Departament de Llenguatges i Sistemes Informàtics
Universitat Jaume I

Análisis del error en redes neuronales: Corrección de los datos y distribuciones no balanceadas

TESIS DOCTORAL

Presentada por:
Roberto Alejo Eleuterio
Dirigida por:
Dr. José Martínez Sotoca

Castellón, Julio de 2010

A mi madre ...

Agradecimientos

Agradezco a mis padres Pablo y Rosa María por el apoyo incondicional que me han brindado siempre, de igual modo a mis hermanos y hermanas que todo el tiempo han estado conmigo.

Así mismo, quiero expresar mi agradecimiento a mis amigos y compañeros Patricio, Jerónimo, Mónica, Marcos, Mauro, Rosa María y Vicente por su amistad y comprensión durante de esta etapa de mi vida.

También quiero agradecer a mi director de tesis el Dr. José Martínez Sotoca por su paciencia en la elaboración de este trabajo de investigación. Así mismo, a los integrantes de la sección Pattern Analysis and Learning Group del grupo de Visión por Ordenador de la Universidad Jaime I, en especial al Dr. José Salvador Sánchez Garreta por el enorme apoyo recibido de él durante mi estancia en la Universidad Jaime I.

Aprovecho para dar las gracias al Dr. Ramón A. Mollineda por su valiosa ayuda en la terminación de este trabajo de investigación y al Dr. J. Miguel Sanchiz.

Finalmente, agradezco a las diferentes instituciones que subvencionaron el desarrollo de este trabajo de investigación: CONACyT, COMECyT, CICYT y la Universidad Jaime I.

Resumen

En los últimos años el desbalance de las clases se ha reconocido como un problema crucial en áreas como el aprendizaje automático y la minería de datos. Este tipo de problema genera una perdida de efectividad del clasificador, porque generalmente asume que los datos de entrada siguen una distribución relativamente balanceada.

El problema del desbalance de las clases aparece cuando existen muchos más elementos de una o algunas clases, que de la otra u otras clases (dos o múltiples clases). Esta desproporción en el tamaño de las diferentes clases en un mismo conjunto de datos, puede ocasionar una disminución en la efectividad del clasificación sobre las clases menos representadas.

En el caso específico de las redes neuronales artificiales, el desbalance de las clases ocasiona lentitud en la convergencia de las clases minoritarias, lo que se traduce en una pobre capacidad de generalización del clasificador.

En este trabajo se estudia el problema del desbalance de las clases en el ámbito de las redes neuronales artificiales. Para ello se entrena la red con el algoritmo back-propagation con procesamiento por grupos desde tres enfoques distintos.

1. Inclusión de funciones de coste al proceso de entrenamiento para disminuir los efectos del desbalance de las clases.

2. Descomposición del problema para simplificar el tratamiento del desbalance de las clases a través del uso de redes neuronales modulares.

3. Reducción de la región de solapamiento de las clases menos representadas a partir de técnicas de corrección de los datos, para mejorar la efectividad del clasificador sobre estas clases.

En síntesis, este trabajo presenta un estudio empírico comparativo de los efectos y posibles tratamientos del problema del desbalance de las clases sobre tres modelos de red neuronal artificial.

Abstract

In recent years the class imbalance problem has been recognized how a crucial problem in areas such as machine learning and data mining. This kind of problems result in loss of performance of the classifier, because it usually assumes that the input data follow a relatively balanced distribution.

The problem of class imbalance appears when there are many more elements from one or several class have more elements than the other or other classes (two or multiple classes). This disproportion of the classes size in a same data set, can diminish the classification accuracy about the classes less represented.

In the specific case of the artificial neuronal networks, the class imbalance causes slow convergence of the minority classes, resulting in a poor generalization capacity of the classifier.

In this work the class imbalance problem in the context of artificial neuronal networks is studied. For that, the network is trained with back-propagation algorithm in batch mode using three different approaches.

1. Cost functions are included in the training process to lessen the class imbalance effects.

2. Decomposition of the problem to simplify the handling of class imbalance though the use of modular neuronal networks.

3. Reducing the overlapping region of minority classes with correction data techniques, to improve the accuracy of these classes.

Summarizing, this work presents a comparative empirical study of the effects and a possible treatments of the class imbalance problem, using three neural network models.

Índice general

Lista de Figuras

Lista de Tablas

Lista de Símbolos

ANN	Artificial Neural Network (Red Neuronal Artificial)
ART	Adaptative Resonance Theory (Teoría de Resonancia Adaptativa)
ART-Map	Adaptative Resonance Theory - Map (El término Map, hace referencia al mapa asociativo incluido para la transformación del espacio)
EW	Edición de Wilson
EWP	EW Ponderada
EW$^+$	EW sobre la clase positiva o minoritaria
EW$^-$	EW sobre la clase negativa o mayoritaria
EWP$^+$	EW Ponderada sobre la clase positiva o minoritaria
EWP$^-$	EW Ponderada sobre la clase negativa o mayoritaria
F1	Criterio de Fisher
g-mean	Media geométrica
RBF	Radial Basis Function (Función de Base Radial)
RBF+VF	Radial Basis Function + Vector Functional (Función de Base Radial + el Vector Funcional de Pao)
LVQ	Learning Vector Quantization (Aprendizaje por cuantización vectorial)
ME	Muestra de Entrenamiento

MLP	Multilayer Perceptron (Perceptron Multicapa)
MSE	Mean Square Error (Error Cuadrático Medio)
NN	Nearest Neighbour (Vecino más cercano)
PC	Precisión en la clasificación
$\mathbf{PC}^+$	PC de la clase positiva o minoritaria
$\mathbf{PC}^-$	PC de la clase negativa o mayoritaria
PDMC	Problemas Desbalanceados de Múltiples Clases
SOM	Self Organizing Maps (Mapas Autoorganizados)
$\mathbf{x}$	Muestra o vector de datos de entrada
$\mathbf{w}$	Vector de pesos
σ	Desviación estándar o ancho de banda de la RBF
η	Razón de aprendizaje
μ	Momento
$\|\cdot\|$	Norma Euclidiana

Capítulo 1

Introducción

Contenido

1.1 Redes Neuronales Artificiales

Una Red Neuronal Artificial (Artificial Neural Network, ANN) es un modelo matemático que trata de emular a los sistemas neuronales biológicos en el procesamiento de la información [Lippmann 1988]. Actualmente, gozan de gran popularidad entre teóricos y especialistas en el área de aprendizaje automático, minería de datos y reconocimiento de formas.

Una de las principales razones del auge de las ANNs es que están dotadas de algoritmos que les permiten aprender a partir de ejemplos o datos de entrada [Russell 1996], por lo que no precisan para su funcionamiento de un modelo a priori del problema a tratar. No obstante, en algunas de ellas se asume que los datos de entrada siguen una distribución de probabilidad específica.

Al día de hoy existen numerosos modelos de ANNs entre los que se encuentran las ANNs basadas en funcion de base radial (Radial Basis Function, RBF) y el perceptron multicapa (Multilayer Perceptron, MLP). Ambos modelos de ANNs son considerados de propagación hacia adelante (feedforward) y comparten varias características en común [Hutchinson 1995]. Por ejemplo, son aproximadores universales

[Haykin 1999], modelos con capas no lineales [Looney 1997] o pueden ser entrenados con métodos similares de descenso por gradiente [Schwenker 2001]. Sin embargo, presentan notables diferencias [Ding 2004].

La principal diferencia entre las redes RBF y el MLP está en la función de activación de los nodos ocultos [Jain 1996]. En las redes RBF esta función depende de la distancia entre los vectores de entrada y los centros de la red, mientras que en la red MLP depende directamente del producto del vector de entrada y el vector de pesos. En otras palabras, en las redes RBF se asume que los datos de entrada siguen alguna distribución de probabilidad particular[1] mientras que en el MLP no se presenta esta característica.

Las redes MLP y RBF han sido ampliamente utilizadas en tareas de clasificación, aproximación de funciones, modelado y problemas de control [Ding 2004]. No obstante, aún se conoce poco de estos modelos lo que se traduce en debilidades tales como la lentitud en el aprendizaje y la pobre capacidad de generalización que se observa en un número importante de aplicaciones prácticas [Barandela 2001].

El incremento de la rapidez del procedimiento de entrenamiento y la mejora de la precisión en el resultado, han motivado un esfuerzo importante en la investigación de criterios más adecuados para definir parámetros y algoritmos vinculados al proceso de aprendizaje.

En [Anand 1993] se propone una modificación al algoritmo back-propagation para acelerar la convergencia del MLP en problemas de dos clases, y en [Anand 1995] se extiende el estudio a problemas de múltiples clases. Otras alternativas para acelerar la convergencia de la red se han presentado en [Jacobs 1988, Fahlman 1988]. En [Lippmann 1988, Cybenko 1989, Funahashi 1989] se ha estudiado la capacidad de representación del MLP en relación al número de capas ocultas presentes, mientras que en [Pao 1989, Kanellopoulos 1997] se ha discutido la estimación del número óptimo de nodos para las capas ocultas.

Por su parte, las redes RBF han sido estudiadas desde diferentes enfoques. Se ha priorizado en el estudio de la configuración de la capa oculta [Uykan 2000] o en su construcción, a partir de otros modelos como por ejemplo las máquinas de soporte vectorial (Support Vector Machines, SVM) [Schölkopf 1997], o los árboles de decisión [Kubat 1998]. También las redes RBF se han estudiado como casos especiales del modelo *Alternative mixture of experts* [Xu 1995, Xu 1998]. Otra tendencia ha sido el uso de algoritmos genéticos para la optimización de los parámetros de la red [Harpham 2004].

A pesar de los numerosos estudios dirigidos a mejorar la efectividad de las ANNs, en la actualidad existen cuestiones que aún no han sido resueltas del todo como el problema ocasionado por distribuciones de clases no balanceadas [Zhou 2006]. Este

[1] Por lo general una distribución normal.

problema aparece cuando existen muchos más elementos de una clase que de la otra. Por ejemplo, en la detección de fraudes en llamadas telefónicas [Fawcett 1997], el diagnóstico de enfermedades raras [Newman 1998] o en la identificación de piezas defectuosas [Murphey 2004].

Las distribuciones de clases no balanceadas no son un factor exclusivo en la resolución de problemas de dos clases, sino que también aparecen con frecuencia en dominios de múltiples clases [Serpico 1993]. Sin embargo, la mayor parte de las investigaciones han sido dirigidas a problemas de dos clases [Zhou 2006].

1.2 El problema del desbalance de las clases

Generalmente, los clasificadores son diseñados para trabajar con bases de datos con clases relativamente balanceadas [Japkowicz 2002], es decir, sobre conjuntos de datos donde no hay diferencia significativa en el número de elementos de las diferentes clases. Sin embargo, en muchas aplicaciones la desproporción del número de muestras entre clases es considerable [Kotsiantis 2003].

Investigaciones recientes reconocen al problema del desbalance de las clases como un factor crítico en el diseño, construcción y entrenamiento del clasificador ANN [Zhou 2006].

Estudios realizados a las ANNs han afirmado que el problema del desbalance de las clases ocasiona que las clases menos representadas tengan una menor participación en el proceso de entrenamiento, lo que se ve reflejado en una disminución de la capacidad de generalización de la red, o en un incremento del tiempo necesario para la convergencia del método [Anand 1993, Bruzzone 1997a, Lu 1998, Murphey 2004].

Básicamente, el problema del desbalance de las clases ocasiona lentitud en la convergencia de la red y reduce la capacidad de generalización de ésta, por lo que una tarea prioritaria es minimizar los efectos del desbalance de las clases sobre las ANNs.

1.3 Objetivos de la tesis

Una de las principales razones que contribuye al bajo rendimiento de una ANN es la falta de representatividad de los datos de entrenamiento. Además, puede ocurrir que existan importantes desproporciones en el número de muestras de las distintas clases (problema del desbalance de las clases) [Murphey 2004], aparición de solapamiento entre clases [Alejo 2006] y ruido[2] en los datos de entrenamiento, o muestras atípicas[3]

[2]Datos con errores originados en su medición o registro.

[3]Excepciones a la regla o elementos que no encajan en un tipo o modelo.

[Barandela 2001].

En este trabajo, se estudia el problema del desbalance de las clases (en dominios de dos y múltiples clases), y se evaluan distintas alternativas para reducir su influencia y efectos sobre el algoritmo back-propagation (con procesamiento por grupos) aplicado a tres arquitecturas distintas de ANN.

En particular, esta investigación se centra en el análisis y evaluación de las siguientes estrategias:

- Inclusión de funciones de coste en el algoritmo de aprendizaje para compensar el desbalance de las clases.
- Empleo de redes neuronales modulares para el tratamiento de problemas de múltiples clases desbalanceados. El objetivo es descomponer los problemas de múltiples clases en subproblemas de dos clases y de esta forma simplificar la resolución del desbalance de las clases.
- Aplicación de técnicas de corrección de los datos para reducir el área de confusión de las clases menos representadas.

El objetivo final de estas estrategias es tratar el problema del desbalance de las clases para mejorar la capacidad de generalización de la red y acelerar su proceso de convergencia.

1.4 Descripción de los datos

En los últimos años los conjuntos de datos almacenados en el depósito de la Universidad de California (UCI Database Repository) [Newman 1998] han sido ampliamente utilizados en diversos estudios sobre aprendizaje automático, minería de datos y reconocimiento de formas. Así mismo, han servido como punto de referencia para distintas investigaciones.

Para evaluar las posibilidades de las estrategias estudiadas en esta investigación se tomaron once conjuntos de datos (de dos y múltiples clases) de la UCI Database Repository [Newman 1998]. Corresponden a diferentes dominios en cuanto a tamaño, dimensionalidad y complejidad.

Adicionalmente, se utilizaron dos bases de datos de múltiples clases (Cayo y Feltwell) relacionadas con imágenes de percepción remota. Estas bases de datos han sido estudiadas en el contexto del problema del desbalance entre clases en dominios de múltiples clases.

En esta sección se comenta brevemente algunas de las principales características de los conjuntos de datos empleados en este trabajo.

Problemas de dos clases

- **Cancer**. Consiste en distinguir entre cáncer de mama maligno o benigno. Las características son obtenidas a partir de una imagen digitalizada de una aspiración con aguja fina (Fine Needle Aspiration, FNA) de un tumor mamario. Estas características describen los núcleos de las células presentes en la imagen. La base de datos Cancer está distribuida en dos clases (cáncer benigno y maligno) con 445 y 238 muestras respectivamente, y cada muestra está representada por 9 características.

- **Pima Indian Diabetes (Diabetes)**. En este conjunto de datos se debe identificar si las pacientes son diabéticas o no de acuerdo a los criterios establecidos por la Organización Mundial de la Salud. Esta base de datos está distribuida en dos clases con 500 pacientes no diabéticos y 268 pacientes diabéticos. Cada muestra está representada por un vector de 8 dimensiones.

- **German Credit (German)**. Esta base de datos sirve para identificar potenciales clientes con bajo o alto riesgo crediticio en función de su solvencia económica. Incluye características como la edad, el estado de la cuenta de ahorro del cliente, tipo de automóvil, empleo, entre otras. En [Newman 1998] están disponibles dos conjuntos de datos de German Credit. Uno con datos mezclados (categóricos y numéricos) y otro con datos numéricos únicamente. En este trabajo se hace referencia a la base de datos numérica que cuenta con 700 clientes buenos y 300 clientes malos identificados por 24 características.

- **Ionosphere**. El objetivo es reconocer electrones libres en la ionosfera como resultado de mediciones que identifican alguna estructura en la ionosfera (mediciones buenas) y aquellas que no (mediciones malas). Está distribuida en 225 mediciones buenas y 126 mediciones malas representadas por 34 características.

- **Liver**. Este conjunto de datos es conocido como BUPA Liver Disorders. Está relacionado a problemas de hígado ocasionados por el consumo excesivo de alcohol. Fue generada por BUPA Medical Research Company. Incluye 345 muestras representadas por seis atributos y dos clases (con 200 y 145 muestras). Cada ejemplo corresponde a una muestra tomada a un hombre soltero.

- **Phoneme**. El conjunto de datos de Phoneme proviene del proyecto ELENA [Guérin-Dugué 1995]. Contiene vocales procedentes de 1809 sílabas aisladas. Cinco atributos caracterizan a cada vocal. El objetivo de esta base de datos es distinguir entre vocales de la clase nasal y la clase oral (3818 y 1586 muestras, respectivamente).

- **Sonar**. Se trata de discernir entre señales de sonar rebotadas en rocas (97) y de las rebotadas de cilindros metálicos (111). Ambas señales son obtenidas desde distintos ángulos y condiciones. Cada muestra es un vector de señales de 60 características en el rango de [0, 1] y representan la energía para una banda de frecuencia integrada durante un determinado lapso de tiempo.

- **Balance**. Este conjunto de datos se generó a partir de los resultados experimentales de un modelo psicológico. Cada ejemplo es clasificado como el extremo de una balanza (derecha, izquierda) o un punto de equilibrio. Los atributos son el peso a la izquierda, la distancia a la izquierda, el peso de la derecha, y la distancia derecha. La forma correcta de encontrar la clase es el mayor de (distancia-izquierda $*$ peso-izquierda) y (distancia-derecha $*$ peso-derecha). Si son iguales está equilibrada. Cada clase está compuesta de 49 (equilibrada), 288 (izquierda) y 288 (derecha) muestras.

- **Vowel**. Este problema consiste en la distinción de los 11 fonemas vocales del inglés (clases). Los datos contienen información de la pronunciación de 15 locutores (8 hombres y 7 mujeres) pronunciando seis veces cada fonema lo que hace un total de $11 * 15 * 6 = 990$ ejemplos. La señal de voz se procesó mediante un filtro de paso bajo y se digitalizaron a 12 bits con una frecuencia de muestreo de 10kHz. Un análisis posterior produjo los 10 atributos (que identifican a cada muestra o ejemplo) a partir de los coeficientes de reflexión.

Los conjuntos de datos Balance y Vowel fueron colocados en la sección de problemas de dos clases porque en este trabajo son tratados como bases de datos de dos clases. En el capítulo 3 se detalla el preprocesamiento realizado a estas bases de datos para transformarlas a problemas de dos clases.

Problemas de múltiples clases

- **Cayo**. Este conjunto de datos representa un cayo de una región del golfo de México. Está distribuida en 11 clases y tiene un total de 6020 muestras etiquetadas y cuatro características. Las clases están distribuidas de la siguiente forma: Nubes (838 muestras), Sombras (293 muestras), Bosque (624 muestras), Yanal (322 muestras), Caminos (133 muestras), Arenas (369 muestras), Mangle (324 muestras), Pantanos (722 muestras), MarNorte(789 muestras), MarSurCercano (833 muestras) y MarSurLejano (772 muestras).

- **Ecoli6**. Fue obtenida a partir de E.coli [Newman 1998]. Esta última es una base de datos biológica generada por el *Institute of Molecular and Cellular*

Biology[4] de la universidad de Osaka, Japón. Está distribuida en 8 clases. y en este trabajo por consideraciones prácticas se eliminaron las clases 7 y 8 que corresponden a las clases *imL* (inner membrane lipoprotein) y *imS* (inner membrane, cleavable signal sequence), dado que cada clase sólo disponen de dos muestras, y este hecho dificulta la aplicación eficiente del método de validación cruzada.

- **Feltwell**. Hace referencia a una sección (de 250 x 350 píxeles) de una imagen de percepción remota de una zona reservada para la agricultura cercana a la villa de Feltwell (Reino Unido) [Serpico 1993]. Para cada píxel se seleccionaron 15 características en función de un criterio heurístico y un análisis de correlación. Se seleccionaron 10944 píxeles que pertenecen a cinco clases relacionadas a la agricultura: remolacha de azúcar (3531 píxeles), hojarasca (2441 píxeles), suelo (896 píxeles), patatas (2295 píxeles) y zanahorias (1781 píxeles). Para más detalle véase [Roli 1996].

- **Satimage**. Fue extraída de [Newman 1998] y corresponde a una imagen de percepción remota. Cada una de las 6435 muestras en esta base de datos está compuesta por cuatro bandas espectrales de la misma escena. El vector de características corresponde a una región cuadrada de 3x3 píxeles. Por lo tanto, las 36 características representan a cada uno de los nueve píxeles en cada una de las cuatro imágenes espectrales. Está dividida en 6 clases con 1533, 703, 1358, 626, 707, y 1508 muestras respectivamente.

1.5 Estructura del documento

Este trabajo ha sido organizado en diferentes capítulos que tratan los principales temas de interés para esta investigación. A continuación, se presenta un bosquejo general de la estructura de este documento.

- **Capítulo 2:** Las bases teóricas fundamentales para el estudio de las redes neuronales artificiales son discutidas en este capítulo. Se hace una descripción de los avances que han sufrido las ANNs, así como de las principales estructuras conexionistas e investigaciones más relevantes en este campo.

- **Capítulo 3:** En este capítulo se analiza el impacto y efecto del problema del desbalance de las clases en las ANNs entrenadas con el algoritmo back-propagation con procesamiento por grupos. El problema del desbalance entre

[4]Información detallada sobre la distribución de los datos, precisión de clasificación y características específicas de la base de datos E.coli se puede encontrar en [Nakai 1991] y [Horton].

clases es estudiado en dominios de dos y múltiples clases. Así mismo, se proponen y evalúan algunas estrategias basadas en funciones de coste para tratar de reducir sus efectos.

- **Capítulo 4:** Las ANNs modulares son una tendencia en el diseño de la ANN. Están basadas en el principio de *divide y vencerás*. En este capítulo se estudia este tipo de redes para tratar de resolver problemas desbalanceados de múltiples clases dividiendolos en subproblemas de dos clases, para así tratar el problema del desbalance con métodos efectivos para dos clases y de forma independiente.

- **Capítulo 5:** En los últimos años ha resurgido el interés por examinar la calidad de la Muestra de Entrenamiento (ME) y la validez de sus elementos. En este capítulo se estudian y adaptan técnicas tomadas del contexto de la regla del vecino más próximo para tratar de eliminar situaciones imperfectamente supervisadas, elementos atípicos y muestras en la zona de solapamiento, con la finalidad de mejorar la efectividad de las ANNs entrenadas con bases de datos desequilibradas. Se busca minimizar el área de confusión de la clase minoritaria y de esta forma reducir los efectos del desbalance.

- **Capítulo 6:** Este capítulo está dedicado a resumir las principales conclusiones obtenidas a lo largo de la investigación, así como las aportaciones más relevantes. Se comentan brevemente los diferentes enfoques utilizados para reducir los efectos del desbalance en las fases de entrenamiento y clasificación de las redes MLP, RBF y RBF+VF. Para finalizar se discuten algunas de las líneas de mayor interés para futuras investigaciones.

En síntesis, esta tesis presenta un estudio empírico comparativo de los efectos y posibles tratamientos del problema del desbalance de las clases (en dominios de dos y múltiples clases) sobre tres modelos de ANNs, entrenadas con el algoritmo back-propagation con procesamiento por grupos.

Capítulo 2

Redes Neuronales Artificiales

Contenido

2.1 Inspiración biológica

La idea que impulsó el desarrollo de las Redes Neuronales Artificiales (Artificial Neural Networks, ANNs) ha sido la de imitar al sistema de procesamiento más complejo que se conoce hasta ahora, el cerebro humano. El cerebro está formado por millones neuronas organizadas en capas [Hilera 1995]. Las neuronas son un tipo especial de células nerviosas que procesan información. Están compuestas de distintas partes: Cuerpo o soma, axón y dendritas [Jain 1996].

Las neuronas se comunican unas con otras y con el exterior sin tocarse a través de unos espacios de intercomunicación llamados sinapsis [Russell 1996]. La comunicación entre neuronas tiene lugar como resultado de la liberación de unas sustancias

electro-químicas llamadas neurotransmisores. Las dendritas reciben estas señales eléctricas (que pueden ser excitadoras o inhibidoras) mediante iones provenientes de otras neuronas, y se acumulan en el cuerpo o soma de la neurona hasta alcanzar un cierto valor umbral[1].

En ese momento se genera un impulso que consiste en liberar neurotransmisores que a través del axón llegarán a las dendritas de otras neuronas (ver Fig. 2.1). En el caso de las neuronas receptoras su axón envía señales hacia el cerebro, y en el de las neuronas motoras manda señales desde el cerebro hacia el resto del cuerpo[2].

La complejidad del cerebro humano es extraordinaria debido al enorme número de neuronas y conexiones existentes entre ellas. Se estima que el cerebro humano contiene más de cien mil millones de neuronas y de 1.000 a 10.000 sinapsis por neurona [Chudler 2006].

Es importante notar que aunque el tiempo de conmutación de la neurona es mucho menor que los elementos de las computadoras [Jain 1996], el cerebro tiene la capacidad de realizar tareas miles de veces más rápidas que las actuales supercomputadoras. Esta versatilidad es utilizada por el cerebro humano a través de la información que recibe por medio de los sentidos para interpretar su entorno: reconocimiento de rostros, comunicación mediante lenguaje natural, etc.

2.2 Bosquejo histórico

En la antigüedad Platón (427-347 a.C) y Aristóteles (384-422 a.C) dieron a conocer las primeras explicaciones teóricas sobre el cerebro y el pensamiento. Las mismas ideas sobre el proceso mental las mantuvieron Descartes (1596-1650) y los filósofos empiristas del siglo XVIII [Hilera 1995]. Sin embargo, la historia del conocimiento de las neuronas comenzó con el científico aragonés Santiago Ramón y Cajal descubridor de la estructura neuronal del sistema nervioso [Martín 2001].

Más adelante, en 1943 el neurofisiólogo Warren McCulloch y el matemático Walter Pitts [McCulloch 1943] conciben una teoría acerca de la forma de trabajar de las neuronas artificiales. En 1949 Donald Hebb [Hebb 1949] publica el libro “The organization of behavior - a neurophysiological theory" en el que se establece la conexión entre la Fisiología y la Psicología. Propone una ley de aprendizaje que le permite explicar cualitativamente algunos ejemplos experimentales de carácter psicológico.

En 1958 Rosenblatt [Rosenblatt 1958] publica “The Perceptron: a probabilistic model for information storage & organization in the brain" donde se aborda uno

[1] Valor mínimo de una magnitud a partir del cual se produce un potencial de acción.

[2] La descripción que se presenta en esta sección es una visión general simplificada; la neurobiología es un tema mucho más complejo de lo que se esboza en este apartado. Información más detallada se puede encontrar en [Brunak 1990].

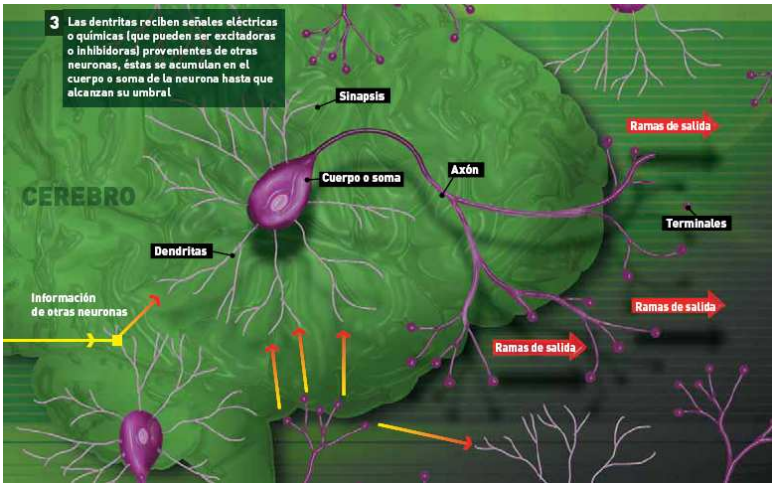


Fig. 2.1: Esquema simplificado de una neurona biológica. Imagen tomada de [Gúzman 2004]

de los modelos neuronales de mayor impacto en los inicios de la neurocomputación. Rosenblatt opinaba que la herramienta de análisis más apropiada era la teoría de probabilidades. Esta teoría se basa en la separabilidad estadística para caracterizar las propiedades más visibles de estas redes de interconexión ligeramente aleatorias.

En 1959 Bernard Widrow y Marcial Hoff [Widrow 1960] desarrollan el modelo ADALINE (ADAptative LINear Elements) equipado con un potente algoritmo de aprendizaje. Es la primera red neuronal aplicada a un problema real[3].

Con la publicación de "Perceptrons: An Introduction to Computational Geometry"[4] por Marvin Minsky y Seymour Papert del Instituto de Tecnología de Massachussets, surgen numerosas críticas que frenaron hasta 1982 el crecimiento que estaban experimentando las investigaciones sobre ANNs. A pesar de esto, algunos investigadores continuaron avanzando en este campo del conocimiento.

James Anderson plantea un modelo lineal llamado Asociador Lineal y en 1977 diseña una potente extensión a este modelo llamado Brain-State-in-a-Box (BSB) [Anderson 1977]. Kunihiko Fukushima [Fukushima 1979] desarrolla el Neocognitron, un modelo de red neuronal para el reconocimiento de patrones visuales. Teuvo Kohonen [Kohonen 1977] propone un modelo similar al propuesto por Anderson de

[3]Fue usada comercialmente durante varias décadas como filtros adaptativos para eliminar ecos en líneas telefónicas.

[4]En este trabajo se presentan las limitaciones del Perceptron de Rosenblatt. Principalmente la imposibilidad de este modelo de resolver problemas no linealmente separables.

forma independiente.

En 1982 coinciden numerosos eventos que hacen resurgir el interés por las ANNs. John Hopfield [Hopfield 1982] presenta su trabajo sobre ANN en la Academia Nacional de Ciencias de los Estados Unidos de America[5]. En este trabajo escribe con claridad y rigor matemático una red a la que se ha dado su nombre, pero además, muestra cómo tales redes pueden trabajar de forma óptima.

En 1987 Stephen Grossberg [Grossberg 1987] aporta importantes innovaciones con su modelo ART (Adaptative Resonance Theory) y junto a Michel Cohen elabora un importante teorema sobre la estabilidad de las ANNs recurrentes en términos de una función de energía. La publicación de "PDP Books"[6] editados por David Rumelhart y James Mc Clelland supone un verdadero acontecimiento por la presentación del método de retropropagación ("back-propagation") [Rumelhart 1986].

Por otra parte, en 1982 se celebra la U.S.-Japan Joint Conference on Cooperative/Competitive Neuronal Networks, y la compañía Fujitsu comienza el desarrollo de computadoras con capacidad de aprendizaje para aplicaciones en robótica.

En 1986 el American Institute of Physics (AIP) inicia lo que ha sido la reunión anual Neural Networks for Computing. En 1987 el IEEE celebra la primera conferencia internacional sobre ANN. Ese mismo año se forma la International Neural Networks Society (INNS). En 1988 surge la International Joint Conference on Neural Networks (IJCNN) como resultado de la unión entre la IEEE y la INNS.

La alternativa europea es la International Conference on Artificial Neural Networks (ICANN) surgida en septiembre de 1991. También merece una referencia aparte la reunión anual Neural Information Processing System (NIPS) celebrada en Denver (Colorado) desde 1987.

2.3 Neurona artificial

El modelo de neurona artificial más conocido es el de McCulloch-Pitts [McCulloch 1943]. La teoría de McCulloch-Pitts se basa en cinco suposiciones [Freeman 1991]:

1. La actividad de una neurona es un proceso todo-nada.

2. Es preciso que un número fijo de sinapsis (> 1) sean excitadas dentro de un periodo de adición latente para que se excite una neurona.

3. El único retraso significativo dentro de un sistema nervioso es el retardo sináptico.

[5] National Academy of Sciences of the United States of America.
[6] Parallel Distributed Processing, Vol. I and II.

4. La actividad de cualquier sinapsis inhibitoria impide por completo la excitación de la neurona en ese momento.

5. La estructura de la red de interconexiones no cambia con el transcurso del tiempo.

En la Fig. 2.2 se ilustra el modelo de neurona artificial de McCulloch-Pitts[7].

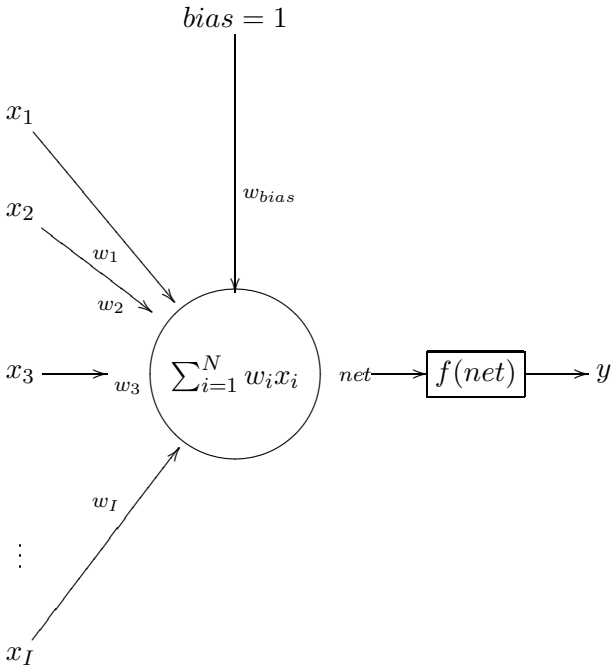


Fig. 2.2: Neurona artificial según el modelo de McCulloch-Pitts.

En este modelo se identifican cinco elementos básicos:

1. Conjunto de x_i señales de entrada, que suministran a la red la información del entorno ($i = 1, ..., I$).

2. Vector de pesos sinápticos $\mathbf{w}$ (en analogía con la sinapsis biológica, Fig. 2.1). El peso w_i está asociado a la sinapsis que conecta a la i-ésima entrada con la neurona.

[7] El modelo de McCulloch-Pitts no contaba con un mecanismo de aprendizaje y el objetivo que motivo su desarrollo fue la representación de funciones boolenas básicas [Russell 1996]. Actualmente, sigue siendo la base de los elementos constitutivos de las ANNs.

3. Umbral o bias. Aumenta la capacidad de procesamiento de la neurona y eleva o reduce la entrada a ésta según sea su valor positivo o negativo.

4. Sumador o integrador (net) que combina linealmente las señales de entrada ponderadas con sus respectivos pesos (w_i) y el umbral.

5. Función de activación $f(net)$ que suele limitar la amplitud de la salida de la neurona (y).

La función de activación es la que define en última instancia la salida de la neurona. En la Tabla 2.1 se muestra la analogía entre las neuronas biológicas y las artificiales que se planteó en el modelo de McCulloch-Pitts.

Tabla 2.1: Analogía entre las neuronas biológicas y las artificiales.

Neurona biológica	Neurona artificial
Sinapsis	Pesos sinápticos($\mathbf{w}$)
Axón	Respuesta de la neurona y (función de activación $f(net)$)
Dendritas	Señales de entrada ($\mathbf{x}$)
Soma	Sumador o integrador (net)

El modelo de McCulloch-Pitts asume varias suposiciones que no reflejan el verdadero comportamiento de las neuronas biológicas [Jain 1996]. Sin embargo, dicho modelo ha sido generalizado de diferentes formas. Actualmente, la función de activación puede admitir distintos modelos. Entre las más comunes podemos destacar:

- Función lineal. Se utiliza cuando no se desea acotar la salida de la neurona.

$$f(net) = c \cdot net \ , \tag{2.1}$$

  donde c es un valor constante.

- Función escalón. Adopta la forma

$$f(net) = \begin{cases} +1 & \text{si } net > 0 \\ -1 & \text{si } net < 0 \end{cases} \tag{2.2}$$

  y proporciona una salida bivaluada.

- Función sigmoidea. Es habitual en muchos modelos neuronales y provoca una transformación no lineal de su argumento. Una de las funciones más utilizadas es la función logística definida por

$$f(net) = \frac{1}{1 + exp(-a \cdot net + b)}\,, a > 0\,, \tag{2.3}$$

  donde a es el parámetro de inclinación que ajusta la pendiente de la función y b el sesgo o bias.

- Función Gaussiana. Su uso es común cuando se requiere de una transformación no lineal del espacio de entrada a otro con mayor dimensionalidad. Se puede expresar como:

$$f(net) = c \cdot exp\left(\frac{-net^2}{2\sigma^2}\right)\,, \tag{2.4}$$

  donde c es una constante y σ^2 la varianza o amplitud de la función.

2.4 Red Neuronal Artificial

Una red neuronal artificial (Artificial Neural Network, ANN) es un prototipo de procesamiento de información inspirado en la estructura del cerebro humano cuyo objetivo es reproducir sus mecanismos de aprendizaje. Esta conformada por un gran número de neuronas artificiales interconectadas en base a un modelo que asigna una respuesta a cada componente, donde lo que se pretende emular no es la estructura biológica sino su funcionamiento.

A lo largo del tiempo se han presentado numerosas definiciones del concepto red neuronal artificial. A continuación se presentan algunas de la más relevantes.

- En el DARPA Neural Network Study [DARPA-USA 1988] se define que una ANN es un sistema compuesto de muchos elementos simples de proceso operando en paralelo y cuya función está determinada por la estructura de la red, los pesos de las conexiones y el procesado realizado en los elementos o nodos de cálculo.

- En [Freeman 1991] se dice que una ANN es un conjunto de procesadores en paralelo conectados entre sí en forma de grafo dirigido. Está organizado de tal modo que la estructura de la red es la adecuada para el problema que se está considerando.

- Zurada [Zurada 1992] establece que los sistemas de ANNs son sistemas celulares físicos que puedan adquirir, almacenar y usar conocimiento empírico.

- Nigrin [Nigrin 1993] define a la ANN como un circuito compuesto de un número elevado de elementos simples de un proceso con una base neurológica. Cada elemento opera con información local. Así, cada elemento opera asíncronamente por lo que no hay un reloj total del sistema.

- Haykin [Haykin 1999] propone que una ANN es un procesador distribuido y con estructura paralela que tiene una tendencia natural a almacenar conocimiento experimental haciéndolo apto para su uso. Se parece al cerebro en dos cosas: 1) el conocimiento es adquirido por la red a través de un proceso de aprendizaje y 2) este conocimiento se almacena en los pesos sinápticos o conexiones entre las neuronas.

Independientemente de la definición a la que se haga referencia en una ANN se pueden identificar las siguientes características:

- Está formada por elementos de procesamiento sencillos.

- Tiene un alto grado de interconexión.

- La comunicación se realiza por mensajes simples escalares.

- La interacción es adaptable entre elementos.

- Presenta una estructura en paralelo o distribuida.

2.4.1 Proceso de aprendizaje

El proceso de aprendizaje o entrenamiento de la ANN consiste en modificar los pesos de las conexiones sistemáticamente para codificar la relación de entrada-salida [Freeman 1991]. En otras palabras, es el mecanismo por el cual los parámetros libres de la ANN son adaptados a través de un procedimiento de estimulación del ambiente en el cual está contenida.

El tipo de aprendizaje está determinado por la manera en la cual el cambio de los parámetros es realizado [Russell 1996]. Este proceso implica la siguiente secuencia de eventos:

- La red es estimulada por su entorno.

- La ANN sufre cambios en sus parámetros como resultado de esta estimulación.

- La ANN responde al entorno de manera diferente por los cambios ocurridos en su estructura interna.

Lo más común es usar la arquitectura y el tipo de aprendizaje como criterios de clasificación de las ANNs (ver Fig. 2.3). Es por ello que un tipo de clasificación que se realiza sobre las ANNs obedece al tipo de aprendizaje utilizado por dichas redes. Así, se pueden distinguir dos tipos de aprendizaje: el supervisado y el no supervisado. La diferencia fundamental entre ambos estriba en la existencia de un agente externo (supervisor) que controle este proceso.

Un tipo especial de aprendizaje es el híbrido que combina el supervisado y el no supervisado. Unas capas de la red utilizan un aprendizaje de naturaleza supervisada mientras que otras capas realizan un aprendizaje de forma no supervisada.

Al conjunto prescrito de reglas definidas para solucionar un problema de adquisición de conocimiento se le llama algoritmo de aprendizaje. No existe un único algoritmo de aprendizaje para el diseño de la red, y cada algoritmo obedece a una regla de aprendizaje específica. Básicamente existen 4 reglas [Jain 1996]:

- Corrección de error: En el paradigma de aprendizaje supervisado se establece un valor deseado (d) como salida de la ANN para cada entrada (x). Durante el proceso de aprendizaje la salida (y) generada por la red suele no ser igual a la salida deseada. El principio básico de la corrección del error es usar el error generado por la red ($d - y$) para modificar las conexiones de los pesos gradualmente y así reducir este error.

- La regla de Boltzmann: Es una regla de aprendizaje estocástica obtenida a partir de principios de la teoría de la información y de la termodinámica. El objetivo del aprendizaje de Boltzmann es ajustar los pesos de conexión de tal forma que el estado de las unidades visibles satisfaga una distribución de probabilidades deseada. Puede ser vista como un caso especial de corrección del error, en el cuál el error no es medido directamente como la diferencia entre la salida deseada y la actual, sino que corresponde a la correlación de las salidas de las neuronas.

- Regla Hebbiana: Es un mecanismo dependiente del tiempo y de ámbito local. Incrementa la eficiencia de una sinapsis en función de la correlación entre las actividades pre y post-sinápticas. Si dos neuronas están simultanea y repetidamente activas, las conexiones entre ellas deben ser fortalecidas y en caso contrario deben ser debilitadas.

- Regla del aprendizaje competitivo: Suele decirse que las neuronas compiten unas contra otras con el fin de llevar a cabo una tarea. Con este tipo de aprendizaje se pretende que cuando se presenta a la red cierta información de entrada, sólo una de las neuronas de salida de la red o un cierto grupo de neuronas sean activadas (hasta alcanzar su valor de respuesta máximo). Por

tanto, las neuronas compiten para activarse quedando finalmente una o un grupo como neuronas vencedoras, y el resto quedan anuladas siendo forzadas a que sus valores de respuesta sean mínimos.

El otro criterio de organización de las ANNs es mediante la arquitectura que disponen (ver Fig. 2.3). Se pueden tener dos posibilidades distintas: a) Si la arquitectura de la red no presenta ciclos[8] la red se llama unidireccional (feedforward), y b) si se puede trazar un camino de una neurona a sí misma entonces este tipo de redes se denominan recurrentes o realimentadas [Russell 1996].

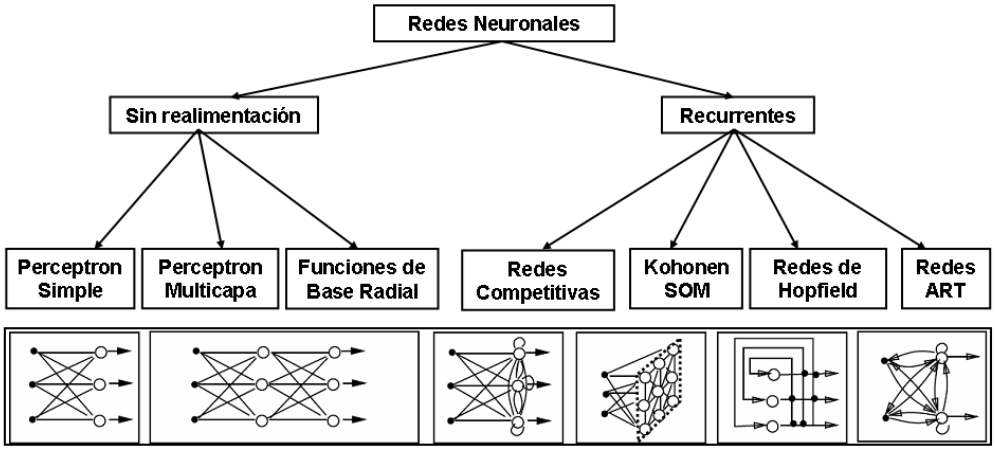


Fig. 2.3: Esquema de las principales arquitecturas de ANN.

2.4.2 Principales estructuras conexionistas

Actualmente, existen una gran cantidad de estructuras conexionistas o ANNs distintas. En este apartado nos limitamos a mostrar algunas de las estructuras más sobresalientes. De [Jain 1996] se toma una visión generalizada de algunas de las características más importantes de las ANNs y se presentan en la Tabla 2.2. Se muestran los tipos de arquitecturas, las reglas de aprendizaje, los algoritmos de entrenamiento y algunas de las aplicaciones en las que se han utilizado estas estructuras conexionistas.

[8] No se puede trazar un camino de una neurona a sí misma.

Algunas aplicaciones de las ANNs

Las aplicaciones prácticas de esta rama de la investigación científica y tecnológica son múltiples y no parecen tener fin. En el terreno de la medicina han apoyado la obtención de diagnósticos con mayor precisión y menor riesgo para los pacientes, además de permitir el análisis de las imágenes de diversos estudios clínicos, y facilitar la elaboración de medicamentos [Papik 1998].

En aplicaciones financieras se examinan diariamente los movimientos del mercado financiero y se realiza la predicción de los índices bursátiles con lo cual se cubre la función de asesoría. También hay programas que apoyan los procesos de transacciones bancarias mediante la detección de fraudes [Chan 1999] no evidentes para otros sistemas de control.

En el tema de la edición de sonido un programa basado en este principio permite discriminar voces o sonidos ambientales sin detrimento de las demás señales de la grabación. Gracias a la aplicación de las redes neuronales también es posible clasificar y buscar imágenes en un catálogo dado, además de predecir cambios en los rasgos faciales o en el comportamiento humano.

Los estudios de nutrición han recibido el apoyo de las redes neuronales artificiales, principalmente para el control de calidad. Otros ejemplos se pueden encontrar en la detección de fraudes en llamadas telefónicas [Fawcett 1997], o en la identificación de productos defectuosos en la línea de ensamblaje para la fabricación de automóviles [Murphey 2004].

Una visión más detallada sobre las ANNs puede encontrarse en la lectura de los trabajos de Lippmann [Lippmann 1988], Jain [Jain 1996], o los libros de Haykin [Haykin 1999] y Looney [Looney 1997].

Tabla 2.2: Algunas características importantes de las principales de las estructuras conexionistas

Paradigma	Regla Aprendizaje	Arquitectura	Algoritmo Entrenamiento	Aplicaciones
Supervisado	Corrección de error	Perceptron mono multicapa	Perceptron Adaline Madaline Back-propagation	Reconocimiento de formas, aproximación de funciones, control
	Boltzman	Recurrente	Algoritmo de Boltzman	Reconocimiento de formas
	Hebbian	Feedforward multicapa	Análisis de discriminación lineal	Análisis de datos, Reconocimiento

Tabla 2.2: Algunas características importantes de las principales de las estructuras conexionistas (continuación).

Paradigma	Regla Aprendizaje	Arquitectura	Algoritmo Entrenamiento	Aplicaciones
				de formas
	Competitiva	Competitiva	Cuantización vectorial	Autoasociación Compresión de datos
		Redes ART	ARTMap	Autoasociación Reconocimiento de formas
No Supervisado	Corrección de error	Feedforward multicapa	Proyección de Sammons	Análisis de datos
	Hebbian	Feedforward o competitiva	Análisis de componentes principales	Análisis y compresión de datos
		Redes de Hopfield	Memoria asociativa	Autoasociación optimización
	Competitiva	Competitiva	Cuantización vectorial	Autoasociación Compresión de datos
		Mapas Autoorganizativos de Kohonen	Kohonen SOM	Categorización, análisis de datos
		Redes ART	ART1, ART2	Categorización
Híbrido	Corrección de error y competitiva	RBF	Back-propagation regresión y técnicas de agrupamiento[9]	Reconocimiento de formas, aproximación de funciones, predicción y control

[9]Las técnicas de agrupamiento tratan de reunir objetos semejantes mediante el uso de criterios de similaridad.

2.5 Perceptron Multicapa

El Perceptron Multicapa (Multilayer Perceptron, MLP) es uno de los modelos de ANNs más conocidos y utilizados en la actualidad [Jain 1996]. Se ha aplicado a tareas de clasificación, predicción, aproximación de funciones y control.

Nace en el campo de la inteligencia artificial como un intento de modelar la estructura neuronal del cerebro y la capacidad de aprendizaje de este sistema biológico [Haykin 1999].

El MLP es una estructura interconexionista compuesta por una o más capas ocultas entre los nodos de entrada y los de salida [Lippmann 1988]. Estas capas están integradas por nodos que pueden o no estar conectados directamente a los de entrada y a los de salida. Se caracteriza fundamentalmente porque no existen conexiones entre neuronas de la misma capa ni conexiones hacia atrás. Cada capa alimenta a todas las neuronas de la capa siguiente.

En el MLP la cantidad de neuronas de entrada se corresponde con la dimensión del vector de entrada ($\mathbf{x}$) y el número de neuronas en la capa de salida con el total de clases en la ME. La cantidad de neuronas y capas ocultas no está predefinida por lo que ha sido objeto de estudio durante muchos años [Haykin 1999]. La Fig. 2.4 muestra un MLP con tres capas: entrada, oculta y salida.

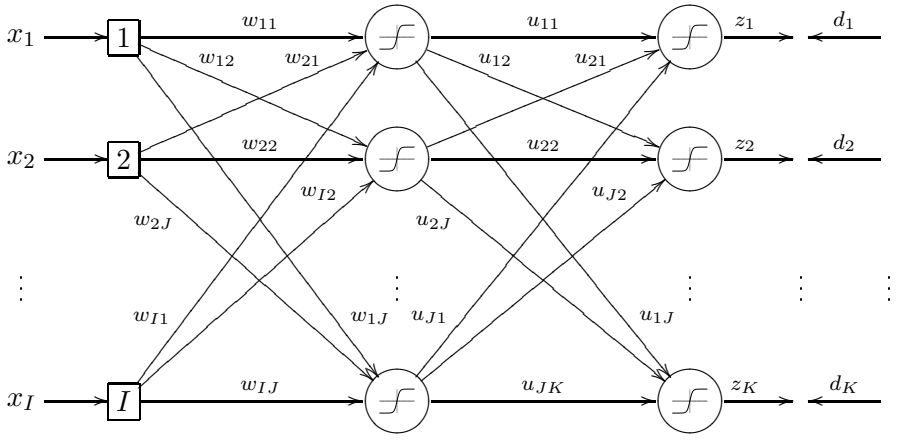


Fig. 2.4: MLP de tres capas con I nodos en la entrada, J neuronas ocultas y K nodos de salida. $\mathbf{z}$ es la salida real de la red y $\mathbf{d}$ la esperada para la entrada $\mathbf{x}$. $\mathbf{W}$ y $\mathbf{U}$ son los pesos de la red para la capa oculta y la de salida respectivamente.

Obsérvese en la Fig. 2.4, que cada neurona proporciona una salida como resultado de la combinación de todas las señales que recibe y su procesamiento por medio de la función de activación $f(\cdot)$.

Un MLP es considerado un prototipo de entrada-salida dado que a cada entrada $\mathbf{x}_n = [x_1, x_2, ..., x_I]$ se produce una salida $\mathbf{z}_n$ donde

$$\mathbf{z}_n = f(\mathbf{y}_n, \mathbf{U}), \quad \text{para} \quad \mathbf{y}_n = f(\mathbf{x}_n, \mathbf{W}), \tag{2.5}$$

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1J} \\ w_{21} & w_{22} & \cdots & w_{2J} \\ \vdots & \vdots & \vdots & \vdots \\ w_{I1} & w_{I2} & \cdots & w_{IJ} \end{pmatrix}, \ \mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1K} \\ u_{21} & u_{22} & \cdots & u_{2K} \\ \vdots & \vdots & \vdots & \vdots \\ u_{J1} & u_{J2} & \cdots & u_{JK} \end{pmatrix},$$

y la función de activación $f(\cdot)$ es definida como

$$f(net) = 1/(1 + exp(-a \cdot net + c)), \tag{2.6}$$

o bien

$$f(net) = a \cdot tanh(b \cdot net + c), \tag{2.7}$$

donde a, b y c son constantes, y net[10] es el sumador o integrador que combina linealmente las señales de entrada de la neurona con sus respectivos pesos. Ambas funciones (Ec. 2.6, Ec. 2.7) cumplen con dos condiciones fundamentales:

- No linealidad: necesaria para superar los inconvenientes de la separabilidad lineal.
- Diferenciabilidad: imprescindible para el uso de algoritmos de entrenamiento o aprendizaje robustos.

El MLP supera muchas de las limitaciones del perceptron de Rosenblatt pero no fue utilizada en el pasado por falta de un algoritmo de entrenamiento efectivo [Lippmann 1988]. Más adelante con la popularización del algoritmo back-propagation [Rumelhart 1986] el MLP se convirtió en una atractiva alternativa para la resolución problemas complejos.

2.5.1 Algoritmo Back-propagation

El proceso de aprendizaje o entrenamiento del MLP consiste en la estimación de sus parámetros libres (pesos de la red). El algoritmo back-propagation descrito (formalmente) en primer lugar por Werbos [Werbos 1974], posteriormente por Parker [Parker 1985] y finalmente por Rumelhart [Rumelhart 1986], es el método de aprendizaje más ampliamente utilizado en el MLP. Esta basado en una técnica de descenso

[10] En el caso de la Fig. 2.4, la variable net puede asumir dos valores dependiendo de la capa en que se encuentre (en la oculta $net_j = \sum_{i=1}^{I} w_{ij} x_i$, o en la salida $net_k = \sum_{j=1}^{J} u_{jk} y_j$).

por gradiente que utiliza la minimización del error cuadrático medio (Mean Square Error, MSE) mediante un proceso iterativo. El MSE es definido como:

$$E(V) = \frac{1}{2N} \sum_{n=1}^{N} \sum_{k=1}^{K} (d_k^n - z_k^n)^2 \,, \tag{2.8}$$

donde $\mathbf{d}^n$ es la salida esperada de la red, $V = \{\mathbf{W}, \mathbf{U}\}$ los parámetros libres de la red, y $\mathbf{z}^n(\cdot)$ la salida real. N es el total de muestras de entrenamiento y K el número de clases.

El método de descenso por gradiente iterativo puede ser formulado como sigue

$$V^{t+1} = V^t + \eta_i \nabla E(V^t) \,, \tag{2.9}$$

donde t es la *t-ésima* iteración y η_i la *razón de aprendizaje* o *learning rate*[11] ($0 < \eta_i \leq 1$). Al pasar de la iteración t a la $t+1$ el algoritmo aplica la corrección

$$\nabla V = V^{t+1} - V^t = \eta_i \nabla E(V^t) \,, \tag{2.10}$$

en la dirección opuesta al vector gradiente $\nabla E(V^t)$.

En términos generales, el algoritmo back-propagation para un MLP de tres capas se puede resumir como sigue:

1. Inicializar aleatoriamente con valores pequeños los pesos de la red. Generalmente con valores entre -0.5 y 0.5.

2. Aleatoriamente elegir una muestra de entrada $\mathbf{x}^{(n)}$.

3. Propagar la señal hacia adelante a través de la red.

$$z_k^{(n)} = g(\sum_{j=1}^{J} u_{jk} h(\sum_{i=1}^{I} w_{ij} x_i^{(n)})) \,, \tag{2.11}$$

   donde $z_k^{(n)}$ es la salida de la red para la entrada $x_i^{(n)}$; $g(\cdot)$ y $h(\cdot)$ representan la función de activación (Ec. 2.6).

4. Calcular δ_k^L para la capa de salida.

$$\delta_k^L = [z_k^{(n)}(1 - z_k^{(n)})](d_k^{(n)} - z_k^{(n)}) \,, \tag{2.12}$$

   donde L es el número de capas ocultas mas 1 (en este caso $L = 2$).

[11] Ésta tiene una enorme influencia en la convergencia del método [Anand 1993, Looney 1997].

5. Calcular las deltas (δ) para las capas previas por propagación del error hacia atrás.

$$\delta_j^l = [y_j^n(1 - y_j^{(n)})] \sum_{k=1}^{K} u_{jk}^{(t)} \delta_k^L, \tag{2.13}$$

para $l = (L-1), ..., 1$, donde $t = (1, ..., T)$ es el número de iteración o repetición del algoritmo.

6. Actualizar los pesos usando

$$u_{jk}^{(t+1)} = u_{jk}^{(t)} + (\eta_i \delta_k^L y_j^{(n)})^{(t)}, \tag{2.14}$$

para la capa de salida y

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + (\eta_i \delta_j^l x_i^{(n)})^{(t)}, \tag{2.15}$$

para la capa oculta.

7. Regresar al paso 2 y repetir para la siguiente muestra hasta alcanzar el mínimo de error (fijado a priori) o hasta alcanzar el número máximo de iteraciones.

En el anexo B se presenta a mayor detalle el desarrollo matemático seguido para la obtención de las reglas de actualización del algoritmo back-propagation.

2.5.2 Razón de aprendizaje y Momento

En el algoritmo back-propagation la elección apropiada de la razón de aprendizaje (η) es un factor crítico para la efectividad de la red [Looney 1997], porque determina la magnitud de las actualizaciones en los pesos. Si η es demasiado pequeña la velocidad de la convergencia es excesivamente lenta y la probabilidad de quedar atrapado en un mínimo local se eleva, mientras que si η es demasiado grande conduce a inestabilidad (oscilaciones) dentro de la función de error con el peligro de pasar por encima del mínimo global [Haykin 1999].

Para ayudar a disminuir las oscilaciones en la función de error de una iteración a otra e incrementar la velocidad de convergencia, en [Rumelhart 1986] se propone la técnica llamada *momento* (μ). Cuando se calcula el valor del cambio de peso $\nabla \mathbf{v}$ se añade una fracción del cambio anterior. Este término adicional tiende a mantener los cambios de peso en la misma dirección: de aquí el término *momento* [Freeman 1991]. La adición del término momento μ da como resultado

$$\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} + \eta \nabla \mathbf{v}^{(t)} + \mu \nabla \mathbf{v}^{(t-1)}, \quad 0 < \mu < 1, \tag{2.16}$$

donde t es la t-ésima iteración y $\mathbf{v}$ el vector de pesos. Los valores más comúnmente usados para el momento son $\mu \approx 0.9$ [Duda 2001].

2.5.3 Arquitectura del MLP

En la estructura del MLP es fundamental la elección del número de capas ocultas y la cantidad de nodos en cada una de ellas. En [Lippmann 1988] se indica que los MLP de una capa oculta son capaces de producir regiones de decisión linealmente separables, las de dos capas pueden generar zonas convexas cerradas o abiertas, y finalmente las de tres capas poseen la capacidad de dividir el espacio de observación en áreas de diversas formas. En [Cybenko 1989] se estudia la capacidad de representación del MLP en relación al número de capas ocultas. Ahora se sabe que un MLP con una capa oculta es capaz de encontrar cualquier relación que pueda ser aproximada por una función continua, y que con dos capas ocultas cualquier relación que implique funciones discontinuas [Russell 1996]. No obstante, se ha demostrado que para la mayoría de los problemas bastará con una capa oculta [Funahashi 1989].

Otro problema relacionado a la topología del MLP es la determinación del número óptimo de nodos por capa oculta. A diferencia de la elección del número de capas ocultas, la identificación del número de neuronas ocultas necesarias para resolver un problema es aún más complicado. Al día de hoy no existe ninguna solución sistematizada que sea completamente aceptada para realizar esta labor. Sin embargo, algunos métodos se han desarrollado con este propósito. En la tabla 2.3 se muestran algunas de estas estrategias.

La importancia de obtener el número idóneo de neuronas ocultas descansa en el hecho de que determinan la capacidad de representación de la red y la complejidad de la frontera de decisión [Duda 2001].

En la actualidad, el proceso más común para la estimación del número de elementos en la capa oculta es realizado mediante prueba y error. En otras palabras, el investigador usando su experiencia fija el número de nodos para cada capa oculta.

2.5.4 Heurísticas para mejorar el rendimiento del algoritmo back-propagation

A menudo se dice que el diseño de una red neuronal es más un arte que una ciencia en el sentido de que el ajuste de muchos de sus parámetros depende directamente de la experiencia personal del diseñador. Sin embargo, existen métodos que mejoran significativamente el desempeño del algoritmo back-propagation. En [Haykin 1999] se describen los siguientes.

Actualización secuencial o procesamiento por grupos ("batch mode")

La principal diferencia entre el procesamiento por grupos y el secuencial reside en la forma de calcular el error de propagación hacia atrás. En el procesamiento secuencial

Tabla 2.3: Número óptimo (N_J) de neuronas ocultas sugerido en algunos trabajos. N_I representa la cantidad de neuronas en la capa de entrada, N_w identifica el número total de conexiones en la red, N es la cantidad de muestras de entrenamiento y N_K las neuronas correspondientes a la capa de salida.

Autor(es)	Estrategia
Hecht-Nielsen [Hecht-Nielsen 1990]	$N_J \leq N_I + 1$
Jadid y Fairbairn [Jadid 1996]	$N_J = \frac{N}{R+N_I+N_K}$ donde $R = -5$
Lachtermacher y Fuller [Lachtermacher 1995]	$\frac{0.11N}{N_I+1} \leq N_J \leq \frac{0.3N}{N_I+1}$
Masters [Masters 1993]	$N_J \approx (N_I \cdot N_K)^{1/2}$
Pao [Pao 1989]	$N_J = 2N_I$; o $N_J = N_I + 1$;
Duda et al. [Duda 2001]	$N_{wk} \approx \frac{N}{10}$ N_{wk} conexiones por unidad oculta
Upadhaya and Eryureka [Upadhyaya 1992]	$N_w = Nlog_2(N)$; N_w esta relacionado a N_J

el error es obtenido a partir de la entrada de cada prototipo $\mathbf{x}_n$, mientras que en el procesamiento por grupos éste es establecido como el promedio de error de todos los prototipos contenidos en la ME.

El procesamiento secuencial presenta dos ventajas fundamentales: 1) Es simple de implementar y 2) proporciona soluciones efectivas a problemas complejos.

Por su parte, el procesamiento por grupos facilita el establecimiento de condiciones teóricas para la convergencia del algoritmo, además de simplificar su paralelización.

Maximización de la información contenida en los datos de entrenamiento

Los datos presentados al algoritmo back-propagation deberían elegirse basándose en que la información contenida en estos datos sea lo más grande posible para cada tarea. Esto pude ser logrado de dos formas distintas: 1) el uso de ejemplos que

resulten en el mayor error de entrenamiento, o 2) utilizando ejemplos radicalmente diferentes de aquellos previamente procesados. Estas dos heurísticas están motivadas por el deseo de buscar en mayor profundidad el espacio de pesos.

En las tareas de clasificación de muestras usando el algoritmo back-propagation en modo secuencial, es común utilizar una técnica simple que consiste en presentar aleatoriamente los datos de entrenamiento en cada iteración.

Una técnica más refinada es presentada en [LeCun 1993]. Para entrenar la red se presentan más muestras "difíciles" que "fáciles". Por "difíciles" entiéndase aquellos elementos que generan mayor error durante el entrenamiento, mientras que los "fáciles" tendrán un comportamiento opuesto. Sin embargo, esta técnica presenta los siguientes inconvenientes 1) altera la distribución de los datos y 2) es muy sensible a la existencia de muestras atípicas o mal clasificadas.

Función de activación

Los MLP entrenados con el algoritmo back-propagation pueden aprender más rápidamente[12], cuando la función de activación corresponde a un modelo de neurona que utilice una función de activaciones antisimétrica[13]. Esta condición no es satisfecha por la función logística (Ec. 2.6), y sí por la función de activación tangente hiperbólica definida en la Ec. 2.7.

Valores objetivo

Es importante que los valores objetivo (las respuestas deseadas de la red) sean elegidos dentro del rango de la función de activación. Más específicamente, la respuesta deseada d_k para la neurona k en la capa de salida del MLP debería ser desplazada mediante un valor de compensación ϵ, alejándola del valor de la función de activación dependiendo de si este valor es positivo o negativo. De lo contrario, el algoritmo back-propagation tiende a llevar a infinito los parámetros libres de la red y el proceso de aprendizaje se hace más lento al ser dirigido por neuronas ocultas en saturación.

Normalización de los datos

Los datos de entrada a la red deberían ser preprocesados de manera que su valor medio sobre el conjunto de entrenamiento completo este cercano a cero o pequeño comparado con su desviación estándar [LeCun 1993]. Para apreciar el significado práctico de esta regla consideremos el caso extremo donde las variables de entrada son consistentemente positivas.

[12] En términos de número de iteraciones.

[13] Se dice que una función de activación $f(v)$ es antisimétrica si $f(-v) = -f(v)$.

En esta situación, los pesos de una neurona en la primer capa oculta sólo pueden incrementarse juntos o decrementarse juntos. En consecuencia, si el vector de pesos de esas neuronas tiene que cambiar de dirección sólo puede hacerlo al zigzaguear su camino a través de la superficie de error, proceso que es normalmente lento y debería ser evitado.

Para acelerar el entrenamiento del back-propagation el proceso de normalización debe incluir dos aspectos: 1) los datos de entrenamiento deben ser *no correlacionados* y 2) sus *covarianzas* deben tomar valores aproximadamente iguales [LeCun 1993].

Inicialización

Una buena elección de los valores iniciales de los pesos y los umbrales puede ser de gran ayuda en el diseño exitoso de la red. La pregunta clave es: ¿Cuál es una buena elección?.

Cuando a los pesos se les asignan valores iniciales grandes es altamente probable que las neuronas de la red entren en saturación. Si esto ocurre los gradientes locales en el algoritmo back-propagation asumen valores pequeños, lo cual a su vez causará que el proceso de aprendizaje sea más lento. Sin embargo, si a los pesos se les asignan valores iniciales pequeños el algoritmo back-propagation puede operar en un área muy aplanada alrededor del origen de la superficie de error; esto es particularmente cierto en el caso de las funciones de activación antisimétricas [Haykin 1999] como la función tangente hiperbólica (Ec. 2.7). Por estos motivos debe evitarse el uso de valores tanto grandes como pequeños para la inicialización de los pesos.

La elección apropiada para la inicialización cae en algún lugar en medio de estos dos casos extremos.

Aprender desde pistas

Al aprender a partir de un conjunto de entrenamiento se trabaja con una función $S(\cdot)$ que establece una relación entrada-salida desconocida. El proceso de aprendizaje explota la información contenida en los ejemplos acerca de la función $S(\cdot)$ para inferir una implementación aproximada de ella.

El proceso de aprendizaje desde ejemplos puede ser generalizado para incluir *aprendizaje desde pistas*, que puede lograrse incluyendo información previa que se pueda obtener de la función $S(\cdot)$ en el proceso de aprendizaje [Abu-Mostafa 1995]. Tal información puede incluir propiedades de la varianza, simetrías, o cualquier otro conocimiento acerca de la función $S(\cdot)$ que puede emplearse para acelerar la búsqueda de su implementación aproximada y aún más importante, mejorar la calidad de la estimación final.

Razón de aprendizaje: mecanismo para acelerar la convergencia

Todas las neuronas en un MLP idealmente aprenderían a la misma velocidad. Sin embargo, las últimas capas usualmente tienen gradientes locales mayores que las capas en la parte inicial de la red. Por lo tanto, se debería asignar una tasa de aprendizaje η con un valor más pequeño en las últimas capas que en las primeras.

Las neuronas con muchas entradas deberían tener una tasa de aprendizaje menor que las neuronas con pocas entradas, de manera que el tiempo de aprendizaje sea similar para todas las neuronas de la red. En [LeCun 1993] se sugiere que para una neurona dada, la tasa de aprendizaje debería ser inversamente proporcional a la raíz cuadrada de los pesos conectados a esa neurona[14].

2.5.5 Modificaciones al algoritmo back-propagation

En la actualidad se han propuesto diversas alternativas al algoritmo back-propagation dirigidas a superar sus deficiencias. Estas propuestas tienen el objetivo de acelerar la convergencia del método. Por ejemplo, la regla *delta-bar-delta* está basada en la utilización de razones de aprendizaje adaptativas [Jacobs 1988], o el *Quickprop* que se centra en la modificación de los pesos en función del valor gradiente obtenido en la iteración actual y la anterior [Fahlman 1988].

Por otra parte se ha propuesto la aplicación del gradiente conjugado para evitar que la red quede atrapada en mínimos locales [Haykin 1999]. El algoritmo basado en gradiente conjugado consiste en el cálculo de la segunda derivada del error con respecto a cada peso, y en obtener el cambio a realizar a partir de este valor y el de la primera derivada.

Algunos trabajos relativamente recientes en redes neuronales utilizan métodos estocásticos de optimización, los cuales no requieren derivadas y tienen la ventaja de que pueden eludir mínimos locales. Entre estas estrategias cabe citar el método de Nelder y Mead combinado con mínimos cuadrados lineales [Hsu 1999], el recocido simulado [Bárdossy 1998] y los algoritmos genéticos [Yao 1993].

La abundancia de trabajos sobre el ajuste de pesos indica que no hay un procedimiento comúnmente aceptado, y que para cada ANN el éxito de la misma depende de la eficiencia del método de optimización y de su capacidad para eludir los mínimos locales.

[14]Para mayor detalle el lector interesado debería estudiar los capítulos V de [Looney 1997], el VIII de [Rojas 1996], la sección 4.6 y 6.8 de [Haykin 1999] y [Duda 2001] respectivamente, que están dedicados a estudiar diversos algoritmos diseñados para dar mayor estabilidad y celeridad a la convergencia del MLP entrenado con métodos de descenso por gradiente.

2.6 Redes Neuronales de Funciones de Base Radial

Las redes reuronales de Funciones de Base Radial (Radial Basis Function, RBF) son un poderoso tipo de redes de propagación hacia adelante [Looney 1997]. Estas redes fueron introducidas a finales de los años 80 en la solución de problemas de interpolación de funciones multivariadas [Powell 1987].

En los últimos años por la simplicidad de su arquitectura y método de entrenamiento, las redes RBF se han convertido en una atractiva alternativa al MLP [Wettschereck 1992, Uykan 2000]. La principal diferencia entre las redes RBF y el MLP esta en la función de activación de los nodos ocultos [Ding 2004].

Una red RBF se caracteriza por tener sólo tres capas que realizan actividades diferentes (Fig. 2.5). La capa de entrada relaciona a la red con el entorno. La segunda capa o también llamada oculta aplica una transformación no lineal al espacio de entrada[15]. La tercera capa o de salida es lineal y sólo da las respuestas de la red a las estimulaciones recibidas del entorno.

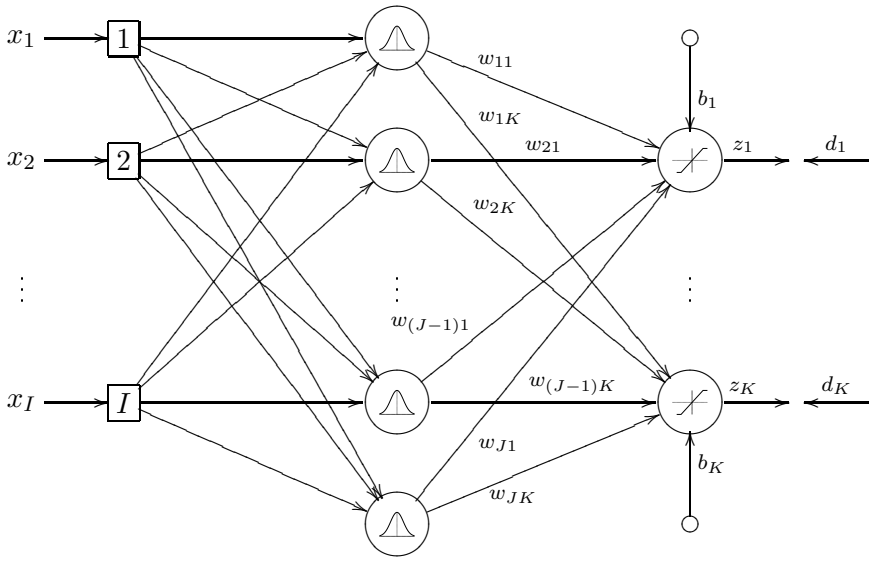


Fig. 2.5: Arquitectura general de una red RBF: I nodos en la entrada, J neuronas ocultas y K nodos de salida. $\mathbf{z}_n$ es la salida real de la red y $\mathbf{d}_n$ la esperada para la entrada $\mathbf{x}_n$. w_{jk} son los pesos de la red ($k = 1...K$; $j = 1...J$).

[15] Observe que no existen pesos entre los datos de entrada y las unidades ocultas.

El modelo de red RBF estándar puede ser formulado como sigue

$$z_k(\mathbf{x}) = \sum_{j=1}^{J} w_{jk} h_j(\|\mathbf{x} - \mathbf{c}_j\|) + w_{0k}\,, \tag{2.17}$$

donde

$$h_j(\|\mathbf{x} - \mathbf{c}_j\|) = exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{2\sigma_j^2}\right)\,, \tag{2.18}$$

$\|\cdot\|$ representa la norma Euclidiana, h_j es la función activación en la capa oculta con centro en $\mathbf{c}_j$ y varianza σ_j^2. El vector de pesos (w_{jk}) establece la relación entre la capa oculta y la de salida, w_{0k} representa el valor asignado al sesgo o "bias".

2.6.1 Proceso de aprendizaje

El entrenamiento o aprendizaje de una red es el proceso por el cual los parámetros libres de la red son adaptados a partir de las estimulaciones recibidas del ambiente, y de esta forma puedan desempeñar las tareas que se les asignen eficientemente [Jain 1996]. En el caso particular de las redes RBF, los parámetros libres que hay que adaptar al problema son: pesos $(\mathbf{w})$, centros $(\mathbf{c})$ y varianzas (σ^2) [Looney 1997].

Antes de iniciar el proceso de entrenamiento de la red se deben considerar dos aspectos básicos: a) seleccionar el modelo de RBF para la capa oculta y b) determinar el número necesario de neuronas ocultas para resolver el problema.

Algunos modelos de RBF se muestran a continuación:

1. Cuadrática: $h(x) = (x^2 + c^2)^{\frac{1}{2}}$ para algún valor $c > 0$ y $x \in \mathbb{R}$
2. Cuadrática Inversa: $h(x) = \frac{1}{(x^2 + c^2)^{\frac{1}{2}}}$ para algún valor $c > 0$ y $x \in \mathbb{R}$
3. Función Gaussiana: $h(x) = exp(-\frac{(x-c)^2}{2\sigma^2})$ para algún valor $c > 0$ y $x \in \mathbb{R}$

En la red RBF es común el uso de funciones basadas en una distribución de probabilidad normal o Gaussiana (Ec. 2.18) [Ghodsi 2003] y determinar empíricamente el número de neuronas de la capa oculta. No obstante, en los últimos años se ha incrementado el interés por automatizar este proceso [Schölkopf 1997, Xu 1998, Harpham 2004].

2.6.2 Aprendizaje Híbrido

Las capas de la red RBF realizan diferentes tareas. Por lo tanto, el entrenamiento de la red puede ser dividido en dos fases [Haykin 1999]. La primera fase consiste en determinar el número y posiciones de los centros, así como la varianza de las RBF en la capa oculta. La segunda fase corresponde a la estimación de los pesos de la red.

2.6.3 Localización de los centros de las RBF

Estudios teóricos y empíricos han mostrado que el rendimiento de las redes RBF dependen directamente de los valores usados por la capa oculta [Uykan 2000].

Existen numerosas formas de entrenar una red RBF y se reconocen por la forma en como los centros son localizados [Schwenker 2001]. A continuación, se describen brevemente dos de las formas más comunes para realizar esta tarea.

Selección aleatoria

Una de las técnicas más sencillas para la selección de los centros consiste en elegirlos de forma aleatoria desde los datos de entrada. Este método no puede ser considerado óptimo [Lowe 1989] ya que se asume que los datos de entrenamiento están dispuestos de acuerdo a la distribución real del problema.

Estrategias *clustering*

Otro enfoque es el uso de estrategias de *clustering* para la ubicación de los centros de la red [Schwenker 2001]. Por ejemplo, el algoritmo k-means[16] [Duda 2001], el LVQ (learning vector quantization) [Gray 1984], o los mapas autoorganizados de Kohonen (Selft Organization Maps, SOM) [Kohonen 1990].

2.6.4 Desviación Estándar

Para fijar el valor de la desviación estándar de las RBF se suele utilizar algún valor heurístico. En [Haykin 1999] se sugiere el uso de un valor proporcional a la distancia máxima entre los centros como se expresa a continuación:

$$\sigma = \frac{d_{max}}{\sqrt{2J}}, \tag{2.19}$$

donde d_{max} es la distancia máxima entre los centros elegidos y J el número de centros. Esta fórmula asegura que la forma de las funciones no sea demasiado suave ni excesivamente pronunciada, condiciones que deben ser evitadas.

No obstante, al establecer el mismo valor de σ para todas las RBF se asume que los datos están distribuidos de manera uniforme, situación que pocas veces ocurre en la práctica [Benoudjit 2003].

En [Saha 1989] se sugiere una σ distinta para cada RBF y se propone el uso de la estrategia heurística *nearest neighbour* (Ec. 2.20). El valor de σ_j se obtiene de

[16] Es el algoritmo más frecuentemente empleado con este propósito.

multiplicar la distancia entre el centro c_j y su vecino más próximo c_p por un valor constante de solapamiento (r).

$$\sigma_j = r \cdot min(\|c_j - c_p\|)\,. \tag{2.20}$$

Se han propuesto otras alternativas heurísticas para la ubicación de σ, por ejemplo, el *p-nearest neighbour* [Moody 1989]. En este procedimiento σ es establecida como el promedio de las distancias del centro c_j y sus p vecinos más próximos (Ec. 2.21).

$$\sigma_j = \frac{1}{p}\left(\sum_{i=1}^{p}\|c_j - c_i\|^2\right)^{\frac{1}{2}}\,. \tag{2.21}$$

2.6.5 Cálculo del vector de pesos

Al obtenerse los parámetros libres de la capa oculta ($\mathbf{c}$ y σ) solo resta estimar los valores de los pesos ($\mathbf{W}$) de la red. Esta tarea se puede realizar a través de la aplicación de algún método de optimización lineal [Jain 1996].

Suponiendo que $f(\mathbf{x}_n) = d_n$ es la salida deseada para la entrada $\mathbf{x}_n$ ($n = 1, ..., N$). La Ec. 2.17 se puede generalizar mediante notación matricial de la siguiente manera:

$$\mathbf{HW} = \mathbf{D} \tag{2.22}$$

en otras palabras,

$$\begin{pmatrix} h(\cdot)_{11} & h(\cdot)_{12} & \cdots & h(\cdot)_{1J} \\ h(\cdot)_{21} & h(\cdot)_{22} & \cdots & h(\cdot)_{2J} \\ \vdots & \vdots & \vdots & \vdots \\ h(\cdot)_{N1} & h(\cdot)_{N2} & \cdots & h(\cdot)_{NJ} \end{pmatrix} \begin{pmatrix} w_{11} & \cdots & w_{1K} \\ w_{21} & \cdots & w_{2K} \\ \vdots & \vdots & \vdots \\ w_{J1} & \cdots & w_{JK} \end{pmatrix} = \begin{pmatrix} d_{11} & \cdots & d_{1K} \\ d_{21} & \cdots & d_{2K} \\ \vdots & \vdots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{pmatrix},$$

donde $\mathbf{H} = (\mathbf{h_n}(\mathbf{x_n}))$, J el número de centros y K el número de clases en la ME o unidades en la capa de salida (ver Fig. 2.5).

A partir de la Ec. 2.22 se puede calcular $\mathbf{W}$ como sigue

$$\mathbf{W} = \mathbf{H}^{+}\mathbf{D} = (\mathbf{H^T H})^{-1}\mathbf{H^T D}\,, \tag{2.23}$$

donde $\mathbf{H}^{+}$ es la matriz *pseudo-inversa* de $\mathbf{H}$. En [Golub 1996] se estudian algunos algoritmos eficientes para el calculo de la matriz *pseudo-inversa*.

2.6.6 Aprendizaje Supervisado

Las redes RBF al igual que el MLP pueden ser entrenadas por métodos similares de descenso por gradiente [Ding 2004]. Así, los parámetros $U = \{w, c, \sigma\}$ de la red RBF son obtenidos simultáneamente. A continuación, se presenta una breve descripción del algoritmo back-propagation en el contexto de las redes RBF .

Considérese el error cometido por la ANN (Ec. 2.8), donde la salida real es:

$$z_k(\mathbf{x}_n) = \sum_{j=1}^{J} w_{jk} h_j(\|\mathbf{x} - \mathbf{c}_j\|) + w_{0k} \,. \tag{2.24}$$

Por lo tanto, nos interesa encontrar una solución que minimice el MSE (Ec. 2.8). Esta situación se puede formular como un problema de optimización sin restricciones. La condición necesaria para el valor óptimo es que $\nabla E(U^*) = 0$ y que $E(U^*) \leq E(U)$.

Una de las estrategias más populares para la minimización del MSE es el método de descenso por gradiente descrito en la sección 2.5.1. La clave en este método es encontrar los ∇ apropiados para actualizar en cada iteración los parámetros $U = \{w, c, \sigma\}$.

Al descomponer la Ec. 2.8 se tiene:

$$E(U) = E^{(1)} + E^{(2)} + \cdots + E^{(N)} \,, \tag{2.25}$$

donde el error parcial es

$$E^{(n)} = \sum_{k=1}^{K} \frac{1}{2}(d_k - z_k)^2 \,, \tag{2.26}$$

y así el problema es simplificado, y a partir de la Ec. 2.26 se pueden obtener las siguientes reglas de actualización para la red RBF.

$$\nabla w_{jk} = -\frac{\partial E}{\partial w_{jk}} = -\sum_{n=1}^{N} h_j(\|\mathbf{x}^n - \mathbf{c}_j\|)(d_k^n - f_k^n) \,, \tag{2.27}$$

$$\nabla c_{ji} = -\frac{\partial E}{\partial c_{ji}} = -\frac{1}{\sigma_j^2} \sum_{n=1}^{N} \left[\sum_{k=1}^{K} (d_k^n - f_k^n) w_{jk} \right] h_j(\|\mathbf{x}^n - \mathbf{c}_j\|)(x^n - c_{ji}) \,, \tag{2.28}$$

$$\nabla \sigma_j = -\frac{\partial E}{\partial \sigma_j} = -\frac{1}{\sigma_j^3} \sum_{n=1}^{N} \sum_{k=1}^{K} (d_k^n - f_k^n) w_{jk} h_j(\|\mathbf{x}^n - \mathbf{c}_j\|) \|\mathbf{x}^n - \mathbf{c}_j\|^2 \,. \tag{2.29}$$

Para profundizar en este tema consúltese el Anexo B donde se presenta el desarrollo seguido para llegar a estos resultados.

Los parámetros libres (U) de la red también pueden ser obtenidos por algún otro método de optimización no lineal, como por ejemplo el método de *Quasi-Newton* [Lowe 1989] o el de gradiente conjugado [Wettschereck 1992].

2.6.7 Otros enfoques de aprendizaje

Múltiples investigaciones han sido dirigidas a la construcción de la red RBF a partir de otros modelos. Por ejemplo, en [Schölkopf 1997] se hace uso de máquinas de vectores soporte (SVM) para determinar el número y localización de los centros de las RBF.

En [Kubat 1998] la fase de entrenamiento no supervisada de la red es realizada a partir de la construcción de un árbol de decisión donde cada *hiperrectángulo* generado por el árbol es interpretado como un *cluster* o nodo de la red.

Por otra parte en [Xu 1998], las redes RBF son presentadas como casos especiales del modelo *Alternative Mixture of Experts (ME)* [Xu 1995], y los parámetros de la capa oculta y de salida son obtenidos por el algoritmo *Expectation-Maximization (EM)* [Jordan 1994, Jordan 1995].

Otra tendencia es la utilización de algoritmos genéticos [Holland 1992]. La determinación del número de centros, así como su localización y varianzas, o la obtención de todos los parámetros de la red se han resuelto con el uso de estos métodos [Harpham 2004].

2.6.8 Red RBF + el Vector Funcional de Pao (red RBF+VF)

Las redes RBF + el vector funcional (functional link nets) (red RBF+VF) es una variante del vector funcional (funtional link net) de Pao [Pao 1994], o de las redes RBF según quiera verse [Dash 2007]. Es más general que las redes RBF porque tiene tanto conexiones no lineales como lineales.

En la Fig. 2.6 se observa que esta estructura de red se diferencia de las redes RBF porque incluye pesos extras ($\mathbf{u}$) entre la capa de entrada y la de salida (conexiones lineales).

Según Looney la principal ventaja de las redes RBF+VF es que necesita de un menor número de nodos ocultos, y la adición de conexiones incrementa su capacidad de aprendizaje [Looney 1997]. Teóricamente, la red RBF+VF debería producir mejores resultados que las redes RBF [Looney 2002].

En general, la red RBF representa un modelo no lineal para la relación entrada-salida, mientras que las redes RBF+VF incluyen un modelo no lineal y otro lineal (conexiones extras $\mathbf{u}$). Así, la parte lineal del problema no necesita ser aproximado por un modelo no lineal [Looney 2002]. Por lo tanto, el modelo red RBF+VF es más completo que el modelo no lineal red RBF. De hecho, una red RBF es la parte no lineal de la red RBF+VF.

De manera formal la salida de una red RBF+VF (ver Fig.2.6) se puede expresar como sigue

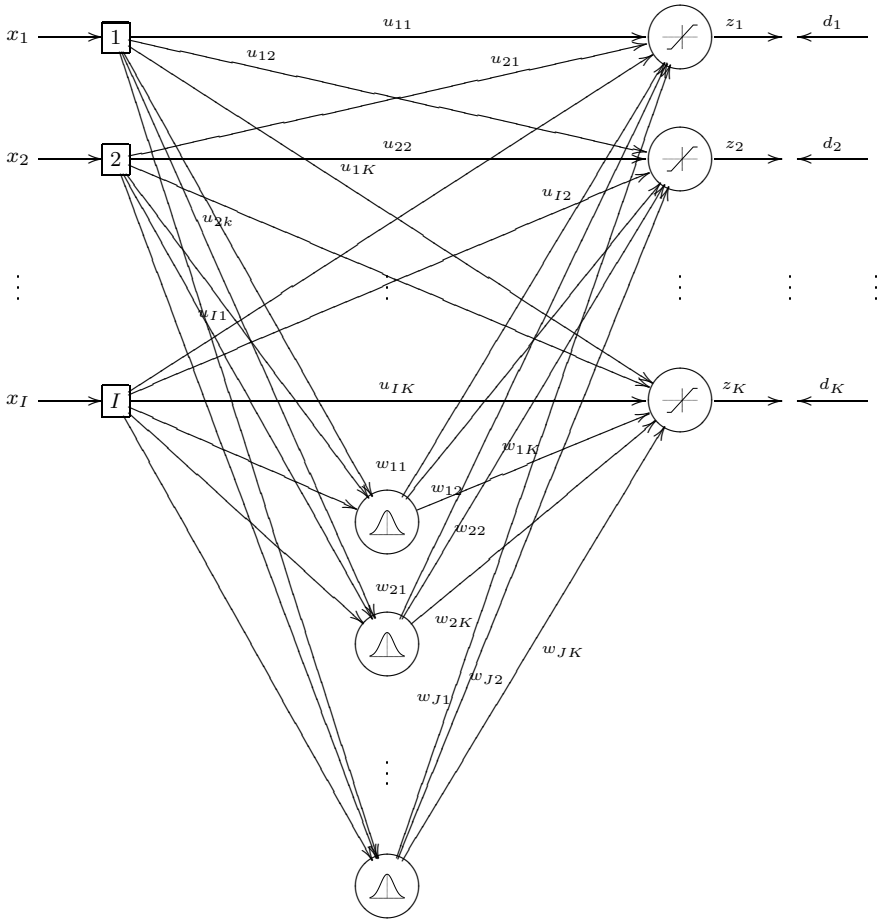


Fig. 2.6: Modelo de red RBF+VF compuesto de I nodos en la entrada, J neuronas ocultas y K nodos de salida. $\mathbf{z}$ es la salida real de la red y $\mathbf{d}$ la esperada para la entrada $\mathbf{x}$. $\mathbf{W}$ y $\mathbf{U}$ son los pesos de la capa oculta y de las conexiones adicionales a la red, respectivamente.

$$z_k(\mathbf{x}_n) = \sum_{j=1}^{J} w_{jk} h_j(\|\mathbf{x} - \mathbf{c}_j\|) + u_{ik} x_i \,. \tag{2.30}$$

Observe que la única diferencia con la Ec. 2.24 es la adición del vector funcional $\mathbf{u}$.

2.7 Red RBF vs MLP

Las redes RBF y el MLP son ejemplos claros de redes de propagación hacia adelante (*feedforward*) con capas no lineales [Haykin 1999]. Ambas redes son consideradas aproximadores universales [Looney 1997]. Otra característica en común, es que pueden ser entrenadas con métodos similares de descenso por gradiente (por ejemplo, con el algoritmo back-propagation) [Schwenker 2001]. No obstante, estos dos modelos neuronales presentan importantes diferencias [Haykin 1999, Ding 2004].

1. Las redes RBF tienen una capa oculta y el MLP pueden tener una o más.

2. Generalmente, en el MLP los nodos ocultos y los de salida tienen el mismo modelo neuronal mientras que en las redes RBF el modelo neuronal de la capa oculta y la de salida es distinto.

3. Los MLP generan una aproximación global de la relación no lineal entrada-salida, en tanto que en las redes RBF esta relación es local.

4. La principal diferencia entre las redes RBF y los MLP está en la función de activación de los nodos ocultos. En las redes RBF depende de la distancia entre los vectores de entrada y los centros de la red, mientras que en el MLP depende de el producto del vector de entrada y el vector de pesos.

2.8 Redes Modulares

En las secciones 2.6 y 2.5 se describen dos de los modelos de ANNs de propagación hacia adelante de mayor popularidad en la actualidad, los MLP y las redes RBF.

El MLP se caracteriza por el hecho de que todos los nodos de la red aportan información para el procesamiento de cada entrada (procesamiento global), mientras que en la red RBF sólo parte de la red se involucra en dicho procesamiento (procesamiento local). Esta situación realza la naturaleza complementaria que existe entre el uno y el otro [Ronco 1995].

En [Jacobs 1990] se destaca la importancia de lograr un nivel intermedio entre procesamiento local y global dado que muchas aplicaciones no son de naturaleza absolutamente local o global. Propone la incorporación de distintas ANNs para lograr una integración de ambas características.

La idea de unir diferentes prototipos de ANNs para formar uno solo da como resultado el concepto de ANN Modular (ANN-M). Motivadas inicialmente por la alta modularidad existente en los sistemas neuronales biológicos y basadas en el principio básico de la ingeniería: Divide y Vencerás [Haykin 1999], las ANN-M representan

una importante tendencia en el desarrollo de ANNs [Auda 1998]. En esta sección se discutirán algunos conceptos básicos de esta tendencia.

2.8.1 Motivación, definición y objetivos

El principio *Divide y Vencerás* consiste en descomponer un problema en subproblemas más simples del mismo tipo, resolverlos de forma independiente y una vez obtenidas las soluciones parciales combinarlas para obtener la solución del problema original.

El equivalente en redes neuronales al principio *Divide y Vencerás* son las arquitecturas que dividen el espacio entrada en diferentes subespacios o regiones. Estas arquitecturas integran varias redes donde cada una corresponde a uno de estos subespacios. Generalmente, estas arquitecturas son denominadas sistemas en comité[17](committee machines)[Haykin 1999].

Las ANN-M según su arquitectura y métodos de aprendizaje se pueden clasificar en dos categorías fundamentales:

1. Estructuras estáticas, en las que las respuestas de los expertos se combinan según un mecanismo en el que no influye el valor de la entrada para dar lugar a la salida final de la red. A esta categoría pertenecen los siguientes métodos:

   - *Arquitectura ventaja del conjunto (Ensemble averaging)*, donde las salidas de las diferentes redes neuronales (expertos) son combinadas linealmente para producir una salida global de la red. Durante el proceso de aprendizaje todos los expertos son entrenados con los mismos datos pero pueden diferir entre sí en las condiciones iniciales del proceso de entrenamiento (ver Fig. 2.7).
   - La estrategia *Boosting*, pretende elevar el desempeño de un algoritmo de aprendizaje débil combinando varias hipótesis adecuadamente para generar un algoritmo de aprendizaje fuerte [Freund 1995].

2. Estructuras dinámicas, donde el valor de la entrada influye en la manera en que se integran las respuestas parciales de los expertos para dar lugar a la salida global de la red. Entre ellas se encuentra el modelo Mezcla de Expertos y sus variantes.

[17]Por simplicidad a lo largo de este trabajo nos referiremos a los sistemas en comité como ANNs Modulares (ANN-M).

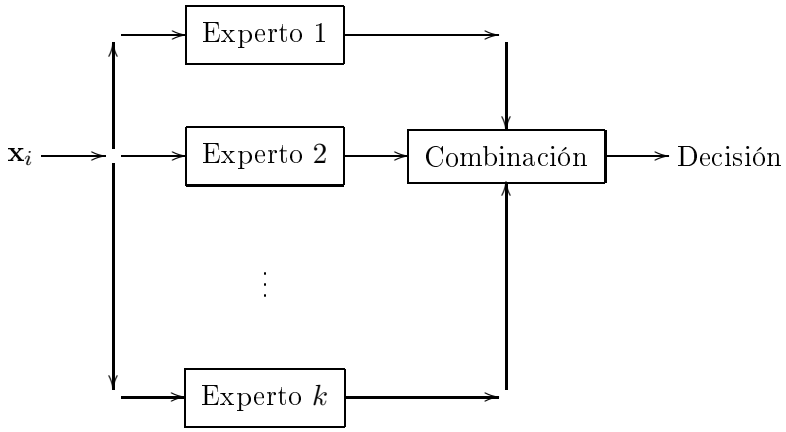


Fig. 2.7: Diagrama de bloques de la arquitectura ventaja del conjunto (Ensemble averaging).

Definición

En [Haykin 1999] se presenta la siguiente definición sobre el concepto ANN-M : "Se dice que una red neuronal es modular si la computación realizada por la red puede verse descompuesta en dos o más módulos (subsistemas), que operan en entradas distintas sin comunicarse con el otro. Las salidas de los módulos son mediadas por una unidad integradora a la que no se le permite realimentar información a los módulos. En particular, la unidad integradora decide como serán combinadas las salidas de los módulos para formar la salida final del sistema, y decide qué módulos deben aprender que muestras."

2.8.2 Diseño de ANN-M

Para el diseño de una ANN-M se deben considerar las siguientes fases:

- Descomposición del problema en subproblemas.
- Organización de la arquitectura modular.
- Comunicación entre módulos.

Descomposición del problema

La descomposición del problema es el primer paso en el desarrollo de una ANN-M. Se deben tener en mente las propiedades físicas y funcionales del problema. Cuando la descomposición del problema depende de la división del espacio de entrada puede

conseguirse haciendo uso de técnicas de agrupamiento (*clustering*), de autoorganización (SOM) o de cuantización vectorial (LVQ).

Para que la descomposición del espacio en grupos sea efectiva el criterio de partición debe ser el apropiado.

Generalmente, el criterio de partición se basa en un principio de vecindad espacial (por ejemplo la distancia euclidiana). Sin embargo, no en todos los casos un criterio de vecindad espacial es pertinente. Por ejemplo, en sistemas dinámicos continuos el criterio de partición debe incluir no sólo aspectos espaciales sino también temporales [Ronco 1995].

Independientemente de los modelos dinámicos, es evidente que un criterio de vecindad espacial no será adecuado en todas las situaciones. Cuando el espacio de entrada es heterogéneo será difícil aplicar un criterio de vecindad espacial efectivo.

En estos casos, la única forma de lograr una descomposición adecuada será haciendo uso de conocimiento *a priori* o previo del problema [Ronco 1995]. La descomposición del problemas en módulos dependerá de sus propiedades físicas o funcionales y por lo tanto, no es generalizable a todas las situaciones.

Antes de concluir este punto es necesario hacer mención del modelo de Jacobs [Jacobs 1991]. La relevancia de este modelo reside en su capacidad de dividir el espacio de entrada de forma autónoma sin necesidad de información *a priori*, o la aplicación de alguna técnica de agrupamiento o de autoasociación. Por lo tanto, puede ser usado cuando no se tiene información *a priori* o el problema no puede ser descompuesto. Este hecho ha convertido modelo de Jacobs en una atractiva alternativa en el desarrollo de ANN-M.

Organización de la arquitectura modular

La organización de la arquitectura modular se refiere a la forma en como las ANN-M deben construirse. De otra forma, ¿de cuantos módulos debe componerse el prototipo?, ¿que tipo de ANN ha de utilizarse en cada módulo?, así como de su configuración particular.

En [Happel 1994] se aplican algoritmos genéticos para encontrar la estructura modular apropiada para el modelo CALM (Categorising And Learning Module) [Murre 1992].

Otros trabajos se han desarrollado con el objetivo de encontrar mejores métodos para la estructuración sistemática de las ANN-M. Por ejemplo en [Dorizzi 1993], el número de módulos es fijado en función del total de clases. Se tendrán tantos módulos como clases se encuentren en el problema a tratar. En este caso la estrategia no consigue superar significativamente los resultados que se pueden obtener con una única ANN.

Si se considera el funcionamiento de los sistemas biológicos neuronales, la mejor forma de lograr la organización de la estructura modular es permitiendo que ella misma se organice progresivamente por su interacción con el ambiente como se sugiere en [Murre 1992]. Para lograr ésto, es necesaria la integración de algunas reglas en el algoritmo de aprendizaje que restrinjan la estructura de la red durante el entrenamiento. Existen dos formas de realizarlo: utilizando métodos de poda o mediante procedimientos incrementales.

Básicamente, las técnicas de poda se centran en la eliminación de conexiones entre elementos o neuronas de la red. En [LeCun 1990] se describe uno de los métodos de poda más usados. Consiste en la ubicación y eliminación de las conexiones entre elementos (o pesos) que al ser removidas causen variaciones mínimas en la función de costo o error de la red. Para mayor información sobre estrategias de poda, véase los trabajos [Reed 1993, Jutten 1995].

Los procedimientos incrementales tratan de adaptar la estructura al problema en particular. Este procedimiento se inicia con una estructura pequeña y se van agregado elementos hasta encontrar una solución al problema, de tal forma que no se necesita hacer una estimación previa sobre el tamaño de la estructura. En [Castillo 1991] se hace una revisión de algunos procedimientos incrementales.

Comunicación entre módulos

Una vez efectuado el análisis del problema, construidos y ajustados los módulos que se hacen responsables de la resolución de cada subproblema, es necesario especificar el mecanismo que integre cada una de las soluciones parciales alcanzadas para crear la solución al problema original.

Generalmente y de acuerdo a la manera en que se ha realizado el reparto de la información, se pueden distinguir diferentes métodos de integración o combinación de los módulos que se hayan considerado. A partir de los trabajos descritos en [Ronco 1995, Auda 1998, Haykin 1999, Happel 1994] se pueden identificar los siguientes:

- Mecanismo "el ganador se lo lleva todo". En este esquema el módulo o experto que muestre el valor de salida mayor es quien toma la decisión. Este método sólo se puede plantear en aquellos sistemas en los que los expertos realizan tareas parecidas y ofrecen resultados homogéneos.

- Sistemas cooperativos. Se fundamentan en el uso de las salidas de los expertos como entradas a otros expertos. En [Bottou 1990] se introduce formalmente el entorno de trabajo de los sistemas cooperativos.

- Combinación lineal de las salidas. En esta estrategia todos los expertos aportan parte de la solución, la cual es ponderada e integrada para determinar la solución global. En [Hashem 1997] se presenta una breve revisión de algunas de las estrategias de combinación lineal más relevantes.

- Combinación no lineal de las salidas. A diferencia del punto anterior las salidas son combinadas de forma no lineal. Un ejemplo claro de este tipo de ANN-M es el modelo de Jacobs [Jacobs 1990], donde la *gating network*[18] modula las respuestas parciales de los expertos antes de dar lugar a la salida global del sistema.

2.8.3 Ventajas y limitaciones de las Redes Modulares

Las ANN-M son modelos muy flexibles que pueden adaptarse a un amplio rango de aplicaciones. Representan una atractiva tendencia en el diseño de ANNs. Basadas en el principio *Divide y Vencerás* las ANN-M ofrecen ventajas desde diferentes enfoques.

- La descomposición del problema proporciona un aumento de la velocidad de aprendizaje [Anand 1995, Auda 1998]. Cada módulo experto suele tener menor tamaño (elementos de proceso) y se encarga de un subproblema que por definición es de resolución más sencilla que la tarea global.

- Las ANN-M facilitan la incorporación de técnicas mixtas [Bottou 1990]. Permiten la combinación de diferentes tipos de estructuras de ANN mientras se respeten las interfaces. Además de admitir reutilizar los módulos (programación orientada a componentes) o la posibilidad de evaluar el funcionamiento de los módulos de forma independiente antes de su integración.

- La representación de los datos de entrada desarrollada por una red modular tiende a ser más fácil de interpretar que en el caso de las estructuras monolíticas. En [Ronco 1995] se destaca que la descomposición del problema permite la inclusión de conocimiento previo sobre la solución del mismo, y de esta forma se ayuda significativamente en su resolución.

Estas son algunas de las ventajas de aplicar ANN-M en lugar de modelos monolíticos pero como se esperaría también presenta serias desventajas. Las principales se centran en dos temas básicos: a) La descomposición del problema, y b) la comunicación entre módulos. Ambos puntos son líneas de investigación que permanecen abiertas.

[18] Prototipo de red no lineal.

2.9 Análisis del error en la ANN

La esencia del aprendizaje de una ANN es codificar la relación *entrada-salida* del conjunto de datos de entrenamiento. Para ello, se requiere que la ANN se entrene de forma que aprenda lo suficiente acerca del pasado (datos de entrenamiento) para responder correctamente en el futuro (datos no vistos previamente[19]) [Haykin 1999]. Por lo tanto, es necesario algún criterio que pueda cuantificar la efectividad del clasificador.

Lo común es medir la efectividad de las ANNs a partir de alguna función error [Looney 1997].

2.9.1 Función de error

El proceso de aprendizaje de la ANN se refiere al problema de encontrar una estructura que aproxime la relación *entrada-salida*. Así, la función de error puede ser expresada como sigue:

$$E(\mathbf{W}) = \frac{1}{2N} \sum_{n=1}^{N} \|\mathbf{d}(\mathbf{x}_n) - \mathbf{f}(\mathbf{W}, \mathbf{x}_n)\|^2, \tag{2.31}$$

donde $\mathbf{d}(\mathbf{x})$ es la salida que se quiere obtener de la ANN para el vector de entrada $\mathbf{x}$. $\mathbf{f}(\mathbf{W}, \mathbf{x})$ corresponde a la salida real y $\mathbf{W}$ a los parámetros libres de la red. N es el número de elementos en la ME.

La Ec. 2.31 es utilizada comúnmente para establecer los criterios de parada o convergencia en el algoritmo back-propagation. En este caso no puede demostrarse su convergencia y no hay un criterio bien definido para detener su operación. Sin embargo, hay algunos criterios razonables que pueden ser utilizados para finalizar el ajuste de los pesos [Haykin 1999].

El proceso de entrenamiento se podría detener cuando $E(\cdot)$ es suficientemente pequeño, es decir, en el momento que $E(\cdot)$ alcance un valor predeterminado (ϵ) [Duda 2001]. No obstante, esto por si solo no es suficiente [Kramer 1988]. En casos donde la función de error alcanza mínimos locales es posible que nunca se llegue al valor ϵ. En [Fahlman 1988] se propone limitar el entrenamiento a un determinado número de iteraciones y de esta forma garantizar la terminación del proceso de aprendizaje a pesar de que no alcance el valor de ϵ.

Para [Kramer 1988] una representación más adecuada del criterio de convergencia es la siguiente:

[19] Aquí se asume que los datos de prueba son obtenidos de la misma población usada para generar los datos de entrenamiento.

- Considérese al vector de pesos $\mathbf{w}^*$ que denota un mínimo ya sea local o global. Una condición necesaria para que $\mathbf{w}^*$ sea un mínimo es que el vector gradiente $g(\mathbf{w}^*) = \nabla E(\mathbf{w}^*) = 0$. En consecuencia se puede formular el criterio de convergencia como $\|g(\mathbf{w}^*)\| \leq \epsilon$ donde ϵ es un umbral de gradiente suficientemente pequeño.

La desventaja de este criterio de convergencia es que para experimentos en los que se minimice el error los tiempos de aprendizaje pueden ser grandes.

Otra propiedad única acerca de la minimización de la función de coste o medida de error (E_{av}) que se puede usar es el hecho de que es estacionaria en el punto $\mathbf{w} = \mathbf{w}^*$. Se puede entonces sugerir un criterio de convergencia distinto [Haykin 1999]:

- Se considera que el algoritmo ha convergido cuando la tasa absoluta de cambio en el error cuadrático promedio $E_{av}()$ por iteración es suficientemente pequeña.

La tasa de cambio en el error cuadrático promedio se considera lo suficiente pequeña si decrece en el rango del 0,1 al 1,0 por ciento por cada iteración. A veces se usa un valor incluso menor como 0,01. Desafortunadamente este criterio puede finalizar de forma prematura y la red podría no aprender lo suficiente [Haykin 1999].

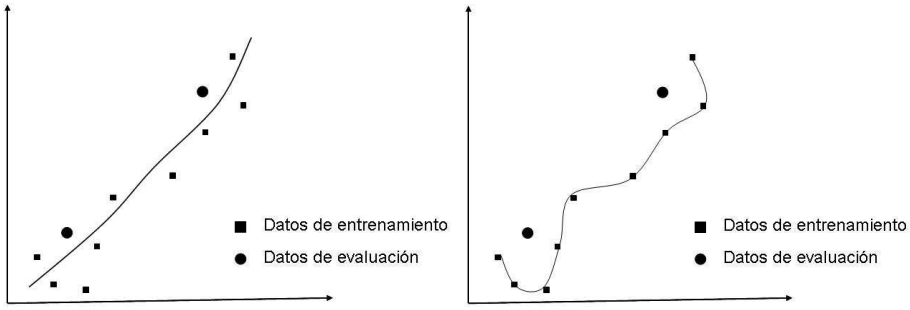
Otro criterio útil para detener el entrenamiento de la ANN es el denominado "Detención Temprana (Early Stopping)". Este criterio será discutido más adelante.

2.9.2 Capacidad de generalización

El concepto *generalizar* se refiere a la capacidad de la ANN de responder correctamente a situaciones no vistas previamente (datos de prueba) o al potencial de la ANN para clasificar correctamente los objetos con los cuales la red no fue entrenada.

Una ANN diseñada para generalizar bien producirá un mapeo *entrada-salida* correcto aún cuando la entrada sea ligeramente distinta de los ejemplos usados para entrenar a la ANN (Fig. 2.8(a)). Sin embargo, si se permite que la ANN cubra con demasiada perfección los datos de entrenamiento, se corre el riesgo de sobre ajuste o sobre entrenamiento (overfitting) y se pierde su habilidad para generalizar entre muestras de entrada-salida similares (Fig. 2.8(b)).

Se puede identificar el comienzo del sobre ajuste [Haykin 1999] a través del uso de técnicas de validación cruzada como la "Detención Temprana (*Early Stopping*)" (Fig. 2.9). En esta estrategia los datos con los que se cuenta son divididos en tres subconjuntos disjuntos: entrenamiento (*tra*), prueba (*tst*) y evaluación (*eval*). La ANN es entrenada con *tra* y después de cada iteración se evalúa la red con *eval*. El proceso de aprendizaje se detiene cuando se alcanza el mínimo $E_{av}()$ al clasificar *eval*.



(a) Generalización (b) Sobreajuste

Fig. 2.8: (a) Datos de entrenamiento ajustados apropiadamente, (b) Sobre ajuste en los datos de entrenamiento.

La capacidad de generalización de la ANN es medida a partir de la clasificación de los datos de prueba *tst*. Este método es utilizado como un criterio de convergencia o parada del algoritmo de aprendizaje de la red [Duda 2001].

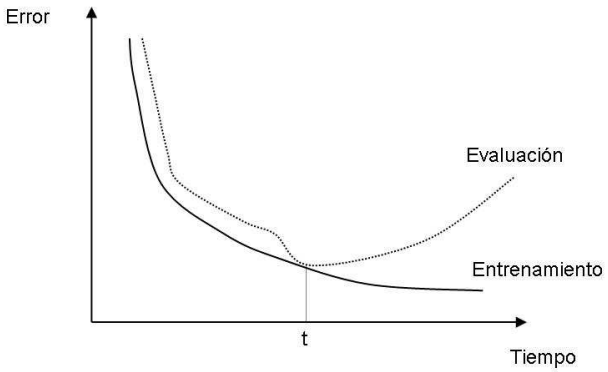


Fig. 2.9: Ilustración del método de "Detención Temprana (Early Stopping)". El índice t indica el punto en el tiempo donde debe detenerse el proceso de entrenamiento.

En [Amari 1996] se presenta una teoría estadística del fenómeno del sobre ajuste donde se advierte de los riesgos en los que se incurren con la utilización del método de "Detención Temprana".

2.9.3 Error de clasificación

Es importante conocer la efectividad del clasificador no sólo en términos de la función de error de la Ec. 2.31. En situaciones donde se quiere evaluar la efectividad del clasificador en términos más generales o se quiere comparar distintos clasificadores, la función de error de la Ec. 2.31 es insuficiente.

Generalmente, el cálculo del *error de clasificación* es utilizado como medida de efectividad. El error de clasificación consiste en identificar el número de errores cometidos por el clasificador (C).

Considérese un conjunto disponible de datos etiquetados ($\mathbf{X}$) para estimar el error de clasificación, entonces la forma más natural de realizar esta tarea es:

$$\text{Error}(C) = \frac{N_{error}}{N_X}, \tag{2.32}$$

donde N_{error} es el número de identificaciones erróneas cometidas por C y N_X la cantidad de muestras en $\mathbf{X}$.

Si $s_n \in \Omega$ es la etiqueta de clase asignada por C al objeto $\mathbf{x}_n$ la Ec. 2.32 se puede reescribir como

$$\text{Error}(C) = \frac{1}{N_X} \sum_{n=1}^{N_X} \{1 - T(l(\mathbf{x}_n), s_n)\}, \tag{2.33}$$

donde $T(a, b)$ es un indicador de la función que toma los valores de 1 si $a = b$ y 0 si $a \neq b$. $l(\cdot)$ obtiene la etiqueta asignada a $\mathbf{x}_n$.

La *Precisión en la Clasificación* (PC) puede obtenerse a partir de la Ec. 2.33 como

$$\text{PC}(C) = 1 - \text{Error}(C)\,. \tag{2.34}$$

Matriz de Confusión

Es común evaluar la precisión del clasificador en forma de matriz de error también denominada tabla de contingencia o matriz de confusión. El ordenamiento de esta matriz suele ser tal que las clases reales aparecen en columnas mientras que las predichas aparecen en las filas de la matriz (ver Tabla 2.4).

La tabla así formada presenta una visión general de las asignaciones tanto de las correctas (elementos de la diagonal) como de las incorrectas (elementos fuera de la diagonal) [Ariza 1996]. De esta forma se recogen los denominados errores de omisión y de comisión.

Los errores de comisión lo forman los elementos que no perteneciendo a una clase aparecen en ella, mientras que los de omisión están formados por los elementos que perteneciendo a esa clase no aparecen en ella por estar incorrectamente incluidos en otra.

Tabla 2.4: Matriz de confusión o tabla de contingencia. K es el número de clases y N el total de elementos.

	Clases reales				
Clases predichas	1	2	$\cdots$	K	total (N_{i+})
1	N_{11}	N_{12}	$\cdots$	N_{1K}	N_{1+}
2	N_{21}	N_{22}	$\cdots$	N_{2K}	N_{2+}
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
K	N_{K1}	N_{K2}	$\cdots$	N_{KK}	N_{K+}
total (N_{+j})	N_{+1}	N_{+2}	$\cdots$	N_{+K}	N

En [Russell 1999] se detalla el uso de la matriz de error o confusión. A continuación se describen algunos datos importantes que pueden obtenerse directamente de la Tabla 2.4:

- Número de muestras de la clase i

$$N_{i+} = \sum_{j=1}^{K} N_{ij} \,. \tag{2.35}$$

- Número de muestras clasificadas dentro de la clase j de los datos de referencia

$$N_{+j} = \sum_{i=1}^{K} N_{ij} \,. \tag{2.36}$$

- PC global

$$\mathrm{PC} = \frac{\sum_{i=1}^{K} N_{ii}}{N} \,, \tag{2.37}$$

  donde N es total de elementos. Obsérvese que esta ecuación es equivalente a la Eq. 2.34.

- PC por clase

$$\mathrm{PC}_i = \frac{N_{ii}}{N_{i+}} \,. \tag{2.38}$$

En [Ariza 1996] se establecen las condiciones necesarias para la construcción de la matriz de confusión:

1. Las clases que se establezcan deben ser independientes, mutuamente excluyentes y exahustivas.

2. Deben usarse métodos de muestreo que excluyan autocorrelación.

3. Conviene el uso de métodos estratificados para asegurar la presencia de clases extrañas o minoritarias.

4. Para comprobar la bondad de un proceso de clasificación supervisado no se deben usar los elementos de entrenamiento del clasificador.

Media Geométrica

El desempeño del clasificador generalmente es medido según el promedio de aciertos (sec. 2.9.3) obtenidos en la fase de clasificación. No obstante, existen situaciones donde no puede ser considerada una medida adecuada [Barandela 2003a].

Por ejemplo, considere el caso de un conjunto de datos de entrenamiento de dos clases donde los datos de la clase A representan el 98% de la ME y el 2% restante a la clase B. En esta situación, obtener un 98% de precisión en la clasificación no supone un buen rendimiento dado que podría estar clasificando solamente una clase. Consecuentemente, otros criterios de medida deben ser adoptados [Kubat 1997].

Uno de los criterios de medida más ampliamente aceptados es la media geométrica (geometric mean, *g-mean*) [Barandela 2004]. La media geométrica es definida como

$$g\text{-}mean = \left(\prod_{k=1}^{K} \left(\frac{aciertos_k}{aciertos_k + errores_k} \right) \right)^{\frac{1}{K}} = \left(\prod_{k=1}^{K} \mathrm{PC}_k \right)^{\frac{1}{K}}, \tag{2.39}$$

donde $aciertos_k$ y $errores_k$ son el número de aciertos y errores de la clase k. Por lo tanto, PC_k representa la precisión de la clase k. Esta medida busca maximizar la efectividad por clase del clasificador.

2.9.4 Evaluación del producto final

Para construir y evaluar la capacidad de generalización del clasificador se han presentado diversas propuestas [Duda 2001]. Se basan en la separación de los datos para así estimar la probabilidad de error o acierto del producto final. En ellas, el conjunto de datos $\mathbf{X}$ juega un papel fundamental a la hora de cuantificar la efectividad del clasificador. Algunas de las principales alternativas [Kuncheva 2004] se resumen a continuación.

- Método R(restitución):
  - Diseñar C con $\mathbf{X}$.
  - Evaluar C con $\mathbf{X}$.

  Desventaja: Error(C) es optimistamente sesgado.

- Método H (Hold-out). En este método se dividen los datos disponibles ($\mathbf{X}$) en dos conjuntos disjuntos $\mathbf{X}_1$ (conjunto de aprendizaje) y $\mathbf{X}_2$ (conjunto de prueba) de tal forma que $\mathbf{X}_1 \cup \mathbf{X}_2 = \mathbf{X}$ y $\mathbf{X}_1 \cap \mathbf{X}_2 = \emptyset$.
  - Diseñar C con $\mathbf{X}_1$.
  - Evaluar C con $\mathbf{X}_2$.

  Desventaja: Da cifras pesimistas (mayores a las reales).

- Método de validación cruzada. Consiste en partir $\mathbf{X}$ en k subconjuntos disjuntos. Donde $k-1$ subconjuntos serán utilizados como datos de entrenamiento y el resto como datos de evaluación ($\mathbf{Z}$). Esto se realiza de tal manera que cada uno de los k subconjuntos actúe como $\mathbf{Z}$ una sola vez y el resto sea utilizado para diseñar C. El proceso termina hasta que cada subconjunto haya participado como muestra de evaluación una vez. Cuando $k = N$ el método es llamado *leave-one-out* o método U.

  La principal desventaja de la validación cruzada es que requiere una cantidad excesiva de computación porque el modelo tiene que ser entrenado k veces (donde $1 < k \leq N$).

- Método Boostrap. Este método se ha diseñado para superar las deficiencias del método-R. Consiste en elegir aleatoriamente n elementos (con remplazo) de X para generar X_a. Posteriormente, C es construido con X_a y es evaluado con X. Esto se repite b veces y se generan b conjuntos Boostrap que son tratados de forma independiente. Este método es muy útil cuando se trabaja con conjuntos de datos muy pequeños.

2.10 Aspectos experimentales

En este trabajo se estudia el problema del desbalance de las clases y sus efectos en tres arquitecturas diferentes de ANNs entrenadas con el algoritmo back-propagation con procesamiento por grupos o "batch mode". Este problema es analizado desde tres enfoques diferentes:

- Inclusión de funciones de coste al algoritmo de entrenamiento.
- Tratamiento del desbalance de las clases a partir del uso de redes neuronales modulares.
- Reducción del área de confusión en las fronteras de decisión.

Para llevar a cabo este estudio se desarrollaron una serie de pruebas con bases de datos sintéticas y reales de dos y múltiples clases. Esta sección está dedicada a describir los detalles relacionados a la experimentación efectuada en este trabajo.

Descripción de los datos

Los experimentos fueron desarrollados con conjuntos de datos extraídos del UCI Database Repository [Newman 1998], excepto por Cayo y Feltwell (para mayor detalle sobre los conjuntos de datos véase la sección 1.4).

En las Tablas 2.5 y 2.6 se resumen las características más relevantes de los conjuntos de datos de dos y de múltiples clases respectivamente. La Tabla 2.7 presenta la distribución de las muestras en las bases de datos de múltiples clases, i.e., indica el número de muestras por clase.

Tabla 2.5: Características relevantes de los conjuntos de datos de dos clases.

Datos	Muestras	Atributos	Clases	Distribución de las clases
B2Cls	625	4	2	49/576
Cancer	683	9	2	238/445
Diabetes	768	8	2	268/500
German	1000	24	2	300/700
Ionosphere	351	34	2	126/225
Liver	345	6	2	145/200
Phoneme	5404	5	2	1586/3818
Sonar	208	60	2	97/111
V2Cls	528	10	2	48/480

Tabla 2.6: Información relevante relacionada a las bases de datos de múltiples clases utilizadas en esta investigación.

Datos	Muestras	Atributos	Clases
Cayo	6019	4	11
Ecoli6	332	7	6
Feltwell	10944	15	5
Satimage	6430	36	6

Tabla 2.7: Número de muestras por categoría en las bases de datos de múltiples clases.

Datos	Distribución de las clases
Cayo	838/293/624/322/133/369/324/722/789/833/772
Ecoli6	5/143/77/52/35/20
Feltwell	3531/2441/896/2295/1781
Satimage	1508/1531/703/1356/625/707

Arquitectura de la ANN

En la parte experimental de este trabajo se usaron las siguientes ANN:

- Redes del tipo Perceptron Multicapa (MLP, sec. 2.5) con una capa oculta.
- Redes de Funciones de Base Radial (red RBF, sec. 2.6).
- Redes de Funciones de Base Radial mas el Vector Funcional de Pao (red RBF+VF, sec. 2.6.8).
- Redes Neuronales Modulares (ANN-M, sec. 2.8).

Todas ellas fueron entrenadas con el algoritmo back-propagation con procesamiento por grupos o “batch mode" (sec. 2.5.1).

El modelo de red modular utilizado es el de Arquitectura Ventaja del Conjunto (ver Fig. 2.7). Los módulos fueron construidos con MLP, redes RBF y redes RBF+VF.

Por simplicidad, para las bases de datos de dos clases se fijo el número de neuronas ocultas en 4 y la configuración inicial de la ANN fue determinada según el criterio de prueba y error.

Para las bases de datos de múltiples clases el número de nodos ocultos y la configuración inicial de las ANNs fueron establecidos por medio del método de prueba

y error. Por lo tanto, el número de neuronas ocultas y la configuración inicial corresponden a las características propias de cada base de datos.

Para aplicar el método de prueba y error se tomo el 30% de cada base de datos, así, se obtuvieron los parámetros iniciales de la ANN y el número de neuronas ocultas. En la tabla 2.8 se resume la configuración inicial del MLP, las redes RBF y las redes RBF+VF.

Tabla 2.8: Información relevante relacionada a la configuración utilizada en las ANNs. NO es en número de neuronas ocultas, η corresponde a la razón de aprendizaje y μ al momento.

Datos	MLP_{NO}	RBF_{NO}	MLP_{η}	MLP_{μ}	RBF_{η}	VF_{η}
Problemas de dos clases						
Cancer	4	4	0.9	0.1	0.0001	0.9
Diabetes	4	4	0.9	0.1	0.0001	0.9
German	4	4	0.9	0.1	0.0001	0.9
Ionosphere	4	4	0.9	0.1	0.0001	0.9
Liver	4	4	0.9	0.1	0.0001	0.9
Phoneme	4	4	0.9	0.1	0.0001	0.9
Sonar	4	4	0.9	0.1	0.0001	0.9
B2Cls	4	4	0.9	0.1	0.0001	0.9
V2Cls	4	4	0.9	0.1	0.0001	0.9
Problemas de múltiples clases						
Cayo	7	22	0.9	0.1	0.00001	0.9
Ecoli6	15	15	0.9	0.1	0.001	0.9
Feltwell	6	6	0.9	0.1	0.00001	0.9
Satimage	12	12	0.9	0.1	0.00001	0.9
Problemas de dos clases con ANN-M						
Cayo	2	4	0.9	0.1	0.00001	0.9
Ecoli6	4	4	0.9	0.1	0.001	0.9
Feltwell	4	4	0.9	0.1	0.00001	0.9
Satimage	4	4	0.9	0.1	0.00001	0.9

Inicialización

El valor inicial de los pesos fue asignado aleatoriamente con valores entre -0.5 y 0.5 como es sugerido en [Haykin 1999].

Para el caso de las redes RBF y RBF+VF la inicialización de los centros fue realizada a partir la estrategia de selección aleatoria (sec. 2.6.3) y los valores iniciales de las varianzas fueron obtenidos a partir del heurístico *nearest neighbour* (Ec. 2.20). El objetivo es evitar que la ANN inicie con una solución muy alejada a la real.

Los centros y varianzas son actualizados solamente después de cada 500 iteraciones con el propósito de prevenir oscilaciones bruscas en el MSE durante el proceso de entrenamiento [Looney 1997].

Finalmente, el criterio de parada se estableció en un máximo de 25000 iteraciones o un error (MSE) inferior a 0.0001 en todos los experimentos.

Criterios de evaluación

El desempeño del clasificador fue cuantificado a partir de los siguientes criterios de medida:

- Precisión global (Ec. 2.34).
- Precisión por clase (Ec. 2.38).
- Media geométrica (Ec. 2.39).
- Matriz de confusión (sec. 2.9.3).

Cada criterio corresponde a las necesidades de los distintos experimentos y lo que se pretende medir.

Evaluación del producto final

Para evaluar la efectividad de las ANNs se hizo uso del método k-fold-cross-validation con $k = 10$ (ver sec. 2.9.4), excepto en Ecoli6 donde $k = 5$. Observe en la Tabla 2.7 que en Ecoli6 sería inefectivo este método si el valor de $k > 5$.

Cada ANN fue entrenada 10 veces con valores iniciales diferentes. Los resultados presentados a lo largo de este trabajo corresponden al promedio obtenido de las k particiones y 10 repeticiones. Por ejemplo, en el caso de $k = 10$ los resultados mostrados son el promedio de 10 inicializaciones distintas y 10 particiones. En otras palabras, corresponden a 100 procesos de entrenamiento y clasificación.

Por otra parte, la base de datos Feltwell fue obtenida de otros trabajos [Alejo 2007], y se conservo de esa forma para permitir la comparación de los resultados experimentales con otras investigaciones. Esta base de datos esta dividida en datos de entrenamiento (5124 muestras) y de prueba (5820 muestras). Satimage fue obtenida de [Newman 1998] y se conservo su partición tal y como se sugiere en ese sitio.

Para Feltwell y Satimage se entrenó la red 30 veces. Así, los resultados presentados en este trabajo para estas bases de datos, hacen referencia al promedio de 30 inicializaciones distintas de la ANN.

Capítulo 3

Distribuciones no balanceadas: Funciones de coste

Contenido

3.1 Introducción

La entrada a una Red Neuronal Artificial (Artificial Neural Network, ANN) con aprendizaje supervisado consiste de una Muestra de Entrenamiento (ME). Una ME es un conjunto de datos previamente identificados por un experto humano que caracteriza un problema a resolver [Barandela 2004]. De manera formal se puede definir como

$$\mathrm{ME} = \mathrm{ME}_1 \cup \mathrm{ME}_2 \cup ... \cup \mathrm{ME}_K \,, \tag{3.1}$$

donde

$$\mathrm{ME}_n = (\mathbf{x}_n, \varphi(\mathbf{x}_n)),\ n = 1, ..., N_k \,, \tag{3.2}$$

$\mathbf{x}_n = [x_1, x_2, ..., x_d]^T$ es el vector de características que identifica una situación particular; $\varphi(\mathbf{x}_n)$ la clase a la que corresponde y N_k el número de muestras de la clase k.

En la Fig. 3.1 se ilustra el papel de la ME en el funcionamiento general de una ANN con aprendizaje supervisado.

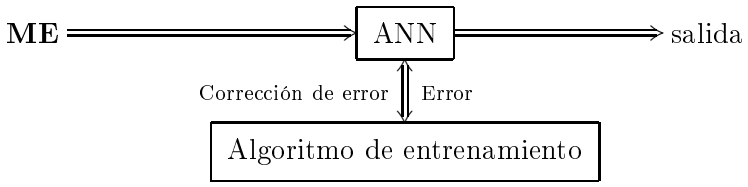


Fig. 3.1: Esquema a bloques del funcionamiento de una ANN entrenada con un algoritmo de corrección de error (por ejemplo el back-propagation con procesamiento por grupos).

Generalmente, los métodos de aprendizaje supervisado como las ANNs están diseñados para trabajar con MEs relativamente equilibradas[1] [Japkowicz 2002]. Sin embargo, existen numerosas aplicaciones donde la desproporción en el número de muestras entre clases es importante [Kotsiantis 2003]. Por ejemplo, en la detección de fraudes en llamadas telefónicas [Fawcett 1997], en la identificación de productos defectuosos en la línea de ensamblaje de partes de automóviles [Murphey 2004], o en detección transacciones ilegales con tarjetas de crédito[2] [Chan 1999]. También se han presentado en problemas médicos o en el diagnóstico de enfermedades raras [Newman 1998].

Una ME no balanceada es aquella donde la diferencia en el número de muestras de las distintas clases es considerable. De manera formal, si para alguna ME_i se cumple que

$$\|\mathrm{ME}_i\| << \|\mathrm{ME}_j\| \; i \neq j;\; i,j = 1,...,K\,, \tag{3.3}$$

donde K es el número total de clases en la ME.

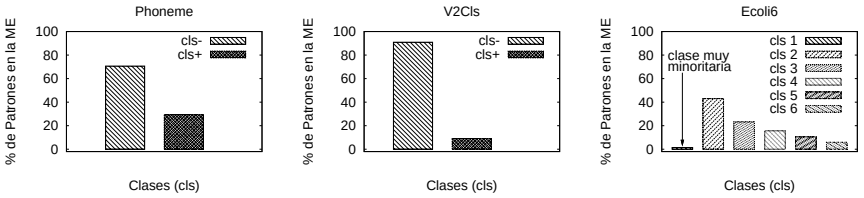
La Fig. 3.2 muestra tres ejemplos típicos de MEs no balanceadas[3]. $cls+$, hace referencia a la clase positiva o minoritaria y $cls-$ a la clase negativa o mayoritaria. Se observa en las Fig. 3.2a y 3.2b correspondientes a Phoneme y V2Cls respectivamente, que no es difícil identificar a la clase minoritaria. Sin embargo, en un problema de múltiples clases como Ecoli6, la definición de que es una clase minoritaria es una tarea complicada, porque no existe un criterio preestablecido que diga cuando una clase es minoritaria o respecto a quien se considera minoritaria. Así, la clase 4 puede ser considerada minoritaria con respecto a la clase 2, pero mayoritaria en relación

[1] Conjuntos de datos donde la diferencia en el número de muestras de las distintas clases no es significativa.

[2] El número de transacciones legales es mucho mayor que el número de transacciones ilegales.

[3] Las bases de datos introducidas en esta sección son utilizadas a lo largo del capítulo y sus características son descritas en las secciones 1.4 y 2.10.

a la clase 1. Esta situación es muy común cuando se trabaja con MEs de múltiples clases [Alejo 2007].



(a) (b) (c)

Fig. 3.2: Ejemplos de MEs no balanceadas.

Recientemente, el problema del desbalance de la ME se ha considerado como un problema crítico en la minería de datos y el aprendizaje automático [Zhou 2006]. Numerosos estudios se han desarrollado con el objetivo de mejorar la efectividad del clasificador cuando se entrena con MEs no balanceadas [He 2009].

En el contexto del MLP entrenado con el algoritmo back-propagation y dominios de dos clases el problema se ha formulado como sigue: La clase mayoritaria domina el proceso de entrenamiento y los elementos de la clase menos representada o minoritaria pueden ser ignorados [Anand 1993, Bruzzone 1997a, Lu 1998, Murphey 2004]. En consecuencia, la convergencia de esta última clase es muy lenta.

Diversos trabajos han sido dirigidos a combatir este problema. En [Anand 1993] se analiza al algoritmo back-propagation y se propone su modificación para acelerar el proceso de convergencia de la red. La idea está centrada en el cálculo del vector gradiente y su dirección, de forma que permita que el error pueda decrecer en la dirección de ambas clases, y se evite que la clase minoritaria pueda ser ignorada en el proceso de entrenamiento. El estudio está limitado a problemas de dos clases.

Más adelante en [Anand 1995] se extiende este enfoque a problemas de múltiples clases y redes modulares. En este último trabajo, el problema de múltiples clases es descompuesto en subproblemas de dos clases, y cada subproblema es resuelto por una red idéntica a la de la propuesta inicial de [Anand 1993] para dos clases.

Posteriormente, las salidas de las diferentes ANNs (expertos) son consideradas para producir una salida global de la red (Fig. 2.7). Los resultados de [Anand 1995] evidencian la conveniencia de descomponer un problema de múltiples clases a subproblemas de dos clases, y la utilidad de recalcular el vector gradiente y su dirección para tratar el desbalance de la ME en problemas de dos clases.

Por otra parte, en [Bruzzone 1997a, Bruzzone 1997b] se expone una alternativa al método anterior. Su objetivo es modificar el algoritmo back-propagation con el propósito de acelerar la convergencia de las clases menos representadas. Esta modificación consiste en incluir una función de coste en el algoritmo de entrenamiento y disminuir su valor a partir de una estrategia heurística, con la finalidad de reducir su impacto en la probabilidad de la distribución de los datos.

La principal ventaja respecto a [Anand 1995] es que no es necesario descomponer el problema de múltiples clases en subproblemas de dos clases. No obstante, la desventaja mas importante de este enfoque reside en la efectividad del mecanismo empleado para reducir el valor de la función de coste.

Otras alternativas para enfrentarse al problema del desbalance en la ME han sido las técnicas de muestreo (submuestreo *under-sampling* o sobremuestreo *over-sampling*) [Alejo 2006]. En las técnicas de *under-sampling* algunas muestras de las clase mayoritaria son eliminadas hasta alcanzar cierto grado de balance en la ME, mientras que en las de *over-sampling* el balance se alcanza duplicando o creando nuevas muestras de la clase minoritaria.

En las redes basadas en MLP, las técnicas de *under-sampling* o de *over-sampling* han mostrado notables mejoras en la efectividad del clasificador [Japkowicz 2002]. No obstante, eliminar muestras de la ME puede causar perdida de información relevante, o si se incrementa el tamaño de la clase minoritaria puede introducirse ruido en la misma, de forma que el tiempo de entrenamiento es incrementado y además se altera la distribución de los datos [Lawrence 1998].

En algunos trabajos [Fu 2002, Zhou 2006, Ling 2007] el problema del desbalance en la ME se plantea como un problema de aprendizaje sensible al coste (*cost-sensitive*). En este enfoque, el precio de cometer un error de clasificación debe ser distinto para cada clase [Kukar 1998]. El principal inconveniente de esta alternativa está en la necesidad de contar con información a priori sobre el problema en cuestión. Así, de antemano se debe cuantificar el coste de cometer cada error, o si es necesario, la implementación de sofisticados mecanismos para la obtención de los costes.

En este capítulo, se estudia y explica el efecto del desbalance de las clases sobre las ANNs entrenadas con el algoritmo back-propagation con procesamiento por grupos en dominios de dos y múltiples clases. Se evalúan las posibilidades de tres estrategias diseñadas para tratar el desbalance de la ME. Básicamente, estas estrategias consisten en la inclusión de funciones de coste en el algoritmo de entrenamiento.

3.2 Efecto del desbalance de la ME en el MSE

La primera pregunta que se debe hacer al iniciar una investigación sobre el desbalance de las clases es: ¿Cómo afecta al clasificador[4]?.

Estudios empíricos realizados al algoritmo back-propagation [Anand 1993] muestran que el desbalance de las clases de la ME genera aportaciones desiguales al error cuadrático medio (Mean Square Error, MSE) en la fase de entrenamiento de la red. La mayor parte de las aportaciones al MSE están dadas por la clase mayoritaria. En consecuencia, el entrenamiento de la red es dominado por las muestras de esta clase.

Considérese una ME de dos clases ($K = 2$) con N muestras de entrenamiento, donde $N = \sum_k^K N_k$ y N_k el número de muestras de la clase k. Supóngase entonces que el MSE por clase puede ser expresado como

$$E_k(U) = \frac{1}{N} \sum_{n=1}^{N_k} (\mathbf{d}^n - \mathbf{f}^n)^2 \,, \tag{3.4}$$

de tal forma que el MSE global puede ser referenciado por

$$E(U) = \sum_{k=1}^{K} E_k = E_1(U) + E_2(U) \,. \tag{3.5}$$

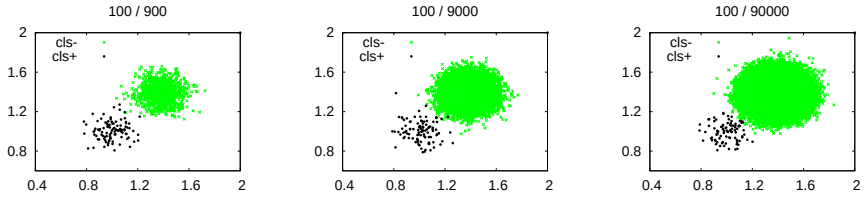
Si $N_1 << N_2$ entonces $E_1(U) << E_2(U)$ y $\|\nabla E_1(U)\| << \|\nabla E_2(U)\|$. Por lo tanto $\nabla E(U) \approx \nabla E_2(U)$. Así, $-\nabla E(U)$ no siempre es la mejor dirección para minimizar el MSE de ambas clases [Anand 1993].

Para ilustrar el comportamiento del MSE cuando se trabaja con MEs no balanceadas se desarrollaron una serie de experimentos con tres bases de datos artificiales (Fig. 3.3) diseñadas según un modelo de dos clases con distribuciones gaussianas bivariadas con medias $\mu_1(1, 1)$ y $\mu_2(1.385, 1.385)$. La distribución de los datos en la ME para la clase minoritaria y mayoritaria siguió la siguiente proporción: (a) 100 - 900, (b) 100 - 9000, y (c) 100 - 90000 (Fig. 3.3a, 3.3b y 3.3c respectivamente).

En la Fig. 3.4 se ilustra el comportamiento del MSE cuando se trabaja con MEs no equilibradas. El eje x ha sido escalado logarítmicamente dado que los principales cambios ocurren durante la primeras iteraciones.

En términos generales se observa que en las primeras iteraciones el MSE de la clase mayoritaria es reducido rápidamente mientras que el de la clase minoritaria es incrementado. Posteriormente, el MSE de esta última clase disminuye muy lentamente.

[4]En todo el trabajo nos referimos como clasificador a las ANNs entrenadas con el algoritmo back-propagation con procesamiento por grupos.



(a) (b) (c)

Fig. 3.3: MEs sintéticas de dos clases no balanceadas.

El efecto del desbalance en el proceso de entrenamiento de la ANN se traduce en un incremento en la cantidad de iteraciones necesarias para alcanzar la convergencia. Una consecuencia inmediata de este hecho es la dificultad para lograr un desempeño efectivo (en términos de clasificación) en un plazo de tiempo "razonable". Sobre todo en situaciones donde se tiene un desbalance extremo en la ME (Fig. 3.4g, 3.4h, 3.4i).

Podemos ver en la Fig. 3.4 que a medida que crece la desproporción en el número de muestras entre clases, se incrementa el número de iteraciones necesarias para alcanzar la convergencia. Para el primer escenario se requieren (en promedio) cerca de 10000 iteraciones mientras que en el segundo al acercarse a las 100000 iteraciones se empiezan a notar descensos importantes en el MSE de la clase minoritaria (cls+). En el tercer escenario no se vislumbra cuando convergerá la clase minoritaria.

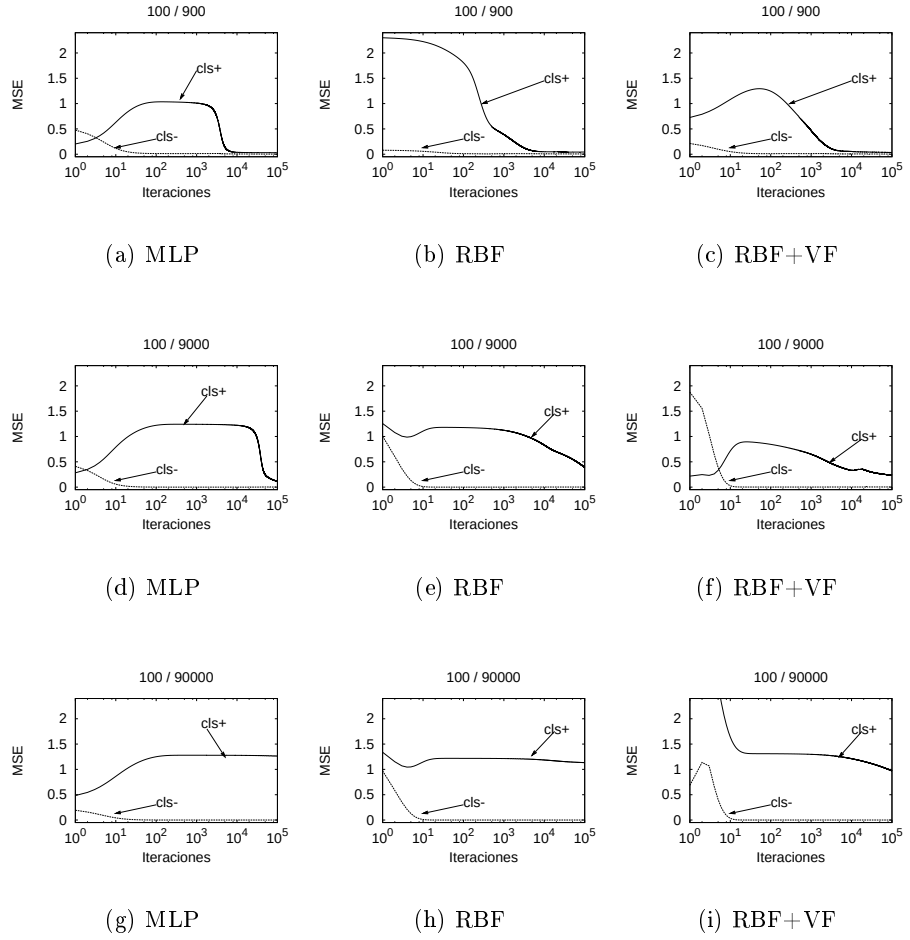
Estos resultados evidencian lo apuntado con anterioridad en el sentido de que el desbalance de las clases relentiza la convergencia de la clase menos representada.

3.3 Equilibrio de las aportaciones al MSE

Considerando que el problema del desbalance de la ME afecta negativamente al algoritmo back-propagation debido a la desproporción de las aportaciones al MSE (Ec. 3.5) por parte de las clases, se puede considerar la inclusión de una función de coste (γ) al algoritmo que compense este efecto como se muestra a continuación:

$$\begin{aligned} E(U) &= \sum_{k=1}^{K} \gamma(k) E_k = \gamma(1)E_1(U) + \gamma(2)E_2(U) \\ &= \frac{1}{N} \sum_{k=1}^{K} \gamma(k) \sum_{n=1}^{N_k} (\mathbf{d}^n - \mathbf{f}^n)^2 \,, \end{aligned} \tag{3.6}$$

de esta forma $\gamma(1)\|\nabla E_1(U)\| \approx \gamma(2)\|\nabla E_2(U)\|$ y puede evitarse que la clase minoritaria sea ignorada y que el entrenamiento este dominado por la clase mayoritaria.



(a) MLP (b) RBF (c) RBF+VF

(d) MLP (e) RBF (f) RBF+VF

(g) MLP (h) RBF (i) RBF+VF

Fig. 3.4: MSE por clase (de las tres bases de datos prototipo de la Fig. 3.3) generado por el MLP, la red RBF y la red RBF+VF. El error es calculado a partir de $E_k(U) = 1/N_k \sum_{n=1}^{N_k}(\mathbf{d}^n - \mathbf{f}^n)^2$ con la finalidad de facilitar el contraste del error de ambas clases.

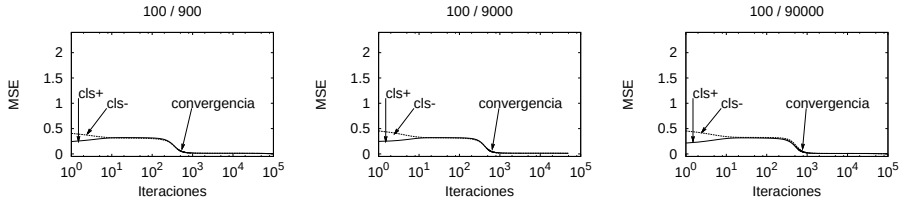
La forma más natural de la función de coste $\gamma(k)$ puede ser expresada como la razón entre el número de muestras de la ME y el de cada clase [Fu 2002]: $\gamma(k) = N/N_k$, donde N_k es el número de muestras de la clase k en la ME y N el total de muestras.

Para ilustrar los efectos de incluir la función de coste $\gamma(k)$ al algoritmo de apren-

dizaje, se realizaron una serie de experimentos con las bases de datos artificiales comentadas en la sección 3.2.

La Fig. 3.5 muestra los resultados de esta experimentación. En esta figura se evidencian los beneficios de compensar las aportaciones al MSE durante el proceso de entrenamiento del MLP. Obsérvese que al incluir la función de coste $\gamma(k)$, la cantidad de iteraciones necesarias para alcanzar la convergencia se redujo a menos de 1000 iteraciones en las tres bases de datos. Esto se ve reflejado en una importante mejora en la eficiencia de la ANN.

En la sección 3.2 se vio que cuando el desbalance de la ME no es tratado en el mejor de los casos y con estas bases de datos se requirieron de al menos 10000 iteraciones para alcanzar la convergencia (Fig. 3.4a, 3.4d y 3.4g).



(a) MLP (b) MLP (c) MLP

Fig. 3.5: Comportamiento del MSE por clase al incluir una función de coste $\gamma(k)$ al proceso de entrenamiento. El error es calculado a partir de $E_k(U) = 1/N_k \sum_{n=1}^{N_k}(\mathbf{d}^n - \mathbf{f}^n)^2$ con la finalidad de facilitar el contraste del error de ambas clases.

3.3.1 Opciones para tratar el desbalance

En este apartado se describen y explican las funciones de coste utilizadas para este estudio. Se centran en la idea de buscar un balance en las aportaciones de error.

- **Opción 0**: $\gamma(k) = 1$. Algoritmo back-propagation sin ninguna modificación.
- **Opción 1**: $\gamma(k) = N_{max}/N_k$; donde $k = 1, ..., K$; K es el total de clases, N_{max} es el número de muestras de la clase mayoritaria y N_k el de la clase k.
- **Opción 2**: $\gamma(k) = N/N_k$, donde N es el número total de muestras.

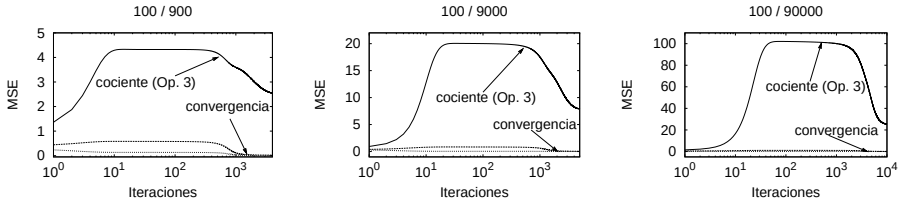
- **Opción 3**: $\gamma(k) = \frac{\|\nabla E_k(U)\|}{\|\nabla E_{max}(U)\|}$, donde $\|\nabla E_{max}(U)\|$ corresponde a la clase mayoritaria. Esta función esta basada en una modificación [Alejo 2008] de la propuesta de [Anand 1993].

La idea de utilizar estas funciones de coste reside en dos aspectos fundamentales: a) El valor de cada función es obtenido automáticamente[5] y b) las funciones son fáciles de implementar. Sin embargo, teóricamente es posible demostrar que incluir la función de coste $\gamma(k)$ al proceso de aprendizaje de la ANN altera la probabilidad de la distribución de los datos [Lawrence 1998]. Esta consideración debe de ser tomada en cuenta según la trascendencia de conservar las probabilidades a priori de los datos.

Se puede observar que la Opción 3 tiene la tendencia a disminuir su impacto en la distribución de los datos debido a que al disminuir el error de ambas clases en proporciones semejantes, el cociente $\|\nabla E_k(U)\|/\|\nabla E_{max}(U)\|$ es reducido.

En la Fig. 3.6 se presenta el valor del cociente $\|\nabla E_k(U)\|/\|\nabla E_{max}(U)\|$ en relación a la convergencia del MLP[6]. Se evidencia que inicialmente el valor de la Opción 3 es incrementado y posteriormente es mantenido hasta que la ANN comienza a converger. Cuando esto ocurre, el valor de la función de costo empieza a decrecer a consecuencia de la reducción del MSE de la clase minoritaria. En la Fig. 3.6a se puede ver con mayor claridad este proceso.

Vemos que el valor de la Opción 3 es directamente proporcional al desbalance de las clases. En otras palabras, cuanto más es la desproporción de elementos de las clases, mayor es el valor de esta función de costo.



(a) MLP (b) MLP (c) MLP

Fig. 3.6: Comportamiento del MSE por clase y el cociente $\frac{\|\nabla E_k(U)\|}{\|\nabla E_{max}(U)\|}$ al incluir la Opción 3 al proceso de entrenamiento. El eje x ha sido escalado logarítmicamente para resaltar los cambios en las curvas de error.

[5]Lo que las hace atractivas cuando se tiene poca información del problema a resolver.

[6]Observe que estos valores corresponden a las bases de datos sintéticas de la sección 3.2.

3.4 Caso de estudio: Problemas de dos clases

Hasta ahora sólo se han discutido los beneficios de equilibrar las aportaciones al MSE durante el entrenamiento de la ANN en conjuntos de datos simulados. Por lo tanto, es necesario ampliar el estudio del efecto del desbalance de las clases y su posible tratamiento (opciones propuestas en la sección 3.3.1) a dominios de bases de datos reales.

Para ello, se discutirán tres bases de datos de estudio (V2Cls, B2Cls y Phomene) que tratan de reflejar el comportamiento común del problema del desbalance de las clases en el contexto de las ANNs entrenadas con el algoritmo back-propagation y problemas de dos clases.

En términos generales se pueden describir los siguientes escenarios:

1. **V2Cls**. Ambas clases convergen en un intervalo de tiempo "razonable".

2. **Phoneme**. El MSE de la clase minoritaria desciende pero no logra la convergencia mientras que el de la mayoritaria converge en las primeras iteraciones. No obstante, se alcanza un relativo equilibrio entre el error de la clase mayoritaria y la clase minoritaria.

3. **B2Cls**. La clase mayoritaria converge al inicio del entrenamiento y el MSE de la clase minoritaria desciende muy lentamente.

La principal diferencia entre estos tres casos se encuentra en la lentitud de la convergencia de la clase minoritaria en las diferentes bases de datos.

Las bases de datos V2Cls, B2Cls y Phoneme disponen de diferentes características en cuanto a representatividad en el espacio, nivel de separabilidad y desbalance (véase la Tabla 3.1). V2Cls y B2Cls corresponden a una modificación de los conjuntos de datos Vowel y Balance. La modificación consistió en convertir cada base de datos de múltiples clases en otra de dos clases.

En Vowel se tomó la clase 1 como minoritaria y el resto de las clases formaron la clase mayoritaria para dar como resultado V2Cls. Para obtener B2Cls se ocupo la clase menos representada de Balance como minoritaria y las otras 2 se identificaron como mayoritaria. De esta forma se obtuvieron bases de datos desbalanceadas de dos clases. Phoneme es una base de datos de dos clases por lo que no sufrió ninguna modificación. Los tres conjuntos de datos fueron extraídos del *UCI Machine Learning Repository* [Newman 1998].

En la Tabla 3.1 se presentan algunas de las características más relevantes de estas bases de datos.

F1 o criterio de Fisher es una medida geométrica de solapamiento que calcula la separabilidad entre dos clases [Sánchez 2007] en función de una característica específica.

$$F1 = \frac{\|m_1 - m_2\|^2}{\sigma_1^2 + \sigma_2^2}, \tag{3.7}$$

donde m representa una media, σ^2 representa una variación, y los subíndices denotan las dos clases.

En la Tabla 3.1 cada conjunto de datos presenta diferentes valores de F1. Estos diferentes grados de separabilidad entre las distribuciones de las clases se corresponde con un mayor o menor nivel de dificultad para ser aprendidas por la ANN.

Tabla 3.1: Características relevantes de las bases de datos. F1 corresponde al criterio de Fisher. Valores grandes de F1 indican alta separabilidad entre clases y valores pequeños corresponde a baja separabilidad.

Datos	Atributos	Distribución de las clases	Razón	Fisher (F1)
V2Cls	10	48/480	0.100	1.812
Phoneme	5	1586/3818	0.415	0.285
B2Cls	4	49/576	0.085	0.001

La idea de hacer uso de bases de datos de dos clases es la de simplificar la interpretación del MSE por clase. Más adelante, el estudio es generalizado a problemas de múltiples clases.

Los detalles experimentales de las pruebas realizadas en este capítulo se pueden consultar en la sección 2.10. Así mismo, en el Apéndice C se encontra información relevante de los resultados experimentales obtenidos con otras bases de datos.

3.4.1 Estudio sobre las bases de datos V2Cls, Phoneme y B2Cls

En la Fig. 3.7 se ilustra el MSE por clase obtenido en la fase de aprendizaje de los modelos de red RBF, RBF+VF y MLP, entrenados con el algoritmo back-propagation[7] y las bases de datos V2Cls, Phoneme y B2Cls.

Nótese que en las tres bases de datos se muestran comportamientos en MSE muy diferentes entre sí. En el caso de V2Cls el desbalance de la ME relentiza la convergencia de la ANN[8], pero al final esta es alcanzada, y por lo tanto no afecta la efectividad del clasificador sobre la clase menos representada (Fig. 3.7a-c).

[7] Para mayor detalle sobre estos modelos de ANN véase el capítulo 2.

[8] Al igual que ocurrió con las bases de datos artificiales de la sección 3.2.

Por otro lado en Phoneme la convergencia del MSE de la clase minoritaria es muy lenta y no se logra en un periodo de 25000 iteraciones[9] (Fig. 3.7d-f). Sin embargo, los resultados en Phoneme no son tan dramáticos como en B2Cls donde el efecto del desbalance de la ME causa que la reducción del MSE sea prácticamente insignificante a lo largo de las 25000 iteraciones (Fig. 3.7g-i).

Este problema aumenta al existir un alto nivel de solapamiento entre clases. Cuanto menor es la separabilidad más difícil es establecer una frontera de decisión capaz de discriminar correctamente las clases, y esto se agrava cuando la ME está desbalanceada (para mayor detalle véase el Anexo A).

Para tratar de medir la separabilidad de los datos y relacionarla con el comportamiento del MSE cuando existe desbalance en las clases de la ME, se utilizó el criterio de medida de Fisher (F1). Obsérvese en la Tabla 3.1 que valor del criterio F1 para B2Cls es muy cercano a cero mientras que para V2Cls es de 1.812.

Por otra parte, como cabría esperar en términos de clasificación en la base de datos V2Cls, se observa un buen desempeño en los tres modelos de red (Tabla 3.2). Tanto en la PC (Precisión en la Clasificación) como en la *g-mean* se tienen porcentajes cercanos al 100%. Valores altos de *g-mean* indican una buena efectividad del clasificador en ambas clases (véase la sección 2.9.3).

En Phoneme y B2Cls (Fig. 3.7) no se alcanza la convergencia de la clase minoritaria y su MSE decrece muy lentamente. Esto trae como consecuencia un desempeño deficiente del clasificador sobre la clase minoritaria. El porcentaje de aciertos para esta clase es mucho menor que el de la clase mayoritaria viéndose reflejado en valores bajos de la *g-mean*.

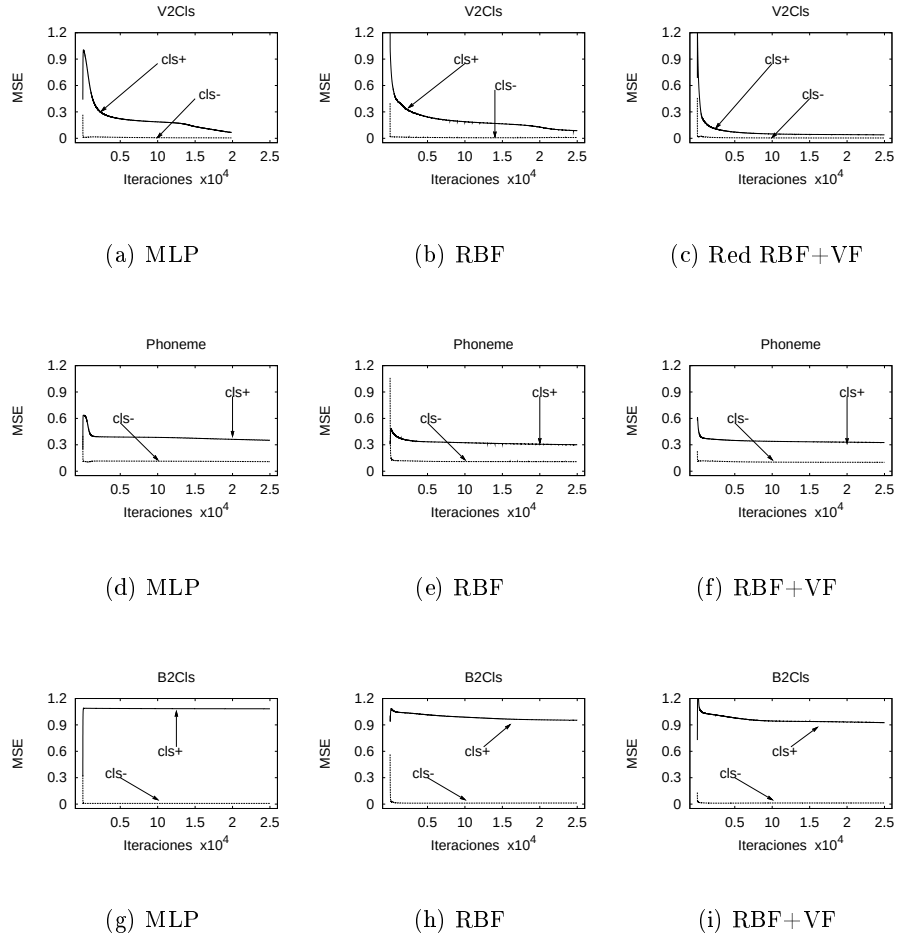
En el caso de Phoneme el desbalance de la ME no perjudica radicalmente el desempeño del clasificador sobre la clase minoritaria. Se tienen valores pequeños de *g-mean* (ver Tabla 3.2) pero no al nivel que ocurre con B2Cls. En esta última base de datos, el porcentaje de aciertos para la clase menos representada es de 0.0%. Esto indica que fue ignorada durante el proceso de entrenamiento. Sin embargo, la PC para B2Cls es superior al 90%.

Así, valores altos de PC no implican un buen desempeño del clasificador sobre ambas clases[10]. Esto es común en bases de datos no balanceados donde la distribución de los datos está muy desequilibrada.

A continuación, se estudiaran los posibles beneficios de tratar el problema del desbalance con las opciones presentadas en la sección 3.3.1.

[9] Criterio de parada establecido para la ANN.

[10] Porque solamente esta aprendiendo a la clase mayoritaria aunque de manera muy pobre.



(a) MLP (b) RBF (c) Red RBF+VF

(d) MLP (e) RBF (f) RBF+VF

(g) MLP (h) RBF (i) RBF+VF

Fig. 3.7: MSE por clase en la fase de entrenamiento de las bases de datos V2Cls, Phoneme y B2Cls. Los resultados corresponden a la Opción 0 (algoritmo back-propagation estándar).

3.4.2 Caso 1: Tratamiento del desbalance en V2Cls

En la Fig. 3.8 se observa que la inclusión de las opciones 1, 2 y 3 al algoritmo de entrenamiento acelera la convergencia de la red de forma considerable. Obsérvese que se requieren de menos de 5000 iteraciones para lograr la convergencia (sobre todo en el MLP y la red RBF+VF) mientras que con el algoritmo de entrenamiento sin

Tabla 3.2: Desempeño de las ANNs en la fase de clasificación de las bases de datos V2Cls, Phoneme y B2Cls. Los valores entre paréntesis hacen referencia a la desviación estándar.

V2Cls	MLP	RBF	RBF+VF
PC	99.43(0.52)	98.67(0.85)	99.81(0.42)
g-mean	99.69(0.28)	95.41(4.25)	99.9(0.23)

Phoneme	MLP	RBF	RBF+VF
PC	80.01(1.41)	79.5(1.26)	80.27(1.53)
g-mean	74.58(2.2)	75.21(1.88)	75.59(2.07)

B2Cls	MLP	RBF	RBF+VF
PC	92.16(0.36)	91.36(1.54)	92.0(0.57)
g-mean	0.00(0.00)	0.00(0.00)	0.00(0.00)

modificar (Opción 0) se requerían de al menos 20000 iteraciones aproximadamente (excepto la red RBF+VF que convergió alrededor de las 10000 iteraciones, véase Fig. 3.8c). Sin embargo, es notable que la red RBF muestra una mayor dificultad para reducir el MSE de la clase minoritaria.

Las tres ANNs aplicadas sobre V2Cls alcanzan la convergencia tanto con el algoritmo de entrenamiento sin modificar, como con la inclusión de las opciones 1, 2 y 3, lo que se significa (en términos de PC y valores de *g-mean*) que los resultados obtenidos en la fase de clasificación son prácticamente los mismos (valores cercanos al 100%). No se observan ni mejoras ni perjuicios en la clasificación (véase la Tabla 3.3).

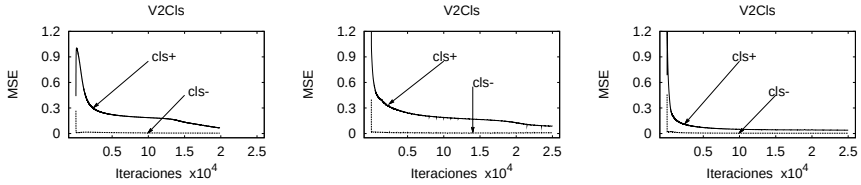
No obstante, las opciones 1-3 reducen el número de iteraciones necesarias para alcanzar la convergencia de la ANN cuando es entrenada con el algoritmo back-propagation con procesamiento por grupos y MEs desequilibradas.

V2Cls es una base de datos con un alto nivel de separabilidad lo que le permite converger más fácilmente al clasificador. Sin embargo, no todos los conjuntos no balanceados presentan esta característica. La siguiente base de datos (Phoneme) muestra características distintas lo que se ve reflejado en la convergencia de la red.

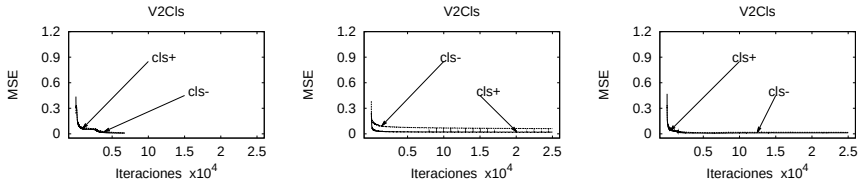
3.4.3 Caso 2: Tratamiento del desbalance en Phoneme

En la Fig. 3.9 se observa que en ninguno de los tres modelos de ANNs se logra la convergencia con el conjunto de datos Phoneme.

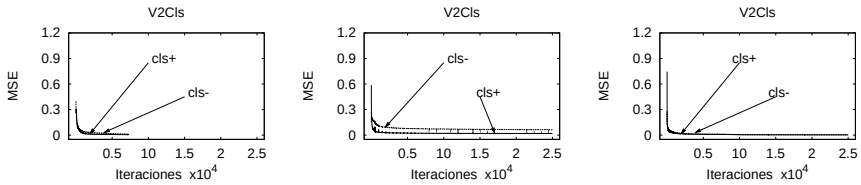
Sin embargo, el comportamiento del MSE por clase cuando las funciones de coste $\gamma(k)$ son incluidas modifican de forma significativa el proceso de entrenamiento de las ANNs.



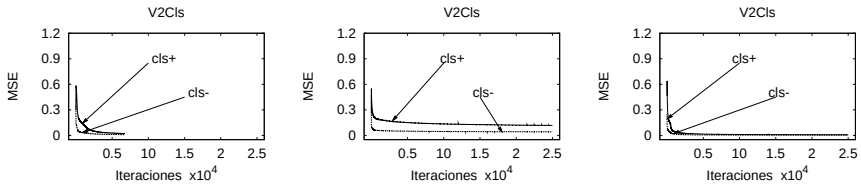
(a) MLP Op. 0 (b) RBF Op. 0 (c) RBF+VF Op. 0



(d) MLP Op. 1 (e) RBF Op. 1 (f) RBF+VF Op. 1



(g) MLP Op. 2 (h) RBF Op. 2 (i) RBF+VF Op. 2



(j) MLP Op. 3 (k) RBF Op. 3 (l) RBF+VF Op. 3

Fig. 3.8: MSE por clase en la fase de entrenamiento de la base de datos V2Cls.

Tabla 3.3: Desempeño en la fase de clasificación de la base de datos V2Cls. Los valores entre paréntesis hacen referencia a la desviación estándar

MLP	Opción 0	Opción 1	Opción 2	Opción 3
PC	99.43(0.52)	99.62(0.51)	99.62(0.52)	99.43(0.52)
g-mean	99.69(0.28)	99.79(0.28)	99.79(0.28)	99.69(0.28)
RBF	Opción 0	Opción 1	Opción 2	Opción 3
PC	98.67(0.85)	99.24(1.04)	98.86(1.57)	98.3(2.05)
g-mean	95.41(4.25)	99.58(0.58)	99.37(0.87)	99.05(1.14)
RBF+VF	Opción 0	Opción 1	Opción 2	Opción 3
PC	99.81(0.42)	99.62(0.52)	99.43(0.85)	99.62(0.52)
g-mean	99.9(0.23)	99.79(0.28)	99.69(0.47)	99.79(0.28)

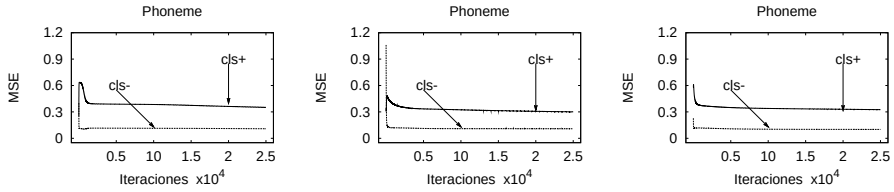
Observe que al aplicarse las opciones 1, 2 y 3, el MSE de la clase minoritaria es disminuido considerablemente en relación al MSE original de esta clase (véase la Fig.3.9a-c). Sin embargo, la convergencia de la ANN no se consigue pero se puede ver una razonable participación de ambas clases en el proceso de entrenamiento. Este hecho trae como consecuencia mejoras en la efectividad del clasificador sobre la clase menos representada en la ME (véase en la Tabla 3.4 los incrementos que se obtienen en los valores de *g-mean*).

Por otro lado puede verse en la Fig. 3.9, que no existe una diferencia significativa entre los resultados (en MSE) obtenidos por la aplicación de las opciones 1, 2 y 3.

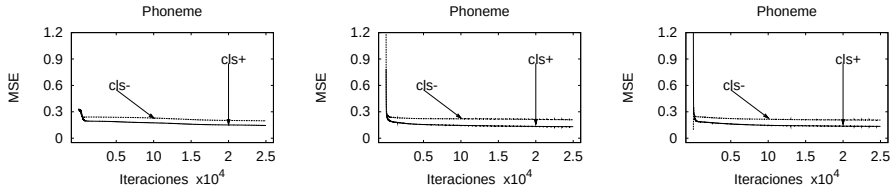
Otro aspecto interesante que se aprecia en la Fig. 3.9 del inciso d al i (opciones 1 y 2) es el incremento en el MSE de la clase mayoritaria. El efecto de las opciones 1 y 2 se ve reflejado en el aumento del MSE de la clase mayoritaria. Este comportamiento se ha observado en otras bases de datos de dos clases, y disminuye la precisión de esta clase a causa del aumento y reducción de la influencia de las clases minoritaria y mayoritaria respectivamente, en el entrenamiento de la red.

En la Tabla 3.4 se aprecia que con el modelo MLP no hubo una diferencia importante entre los resultados de PC obtenidos. Esto significa que no fue afectada la precisión del clasificador por la aplicación de las opciones 1, 2 y 3. En cuanto a los valores de *g-mean* se observan mejoras significativas al aplicar las tres opciones, es decir, se incrementó el porcentaje de aciertos de la clase minoritaria.

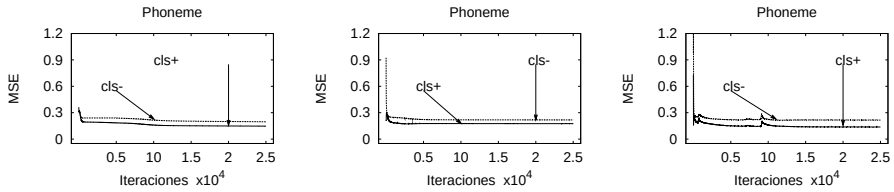
La red RBF presenta un comportamiento semejante al no mostrar diferencias importantes en la PC. Los valores de *g-mean* fueron incrementados al aplicar las opciones 1, 2 y 3. De igual manera, el modelo red RBF+VF no mostró diferencia significativa en cuanto a precisión y los valores de *g-mean* fueron incrementados de forma importante.



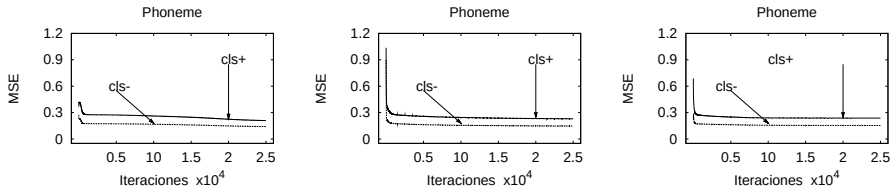
(a) MLP Op. 0 (b) RBF Op. 0 (c) RBF+VF Op. 0



(d) MLP Op. 1 (e) RBF Op. 1 (f) RBF+VF Op. 1



(g) MLP Op. 2 (h) RBF Op. 2 (i) RBF+VF Op. 2



(j) MLP Op. 3 (k) RBF Op. 3 (l) RBF+VF Op.3

Fig. 3.9: MSE por clase en la fase de entrenamiento de la base de datos Phoneme.

La tendencia a reducir la PC global por la aplicación de las opciones 1, 2 y 3 es resultado de la disminución de la PC de la clase mayoritaria. No obstante, el incremento de la PC de la minoritaria evita que esta reducción sea importante. En el caso del MLP esta reducción es prácticamente insignificante.

En términos generales se observó que al aplicar las opciones 1, 2 y 3 se incrementó la precisión del clasificador sobre la clase minoritaria, y se acelero la reducción del MSE de esta clase como lo evidenciaron los resultados de la Tabla 3.4 (especialmente los resultados reflejados por el criterio de medida *g-mean*).

Tabla 3.4: Desempeño en la fase de clasificación de la base de datos Phoneme. Los valores entre paréntesis hacen referencia a la desviación estándar.

MLP	Opción 0	Opción 1	Opción 2	Opción 3
PC	80.01(1.41)	80.33(1.68)	80.87(1.87)	81.35(1.84)
g-mean	74.58(2.2)	82.16(1.6)	82.59(1.87)	80.62(2.17)
RBF	Opción 0	Opción 1	Opción 2	Opción 3
PC	79.5(1.26)	77.07(2.27)	78.81(1.17)	79.55(1.46)
g-mean	75.21(1.88)	80.19(2.01)	81.62(1.5)	79.81(1.45)
RBF+VF	Opción 0	Opción 1	Opción 2	Opción 3
PC	80.27(1.53)	78.94(2.24)	79.22(1.97)	80.18(1.46)
g-mean	75.59(2.07)	81.45(1.7)	81.82(1.89)	80.02(1.22)

La base de datos Phoneme es un ejemplo claro donde el desbalance afecta el desempeño del clasificador pero no de forma radical. La clase menos representada es identificada en menor medida que la mayoritaria pero no hasta el punto de llegar a ser ignorada su participación en el entrenamiento de la ANN. Así, su valor de F1 = 0.285 es muy por debajo del mostrado por V2Cls (F1 = 1.812) y su nivel de desbalance no es tan considerable como en el caso de V2Cls o B2Cls (véase la Tabla 3.1).

3.4.4 Caso 3: Tratamiento del desbalance en B2Cls

A diferencia de V2Cls y Phoneme, B2Cls (Fig. 3.10) presenta un comportamiento muy diferente. Se observa en la Fig. 3.10a-c que el MSE de la clase minoritaria se incrementó en mayor medida y decreció más lentamente en comparación con las bases de datos anteriores.

Este hecho trajo como consecuencia que la clase minoritaria fuera ignorada durante el proceso de entrenamiento, y por lo tanto el porcentaje de aciertos para esta clase fuera cero, con valores de *g-mean* nulos (véase la Tabla 3.5).

En el caso particular del MLP se observan mejoras muy significativas cuando las opciones 1, 2 y 3 son aplicadas durante el proceso de entrenamiento. Se muestra en la Fig. 3.10 como el MSE de la clase minoritaria es reducido de manera importante.

En términos de PC con el modelo MLP no existe diferencia entre los resultados obtenidos con las opciones 0, 1, 2 y 3. Sin embargo, en los valores de *g-mean* se producen incrementos sustanciales. Obsérvese que con la Opción 0 se obtienen valores de *g-mean* iguales a cero. Sin embargo, al aplicarse las opciones 1, 2 y 3 el porcentaje de aciertos para esta clase se incrementó en promedio en torno al 90% (véase la Tabla 3.5).

Las redes RBF y RBF+VF (Fig. 3.10) presentan resultados muy similares entre ellas. Al igual que ocurrió en otros casos, las opciones 1, 2 y 3 incrementan la velocidad de convergencia de la clase menos representada, aunque no se tienen resultados de similar importancia como en el caso del MLP. La Opción 3 (en ambos casos) mostró serios problemas en la reducción del MSE de la clase minoritaria, y en consecuencia un desempeño deficiente en la clasificación de esta clase.

En la Tabla 3.5 se observa que con la red RBF y la red RBF+VF el porcentaje de aciertos de la clase minoritaria se incrementó[11] al aplicarse las opciones 1 y 2, pero también es notable la reducción de la PC del clasificador.

Tabla 3.5: Desempeño en la fase de clasificación de la base de datos B2Cls. Los valores entre paréntesis hacen referencia a la desviación estándar

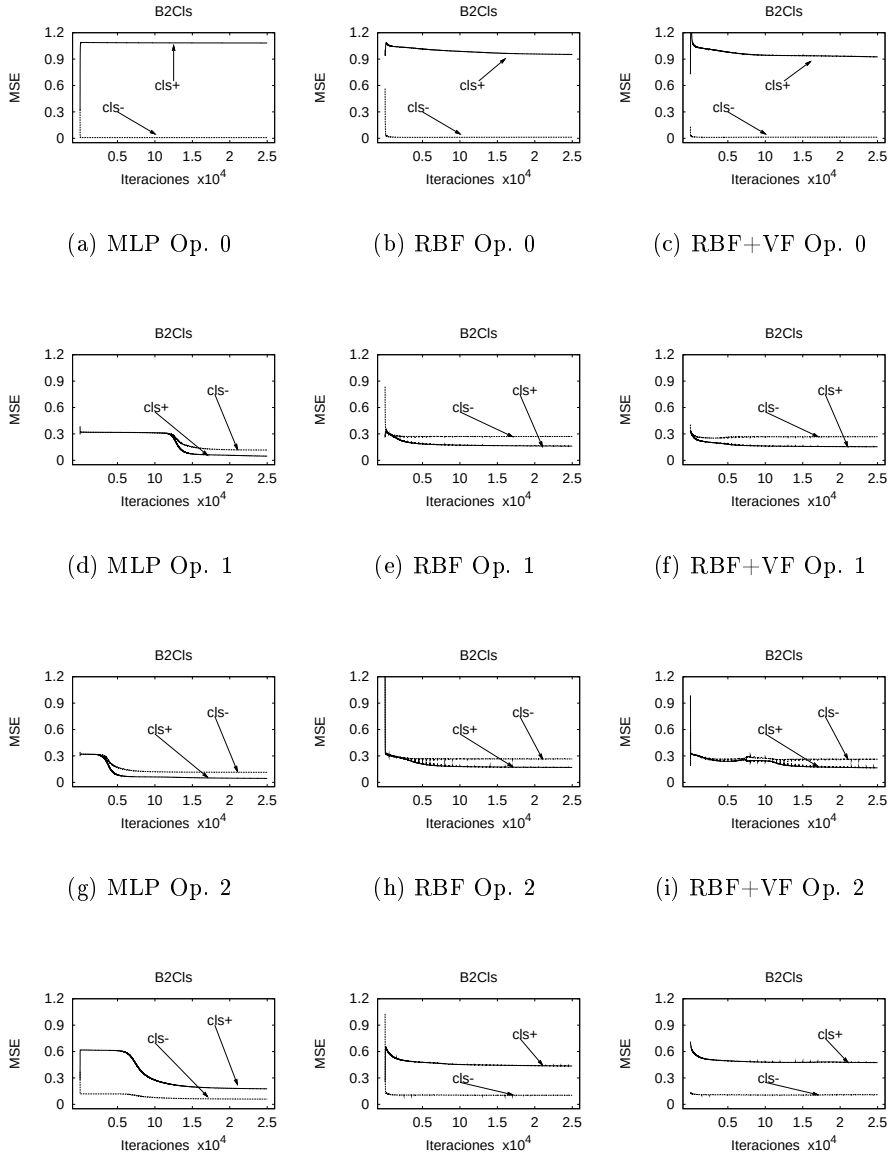
MLP	Opción 0	Opción 1	Opción 2	Opción 3
PC	92.16(0.36)	90.88(1.66)	91.36(1.99)	92.8(2.33)
g-mean	0.0(0.0)	91.27(5.71)	91.54(5.79)	92.33(6.13)
RBF	Opción 0	Opción 1	Opción 2	Opción 3
PC	91.36(1.54)	69.28(4.58)	65.28(4.26)	86.4(6.22)
g-mean	0.0(0.0)	72.62(9.74)	71.01(4.2)	18.47(27.3)
RBF+VF	Opción 0	Opción 1	Opción 2	Opción 3
PC	92.0(0.57)	64.64(9.97)	66.56(4.61)	86.08(6.46)
g-mean	0.0(0.0)	65.98(10.04)	74.2(5.03)	18.91(17.26)

Nótese que el comportamiento de las opciones 1, 2 y 3 en el MLP es muy distinto al mostrado en las redes RBF y RBF+VF.

El efecto de las opciones en estos dos últimos modelos de ANNs sobre la clase mayoritaria fue negativo al reducir el PC de esta clase.

Sin embargo, la clase minoritaria fue beneficiada por estas opciones, aunque mínimamente en el caso de la Opción 3.

[11] Esto es reflejado en los incrementos de los valores de la *g-mean*.



(a) MLP Op. 0 (b) RBF Op. 0 (c) RBF+VF Op. 0

(d) MLP Op. 1 (e) RBF Op. 1 (f) RBF+VF Op. 1

(g) MLP Op. 2 (h) RBF Op. 2 (i) RBF+VF Op. 2

(j) MLP Op. 3 (k) RBF Op. 3 (l) RBF+VF Op. 3

Fig. 3.10: MSE por clase en la fase de entrenamiento de la base de datos B2Cls.

La causa más probable de esta situación sea la naturaleza de cada ANN. Mientras que para el MLP la transformación del espacio se realiza por atributo o característica, en las redes RBF y RBF+VF se realiza por prototipo. Esto produce que este tipo de redes sean más sensibles a las actualizaciones de los parámetros libres de la ANN sobre todo en los centros y varianzas. Cuando la separabilidad en los datos es menor[12] los cambios de una iteración a otra pueden ser de mayor magnitud lo que genera mayor inestabilidad en el aprendizaje de la red y las hacen más vulnerables a caer en mínimos locales.

Por otro lado deben considerarse las siguientes características particulares del conjunto de datos B2Cls:

1. Muestra niveles de separabilidad de los datos muy cercanos a cero.

2. Presenta un alto nivel de solapamiento. Al aplicar el clasificador 3-NN se observó que prácticamente todos los vecinos más próximos a cada uno de los elementos de la clase minoritaria pertenecían a la clase mayoritaria.

3. Los elementos pertenecientes a la clase minoritaria (aproximadamente un 8%) son pocos en relación a la clase mayoritaria (aproximadamente 92%), y además los datos que contiene están muy poco representados (véase la Tabla 3.1).

   Esto realmente no supone un problema si los elementos contenidos en la clase minoritaria son altamente discriminantes. En la Tabla 3.1, la base de datos V2Cls presenta características similares en cuanto a representación. Sin embargo, su comportamiento es diametralmente distinto y se debe a que V2Cls es altamente separable y no presenta solapamiento entre clases[13].

Cuando la base de datos presenta características como las de B2Cls, las ANNs basadas en funciones de base radial presentan una pobre capacidad en el aprendizaje dado que lo buscan es encontrar los parámetros que ajusten los datos a una distribución normal y posiblemente no sigan necesariamente esta distribución.

Por otro lado, el MLP utiliza únicamente la información contenida en los datos para establecer las fronteras de decisión, y por lo tanto en este contexto logra mejores resultados. Esto significa que ante el problema del solapamiento entre clases, el comportamiento de la red es distinto para un MLP respecto a una red RBF. Para mayor detalle sobre este tema véase el Anexo A. Así, si el desbalance de las clases no es tratado, los resultados no serían los obtenidos por las opciones 1, 2 y 3, sino que al contrario se tendría un aprendizaje pobre sobre la clase menos representada[14].

En síntesis, de este estudio se puede concluir lo siguiente:

[12] Recuérdese que B2Cls es la que presenta menor separabilidad en los datos.

[13] Esto fue observado cuando se le aplicó el clasificador 3-NN.

[14] Los clasificadores utilizados fueron: MLP entrenada con el back-propagation con procesamiento

- El desbalance de la ME origina lentitud en al convergencia de la clase menos representada.
- El problema es generado por las aportaciones desiguales al MSE durante el proceso de entrenamiento.
- Equilibrar las aportaciones al MSE (opciones 1, 2 y 3) durante el entrenamiento ayuda a acelerar la convergencia de la clase minoritaria.
- El desbalance de las clases afecta en mayor o menor medida dependiendo de la separabilidad de las clases.
- Cuando la separabilidad de las clases es menor, las redes RBF y RBF+VF son más sensibles al desbalance de la ME.
- La inclusión de funciones de coste al algoritmo de entrenamiento incrementa la influencia de la clase minoritaria en el proceso de aprendizaje. Así mismo, reduce la participación de la clase mayoritaria y esto trae como consecuencia la disminución en la PC de esta última.

El análisis del desbalance de las clases es un trabajo complejo y debe evaluarse desde diferentes contextos y dominios. En la siguiente sección se ampliará el estudio a problemas de múltiples clases y distintos niveles de desbalance entre clases. Además, se relacionará este efecto con cuestiones de separabilidad y solapamiento entre clases.

3.5 Caso de estudio: Problemas de múltiples clases

La mayor parte de los trabajos realizados para tratar el desbalance están dirigidos a problemas de dos clases y pocos lo han abordado en el contexto de múltiples clases [Zhou 2006]. En esta sección se estudia el problema del desbalance en dominios de múltiples clases y se analiza su efecto sobre el clasificador. Posteriormente se evalúan las posibilidades de las estrategias presentadas en la sección 3.3.1 para tratar el problema de las clases no balanceadas en contextos de múltiples clases.

En la sección anterior se mostró el efecto del desbalance de las clases en el rendimiento y efectividad de las redes entrenadas con el algoritmo back-propagation en dominios de dos clases. Se observó que causa lentitud en la convergencia de la clase menos representada. En este sentido se vio que el uso de funciones de coste (Opción 1, 2 y 3) para tratar el desbalance de la ME puede causar:

secuencial, el Nearest-neighbor-like, máquinas de vectores soporte, árboles de decisión y redes RBF entrenadas con regresión logística. Estos clasificadores fueron tomados de la herramienta WEKA [Ian 2005].

- Celeridad en la convergencia de la clase minoritaria.
- Incrementos en la PC de la clase minoritaria.
- Aumento del MSE de la clase mayoritaria.
- Reducciones en la PC de la clase mayoritaria.

En este estudio se analiza la viabilidad de generalizar el comportamiento observado en problemas de dos clases a problemas de múltiples clases. En esta sección se muestran los resultados sobre Ecoli6 y Cayo. Estas bases de datos son de especial interés por las siguientes razones:

- Son bases de datos de múltiples clases.
- Presentan un alto desbalance entre clases.
- Ecoli6 corresponde al contexto de las bases de datos poco representadas en la ME y Cayo al caso opuesto.

Más adelante en el Anexo C se presentan algunos resultados obtenidos con otras bases de datos de múltiples clases que dan soporte a las conclusiones presentadas en este capítulo. Para mayor detalle acerca de los aspectos experimentales consúltese la sección 2.10.

3.5.1 Base de datos Ecoli6

La base de datos Ecoli6 presenta diferentes niveles de desbalance entre clases. Se observa en la Tabla 3.6, que la clase 1 y 2 muestran un desbalance severo entre ellas. La clase mayoritaria tiene 143 muestras y la minoritaria sólo tiene 5. Sin embargo, entre las clases 5 y 6 el desbalance no es tan considerable. Mediante la razón se representa el tamaño que tiene cada clase respecto al total de muestras contenidas en la ME. Aquí nos enfrentamos al problema de identificar a las clases minoritarias o bajo que criterio decidir si corresponde a una clase minoritaria o no.

Al observar la Fig. 3.11 se puede ver claramente que las clases que inicialmente presentan una mayor lentitud en la convergencia de su MSE son las clases 1 y 5. Corresponde con el hecho de que son clases poco representadas en la ME (razón de 0.02 y 0.11 respectivamente). En especial la clase 1 muestra un comportamiento bastante diferente en relación al resto de las clases.

Contradictoriamente como se observa en la Tabla 3.6, aunque la clase 6 se encuentra menos representada en la ME que la clase 5, su MSE es mucho menor. Estos resultados sugieren que el desbalance presentado por las clases 5 y 6 no es la causa

Tabla 3.6: Características relevantes de la Base de datos Ecoli6

Clase	F1	Patrones	Razón	Atributos
1	52.65	5	0.02	7
2	10.96	143	0.42	7
3	10.66	77	0.24	7
4	10.96	52	0.15	7
5	10.33	35	0.11	7
6	9.93	20	0.06	7

de la lentitud en la convergencia de estas clases, sino de otros factores como su solapamiento entre ellas. Obsérvese en la Tabla 3.7 que la clase 5 presenta un alto grado de solapamiento con la clase 3.

Tabla 3.7: Los resultados de esta tabla representan el porcentaje de elementos identificados por el clasificador no paramétrico k nearest-neighbours (k-NN) para $k = 3$. Los resultados fuera de la diagonal se pueden interpretar como el grado de solapamiento entre clases.

Clase	1	2	3	4	5	6
1	**100**	0.00	0.00	0.00	0.00	0.00
2	0.00	**97.9**	0.00	2.10	0.00	0.00
3	1.30	5.19	**80.52**	0.00	12.99	0.00
4	0.00	11.54	1.92	**84.62**	0.00	1.92
5	2.86	2.86	42.86	0.00	**51.43**	0.00
6	5.00	5.00	0.00	10.00	0.00	**80.00**

El comportamiento de la clase 1 confirma lo mencionado con anterioridad en el sentido de que las clases menos representadas presentan una mayor dificultad para converger. Sin embargo, cuando se tiene una "alta" separabilidad con el resto de las clases, la convergencia es alcanzada en un periodo de tiempo "razonable", como ocurre con esta clase que presenta un valor de F1 = 52.648 (véase la Fig. 3.11).

Obsérvese que en los tres modelos de ANNs, la clase 2 se converge en las primeras iteraciones. Esto coincide con el hecho de que es la clase más representada en la ME (razón de 0.42).

En la Tabla 3.8 se presenta el desempeño de la red en la fase de clasificación con la Opción 0. Al analizar estos resultados se observa que no existe contradicción alguna con lo discutido con anterioridad. En el caso de la clase 1 se observa un alto porcentaje de aciertos (véase la Tabla 3.8).

Como se vio en la Fig. 3.11 la convergencia de esta clase fue lenta pero al final se alcanzó. Este comportamiento es diferente al que se observa en la clase 5 (ver Fig.

3.11), donde se muestra una mayor dificultad para converger, y por tanto presenta el menor porcentaje de aciertos (ver Tabla 3.8).

Tabla 3.8: Resultados obtenidos en la fase de clasificación con la base de datos Ecoli6 y la Opción 0.

	Clase	F1	Ratio	PC	% confusión (> 10 %)
MLP	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	93.61	
	cls 3	10.66	0.23	82.96	cls 5 (11.91)
	cls 4	10.96	0.16	86.33	
	cls 5	10.33	0.11	64.44	cls 3 (30.09)
	cls 6	9.93	0.06	85.64	
RBF	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	91.46	
	cls 3	10.66	0.23	84.39	cls 5 (12.43)
	cls 4	10.96	0.16	87.18	
	cls 5	10.33	0.11	62.54	cls 3 (34.60)
	cls 6	9.93	0.06	85.56	cls 4 (10.00)
RBF+VF	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	92.13	
	cls 3	10.66	0.23	81.35	cls 5 (12.84)
	cls 4	10.96	0.16	87.55	
	cls 5	10.33	0.11	61.01	cls 3 (35.12)
	cls 6	9.93	0.06	85.42	

En la Tabla 3.8 las tres arquitecturas de ANNs presentaron resultados muy semejantes en la fase de clasificación. Es notable que las clases que generaron un mayor nivel de confusión fueron las clases 3, 5 y 6. Estos resultados tienen poco que ver con el desbalance de las clases y más con problemas de solapamiento (véase las Tablas 3.7 y 3.9).

En la Tabla 3.6 los valores de F1 globales no dan suficiente información como para poder afirmar que las clases 3 y 5 presentan un nivel bajo de separabilidad. Para profundizar en nuestro análisis se realizó un estudio de la separabilidad por pares de clases (ver Tabla 3.9). Se observa claramente que las clases con menor índice de separabilidad son las clases 3 y 5. Así mismo, la clase que presenta mayor separabilidad es la clase 1.

En las Fig. 3.12, 3.13 y 3.14 se presenta el comportamiento del MSE por clase al aplicar las opciones 1, 2 y 3 durante el entrenamiento de la red. A nivel de terminología se escribe Op. como abreviatura de Opción. En el eje x se representa el número de iteraciones, mientras que el eje y se muestra el MSE por clase.

En términos generales se puede advertir que las opciones 1, 2 y 3 ayudan a acelerar la convergencia de las clases menos representadas en la ME (clases 1 y 5, en especial la clase 1).

Tabla 3.9: Valores de F1 presentados en forma de matriz de confusión. El objetivo es mostrar los valores de F1 por cada par de clases. Recuérdese que a mayor magnitud de F1 más separables son las clases.

Clase	1	2	3	4	5	6
1	—	$3.77\ 10^{14}$	75.30	$6.31\ 10^{14}$	34.70	18.35
2	$3.77\ 10^{14}$	—	10.14	**5.06**	13.79	8.03
3	75.30	10.14	—	**5.37**	**1.24**	7.67
4	$6.31\ 10^{14}$	**5.06**	**5.37**	—	7.86	**5.66**
5	34.70	13.79	**1.24**	7.86	—	13.66
6	18.35	8.03	7.67	**5.66**	13.66	—

Por otro lado, en la clase 3 se incrementa el MSE al aplicar las opciones 1 y 2, mientras que la clase 2 queda afectada mínimamente cuando se incluye las funciones de coste en el proceso de entrenamiento de la ANN. Lo mismo ocurre con la clase 4 en las redes RBF y RBF+VF.

La razón de ello es que aumenta el nivel de confusión entre clases cuando tienen un alto nivel de solapamiento. Un ejemplo de esta situación se puede ver en el incremento de la confusión de la clase 3 con respecto de la clase 5.

Los resultados evidencian que la Opción 3 no genera efectos negativos sobre el MSE de las clases más representadas (ver Fig. 3.12, 3.13 y 3.14) independientemente de si están solapadas o no. No obstante, sus resultados en términos de reducción del MSE siempre son iguales o peores que las opciones 1 y 2. La posible ventaja de la Opción 3 sobre el resto es que no se incrementa el MSE a consecuencia del solapamiento entre clases.

En resumen, es claro el efecto de las opciones 1, 2 y 3 para tratar el desbalance de las clases en cuestiones de convergencia, pero ahora habrá que preguntarse ¿cuál es su efecto sobre la efectividad de la ANN en la fase de clasificación?.

En el caso específico del MLP (ver Tabla 3.10) se observa que sólo hubo mejoras en la PC de la clase 5, mientras que en el resto de las clases se mantuvo o se redujo su PC. Así, la confusión entre clases se incrementó al aplicar las opciones 1, 2 y 3.

Una explicación a este hecho es que la base de datos Ecoli6 está poco representada en el espacio, y la introducción de las funciones de coste puede aumentar la confusión de las clases mayoritarias, y por tanto modificar de forma significativa su precisión.

El efecto que se produce es parecido a cuando se modifica de forma artificial las probabilidades a priori de las clases en un clasificador bayesiano desplazando sus fronteras de decisión. De esta forma intentando balancear las clases minoritarias se genera un efecto secundario claro que modifica de forma significativa lo que entiende la red como la distribución de los datos de las clases mayoritarias, y que objetivamente

disminuye la precisión de las mismas. Este efecto se manifiesta de forma más clara cuanto mayor sea el grado de solapamiento en la frontera entre pares de clases.

Tabla 3.10: Resultados obtenidos al clasificar la base de datos Ecoli6 con el MLP.

	Clase	F1	Ratio	PC	% confusión (> 10 %)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	93.61	
Opción 0	cls 3	10.66	0.23	82.96	cls 5 (11.91)
	cls 4	10.96	0.16	86.33	
	cls 5	10.33	0.11	64.44	cls 3 (30.09)
	cls 6	9.93	0.06	85.64	
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	87.38	
Opción 1	cls 3	10.66	0.23	77.06	cls 5 (17.24)
	cls 4	10.96	0.16	77.06	cls 6 (14.12)
	cls 5	10.33	0.11	68.51	cls 3 (23.91)
	cls 6	9.93	0.06	80.10	cls 4 (14.80)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	88.78	
Opción 2	cls 3	10.66	0.23	80.47	cls 5 (14.82)
	cls 4	10.96	0.16	77.55	cls 6 (13.88)
	cls 5	10.33	0.11	65.96	cls 3 (27.36)
	cls 6	9.93	0.06	80.85	cls 4 (13.83)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	88.13	
Opción 3	cls 3	10.66	0.23	82.54	cls 5 (12.45)
	cls 4	10.96	0.16	85.57	
	cls 5	10.33	0.11	66.37	cls 3 (28.57)
	cls 6	9.93	0.06	82.81	cls 4 (12.50)

Por otro lado, en las redes RBF y RBF+VF (Tablas 3.11 y 3.12) se observan mejoras en la PC de la clase 5, mientras que en resto de las clases la PC fue disminuida o no fue afectada de manera importante. El comportamiento mostrado en la clase 3 y 5 obedece a las mismas causas descritas para el MLP.

Un ejemplo de esta situación puede observarse entre las clases 4 y 6, cuyo nivel de separabilidad en F1 es 5.663 (ver Tabla 3.9), y que aun siendo baja se consigue un nivel de precisión aceptable y sólo alcanza una confusión del 10.00 % en la Opción 0 de la clase 6 con la clase 4 en la red RBF. Esto es normal ya que la clase 6 tiene prácticamente un tercio de puntos que la clase 4 (la clase 4 contiene 52 muestras y la clase 6 contiene 20 muestras). Sin embargo, cuando entrenamos con funciones de coste aumenta la confusión entre las clases 6 y 4.

En la Tabla 3.13 se muestra el desempeño global del clasificador medido a través de dos criterios de medida: PC global y *g-mean*. Esta última medida nos da una idea del comportamiento general del clasificador sobre todas las clases.

En términos de PC y *g-mean*, los modelos MLP, RBF y RBF+VF no presentaron

Tabla 3.11: Resultados obtenidos al clasificar la base de datos Ecoli6 con la red RBF.

	Clase	F1	Ratio	PC	% confusión (> 10 %)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	91.46	
Opción 0	cls 3	10.66	0.23	84.39	cls 5 (12.43)
	cls 4	10.96	0.16	87.18	
	cls 5	10.33	0.11	62.54	cls 3 (34.60)
	cls 6	9.93	0.06	85.56	cls 4 (10.00)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	87.29	
Opción 1	cls 3	10.66	0.23	76.56	cls 5 (18.28)
	cls 4	10.96	0.16	78.78	cls 6 (11.98)
	cls 5	10.33	0.11	74.05	cls 3 (22.45)
	cls 6	9.93	0.06	85.20	cls 4 (10.20)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	88.49	
Opción 2	cls 3	10.66	0.23	75.64	cls 5 (19.62)
	cls 4	10.96	0.16	77.76	cls 6 (12.83)
	cls 5	10.33	0.11	69.35	cls 3 (24.70)
	cls 6	9.93	0.06	84.90	cls 4 (10.42)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	87.91	
Opción 3	cls 3	10.66	0.23	82.68	cls 5 (14.34)
	cls 4	10.96	0.16	86.77	
	cls 5	10.33	0.11	64.58	cls 3 (35.42)
	cls 6	9.93	0.06	85.42	cls 4 (10.42)

mejoras o reducciones significativas al aplicarse las opciones 1, 2 y 3.

La Opción 3 muestra una tendencia más estable y los incrementos de confusión en relación a la Opción 0, son de menor consideración que los que se generaron con las opciones (Opción 1 y 2). Esto es debido a que el valor de ponderación de la Opción 3 es obtenido a partir del MSE por clase, que a su vez es disminuido al avanzar en el entrenamiento de la ANN.

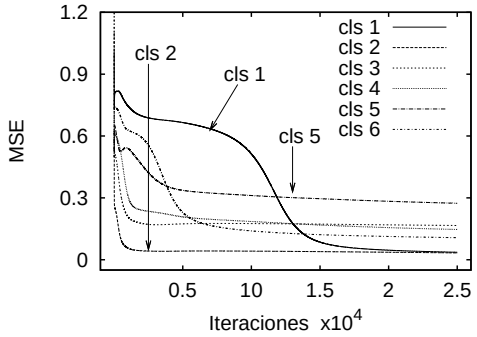
Así, en las primeras iteraciones se reduce el problema del desbalance de las clases significativamente, y posteriormente el entrenamiento de la ANN es afectado en menor medida por la función de coste (ver figuras donde se analizó el MSE por clase). Sin embargo, esta opción presenta menores ventajas en cuanto a PC y valores de *g-mean*.

Tabla 3.12: Resultados obtenidos al clasificar la base de datos Ecoli6 con la red RBF+VF.

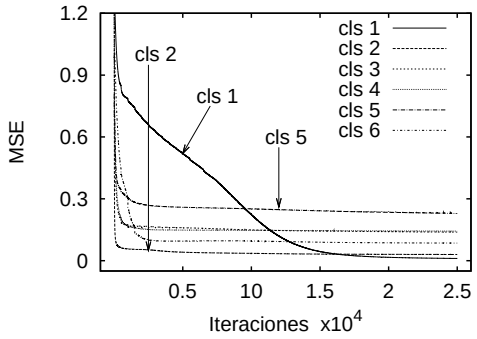
	Clase	F1	Ratio	PC	% confusión (> 10 %)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	92.13	
Opción 0	cls 3	10.66	0.23	81.35	cls 5 (12.84)
	cls 4	10.96	0.16	87.55	
	cls 5	10.33	0.11	61.01	cls 3 (35.12)
	cls 6	9.93	0.06	85.42	
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	88.73	
Opción 1	cls 3	10.66	0.23	76.92	cls 5 (18.44)
	cls 4	10.96	0.16	77.84	cls 6 (12.75)
	cls 5	10.33	0.11	72.89	cls 3 (20.99)
	cls 6	9.93	0.06	83.67	cls 4 (11.22)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	88.66	
Opción 2	cls 3	10.66	0.23	76.79	cls 5 (18.57)
	cls 4	10.96	0.16	78.43	cls 6 (12.16)
	cls 5	10.33	0.11	71.72	cls 3 (23.62)
	cls 6	9.93	0.06	83.67	cls 4 (11.22)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	87.79	
Opción 3	cls 3	10.66	0.23	82.07	cls 5 (14.62)
	cls 4	10.96	0.16	87.89	
	cls 5	10.33	0.11	64.13	cls 3 (34.65)
	cls 6	9.93	0.06	85.64	cls 4 (10.11)

Tabla 3.13: Desempeño global del clasificador: Base de datos Ecoli6. Los valores entre paréntesis hacen referencia a la desviación estándar.

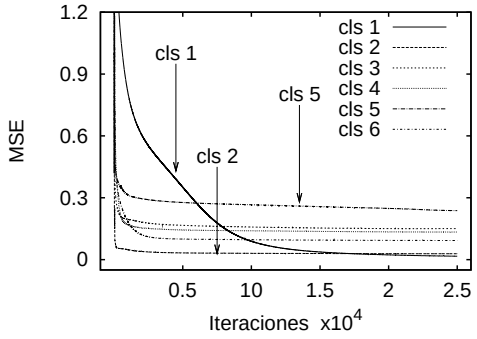
MLP	Opción 0	Opción 1	Opción 2	Opción 3
PC	86.56(2.78)	81.12(5.96)	82.38(6.25)	84.00(5.19)
g-mean	83.76(5.57)	80.25(5.69)	80.91(6.04)	83.02(5.67)
RBF	Opción 0	Opción 1	Opción 2	Opción 3
PC	85.87(3.36)	82.14(5.28)	81.77(5.43)	84.10(4.51)
g-mean	83.50(5.29)	82.29(5.09)	81.19(5.49)	83.01(5.49)
RBF+VF	Opción 0	Opción 1	Opción 2	Opción 3
PC	85.34(3.69)	82.48(5.53)	82.39(4.82)	84.03(4.79)
g-mean	82.59(5.62)	81.98(5.63)	81.77(5.01)	83.14(5.40)



(a) MLP



(b) RBF



(c) RBF+VF

Fig. 3.11: MSE por clase de la base de datos Ecoli6 con la Opción 0.

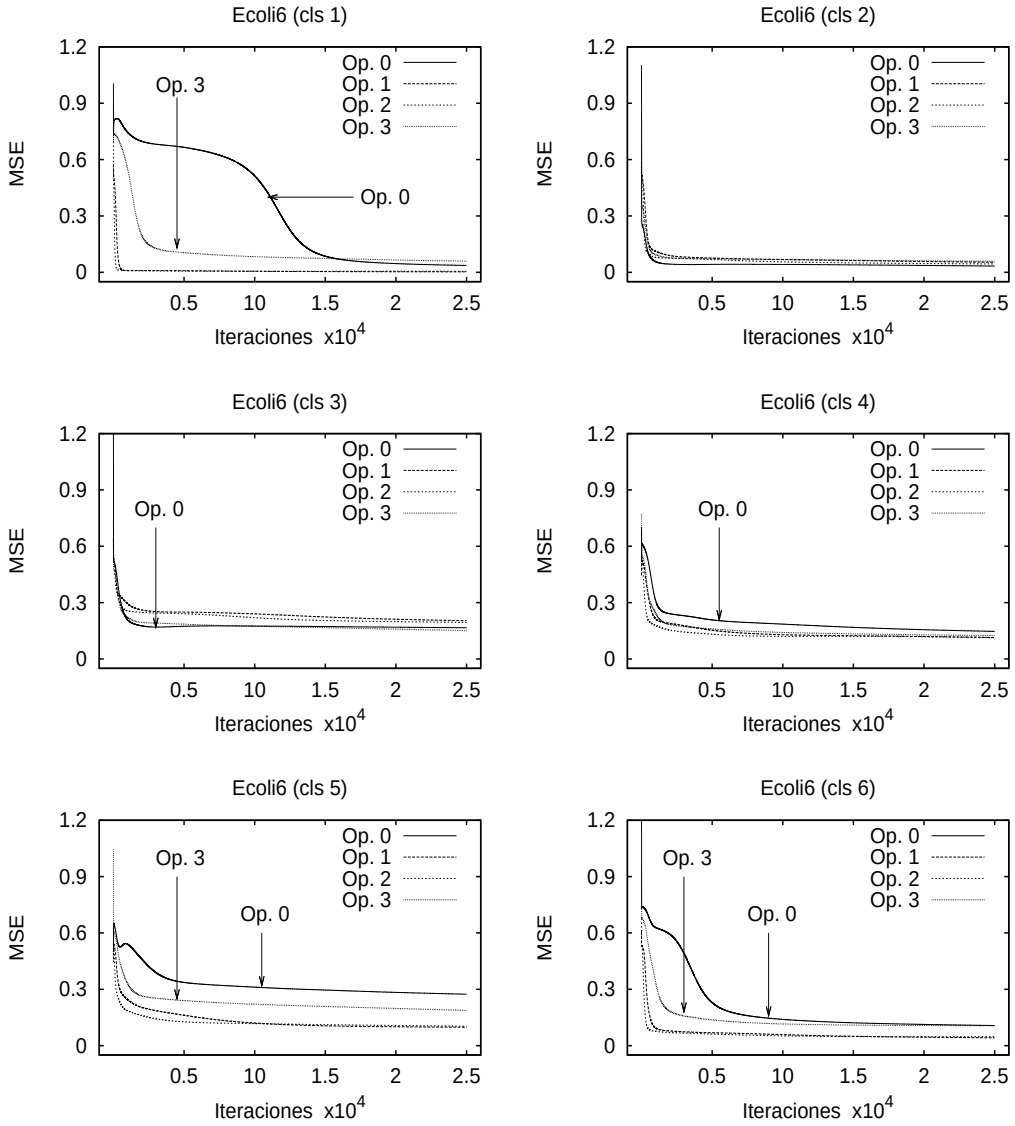


Fig. 3.12: MLP: MSE por clase de la base de datos Ecoli6.

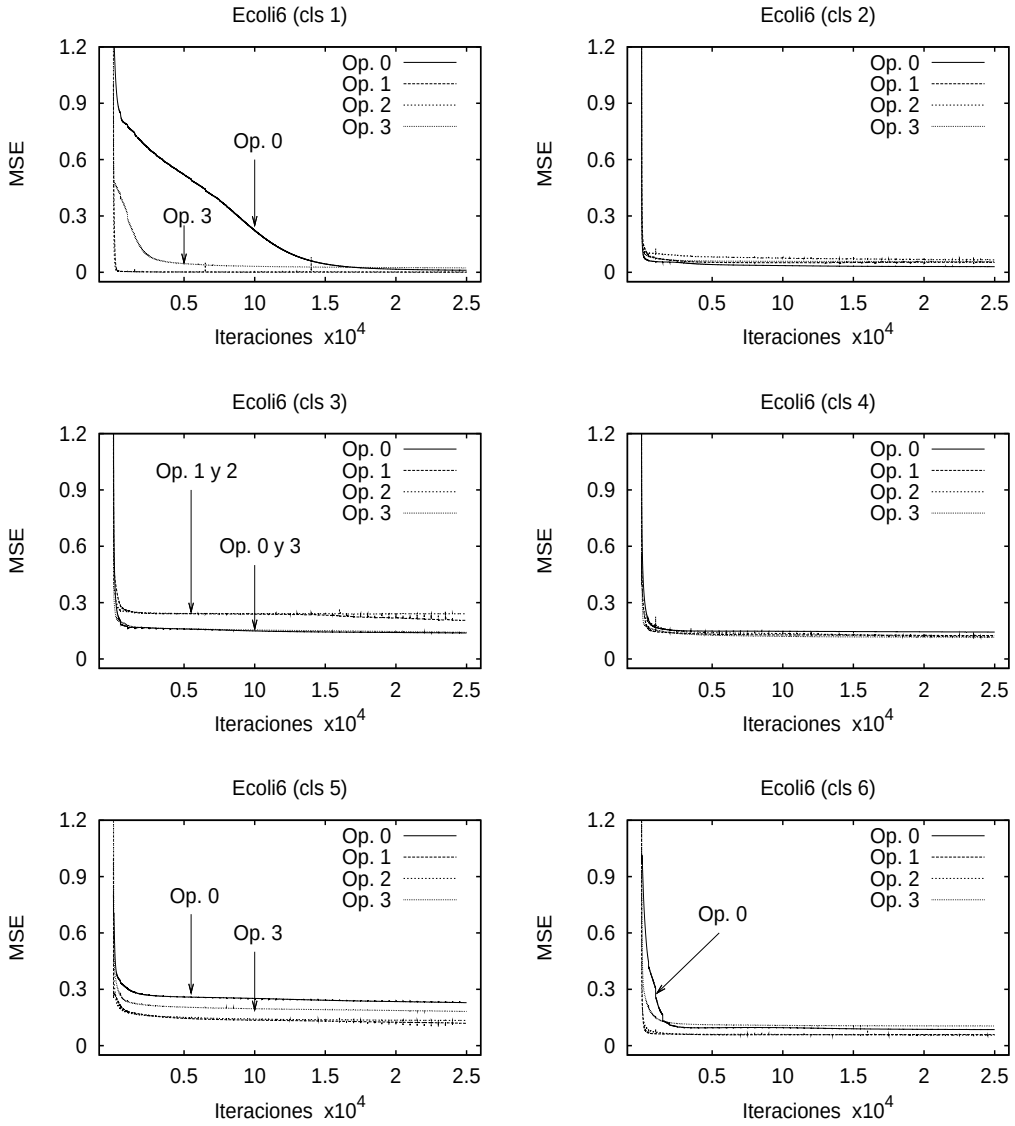


Fig. 3.13: Red RBF: MSE por clase de la base de datos Ecoli6.

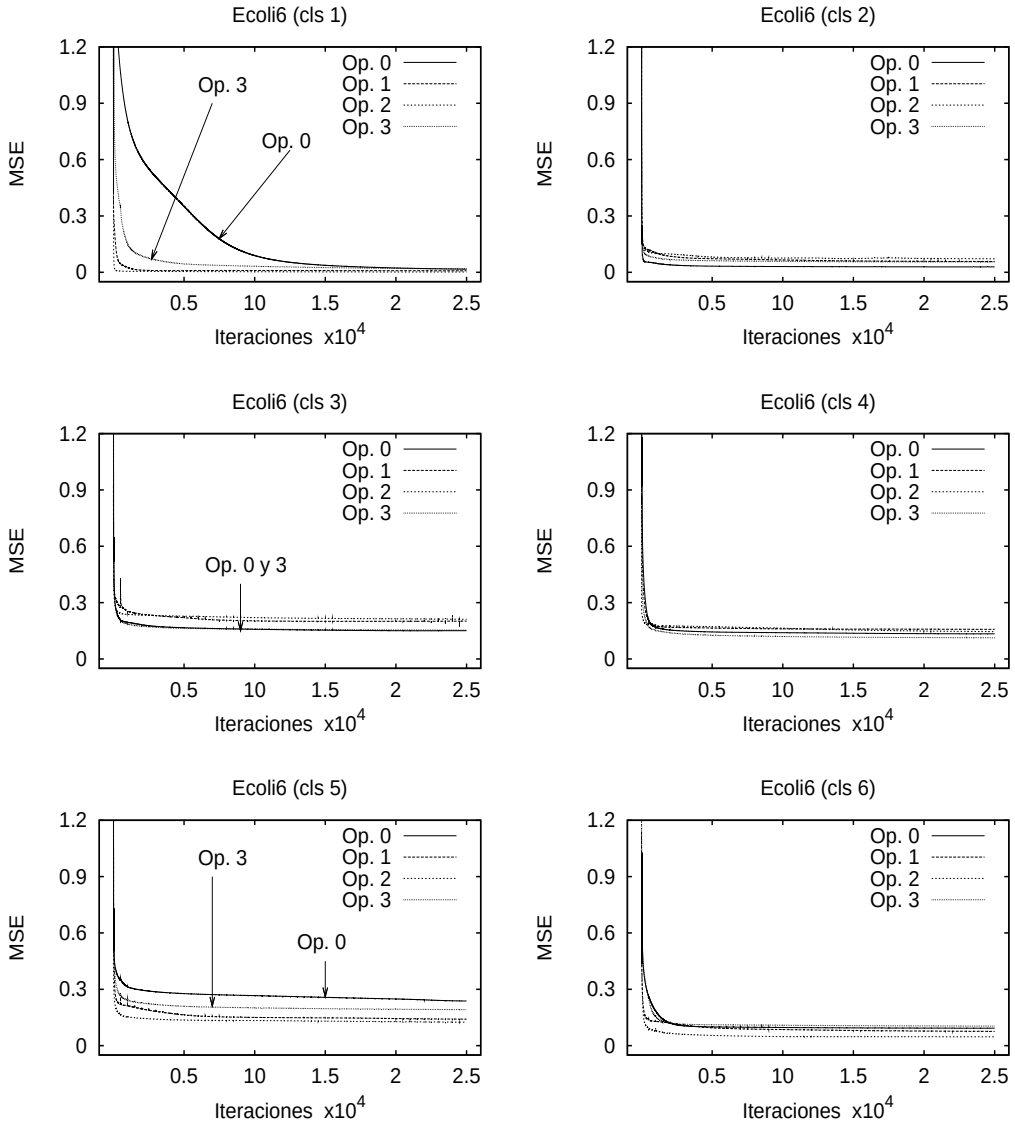


Fig. 3.14: Red RBF+VF: MSE por clase de la base de datos Ecoli6.

3.5.2 Base de datos Cayo

Para ampliar el estudio en esta sección se analiza el comportamiento[15] de la base de datos desbalanceada y de múltiples clases Cayo.

En la Tabla 3.14 se resumen algunas de las principales características del conjunto de datos Cayo. Obsérvese que la clase 1 es la única que presenta un valor de F1 que la hace parecer distinta a las demás.

Tabla 3.14: Características relevantes de la Base de datos Cayo

Clase	F1	Patrones	Ratio	Atributos
1	32.57	838	0.14	4
2	5.41	293	0.05	4
3	4.57	624	0.10	4
4	4.53	322	0.05	4
5	10.75	133	0.02	4
6	6.18	369	0.06	4
7	5.27	324	0.05	4
8	5.46	722	0.12	4
9	8.05	789	0.13	4
10	8.31	833	0.14	4
11	8.37	772	0.13	4

En la Tabla 3.15 se muestran los resultados obtenidos al experimentar con la base de datos Cayo y el MLP. Se observa que las clases que presentan un menor porcentaje de aciertos o precisión corresponden con las clases menos representadas en la ME (clases 2, 4, 5 y 6), excepto por la clase 7 que muestra una precisión superior al 90%.

En las Tablas 3.16 y 3.17 se muestran los valores separabilidad entre clases (F1) y de nivel de solapamiento (3-NN).

Obsérvese en la Tabla 3.16, que los valores en negrita corresponden a las clases que presentan mayor nivel de confusión cuando se utiliza el clasificador k-NN. Con ello se busca relacionar ambos criterios de medida (F1 y k-NN).

En la Tabla 3.17 se muestra el porcentaje de elementos identificados por el clasificador no paramétrico k nearest-neighbours (k-NN) para $k = 3$. Los resultados fuera de la diagonal se pueden interpretar como el grado de confusión entre clases. Los valores subrayados representan los niveles de confusión entre los pares de clases más significativos.

Nótese que la clase 7 es altamente separable o con un bajo nivel de confusión en términos de PC, y sin embargo la información aportada por las tablas anteriores no es suficiente para explicar este comportamiento. La hipótesis más probable es que se trata de una clase con un alto nivel de densidad en los datos o al menos, que el

[15] Tanto en la fase de entrenamiento como en la de clasificación.

clasificador la discrimina correctamente cuando se compara con la clase más cercana (clase 6).

Tabla 3.15: Resultados de la fase de clasificación del MLP con la base de datos Cayo.

	Clase	F1	Ratio	PC	% confusión (> 10 %)
Opción 0	cls 1	32.57	0.14	89.74	
	cls 2	5.41	0.05	51.20	cls 3 (48.63)
	cls 3	4.57	0.10	95.69	
	cls 4	4.53	0.05	70.99	cls 3 (12.61) cls 8 (11.43)
	cls 5	10.75	0.02	19.92	cls 1 (50.30) cls 3 (19.39)
	cls 6	6.18	0.06	56.44	cls 7 (31.90)
	cls 7	5.27	0.05	95.40	
	cls 8	5.46	0.12	98.55	
	cls 9	8.05	0.13	87.56	cls 10 (12.44)
	cls 10	8.31	0.14	77.03	cls 11 (21.80)
	cls 11	8.37	0.13	89.40	cls 10 (10.14)
Opción 1	cls 1	32.57	0.14	86.80	
	cls 2	5.41	0.05	56.92	cls 3 (43.08)
	cls 3	4.57	0.10	86.92	
	cls 4	4.53	0.05	93.73	
	cls 5	10.75	0.02	92.88	
	cls 6	6.18	0.06	60.84	cls 7 (31.71)
	cls 7	5.27	0.05	96.44	
	cls 8	5.46	0.12	95.98	
	cls 9	8.05	0.13	87.53	cls 10 (12.44)
	cls 10	8.31	0.14	75.29	cls 11 (22.49)
	cls 11	8.37	0.13	91.40	
Opción 2	cls 1	32.57	0.14	91.42	
	cls 2	5.41	0.05	65.48	cls 3 (34.52)
	cls 3	4.57	0.10	83.48	
	cls 4	4.53	0.05	96.77	
	cls 5	10.75	0.02	92.65	
	cls 6	6.18	0.06	60.54	cls 7 (30.57)
	cls 7	5.27	0.05	95.46	
	cls 8	5.46	0.12	96.90	
	cls 9	8.05	0.13	87.56	cls 10 (12.44)
	cls 10	8.31	0.14	73.25	cls 11 (25.73)
	cls 11	8.37	0.13	96.37	
Opción 3	cls 1	32.57	0.14	88.10	
	cls 2	5.41	0.05	51.37	cls 3 (48.63)
	cls 3	4.57	0.10	93.42	
	cls 4	4.53	0.05	93.54	
	cls 5	10.75	0.02	73.79	cls 1 (14.39) cls 3 (11.52)
	cls 6	6.18	0.06	60.43	cls 7 (30.82)
	cls 7	5.27	0.05	95.31	
	cls 8	5.46	0.12	94.86	
	cls 9	8.05	0.13	87.56	cls 10 (12.44)
	cls 10	8.31	0.14	76.36	cls 11 (22.99)
	cls 11	8.37	0.13	91.89	

Otro aspecto interesante que puede verse en la Tabla 3.15 es que al aplicarse

Tabla 3.16: Valores de F1 para la base de datos Cayo representados en forma de matriz de confusión.

Clase	1	2	3	4	5	6	7	8	9	10	11
1	—-	8.91	7.44	7.90	**2.03**	7.49	8.81	5.37	11.89	10.88	11.95
2	8.91	—-	**2.13**	1.90	9.07	3.55	3.69	14.25	14.37	13.01	25.10
3	7.44	**2.13**	—-	4.19	6.39	4.11	3.27	11.11	58.38	22.05	61.17
4	7.90	1.90	4.19	—-	9.36	2.77	1.80	6.90	59.66	15.30	61.17
5	**2.03**	9.07	6.39	9.36	—-	8.90	12.21	6.14	34.92	24.72	35.48
6	7.49	3.55	4.11	2.77	8.90	—-	**2.72**	2.48	1.53	0.68	1.37
7	8.81	3.69	3.27	1.80	12.21	**2.72**	—-	9.59	42.44	16.99	45.07
8	5.37	14.25	11.11	6.90	6.14	2.48	9.59	—-	14.63	7.77	14.88
9	11.89	14.37	58.38	59.66	34.92	1.53	42.44	14.63	—-	**2.11**	0.77
10	10.88	13.01	22.05	15.30	24.72	0.68	16.99	7.77	**2.11**	—-	**1.70**
11	11.95	25.10	61.17	61.17	35.48	1.37	45.07	14.88	0.77	**1.70**	—-

Tabla 3.17: Porcentaje de elementos identificados por el clasificador no paramétrico k nearest-neighbours (k-NN) para $k = 3$.

Clase	1	2	3	4	5	6	7	8	9	10	11
1	**98.81**	0.00	0.00	0.24	0.60	0.00	0.00	0.36	0.00	0.00	0.00
2	0.68	**82.94**	16.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	6.89	**91.35**	1.44	0.32	0.00	0.00	0.00	0.00	0.00	0.00
4	0.93	0.00	4.04	**92.86**	0.00	0.31	0.31	1.55	0.00	0.00	0.00
5	9.77	0.00	5.26	0.00	**81.95**	2.26	0.00	0.75	0.00	0.00	0.00
6	0.54	0.00	0.00	0.27	2.44	**84.28**	10.84	0.00	0.00	1.63	0.00
7	0.00	0.00	0.00	0.00	0.00	13.27	**86.42**	0.00	0.31	0.00	0.00
8	0.00	0.00	0.00	0.55	0.00	0.00	0.00	**99.31**	0.14	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**97.97**	2.03	0.00
10	0.00	0.00	0.00	0.00	0.12	0.24	0.00	0.24	4.80	**85.59**	9.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.77	**92.23**

las funciones de coste la confusión entre clases se reduce. Así, pueden observarse incrementos en la PC de las clases minoritarias 2, 4, 5 y 6.

En el caso de la clase 5 con Opción 0 se presenta una PC = 19.92%, mientras que al aplicarse las opciones 1 y 2 este valor aumenta en más de 70%.

Además, algunas de las clases mayoritarias también se ven beneficiadas por la aplicación de las opciones 1, 2 y 3. Tal es el caso de la clase 11 que incrementa su precisión con las tres opciones, aunque sólo la Opción 2 muestra un incremento significativo.

Por otro lado un efecto negativo en la aplicación de estas opciones es la reducción del porcentaje de aciertos de algunas clases consideradas mayoritarias (véase las clases 3, 8 y 10).

Sin embargo, las mejoras obtenidas en las clases minoritarias evitan que estas reducciones ocasionen mermas significativas en el desempeño global del clasificador, tal y como puede verse en la Tabla 3.18. Los tres modelos de ANN tienen incrementos en la PC global del clasificador. Mas aún, el desempeño de las clases medido a través de la *g-mean* es incrementado de forma importante.

Tabla 3.18: Desempeño global del clasificador: Base de datos Cayo. Los valores entre paréntesis hacen referencia a la desviación estándar.

MLP	Opción 0	Opción 1	Opción 2	Opción 3
PC	83.58(0.77)	85.00(0.94)	86.25(0.63)	85.15(0.27)
g-mean	70.17(6.28)	82.88(0.64)	84.41(0.59)	80.96(0.44)
RBF	Opción 0	Opción 1	Opción 2	Opción 3
PC	79.9(1.63)	81.51(0.97)	82.03(0.79)	83.33(0.47)
g-mean	58.97(3.94)	78.98(1.4)	78.01(4.02)	78.64(0.70)
RBF+VF	Opción 0	Opción 1	Opción 2	Opción 3
PC	76.92(2.92)	81.78(1.17)	81.23(2.05)	83.08(0.81)
g-mean	66.99(7.58)	78.3(3.94)	78.15(3.09)	77.68(2.39)

Al aplicarse las opciones 1 y 2, la confusión generada por el desbalance entre clases es reducida quedando a descubierto las clases que se confunden por la naturaleza de sus datos y no por el desbalance de las clases. Observando la Tabla 3.15, las clases que presentan mayor confusión son las siguientes: la clase 2 con la clase 3, la clase 6 con la clase 7, la clase 9 con la clase 10 y la clase 10 con la clase 11. Estos resultados corresponden a las clases que en la Tabla 3.17 presentan mayor nivel de solapamiento entre ellas[16].

La Tabla 3.19 corresponde a los resultados obtenidos con la red RBF. En esta red se puede observar que al igual que con el MLP, las clases 2, 4, 5 y 6 (consideradas minoritarias) presentan bajos porcentajes de aciertos en el caso de la Opción 0. Al aplicarse las opciones 1, 2 y 3, la precisión por clase se incrementa considerablemente y la confusión entre clases se ve disminuida, especialmente con las opciones 1 y 2.

En el caso de la clase 7 (clase considerada minoritaria) el modelo de red RBF presenta un menor porcentaje de aciertos para esta clase que cuando se aplicó el modelo MLP con la Opción 0. Después de aplicarse las opciones 1, 2 y 3 se obtienen valores de PC muy similares en ambas redes.

Los resultados obtenidos con el modelo de red RBF+VF (Tabla 3.20) siguieron el mismo patrón de comportamiento que la red RBF. Este modelo al igual que la red RBF es afectado de manera muy significativa por el desbalance de la ME. También

[16]Recuérdese que la estrategia no paramétrica utilizada para visualizar el solapamiento entre clases es mucho menos sensible al desbalance global de las clases que los modelos neuronales que se están estudiando.

Tabla 3.19: Resultados de la fase de clasificación de la red RBF con la base de datos Cayo.

	Clase	F1	Ratio	PC	% confusión (> 10 %)
	cls 1	32.57	0.14	89.67	
	cls 2	5.41	0.05	31.51	cls 3 (48.68) cls 9 (11.05)
	cls 3	4.57	0.10	96.17	
	cls 4	4.53	0.05	25.80	cls 3 (40.66) cls 7 (14.45) cls 8 (17.93)
	cls 5	10.75	0.02	12.02	cls 1 (47.47) cls 3 (16.06) cls 8 (23.94)
Opción 0	cls 6	6.18	0.06	45.83	cls 7 (28.55) cls 10 (14.71)
	cls 7	5.27	0.05	85.97	cls 3 (10.96)
	cls 8	5.46	0.12	98.93	
	cls 9	8.05	0.13	87.56	cls 10 (12.44)
	cls 10	8.31	0.14	75.02	cls 11 (18.04)
	cls 11	8.37	0.13	74.16	cls 10 (16.46)
	cls 1	32.57	0.14	82.26	
	cls 2	5.41	0.05	52.44	cls 3 (47.56)
	cls 3	4.57	0.10	84.95	
	cls 4	4.53	0.05	83.37	
	cls 5	10.75	0.02	86.36	
Opción 1	cls 6	6.18	0.06	60.45	cls 7 (31.46)
	cls 7	5.27	0.05	95.16	
	cls 8	5.46	0.12	94.83	
	cls 9	8.05	0.13	87.54	cls 10 (12.44)
	cls 10	8.31	0.14	66.05	cls 11 (22.68)
	cls 11	8.37	0.13	89.64	
	cls 1	32.57	0.14	85.50	
	cls 2	5.41	0.05	50.00	cls 3 (48.63)
	cls 3	4.57	0.10	89.50	
	cls 4	4.53	0.05	78.11	
	cls 5	10.75	0.02	90.53	
Opción 2	cls 6	6.18	0.06	61.21	cls 7 (29.76)
	cls 7	5.27	0.05	91.03	
	cls 8	5.46	0.12	94.18	
	cls 9	8.05	0.13	87.56	cls 10 (12.44)
	cls 10	8.31	0.14	65.72	cls 11 (24.28)
	cls 11	8.37	0.13	91.19	
	cls 1	32.57	0.14	86.70	
	cls 2	5.41	0.05	51.37	cls 3 (48.63)
	cls 3	4.57	0.10	92.15	
	cls 4	4.53	0.05	91.39	
	cls 5	10.75	0.02	66.67	cls 1 (16.45) cls 3 (11.26)
Opción 3	cls 6	6.18	0.06	56.13	cls 7 (31.91)
	cls 7	5.27	0.05	93.08	
	cls 8	5.46	0.12	93.23	
	cls 9	8.05	0.13	87.56	cls 10 (12.44)
	cls 10	8.31	0.14	74.28	cls 11 (21.91)
	cls 11	8.37	0.13	89.08	

puede verse que la aplicación de las opciones 1, 2 y 3 ayuda a reducir los efectos del desbalance en la fase de entrenamiento de la red, acelerando la convergencia de las

clases menos representadas en la ME y originando que se tengan mejores resultados con un menor o igual número de iteraciones.

Analizando los resultados en PC de las diferentes clases en las Tablas 3.19, 3.20 podemos afirmar que los modelos de red RBF y RBF+VF son más sensibles y obtienen menos mejoras al aplicar las funciones de coste para resolver el desbalance de las clases.

A partir de los resultados presentados en las Tablas 3.15, 3.19 y 3.20 se puede resumir lo siguiente:

1. Las clases menos representadas en la ME (ratio < 0.1) muestran bajos porcentajes de PC (excepto la clase 7).

2. Las clases minoritarias 4 y 5 presentan confusión a consecuencia de su bajo nivel de representatividad en la ME (desbalance de las clases).

3. Al aplicar las opciones 1, 2 y 3 se incrementan los valores de PC para las clases minoritarias 2, 4, 5, 6 y 7 (en esta última sólo en las redes RBF y RBF+VF).

4. El efecto de las opciones sobre las clase mayoritarias 1, 3, 8 y 10 reduce su PC debido a la disminución de su influencia en el aprendizaje de la ANN.

5. La PC de la clase mayoritaria 11 es incrementado y la de la clase 9 no se ve afectada.

6. Visto de conjunto, la confusión entre clases disminuye al aplicar las opciones 1, 2 y 3, y la confusión existente en los resultados después de aplicar las opciones 1, 2 y 3 está generada más por la naturaleza de los datos que por el desbalance de las clases.

7. La Opción 3 no alcanza a reducir al máximo el efecto del desbalance de la ME.

Para contrastar los resultados anteriores se analizó el comportamiento del MSE en la Fig. 3.15 para las clases 1, 5 y 11. Se eligieron estas tres clases porque en los tres modelos de ANNs presentan conductas similares. El MSE de la red RBF+VF no es mostrado porque presenta una conducta muy semejante al de la red RBF.

El objetivo que se persigue es el de mostrar los tres posibles escenarios de interés que ocurren cuando se pondera el MSE, para tratar el problema del desbalance de la ME en problemas de múltiples clases.

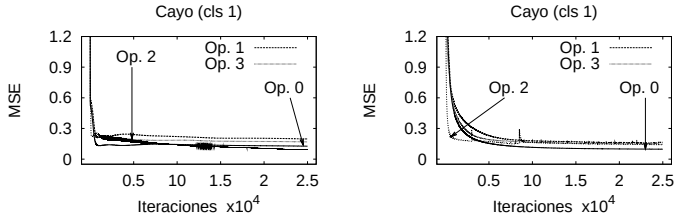
Al aplicarse las opciones 1, 2 y 3 se tiene el siguiente comportamiento sobre las las clases 1, 5 y 11:

- La clase 1 es mayoritaria y su PC es reducido.

Tabla 3.20: Resultados de la fase de clasificación de la red RBF+VF con la base de datos Cayo.

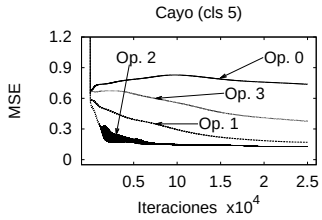
	Clase	F1	Ratio	PC	% confusión (> 10 %)
	cls 1	32.57	0.14	88.38	
	cls 2	5.41	0.05	24.88	cls 3 (48.66) cls 9 (11.89)
	cls 3	4.57	0.10	93.71	
	cls 4	4.53	0.05	43.68	cls 3 (31.54) cls 8 (15.44)
	cls 5	10.75	0.02	25.28	cls 1 (32.58) cls 3 (11.23) cls 8 (28.93)
Opción 0	cls 6	6.18	0.06	53.56	cls 7 (26.04)
	cls 7	5.27	0.05	79.20	cls 3 (16.15)
	cls 8	5.46	0.12	98.64	
	cls 9	8.05	0.13	87.56	cls 10 (12.44)
	cls 10	8.31	0.14	77.40	cls 11 (16.64)
	cls 11	8.37	0.13	71.74	cls 10 (22.13)
	cls 1	32.57	0.14	82.17	
	cls 2	5.41	0.05	48.58	cls 3 (47.47)
	cls 3	4.57	0.10	85.35	
	cls 4	4.53	0.05	87.48	
	cls 5	10.75	0.02	89.86	
Opción 1	cls 6	6.18	0.06	60.33	cls 7 (30.43)
	cls 7	5.27	0.05	91.65	
	cls 8	5.46	0.12	94.93	
	cls 9	8.05	0.13	87.27	cls 10 (12.44)
	cls 10	8.31	0.14	68.71	cls 11 (22.98)
	cls 11	8.37	0.13	89.46	
	cls 1	32.57	0.14	85.94	
	cls 2	5.41	0.05	49.09	cls 3 (47.43)
	cls 3	4.57	0.10	86.47	cls 4 (10.89)
	cls 4	4.53	0.05	87.11	
	cls 5	10.75	0.02	90.66	
Opción 2	cls 6	6.18	0.06	60.73	cls 7 (31.07)
	cls 7	5.27	0.05	94.33	
	cls 8	5.46	0.12	92.38	
	cls 9	8.05	0.13	86.93	cls 10 (11.70)
	cls 10	8.31	0.14	63.28	cls 11 (27.68)
	cls 11	8.37	0.13	87.31	
	cls 1	32.57	0.14	86.30	
	cls 2	5.41	0.05	47.31	cls 3 (48.63)
	cls 3	4.57	0.10	91.57	
	cls 4	4.53	0.05	84.52	
	cls 5	10.75	0.02	67.72	cls 1 (13.40) cls 3 (11.19)
Opción 3	cls 6	6.18	0.06	57.94	cls 7 (30.94)
	cls 7	5.27	0.05	93.77	
	cls 8	5.46	0.12	95.50	
	cls 9	8.05	0.13	87.56	cls 10 (12.44)
	cls 10	8.31	0.14	73.35	cls 11 (22.67)
	cls 11	8.37	0.13	90.02	

- La clase 5 es minoritaria y su PC es incrementado.

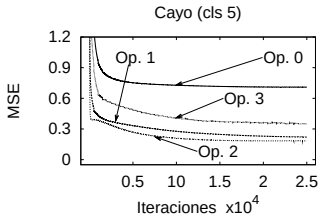


(a) MLP: Clase 1

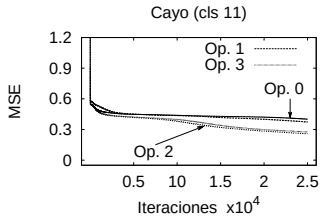
(b) RBF: Clase 1



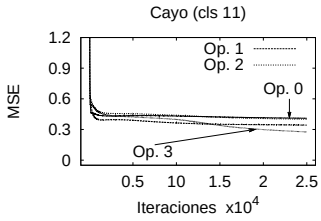
(c) MLP: Clase 5



(d) RBF: Clase 5



(e) MLP: Clase 11



(f) RBF: Clase 11

Fig. 3.15: MSE de las clases 1, 5 y 11, de la base de datos Cayo. El MSE de la red RBF+VF, presenta una conducta muy semejante a la de la red RBF por lo que no es presentado para evitar redundancia en los resultados.

- La clase 11 es mayoritaria y su PC es incrementado.

En la Fig. 3.15a y 3.15b, el incremento del MSE de la clase 1 es prácticamente

insignificante. Es difícil relacionar este incremento con la reducción de su PC. En la clase 11 no se observa una disminución significativa en el MSE como para justificar el incremento de la PC (Fig. 3.15e y 3.15f). Sin embargo, en la Fig. 3.15c y 3.15d están claros los efectos de las opciones 1, 2 y 3 sobre el MSE de la clase 5. Se tienen importantes reducciones en el MSE de esta clase coincidiendo con notables incrementos de PC que se observaron previamente en las Tablas 3.15, 3.19 y 3.20. Este comportamiento ya sido observado con anterioridad en las bases de datos de dos clases y en Ecoli6.

Al evaluar el desempeño global del clasificador (Tabla 3.18) se observan notables mejoras tanto en PC global como en valores de *g-mean* en los tres modelos de ANNs y las opciones (1, 2 y 3). Obsérvese que en los tres modelos de ANN se tienen mejoras significativas en resultados de *g-mean*, lo que se corresponde con incrementos en los valores de PC de las clases minoritarias.

3.6 Conclusión

En este capítulo se ha estudiado empíricamente el efecto del desbalance de las clases sobre el algoritmo back-propagation con procesamiento por grupos. Se utilizaron tres arquitecturas distintas de ANN y se realizaron experimentos con conjuntos de datos reales y artificiales. Así mismo, se analizó este problema en el contexto de dos y múltiples clases.

El estudio fue dirigido desde dos perspectivas: a) análisis del MSE, y b) precisión en la clasificación. Se propusieron tres funciones de coste dirigidas a reducir el efecto del desbalance de las clases.

A partir de la resultados presentados en este capítulo se puede concluir lo siguiente:

- El desbalance de las clases sobre el algoritmo back-propagation (con procesamiento por grupos) retarda la convergencia de las clases minoritarias.
- Cuando las clases minoritarias presentan bajos niveles de solapamiento y alta separabilidad entre las clases, convergen en un menor número de iteraciones.
- El problema de no alcanzar la convergencia está más relacionado con cuestiones de solapamiento y separabilidad entre clases que con el desbalance.
- Incluir funciones de coste para compensar el desbalance de la ME tiene básicamente dos consecuencias: a) acelerar la convergencia de las clases menos representadas y b) reducir la influencia de las clases mayoritarias en el proceso de entrenamiento.

- Si las clases mayoritarias y minoritarias presentan bajo nivel de separabilidad entre clases, el efecto de las opciones 1 y 2 puede acentuar este solapamiento reduciendolo en las clases minoritarias e incrementandolo en la clases mayoritarias.

- Si los datos se encuentran poco representados en el espacio, el efecto de aplicar funciones de coste con las opciones 1, 2 y 3 puede modificar significativamente la interpretación de la red acerca de la distribución espacial de las clases, perjudicando significativamente las clases mayoritarias. Este efecto disminuye a medida que aumenta la densidad de los datos en estas clases.

- La Opción 3 genera menos efectos negativos en el proceso de aprendizaje de la red. No obstante, es la que obtiene menores beneficios sobre las clases minoritarias.

- Los MLP son más efectivos que las redes RBF y RBF+VF cuando se trabaja con bases de datos desbalanceadas y con alto nivel de solapamiento o baja separabilidad. En otras palabras, el MLP es menos sensible al desbalance de la ME y al solapamiento entre clases.

  Esto puede deberse a la forma en que realiza la transformación del espacio de entrada en relación a la forma en como es realizado por las redes basadas en funciones de base radial.

  Por otro lado, las redes RBF y RBF+VF asumen una distribución normal. Otra evidencia que reafirma lo anterior, es que el MLP requiere de menos neuronas ocultas que las redes basadas en funciones de base radial para resolver el mismo problema.

En el siguiente capítulo se continuará con el estudio del desbalance del las clases pero desde un enfoque modular. Se aplicará el criterio de divide y vencerás para reducir la influencia de unas clases sobre otras (nivel de confusión), y de esta forma intentar simplificar el problema del desbalance de la ME.

Capítulo 4

Tratamiento del desbalance de las clases con ANN-M

Contenido

4.1 Redes Neuronales Modulares

Las ANN Modulares (ANN-M) son una tendencia en el diseño de arquitecturas de ANN [Auda 1998]. Estos modelos han sido inspirados por la alta modularidad de las redes neuronales biológicas y están basados en el principio de básico de la ingeniería: "divide y vencerás" [Ronco 1995].

En ocasiones, el uso de ANN-M puede implicar una mejora significativa en comparación a la aplicación de una ANN global (ANN-G)[1] [Ronco 1995]. Estas últimas presentan una alta tendencia a introducir interferencia interna a causa del fuerte acoplamiento entre los pesos de la capa oculta [Jacobs 1991].

En términos de eficiencia, las ANN-M presentan importantes ventajas computacionales frente a las ANN-G.

[1] Únicamente en este capítulo se hará referencia como ANN-G a las ANNs no modulares. En el resto de la tesis no se hará referencia a las ANNs monolíticas como ANN-G.

- El número de iteraciones necesarias para entrenar los módulos es menor que para entrenar una ANN-G que resuelva el mismo problema.
- Los módulos en la ANN-M son más pequeños en comparación con las ANN-G.
- Los módulos pueden ser entrenados de forma paralela e independiente.

En [Anand 1995] se estudia la forma de acelerar la convergencia del MLP cuando se trabaja con bases de datos de múltiples clases. La idea es descomponer el problema en subproblemas de dos clases y de esta forma construir una arquitectura específica de ANN-M.

El desbalance entre clases generado al transformar el problema de múltiples clases en otros de dos clases, es tratado por medio de un método efectivo que equilibra las proporciones de error de las clases minoritaria y mayoritaria. Este proceso acelera la convergencia de la ANN. En la Fig. 4.1 se muestra la estructura de ANN-M empleada en [Anand 1995].

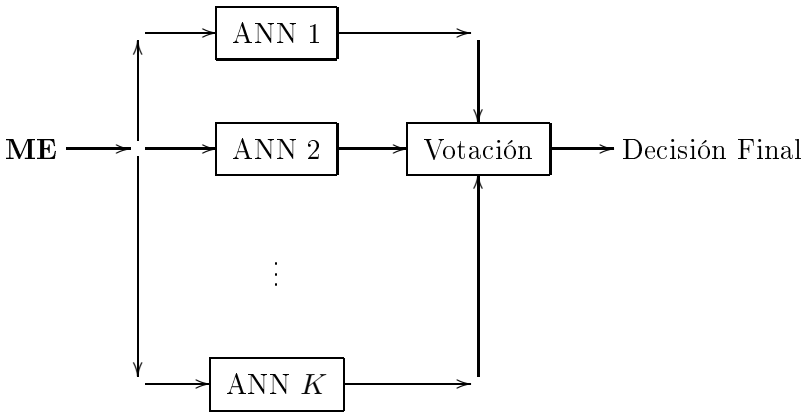


Fig. 4.1: Esquema simplificado de una red modular. Cada experto corresponde a una ANN del mismo tipo (MLP, RBF o RBF+VF), no existen combinaciones de ANN de diferentes tipos. Los módulos se especializan en aprender cada una de las K clases.

En este capítulo se extiende la idea de [Anand 1995] a problemas desbalanceados de múltiples clases (PDMC) y distintas arquitecturas de ANN. Así mismo, se utilizan diferentes mecanismos para la compensación del desbalance.

La estructura de este capítulo es la siguiente. La sección 4.2 esta dedicada a explicar los objetivos perseguidos al descomponer un PDMC en subproblemas más simples, así como las principales desventajas de este proceso. En la sección 4.3 se analiza la reducción de la interferencia entre clases y se trata el desbalance de la ME a

partir de la inclusión de alguna función de coste (opciones 1, 2 o 3, ver sección 3.3.1) al proceso de entrenamiento de la ANN. En la sección 4.4 se investiga el problema de la comunicación entre módulos, y finalmente en la sección 4.5 se presentan algunas de las principales conclusiones obtenidas.

El estudio se realizó sobre las redes MLP, RBF y RBF+VF entrenadas, con el algoritmo back-propagation con procesamiento por grupos. La experimentación fue realizada sobre cuatro bases de datos reales de múltiples clases.

4.2 Descomposición del problema

La descomposición del problema se realizó de la siguiente manera: Cada subconjunto ME_k es extraído de la ME. Posteriormente $K-1$ subcojuntos son unidos en uno solo para formar la clase mayoritaria y el subconjunto restante es utilizado como clase minoritaria. Finalmente ambas clases (mayoritaria y minoritaria) son unidas en una nueva ME de dos clases. Este procedimiento es realizado para $k = 1, 2, ...K$ donde K es el total de clases. Observe que se generarán K nuevas ME y cada una de ellas contendrá como clase minoritaria a una de las clases de la ME original.

Los objetivos que se persiguen al descomponer un PDMC en subproblemas de dos clases y tratarlos de forma independiente son los siguientes:

- Reducir la interferencia entre clases para así tratar exclusivamente con el desbalance de la ME.
- Disminuir el coste computacional asociado a la resolución de PDMC por medio de ANN.
- Acelerar la convergencia de la ANN.

Sin embargo, al realizar la transformación de un PDMC a varios de dos y tratarlos de forma independiente, se presentan dos problemas fundamentales:

1. El desbalance entre clases se incrementa considerablemente.
2. ¿Cómo conseguir una comunicación efectiva entre los módulos independientes?

Estas cuestiones se estudian por separado con la finalidad de simplificar su estudio.

4.3 Desbalance e interferencia de las clases

En esta sección se estudia el incremento del desbalance de las clases generado al transformar un PDMC a subproblemas de dos clases, y se evalúan las posibilidades de las opciones 1, 2 y 3 para tratar este problema. Así mismo, se estudia empíricamente el impacto de la interferencia entre clases cuando un PDMC es tratado de forma global y modular.

4.3.1 Aspectos metodológicos

Para realizar este estudio se siguió el siguiente procedimiento: El PDMC es descompuesto en subproblemas de dos clases y cada uno de ellos es resuelto por una ANN independiente. Posteriormente, las salidas de las diferentes ANNs (expertos) son evaluadas para producir una salida global de la red como se ilustra en la Fig. 4.1.

A cada módulo se le incluye alguna de las opciones 1-3 con la finalidad de contrarrestar el efecto del desbalance de las clases. La decisión final del clasificador es establecida bajo el esquema de votación simple (winners-take-all).

Nuevamente las bases de datos Ecoli6 y Cayo son utilizadas como conjuntos de datos prototipo. En el apartado 4.4 de este capítulo y en el apéndice C, se incluyen los conjuntos de datos (de múltiples clases) Feltwell y Satimage para profundizar en este estudio. Para mayor información sobre los aspectos experimentales consúltese la sección 2.10.

4.3.2 Base de datos Ecoli6: ANN-M

A continuación se analiza sobre la base de datos Ecoli6, el efecto de la interferencia de las clases, así como las ventajas de las opciones 1, 2 y 3 para tratar el desbalance de la ME en las fases de entrenamiento y clasificación de la ANN-M sobre las redes MLP, RBF y RBF+VF.

En la Tabla 4.1 se muestra la PC por clase obtenida al clasificar con las ANN-M sobre MLP, RBF y RBF+VF. Estos resultados corresponden a los generados de integrar los módulos independientes especializados en cada una de las clases.

Obsérvese que los valores de F1[2] corresponden a problemas de dos clases, i.e., a la clase en la que se especializa el módulo (clase minoritaria) y al resto de las clases unidas en una sola (clase mayoritaria). La razón es la proporción de elementos de la clase minoritaria en relación al total de muestras[3].

[2]En la sección 3.4 se definió a F1 como criterio de medida de la separabilidad entre clases. Una medida que da una idea acerca de que tan alejadas se encuentran las clases entre sí.

[3]Expresa las probabilidades a priori de cada una de las clases.

Los valores de "confusión" indican el porcentaje de elementos con respecto a las otras clases que son identificados con la clase en cuestión. Por ejemplo, en la Tabla 4.1 con la ANN-M sobre MLP en la clase 5, el valor de 31.20 de la clase 3 dice que el 31.20% de las muestras de la clase 5 son identificados como ejemplos de la clase 3. Nótese que la confusión se evidencia al integrar las salidas de cada uno de los módulos.

Al transformar el problema de múltiples clases en subproblemas de dos clases con la estrategia propuesta, el desbalance de las clases es incrementado. No obstante, este aumento no logra mermar la efectividad del clasificador sobre la base de datos Ecoli6. Se observa en la Tabla 4.1, que los resultados de PC y confusión son similares a los presentados por la ANN-G (ver Tabla 3.8).

No hay una evidencia clara (ver Tabla 4.1) que nos indique que es mejor en términos de efectividad (de PC y confusión) la descomposición del problema en subproblemas de dos clases para esta base de datos.

Tabla 4.1: Resultados obtenidos en la fase de clasificación por la ANN-M.

	Módulo de Clase	F1	Razón	PC	% confusión (> 10 %)
MLP	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	94.15	
	cls 3	2.74	0.23	83.82	cls 5 (10.88)
	cls 4	1.82	0.16	82.55	
	cls 5	1.60	0.11	62.10	cls 3 (31.20)
	cls 6	3.23	0.06	85.20	
RBF	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	92.34	
	cls 3	2.74	0.23	82.32	cls 5 (12.71)
	cls 4	1.82	0.16	86.07	
	cls 5	1.60	0.11	62.61	cls 3 (31.91)
	cls 6	3.23	0.06	86.17	cls 4 (10.11)
RBF+VF	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	92.18	
	cls 3	2.74	0.23	81.38	cls 5 (12.69)
	cls 4	1.82	0.16	87.68	
	cls 5	1.60	0.11	62.61	cls 3 (31.31)
	cls 6	3.23	0.06	87.23	

La cuestión es ¿cuál fue el beneficio de dividir un PDMC en subproblemas de dos clases?.

Si se analiza el MSE (Mean Square Error, error cuadrático medio) de la clase 1[4] en el modelo ANN-M sobre MLP, se observa que el aumento del desbalance de la clase 1 en su correspondiente módulo, genera un incremento en el MSE de esta clase

[4]Recuerde que se trata de la clase menor en la ME.

y en consecuencia su convergencia se hace más lenta (Fig 4.2). No obstante, al final del proceso la convergencia es alcanzada.

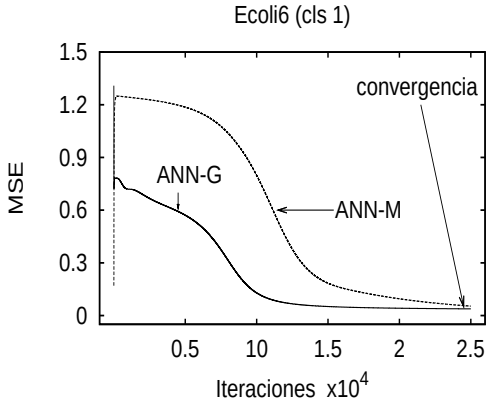
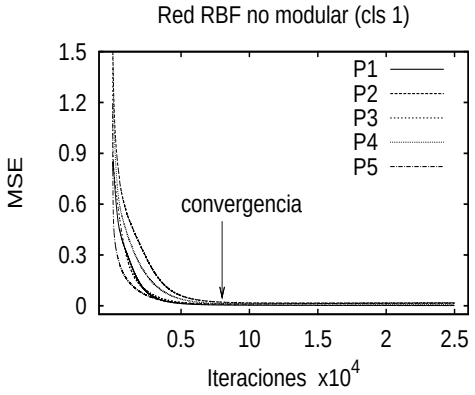


Fig. 4.2: Comparación del MSE de la clase 1 de la base de datos Ecoli6 obtenido con ANN-M y ANN-G. El ejemplo corresponde al modelo MLP y la Opción 0.

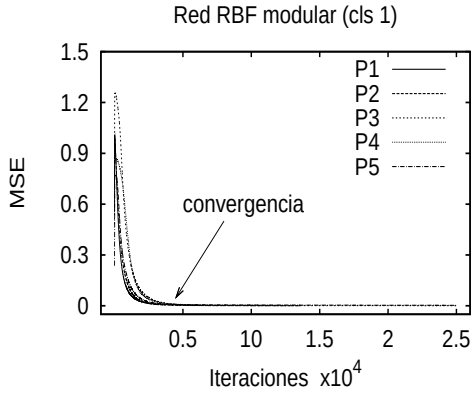
Por otro lado, en la Fig. 4.3 se observan los valores de MSE obtenidos por los modelos ANN-M sobre RBF y RBF+VF con la clase menos representada (clase 1). En esta clase el comportamiento de estas dos redes es muy diferente y se evidencian los beneficios de descomponer el problema de múltiples clases en subproblemas de dos clases, que a diferencia de la ANN-M sobre MLP, el MSE de esta clase presenta una mayor estabilidad en las primeras iteraciones del proceso de entrenamiento, lo que se traduce en una disminución más rápida del MSE.

Obsérvese en la Fig. 4.3 que el efecto de entrenar por separado módulos especializados por clase es el de acelerar la convergencia de la ANN en este tipo de redes. Estos resultados sugieren que los modelos ANN-M sobre RBF y RBF+VF son afectados en mayor medida por la interferencia de las clases en relación al modelo ANN-M sobre MLP, y por lo tanto, al simplificar el problema a dos clases se facilita su entrenamiento.

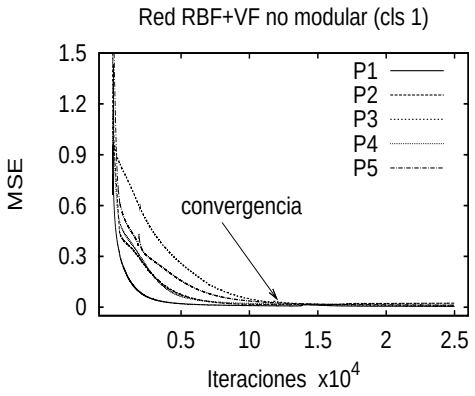
En la Fig. 4.4 se presenta el número promedio de iteraciones necesarias para alcanzar un valor medio mínimo de MSE. Se procedió de esta forma para visualizar la diferencia en número de iteraciones por clase entre la ANN-M (color oscuro) y la ANN-G (color claro). En el caso de la clase 1, los modelos neuronales RBF y RBF+VF requieren de menos iteraciones cuando el problema es descompuesto en subproblemas de dos clases.



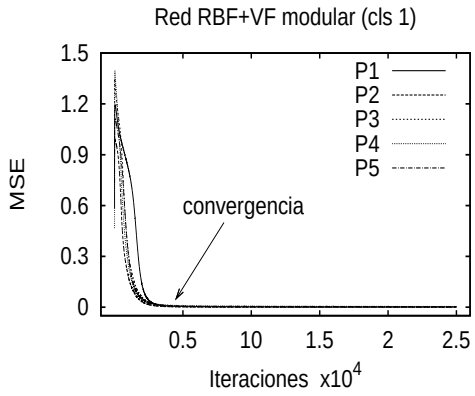
(a) No Modular



(b) Modular

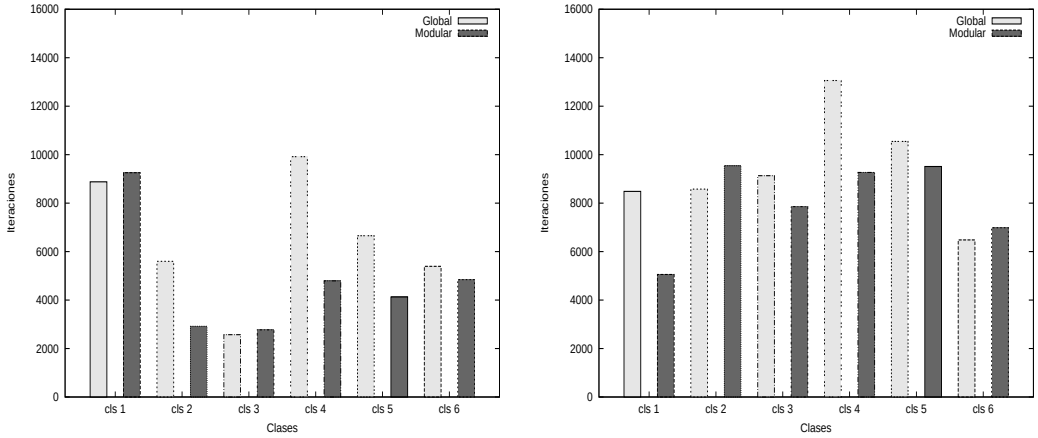


(c) No Modular



(d) Modular

Fig. 4.3: Comparación entre el MSE de la clase 1 de la base de datos Ecoli6 obtenido con ANN-M y ANN-G. El ejemplo corresponde a los modelos RBF y RBF+VF con la Opción 0. P1, P2, ... y P5 corresponden a las 5 particiones de la base de datos Ecoli6 al aplicar k-fold-cross-validation para $k = 5$.



(a) MLP

(b) RBF

(c) RBF+VF

Fig. 4.4: Comparación del número de iteraciones necesarias (por clase) para alcanzar un valor promedio mínimo de MSE en redes no modulares (color claro) y modulares (color oscuro) con la base de datos Ecoli6 y la Opción 0. El eje x indica cada una de las clases de la ME mientras que el eje y representa el número de iteraciones.

Por otra parte al incluir las opciones 1, 2 y 3 al proceso de entrenamiento, la convergencia de las clases menos representadas se acelera y de esta forma se logra reducir el problema originado por la transformación del PDMC a subproblemas desequilibrados de dos clases.

En las figuras 4.5, 4.6 y 4.7 se presenta el MSE por clase generado por los modelos de ANN-M sobre MLP, RBF y RBF+VF respectivamente. Las abreviaturas cls 1, cls 2, ...,cls 6, nos indica el MSE de cada una de las clases en su correspondiente módulo.

La finalidad de mostrar estos resultados no es la de analizar el MSE por clase, sino la de evidenciar los beneficios en cuestiones de convergencia que ofrecen las opciones 1, 2 y 3. Obsérvese en estas figuras la tendencia a reducir el MSE por clase de manera más homogénea cuando las opciones 1, 2 y 3 son incluidas en el proceso de entrenamiento.

En la Tabla 4.1 se muestra en la columna correspondiente a F1, el grado de separabilidad de cada módulo y la clase en cuestión, y el resto de las clases unidas en una sola clase. Así, se observa como esta medida presenta valores bajos para los módulos de las clases 2, 3, 4, 5 y 6, lo que plantea que existe un importante grado de solapamiento con las muestras de alguna de las clases incorporadas al resto de las clases que están unidas en una sola clase dentro de cada módulo.

En el capítulo anterior se vió la separabilidad entre pares de clases, existiendo un nivel bajo de separabilidad entre las clases 3 y 5 y en menor medida entre las clases 4 y 6 (ver Tabla 3.9). Además se observó como al clasificar entre pares de clases con el clasificador no paramétrico 3-NN se confirmaba este hecho.

En las Tablas 4.2, 4.3 y 4.4 se presentan algunos de los resultados obtenidos al clasificar la base de datos Ecoli6 con la ANN-M.

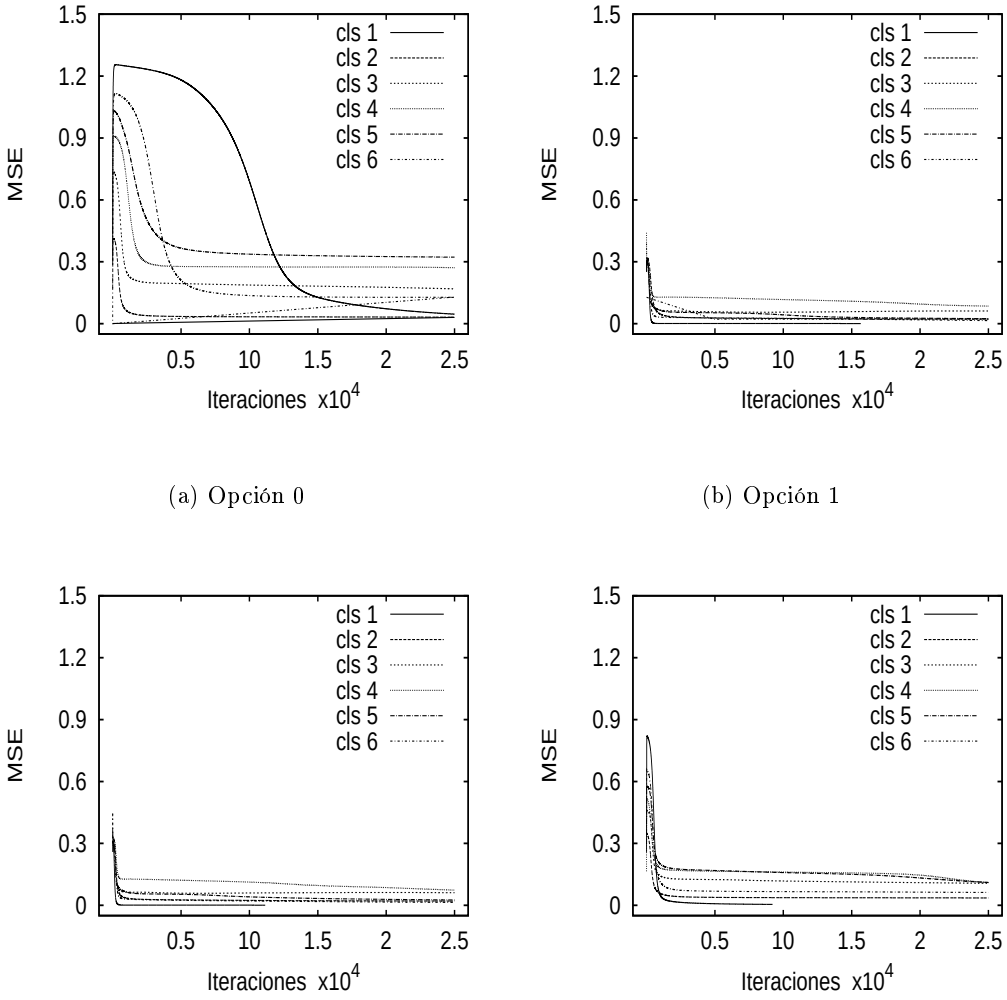
A nivel de efectividad (PC o confusión con otras clases) se observa una ligera mejora la PC de la ANN-M sobre las redes RBF y RBF+VF respecto a sus versiones no modulares[5]. Esta tendencia no se observo para el caso de la ANN-M sobre MLP[6].

Analizando con más detalle las Tablas 4.2, 4.3 y 4.4, se observa que la clase 5 presenta una baja PC en los tres modelos neuronales. Dicha clase presenta el valor más bajo de F1 en su módulo lo que supone un nivel de solapamiento importante con otras clases (clase 3), además de un desbalance importante dentro de su módulo.

Al evaluar por separado el módulo correspondiente a la clase 5, se obtuvieron con la Opción 0 valores de PC = 58.28 %. Después de aplicar las funciones de coste se incrementó la PC de la clase 5 en este módulo. Los resultados obtenidos para esta clase son los siguientes: Opción 1 (PC = 70.65%), Opción 2 (PC 68.81%) y Opción

[5]En las Tablas 4.3 y 4.4, los niveles de confusión son menores cuando se aplica el modelo ANN-M respecto a aplicar la ANN-G del capítulo 3 (Tablas 3.11 y 3.12).

[6]Compárese las Tablas 3.10 y 4.2.



(a) Opción 0

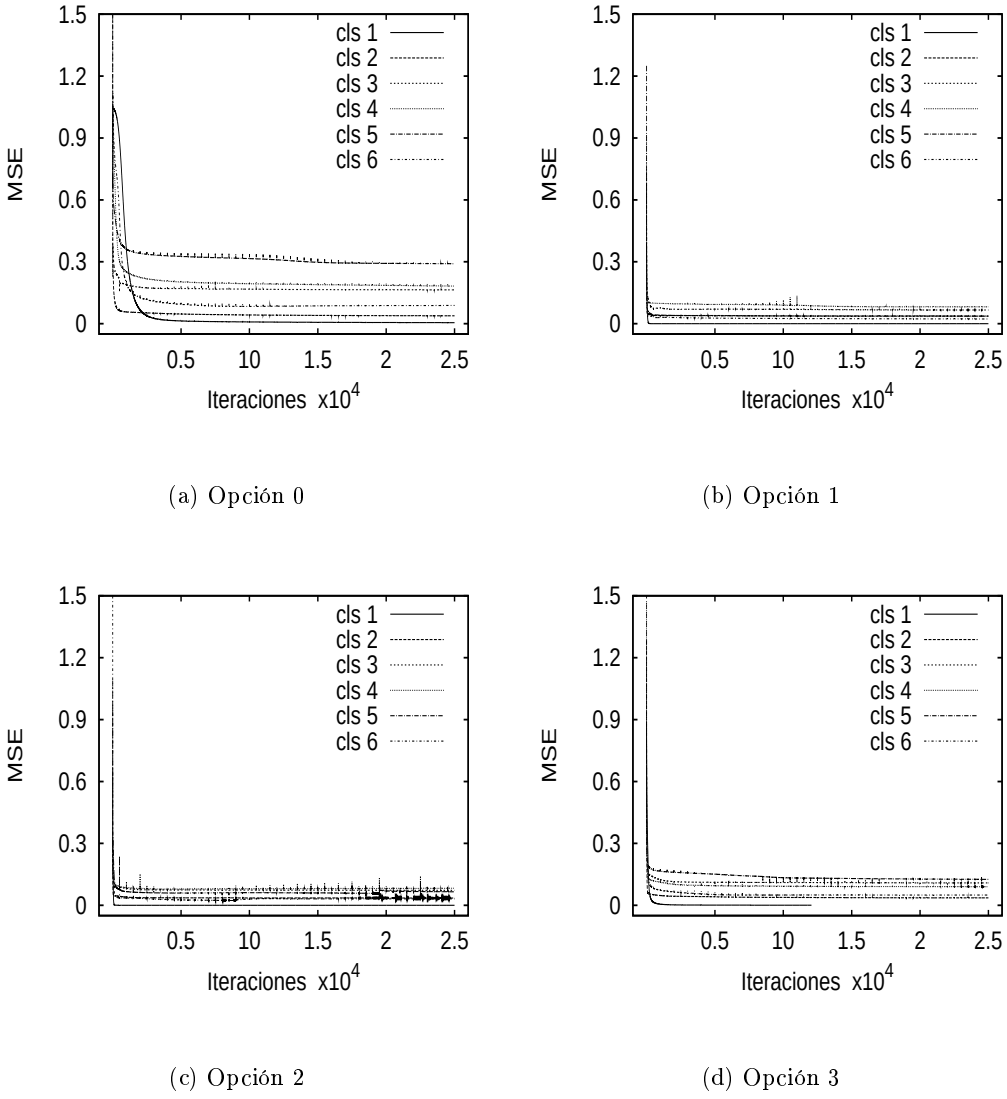
(b) Opción 1

(c) Opción 2

(d) Opción 3

Fig. 4.5: MSE de cada una de las clases de la base de datos Ecoli6 obtenido en el proceso de aprendizaje de la ANN-M sobre MLP.

3 (PC 72.5%). Se incrementó la PC individual para este módulo. Sin embargo, estas mejoras no se trasladaron en un incremento de PC para esta clase cuando



(a) Opción 0

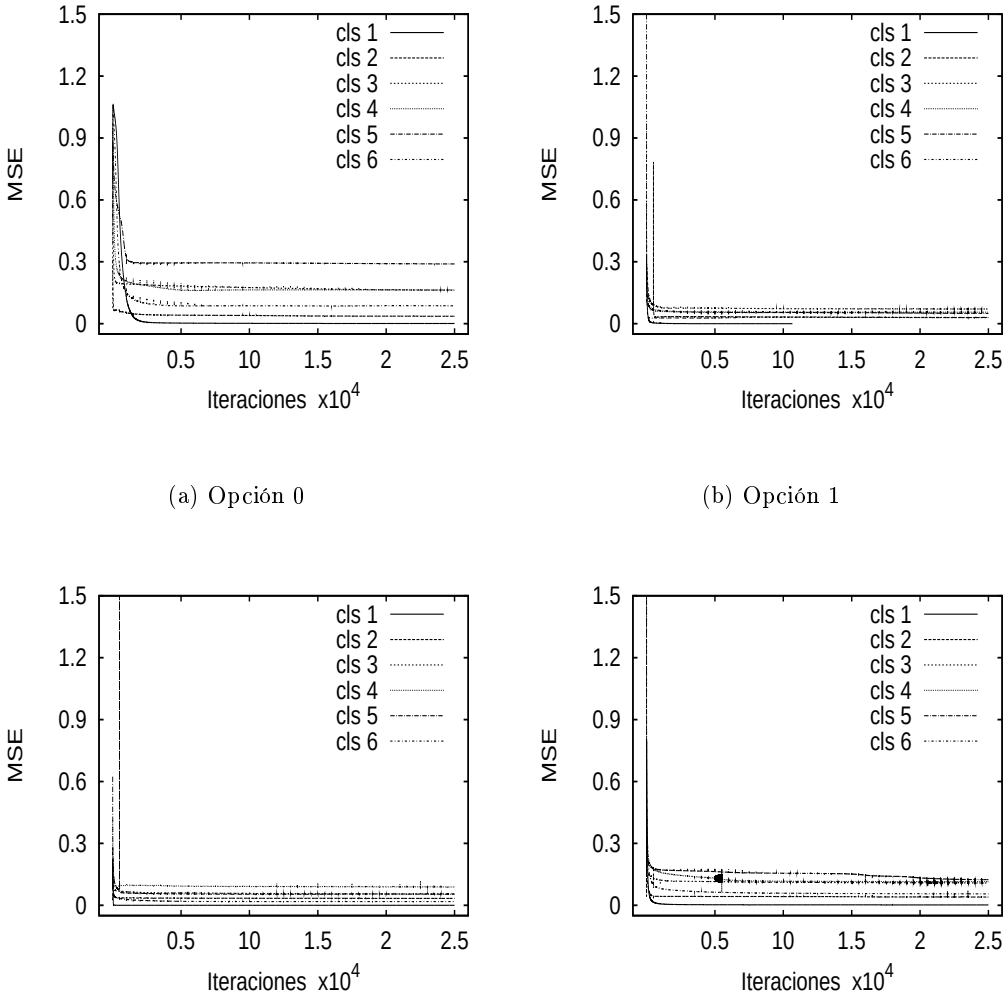
(b) Opción 1

(c) Opción 2

(d) Opción 3

Fig. 4.6: MSE de cada una de las clases de la base de datos Ecoli6 obtenido en el proceso de aprendizaje de la ANN-M sobre RBF.

se integraron las salidas de la ANN-M. Nuevamente aparece la interferencia entre clases, esta vez trasladado al esquema de votación. Así, sólo la Opción 3 consiguió



(a) Opción 0

(b) Opción 1

(c) Opción 2

(d) Opción 3

Fig. 4.7: MSE por clase de la base de datos Ecoli6 obtenido en la fase de entrenamiento de la ANN-M sobre RBF+VF.

una mejora en la precisión de esta clase.

De la misma forma que en la clase 5, en el resto de las clases el efecto de las

Tabla 4.2: Resultados obtenidos en la fase de clasificación por la ANN-M sobre MLP.

	Módulo de Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	94.15	
	cls 3	2.74	0.23	83.82	cls 5 (10.88)
	cls 4	1.82	0.16	82.55	
	cls 5	1.60	0.11	62.10	cls 3 (31.20)
	cls 6	3.23	0.06	85.20	
Opción 1	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	90.11	
	cls 3	2.74	0.23	78.72	cls 5 (11.25)
	cls 4	1.82	0.16	79.78	cls 6 (10.44)
	cls 5	1.60	0.11	60.47	cls 3 (28.57)
	cls 6	3.23	0.06	84.88	cls 4 (12.79)
Opción 2	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	89.22	
	cls 3	2.74	0.23	78.82	cls 5 (10.90)
	cls 4	1.82	0.16	78.41	cls 6 (10.68)
	cls 5	1.60	0.11	61.22	cls 3 (29.25)
	cls 6	3.23	0.06	85.71	cls 4 (10.12)
Opción 3	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	91.09	
	cls 3	2.74	0.23	78.55	cls 5 (15.65)
	cls 4	1.82	0.16	84.26	
	cls 5	1.60	0.11	65.08	cls 3 (26.98)
	cls 6	3.23	0.06	81.11	

funciones de coste produce una mejora de la PC de estas clases dentro del módulo cuando se clasifica individualmente. Sin embargo, se genera un aumento de la incertidumbre en la salida de la red cuando las muestras a evaluar pertenecen a otras clases, i.e., a las muestras que no son de la clase en la que se especializa el módulo.

Este hecho hace que cuando se unen todos los módulos y se aplica una estrategia basada en un esquema de votación simple aumente el nivel de error en las clases mayoritarias.

Para ilustrar lo anterior, en la Fig. 4.8 se muestran las salidas de los módulos correspondientes a las clases mayoritarias 2 y 3[7]. El eje x representa las muestras del conjunto de evaluación y el eje y indica el valor de la salida de la red para cada una de las muestras. Se muestra las salidas de la red del conjunto de muestras de test en los módulos de las clases 2 y 3. Los datos de evaluación han sido ordenados de izquierda a derecha desde la clase 1 a la 6 en función de su etiqueta. Además,

[7]La Fig. 4.8 corresponde al MLP y se muestra sobre dos clases únicamente con el objetivo de ejemplificar este comportamiento. Sin embargo, para las tres arquitecturas de ANN y el resto de las clases se tiene la misma tendencia en mayor o menor medida, por lo que no se consideró importante incluir sus respectivas figuras.

Tabla 4.3: Resultados obtenidos en la fase de clasificación por la ANN-M sobre RBF.

	Módulo de Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	92.34	
	cls 3	2.74	0.23	82.32	cls 5 (12.71)
	cls 4	1.82	0.16	86.07	
	cls 5	1.60	0.11	62.61	cls 3 (31.91)
	cls 6	3.23	0.06	86.17	cls 4 (10.11)
Opción 1	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	90.45	
	cls 3	2.74	0.23	83.67	cls 5 (12.57)
	cls 4	1.82	0.16	84.40	
	cls 5	1.60	0.11	60.00	cls 3 (35.87)
	cls 6	3.23	0.06	83.33	
Opción 2	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	92.04	
	cls 3	2.74	0.23	82.01	cls 5 (12.97)
	cls 4	1.82	0.16	84.02	
	cls 5	1.60	0.11	56.43	cls 3 (37.50)
	cls 6	3.23	0.06	87.50	
Opción 3	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	90.11	
	cls 3	2.74	0.23	83.40	cls 5 (13.00)
	cls 4	1.82	0.16	84.66	
	cls 5	1.60	0.11	65.05	cls 3 (31.91)
	cls 6	3.23	0.06	83.51	cls 4 (11.70)

se ha representado la Opción 0 (Op. 0) con linea continua y la Opción 2 con línea discontinua (Op. 2)

En estas imágenes se evidencia que la causa de la perdida de la efectividad del clasificador sobre las clases mayoritarias cuando las funciones de coste son aplicadas, es el incremento en la incertidumbre de las salidas de la red sobre las muestras que no corresponden a la clase en la que se especializó el módulo (véase la Fig. 4.8). Este efecto esta directamente relacionado con los niveles de confusión o solapamiento entre clases.

En el caso de la clase 2, que presenta bajos niveles de solapamiento, el aumento de la incertidumbre es mínimo (Fig. 4.8a), mientras que en la clase 3 que se encuentra fuertemente solapada, es más acentuado (Fig. 4.8b). En este último caso, aparecen valores altos en las salidas para el módulo de la clase 3 sobre muestras que no pertenecen a esta clase. Este hecho afecta de manera clara en un aumento del error cuando estas salidas obtenidas por cada módulo son introducidas en la fase de votación

Por otro lado, en la Tabla 4.5 se puede analizar la efectividad de los tres clasificadores, observándose como se reduce la precisión de clasificación cuando las opciones

Tabla 4.4: Resultados obtenidos en la fase de clasificación por la ANN-M sobre RBF+VF.

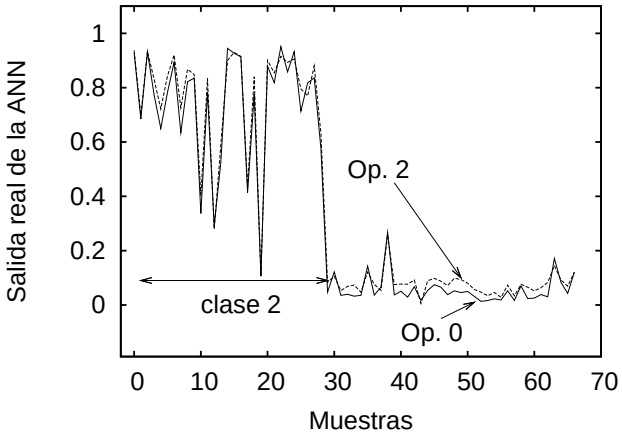
	Módulo de Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	92.18	
	cls 3	2.74	0.23	81.38	cls 5 (12.69)
	cls 4	1.82	0.16	87.68	
	cls 5	1.60	0.11	62.61	cls 3 (31.31)
	cls 6	3.23	0.06	87.23	
Opción 1	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	89.67	
	cls 3	2.74	0.23	83.78	cls 5 (11.36)
	cls 4	1.82	0.16	85.31	
	cls 5	1.60	0.11	57.47	cls 3 (37.34)
	cls 6	3.23	0.06	82.95	cls 4 (11.36)
Opción 2	cls 1	107.90	0.02	100.00	
	cls 2	2.79	0.43	90.75	
	cls 3	2.74	0.23	83.90	cls 5 (10.42)
	cls 4	1.82	0.16	83.57	
	cls 5	1.60	0.11	52.08	cls 3 (40.48)
	cls 6	3.23	0.06	83.85	cls 4 (10.42)
Opción 3	cls 1	107.90	0.02	98.00	
	cls 2	2.79	0.43	90.21	
	cls 3	2.74	0.23	83.38	cls 5 (11.82)
	cls 4	1.82	0.16	86.54	
	cls 5	1.60	0.11	67.14	cls 3 (28.29)
	cls 6	3.23	0.06	83.50	cls 4 (12.50)

1, 2 y 3 son incluidas al proceso de entrenamiento. Esta reducción está en coherencia con lo explicado en párrafos anteriores.

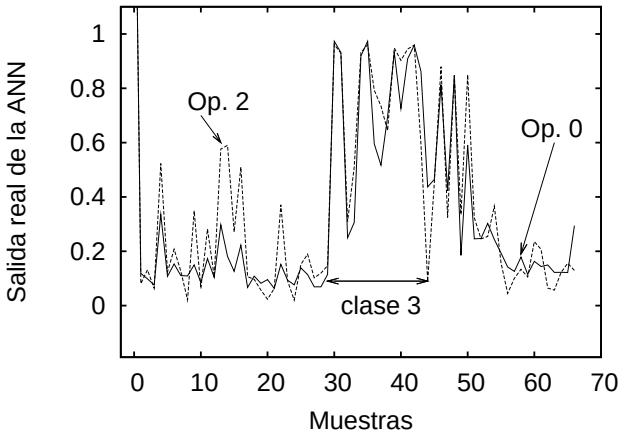
No obstante, no se presenta una diferencia significativa entre los resultados obtenidos al aplicar las opciones 1, 2 y 3, y los generados por la Opción 0. Este comportamiento fue visto previamente en el clasificador no modular del capitulo 3.

En esta sección se estudiaron dos cuestiones: convergencia y clasificación con la base de Ecoli6. En la siguiente sección, se continuará con este estudio pero sobre la base de datos Cayo que a diferencia de Ecoli6 está más representada y tiene 5 clases adicionales[8]. De esta forma se pretende mostrar a través de dos ejemplos prototipo (Ecoli6 y Cayo) desde un enfoque modular, el problema del desbalance de las clases en el contexto de tres arquitecturas distintas de ANN entrenadas con el algoritmo back-propagation con procesamiento por grupos.

[8] Además de presentar problemas serios de desbalance entre clases.



(a) Módulo especializado en la clase 2



(b) Módulo especializado en la clase 3

Fig. 4.8: Salidas de los módulos MLP asignados a las clases mayoritarias 2 y 3 con las opciones 0 y 2.

Tabla 4.5: Desempeño global del clasificador ANN-M con la base de datos Ecoli6. Los valores entre paréntesis hacen referencia a la desviación estándar.

ANN-M MLP	Opción 0	Opción 1	Opción 2	Opción 3
PC	86.12(2.79)	82.56(4.49)	82.11(5.23)	83.92(3.49
g-mean	82.90(5.35)	80.84(4.11)	80.83(4.61)	81.98(5.07)
ANN-M RBF	Opción 0	Opción 1	Opción 2	Opción 3
PC	85.64(4.15)	84.44(4.95)	84.54(4.91)	84.81(4.86)
g-mean	83.02(6.06)	81.94(6.38)	80.89(7.21)	82.91(6.00)
ANN-M RBF+VF	Opción 0	Opción 1	Opción 2	Opción 3
PC	85.66(4.36)	83.97(5.13)	83.68(4.61)	85.34(5.12)
g-mean	83.33(5.80)	80.94(6.83)	79.28(7.18)	83.68(6.15)

4.3.3 Base de datos Cayo: ANN-M

La base de datos Cayo ha sido objeto de estudio en otras investigaciones [Alejo 2008, Alejo 2009] debido a su naturaleza de múltiples clases y a su alto nivel desbalance. En esta sección se analiza el comportamiento de esta base de datos en el contexto de ANN-M.

La metodología empleada para estudiar al conjunto de datos Cayo fue la siguiente: Se dividió en 11 subconjuntos de datos de dos clases y posteriormente cada uno de estos conjuntos se utilizó para construir módulos especializados por clase como se observa en la Fig. 4.1. La finalidad de realizar este proceso es la de excluir la interferencia entre clases que existe cuando se trabaja con redes no modulares, y de esta forma centrarse en el tratamiento del desbalance de las clases. Para mayor detalle de los aspectos experimentales véase la sección 2.10.

El comportamiento de la ANN-M sobre MLP en esta base de datos no siguió el mostrado en la sección anterior con la base de datos Ecoli6 en términos de PC y *g-mean*. Se puede observar en la Tabla 4.6 que la ANN-M sobre MLP presentó un desempeño en la fase de clasificación inferior al mostrado por la ANN-G (Tabla 3.18) con la Opción 0. Esto se debe a que al pasar una base de datos de múltiples clases a 11 de dos clases el desbalance se acentúa considerablemente y la convergencia de las clases resulta ser mucho más lenta. Este hecho hace que en determinadas clases no se alcance un MSE tan bajo o tan reducido como el que se obtuvo para esas mismas clases en la ANN-G.

Por otro lado, las ANN-M sobre RBF y RBF+VF con la Opción 0 (Tabla 4.6) presentaron mejores resultados en términos de PC y *g-mean* que los resultados obtenidos con la ANN-G (ver Tabla 3.18). Al igual que ocurrió con Ecoli6 (ver Fig. 4.3), el MSE no se incremento y se aceleró la convergencia de las clases.

En la Fig. 4.9 se presenta el promedio de iteraciones necesarias para alcanzar un valor mínimo medio de MSE (por clase) tanto por el modelo ANN-M (color oscuro) como el ANN-G (color claro). Obsérvese que el modelo ANN-M (en términos generales) requiere de menos iteraciones que el ANN-G para alcanzar el promedio mínimo de MSE en la fase de entrenamiento.

Así mismo, esta figura evidencia que la ANN-M sobre MLP requieren de más iteraciones en los diferentes módulos para lograr el promedio mínimo de MSE en la fase de entrenamiento que la ANN-M sobre RBF y RBF+VF. Este hecho evidencia que en algunos de los módulos entrenados, las ANNs requieren de más iteraciones y en algunas ocasiones no alcanzan el mínimo de MSE deseado afectando a la PC.

En la Tabla 4.6 se muestran los resultados a nivel de PC global sobre las tres arquitecturas de ANN-M cuando se aplican las opciones 1, 2 y 3. Puede observarse que en el caso de la aplicación de la ANN-M sobre MLP se obtienen mejoras significativas al aplicar las funciones de coste tanto en valores de PC como de *g-mean*,

aunque son ligeramente inferiores a los mostrados por la ANN-G.

En el caso de las ANN-M sobre RBF y RBF+VF se puede observar que aunque a nivel de PC global no hay mejoras significativas respecto a la Opción 0, si que se obtiene mejores resultados de conjunto con mejoras en su *g-mean*.

Esto significa que el desbalance generado por la transformación de múltiples clases a dos clases no afectó la efectividad del clasificador modular, y además la inclusión de las opciones 1, 2 y 3 (al igual que en Ecoli6) aceleró la convergencia de las clases menos representadas en la ME.

Estos resultados confirman que las redes RBF y RBF+VF son fuertemente afectadas por la interferencia interna de las clases.

Tabla 4.6: Desempeño global del clasificador modular con la base de datos Cayo. Observe que los resultados corresponde a las tres arquitecturas de ANN-M sobre MLP, RBF y RBF+VF, así como a las opciones 1, 2 y 3. Los valores entre paréntesis hacen referencia a la desviación estándar.

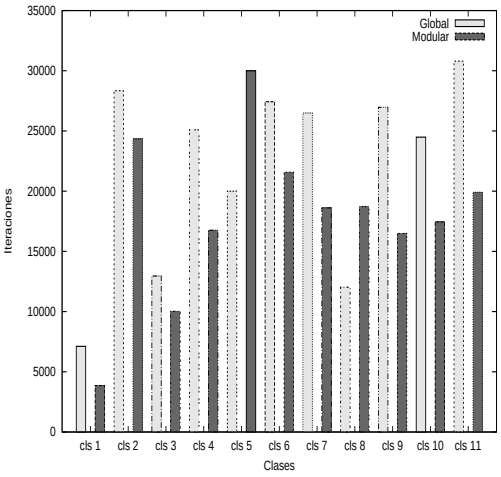
ANN-M MLP	Opción 0	Opción 1	Opción 2	Opción 3
PC	78.03(0.54)	83.34(0.68)	83.77(0.97)	84.40(0.40)
g-mean	46.70(4.70)	80.23(0.41)	80.58(0.42)	78.27(1.36)
ANN-M RBF	Opción 0	Opción 1	Opción 2	Opción 3
PC	83.48(0.85)	83.21(0.81)	82.94(1.37)	83.09(2.29)
g-mean	76.47(1.38)	78.93(1.23)	79.04(1.52)	78.28(2.66)
ANN-M RBF+VF	Opción 0	Opción 1	Opción 2	Opción 3
PC	82.56(1.14)	82.80(1.94)	83.70(0.90)	83.61(0.74)
g-mean	74.10(2.20)	77.71(3.67)	79.66(1.00)	78.92(0.64)

En las Tablas 4.7, 4.8 y 4.9 se muestra un mayor detalle de los resultados obtenidos con el clasificador ANN-M y la base de datos Cayo, mostrando la PC de cada uno de los módulos de clase y su nivel de confusión cuando el error es mayor que un 10%. Además, se puede ver en las Tablas 4.7, 4.8 y 4.9 que en las clases (a diferencia de Ecoli6) no se incrementó el nivel de confusión después de incluir las opciones 1, 2 y 3 al proceso de entrenamiento[9] en los tres modelos de ANN-M.

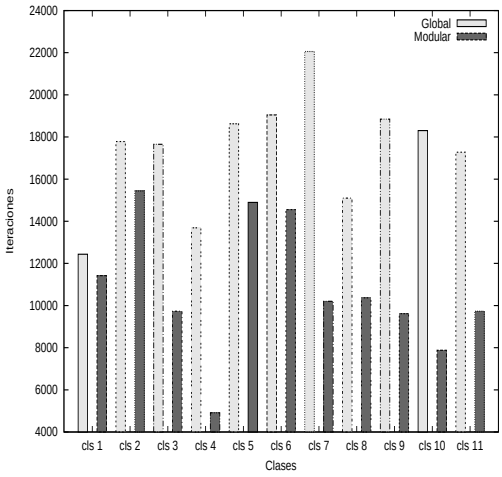
Las clases 2, 5, 6, 9 y 10 son las que muestran mayor confusión después de aplicar las funciones de coste. Estos resultados coinciden con los generados por los modelos ANN-G (Tablas 3.15, 3.19 y 3.20). Excepto por la clase 5 que en los modelos ANN-G al aplicar las opciones 1 y 2 se obtuvieron valores de PC aproximados al 90%.

En la ANN-M sobre MLP se puede observar como la aplicación de las opciones 1, 2 y 3 producen una mejora significativa en el PC de cada módulo de clase y disminuye de forma significativa el nivel de confusión.

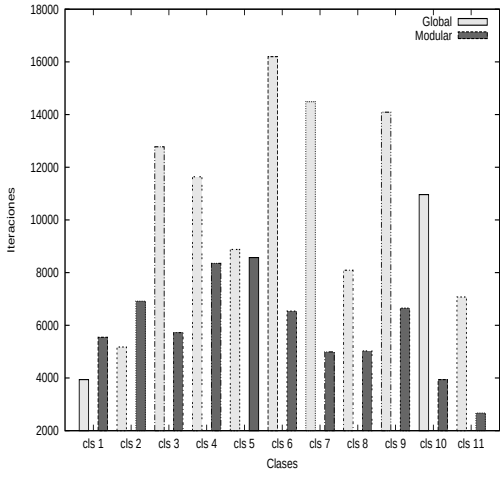
[9]Luego de reducir el problema del desbalance de las clases.



(a) ANN-M MLP



(b) ANN-M RBF



(c) ANN-M RBF+VF

Fig. 4.9: Comparación del número de iteraciones necesarias para alcanzar un valor promedio mínimo de MSE en ANN-M y ANN-G con la base de datos Cayo y la Opción 0.

Por ejemplo, la clase 5 (clase más minoritaria) evidencia una mejora significativa al incluir las opciones 1, 2 y 3 al proceso de entrenamiento. Obsérvese que para este modelo de ANN-M y con Opción 0, los valores de PC para esta clase son de aproximadamente 7.88%, y al aplicar las opciones 1, 2 y 3 se obtienen incrementos significativos alcanzando en el peor de los casos un 68.69% y en el mejor casos un 81.82%.

Las ANN-M sobre RBF y RBF+VF las opciones 1, 2 y 3 generan mejoras en la PC de las clases minoritarias (2, 4, 5 y 6), excepto la clase 7. En el caso específico de la clase 2 se presenta bajos valores de PC al aplicar las opciones 1, 2 y 3. Lo interesante de este caso reside en:

- Se trata de una clase minoritaria.
- En la fase de clasificación con la Opción 0 presenta valores (promedio de las tres ANN-M) de PC $\approx$ 48.7%.
- Al aplicar la opciones 1, 2 y 3 muestra valores (promedio de las tres ANN-M) de PC $\approx$ 54.3%.

Esto es debido a que se trata de una clase altamente solapada con la clase 3 pero la zona de la clase que no se encuentra en el área de confusión es lo suficientemente discriminante.

No obstante, los valores de F1 y los obtenidos por el criterio de los 3-NN no sugieren que exista un alto nivel de solapamiento entre este par de clases. Así, en este caso el clasificador k-NN y la medida F1 no aportan información suficiente para identificar el solapamiento o confusión entre clases en el contexto de las ANNs.

En el caso de las clases mayoritarias (1, 8, 9, 10 y 11) la aplicación de las funciones de coste no afecta de manera significativa su PC, ya que la base de datos Cayo a diferencia de Ecoli6, contiene una suficiente representatividad de muestras en las distribuciones de sus clases.

Tabla 4.7: Resultados obtenidos en la fase de clasificación por la ANN-M sobre MLP con la base de datos Cayo.

	Módulo de Clase	F1	Razón	PC	% confusión (> 10 %)
	cls 1	4.64	0.14	90.12	
	cls 2	0.83	0.05	45.66	cls 3 (48.63)
	cls 3	0.69	0.10	99.72	
	cls 4	0.56	0.05	4.47	cls 3 (68.90) cls 8 (23.64)
	cls 5	0.95	0.02	7.88	cls 1 (46.87) cls 3 (26.87) cls 8 (16.97)
Opción 0	cls 6	0.22	0.06	48.04	cls 7 (32.61)
	cls 7	0.54	0.05	79.47	cls 2 (16.48)
	cls 8	0.13	0.12	99.50	
	cls 9	0.93	0.13	87.56	cls 10 (12.44)
	cls 10	0.69	0.14	78.57	cls 11 (20.54)
	cls 11	0.95	0.13	85.56	cls 10 (14.44)
	cls 1	4.64	0.14	90.10	
	cls 2	0.83	0.05	54.68	cls 3 (40.53)
	cls 3	0.69	0.10	87.38	
	cls 4	0.56	0.05	95.03	
	cls 5	0.95	0.02	81.57	cls 1 (10.86)
Opción 1	cls 6	0.22	0.06	53.62	cls 7 (31.79)
	cls 7	0.54	0.05	98.67	
	cls 8	0.13	0.12	96.31	
	cls 9	0.93	0.13	86.84	cls 10 (12.44)
	cls 10	0.69	0.14	65.58	cls 11 (24.80)
	cls 11	0.95	0.13	90.16	
	cls 1	4.64	0.14	92.18	
	cls 2	0.83	0.05	55.48	cls 3 (40.41)
	cls 3	0.69	0.10	87.06	
	cls 4	0.56	0.05	95.03	
	cls 5	0.95	0.02	81.82	cls 1 (12.12)
Opción 2	cls 6	0.22	0.06	52.72	cls 7 (31.66)
	cls 7	0.54	0.05	98.93	
	cls 8	0.13	0.12	96.47	
	cls 9	0.93	0.13	86.93	cls 10 (12.44)
	cls 10	0.69	0.14	66.35	cls 11 (24.22)
	cls 11	0.95	0.13	90.35	
	cls 1	4.64	0.14	93.72	
	cls 2	0.83	0.05	45.09	cls 3 (48.40)
	cls 3	0.69	0.10	93.29	
	cls 4	0.56	0.05	95.65	
	cls 5	0.95	0.02	68.69	cls 1 (22.22)
Opción 3	cls 6	0.22	0.06	50.00	cls 7 (31.43)
	cls 7	0.54	0.05	93.66	
	cls 8	0.13	0.12	97.37	
	cls 9	0.93	0.13	87.56	cls 10 (12.44)
	cls 10	0.69	0.14	73.64	cls 11 (24.68)
	cls 11	0.95	0.13	88.64	

Tabla 4.8: Resultados obtenidos en la fase de clasificación por la ANN-M sobre RBF con la base de datos Cayo.

	Módulo de Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	4.64	0.14	89.67	
	cls 2	0.83	0.05	50.73	cls 3 (48.58)
	cls 3	0.69	0.10	91.88	
	cls 4	0.56	0.05	84.18	
	cls 5	0.95	0.02	49.80	cls 1 (28.99) cls 3 (12.53)
	cls 6	0.22	0.06	58.33	cls 7 (30.29)
	cls 7	0.54	0.05	94.27	
	cls 8	0.13	0.12	97.62	
	cls 9	0.93	0.13	87.53	cls 10 (12.44)
	cls 10	0.69	0.14	70.42	cls 11 (23.51)
	cls 11	0.95	0.13	91.92	
Opción 1	cls 1	4.64	0.14	89.47	
	cls 2	0.83	0.05	56.16	cls 3 (41.32)
	cls 3	0.69	0.10	85.48	
	cls 4	0.56	0.05	85.44	
	cls 5	0.95	0.02	79.63	cls 1 (13.64)
	cls 6	0.22	0.06	57.00	cls 7 (29.53)
	cls 7	0.54	0.05	86.16	
	cls 8	0.13	0.12	97.38	
	cls 9	0.93	0.13	87.56	cls 10 (12.44)
	cls 10	0.69	0.14	68.54	cls 11 (25.51)
	cls 11	0.95	0.13	93.84	
Opción 2	cls 1	4.64	0.14	89.66	
	cls 2	0.83	0.05	58.72	cls 3 (40.91)
	cls 3	0.69	0.10	84.71	
	cls 4	0.56	0.05	87.25	
	cls 5	0.95	0.02	77.78	cls 1 (13.43)
	cls 6	0.22	0.06	53.51	cls 7 (31.20)
	cls 7	0.54	0.05	92.92	
	cls 8	0.13	0.12	96.99	
	cls 9	0.93	0.13	86.72	cls 10 (11.93)
	cls 10	0.69	0.14	68.73	cls 11 (26.19)
	cls 11	0.95	0.13	90.62	
Opción 3	cls 1	4.64	0.14	89.27	
	cls 2	0.83	0.05	51.33	cls 3 (48.59)
	cls 3	0.69	0.10	92.16	
	cls 4	0.56	0.05	87.72	
	cls 5	0.95	0.02	70.14	cls 1 (20.41)
	cls 6	0.22	0.06	60.52	cls 7 (29.70)
	cls 7	0.54	0.05	91.05	
	cls 8	0.13	0.12	97.33	
	cls 9	0.93	0.13	87.56	cls 10 (12.44)
	cls 10	0.69	0.14	70.55	cls 11 (22.88)
	cls 11	0.95	0.13	84.35	

Tabla 4.9: Resultados obtenidos en la fase de clasificación por la ANN-M sobre RBF+VF con la base de datos Cayo.

	Módulo de Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	4.64	0.14	88.94	
	cls 2	0.83	0.05	49.71	cls 3 (48.63)
	cls 3	0.69	0.10	93.40	
	cls 4	0.56	0.05	77.38	
	cls 5	0.95	0.02	43.18	cls 1 (30.43) cls 3 (13.13) cls 8 (13.26)
	cls 6	0.22	0.06	57.25	cls 7 (31.11)
	cls 7	0.54	0.05	92.84	
	cls 8	0.13	0.12	98.15	
	cls 9	0.93	0.13	87.56	cls 10 (12.44)
	cls 10	0.69	0.14	68.99	cls 11 (23.28)
	cls 11	0.95	0.13	90.80	
Opción 1	cls 1	4.64	0.14	90.65	
	cls 2	0.83	0.05	62.10	cls 3 (36.07)
	cls 3	0.69	0.10	83.60	
	cls 4	0.56	0.05	84.47	
	cls 5	0.95	0.02	71.46	cls 1 (15.78)
	cls 6	0.22	0.06	56.30	cls 7 (28.17)
	cls 7	0.54	0.05	84.82	
	cls 8	0.13	0.12	97.30	
	cls 9	0.93	0.13	87.46	cls 10 (12.44)
	cls 10	0.69	0.14	69.71	cls 11 (24.28)
	cls 11	0.95	0.13	90.31	
Opción 2	cls 1	4.64	0.14	89.15	
	cls 2	0.83	0.05	54.41	cls 3 (44.98)
	cls 3	0.69	0.10	89.21	
	cls 4	0.56	0.05	87.99	
	cls 5	0.95	0.02	78.62	cls 1 (14.31)
	cls 6	0.22	0.06	54.47	cls 7 (31.76)
	cls 7	0.54	0.05	95.16	
	cls 8	0.13	0.12	96.77	
	cls 9	0.93	0.13	87.14	cls 10 (12.44)
	cls 10	0.69	0.14	69.87	cls 11 (24.97)
	cls 11	0.95	0.13	91.77	
Opción 3	cls 1	4.64	0.14	88.74	
	cls 2	0.83	0.05	51.16	cls 3 (48.63)
	cls 3	0.69	0.10	91.02	
	cls 4	0.56	0.05	87.27	
	cls 5	0.95	0.02	69.09	cls 1 (22.73)
	cls 6	0.22	0.06	60.16	cls 7 (29.84)
	cls 7	0.54	0.05	92.09	
	cls 8	0.13	0.12	97.59	
	cls 9	0.93	0.13	86.19	cls 10 (12.44)
	cls 10	0.69	0.14	68.49	cls 11 (24.64)
	cls 11	0.95	0.13	93.45	

4.4 Comunicación entre módulos

Una vez efectuado el análisis del problema, construidos y ajustados los módulos que se hacen responsables de la resolución de cada subproblema, es necesario especificar el mecanismo que integre cada una de las soluciones parciales alcanzadas para crear la solución al problema original.

De acuerdo a la manera en que se ha realizado el reparto de la información, se pueden distinguir diferentes métodos de integración o combinación de los módulos que se hayan considerado [Haykin 1999]. El esquema de votación simple es uno de los métodos más utilizado para realizar la combinación de las decisiones individuales. Sin embargo, estudios realizados han demostrado la debilidad de este método cuando el desempeño de los componentes individuales no es uniforme [Valdovinos 2006].

En este trabajo se ha utilizado el esquema de votación simple (winners-take-all) para la integración de las salidas de los módulos independientes de la ANN-M. Sin embargo, es necesario el estudio y evaluación de otros mecanismos para la integración de las salidas.

En esta sección se explora la posibilidad de una alternativa adicional a este mecanismo. El objetivo que se persigue con esta experimentación es explicar porque la construcción adecuada de cada módulo independiente, así como, la integración apropiada de sus salidas puede mejorar los resultados.

Una opción al esquema de votación simple es la de los sistemas cooperativos. En estos sistemas se hace uso de las salidas de los expertos como entradas a otros expertos [Bottou 1990].

Supóngase que $r_k(\mathbf{x}_n)$ es la salida a la entrada $\mathbf{x}_n$ ($n = 1, ..., N$; y $N = \|\mathrm{ME}\|$) al experto r_k y $\mathbf{d}_n$ corresponde a la salida deseada para $\mathbf{x}_n$, entonces se puede incluir un factor de ponderación $\mathbf{P}$ que optimice las salidas de los expertos r_k antes de que se tome la decisión final. Esto se puede expresar de forma matricial como sigue:

$$\mathbf{R}(\mathbf{X})\mathbf{P} = \mathbf{D}\,. \tag{4.1}$$

La Ec. 4.1 representa un problema de optimización lineal y el factor de ponderación $\mathbf{P}$ puede ser obtenido de forma automática.

A partir de la expresión 4.1 se puede calcular $\mathbf{P}$ como sigue

$$\mathbf{P} = \mathbf{R}(\mathbf{X})^{+}\mathbf{D} = (\mathbf{R}(\mathbf{X})^{\mathbf{T}}\mathbf{H})^{-\mathbf{1}}\mathbf{R}(\mathbf{X})^{\mathbf{T}}\mathbf{D}\,, \tag{4.2}$$

donde $\mathbf{R}(\mathbf{X})^{+}$ es la matriz *pseudo-inversa* de $\mathbf{R}(\mathbf{X})$. No obstante, esta forma de calcular el factor de ponderación $\mathbf{P}$ es susceptible al problema del desbalance de las clases [Fu 2002]. Por lo tanto, es necesario incluir en este método un elemento adicional que compense el desbalance de las clases.

Considérese que el MSE asociado a la Ec. 4.1 puede ser expresado como sigue:

$$E(P) = \frac{1}{2}\sum_{i=1}^{M}\sum_{n_i=1}^{N_i}\sum_{m=1}^{M}\left\{\sum_{j=0}^{K} p_{mj} r_j(\mathbf{x}_m^{n_i}) - d_m^{n_i}\right\}^2 , \tag{4.3}$$

donde M es el número de salidas o clases, K el número de expertos, N_i el número de muestras de la clase i y N el total de muestras en la ME.

Para incrementar la contribución de las clases menos representadas en la ME se sustituye la Ec. 4.3 por

$$E(P) = \frac{1}{2}\sum_{i=1}^{M}\beta_i\sum_{n_i=1}^{N_i}\sum_{m=1}^{M}\{\sum_{j=0}^{K} p_{mj} r_j(\mathbf{x}_m^{n_i}) - d_m^{n_i}\}^2 , \tag{4.4}$$

donde $\beta_i = \frac{N}{N_i}$, $i = 1, 2, ..., M$. β_i corresponde al factor de compensación del desbalance de la ME. Al calcular $\frac{\partial E(P)}{\partial p_{mj}} = 0$ se tiene

$$\sum_{i=1}^{M}\beta_i\sum_{n_i=1}^{N_i}\{\sum_{j'=0}^{K} p_{mj} r_{j'}(\mathbf{x}_m^{n_i}) - d_m^{n_i}\} r_j(\mathbf{x}_m^{n_i}) = 0 . \tag{4.5}$$

Si $c_n = \beta_i$ donde $\mathbf{x}^n \in A_i$ y A_i es la clase i, entonces se puede substituir c_n en la Ec. 4.5 para obtener

$$\sum_{n=1}^{N} c_n\{\sum_{j'=0}^{K} p_{mj} r_{j'}(\mathbf{x}_m^{n_i}) - d_m^{n_i}\} r_j(\mathbf{x}_m^{n_i}) = 0 , \tag{4.6}$$

reemplazando c_n con $\sqrt{c_n}\cdot\sqrt{c_n}$ se obtiene

$$\sum_{n=1}^{N}\{\sum_{j'=0}^{K} p_{mj} r_{j'}(\mathbf{x}_m^{n_i})\cdot\sqrt{c_n} - d_m^{n_i}\cdot\sqrt{c_n}\} r_j(\mathbf{x}_m^{n_i})\cdot\sqrt{c_n} = 0 , \tag{4.7}$$

y de esta forma se puede establecer $r \rightarrow r_j^n \cdot \sqrt{c_n}$ y $d \rightarrow d_j^n \cdot \sqrt{c_n}$, y así, replantear la Ec. 4.1 con la diferencia de que ahora se esta considerando el desbalance de las clases.

Para evaluar las posibilidades de la propuesta anterior se desarrollaron una serie de experimentos[10] con las bases de datos Cayo, Ecoli6, Feltwell y Satimage. Obsérvese que se trata de cuatro conjuntos de datos desequilibrados y de múltiples clases.

[10] Para mayor detalle acerca de los aspectos experimentales consúltese la sección 2.10.

El procedimiento que se siguió para este análisis consistió en dividir cada conjunto de datos en K subproblemas de dos clases, y cada uno de ellos fue resuelto por una ANN independiente de la misma forma que se hizo en las subsecciones 4.3.2 y 4.3.3.

Una vez construido y entrenado cada módulo de la ANN-M se procedió a aplicar la Ec. 4.1 y a partir de la Ec. 2.23 (considerándose los factores de la Ec. 4.7) se obtuvo la matriz de optimización ($\mathbf{P}$), y desde $\mathbf{P}$ se generaron los resultados de la Tabla 4.11. La Tabla 4.10 es presentada con la finalidad de comparar los resultados obtenidos por el esquema de votación simple con el mecanismo de sistemas cooperativos (Tabla 4.11).

Tabla 4.10: Resultados de la fase de clasificación de la ANN-M (en sus tres versiones) con el *esquema de votación simple*. Los valores entre paréntesis hacen referencia a la desviación estándar.

	ANN-M MLP		ANN-M RBF		ANN-M RBF+VF	
	Salidas SIN Ponderar					
Opción 0	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cayo	78.03(0.54)	46.70(4.70)	83.48(0.85)	76.47(1.38)	82.56(1.14)	74.10(2.20)
Ecoli6	86.12(2.79)	82.90(5.35)	85.64(4.15)	83.02(6.06)	85.66(4.36)	83.33(5.80)
Feltwell	89.63(0.66)	86.93(0.88)	87.02(2.86)	80.98(6.87)	88.42(0.81)	84.72(1.33)
Satimage	81.70(0.80)	48.27(5.18)	83.63(2.22)	73.86(7.76)	84.85(0.77)	77.31(1.35)
Opción 1	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cayo	83.34(0.68)	80.23(0.41)	83.21(0.81)	78.93(1.23)	82.80(1.94)	77.71(3.67)
Ecoli6	82.56(4.49)	80.84(4.11)	84.44(4.95)	81.94(6.38)	83.97(5.13)	80.94(6.83)
Feltwell	86.47(0.84)	83.94(0.89)	88.54(1.35)	86.61(1.79)	88.00(0.97)	84.87(1.40)
Satimage	84.88(0.79)	82.91(0.90)	86.15(0.85)	84.10(0.59)	86.04(0.75)	84.35(0.90)
Opción 2	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cayo	83.77(0.97)	80.58(0.42)	82.94(1.37)	79.04(1.52)	83.70(0.90)	79.66(1.00)
Ecoli6	82.11(5.23)	80.83(4.61)	84.54(4.91)	80.89(7.21)	83.68(4.61)	79.28(7.18)
Feltwell	86.12(0.85)	83.46(0.88)	88.26(1.07)	86.38(1.30)	88.19(0.89)	84.93(1.19)
Satimage	85.78(0.58)	83.84(0.46)	86.08(0.88)	84.48(0.77)	86.14(0.94)	84.59(1.06)
Opción 3	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cayo	84.40(0.40)	78.27(1.36)	83.09(2.29)	78.28(2.66)	83.61(0.74)	78.92(0.64)
Ecoli6	83.92(3.49)	81.98(5.07)	84.81(4.86)	82.91(6.00)	85.34(5.12)	83.68(6.15)
Feltwell	88.34(0.57)	85.67(0.65)	90.09(0.47)	87.97(0.59)	89.39(1.21)	86.63(1.65)
Satimage	85.16(0.54)	81.80(1.23)	85.55(0.94)	82.20(1.54)	85.74(0.98)	81.94(1.41)

Al comparar las Tablas 4.10 y 4.11 no se observa en la mayoría de los experimentos una diferencia significativa en cuanto a la PC y los valores de *g-mean* en ambos enfoques. Esto quiere decir que la ponderación de las salidas de los expertos no aportó información lo suficientemente relevante como para incrementar la efectividad del clasificador.

Como punto a favor de los sistemas cooperativos se puede analizar la base de

Tabla 4.11: Resultados obtenidos en la fase de clasificación por la red neuronal modular (en sus tres versiones) con el enfoque de *sistemas cooperativos*. Los valores entre paréntesis hacen referencia a la desviación estándar.

	ANN-M MLP		ANN-M RBF		ANN-M RBF+VF	
	Salidas Ponderadas					
Opción 0	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cayo	83.98(0.36)	79.64(0.65)	83.38(1.29)	80.33(1.46)	82.95(0.80)	80.15(0.99)
Ecoli6	85.88(2.67)	82.60(5.08)	85.45(4.35)	83.13(5.96)	85.36(4.56)	82.94(5.76)
Feltwell	89.88(0.65)	87.33(0.84)	88.18(1.92)	85.22(2.83)	88.99(0.71)	86.12(0.94)
Satimage	82.43(0.92)	74.73(1.99)	84.74(0.99)	81.46(1.19)	85.53(0.58)	81.94(0.96)
Opción 1	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cayo	82.83(0.67)	82.35(0.45)	83.49(0.81)	81.02(0.76)	84.00(1.13)	81.16(1.25)
Ecoli6	83.18(4.30)	80.28(4.72)	84.94(4.87)	82.73(6.24)	84.37(5.13)	81.50(7.13)
Feltwell	87.11(0.90)	84.43(0.94)	87.83(1.22)	86.05(1.43)	87.66(0.87)	84.53(1.25)
Satimage	83.82(0.35)	83.32(0.36)	85.92(0.65)	84.34(0.68)	85.11(0.59)	83.96(0.76)
Opción 2	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cayo	83.15(0.61)	82.69(0.44)	83.38(1.00)	81.05(0.62)	83.51(0.82)	81.18(0.81)
Ecoli6	83.12(4.60)	80.29(4.95)	84.13(5.17)	80.75(8.01)	84.73(4.63)	82.21(5.92)
Feltwell	86.64(0.86)	83.78(0.91)	87.92(0.88)	86.06(1.01)	87.85(0.87)	84.59(1.11)
Satimage	84.37(0.60)	83.91(0.64)	85.69(0.63)	84.48(0.79)	85.47(0.90)	84.43(1.03)
Opción 3	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cayo	83.36(0.45)	81.07(0.89)	84.14(0.74)	81.23(0.71)	83.82(0.68)	81.08(0.62)
Ecoli6	83.62(3.74)	81.26(4.88)	84.88(4.60)	82.87(6.06)	85.12(5.22)	83.37(6.20)
Feltwell	88.55(0.54)	85.92(0.60)	89.25(0.69)	87.31(0.79)	88.90(1.23)	86.27(1.60)
Satimage	84.53(0.58)	83.49(1.16)	85.28(0.56)	83.59(0.63)	85.69(0.65)	83.77(0.68)

datos Cayo donde al optimizar las salidas de los módulos se logró incrementar el porcentaje de aciertos de algunas clases, especialmente la clase 5. Esto trajo como consecuencia un incremento significativo en sus valores de *g-mean*.

Estos resultados muestran el interés de construir y entrenar adecuadamente cada módulo de la red para obtener un buen resultado. Así, se observa que si han sido diseñados y ajustados correctamente cada módulo, es suficiente la aplicación de un mecanismo de integración sencillo como el esquema de votación simple.

En esta experimentación, el uso de un mecanismo de ponderación de las salidas de los módulos no resuelve todas las deficiencias ocasionadas por un mal diseño o entrenamiento de los módulos individuales. Sin embargo, la aplicación de este tipo de mecanismos ayudan a reducir algunas imperfecciones generadas en el proceso de aprendizaje. En el caso de la base de datos Cayo, las deficiencias de los expertos individuales fueron mejoradas al aplicar el factor de ponderación $\mathbf{P}$.

Es necesario prestar mucha atención al ajuste y construcción correcta de los

módulos de la red. De esta forma, el estudio de las técnicas de integración de las salidas de los expertos es un tema de amplio estudio en la actualidad[11] y no se le debe restar importancia.

4.5 Conclusión

A lo largo de este capítulo se ha discutido la viabilidad de descomponer un problema de múltiples clases en subproblemas de dos clases para tratar de reducir la interferencia interna entre las clases.

Se han visto notables beneficios en cuestiones de convergencia al entrenar módulos especializados por clase. Así mismo, se ha evidenciado que al descomponer un PDMC en subproblemas de dos clases, se tienen resultados en la fase de clasificación en valores de PC y *g-mean* de similar magnitud a los obtenidos con el PDMC original. No obstante, en las redes RBF y RBF+VF se ha detectado la tendencia a producir mejores resultados cuando el enfoque modular es aplicado. En el caso del MLP no esta clara esta tendencia en la Opción 0.

Además, se ha observado que entrenar de manera adecuada los módulos independientes combinando las salidas a través de un mecanismo simple (por ejemplo por *votación*) funciona adecuadamente. La utilidad de los mecanismos de integración de las salidas en ciertas situaciones pueden ayudar a superar determinadas deficiencias del entrenamiento y construcción de algunos de los módulos.

En términos generales, y a partir de los resultados presentados en este capítulo se puede concluir lo siguiente:

- Tratar un PDMC como subproblemas de dos clases reduce la interferencia interna entre clases, lo que permite que el problema sea más fácil de aprender por la red neuronal. Esto se ve reflejado en cuestiones de convergencia y clasificación donde al aplicar este enfoque se mejora la PC en los modelos de ANN-M sobre RBF y RBF+VF.

- En la ANN-M sobre MLP esta descomposición incrementa el MSE asociado a las clases menos representadas. No obstante, la aplicación de funciones de coste permiten tratar el desbalance de las clases (opciones 1, 2 y 3).

- En algunas bases de datos como Ecoli6 y Feltwell, la aplicación de las opciones 1 y 2 causan una perdida de efectividad en el clasificador. Esto es debido a que estos conjuntos de datos son afectados en menor medida por el desbalance de las clases, y al incluir funciones de coste en el proceso de entrenamiento se puede estar sobre ajustando la ANN.

[11] Por ejemplo en [Valdovinos 2006] se puede encontrar una discusión detallada sobre este tema.

Es indudable que tener módulos independientes es de gran utilidad porque pueden ser tratados de forma distinta dependiendo de la clase que se trate, es decir, si es una clase solapada o con ruido. En estos casos se le podría aplicar alguna estrategia apropiada (edición de datos, transformación del espacio, etc.) para tratar de disminuir estos problemas.

La clave en la construcción de las ANN-M está en un diseño correcto de cada módulo y su respectivo tratamiento respecto al desbalance de las clases. Así mismo, es importante la aplicación de algún mecanismo para la combinación de las salidas de los módulos, que supere algunos de los inconvenientes generados por un mal diseño.

Nótese que al descomponer un PDMC en problemas de dos clases se reduce la dificultad en el entrenamiento de la red y se consume menos recursos en tiempo de computación.

Capítulo 5

Corrección de los datos

Contenido

5.1 Introducción

En capítulos previos se ha discutido que es suficiente la aplicación de alguna estrategia para equilibrar las aportaciones de error al proceso de entrenamiento, acelerando la convergencia de las clases menos representadas, y así obtener mejores resultados en la fase de clasificación.

No obstante, cuando aparece el factor solapamiento o separabilidad entre clases, el desbalance de las clases afecta en mayor medida al clasificador, y no es suficiente la aplicación de una función de coste para tratar el desequilibrio de las aportaciones de error al proceso de aprendizaje. Por ello, es de interés estudiar algunas alternativas que ayuden reducir el área de solapamiento entre clases, y así mejorar la efectividad del clasificador sobre las clases minoritarias.

En este capítulo se estudia la idea de reducir la región de solapamiento entre clases a partir del uso de técnicas de tomadas del contexto de la regla del vecino más próximo, para incrementar la precisión del clasificador sobre las clases menos representadas. Estas técnicas se basan en la eliminación de aquellos elementos de la

ME que puedan ser considerados sospechosos de atipicidad, estar mal etiquetados o encontrarse en el área de confusión. Se debe aclarar que el objetivo de este enfoque no es lograr el balance de las clases.

El estudio fue realizado sobre tres arquitecturas distintas de ANNs (MLP, RBF y RBF+VF) entrenadas con el algoritmo back-propagation con procesamiento por grupos.

5.2 Edición de Wilson y algunas variantes

La Edición de Wilson (EW) [Wilson 1972] fue diseñada con el objetivo de incrementar la capacidad de generalización de la regla NN. Sin embargo, es reportada con frecuencia como otra técnica para la reducción del tamaño de la ME por tener también esa propiedad [Wilson 2000]. Se han publicado diversas modificaciones o variantes de la EW y todas ellas han mostrado una disminución importante de los errores de clasificación, pero la reducción en el tamaño de la ME no ha sido apreciable (sólo se reduce en aproximadamente 25% [Barandela 2003b]).

El procedimiento consiste en eliminar aquellas muestras de la ME de las que se tenga la sospecha de la autenticidad de sus etiquetas, de ser atípicas o de ubicarse en el área de solapamiento entre clases, y que consecuentemente originen errores en la clasificación. Este método puede ser expresado algorítmicamente como:

- Inicialización: $S \leftarrow$ ME
- Para cada elemento $\mathbf{x}_i \in$ ME
  - Buscar $K-$NN a $\mathbf{x}_i$ ($\mathbf{x}_j^k \in$ ME, $i \neq j$ y $\mathbf{x}_j^k = \min\|(\mathbf{x}_i, \mathbf{x}_j^k)\|$ donde $k = 1, ..., K$)
  - Si la mayoría de los $K-$NN son de distinta clase a $\mathbf{x}_i$, el elemento es eliminado de S
- Finalización: ME $\leftarrow S$

En este capítulo la EW es modificada con el objetivo de tratar el problema del desbalance de las clases desde un enfoque distinto, i.e., no busca equilibrar de alguna forma las clases de la ME, sino trata de mejorar la calidad de ésta reduciendo el área de solapamiento de las clases minoritarias, eliminando aquellas muestras de la clase mayoritaria que se consideren atípicas o incorrectamente etiquetadas. Las técnicas propuestas en este trabajo son las siguientes:

- EW sobre la clase mayoritaria (EW$^-$). El objetivo de esta técnica es impedir la eliminación de muestras de la clase minoritaria. No se eliminan elementos

de la clase minoritaria para prevenir la acentuación del desbalance y evitar la perdida de información relevante.

Al reducir la región de confusión de la clase menos representada se busca mejorar la precisión del clasificador sobre esta clase, y perjudicar lo menos posible a la PC de la clase mayoritaria.

- Edición de Wilson con distancia ponderada (EWP) y la edición de Wilson con distancia ponderada sobre la clase mayoritaria (EWP$^-$). Estas dos técnicas corresponden a la modificación de la EW en la que se incluye un factor de ponderación al cálculo de las distancias entre muestras como se ilustra en la Ec. 5.1.

$$d_P(\mathbf{x}_i, \mathbf{x}_j) = \left(\frac{n_c}{N}\right)^{1/m} \cdot d_E(\mathbf{x}_i, \mathbf{x}_j)\,, \tag{5.1}$$

donde d_E representa la distancia euclídea entre $\{\mathbf{x}_i, \mathbf{x}_j\} \in$ ME, $i \neq j$, m la dimensión de $\mathbf{x}$, n_c es el número de muestras de la clase c y N el total de muestras en la ME.

La ponderación fue propuesta en el contexto de la regla NN para compensar el desbalance de las clases [Barandela 2004].

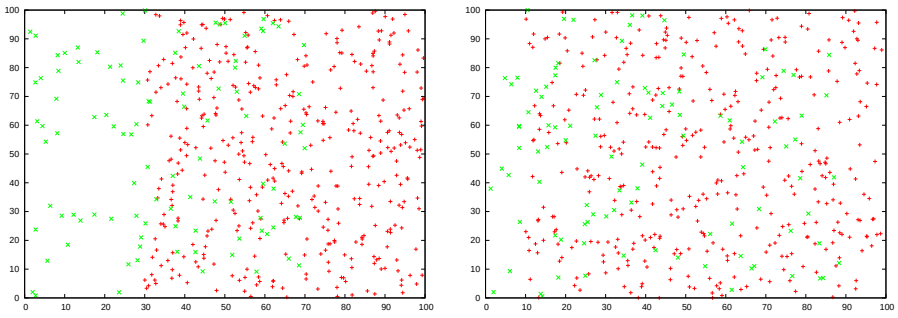
5.3 Aspectos metodológicos

Para estudiar y evaluar la conveniencia de las estrategias propuestas en la sección 5.2 se desarrollaron una serie de experimentos con bases de datos sintéticas y reales.

Se generaron seis conjuntos de datos artificiales: A0, A20, A40, A60, A80 y A100. Cada uno de ellos cuenta con 20 bases de datos (10 entrenamiento y 10 de evaluación) de dos clases distribuidas uniformemente (con 100 y 400 muestras respectivamente) y dos atributos o características.

Los valores 0, 20, 40, 60, 80 y 100 corresponden a los diferentes niveles de solapamiento entre clases. Por ejemplo, A0 representa una base de datos sin solapamiento entre clases, A40 es una base de datos con el 40% de sus elementos ubicados en el área de solapamiento (Fig. 5.1a), y A80 corresponde a una base de datos con el 80% de sus elementos en la región de solape (Fig. 5.1b). Cada base de datos contienen 500 muestras para el conjunto de entrenamiento y 500 muestras para el conjunto de evaluación. El entrenamiento fue repetido 10 veces en todas las ANNs.

El objetivo de utilizar bases de datos sintéticas y solapadas es la de observar en condiciones controladas las mejoras que se tienen en la PC de la clase minoritaria, y los efectos sobre de la PC de la clase mayoritaria al reducir el área de confusión de la clase menos representada.



(a) 40% de solapamiento (b) 80% de solapamiento

Fig. 5.1: Ejemplos de bases de datos artificiales con diferentes niveles de solapamiento: 40% y 80% respectivamente.

Posteriormente, el estudio es extendido a bases de datos reales. Se incluyeron cuatro bases de datos de dos clases (German, Diabetes, Ionosphere y Phoneme) del UCI Database Repository [Newman 1998], y dos de múltiples clases (Ecoli6 y Cayo). Para evaluar la conveniencia de preprocesar la ME con técnicas tomadas en el contexto de la regla NN y así tratar el desbalance, se utilizaron los criterios de medida PC global, PC por clase y *g-mean*. Para mayor detalle acerca de los aspectos experimentales consúltese la sección 2.10.

Por otra parte, es necesario aclarar que en las secciones dedicadas a problemas de dos clases (5.4 y 5.5.1), se incluyeron los resultados obtenidos por el clasificador 3-NN con el objetivo de que el lector pueda observar en términos generales, que las estrategias propuestas en este capítulo presentan una tendencia muy similar tanto en el contexto de la regla NN como en el de las ANNs. Sin embargo, la discusión a lo largo de este capítulo se centra en el ámbito de las ANNs (que son el modelo estudiado en esta investigación) y se hace poco uso de los resultados correspondientes al 3-NN.

5.4 Estudio con conjuntos de datos sintéticos

En esta sección se discuten brevemente algunos de los principales resultados generados al experimentar con bases de datos sintéticas. A cada base de datos se le aplicó EW, EWP (EW con distancia ponderada), EW^- y EWP^-. Las técnicas EW y EWP son utilizadas con fines comparativos y exploratorios.

La EW es un mecanismo que es aplicado a ambas clases (en problemas de dos clases), y esta diseñado para trabajar con bases de datos relativamente equilibradas. Por lo tanto, la primera cuestión que se debe resolver al iniciar la discusión es ¿cuál es el efecto de la EW sobre bases de datos equilibradas y con diferentes niveles de solapamiento?.

La respuesta a esta pregunta se encuentra en la Tabla 5.1[1]. Obsérvese en la experimentación mostrada que para el caso de las ANNs se obtienen resultados similares en PC y *g-mean*, mientras que para la regla NN sólo en algunas situaciones se obtuvieron mejoras. En el caso de EW para ANN no aparecen cambios significativos en la efectividad del clasificador cuando la base de datos es equilibrada y existe solapamiento entre clases. Así, al existir igual cantidad de elementos tanto de una clase como de la otra, se eliminan proporciones semejantes de prototipos. En consecuencia aunque se reduce la zona de incertidumbre no se produce un incremento de PC en la red para esta experimentación.

A continuación se presentan las bases de datos sintéticas propuestas en la metodología en situación de desbalance entre clases.

Los resultados obtenidos por las ANNs del tipo MLP, RBF y RBF+VF en la fase de clasificación de los conjuntos de datos desequilibrados y con solapamiento son presentados en las Tablas 5.2 y 5.3 . En los tres modelos neuronales se presenta un comportamiento muy similar en cuanto a valores de PC y *g-mean*.

En particular, se observa que la EW es la técnica que muestra menor efectividad en cuestiones de *g-mean*, incluso por debajo de los valores originales. Esto se debe a la considerable eliminación de elementos de la clase minoritaria producida por esta técnica (véase la Tabla 5.4).

A diferencia del caso con clases balanceadas donde el número de elementos de ambas clases está en equilibrio en la frontera de decisión, nos encontramos que existe una menor cantidad de prototipos de la clase minoritaria en esta región, y la reducción de elementos de esta clase se ve más afectada. El efecto que se genera es un desplazamiento de la frontera de decisión hacia la clase minoritaria al realizarse el proceso de edición.

En el caso de utilizar las técnicas EW^- y EWP^- se produce el efecto contrario, ya que como estas técnicas sólo eliminan muestras de la clase mayoritaria manteniéndose la zona de influencia de la clase minoritaria dentro de la región de solapamiento. Así, se produce un desplazamiento de la frontera de decisión hacia la clase mayoritaria después de realizarse el proceso de edición. En este sentido, se observó que al utilizar la técnica EWP se presenta esta misma tendencia en la efectividad del clasificador, aunque su impacto es menor.

[1]Los valores entre paréntesis hacen referencia a la desviación estándar. En el resto del capítulo se seguirá la misma nomenclatura.

Tabla 5.1: Efectividad del clasificador sobre las bases de datos sintéticas balanceadas (250-250).

MLP

PC	A0	A20	A40	A60	A80	A100
Original	99.54(0.30)	83.47(1.31)	71.53(2.06)	63.32(1.86)	54.53(1.50)	49.23(1.86)
EW	99.18(0.68)	83.65(1.10)	71.50(1.87)	63.09(2.16)	54.28(1.92)	49.45(1.50)
g-mean	A0	A20	A40	A60	A80	A100
Original	99.55(0.30)	83.42(1.35)	71.48(2.05)	63.12(2.06)	54.43(1.53)	49.12(1.79)
EW	99.19(0.69)	83.58(1.09)	71.34(1.78)	62.57(2.44)	52.29(2.35)	48.59(1.60)

RBF

PC	A0	A20	A40	A60	A80	A100
Original	99.05(0.53)	83.44(1.35)	71.55(1.97)	62.99(1.99)	53.84(2.21)	49.87(1.95)
EW	98.99(0.63)	83.62(1.23)	71.18(1.96)	62.58(2.28)	53.61(2.20)	49.53(1.81)
g-mean	A0	A20	A40	A60	A80	A100
Original	99.05(0.53)	83.39(1.38)	71.42(1.97)	62.36(2.58)	52.20(2.82)	48.88(2.95)
EW	99.00(0.63)	83.56(1.24)	71.03(1.93)	61.82(2.80)	50.96(3.69)	48.47(2.17)

RBF+VF

PC	A0	A20	A40	A60	A80	A100
Original	98.92(0.51)	83.47(1.37)	71.52(2.01)	62.81(2.01)	54.15(2.39)	49.38(1.90)
EW	98.71(0.66)	83.56(1.41)	71.32(2.02)	62.81(2.05)	53.91(2.45)	49.41(1.79)
g-mean	A0	A20	A40	A60	A80	A100
Original	98.93(0.51)	83.42(1.40)	71.45(2.02)	62.35(2.40)	53.25(2.67)	48.76(2.03)
EW	98.72(0.66)	83.50(1.42)	71.17(1.97)	62.35(2.31)	52.18(3.25)	48.71(1.90)

3-NN

PC	A0	A20	A40	A60	A80	A100
Original	98.60(0.52)	83.00(1.71)	71.00(2.35)	60.38(1.18)	54.02(2.14)	50.12(2.90)
EW	98.38(0.85)	83.06(1.72)	71.24(2.01)	60.72(1.54)	54.02(2.95)	49.36(3.27)
g-mean	A0	A20	A40	A60	A80	A100
Original	98.71(0.52)	83.14(1.71)	71.13(2.39)	60.49(1.20)	54.08(2.15)	50.17(2.91)
EW	98.50(0.82)	83.25(1.78)	71.40(2.05)	60.85(1.52)	54.13(3.00)	49.39(3.34)

En la Tabla 5.4 se presentan el número de muestras por clase, antes y después de haberse aplicado las técnicas de edición evaluadas en este trabajo. El primer número corresponde a la clase minoritaria y el segundo a la mayoritaria. Estos resultados pueden relacionarse directamente con la efectividad del clasificador tal y como se discute a continuación.

En las Tablas 5.2 y 5.3 se observan que al aplicar la EW^{-} se obtienen resultados semejantes a los de la ME original en el caso de las ANNs, además de pequeñas mejoras en la medida de *g-mean*. Sin embargo, la aplicación de esta técnica de edición sobre la regla 3-NN produce un empeoramiento del clasificador. Mientras que las ANNs son clasificadores de naturaleza global, es decir, tiene en cuenta la distribución de los datos de las diferentes clases y de forma conexionista actúan en

Tabla 5.2: Precisión en la fase de clasificación obtenida con bases de datos sintéticas no balanceadas sobre MLP y RBF.

MLP

PC	A0	A20	A40	A60	A80	A100
Original	98.53(0.61)	92.40(1.28)	87.81(1.05)	82.41(1.25)	80.00(0.00)	80.00(0.00)
EW	98.23(0.61)	91.49(1.16)	87.07(0.99)	81.40(1.45)	80.00(0.00)	80.00(0.00)
EWP	98.62(0.68)	91.26(2.16)	86.81(2.04)	82.97(1.99)	78.59(1.94)	80.00(0.00)
EW^-	98.54(0.59)	92.02(1.42)	87.25(1.66)	82.97(1.04)	80.20(0.57)	80.00(0.00)
EWP^-	98.69(0.65)	87.25(2.41)	80.23(3.27)	73.80(4.20)	70.83(5.78)	78.37(3.96)
g-mean	A0	A20	A40	A60	A80	A100
Original	96.24(1.58)	78.89(3.96)	63.44(3.80)	32.70(12.05)	0.00(0.00)	0.00(0.00)
EW	95.46(1.60)	75.69(3.82)	59.37(4.22)	19.77(17.62)	0.00(0.00)	0.00(0.00)
EWP	96.49(1.78)	79.75(3.80)	65.65(3.72)	48.44(5.95)	21.56(17.21)	0.00(0.00)
EW^-	96.28(1.52)	79.52(4.17)	65.15(3.97)	40.53(8.28)	6.49(13.30)	0.00(0.00)
EWP^-	96.66(1.68)	82.13(2.95)	68.58(3.22)	57.74(3.50)	43.60(12.05)	6.31(13.79)

RBF

PC	A0	A20	A40	A60	A80	A100
Original	97.95(0.62)	91.78(1.33)	86.48(1.68)	81.28(1.09)	80.03(0.12)	79.99(0.14)
EW	97.58(0.66)	90.84(1.17)	85.46(1.51)	80.66(0.82)	80.04(0.15)	79.98(0.12)
EWP	98.09(0.59)	90.38(1.82)	84.72(2.22)	80.15(1.70)	78.55(2.05)	79.04(1.79)
EW^-	97.97(0.61)	91.08(1.62)	85.89(2.00)	81.34(1.11)	79.61(0.86)	79.41(1.32)
EWP^-	98.15(0.62)	86.35(2.59)	78.75(3.58)	73.33(4.20)	69.84(5.61)	70.48(5.53)
g-mean	A0	A20	A40	A60	A80	A100
Original	94.72(1.66)	78.42(4.05)	58.93(8.80)	22.03(15.10)	2.46(6.83)	0.00(0.00)
EW	93.74(1.78)	73.70(3.85)	51.94(7.64)	12.52(13.96)	1.28(5.14)	0.00(0.00)
EWP	95.13(1.53)	79.77(3.94)	63.33(5.60)	36.85(12.63)	21.55(16.42)	6.47(10.83)
EW^-	94.79(1.61)	79.14(4.48)	61.82(6.95)	29.63(14.56)	10.68(13.82)	4.14(7.70)
EWP^-	95.30(1.62)	81.86(2.90)	68.58(3.73)	54.40(6.03)	43.57(11.96)	34.75(11.47)

la toma de decisiones en la clasificación, en la regla 3-NN esta decisión es realizada de forma local entorno al prototipo a clasificar.

Así, al eliminar puntos sólo de la clase mayoritaria se produce un desplazamiento de la frontera hacia esta clase manteniéndose la influencia de la clase minoritaria en la zona de confusión. Por ello la regla 3-NN al clasificar en torno a la vecindad de la muestra produce un aumento en el número de muestras de la clase mayoritaria mal clasificadas. Este efecto se acentúa cuando se aplica la EWP^-.

En la Tabla 5.4 se observa como la EW^- elimina un 15% de muestras de la clase mayoritaria, mientras que para la EWP^- se llega a eliminar hasta un 50% de las muestras. Posiblemente, lo más conveniente no es alcanzar un equilibrio entre las clases sino minimizar la diferencia entre ellas. Desde este punto de vista, la estrategia EWP^- sería la más efectiva en eliminar muestras de la clase mayoritaria minimizando la confusión de la clase minoritaria y reduciendo el desbalance.

Tabla 5.3: Precisión en la fase de clasificación obtenida con bases de datos sintéticas no balanceadas sobre RBF+VF y 3-NN.

RBF+VF

PC	A0	A20	A40	A60	A80	A100
Original	98.06(0.53)	91.74(1.41)	87.06(1.33)	82.92(0.95)	80.26(0.46)	79.99(0.07)
EW	97.66(0.71)	90.90(1.12)	86.18(1.27)	81.64(0.99)	80.11(0.35)	79.99(0.06)
EWP	98.13(0.58)	90.53(2.11)	85.78(2.17)	81.39(2.02)	77.83(2.48)	78.72(1.80)
EW$^-$	98.04(0.50)	91.16(1.76)	86.40(1.66)	82.36(1.27)	79.67(1.12)	79.47(1.02)
EWP$^-$	98.22(0.52)	86.53(2.43)	79.58(3.72)	73.28(4.42)	69.10(6.03)	68.69(5.03)
g-mean	A0	A20	A40	A60	A80	A100
Original	95.04(1.39)	78.10(4.07)	60.52(5.35)	39.10(6.49)	8.66(10.99)	0.30(1.71)
EW	93.96(1.91)	73.96(3.55)	55.61(6.15)	26.15(12.43)	3.32(8.49)	0.00(0.00)
EWP	95.27(1.50)	79.89(4.33)	64.33(4.26)	46.11(6.01)	34.02(10.15)	9.51(11.79)
EW$^-$	94.98(1.31)	79.40(4.50)	62.76(5.19)	42.63(7.55)	21.16(14.17)	4.80(8.33)
EWP$^-$	95.52(1.32)	81.68(3.23)	68.78(3.38)	58.80(3.86)	49.37(5.52)	37.71(9.02)

3-NN

PC	A0	A20	A40	A60	A80	A100
Original	98.72(0.53)	90.20(1.73)	83.66(1.67)	79.32(1.18)	75.40(0.69)	73.62(2.34)
EW	98.50(0.61)	90.82(1.26)	86.04(1.56)	82.22(0.76)	79.06(1.11)	78.92(0.80)
EWP	98.42(0.81)	87.80(2.61)	80.36(1.53)	73.40(2.54)	68.66(1.93)	65.54(4.21)
EW$^-$	98.66(0.56)	88.48(1.82)	80.60(2.24)	74.72(2.27)	70.30(1.75)	67.68(3.69)
EWP$^-$	98.44(0.85)	83.34(2.87)	73.16(2.52)	63.00(3.45)	56.58(4.21)	53.02(3.12)
g-mean	A0	A20	A40	A60	A80	A100
Original	97.11(1.43)	78.97(4.25)	63.35(3.99)	50.54(3.58)	39.54(5.16)	28.62(4.73)
EW	96.34(1.57)	76.61(3.60)	60.71(4.74)	42.04(4.75)	23.19(8.40)	14.11(6.28)
EWP	96.92(1.69)	80.18(3.68)	68.46(2.87)	58.43(2.88)	50.35(3.25)	40.69(3.03)
EW$^-$	97.07(1.43)	80.11(4.25)	66.54(4.10)	55.06(4.13)	46.76(4.91)	37.94(4.64)
EWP$^-$	97.29(1.63)	82.45(2.75)	70.48(2.57)	61.15(3.43)	55.28(3.34)	49.59(2.83)

Sin embargo, como puede verse en la Tabla 5.5 se reduce la PC de la clase mayoritaria lo que confirma que estamos desplazando la frontera y perjudicando la precisión de esta clase. La tendencia en la EWP$^-$ es aumentar la precisión de la clase minoritaria (lo que se ve reflejado en los incrementos de los valores de *g-mean*) mientras que se produce una disminución en la PC global.

En términos generales, y a partir de los resultados discutidos hasta este momento se puede afirmar lo siguiente:

1. La EW al eliminar tanto muestras de la clase mayoritaria como de la minoritaria desplaza la frontera de decisión hacia la clase minoritaria, y se pierde información relevante de esta última, lo que se traduce en una disminución de la PC de esta clase.

2. La EWP evita, debido a la ponderación, que se descarten las mismas proporciones de muestras de la clase minoritaria que con la EW.

Tabla 5.4: Número de muestras en la ME antes y después aplicar las técnicas de corrección de datos. El primer número corresponde a la clase minoritaria y el segundo a la mayoritaria.

PC	A0	A20	A40	A60	A80	A100
Original	100/400	100/400	100/400	100/400	100/400	100/400
EW	94/399	67/386	45/374	26/366	14/361	9/360
EWP	98/393	80/330	71/290	61/243	54/223	47/204
EW^-	100/399	100/380	100/365	100/351	100/345	100/343
EWP^-	100/392	100/317	100/271	100/221	100/198	100/179

Tabla 5.5: Precisión por clase obtenida con la base de datos A40. PC^- hace referencia a la precisión de la clase mayoritaria y PC^+ a la de la minoritaria.

	MLP		Red RBF		Red RBF+VF	
	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+
Original	99.63	40.53	99.15	35.80	99.56	37.07
EW	99.97	35.43	99.93	27.57	99.90	31.33
EWP	97.43	44.33	95.30	42.40	96.47	43.05
EW^-	98.22	43.34	97.44	39.73	97.86	40.52
EWP^-	86.70	54.35	84.44	56.00	85.61	55.43

3. La EW^- y la EWP^- al no eliminar prototipos de la minoritaria ayudan a incrementar la PC de esta clase manteniendo su zona de influencia. De esta forma, cuanto más crece el nivel de solapamiento más se perjudica a la clase mayoritaria desplazándose la frontera de decisión hacia esta clase.

Para confirmar estas conclusiones, en la siguiente sección se evalúa las técnicas propuestas en bases de datos reales.

5.5 Bases de datos reales

En esta sección se presentan algunos resultados obtenidos al experimentar con bases de datos reales de dos y múltiples clases. Se utilizaron los conjuntos de datos German, Diabetes, Ionosphere, Phoneme, Ecoli6 y Cayo. Obsérvese que no se incluyen las bases de datos B2Cls y V2Cls. Esto es debido a que por un lado V2Cls no presenta solapamiento entre clases, y por el otro B2Cls se encuentra completamente solapada.

5.5.1 Problemas de dos clases

En la Tabla 5.6 se presenta la PC por clase de las bases de datos de dos clases. PC^- hace referencia a la PC de la clase mayoritaria mientras que PC^+ a la de la minoritaria. El objetivo que se persigue con esta tabla es mostrar las modificaciones que se producen en PC^- y PC^+ cuando se aplica las diferentes técnicas de edición para las cuatro bases de datos reales.

Tabla 5.6: Precisión por clase obtenida por la ANN. PC^- hace referencia a la precisión de la clase mayoritaria y PC^+ a la de la minoritaria.

	Original		EW		EWP		EW^-		EWP^-	
MLP	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+
Diabetes	65.60	81.72	74.80	75.90	63.66	84.89	48.32	90.45	48.14	91.90
German	82.19	55.43	90.70	34.40	86.03	42.23	73.19	66.63	72.00	68.97
Ionosphere	97.56	80.95	98.53	57.30	98.84	63.17	96.93	80.24	96.84	80.63
Phoneme	85.49	62.62	84.90	62.11	83.14	67.01	83.39	68.82	82.13	70.36
RBF	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+
Diabetes	60.08	81.64	68.10	77.95	55.70	88.62	34.92	96.79	33.60	96.60
German	92.77	28.03	99.16	4.97	97.66	10.73	83.27	49.80	81.21	52.97
Ionosphere	84.49	92.94	93.20	84.76	90.80	87.86	84.22	93.41	84.31	93.33
Phoneme	88.10	61.38	89.56	56.18	85.66	68.85	85.03	69.81	82.35	75.81
RBF+VF	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+
Diabetes	64.54	82.87	75.20	74.18	64.94	84.07	47.90	89.18	44.98	91.46
German	85.76	53.03	92.77	27.97	88.63	38.43	75.59	67.47	74.11	68.03
Ionosphere	95.51	75.87	97.11	61.27	96.93	63.49	93.87	76.98	93.29	75.16
Phoneme	87.79	63.11	88.89	58.98	85.66	68.76	84.39	71.01	82.61	75.62
3-NN	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+	PC^-	PC^+
Diabetes	77.40	51.87	83.80	48.88	79.60	57.46	64.00	71.64	61.80	72.76
German	82.43	35.00	88.43	26.67	86.71	30.00	70.43	54.00	68.14	56.33
Ionosphere	98.22	59.52	97.78	57.14	97.78	57.14	97.33	66.67	97.33	65.87
Phoneme	93.43	78.25	93.22	73.39	90.70	80.20	89.89	83.10	87.64	85.25

En la Tabla 5.7 se muestra en número de muestras por clase antes y después de aplicar las técnicas de corrección de datos. El primer número corresponde a la clase minoritaria y el segundo a la mayoritaria. Además, se muestra el porcentaje de reducción que se genera para la base de datos después de aplicar la técnica de edición respecto del conjunto original. Se puede observar que la cantidad de muestras de ambas clases en las bases de datos varía significativamente y que el número de muestras eliminadas es diferente dependiendo de la base de datos.

En la Tabla 5.8 se presentan los valores de PC global y *g-mean* en bases de datos reales de dos clases, respecto a las diferentes técnicas de edición propuestas. Se observa como el comportamiento en valores de PC global y *g-mean* es bastante diferente de unas bases de datos a otras.

Así, en el caso de la base de datos Phoneme se puede ver como el valor de *g-*

Tabla 5.7: Número de muestras y porcentaje de reducción en la ME antes y después aplicar las técnicas de corrección de datos.

	German	%	Diabetes	%	Ionosphere	%	Phoneme	%
Original	270/630	100	241/450	100	113/202	100	1427/3436	100
EW	99/517	68	128/351	69	68/198	84	1052/3201	87
EWP	111/488	67	149/324	71	74/198	98	1198/3035	87
EW^-	270/477	83	241/311	80	113/196	98	1427/3175	95
EWP^-	270/442	79	241/287	76	113/197	98	1427/3019	91

mean tiene una variación importante dependiendo de la técnica de edición aplicada, obteniéndose los valores más altos cuando se aplica la EWP^- y EW^-. En estos dos casos se produce un aumento significativo del PC^+ sin que esto suponga perdidas excesivamente graves en el PC^- (ver Tabla 5.6).

Por otro lado, la EWP y la EW equilibran levemente la PC a favor de la clase mayoritaria y minoritaria respectivamente. En su conjunto se puede decir que en esta base de datos, las técnicas de edición propuestas (EWP y EW) son capaces de lograr un relativo balance entre clases favoreciendo la mejora del PC^- sin que esto suponga una perdida significativa del PC^+, y por tanto se mantiene la PC global para la ANN.

La explicación de este comportamiento se debe a que esta base de datos esta bien representada (4863 muestras en 5 dimensiones) siendo además su nivel de solapamiento no excesivamente alto. Así, en la Tabla 5.7 el porcentaje de reducción de muestras que se produce respecto del conjunto original al aplicar las técnicas de edición es relativamente bajo, lo que confirma una alta separabilidad entre clases.

En el caso de las bases de datos German e Ionosphere, las técnicas de edición propuestas producen peores resultados con respecto al conjunto original en PC global. En el caso de German, las técnicas de EW^- y EWP^- reducen de forma significativa la PC^-, aunque la clase minoritaria es beneficiada. Visto en su conjunto, el resultado no es satisfactorio, y la razón vendría dada debido a que las técnicas de edición en esta base de datos desplazan la frontera de decisión hacia la clase mayoritaria (ver Tabla 5.7). Considérese además que esta base de datos tiene una representatividad relativamente baja con 1000 muestras en 24 dimensiones y un nivel de solapamiento alto.

En el caso de Ionosphere sólo la EW^- consigue mantener un PC global al mismo nivel que el conjunto original. Sin embargo, el efecto que tiene esta técnica tampoco es satisfactoria pues no consigue aumentar el *g-mean* con respecto al conjunto original. Al igual que en German estamos tratando una base de datos muy poco representada (con solo 351 muestras en 34 dimensiones).

En el caso de la base de datos Diabetes, el comportamiento que se obtiene a nivel de PC global al aplicar la técnica de EW es similar al que se obtiene cuando se aplica el clasificador 3-NN, en el sentido de que esta técnica de edición mejora la PC global respecto al conjunto original. Es interesante notar que en esta base de datos a diferencia de las otras bases de datos la clasificación mediante ANNs sobre el conjunto original se comporta de manera distinta.

La clase minoritaria presenta un mejor porcentaje de aciertos respecto de la clase mayoritaria, mientras que en el clasificador 3-NN no se muestra esta tendencia (ver Tabla 5.6). No se tiene claro la causa de este comportamiento. El resto de técnicas mejoran el PC^+ y empeoran el PC^- desplazando la frontera de decisión hacia la clase mayoritaria disminuyendo la PC global.

En general, se puede decir que la técnica EW elimina más muestras de la clase minoritaria de forma que la PC^- es incrementada y la PC^+ es disminuida. De esta forma, se acentúa el desbalance de las clases. El caso más notorio es el de la base de datos German. En el caso de la EWP en algunas situaciones es la clase minoritaria la que se ve favorecida, mientras que en otros casos es la clase mayoritaria al aplicar esta técnica de edición (ver Tabla 5.6).

Por otro lado, la EW^- al eliminar exclusivamente elementos de la clase mayoritaria incrementa la PC^+ mientras reduce la PC^-. El caso más extremo lo ilustra diabetes donde la PC^- se reduce en casi un 17% con el MLP y la red RBF+VF, y en aproximadamente 25% con la red RBF. Sin embargo, con german y phoneme la situación no es tan dramática.

La EWP^- muestra la misma tendencia que la EW^- pero de una forma más acentuada.

Son evidentes las mejoras que se obtienen en la PC^+ al eliminar elementos de la clase mayoritaria de forma dirigida[2] y también es indiscutible la perdida de efectividad del clasificador sobre la clase más representada. No obstante, cuando lo que se desea es dar prioridad a la clase menos representada, las estrategias EW^- y EWP^- resultan atractivas.

Así, se podría esperar que dado que las técnicas de edición disminuyen la región de incertidumbre entre las clases, aumentará la precisión de clasificación. Sin embargo, como en problemas desequilibrados se produce un desplazamiento de la frontera de decisión en un sentido u otro dependiendo de la técnica de edición elegida, en algunas situaciones se produce un efecto pernicioso que disminuye la PC global.

[2] Al utilizar mecanismos heurísticos para la identificaciones de elementos en la zona de confusión.

Tabla 5.8: Valores globales de PC y *g-mean* de las bases reales de dos clases.

	MLP		RBF		RBF+VF		3-NN	
Opción 0	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Diabetes	71.23(7.02)	72.19(6.60)	67.61(7.99)	68.36(7.77)	70.94(7.10)	72.23(6.46)	68.49(3.71)	63.25(4.81)
German	74.16(3.96)	67.09(6.03)	73.35(2.95)	49.41(12.06)	75.94(3.09)	66.93(6.29)	68.20(3.61)	53.49(5.42)
Ionosphere	91.58(6.13)	88.31(9.17)	87.51(5.74)	88.39(5.33)	88.44(5.80)	84.77(7.99)	84.32(6.52)	75.65(12.00)
Phoneme	78.78(1.72)	73.14(1.94)	80.26(1.67)	73.43(3.15)	80.55(1.70)	74.36(2.83)	88.97(1.66)	85.54(2.52)
EW	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Diabetes	75.19(5.13)	74.81(5.01)	71.55(6.88)	71.79(7.11)	74.85(5.29)	74.10(5.46)	71.62(3.94)	63.78(6.24)
German	73.81(3.54)	54.71(10.14)	70.90(1.28)	16.81(14.49)	73.33(3.26)	48.33(14.69)	69.90(3.78)	47.80(9.54)
Ionosphere	83.71(5.45)	74.44(10.00)	90.15(7.02)	87.97(12.32)	84.22(6.54)	76.34(11.01)	83.18(6.11)	74.16(10.77)
Phoneme	78.21(1.66)	72.56(2.17)	79.76(1.69)	70.79(3.55)	80.11(1.59)	72.34(2.74)	87.40(1.55)	82.74(2.54)
EWP	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Diabetes	71.08(6.66)	72.74(6.10)	67.19(6.52)	69.46(6.15)	71.62(5.97)	73.29(5.49)	71.88(5.30)	67.44(7.26)
German	72.89(4.08)	59.34(8.90)	71.58(1.58)	28.48(15.15)	73.57(4.06)	57.16(10.37)	69.70(2.83)	50.44(8.04)
Ionosphere	86.02(5.67)	78.42(9.92)	89.72(5.98)	89.04(7.76)	84.91(5.86)	77.80(9.64)	83.18(6.11)	74.16(10.77)
Phoneme	78.40(1.51)	74.62(1.88)	80.72(1.92)	76.70(2.98)	80.69(2.06)	76.72(2.66)	87.62(2.12)	85.33(3.07)
EW^-	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Diabetes	63.02(5.72)	65.28(5.79)	56.52(6.53)	57.22(8.42)	62.31(6.45)	64.52(6.59)	66.67(5.18)	67.66(4.68)
German	71.22(4.71)	69.41(4.40)	73.23(4.19)	64.03(6.27)	73.15(4.40)	71.14(4.92)	65.50(4.45)	61.50(5.43)
Ionosphere	90.92(6.61)	87.61(9.75)	87.51(5.38)	88.53(4.93)	87.78(6.53)	84.67(8.50)	86.31(6.31)	80.02(10.43)
Phoneme	79.11(1.95)	75.73(2.26)	80.57(1.68)	76.95(3.02)	80.46(2.03)	77.33(2.95)	87.90(1.99)	86.48(2.50)
EWP^-	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Diabetes	63.42(6.24)	65.86(6.61)	55.60(6.72)	55.90(8.48)	61.20(6.66)	63.40(7.25)	65.63(5.13)	66.98(4.62)
German	71.09(4.74)	70.12(4.57)	72.74(3.53)	65.21(4.95)	72.29(4.39)	70.58(6.50)	64.60(4.55)	61.74(5.72)
Ionosphere	91.00(6.89)	87.69(10.41)	87.54(5.16)	88.50(4.82)	86.75(6.68)	83.32(9.04)	86.03(6.11)	79.60(10.06)
Phoneme	78.67(1.74)	75.99(2.18)	80.43(1.86)	78.91(2.60)	80.56(2.06)	78.99(2.55)	86.93(2.15)	86.49(2.56)

5.5.2 Problemas de múltiples clases

En esta sección se evaluará la utilidad de la estrategia EWP$^-$ para reducir el solapamiento que existe entre pares de clases, manteniendo el área de influencia de las clases minoritarias en la zona de solapamiento e incrementando la efectividad del clasificador sobre estas clases. La EW$^-$ actúa igual que la EWP$^-$ pero el incremento del PC sobre las clases minoritarias no es tan importante y no se incluyó en la experimentación.

Un problema que surge en el aprendizaje de una ANN con bases de datos desbalanceadas es la lentitud en la convergencia de las clases menos representadas. Por ello en este sección se combinaran ambos enfoques: reducir la confusión de la clases minoritarias en la frontera de decisión e incluir alguna función de coste que nos permita acelerar la convergencia en el entrenamiento de la ANN.

En esta experimentación se ha elegido la Opción 3 como función de coste ya que aunque esta función favorece el aprendizaje de las clases minoritarias al principio del entrenamiento, este beneficio disminuye a medida que avanza el proceso de aprendizaje. En este sentido los efectos de la estrategia EWP$^-$ son menos influenciados por la función de coste elegida. Se han realizado otros experimentos con las opciones 1 y 2 obteniendo niveles de PC global similares cuando se incluye la EWP$^-$ o no se incluye, aunque la convergencia de la red se acelera con las opciones 1 y 2, y el valor de MSE (Mean Square Error, error cuadrático medio) que se obtiene es menor.

Ecoli6

Una de las principales características que se ha evidenciado en esta base de datos es el fuerte solapamiento que existe entre las clases 3 y 5. En esta sección se utiliza la estrategia EWP$^-$ para reducir la confusión entre estas clases.

El objetivo que se busca es reducir el área de solapamiento de la clase 5 o clase minoritaria incrementando la precisión de esta clase.

En la Tabla 5.9 se muestran los resultados por clase obtenidos al aplicar la estrategia EWP$^-$ sobre las clases 3 y 5. Se observa la misma tendencia que para problemas de dos clases: la PC de la minoritaria se incrementa mientras que la PC de la mayoritaria se reduce. Nótese que la confusión de la clase minoritaria (clase 5) disminuyó[3] lo que se tradujo en incrementos de PC para esta clase. Este comportamiento se observa en los tres modelos neuronales.

En la Tabla 5.10 se presentan los resultados de clasificación obtenidos con la ANN sobre Ecoli6 editada con EWP$^-$ y aplicando la función de coste Opción 3. Obsérvese que en las ANNs del tipo MLP y RBF+VF incrementan la PC de la clase 5.

[3]En relación a los resultados obtenidos con la ME sin editar, Tabla 3.8 del capítulo 3.

Tabla 5.9: Resultados de clasificación sobre Ecoli6 con la Opción 0 editando con EWP^{-} sobre las clases 3 y 5.

	Clase	F1	Ratio	PC	% confusión (> 10 %)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	97.14	
MLP	cls 3	10.66	0.23	76.18	cls 5 (18.05)
	cls 4	10.96	0.16	82.31	
	cls 5	10.33	0.11	67.86	cls 3 (26.62)
	cls 6	9.33	0.06	89.20	
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	95.17	
Red RBF	cls 3	10.66	0.23	78.70	cls 5 (17.70)
	cls 4	10.96	0.16	84.87	
	cls 5	10.33	0.11	70.21	cls 3 (27.05)
	cls 6	9.33	0.06	86.17	
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	94.68	
RBF+VF	cls 3	10.66	0.23	76.69	cls 5 (18.22)
	cls 4	10.96	0.16	84.10	
	cls 5	10.33	0.11	69.88	cls 3 (26.09)
	cls 6	9.33	0.06	86.96	

Tabla 5.10: Resultados obtenidos en la fase de clasificación con la base de datos Ecoli6 y la Opción 3. Observe que la clase 5 ha sido editada en relación a la clase 3.

	Clase	F1	Ratio	PC	% confusión (> 10 %)
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	92.71	
MLP	cls 3	10.66	0.23	75.67	cls 5 (18.67)
	cls 4	10.96	0.16	81.42	
	cls 5	10.33	0.11	70.19	cls 3 (22.36)
	cls 6	10.93	0.06	87.50	
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	91.98	
RBF	cls 3	10.66	0.23	77.03	cls 5 (19.46)
	cls 4	10.96	0.16	85.74	
	cls 5	10.33	0.11	70.83	cls 3 (27.68)
	cls 6	10.93	0.06	86.98	
	cls 1	52.65	0.02	100.00	
	cls 2	10.96	0.43	91.21	
RBF+VF	cls 3	10.66	0.23	73.38	cls 5 (20.97)
	cls 4	10.96	0.16	85.83	
	cls 5	10.33	0.11	72.95	cls 3 (21.88)
	cls 6	10.93	0.06	85.64	cls 4 (10.64)

En la Tabla 5.11 se muestran los resultados de clasificación global sobre la base

de datos Ecoli6. La tendencia en esta base de datos es la de reducir tanto los valores de PC y *g-mean* en la combinación EWP$^-$ y Opción 3. Esto es debido a que los datos tienen una baja representatividad y la combinación EWP$^-$ y Opción 3 aumenta la perdida de representatividad de los datos en las diferentes clases.

Tabla 5.11: Efectividad del clasificador sobre la base de datos Ecoli6.

	PC			*g-mean*		
	Op. 0	Op. 0	Op. 3	Op. 0	Op. 0	Op. 3
ANN	Original	EWP$^-$	EWP$^-$	Original	EWP$^-$	EWP$^-$
MLP	86.56(2.78)	86.45(2.43)	84.41(4.42)	83.76(5.57)	83.93(4.58)	83.24(4.41)
RBF	85.87(3.36)	86.64(3.62)	85.13(3.84)	83.50(5.29)	84.43(5.01)	84.18(4.08)
RBF+VF	85.34(3.69)	85.86(3.68)	84.10(4.50)	82.59(5.62)	84.04(4.43)	83.42(4.81)

Cayo

En esta base de datos se decidió aplicar EWP$^-$ sobre las clases consideradas minoritarias (clases 2, 4, 5, 6 y 7)[4] con respecto al resto de clases que se consideran mayoritarias.

En la Tabla 5.12 se presentan los resultados de clasificar la base de datos Cayo editada con EWP$^-$ en las ANNs del tipo MLP, RBF y RBF+VF. Si se comparan estos resultados con los presentados en las Tablas 3.15, 3.19 y 3.20 con Opción 0 del capítulo 3 se observa que mejoran los resultados de PC en las clases minoritarias cuando la ME es editada, sobre todo en las ANNs del tipo RBF y RBF+VF. Además, se puede observar una ligera mejora de la *g-mean* sobre esta base de datos (véase Tabla 5.14).

En la Tabla 5.13 se muestran los resultados de clasificar las ANNs mediante la combinación de la EWP$^-$ y la Opción 3.

En general, analizando la Tabla 5.13 se puede observar que las clases minoritarias incrementan su PC (clases 2, 4, 5, 6 y 7), y que algunas de las clases mayoritarias disminuye su PC (véase las clases 1, 3 y 10). En el caso de las clases 8 y 9 los resultados en PC se mantienen respecto a cuando no se aplica la combinación la EWP$^-$ y la Opción 3. Por último, la clase 11 aún siendo una clase mayoritaria mejora su PC seguramente porque se acelera su convergencia al aplicar la Opción 3

En particular, se puede afirmar que las clases minoritarias mejoran o empatan su PC en las ANNs del tipo MLP y RBF+VF, aunque esto no sucede en todas las clases para la red RBF (ver clases 2 y 4). Así, en las clases minoritarias disminuye el nivel de confusión con respecto a las clases mayoritarias. Por ejemplo, en la clase

[4]De la misma forma que se hizo en los capítulos 3 y 4.

Tabla 5.12: Resultados obtenidos en la fase de clasificación de la base de datos Cayo editada con la estrategia EWP$^-$. Los resultados corresponden a la Opción 0.

	Clase	F1	Ratio	PC	% confusión (> 10 %)
	cls 1	32.57	0.14	89.20	
	cls 2	5.41	0.05	50.14	cls 3 (48.63)
	cls 3	4.57	0.10	94.75	
	cls 4	4.53	0.05	81.01	
	cls 5	10.75	0.02	31.26	cls 1 (38.36) cls 3 (18.74)
MLP	cls 6	6.18	0.06	58.35	cls 7 (32.52)
	cls 7	5.27	0.05	96.48	
	cls 8	5.46	0.12	97.94	
	cls 9	8.05	0.13	87.56	cls 10 (12.44)
	cls 10	8.31	0.14	74.32	cls 11 (24.75)
	cls 11	8.37	0.13	95.02	
	cls 1	32.57	0.14	88.67	
	cls 2	5.41	0.05	43.09	cls 3 (48.63)
	cls 3	4.57	0.10	91.64	
	cls 4	4.53	0.05	57.14	cls 3 (23.43) cls 8 (14.74)
	cls 5	10.75	0.02	36.50	cls 1 (32.09) cls 8 (22.18)
RBF	cls 6	6.18	0.06	49.21	cls 7 (29.00) cls 10 (10.82)
	cls 7	5.27	0.05	88.51	
	cls 8	5.46	0.12	98.29	
	cls 9	8.05	0.13	87.47	cls 10 (12.44)
	cls 10	8.31	0.14	74.48	cls 11 (18.16)
	cls 11	8.37	0.13	69.74	cls 9 (19.38) cls 10 (10.88)
	cls 1	32.57	0.14	87.45	
	cls 2	5.41	0.05	45.89	cls 3 (48.63)
	cls 3	4.57	0.10	94.11	
	cls 4	4.53	0.05	73.20	cls 7 (10.38)
	cls 5	10.75	0.02	30.95	cls 1 (24.03) cls 3 (16.23) cls 8 (26.84)
RBF+VF	cls 6	6.18	0.06	57.45	cls 7 (32.30)
	cls 7	5.27	0.05	94.30	
	cls 8	5.46	0.12	97.59	
	cls 9	8.05	0.13	87.56	cls 10 (12.44)
	cls 10	8.31	0.14	74.69	cls 11 (12.23)
	cls 11	8.37	0.13	60.40	cls 9 (25.98) cls 10 (13.62)

5 con la MLP sin editar presenta un nivel de confusión importante con las clases 1 y 3, disminuyendo esta confusión significativamente cuando se aplica la combinación de EWP$^-$ y la Opción 3 (ver Tabla 5.13).

En otras situaciones cuando lo que están muy confundidas son dos clases minoritarias (véase las clases 6 y 7), o dos clases mayoritarias (véase las clases 10 y 11) la técnica propuesta no va a generar ningún efecto. Nótese que la clase 7 se confunde con la clase 6 en torno a un 31% cuando se supone que tienen el mismo tamaño, lo cual hace sugerir que la clase 7 es más densa que la clase 6.

En la Tabla 5.14 se resumen los resultados globales obtenidos tanto en valores

Tabla 5.13: Resultados obtenidos en la fase de clasificación de la base de datos Cayo editada con la estrategia EWP$^-$. Los resultados corresponden a la Opción 3.

	Clase	F1	Ratio	PC	% confusión (> 10 %)
	cls 1	4.648	0.14	87.24	
	cls 2	0.831	0.05	52.12	cls 3 (47.88)
	cls 3	0.696	0.10	92.91	
	cls 4	0.563	0.05	94.35	
	cls 5	0.955	0.02	81.36	
MLP	cls 6	0.228	0.06	60.30	cls 7 (31.06)
	cls 7	0.543	0.05	95.61	
	cls 8	0.139	0.12	94.56	
	cls 9	0.939	0.13	87.56	cls 10 (12.44)
	cls 10	0.697	0.14	75.73	cls 11 (23.74)
	cls 11	0.957	0.13	93.72	
	cls 1	4.648	0.14	85.24	
	cls 2	0.831	0.05	50.07	cls 3 (48.63)
	cls 3	0.696	0.10	91.00	
	cls 4	0.563	0.05	89.15	
	cls 5	0.955	0.02	76.79	
RBF	cls 6	0.228	0.06	57.01	cls 7 (31.46)
	cls 7	0.543	0.05	94.38	
	cls 8	0.139	0.12	94.84	
	cls 9	0.939	0.13	87.56	cls 10 (12.44)
	cls 10	0.697	0.14	73.24	cls 11 (24.28)
	cls 11	0.957	0.13	92.81	
	cls 1	4.648	0.14	83.84	
	cls 2	0.831	0.05	50.16	cls 3 (48.63)
	cls 3	0.696	0.10	90.49	
	cls 4	0.563	0.05	89.33	
	cls 5	0.955	0.02	75.85	cls 3 (11.94)
RBF+VF	cls 6	0.228	0.06	57.83	cls 7 (30.50)
	cls 7	0.543	0.05	94.66	
	cls 8	0.139	0.12	93.56	
	cls 9	0.939	0.13	87.56	cls 10 (12.44)
	cls 10	0.697	0.14	72.88	cls 11 (23.70)
	cls 11	0.957	0.13	91.44	

de PC como de *g-mean*. En general, no se observa una diferencia significativa entre los resultados generados por la ME editada y sin editar. Esta similitud entre los valores obtenidos con la ME editada y sin editar se deben a que la base de datos Cayo no muestra un fuerte solapamiento entre clases (mayoritaria y minoritaria), y en consecuencia el número de muestras eliminadas fue mínimo. En este caso sólo se eliminaron (aproximadamente) el 7% de los elementos de la ME[5].

[5]En detalle, el porcentaje de elementos eliminados por clase es el siguiente. Clase 1: 5.01%, clase 3: 16.72%, clase 8: 2.21%, clase 9: 1.77%, clase 10: 20.38%, clase 11: 9.58%. En el resto de las clases no hubo eliminaciones por considerarse clases minoritarias.

Sin embargo, los resultados obtenidos por clase indican que existe fuerte solapamiento entre algunos pares de clases. Por ejemplo, la clase 2 está confundida para el MLP sobre el conjunto original a un 48.63% respecto de la clase 3. El problema es que este tipo de solapamiento no es identificado por completo por el criterio de los k-NN[6].

Se puede concluir que aunque la confusión existente en algunas clases se mantiene cuando se aplica la combinación EWP$^-$ y la Opción 3, en otras situaciones entre pares de clases si se disminuye el error y por tanto aumenta el PC para esas clases. Visto en su conjunto, la combinación de ambas estrategias genera una mejora tanto a nivel de PC global como sobre todo del valor de *g-mean*.

Tabla 5.14: Desempeño global del clasificador: Base de datos Cayo

	PC			*g-mean*		
	Op. 0	Op. 0	Op. 3	Op. 0	Op. 0	Op. 3
ANN	Original	EWP$^-$	EWP$^-$	Original	EWP$^-$	EWP$^-$
MLP	83.58(0.77)	84.21(1.10)	85.34(0.41)	70.17(6.28)	71.87(6.88)	81.83(0.65)
RBF	79.9(1.63)	78.12(3.37)	83.69(0.62)	58.97(3.94)	61.90(8.89)	79.53(1.12)
RBF+VF	76.92(2.92)	78.66(4.15)	83.13(0.65)	66.99(7.58)	66.09(8.11)	79.17(0.92)

5.6 Conclusión

En este capítulo se estudió la idea de reducir el área de confusión o solapamiento de las clases minoritarias (en relación a las mayoritarias) con el objetivo de incrementar la precisión del clasificador sobre éstas.

El estudio se realizó con bases de datos sintéticas y reales de dos clases, y múltiples clases, sobre las redes MLP, RBF y RBF+VF y el clasificador 3-NN, con las técnicas de edición: EW, EWP, EW$^-$ y EWP$^-$.

En términos generales, y a partir de los resultados mostrados en este capítulo podemos concluir lo siguiente:

- Las técnicas de edición disminuyen la zona de confusión mejorando la PC global del clasificador. Además, cuando se aplican sobre bases de datos desbalanceadas producen un desplazamiento de la frontera de decisión, siendo este desplazamiento mayor cuanto menos están representadas las clases en estas bases de datos.

[6]El porcentaje de elementos eliminados de la clase 3 es el segundo más alto (aproximadamente 16.72%), lo que no se corresponde con los resultados de clasificación encontrados con ANN.

- La EW tiende a desplazar la frontera hacia la zona de influencia de la clase minoritaria, mientras que las otras tres técnicas propuestas EWP, EW$^-$ y EWP$^-$, tienden a desplazar la frontera hacia la zona de influencia de la clase mayoritaria. Estos hechos se ven claramente corroborados en el caso de las bases de datos sintéticas.

- Cuando se aplican sobre problemas reales de dos clases con la regla 3-NN, la EW y la EWP tienden a mejorar la PC global aunque esto suponga un empeoramiento en su *g-mean*. En el caso de la EW$^-$ y EWP$^-$ se empeora la PC global mejorándose el *g-mean*[7].

- En las ANNs con bases de datos reales de dos clases se comportan de forma diferente a nivel de PC y *g-mean*. En el caso de las bases de datos German e Ionosphere, producen peores resultados con respecto al conjunto original en PC global. Aunque se disminuye la incertidumbre en la zona de confusión, el desplazamiento que se genera en la frontera produce resultados poco satisfactorios en términos de PC global. La razón viene dada por la baja representatividad de las clases.

  En el caso de Diabetes el PC global solo mejora con EW, mientras que el resto de las técnicas de edición producen peores resultados. Con Phoneme, las técnicas de edición propuestas mantienen el PC global con el conjunto original, y logran un relativo balance entre clases mejorando la *g-mean*, debido a que está base de datos esta bien representada en cantidad de muestras.

- En dominios de múltiples clases, editar sobre la mayoritaria con EWP$^-$ e incluir la Opción 3 acelera la convergencia de la ANN. En la base de datos Ecoli6, la aplicación de esta estrategia reduce los valores de PC como de *g-mean* debido a la baja representatividad de las clases.

  En la base de datos Cayo produce efectos beneficiosos, aunque no resuelve la confusión existente en algunas clases. Visto en su conjunto, esta estrategia genera una mejora tanto a nivel de PC global como sobre todo del valor de *g-mean*.

En conclusión, si lo que se desea es dar prioridad a las clases minoritarias, editar la región de solapamiento de las clases minoritarias en relación a las mayoritarias, es una alternativa efectiva que aunque no genere una PC global significativa reduce la zona de confusión para las minoritarias.

[7] Estas dos últimas técnicas de edición al favorecer a la clase minoritaria equilibran ambas clases y aumenta el valor del *g-mean*.

Es indudable que se requiere seguir estudiando en esta dirección con el objetivo de idear nuevos mecanismos que permitan identificar correctamente los elementos en la zona de confusión, y de esta forma dar prioridad a las clases minoritarias. Una propuesta inicial sería la de trabajar en el espacio oculto de la red, para identificar aquellos elementos de la ME que puedan ser sospechosos de estar situados en la zona de solapamiento de la clase minoritaria.

Capítulo 6

Aportaciones, Conclusiones y Trabajos Futuros

En los últimos años el problema del desbalance de las clases ha sido el tema central de numerosas investigaciones en áreas como el reconocimiento de formas, el aprendizaje automático y la minería de datos.

En el ámbito del reconocimiento de formas a través del uso de ANN, el desbalance se ha reconocido como un problema crítico que afecta directamente la eficiencia y efectividad del clasificador. Específicamente, en las ANNs entrenadas con el algoritmo back-propagation con procesamiento por grupos, el problema del desbalance de las clases ralentiza la convergencia de las clases menos representadas, lo que se traduce en una pérdida de la efectividad del clasificador sobre estas clases.

En este trabajo se ha investigado empíricamente el problema del desbalance de las clases sobre tres modelos de ANNs entrenadas con el algoritmo back-propagation con procesamiento por grupos. Así mismo, se han estudiado diferentes enfoques para tratar este problema.

6.1 Aportaciones y conclusiones

La principal aportación de esta tesis es la presentación de un estudio detallado del problema del desbalance de las clases en el contexto de las ANNs entrenadas con el algoritmo back-propagation con procesamiento por grupos. Los modelos neuronales utilizados fueron las redes MLP, RBF y RBF+VF. Obsérvese que se trata de tres estructuras distintas.

En general, las aportaciones más importantes de este trabajo se pueden clasificar en los siguientes puntos de interés.

6.1.1 Estudio de los efectos del desbalance de las clases

En este punto se estudió como el desbalance de las clases en la ME impacta en la convergencia de la ANN. Se evidenció la fuerte relación que existe entre el desbalance y la convergencia de la ANN sobre las clases menos representadas. Así mismo, se observó que este problema viene dado por la desproporción del MSE de las clases minoritarias en relación a las mayoritarias. Esto es de especial importancia dado que el algoritmo back-propagation se basa en un criterio de minimización del MSE.

La aportación más relevante de esta parte de la investigación es que sirvió como punto de partida para el desarrollo de esta tesis.

6.1.2 Tratamiento del desbalance de las clases a partir de la inclusión de funciones de coste

A partir de las observaciones del punto anterior se decidió analizar y evaluar diferentes mecanismos para equilibrar las proporciones de MSE de las distintas clases. Para ello, se realizaron experimentos con conjuntos de datos desbalanceados de dos y múltiples clases. La aportación más importante en este punto fue el análisis de este problema en dominios de múltiples clases y la propuesta de una función de coste basada en el MSE.

Del análisis de los resultados experimentales se obtuvieron las siguientes conclusiones:

- La inclusión de funciones de coste al proceso de aprendizaje de la ANN tiene dos consecuencias: a) acelerar la convergencia de las clases menos representadas y b) reducir la influencia de las clases mayoritarias en el proceso de entrenamiento, lo que se ve reflejado en la efectividad de la ANN en la fase de clasificación.
- Los efectos negativos de incluir funciones de coste al proceso de entrenamiento de la ANN son observados principalmente en situaciones donde las bases de datos están poco y mal representadas y/o existe alto solapamiento entre clases.
- La inclusión de funciones de coste al proceso de entrenamiento prioriza a las clases minoritarias lo que se traduce en incrementos de la efectividad del clasificador sobre estas clases.

6.1.3 Descomposición del problema (ANN-M) para tratar el desbalance de las clases

El estudio del problema del desbalance de la ME en dominios de múltiples clases fue la motivación para el desarrollo de esta parte de la investigación. En otros

trabajos se ha propuesto la idea de descomponer un problema de múltiples clases en subproblemas de dos para simplificar su resolución.

En este punto de la investigación se tomó como base esta idea con el objetivo de resolver problemas desbalanceados de múltiples clases. Para esto se hizo uso de ANN-M. La aportación más relevante de este estudio fue la presentación de un extenso estudio sobre la construcción, entrenamiento y combinación de salidas de los módulos que componen la ANN-M, cuando los conjuntos de datos de entrenamiento presentan desbalance entre clases.

A partir de este estudio se desprendieron interesantes conclusiones que pueden ser resumidas como sigue:

- Descomponer el problema en subproblemas de dos clases reduce la interferencia entre clases, lo que permite que el problema sea más fácil de aprender por la ANN. Esto se ve reflejado en las mejoras observadas en cuestiones de convergencia.

- La descomposición del problema puede incrementar el MSE asociado a la clases menos representadas (por la acentuación del desbalance de las clases). No obstante, es suficiente la aplicación de alguna estrategia como el uso de funciones de coste para reducir este problema.

- Para un rendimiento efectivo por parte la ANN-M, los módulos deben ser construidos y entrenados correctamente obedeciendo a las características individuales de cada subproblema. Sin embargo, el uso de una estrategia adecuada para la combinación de las salidas puede ayudar a reducir algunas de las deficiencias ocurridas durante la construcción y entrenamiento de los módulos.

- La tendencia de las ANN-M es producir iguales o mejores resultados de clasificación que los modelos de ANNs globales.

6.1.4 Reducción de la región de confusión para disminuir los efectos del desbalance de las clases

A lo largo de esta investigación se observó que en muchas ocasiones, el problema de la pobre efectividad del clasificador tiene que ver más con problemas de solapamiento entre clases que con cuestiones de desbalance. La principal aportación de esta parte de la investigación fue la propuesta de algunas estrategias (tomadas del contexto de la regla del vecino más próximo) para reducir la región de confusión. Así mismo, se sugirió la combinación de funciones de coste con técnicas de reducción del solapamiento entre clases.

Se observó que reducir la región de confusión a partir de técnicas tomadas del contexto de la regla del vecino más próximo tiene dos efectos:

- Incrementar la participación de las clases menos representadas en el proceso de entrenamiento.
- Reducir la influencia de las clases mayoritarias.

Estas técnicas pueden ser consideradas bastante efectivas si lo que se desea es dar prioridad a las clases menos representadas. Sin embargo, en algunas situaciones estas técnicas no fueron efectivas para la identificación de elementos en la región de confusión, por lo que es necesaria la búsqueda de otros mecanismos más apropiados para la ubicación de estos elementos.

Por otro lado, la combinación de funciones de coste y técnicas para reducir la región de solapamiento logró disminuir el área de confusión y al mismo tiempo acelerar la convergencia de las clases menos representadas. Esto representa una forma original de tratar el desbalance de las clases.

En general, las estrategias estudiadas en esta tesis para tratar el problema del desbalance incrementan la efectividad del clasificador sobre las clases menos representadas. No obstante, cuando los problemas a resolver presentan características como solapamiento, problemas de representatividad (pocas muestras o mal muestreadas) o ruido en los datos estas estrategias producen efectos negativos en la efectividad de la ANN.

6.2 Futuras líneas de investigación

A lo largo de esta investigación se observaron resultados interesantes, que permitieron vislumbrar posibles escenarios hacia donde encaminar futuras investigaciones sobre el problema del desbalance. Algunas de estas ideas son de especial interés porque han sido poco exploradas y obedecen a problemáticas actuales. A continuación se resumen algunas de ellas.

- Estudiar mecanismos que aceleren la convergencia de las clases minoritarias, y que a medida que el entrenamiento avance disminuyan los efectos negativos de las funciones de coste sobre las clases mayoritarias. Esto implica el desarrollo de nuevas funciones de coste basadas en el MSE que logren este objetivo.
- Los criterios de medida F1 y el de los k-NN para medir el nivel de solapamiento entre clases, en algunos casos no fueron suficientes. Es necesario buscar estrategias específicas para medir los niveles de solapamiento o separabilidad entre clases dentro del contexto de las ANNs.

Si se cuenta con estos mecanismos se pueden optimizar las estrategias diseñadas para reducir la región de confusión, o identificar con mayor precisión el tratamiento que se le deba dar a cada conjunto de datos en función de sus características.

Por otro lado, la trascendencia real de esta propuesta está en que los criterios empleados para identificar los elementos redundantes o en la frontera de decisión incluyan caracteristicas propias de la ANN. Por ejemplo, la búsqueda de los elementos representativos puede darse en el espacio oculto o de salida, en lugar del espacio de características, y el criterio de medida puede estar basado en valores de MSE.

- La tendencia actual en el diseño y construcción de ANNs para la resolución de problemas complejos, es la utilización de paradigmas modulares. Sin embargo, es necesario continuar con el estudio de modelos monolíticos ya que son la parte más básica de los paradigmas modulares. Se vio en este trabajo la trascendencia de construir y entrenar correctamente cada módulo para que la ANN-M fuera efectiva.

  En este punto se propone profundizar en la combinación de técnicas de reducción de la región de confusión, funciones de coste y minimización de la talla del conjunto de datos, para la construcción de los módulos individuales[1]. La implementación de estas ideas en los módulos independientes puede dar lugar a resultados interesantes.

6.3 Publicaciones

Parte de este trabajo de tesis ha sido presentado en diversos congresos y revistas de divulgación tanto nacional como internacional. Así mismo, algunos de los resultados generados a partir de las ideas expuestas en esta tesis han sido contribuciones importantes para otras publicaciones. En esta sección se enumeran algunos de estos trabajos.

Artículos en la serie Lecture Notes in Computer Science

1. **R. Alejo, J.M. Sotoca, V. García and R.M. Valdovinos** *Training cost-sensitive neural networks with editing techniques for imbalanced classes.* In the 2nd Mexican Conference on Pattern Recognition (MCPR 2010). Springer-Verlag, 2010 (In press).

[1] Por ejemplo, en otras técnicas de clasificación como las SVM se establece un submuestreo en pares de clases para luego aplicar una regla de votación

2. **R. Alejo, J.M. Sotoca, R.M. Valdovinos and P. Toribio.** *Edited Nearest Neighbor Rule for Improving Neural Networks Classifications.* In the Seventh International Symposium on Neural Networks (ISNN 2010). LNCS 6063. Pág. 303–310. Springer-Verlag, 2010.

3. **L. Cleofas, R.M. Valdovinos, V. García and R. Alejo.** *Use of Ensemble Based on GA for Imbalance Problem.* In the Sixth International Symposium on Neural Networks (ISNN 2009). LNCS 5552. Pág. 547–554. Springer-Verlag, 2009.

4. **R. Alejo, J.M. Sotoca and G.A. Casañ.** *An empirical study for the multi-class imbalance problem with neural networks.* In 13th Iberoamerican Congress on Pattern Recognition (CIARP 2008). LNCS 5197. Pág. 479–486. Springer-Verlag, 2008.

5. **R. Alejo, V. García, J.M. Sotoca, R.A. Mollineda and J.S. Sánchez.** *Improving the performance of the RBF neural network trained with imbalanced samples.* In 9th International Work-Conference on Artificial Neural Networks (IWANN 2007). LNCS 4507. Pág. 162–169. Springer-Verlag, 2007.

6. **V. García, R.A. Mollineda, J.S. Sánchez, R. Alejo and J.M. Sotoca.** *When overlapping unexpectedly alters the class imbalance effects.* In third Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2007). LNCS 4478. Pág. 499–506. Springer-Verlag, 2007.

7. **R. Alejo, V. García and J.M. Sotoca, R.A. Mollineda and J.S. Sánchez.** *Improving the classification accuracy of RBF and MLP neural networks trained with imbalanced samples.* In Intelligent Data Engineering and Automated Learning (IDEAL 2006). LNCS 4224. Pág. 464–471. Springer-Verlag, 2006.

8. **V. García, R. Alejo, J.S. Sánchez, J.M. Sotoca and R.A. Mollineda.** *Combined Effects of Class Imbalance and Class Overlap on Instance-Based Classification.* In Intelligent Data Engineering and Automated Learning (IDEAL 2006). LNCS 4224. Pág. 371–378. Springer-Verlag, 2006.

Otros artículos internacionales

1. **R. Alejo, J.M. Sotoca, R.M. Valdovinos and G.A. Casañ.** *The multi-class imbalance problem: cost functions with modular and non-modular neural networks.* In the Sixth International Symposium on Neural Networks (ISNN 2009). ASC 56. Pág. 421–431. Springer-Verlag, 2009.

2. **R. Alejo, J.M. Sotoca and G.A. Casañ.** *Error Analysis in Artificial Neural Networks: the Imbalanced Distribution Case.* Simulation modelling and optimization, WSEAS, ISBN 978-960-474-007-9 Pág. 401–407, 2008.

3. **R. Barandela, E. Gasca and R. Alejo.** *Correcting the Training Data.* Pattern Recognition and String Matching, Series in Combinatorial Optimization, Vol. 13, D. Chen, X. Cheng (eds.), Kluwer Academic Publishers, ISBN 978-1-4020-0953-2, Pág. 1–42, 2003.

4. **J.S. Sánchez, R. Barandela, A.I. Marqués and R. Alejo.** *Performance Evaluation of Prototype Selection Algorithms for Nearest Neighbor Classification.* Computer Graphics and Image Processing, IEEE Computer Society ISBN: 0-7695-1330-1 Pág. 44–50, 2001.

Artículos en revistas de divulgación nacional

1. **P. Toribio, B. Rodríguez, R.M. Valdovinos and R. Alejo** *Training Optimization for Artificial Neural Networks.* CIENCIA ergo sum, 2010. In press.

2. **R. Alejo, V. García y F. García.** *Redes Neuronales Artificiales y Distribuciones no Balanceadas.* Programación Matemática y Software. Vol. 1(1), Pág. 13–32. 2009.

3. **R. Alejo, P. Toribio y F. García.** *Análisis del Error en Redes Neuronales: Distribuciones no Balanceadas.* Research in Computing Science, Pág. 31–40, 2008.

4. **R. Alejo, F. García, y M.G. de la Rosa.** *Distribuciones no balanceadas en redes neuronales artificiales de funciones de base radial.* Ciencia y tecnología en la frontera, Pág. 86–93, 2008.

5. **R. Barandela, E. Gasca, y R. Alejo** *Corrección de la Muestra para el Aprendizaje del Perceptron Multicapa.* Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. Vol. 13, Pág. 2–9, 2001.

Artículos en revistas de divulgación internacional

1. **J.S. Sánchez, R. Barandela, A.I. Marqués, R. Alejo, R. and J. Badenas.** *Analysis of new techniques to obtain quality training sets.* Pattern Recognition Letters Vol. 24(7), Pág. 1015–1022, 2003.

Apéndice A

Desbalance de las clases: MLP vs Redes RBF

Contenido

A.1 Introducción

En los últimos años se ha popularizado el empleo de redes neuronales artificiales en tareas de aprendizaje automático, reconocimiento de formas y minería de datos. En especial las redes RBF y el MLP. Éstas comparten varias características en común. Por ejemplo, son de redes de propagación hacia adelante (*feedforward*) con capas no lineales [Haykin 1999], aproximadores universales [Looney 1997] y modelos que pueden ser entrenados con métodos similares de descenso por gradiente (por ejemplo, con el algoritmo back-propagation) [Schwenker 2001]. No obstante, ambas presentan importantes diferencias [Ding 2004]:

1. Las redes RBF tienen una capa oculta y el MLP pueden tener una o más.

2. Generalmente, en el modelo MLP los nodos ocultos y los de salida tienen el mismo modelo neuronal, mientras que en las redes RBF el modelo neuronal de la capa oculta y el de salida es distinto.

3. Los MLP generan una aproximación global de la relación no lineal entrada-salida en tanto que en las redes RBF esta relación es local.

4. La principal diferencia entre la red RBF y el MLP está en la función de activación de los nodos ocultos. En las redes RBF depende de la distancia entre los vectores de entrada y los centros de la red. En el MLP depende del producto del vector de entrada y el vector de pesos.

Las redes RBF y el MLP, al igual que otros mecanismos de aprendizaje automático, muestran desempeños deficientes en contextos donde los datos de entrenamiento presentan desbalance en la distribución de las clases[1] [Japkowicz 2002]. Sin embargo, no esta claro si el efecto del desbalance de las clases es el mismo en ambas redes o cuál es la diferencia entre la una y la otra.

En investigaciones recientes [Alejo 2006, Alejo 2008, Alejo 2009] se ha observado que en la resolución de algunos problemas que involucran bases de datos desbalanceadas, el desempeño de las redes RBF es inferior al mostrado por el MLP. Por lo que surgen la pregunta ¿qué hace la diferencia entre ambas?.

En esta sección, se desarrolla un estudio empírico (con bases de datos artificiales y reales) de las diferencias fundamentales entre el MLP y la red RBF, cuando son entrenadas con el algoritmo back-propagation con procesamiento por grupos y bases de datos desbalanceadas.

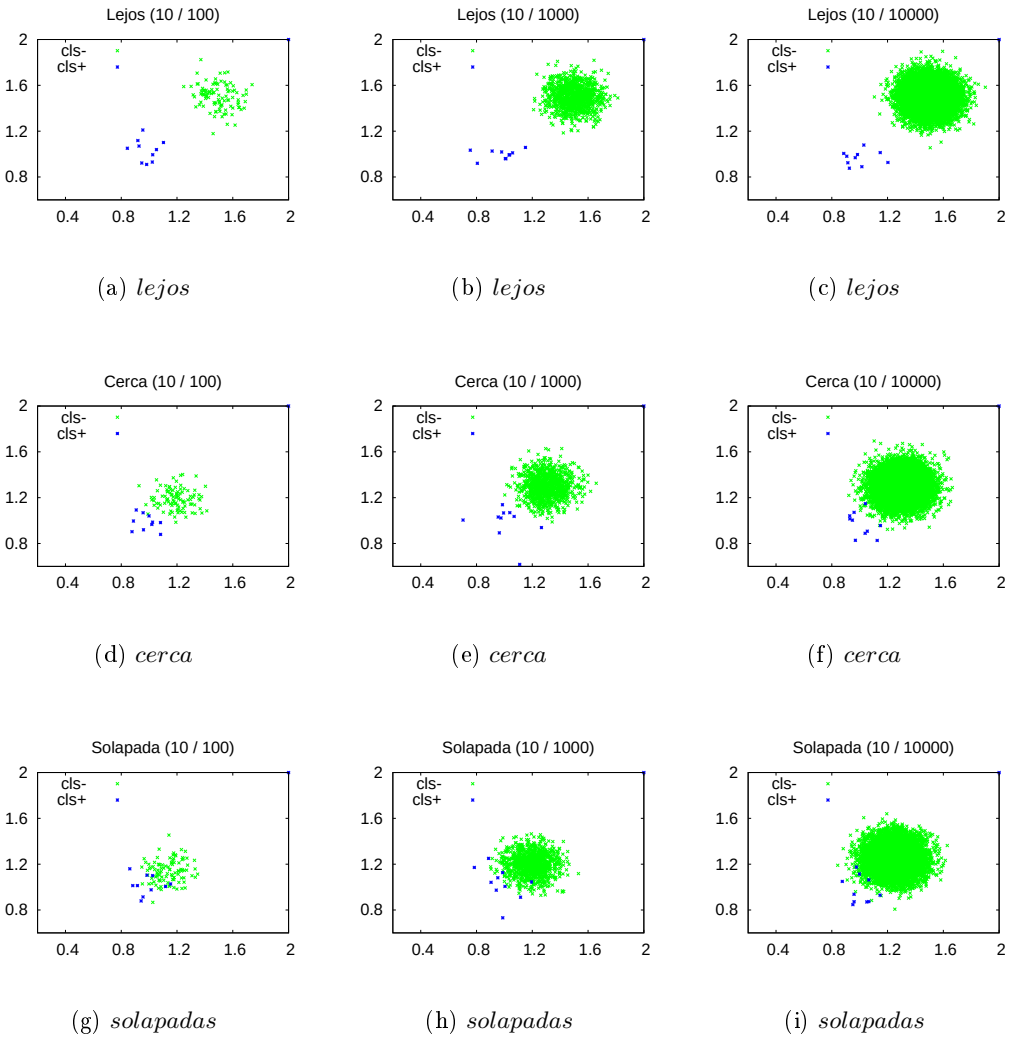
A.2 Aspectos metodológicos

Trabajos anteriores [Alejo 2006, Alejo 2008, Alejo 2009] sugieren que la diferencia básica entre el MLP y la red RBF cuando son entrenadas con el algoritmo back-propagation y bases de datos desequilibradas, tiene que ver más con problemas de solapamiento o separabilidad entre clases que con cuestiones de desbalance.

Para estudiar esta hipótesis se desarrollaron diversos experimentos con bases de datos sintéticas y desequilibradas en tres escenarios de separabilidad distintos. Las bases de datos artificiales fueron diseñadas según un modelo de dos clases con distribuciones gaussianas bivariadas.

En la Fig. A.1a-c se presenta el primer escenario donde las clases muestran un alto nivel de separabilidad. La Fig. A.1d-f corresponde al segundo escenario en el que las clases se encuentran a una menor distancia pero sin llegar al solapamiento. En la Fig. A.1g-i se observa solapamiento entre clases. En cada uno de los escenarios el desbalance entre clases es de 1:10, 1:100 y 1:1000.

[1] Problema del desbalance de las clases.



(a) *lejos* (b) *lejos* (c) *lejos*

(d) *cerca* (e) *cerca* (f) *cerca*

(g) *solapadas* (h) *solapadas* (i) *solapadas*

Fig. A.1: Bases de datos sintéticas de dos clases con diferentes niveles de separabilidad y desbalance entre clases.

Se entrenaron las ANNs con el algoritmo back-propagation con procesamiento por grupos y se estableció el criterio de parada en 100000 iteraciones o un error inferior a 0.0001. Para el MLP la razón de aprendizaje se fijó en 0.9 mientras que en las redes RBF fue de 0.00001.

El objetivo de utilizar un bajo valor en la razón de aprendizaje de la red RBF es el de evitar oscilaciones en el MSE a causa del ajuste de los centros y varianzas de la red en cada una de las iteraciones. En ambos modelos se utilizaron dos neuronas en la capa oculta.

A.3 Resultados experimentales

En esta sección se presentan algunos resultados obtenidos al experimentar con el MLP y la red RBF con bases de datos desequilibradas en los escenarios descritos anteriormente. El objetivo de mostrar estos resultados es el de tratar de responder algunas interrogantes en relación a los efectos del desbalance de las clases sobre las redes RBF y el MLP.

A.3.1 ¿Es más sensible la red RBF al desbalance de la ME que el MLP?

En las Fig. A.2, A.3 y A.4 se presenta el MSE de la clase minoritaria correspondiente a las tres bases de datos sintéticas de las Fig. A.1a, A.1b y A.1c. Cada linea pertenece a una inicialización distinta de la red. Observe que el eje x ha sido escalado logarítmicamente dado que los principales cambios ocurren durante las primeras iteraciones.

En este escenario las clases son altamente separables y se tienen tres niveles de desbalance. La finalidad de utilizar estas bases de datos es que no existen factores como ruido, atípicos o solapamiento en las muestras de entrenamiento que puedan interferir en el aprendizaje de la ANN. Por lo tanto, el único factor que puede considerarse como problemático para el aprendizaje es el desbalance de las clases.

Si se observan las Fig. A.2, A.3 y A.4 se podría pensar que el desbalance afecta más a las redes RBF que al MLP. Sin embargo, el considerable incremento del MSE y la variabilidad del mismo que se observa en las redes RBF durante las primeras iteraciones, es a consecuencia de la inicialización de la red RBF y no del desbalance.

Al inicializar aleatoriamente la red RBF se tienen valores iniciales de MSE distintos y esto es debido a que tanto pesos como los centros y varianzas son determinados aleatoriamente. Estos resultados evidencian que el MLP es menos sensible a la inicialización de sus parámetros libres que la red RBF.

Ahora bien, si se excluye el problema del desbalance (al aplicar la Opción 1) se puede observar que en ambas ANNs la convergencia del MSE de la clase minoritaria es alcanzada en prácticamente en el mismo número de iteraciones (excepto por la red RBF con la base de datos *lejos* 10/100), independientemente de la inicialización de la ANN.

Sin embargo, al observar el MSE de la clase minoritaria obtenido sin ningún tipo de compensación del error, se puede apreciar una mayor inestabilidad en el MSE. Esto quiere decir que el incremento de la inestabilidad en el MSE de la clase minoritaria es ocasionado por el desbalance de las clases.

Obsérvese en las Fig. A.2, A.3 y A.4 que el MSE de la clase minoritaria es mucho más estable cuando las proporciones de error son equilibradas por la inclusión de la función de costo (Opción 1).

A.3.2 ¿Cómo afecta la separabilidad de las clases a las redes MLP y RBF cuando se tiene desbalance en las clases?

Para evaluar los efectos de la separabilidad de las clases en el MLP y las redes RBF se utilizaron las bases de datos de las Fig. A.1d-i. En esta última existe solapamiento entre clases mientras que en la primera, las clases se encuentran muy cerca entre si pero sin llegar al solapamiento de las clases.

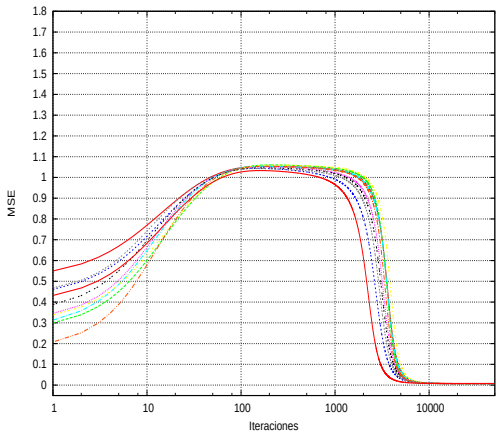
En las Fig. A.5, A.6, A.7, A.8, A.9 y A.10 se presenta el MSE de la clase minoritaria para diferentes inicializaciones de la ANN con las bases de datos *cerca* y *solapadas* (correspondientes a las Fig. A.1d-f y A.1g-i respectivamente).

En estas figuras se observa que a medida que se reduce la separabilidad entre clases, el número de iteraciones necesarias para alcanzar la convergencia (cuando las aportaciones de error son equilibradas con Opción 1) es mayor en la red RBF que en el MLP. Esto evidencia lo sugerido en trabajos previos en el sentido de que las redes RBF son más vulnerables a la separabilidad entre clases cuando presentan distribuciones desbalanceadas.

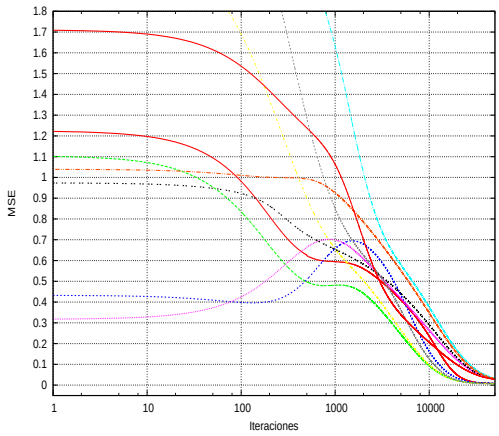
La Tabla A.1 presenta los resultado obtenidos en la fase de clasificación por las redes MLP y RBF. Estos resultados corresponden con el comportamiento el MSE presentado en las figuras anteriores. Sin embargo, pueden observarse dos valores de *g-mean* (de las bases de datos *cerca* –desbalance de 10:1000– y *lejos* –desbalance de 10:10000– en la red RBF con la Opción 0) que parecen contradecir lo discutido hasta ahora.

Estos resultados pudieran sugerir que el descenso del error cuando este no es compensado (Opción 0) es más rápido en la red RBF. No obstante, al analizar la desviación estándar correspondiente a cada valor de *g-mean* se observa que este comportamiento no es una tendencia, sin más bien algo fortuito debido al problema de inicialización de la red. Antes se afirmo que la red RBF es más sensible a la inicialización de sus parámetros libres.

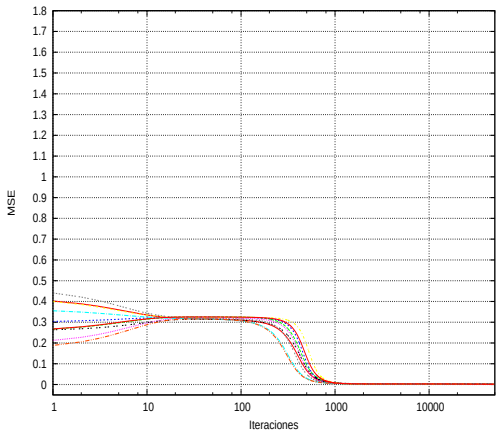
La búsqueda de la configuración inicial de la ANN es una línea de investigación que desde hace muchos años esta presente en los diferentes modelos de redes neuronales, y que ahora debe considerar el desbalance de las clases por que es un factor



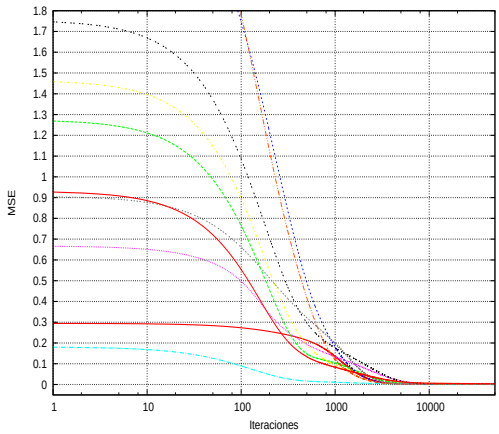
(a) MLP (Op. 0)



(b) RBF (Op. 0)



(c) MLP (Op. 1)



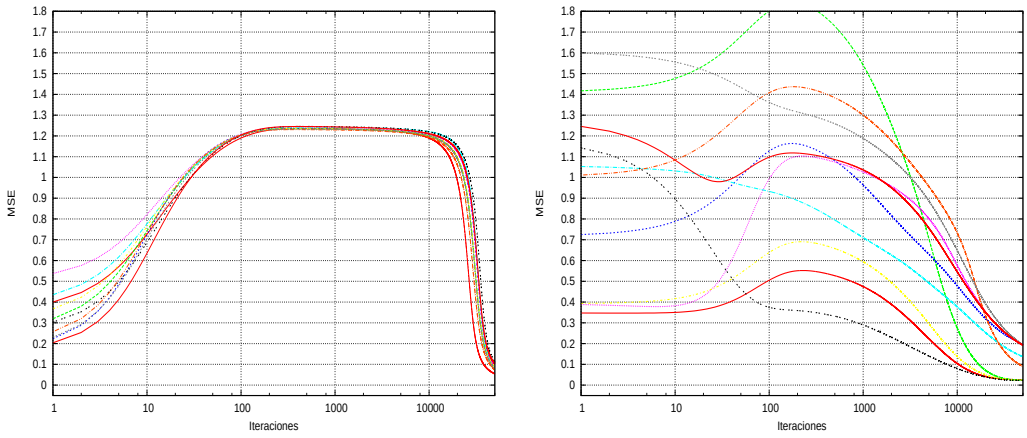
(d) RBF (Op. 1)

Fig. A.2: MSE correspondiente a la clase minoritaria de las bases de datos sintéticas *lejos* (10-100) obtenido por los clasificadores MLP y redes RBF con el algoritmo back-propagation estándar con procesamiento por grupos.

decisivo en los algoritmos basados en reglas de corrección de error.

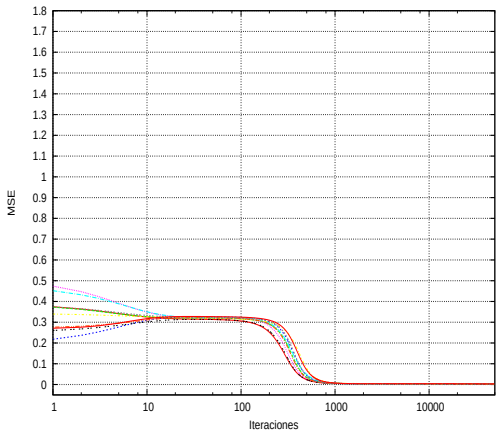
Tabla A.1: Resultados obtenidos en la fase de clasificación por la red neuronal MLP y RBF, con las bases de datos *lejos*, *cerca* y *solapada* correspondientes a las figuras A.1a-c, A.1d-f, A.1g-i respectivamente.Los valores entre paréntesis hacen referencia a la desviación estándar.

	10 / 100		10 / 1000		10 / 10000	
			Opción 0			
MLP	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Lejos	100.00(0.00)	100.00(0.00)	99.98(0.04)	98.98(2.06)	99.90(0.00)	0.00(0.00)
Cerca	99.09(0.00)	99.51(0.00)	99.01(0.00)	0.00(0.00)	99.90(0.00)	0.00(0.00)
Solapada	94.55(0.00)	70.36(0.00)	99.01(0.00)	0.00(0.00)	99.90(0.00)	0.00(0.00)
			Opción 1			
MLP	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Lejos	100.00(0.00)	100.00(0.00)	100.00(0.00)	100.00(0.00)	100.00(0.00)	100.00(0.00)
Cerca	99.09(0.00)	99.51(0.00)	99.76(0.05)	99.89(0.02)	99.04(0.01)	99.53(0.01)
Solapada	81.91(1.66)	85.44(0.96)	92.34(0.66)	91.18(0.33)	98.60(0.08)	99.31(0.04)
			Opción 0			
RBF	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Lejos	100.00(0.00)	100.00(0.00)	99.96(0.05)	97.96(2.53)	99.92(0.04)	18.98(38.14)
Cerca	97.55(1.42)	87.81(11.63)	99.05(0.08)	8.95(17.98)	99.90(0.00)	0.00(0.00)
Solapada	92.45(1.59)	30.72(30.96)	99.01(0.00)	0.00(0.00)	99.90(0.00)	0.00(0.00)
			Opción 1			
RBF	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Lejos	100.00(0.00)	100.00(0.00)	100.00(0.00)	100.00(0.00)	100.00(0.00)	100.00(0.00)
Cerca	97.73(1.17)	98.75(0.65)	98.66(0.54)	99.33(0.28)	99.04(0.29)	99.53(0.15)
Solapada	84.64(2.43)	83.40(2.11)	92.18(0.49)	91.10(0.24)	96.81(0.66)	98.40(0.34)

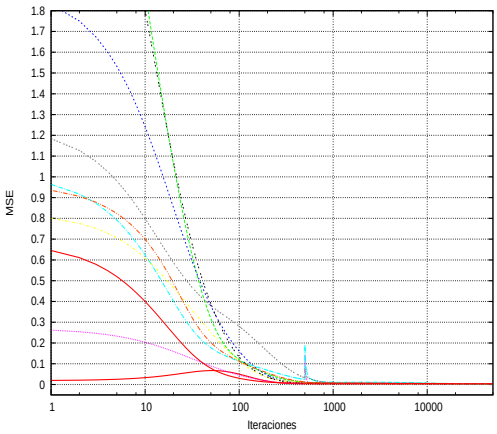


(a) MLP (Op. 0)

(b) RBF (Op. 0)



(c) MLP (Op. 1)

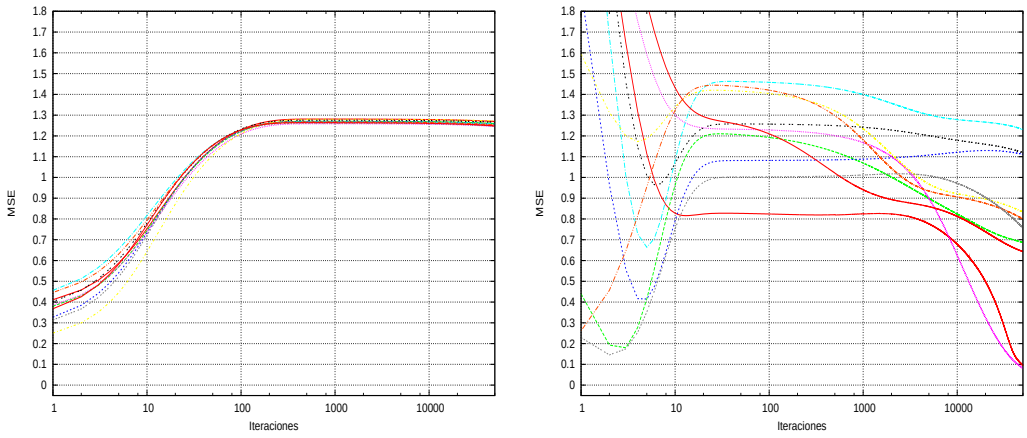


(d) RBF (Op. 1)

Fig. A.3: MSE correspondiente a la clase minoritaria de las bases de datos sintéticas *lejos* (10-1000) obtenido por los clasificadores MLP y redes RBF con el algoritmo back-propagation estándar con procesamiento por grupos.

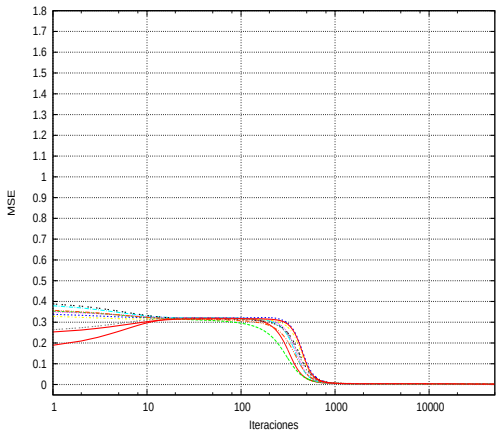
A.3.3 Caso de estudio B2Cls

Los resultados obtenidos con la base de datos B2Cls del capítulo 3 motivaron el desarrollo de este trabajo. Se observó en la Tabla 3.5 de la sección 3.4.4 una notable

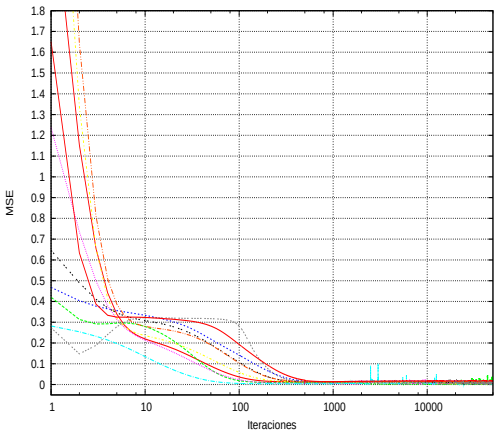


(a) MLP (Op. 0)

(b) RBF (Op. 0)



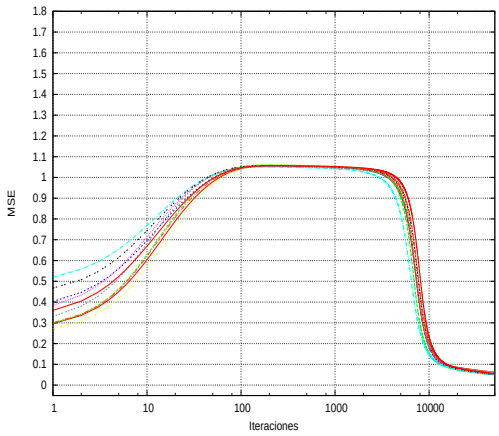
(c) MLP (Op. 1)



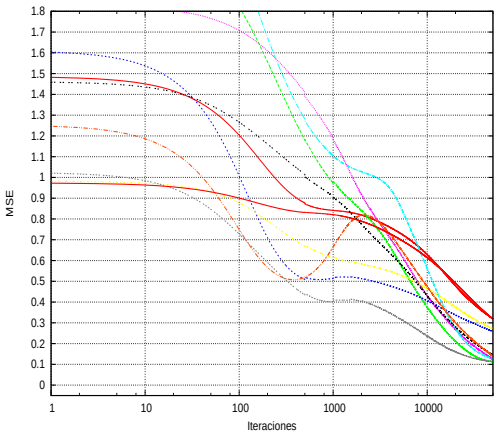
(d) RBF (Op. 1)

Fig. A.4: MSE correspondiente a la clase minoritaria de las bases de datos sintéticas *lejos* (10-10000) obtenido por los clasificadores MLP y redes RBF con el algoritmo back-propagation estándar con procesamiento por grupos.

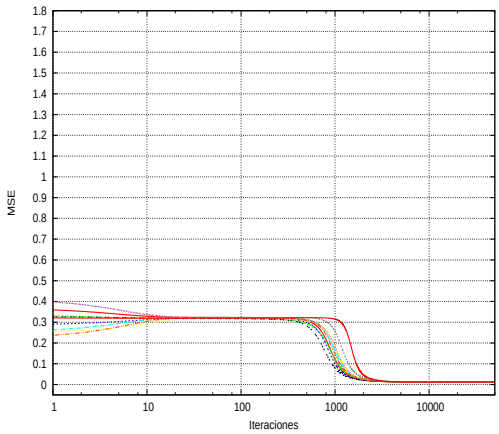
diferencia entre los valores de PC (Precisión en la clasificación) y *g-mean* obtenidos por el MLP y la red RBF. Por lo que fue necesario el desarrollo de estos experimentos.



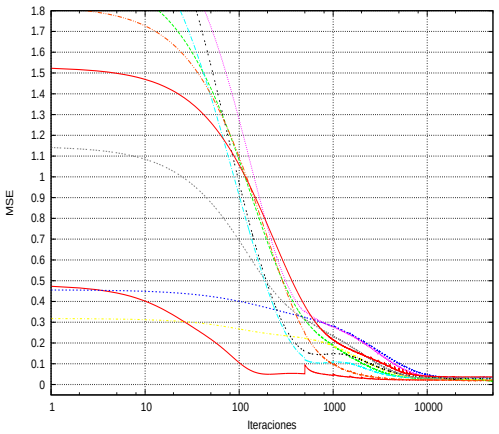
(a) MLP (Op. 0)



(b) RBF (Op. 0)



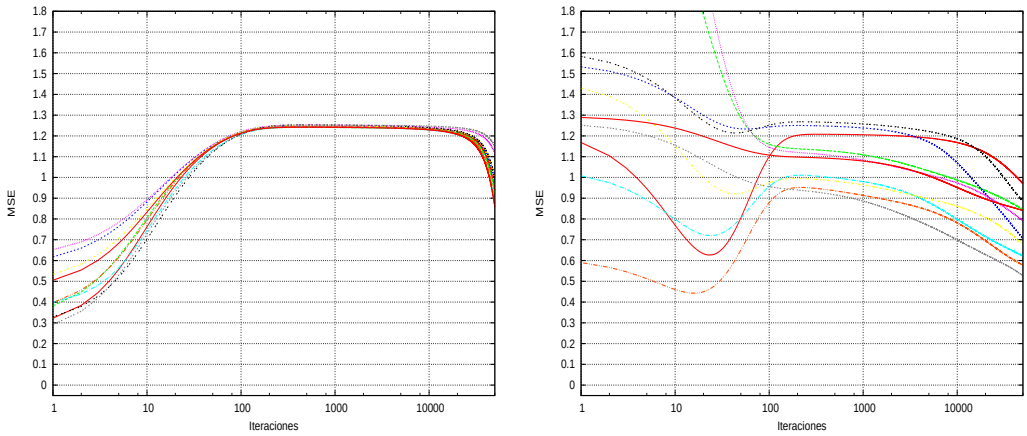
(c) MLP (Op. 1)



(d) RBF (Op. 1)

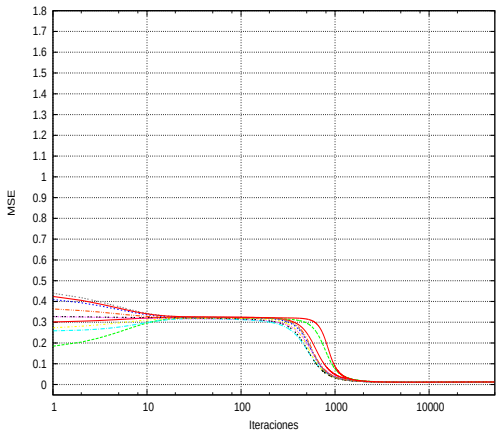
Fig. A.5: MSE correspondiente a la clase minoritaria de las bases de datos sintéticas *cerca* (10-100) obtenido por los clasificadores MLP y redes RBF con diferentes inicializaciones.

La pregunta que surgió fue ¿cuál es la causa de esta diferencia entre el MLP y la red RBF al aplicar las Opciones 1 y 2, es decir, al compensar las aportaciones al

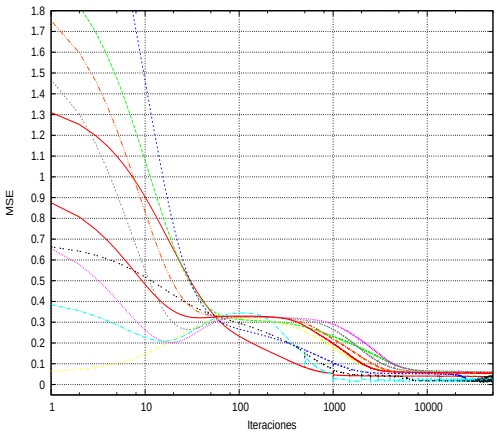


(a) MLP (Op. 0)

(b) RBF (Op. 0)



(c) MLP (Op. 1)

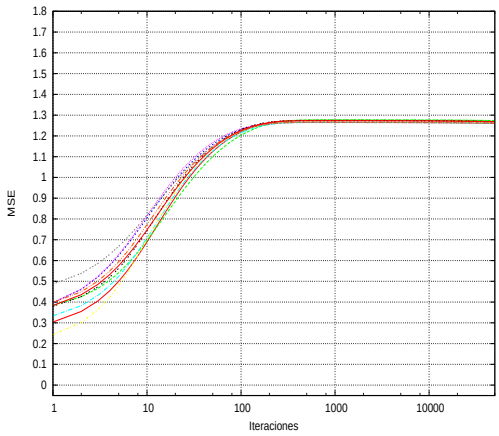


(d) RBF (Op. 1)

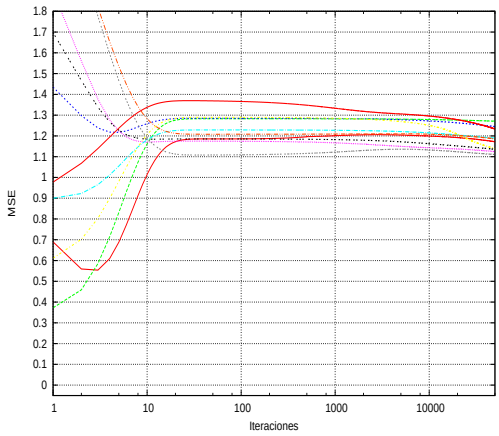
Fig. A.6: MSE correspondiente a la clase minoritaria de las bases de datos sintéticas *cerca* (10-1000) obtenido por los clasificadores MLP y redes RBF con diferentes inicializaciones.

MSE?.

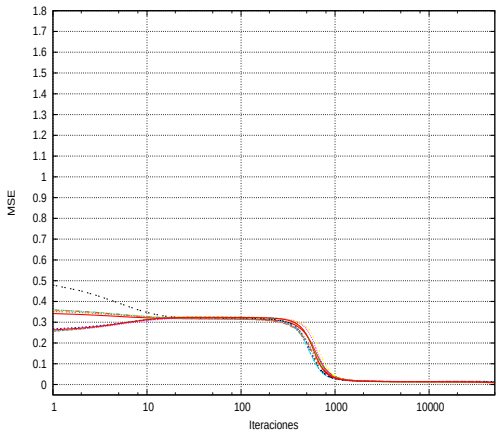
Los resultados discutidos hasta el momento sugieren que el problema es que la



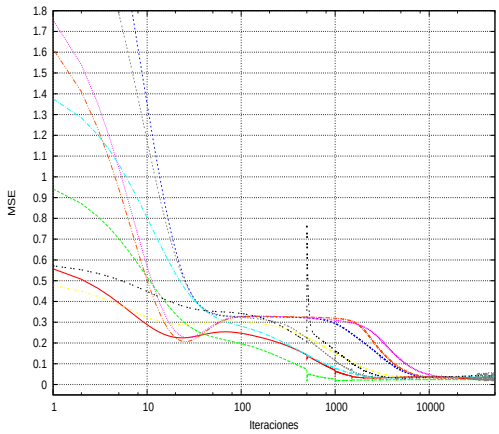
(a) MLP (Op. 0)



(b) RBF (Op. 0)



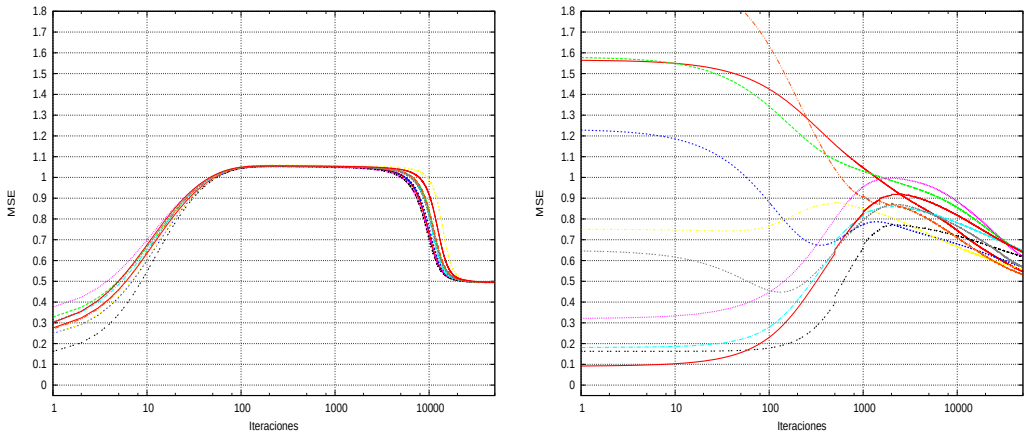
(c) MLP (Op. 1)



(d) RBF (Op. 1)

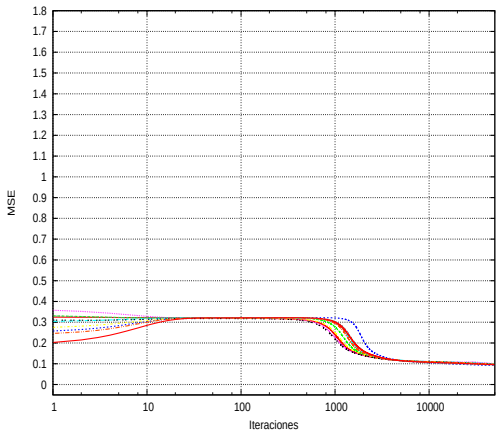
Fig. A.7: MSE correspondiente a la clase minoritaria de las bases de datos sintéticas *cerca* (10-10000) obtenido por los clasificadores MLP y redes RBF con diferentes inicializaciones.

red RBF es más sensibles al solapamiento o separabilidad entre clases. La causa de esta disparidad entre valores de PC y *g-mean* es el solapamiento entre clases.

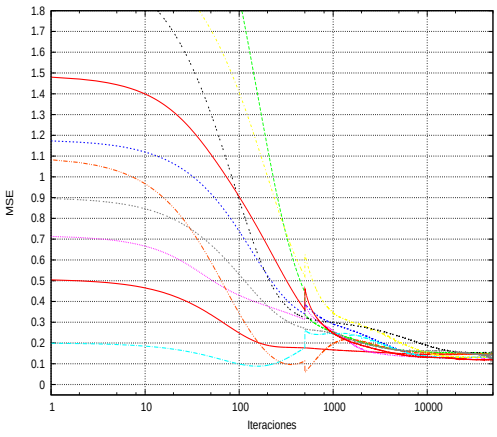


(a) MLP (Op. 0)

(b) RBF (Op. 0)



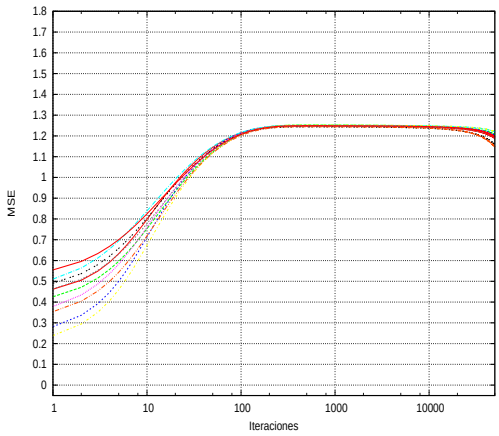
(c) MLP (Op. 1)



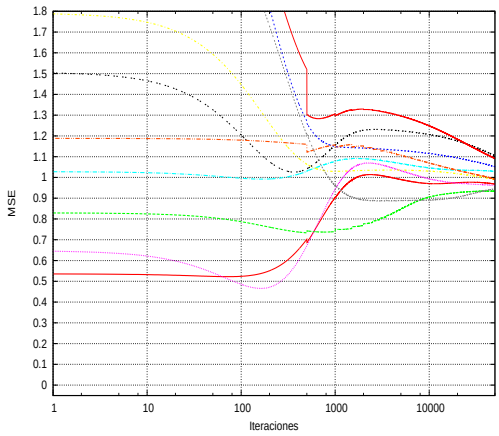
(d) RBF (Op. 1)

Fig. A.8: MSE correspondiente a la clase minoritaria de las bases de datos sintéticas *solapadas* (10-100) obtenido por los clasificadores MLP y redes RBF con diferentes inicializaciones.

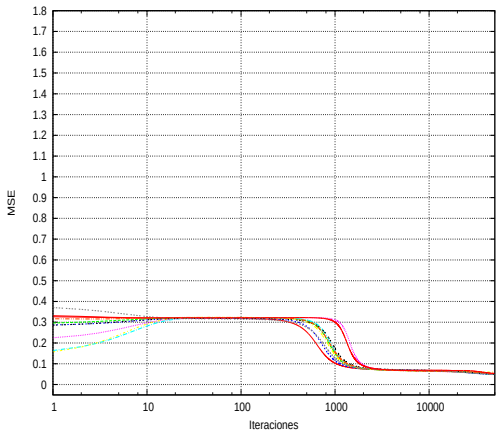
Según el criterio de medida de solapamiento basado en los k vecinos más próximos, esta base de datos presenta altos niveles de solapamiento. Sin embargo, se debe tener



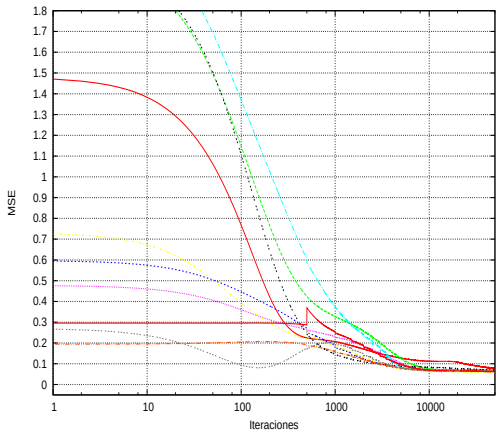
(a) MLP (Op. 0)



(b) RBF (Op. 0)



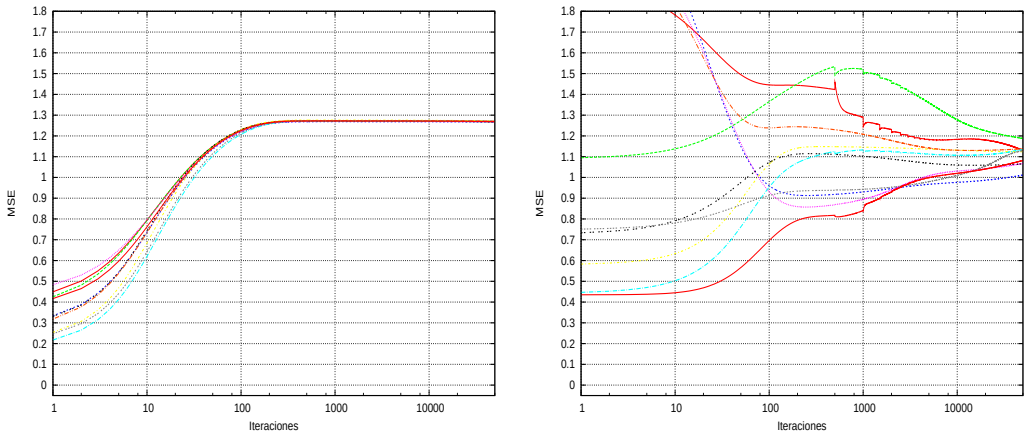
(c) MLP (Op. 1)



(d) RBF (Op. 1)

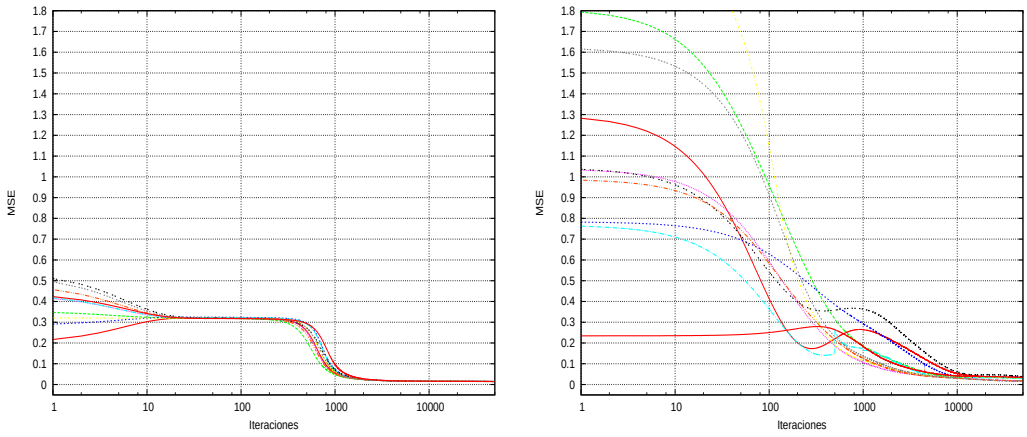
Fig. A.9: MSE correspondiente a la clase minoritaria de las bases de datos sintéticas *solapadas* (10-1000) obtenido por los clasificadores MLP y redes RBF con diferentes inicializaciones.

certeza en esta afirmación y para tratar de evidenciar esta hipótesis se desarrollaron los siguientes experimentos.



(a) MLP (Op. 0)

(b) RBF (Op. 0)



(c) MLP (Op. 1)

(d) RBF (Op. 1)

Fig. A.10: MSE correspondiente a la clase minoritaria de las bases de datos sintéticas *solapadas* (10-10000) obtenido por los clasificadores MLP y redes RBF con diferentes inicializaciones.

La base de datos B2Cls cuenta con 2 clases y 4 atributos por lo que visualmente es muy difícil determinar de que se trata de un conjunto de datos solapado. Para

excluir por completo que el problema fuese el desbalance de la ME, se prosiguió de la siguiente manera: A la base de datos B2Cls se le aplico una estrategia de submuestreo aleatorio de tal manera que se genero un conjunto de datos balanceado (Bal-01).

Posteriormente se fue incrementando (aleatoriamente) el número de muestras de la clase que inicialmente era la mayoritaria hasta regresarla a su estado original. En este proceso gradual se obtuvieron Bal-03, Bal-05, Bal-07 y Bal-10. Los valores 03, 05, ..., 10 corresponde a la proporción de la clase minoritaria en relación a la mayoritaria. En 03 la mayoritaria es tres veces el tamaño de la minoritaria, en 05 cinco veces y así sucesivamente.

A cada base de datos se le aplico la estrategia de validación cruzada $k - fold - crossvalidation$ con $k = 10$ y cada ejecución de la ANN se repitió 10 veces.

Obsérvese en la Tabla A.2 (con la Opción 0) que cuando no existe desbalance (Bal-01), el resultado coincide con el presentado por la base de datos original. La eficacia del MLP es superior a la de la red RBF. Esta tendencia se observa al irse incrementado el desbalance de las clases.

Al aplicarse la Opción 1 este comportamiento sigue observándose y se evidencia que el rendimiento obtenido con la Opción 0 a causa del desbalance de las clases es mejorado al aplicarse la Opción 1. Sin embargo, en ningún caso se alcanzan los valores de PC y *g-mean* obtenidos por el MLP.

Estos resultados sugieren que B2Cls es un problema más difícil de aprender por la red RBF que por el MLP, siendo la red RBF más sensible al solapamiento o separabilidad entre clases.

Tabla A.2: Resultados de la fase de clasificación de las redes MLP y RBF, con la base de datos B2Cls. Los valores entre paréntesis hacen referencia a la desviación estándar.

	MLP		RBF	
Opción 0	PC	*g-mean*	PC	*g-mean*
Bal-01	85.58(14.68)	84.70(15.87)	62.71(17.39)	58.98(20.15)
Bal-03	86.32(12.34)	62.34(43.10)	72.22(5.21)	6.80(16.31)
Bal-05	83.13(1.35)	0.45(4.47)	82.29(2.72)	2.68(10.67)
Bal-07	87.49(0.85)	0.00(0.00)	87.09(1.34)	1.79(8.81)
Bal-10	92.16(0.50)	0.00(0.00)	92.16(0.50)	0.00(0.00)
Opción 1	PC	*g-mean*	PC	*g-mean*
Bal-01	85.29(13.37)	84.50(13.87)	62.77(16.97)	59.46(18.82)
Bal-03	94.34(5.54)	93.30(7.82)	63.42(9.90)	59.68(20.20)
Bal-05	89.63(9.51)	88.61(11.67)	66.36(7.29)	63.57(13.95)
Bal-07	90.44(2.43)	90.81(6.70)	68.78(5.77)	68.68(11.18)
Bal-10	89.51(6.40)	85.82(20.47)	68.29(6.87)	52.81(28.98)

A.4 Conclusión

Se ha afirmado que siempre existe una red RBF capaz de igualar la eficacia de un MLP. Sin embargo, la presencia de algunos factores como el desbalance de las clases, el solapamiento o la baja separabilidad entre clases, ocasiona que los problemas sean más difíciles de aprender por la red RBF.

En esta sección se han desarrollado una serie de experimentos con datos sintéticos y reales para tratar de confirmar esta hipótesis.

En términos generales se observó lo siguiente:

- La red RBF es más sensible a la configuración inicial de la red que el MLP.
- El desbalance de las clases ocasiona una mayor inestabilidad en el MSE de la clase minoritaria de la red RBF.
- A medida que se reduce la separabilidad entre clases la red RBF requiere de más iteraciones para alcanzar valores semejantes de MSE que el MLP.
- El solapamiento entre clases ocasiona que los problemas de clasificación sean más difíciles de aprender para la red RBF.

Es indudable que se requiere profundizar en el tema no solo por la importancia del mismo, sino por su relación con otras áreas del reconocimiento de formas. Una de las principales líneas de investigación que permanecen abiertas es el desarrollo de estrategias que permitan una inicialización óptima de la red RBF, considerando factores como el desbalance o el solapamiento entre clases.

Apéndice B

Algoritmo back-propagation

Contenido

B.1 Introducción

El proceso de aprendizaje o entrenamiento de una ANN consiste en la estimación de sus parámetros libres. Los pesos de la red en el caso del MLP o los pesos, centros y varianzas para la red RBF.

El algoritmo back-propagation descrito formalmente en primer lugar por Werbos [Werbos 1974], posteriormente por Parker [Parker 1985], y finalmente por Rumelhart [Rumelhart 1986] es el método de aprendizaje más ampliamente utilizado en el MLP. Esta basado en una técnica de descenso por gradiente que utiliza la minimización del error cuadrático medio (Mean Square Error, MSE) mediante un proceso iterativo.

En este apéndice se presenta el desarrollo matemático seguido para obtener las reglas de actualización del algoritmo back-propagation aplicado a dos arquitecturas de red distintas: el MLP y la red RBF.

B.2 Algoritmo back-propagation (MLP)

En esta sección se describe brevemente el desarrollo matemático realizado para la obtención de las reglas de actualización del algoritmo back-propagation para un MLP

de tres capas: entrada, salida y oculta (Fig. B.1).

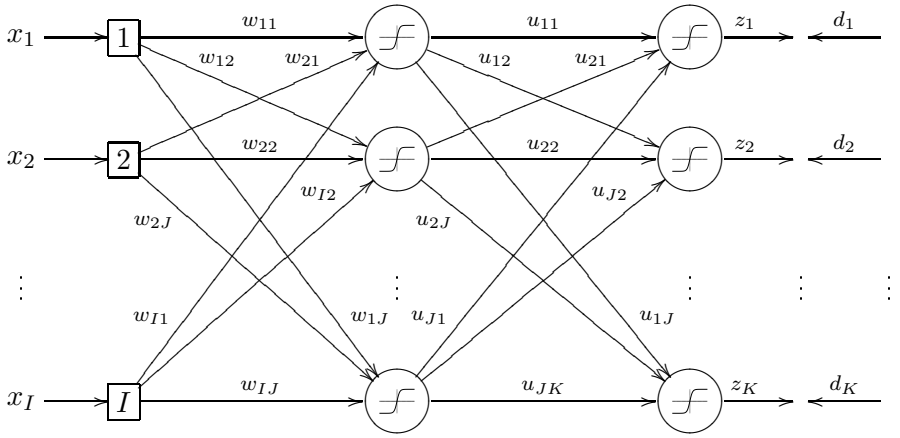


Fig. B.1: MLP de tres capas con I nodos en la entrada, J neuronas ocultas y K nodos de salida. $\mathbf{z}$ es la salida real de la red y $\mathbf{d}$ la esperada para la entrada $\mathbf{x}$. $\mathbf{W}$ y $\mathbf{U}$ son los pesos de la red para la capa oculta y la de salida respectivamente.

La nomenclatura empleada es la siguiente:

- $\mathbf{x}$: Vector de entrada o de características.
- I: Dimensión del vector $\mathbf{x}$.
- J: Número de neuronas ocultas.
- K: Número de clases en la ME.
- $\mathbf{w}$: Vector de pesos correspondientes a la capa de entrada. Observe que $\mathbf{w}$ se puede generalizar a una matriz $\mathbf{W}$ de dimensión $I \times J$.
- $\mathbf{u}$: Vector de pesos correspondientes a la capa de salida. Si se generaliza $\mathbf{u}$ se puede obtener a una matriz $\mathbf{U}$ de dimensión $J \times K$.
- N: Número de prototipos en la ME.
- z_n: Salida real de la red para el vector $\mathbf{x}_n$.
- d_n: Salida deseada de la red para el vector $\mathbf{x}_n$.
- L: Número total de capas ocultas mas 1. Para un MLP de una capa oculta L=2.

Considérese la Fig. B.1, y supóngase que el MSE puede ser descompuesto como sigue

$$E = E^{(1)} + \cdots + E^{N} \tag{B.1}$$

en el cual cada sumando es el MSE sobre el prototipo n y es definido por la Ec. B.2.

$$E^n = E^n(\mathbf{w}, \mathbf{u}) = \frac{1}{2}\sum_{k=1}^{K}(d_k^n - z_k^n)^2 \tag{B.2}$$

Las funciones de activación son definidas de acuerdo a la capa en que se encuentren:

$$\begin{array}{rcl} h(r_j) & = & 1/[1 + exp(a_1 r_j + c_1)] \\ g(s_k) & = & 1/[1 + exp(a_2 s_k + c_2)] \end{array} \tag{B.3}$$

donde a_l es el factor de corrección, c_l el sesgo o bias y

$$\begin{array}{rcl} r_j & = & \sum_{i=1}^{I} w_{ij} x_n \\ s_k & = & \sum_{j=1}^{J} u_{jk} y_j \\ & & \\ y_j & = & h(\sum_{i=1}^{I} w_{ij} x_n) \\ z_k & = & g(\sum_{j=1}^{J} u_{jk} y_j) \end{array} \tag{B.4}$$

Considerando la Ec. B.3 y la Ec. B.4, la Ec. B.2 pude ser reescrita como

$$E^n = \frac{1}{2}\sum_{k=1}^{K}[d_k - g(\sum_{j=1}^{J} u_{jk} h(\sum_{i=1}^{I} w_{ij} x_n))]^2 \tag{B.5}$$

donde por conveniencia $a_l = 1$, $c_l = c$. De esta forma las reglas de actualización de los parámetros libres de la red se formulan como sigue:

$$\begin{array}{rcl} \mathbf{w}^{t+1} & = & \mathbf{w}^t - \eta[\Delta E(\mathbf{w}^t)] \\ & = & \mathbf{w}^t + \Delta\mathbf{w} \\ & & \\ \mathbf{u}^{t+1} & = & \mathbf{u}^t - \eta[\Delta E(\mathbf{u}^t)] \\ & = & \mathbf{u}^t + \Delta\mathbf{u} \end{array} \tag{B.6}$$

o bien

$$\begin{array}{rcl} w_{ij}^{t+1} & = & w_{ij}^t - \eta[\partial E(\mathbf{w}^t, \mathbf{u}^t)/\partial w_{ij}] \\ u_{jk}^{t+1} & = & u_{jk}^t - \eta[\partial E(\mathbf{w}^t, \mathbf{u}^t)/\partial u_{jk}] \end{array}$$

A partir de B.5, se pueden calcular los valores gradiente de los parámetros libres de la ANN como se ilustra a continuación.

$$\begin{aligned}
\frac{\partial E}{\partial u_{jk}} &= \frac{\partial E}{\partial s_k}\frac{\partial s_k}{\partial u_{jk}} \\
&= [\frac{\partial}{z_k}(\frac{1}{2}\sum_{k=1}^{K}(d_k - z_k)^2)\frac{\partial}{s_k}g(s_k)][\frac{\partial s_k}{\partial u_{jk}}] \\
&= [(-1)(d_k - z_k)g'(s_k)][\frac{\partial}{\partial u_{jk}}\sum_{j=1}^{J} u_{jk}y_j] \\
&= -(d_k - z_k)g'(s_k)y_j
\end{aligned} \tag{B.7}$$

$g(s_k)$ es definida por la Ec. B.3 y $z_k = g(s_k)$, $a_2 = 1$, $c_2 = c$. Por lo tanto, la derivada de la función de activación $g(\cdot)$ se pude obtener como sigue

$$\begin{aligned}
g'(s_k) &= \frac{\partial}{\partial s_k}g(s_k) \\
&= \frac{\partial}{\partial s_k}[1 + exp(-s_k + c)]^{-1} \\
&= (-1)[1 + exp(-s_k + c)]^{-2}exp(-s_j + c)(-1) \\
&= [1 + exp(-s_k + c)]^{-2}exp(-s_j + c) \\
&= z_k^2[1/z_k - 1] \\
&= z_k^2[(1 - z_j)/z_j] \\
&= z_k(1 - z_k)
\end{aligned} \tag{B.8}$$

donde $z_k = [1 + exp(-s_k + c)]^{-1}$ y $exp(-s_k + c) = \frac{1}{z_k}$ -1. Ahora si se sustituye la Ec. B.8 en la Ec. B.7 se obtiene

$$\frac{\partial E}{\partial u_{jk}} = -(d_k - z_k)z_k(1 - z_k)y_j \tag{B.9}$$

Para obtener $\frac{\partial E}{\partial w_{ij}}$ se procede de la siguiente manera

$$\begin{aligned}
\frac{\partial E}{\partial w_{ij}} &= \frac{\partial E}{\partial r_j}\frac{\partial r_j}{\partial w_{ij}} \\
&= [(\frac{\partial E}{\partial y_j})(\frac{\partial y_j}{\partial r_j})]\frac{\partial r_j}{\partial w_{ij}} \\
&= [\frac{\partial}{y_j}(\frac{1}{2}\sum_{k=1}^{K}(d_k - z_k)^2)\cdot(\frac{\partial}{\partial r_j}y_j)][\frac{\partial r_j}{\partial w_{ij}}] \\
&= \frac{\partial}{y_j}(\frac{1}{2}\sum_{k=1}^{K}(d_k - z_k)^2)[\frac{\partial}{\partial r_j}h(r_j)]\cdot[\frac{\partial}{\partial w_{ij}}(\sum_{i=1}^{I} w_{ij}x_i)] \\
&= \frac{\partial}{y_j}(\frac{1}{2}\sum_{k=1}^{K}(d_k - z_k)^2)[h'(r_j)][x_i] \\
&= \frac{\partial}{y_j}[E(\mathrm{s}(y_j))]h'(r_j)x_i \\
&= \{\sum_{k=1}^{K}\frac{\partial E}{\partial s_k}\frac{\partial s_k}{\partial y_j}\}h'(r_j)x_i \\
&= \frac{1}{2}\{\sum_{k=1}^{K}\frac{\partial}{\partial s_k}[(d_k - z_k)^2]\cdot\frac{\partial s_k}{\partial y_j}\}h'(r_j)x_i \\
&= \{\sum_{k=1}^{K}(d_k - z_k)(-1)g'(s_k)\cdot\frac{\partial s_k}{\partial y_j}\}h'(r_j)x_i \\
&= \{\sum_{k=1}^{K}(-1)(d_k - z_k)g'(s_k)\cdot[(\frac{\partial}{\partial y_j})\sum_{j=1}^{J} y_j u_{jk}]\}h'(r_j)x_i \\
&= (-1)\{\sum_{k=1}^{K}(d_k - z_k)[z_k(1 - z_k)][u_{jk}]\}\cdot h'(r_j)x_i
\end{aligned} \tag{B.10}$$

donde $E = E(\mathrm{s}(y_j)) = E(s_1(y_j), s_1(y_j), ..., s_k(y_j))$, $z_k = g(s_k)$ y $s_k = \sum_{j=1}^{J} u_{jk} y_j$. Por analogía con la Ec. B.8: $h'(r_j) = y_j(1 - y_j)$ y de esta forma $\frac{\partial E}{\partial w_{ij}}$ queda definida como

$$\frac{\partial E}{\partial w_{ij}} = -\{\sum_{k=1}^{K}(d_k - z_k)[z_k(1 - z_k)]u_{jk}\}[y_j(1 - y_j)]x_i \tag{B.11}$$

Si se sustituye la Ec. B.9 y la Ec. B.11 en la Ec. B.6 se obtienen las reglas de actualización para un MLP con una capa oculta.

$$\begin{array}{rcl} u_{jk}^{t+1} & = & u_{jk}^{t} + \eta(d_k - z_k)z_k(1 - z_k)y_j \\ w_{ij}^{t+1} & = & w_{ij}^{t} + \eta\{\sum_{k=1}^{K}(d_k - z_k)[z_k(1 - z_k)]u_{jk}\}[y_j(1 - y_j)]x_i \end{array} \tag{B.12}$$

B.3 Algoritmo back-propagation (red RBF)

Las redes RBF al igual que el MLP pueden ser entrenadas por métodos similares de descenso por gradiente como por ejemplo el algoritmo back-propagation [Ding 2004]. Así, los parámetros $U = \{w, c, \sigma\}$ de la red RBF son obtenidos simultáneamente. A continuación se presenta una breve descripción del desarrollo matemático seguido

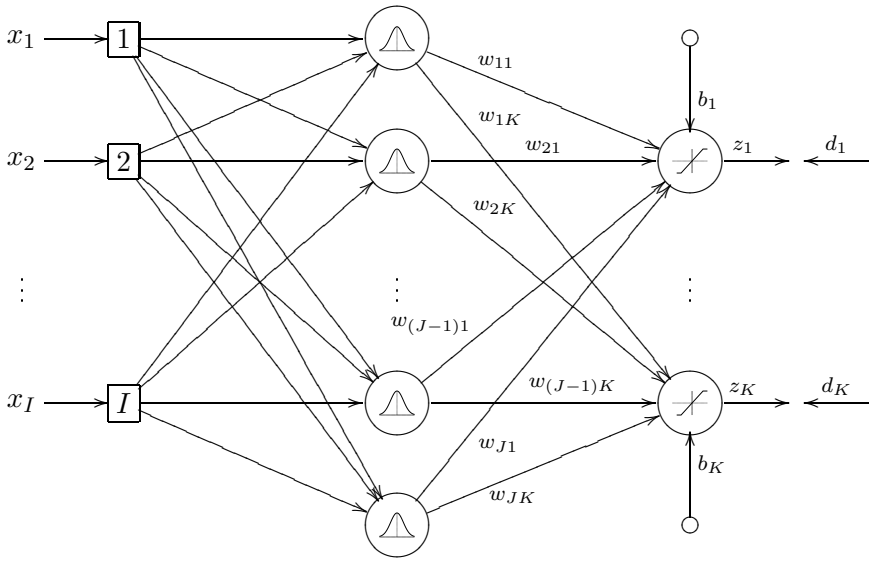


Fig. B.2: Arquitectura general de una red RBF: I nodos en la entrada, J neuronas ocultas y K nodos de salida. $\mathbf{z}_n$ es la salida real de la red y $\mathbf{d}_n$ la esperada para la entrada $\mathbf{x}_n$. w_{jk} son los pesos de la red ($k = 1...K$; $j = 1...J$).

para la obtención de las reglas de actualización del algoritmo back-propagation para la red RBF de la Fig. B.2. La nomenclatura utilizada en esta sección fue la siguiente:

- $\mathbf{x}$: Vector de entrada o de características.
- I: Dimensión del vector $\mathbf{x}$.
- J: Número de neuronas ocultas.
- K: Número de clases en la ME.
- $\mathbf{w}$: Vector de pesos correspondientes a la capa oculta. Observe que $\mathbf{w}$ se puede generalizar a una matriz $\mathbf{W}$ de dimensión $J \times K$.
- N: Número de prototipos en la ME.
- z_n: Salida real de la red para el vector $\mathbf{x}_n$.
- d_n: Salida deseada de la red para el vector $\mathbf{x}_n$.

Supóngase que el MSE puede ser descompuesto como sigue

$$E = E^{(1)} + \cdots + E^N \tag{B.13}$$

en el cual cada sumando es el MSE sobre el prototipo n y es definido por la Ec. B.14.

$$E^n = E^n(\mathbf{w}, \mathbf{d}, \sigma) = \sum_{k=1}^{K} \frac{1}{2}(d_k^n - z_k^n)^2 \tag{B.14}$$

donde z_k queda expresado como

$$z_k(\mathbf{x}_n) = \sum_{j=1}^{J} w_{jk} h_j(u) + h_0(u) w_0 = \sum_{j=0}^{J} w_{jk} h_j(u)\,. \tag{B.15}$$

w_0 es el bias o sesgo. Por cuestiones de notación se define $h_0(u) = 1$ y

$$\begin{array}{lcl} h_j(u) & = & exp(u) \\ u & = & -\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2\sigma^2} \end{array} \tag{B.16}$$

para $j = 1, ..., J$.

Las reglas de actualización para el algoritmo back-propagation en el contexto de la red RBF se formulan de la siguiente manera

$$\begin{aligned}
\mathbf{w}^{t+1} &= \mathbf{w}^t - \eta[\Delta E(\mathbf{w}^t)] \\
&= \mathbf{w}^t + \Delta\mathbf{w} \\
\\
\mathbf{c}^{t+1} &= \mathbf{c}^t - \eta[\Delta E(\mathbf{c}^t)] \\
&= \mathbf{c}^t + \Delta\mathbf{c} \\
\\
\sigma^{t+1} &= \sigma^t - \eta[\Delta E(\sigma^t)] \\
&= \sigma^t + \Delta\sigma
\end{aligned} \tag{B.17}$$

o bien

$$\begin{aligned}
w_{jk}^{t+1} &= w_{jk}^t - \eta[\partial E(\mathbf{w}^t, \mathbf{c}^t, \sigma^t)/\partial w_{jk}] \\
c_{ji}^{t+1} &= c_{ji}^t - \eta[\partial E(\mathbf{w}^t, \mathbf{c}^t, \sigma^t)/\partial c_{ji}] \\
\sigma_j^{t+1} &= \sigma_j^t - \eta[\partial E(\mathbf{w}^t, c^t, \sigma^t)/\partial \sigma_j]
\end{aligned}$$

Si se considera la Ec. B.16, la Ec. B.17 se puede resolver como sigue:

$$\begin{aligned}
\frac{\partial E}{\partial w_{jk}} &= \frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial w_{jk}} \\
&= \frac{\partial}{\partial z_k}(\frac{1}{2}(d_k - z_k)^2)\frac{\partial z_k}{\partial w_{jk}} \\
&= (-1)(d_k - z_k)\frac{\partial}{\partial w_{jk}}(w_{jk}h_j(u)) \\
&= -(d_k - z_k)h_j(u)
\end{aligned} \tag{B.18}$$

Para los centros de la red RBF se procede de esta otra forma:

$$\begin{aligned}
\frac{\partial E}{\partial c_{ji}} &= [\frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial h_j}]\frac{\partial h_j}{\partial u}\frac{\partial u}{\partial c_{ji}} \\
&= [\textstyle\sum_{k=1}^{K} -(d_k - z_k)\frac{\partial}{\partial h_j}(w_{jk}h_j(u))]\frac{\partial u}{c_{ji}} \\
&= [\textstyle\sum_{k=1}^{K}(-1)(d_k - z_k)w_{jk}]\frac{\partial}{\partial u}\{exp(u_j)\}\frac{\partial u}{c_{ji}} \\
&= [\textstyle\sum_{k=1}^{K} -(d_k - z_k)w_{jk}] \cdot exp(-\frac{\|\mathbf{x}-\mathbf{c}_j\|}{2\sigma_j^2})\frac{\partial u}{c_{ji}} \\
&= [\textstyle\sum_{k=1}^{K} -(d_k - z_k)w_{jk}] \cdot exp(-\frac{\|\mathbf{x}-\mathbf{c}_j\|}{2\sigma_j^2})\frac{\partial}{c_{ji}}(-\frac{\sum_{i=1}^{I}(x_i - c_{ji})^2}{2\sigma_j^2}) \\
&= [\textstyle\sum_{k=1}^{K} -(d_k - z_k)w_{jk}] \cdot exp(-\frac{\|\mathbf{x}-\mathbf{c}_j\|}{2\sigma_j^2})(-1)(-2)\frac{(x_i - c_{ji})}{2\sigma_j^2} \\
&= -\frac{1}{\sigma_j^2}[\textstyle\sum_{k=1}^{K}(d_k - z_k)w_{jk}] \cdot h_j(\|\mathbf{x} - \mathbf{c}_j\|)(x_i - c_{ji})
\end{aligned} \tag{B.19}$$

El valor de $\frac{\partial E}{\partial \sigma_j}$ es obtenido como sigue:

$$\begin{aligned}
\frac{\partial E}{\partial \sigma_j} &= [\frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial h_j}]\frac{\partial h_j}{\partial u}\frac{\partial u}{\partial \sigma_j} \\
&= [\sum_{k=1}^{K} -(d_k - z_k)w_{jk}]h_j(\|\mathbf{x}-\mathbf{c}_j\|)\frac{\partial}{\partial \sigma_j}(-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2\sigma^2}) \\
&= [\sum_{k=1}^{K} -(d_k - z_k)w_{jk}]h_j(\|\mathbf{x}-\mathbf{c}_j\|)[-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2}]\frac{\partial}{\partial \sigma_j}(\sigma^{-2}) \\
&= [\sum_{k=1}^{K} -(d_k - z_k)w_{jk}]h_j(\|\mathbf{x}-\mathbf{c}_j\|)(-1)(-2)\frac{(\|\mathbf{x}-\mathbf{c}_j\|^2)}{2\sigma_j^3} \\
&= -\frac{1}{\sigma_j^3}[\sum_{k=1}^{K}(d_k - z_k)w_{jk}]h_j(\|\mathbf{x}-\mathbf{c}_j\|)\|\mathbf{x}-\mathbf{c}_j\|^2
\end{aligned} \tag{B.20}$$

Así las reglas de actualización para el algoritmo back-propagation en el contexto de la red RBF se pueden definir de la siguiente manera:

$$w_{jk}^{t+1} = w_{jk}^{t} + \eta\{\sum_{n=1}^{N} h_j(\|\mathbf{x}^n - \mathbf{c}_j\|)(d_k^n - f_k^n)\}\,, \tag{B.21}$$

$$c_{ji}^{t+1} = c_{ji}^{t} + \eta\{\frac{1}{\sigma_j^2}\sum_{n=1}^{N}\left[\sum_{k=1}^{K}(d_k^n - z_k^n)w_{jk}\right] h_j(\|\mathbf{x}^n - \mathbf{c}_j\|)(x^n - c_{ji})\}\,, \tag{B.22}$$

$$\sigma_j^{t+1} = \sigma_j^{t+1} + \eta\{\frac{1}{\sigma_j^3}\sum_{n=1}^{N}\sum_{k=1}^{K}(d_k^n - f_k^n)w_{jk}h_j(\|\mathbf{x}^n - \mathbf{c}_j\|)\|\mathbf{x}^n - \mathbf{c}_j\|^2\}\,. \tag{B.23}$$

Apéndice C

Otras bases de datos de dos y múltiples clases

Este apéndice esta dedicado a presentar algunos resultados que fueron excluidos en capítulos anteriores. Se organiza en dos apartados: Redes no modulares (sec. C.1) y modulares (sec. C.2). Los resultados corresponden a conjuntos de datos de dos y múltiples clases. El objetivo que se persigue al presentar esta información es dar mayor soporte a las conclusiones presentadas en los capítulos 3 y 4.

C.1 Clasificadores Globales

En el capítulo 3 se observó que la inclusión de funciones de coste al proceso de aprendizaje de la RNA[1] ayuda a acelerar la convergencia de las clases menos representadas en la ME, lo que se traduce en incrementos sustanciales en la PC de las clases minoritarias.

No obstante, en algunas situaciones la inclusión de funciones de coste ocasiona perdida en la efectividad del clasificador. En estos casos se observa una tendencia a dar prioridad a las clases menos representadas en el proceso de aprendizaje, y a disminuir la participación de las clases mayoritarias.

Los resultados presentados en la Tabla C.1 vienen a confirman lo anterior, en el sentido de que en la mayoría de los problemas se presenta un incremento de la efectividad del clasificador. Obsérvese el aumento en el valor de la *g-mean* después de aplicar las funciones de coste. Sin embargo, en algunos casos se observan reducciones importantes en la PC a causa de la perdida de la efectividad del clasificador sobre la clase mayoritaria (por ejemplo, vea los resultados de la base de datos Diabetes).

[1] Entrenada con el algoritmo back-propagation con procesamiento por grupos.

En el caso particular de los problemas de múltiples clases Feltwell y Satimage, la tendencia a la perdida de efectividad del clasificador por el uso de las funciones de coste, no se observa de manera contundente. En otras palabras, se tienen valores de PC y *g-mean* de igual o mayor magnitud antes y después de aplicar las funciones de coste.

En las Tablas C.2, C.3, C.4, C.5, C.6 y C.7 se presenta los valores de PC por clase de los conjuntos de datos Feltwell y Satimage.

En los tres modelos de ANNs la aplicación de las funciones de coste incrementó la PC de las clases minoritarias. Así mismo, se incrementó la confusión de algunas de las clases consideradas mayoritarias. Por ejemplo, observe la clase 2 para Feltwell y la clase 1 para Satimage. En general, estos resultados sólo vienen a confirmar lo ya sugerido previamente.

Nótese que el comportamiento del clasificador sobre estos conjuntos de datos está en un punto entre el comportamiento observado en las bases de datos V2Cls y B2Cls (para el caso de dos clases), y en el observado en los conjuntos Ecoli6 y Cayo (para el caso de múltiples clases), es decir, que no corresponde directamente a cada uno de los casos discutidos en el capítulo 3, sino que se encuentran o más alejados o más cercanos a cada una de las bases de datos prototipo.

Tabla C.1: Resultados de la fase de clasificación de la red neuronal (en sus tres versiones) con el contexto de funciones de costo. Los valores entre paréntesis hacen referencia a la desviación estándar.

	MLP		RBF		RBF+VF	
Opción 0	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cancer	96.57(1.94)	96.44(2.37)	96.00(2.53)	96.16(2.88)	96.72(1.80)	96.71(2.10)
Diabetes	69.91(4.91)	71.51(4.75)	67.36(6.71)	68.36(6.13)	70.23(6.22)	71.60(6.31)
German	74.16(3.96)	67.09(6.03)	73.35(2.95)	49.41(12.06)	75.94(3.09)	66.93(6.29)
Ionosphere	90.94(4.51)	87.59(8.48)	87.53(5.63)	88.20(5.43)	87.73(4.62)	83.89(6.20)
Liver	68.22(8.73)	61.94(12.16)	65.55(7.88)	56.99(11.63)	66.90(7.53)	57.72(11.69)
Sonar	73.08(11.70)	64.85(18.03)	61.57(8.43)	49.17(18.60)	71.92(9.06)	63.52(13.74)
Feltwell	89.38(0.95)	86.60(1.64)	84.36(2.03)	62.21(25.96)	86.15(1.90)	70.78(25.85)
Satimage	82.26(0.31)	47.31(5.72)	83.28(0.94)	77.44(1.46)	82.33(1.24)	71.96(7.04)
Opción 1	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cancer	96.50(1.90)	96.70(1.79)	95.50(2.66)	95.84(2.67)	95.87(6.31)	95.92(7.57)
Diabetes	62.00(4.48)	64.37(4.89)	56.87(7.49)	58.17(7.94)	62.10(6.12)	64.36(6.37)
German	67.52(4.62)	68.00(4.11)	68.45(4.22)	69.74(4.03)	69.43(4.03)	70.81(3.99)
Ionosphere	91.02(4.81)	88.04(8.70)	85.53(5.31)	87.14(4.69)	87.39(5.51)	84.21(8.00)
Liver	68.48(10.23)	64.54(12.72)	65.92(10.00)	63.58(11.04)	66.47(9.24)	62.19(11.03)
Sonar	71.84(10.47)	63.18(17.29)	61.90(8.08)	53.21(12.66)	72.75(8.35)	64.58(13.13)
Feltwell	88.93(0.66)	87.06(0.73)	85.02(1.31)	83.16(2.06)	88.41(1.15)	86.38(1.35)
Satimage	85.49(0.41)	84.45(0.50)	83.38(0.99)	81.88(1.18)	83.78(0.63)	81.80(0.75)
Opción 2	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cancer	96.57(1.83)	96.85(1.90)	96.11(2.52)	96.40(2.53)	96.08(6.02)	96.25(6.62)
Diabetes	61.65(4.92)	63.96(5.34)	57.19(6.63)	58.63(7.09)	61.68(6.79)	63.93(7.12)
German	68.05(4.57)	67.26(4.32)	68.73(4.13)	70.21(3.89)	69.12(4.05)	70.39(3.89)
Ionosphere	90.33(4.89)	86.90(9.59)	86.19(5.63)	87.46(5.32)	87.91(5.46)	84.43(8.72)
Liver	66.48(10.47)	62.14(14.07)	67.27(8.86)	64.18(11.01)	66.27(9.31)	62.52(11.20)
Sonar	73.35(10.67)	65.63(16.34)	64.08(8.56)	53.76(15.54)	70.97(9.63)	62.07(14.64)
Feltwell	88.59(1.28)	86.24(2.10)	87.74(1.42)	86.32(1.44)	87.70(1.22)	85.69(1.11)
Satimage	87.48(0.71)	86.18(0.92)	85.18(1.16)	84.09(1.51)	84.76(0.58)	83.23(0.78)
Opción 3	PC	*g-mean*	PC	*g-mean*	PC	*g-mean*
Cancer	96.63(1.78)	96.46(2.16)	96.08(2.55)	96.19(2.92)	96.71(1.86)	96.62(2.23)
Diabetes	65.10(4.41)	67.34(4.82)	62.05(7.61)	64.13(7.59)	65.89(7.05)	67.94(6.79)
German	71.43(4.39)	68.96(4.19)	73.49(3.91)	68.41(5.14)	72.94(2.57)	71.35(3.41)
Ionosphere	90.40(4.75)	87.13(8.49)	87.39(5.34)	88.19(5.24)	87.85(5.13)	84.42(7.64)
Liver	68.41(9.62)	63.97(12.44)	65.88(8.41)	61.51(11.28)	66.79(8.89)	61.66(11.70)
Sonar	72.88(10.48)	64.71(17.81)	62.41(8.18)	51.69(13.98)	72.23(8.54)	64.01(13.29)
Feltwell	89.01(0.51)	86.79(0.75)	86.51(1.94)	82.41(4.41)	88.93(1.29)	85.60(2.86)
Satimage	86.07(0.34)	83.34(0.95)	83.16(1.30)	80.95(1.20)	84.48(0.77)	81.65(0.87)

Tabla C.2: Feltwell: Resultados obtenidos en la fase de clasificación con el MLP (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	6.06	0.35	99.07	
	cls 2	7.80	0.24	81.97	cls 3 (11.72)
	cls 3	4.78	0.10	78.86	cls 1 (10.70)
	cls 4	7.99	0.15	83.91	cls 1 (11.48)
	cls 5	3.03	0.17	90.43	
Opción 1	cls 1	6.06	0.35	98.59	
	cls 2	7.80	0.24	77.92	cls 3 (19.26)
	cls 3	4.78	0.10	84.90	
	cls 4	7.99	0.15	83.90	
	cls 5	3.03	0.17	91.05	
Opción 2	cls 1	6.06	0.35	98.24	
	cls 2	7.80	0.24	79.02	cls 3 (18.41)
	cls 3	4.78	0.10	81.26	
	cls 4	7.99	0.15	86.00	
	cls 5	3.03	0.17	88.34	
Opción 3	cls 1	6.06	0.35	98.58	
	cls 2	7.80	0.24	80.85	cls 3 (14.85)
	cls 3	4.78	0.10	83.08	
	cls 4	7.99	0.15	83.35	cls 1 (10.74)
	cls 5	3.03	0.17	88.92	cls 1 (10.55)

Tabla C.3: Feltwell: Resultados obtenidos en la fase de clasificación con la red RBF (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
	cls 1	6.06	0.35	96.49	
	cls 2	7.80	0.24	89.58	
Opción 0	cls 3	4.78	0.10	32.90	cls 2 (37.57) cls 4 (12.85) cls 5 (13.50)
	cls 4	7.99	0.15	83.94	cls 1 (10.19)
	cls 5	3.03	0.17	81.27	cls 1 (12.19)
	cls 1	6.06	0.35	93.79	
	cls 2	7.80	0.24	73.57	cls 3 (24.19)
Opción 1	cls 3	4.78	0.10	79.46	cls 4 (13.08)
	cls 4	7.99	0.15	80.97	
	cls 5	3.03	0.17	89.63	
	cls 1	6.06	0.35	96.51	
	cls 2	7.80	0.24	74.76	cls 3 (22.45)
Opción 2	cls 3	4.78	0.10	84.45	cls 4 (12.94)
	cls 4	7.99	0.15	85.74	
	cls 5	3.03	0.17	91.37	
	cls 1	6.06	0.35	94.93	
	cls 2	7.80	0.24	79.46	cls 3 (15.51)
Opción 3	cls 3	4.78	0.10	65.98	cls 2 (15.05) cls 4 (12.76)
	cls 4	7.99	0.15	83.83	
	cls 5	3.03	0.17	92.94	

Tabla C.4: Feltwell: Resultados obtenidos en la fase de clasificación con la red RBF+VF (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	6.062	0.35	99.26	
	cls 2	7.805	0.24	86.37	
	cls 3	4.78	0.10	53.95	cls 2 (14.52) cls 4 (11.32) cls 5 (15.41)
	cls 4	7.999	0.15	83.88	cls 1 (11.89)
	cls 5	3.038	0.17	78.71	cls 1 (19.32)
Opción 1	cls 1	6.062	0.35	98.55	
	cls 2	7.805	0.24	79.17	cls 3 (18.94)
	cls 3	4.78	0.10	85.01	cls 4 (12.09)
	cls 4	7.999	0.15	84.73	
	cls 5	3.038	0.17	85.40	cls 1 (12.60)
Opción 2	cls 1	6.062	0.35	97.70	
	cls 2	7.805	0.24	79.58	cls 3 (18.09)
	cls 3	4.78	0.10	84.74	cls 4 (11.24)
	cls 4	7.999	0.15	85.84	
	cls 5	3.038	0.17	81.49	cls 1 (16.56)
Opción 3	cls 1	6.062	0.35	98.57	
	cls 2	7.805	0.24	85.12	cls 3 (12.42)
	cls 3	4.78	0.10	77.12	cls 4 (11.59)
	cls 4	7.999	0.15	83.37	cls 1 (11.23)
	cls 5	3.038	0.17	85.84	cls 1 (12.90)

Tabla C.5: Satimage: Resultados obtenidos en la fase de clasificación con el MLP (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	5.629	0.23	90.87	
	cls 2	6.758	0.23	98.83	
	cls 3	17.493	0.11	90.71	
	cls 4	13.707	0.20	97.71	
	cls 5	6.198	0.11	2.37	cls 1 (61.04) cls 4 (33.03)
	cls 6	6.532	0.12	70.25	cls 1 (15.74)
Opción 1	cls 1	5.629	0.23	73.49	cls 5 (20.77)
	cls 2	6.758	0.23	97.98	
	cls 3	17.493	0.11	93.26	
	cls 4	13.707	0.20	89.92	
	cls 5	6.198	0.11	75.69	cls 1 (10.76) cls 4 (12.09)
	cls 6	6.532	0.12	78.95	
Opción 2	cls 1	5.629	0.23	78.98	cls 5 (15.81)
	cls 2	6.758	0.23	98.22	
	cls 3	17.493	0.11	97.50	
	cls 4	13.707	0.20	90.60	
	cls 5	6.198	0.11	72.37	cls 1 (13.74) cls 4 (12.51)
	cls 6	6.532	0.12	82.24	
Opción 3	cls 1	5.629	0.23	81.89	cls 5 (13.83)
	cls 2	6.758	0.23	97.51	
	cls 3	17.493	0.11	90.54	
	cls 4	13.707	0.20	91.61	
	cls 5	6.198	0.11	65.73	cls 1 (19.91) cls 4 (13.22)
	cls 6	6.532	0.12	76.71	cls 1 (13.50)

Tabla C.6: Satimage: Resultados obtenidos en la fase de clasificación con la red RBF (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
	cls 1	5.629	0.23	79.21	cls 5 (11.45)
	cls 2	6.758	0.23	97.64	
	cls 3	17.493	0.11	92.81	
Opción 0	cls 4	13.707	0.20	95.89	
	cls 5	6.198	0.11	49.34	cls 1 (16.92) cls 4 (29.29)
	cls 6	6.532	0.12	63.50	cls 1 (14.01) cls 2 (13.88)
	cls 1	5.629	0.23	68.28	cls 5 (22.62)
	cls 2	6.758	0.23	95.88	
	cls 3	17.493	0.11	95.22	
Opción 1	cls 4	13.707	0.20	92.37	
	cls 5	6.198	0.11	69.19	cls 4 (18.44)
	cls 6	6.532	0.12	75.36	
	cls 1	5.629	0.23	69.96	cls 5 (21.02)
	cls 2	6.758	0.23	97.16	
	cls 3	17.493	0.11	96.29	
Opción 2	cls 4	13.707	0.20	92.44	
	cls 5	6.198	0.11	70.66	cls 4 (17.49)
	cls 6	6.532	0.12	82.36	
	cls 1	5.629	0.23	73.45	cls 5 (20.06)
	cls 2	6.758	0.23	95.81	
	cls 3	17.493	0.11	90.31	
Opción 3	cls 4	13.707	0.20	91.26	
	cls 5	6.198	0.11	69.10	cls 1 (10.90) cls 4 (16.68)
	cls 6	6.532	0.12	70.00	

Tabla C.7: Satimage: Resultados obtenidos en la fase de clasificación con la red RBF+VF (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
	cls 1	5.629	0.23	83.00	
	cls 2	6.758	0.23	98.42	
	cls 3	17.493	0.11	90.62	
Opción 0	cls 4	13.707	0.20	97.10	
	cls 5	6.198	0.11	36.73	cls 1 (25.83) cls 4 (33.98)
	cls 6	6.532	0.12	57.68	cls 1 (21.73) cls 2 (12.19)
	cls 1	5.629	0.23	68.77	cls 5 (22.00)
	cls 2	6.758	0.23	98.13	
	cls 3	17.493	0.11	93.79	
Opción 1	cls 4	13.707	0.20	92.77	
	cls 5	6.198	0.11	67.54	cls 4 (20.09)
	cls 6	6.532	0.12	75.61	
	cls 1	5.629	0.23	70.15	cls 5 (21.72)
	cls 2	6.758	0.23	98.48	
	cls 3	17.493	0.11	92.54	
Opción 2	cls 4	13.707	0.20	92.77	
	cls 5	6.198	0.11	72.09	cls 4 (16.30)
	cls 6	6.532	0.12	77.55	
	cls 1	5.629	0.23	76.12	cls 5 (16.88)
	cls 2	6.758	0.23	97.69	
	cls 3	17.493	0.11	91.07	
Opción 3	cls 4	13.707	0.20	93.23	
	cls 5	6.198	0.11	65.51	cls 1 (12.27) cls 4 (18.69)
	cls 6	6.532	0.12	71.40	cls 1 (12.56)

C.2 Clasificadores Modulares (votación simple)

En esta sección se presentan a detalle algunos resultados obtenidos al clasificar con las bases de datos Feltwell y Satimage en el contexto de las redes neuronales modulares. Estos resultados amplían la información discutida previamente en el capítulo 4.

Se observa en las Tablas C.8, C.9 C.10 correspondientes a la base datos Feltwell que al aplicar la ANN-M sobre MLP y RBF+VF, el incremento del desbalance de las clases no afecta la efectividad del clasificador. Esto es debido a que la simplificación del problema permite que pueda ser aprendido más fácilmente.

En el caso de la ANN-M sobre RBF, la clase menos representada se ve afectada por el aumento del desbalance de las clases. Así, este tipo de modelos son más sensibles a este problema. No obstante, al aplicar las funciones de coste este problema se reduce.

También se observa que la inclusión de funciones de coste tiene efectos negativos, como el de incrementar la confusión de las clases como consecuencia del aumento y decremento de la influencia de las clases menos y más representadas (respectivamente) en el proceso de entrenamiento.

El problema real de esta situación es que cuando se tienen bases de datos muy desbalanceadas la clase mayoritaria es aprendida en menor medida (como efecto de la función de coste), y esto ocasiona que a la hora de integrar las respuestas de las distintas clases se tengan más errores de clasificación.

Por otra parte, en el caso de la base de datos Satimage (Tablas C.11, C.12 y C.13) se observa que una de las clases menos representada (clase 5) muestra un bajo nivel de PC, lo que podría llevar a sugerir que la acentuación del desbalance de las clases afecta en gran medida al clasificador. Sin embargo, la clase 3 está igual de representada y la efectividad del clasificador sobre esta clase es superior al 90%.

Este hecho viene a reafirmar la idea de que la bajada de PC en la clase 5 no es ocasionado únicamente por el desbalance, sino por su combinación con otras cuestiones como el solapamiento. Al aplicar las funciones de costo, la efectividad del clasificador sobre esta clase se ve incrementada, aunque no supera en el mejor de los casos el 73% en la clasificación.

El comportamiento del clasificador sobre las bases de datos Feltwell y Satimage se encuentra en un punto intermedio entre el mostrado sobre los conjuntos de datos Ecoli6 y Cayo (ver capítulo 4).

Tabla C.8: Feltwell: Resultados obtenidos en la fase de clasificación con el MLP (Modular).

	Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	1.256	0.35	99.53	
	cls 2	3.019	0.24	85.70	cls 3 (11.72)
	cls 3	3.058	0.10	82.77	
	cls 4	3.004	0.15	83.95	cls 1 (11.22)
	cls 5	0.604	0.17	83.43	cls 1 (16.00)
Opción 1	cls 1	1.256	0.35	99.23	
	cls 2	3.019	0.24	77.32	cls 3 (20.65)
	cls 3	3.058	0.10	85.46	cls 4 (10.67)
	cls 4	3.004	0.15	82.17	cls 1 (11.69)
	cls 5	0.604	0.17	77.01	cls 1 (20.20)
Opción 2	cls 1	1.256	0.35	99.23	
	cls 2	3.019	0.24	77.34	cls 3 (20.37)
	cls 3	3.058	0.10	85.39	
	cls 4	3.004	0.15	80.92	cls 1 (13.02)
	cls 5	0.604	0.17	76.04	cls 1 (20.72)
Opción 3	cls 1	1.256	0.35	99.49	
	cls 2	3.019	0.24	83.56	cls 3 (13.62)
	cls 3	3.058	0.10	85.06	
	cls 4	3.004	0.15	81.53	cls 1 (12.85)
	cls 5	0.604	0.17	79.69	cls 1 (19.34)

Tabla C.9: Feltwell: Resultados obtenidos en la fase de clasificación con la red RBF (Modular).

	Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	1.256	0.35	97.92	
	cls 2	3.019	0.24	85.94	cls 3 (10.69)
	cls 3	3.058	0.10	62.23	cls 4 (12.54) cls 5 (11.06)
	cls 4	3.004	0.15	82.96	cls 1 (12.38)
	cls 5	0.604	0.17	83.41	cls 1 (14.17)
Opción 1	cls 1	1.256	0.35	97.49	
	cls 2	3.019	0.24	80.54	cls 3 (16.74)
	cls 3	3.058	0.10	84.58	cls 4 (11.93)
	cls 4	3.004	0.15	82.85	cls 1 (11.02)
	cls 5	0.604	0.17	88.47	
Opción 2	cls 1	1.256	0.35	97.84	
	cls 2	3.019	0.24	79.34	cls 3 (18.69)
	cls 3	3.058	0.10	85.39	cls 4 (12.07)
	cls 4	3.004	0.15	83.70	cls 1 (11.12)
	cls 5	0.604	0.17	86.46	
Opción 3	cls 1	1.256	0.35	98.53	
	cls 2	3.019	0.24	84.09	cls 3 (13.46)
	cls 3	3.058	0.10	84.18	cls 4 (11.69)
	cls 4	3.004	0.15	82.75	cls 1 (11.95)
	cls 5	0.604	0.17	90.86	

Tabla C.10: Feltwell: Resultados obtenidos en la fase de clasificación con la red RBF+VF (Modular).

	Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	1.256	0.35	99.50	
	cls 2	3.019	0.24	86.78	cls 3 (11.39)
	cls 3	3.058	0.10	77.15	cls 4 (11.14)
	cls 4	3.004	0.15	82.26	cls 1 (13.43)
	cls 5	0.604	0.17	79.44	cls 1 (19.21)
Opción 1	cls 1	1.256	0.35	99.15	
	cls 2	3.019	0.24	84.03	cls 3 (13.70)
	cls 3	3.058	0.10	84.09	
	cls 4	3.004	0.15	73.00	cls 1 (21.43)
	cls 5	0.604	0.17	86.03	cls 1 (12.59)
Opción 2	cls 1	1.256	0.35	99.16	
	cls 2	3.019	0.24	85.08	cls 3 (12.55)
	cls 3	3.058	0.10	84.07	
	cls 4	3.004	0.15	71.70	cls 1 (22.34)
	cls 5	0.604	0.17	86.86	cls 1 (11.52)
Opción 3	cls 1	1.256	0.35	99.11	
	cls 2	3.019	0.24	86.16	cls 3 (11.83)
	cls 3	3.058	0.10	83.62	cls 4 (10.43)
	cls 4	3.004	0.15	79.14	cls 1 (15.88)
	cls 5	0.604	0.17	86.11	cls 1 (13.01)

Tabla C.11: Satimage: Resultados obtenidos en la fase de clasificación con el MLP (Modular).

	Clase	F1	Razón	PC	% confusión (> 10 %)
	cls 1	1.805	0.23	90.64	
	cls 2	0.317	0.23	98.61	
	cls 3	4.894	0.11	94.42	
Opción 0	cls 4	4.090	0.20	96.47	
	cls 5	0.375	0.11	2.89	cls 1 (65.21) cls 4 (28.58)
	cls 6	0.872	0.12	64.51	cls 1 (15.57)
	cls 1	1.805	0.23	75.28	cls 5 (19.94)
	cls 2	0.317	0.23	98.59	
	cls 3	4.894	0.11	92.28	
Opción 1	cls 4	4.090	0.20	90.05	
	cls 5	0.375	0.11	71.47	cls 1 (13.13) cls 4 (13.08)
	cls 6	0.872	0.12	73.54	cls 1 (14.39)
	cls 1	1.805	0.23	77.57	cls 5 (17.85)
	cls 2	0.317	0.23	98.61	
	cls 3	4.894	0.11	92.14	
Opción 2	cls 4	4.090	0.20	90.23	
	cls 5	0.375	0.11	70.57	cls 1 (15.17) cls 4 (11.99)
	cls 6	0.872	0.12	77.17	cls 1 (12.24)
	cls 1	1.805	0.23	79.06	cls 5 (16.15)
	cls 2	0.317	0.23	98.63	
	cls 3	4.894	0.11	94.06	
Opción 3	cls 4	4.090	0.20	92.29	
	cls 5	0.375	0.11	62.37	cls 1 (21.42) cls 4 (14.41)
	cls 6	0.872	0.12	70.97	cls 1 (12.74)

Tabla C.12: Satimage: Resultados obtenidos en la fase de clasificación con la red RBF (Modular).

	Clase	F1	Razón	PC	% confusión (> 10 %)
	cls 1	1.805	0.23	84.98	
	cls 2	0.317	0.23	98.63	
	cls 3	4.894	0.11	94.69	
Opción 0	cls 4	4.090	0.20	95.94	
	cls 5	0.375	0.11	37.96	cls 1 (26.59) cls 4 (31.56)
	cls 6	0.872	0.12	61.39	cls 1 (18.23) cls 2 (14.09)
	cls 1	1.805	0.23	77.57	cls 5 (16.21)
	cls 2	0.317	0.23	97.38	
	cls 3	4.894	0.11	95.31	
Opción 1	cls 4	4.090	0.20	92.62	
	cls 5	0.375	0.11	67.91	cls 1 (13.89) cls 4 (15.55)
	cls 6	0.872	0.12	78.02	
	cls 1	1.805	0.23	76.09	cls 5 (17.89)
	cls 2	0.317	0.23	97.53	
	cls 3	4.894	0.11	94.69	
Opción 2	cls 4	4.090	0.20	92.32	
	cls 5	0.375	0.11	71.28	cls 1 (11.99) cls 4 (14.27)
	cls 6	0.872	0.12	78.23	
	cls 1	1.805	0.23	78.96	cls 5 (14.36)
	cls 2	0.317	0.23	98.31	
	cls 3	4.894	0.11	94.51	
Opción 3	cls 4	4.090	0.20	93.88	
	cls 5	0.375	0.11	61.33	cls 1 (16.35) cls 4 (19.00)
	cls 6	0.872	0.12	72.95	

Tabla C.13: Satimage: Resultados obtenidos en la fase de clasificación con la red RBF+VF (Modular).

	Clase	F1	Razón	PC	% confusión (> 10 %)
	cls 1	1.805	0.23	85.13	
	cls 2	0.317	0.23	98.79	
	cls 3	4.894	0.11	92.63	
Opción 0	cls 4	4.090	0.20	98.09	
	cls 5	0.375	0.11	42.80	cls 1 (24.17) cls 4 (30.85)
	cls 6	0.872	0.12	65.11	cls 1 (19.66)
	cls 1	1.805	0.23	74.79	cls 5 (20.04)
	cls 2	0.317	0.23	98.66	
	cls 3	4.894	0.11	96.16	
Opción 1	cls 4	4.090	0.20	92.49	
	cls 5	0.375	0.11	73.18	cls 1 (11.14) cls 4 (14.03)
	cls 6	0.872	0.12	74.85	
	cls 1	1.805	0.23	74.47	cls 5 (19.49)
	cls 2	0.317	0.23	98.57	
	cls 3	4.894	0.11	95.13	
Opción 2	cls 4	4.090	0.20	92.77	
	cls 5	0.375	0.11	73.22	cls 1 (11.37) cls 4 (13.13)
	cls 6	0.872	0.12	77.00	
	cls 1	1.805	0.23	79.87	cls 5 (13.96)
	cls 2	0.317	0.23	98.76	
	cls 3	4.894	0.11	95.94	
Opción 3	cls 4	4.090	0.20	94.08	
	cls 5	0.375	0.11	58.01	cls 1 (19.38) cls 4 (20.24)
	cls 6	0.872	0.12	73.12	cls 1 (11.22)

C.3 Corrección de los datos

En esta sección se presentan los resultados obtenidos al evaluar la estrategia de Corrección de los datos (presentada en el capítulo 5) sobre las bases de datos de múltiples clases Feltwell y Satimage.

Los resultados contenidos en las Tablas C.14, C.15, C.16, C.17, C.18, C.19, C.20 y C.21, confirman las conclusiones presentadas en el capítulo 5, i.e., evidencian la efectividad de la estrategia de edición para tratar problemas desbalanceados de multiples clases.

Obsérvese que la edición de las clases mayoritarias ayuda a reducir la zona de confusión de las clases minoritarias.

Estos resultados muestran que entrenar la ANN con estrategias de edición genera mejores resultados que los obtenidos con la ME original, en el sentido de que la PC de las clases minoritarias es incrementada, traduciendoce en un aumento de sus valores de *g-mean*. No obstante, se observa una reducción en los valores globales de PC. Esto es ocasionado por la eliminación de muestras de las clases mayoritarias. En la Tabla C.22 se presenta el número de muestras antes y después de editar la ME.

Por otro lado, se observa que la combinación de editar e incluir funciones de coste al proceso de entrenamiento produce mejores resultados en dos direcciones.

1. Editar las clases mayoritarias ayuda a reducir la confusion entre clases minoritarias y mayoritarias.

2. Modificar el algoritmo de entrenamiento (incluyendo una función de coste) ayuda a incrementar la participación de las clases menos representadas en el proceso de aprendizaje, acelerando la convergencia de la red, lo que se traduce en incrementos en la precisón del clasificador sobre estas clases.

En síntesis, si lo que se desea es dar prioridad a las clases minoritarias, editar la región de solapamiento de las clases minoritarias en relación a las mayoritarias, es una alternativa efectiva. Por otro lado, la combinación de funciones de coste y técnicas para reducir la región de solapamiento logró disminuir el área de confusión y al mismo tiempo acelerar la convergencia de las clases menos representadas. Esto representa una forma original de tratar el problema del desbalance de las clases.

Tabla C.14: Feltwell: Resultados obtenidos en la fase de clasificación con el MLP (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	6.06	0.35	99.07	
	cls 2	7.80	0.24	81.97	cls 3 (11.72)
	cls 3	4.78	0.10	78.86	cls 1 (10.70)
	cls 4	7.99	0.15	83.91	cls 1 (11.48)
	cls 5	3.03	0.17	90.43	
Opción 3	cls 1	6.06	0.35	98.58	
	cls 2	7.80	0.24	80.85	cls 3 (14.85)
	cls 3	4.78	0.10	83.08	
	cls 4	7.99	0.15	83.35	cls 1 (10.74)
	cls 5	3.03	0.17	88.92	cls 1 (10.55)
EW (Opción 0)	cls 1	1.256	0.35	97.63	
	cls 2	3.019	0.24	73.62	cls 3 (13.99) cls 5 (11.17)
	cls 3	3.058	0.10	81.48	cls 4 (10.09)
	cls 4	3.004	0.15	83.19	
	cls 5	0.604	0.17	96.12	
EW (Opción 3)	cls 1	1.256	0.35	97.45	
	cls 2	3.019	0.24	69.70	cls 3 (23.85)
	cls 3	3.058	0.10	84.70	
	cls 4	3.004	0.15	81.80	
	cls 5	0.604	0.17	95.76	

Tabla C.15: Feltwell: Resultados obtenidos en la fase de clasificación con la red RBF (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
	cls 1	6.06	0.35	96.49	
	cls 2	7.80	0.24	89.58	
Opción 0	cls 3	4.78	0.10	32.90	cls 2 (37.57) cls 4 (12.85) cls 5 (13.50)
	cls 4	7.99	0.15	83.94	cls 1 (10.19)
	cls 5	3.03	0.17	81.27	cls 1 (12.19)
	cls 1	6.06	0.35	94.93	
	cls 2	7.80	0.24	79.46	cls 3 (15.51)
Opción 3	cls 3	4.78	0.10	65.98	cls 2 (15.05) cls 4 (12.76)
	cls 4	7.99	0.15	83.83	
	cls 5	3.03	0.17	92.94	
	cls 1	1.256	0.35	95.01	
	cls 2	3.019	0.24	75.56	cls 3 (17.68)
EW (Opción 0)	cls 3	3.058	0.10	60.63	cls 2 (16.77)
	cls 4	3.004	0.15	72.31	cls 1 (16.19)
	cls 5	0.604	0.17	89.51	
	cls 1	1.256	0.35	93.06	
	cls 2	3.019	0.24	71.45	cls 3 (20.42)
EW (Opción 3)	cls 3	3.058	0.10	76.77	cls 4 (10.90)
	cls 4	3.004	0.15	72.34	cls 1 (13.52)
	cls 5	0.604	0.17	91.08	

Tabla C.16: Feltwell: Resultados obtenidos en la fase de clasificación con la red RBF+VF (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	6.062	0.35	99.26	
	cls 2	7.805	0.24	86.37	
	cls 3	4.78	0.10	53.95	cls 2 (14.52) cls 4 (11.32) cls 5 (15.41)
	cls 4	7.999	0.15	83.88	cls 1 (11.89)
	cls 5	3.038	0.17	78.71	cls 1 (19.32)
Opción 3	cls 1	6.062	0.35	98.57	
	cls 2	7.805	0.24	85.12	cls 3 (12.42)
	cls 3	4.78	0.10	77.12	cls 4 (11.59)
	cls 4	7.999	0.15	83.37	cls 1 (11.23)
	cls 5	3.038	0.17	85.84	cls 1 (12.90)
EW (Opción 0)	cls 1	1.256	0.35	97.83	
	cls 2	3.019	0.24	84.78	cls 3 (11.47)
	cls 3	3.058	0.10	64.11	cls 2 (10.97) cls 4 (11.42) cls 5 (11.01)
	cls 4	3.004	0.15	81.82	
	cls 5	0.604	0.17	88.12	
EW (Opción 3)	cls 1	1.256	0.35	96.58	
	cls 2	3.019	0.24	77.34	cls 3 (17.92)
	cls 3	3.058	0.10	84.22	cls 4 (11.33)
	cls 4	3.004	0.15	77.78	
	cls 5	0.604	0.17	88.54	

Tabla C.17: Desempeño global del clasificador: Base de datos Feltwell. Los valores entre paréntesis hacen referencia a la desviación estándar.

MLP	Opción 0	Opción 1	EW (Opción 0)	EW (Opción 3)
PC	89.38(0.95)	89.01(0.51)	87.99(1.21)	87.04(0.71)
g-mean	86.60(1.64)	86.79(0.75)	85.97(1.47)	85.33(0.98)
RBF	Opción 0	Opción 1	EW (Opción 0)	EW (Opción 3)
PC	84.36(2.03)	86.51(1.94)	82.79(1.79)	82.94(1.17)
g-mean	62.21(25.96)	82.41(4.41)	76.52(6.55)	80.16(3.04)
RBF+VF	Opción 0	Opción 1	EW (Opción 0)	EW (Opción 3)
PC	86.15(1.90)	88.93(1.29)	87.49(1.85)	86.68(1.89)
g-mean	70.78(25.85)	85.60(2.86)	81.89(6.14)	84.63(1.80)

Tabla C.18: Satimage: Resultados obtenidos en la fase de clasificación con el MLP (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
	cls 1	5.629	0.23	90.87	
	cls 2	6.758	0.23	98.83	
	cls 3	17.493	0.11	90.71	
Opción 0	cls 4	13.707	0.20	97.71	
	cls 5	6.198	0.11	2.37	cls 1 (61.04) cls 4 (33.03)
	cls 6	6.532	0.12	70.25	cls 1 (15.74)
	cls 1	5.629	0.23	81.89	cls 5 (13.83)
	cls 2	6.758	0.23	97.51	
	cls 3	17.493	0.11	90.54	
Opción 3	cls 4	13.707	0.20	91.61	
	cls 5	6.198	0.11	65.73	cls 1 (19.91) cls 4 (13.22)
	cls 6	6.532	0.12	76.71	cls 1 (13.50)
	cls 1	1.805	0.23	75.21	cls 5 (18.66)
	cls 2	0.317	0.23	98.18	
	cls 3	4.894	0.11	91.43	
EW (Opción 0)	cls 4	4.090	0.20	88.46	cls 5 (10.18)
	cls 5	0.375	0.11	60.95	cls 1 (25.45) cls 4 (11.04)
	cls 6	0.872	0.12	77.05	
	cls 1	1.805	0.23	71.47	cls 5 (23.09)
	cls 2	0.317	0.23	96.83	
	cls 3	4.894	0.11	93.39	
EW (Opción 3)	cls 4	4.090	0.20	83.27	cls 5 (15.57)
	cls 5	0.375	0.11	84.08	
	cls 6	0.872	0.12	81.14	

Tabla C.19: Satimage: Resultados obtenidos en la fase de clasificación con la red RBF (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
	cls 1	5.629	0.23	79.21	cls 5 (11.45)
	cls 2	6.758	0.23	97.64	
	cls 3	17.493	0.11	92.81	
Opción 0	cls 4	13.707	0.20	95.89	
	cls 5	6.198	0.11	49.34	cls 1 (16.92) cls 4 (29.29)
	cls 6	6.532	0.12	63.50	cls 1 (14.01) cls 2 (13.88)
	cls 1	5.629	0.23	73.45	cls 5 (20.06)
	cls 2	6.758	0.23	95.81	
	cls 3	17.493	0.11	90.31	
Opción 3	cls 4	13.707	0.20	91.26	
	cls 5	6.198	0.11	69.10	cls 1 (10.90) cls 4 (16.68)
	cls 6	6.532	0.12	70.00	
	cls 1	1.805	0.23	67.49	cls 5 (23.91)
	cls 2	0.317	0.23	96.77	
	cls 3	4.894	0.11	92.50	
EW (Opción 0)	cls 4	4.090	0.20	90.45	
	cls 5	0.375	0.11	72.65	cls 4 (14.88)
	cls 6	0.872	0.12	70.46	
	cls 1	1.805	0.23	62.30	cls 5 (29.96)
	cls 2	0.317	0.23	93.23	
	cls 3	4.894	0.11	89.46	
EW (Opción 3)	cls 4	4.090	0.20	85.57	cls 5 (10.76)
	cls 5	0.375	0.11	79.95	cls 4 (11.04)
	cls 6	0.872	0.12	75.44	

Tabla C.20: Satimage: Resultados obtenidos en la fase de clasificación con la red RBF+VF (Global).

	Clase	F1	Razón	PC	% confusión (> 10 %)
Opción 0	cls 1	5.629	0.23	83.00	
	cls 2	6.758	0.23	98.42	
	cls 3	17.493	0.11	90.62	
	cls 4	13.707	0.20	97.10	
	cls 5	6.198	0.11	36.73	cls 1 (25.83) cls 4 (33.98)
	cls 6	6.532	0.12	57.68	cls 1 (21.73) cls 2 (12.19)
Opción 3	cls 1	5.629	0.23	76.12	cls 5 (16.88)
	cls 2	6.758	0.23	97.69	
	cls 3	17.493	0.11	91.07	
	cls 4	13.707	0.20	93.23	
	cls 5	6.198	0.11	65.51	cls 1 (12.27) cls 4 (18.69)
	cls 6	6.532	0.12	71.40	cls 1 (12.56)
EW (Opción 0)	cls 1	1.805	0.23	69.28	cls 5 (22.60)
	cls 2	0.317	0.23	98.16	
	cls 3	4.894	0.11	93.97	
	cls 4	4.090	0.20	91.16	
	cls 5	0.375	0.11	66.26	cls 4 (20.33)
	cls 6	0.872	0.12	69.92	
EW (Opción 3)	cls 1	1.805	0.23	64.17	cls 5 (27.98)
	cls 2	0.317	0.23	96.05	
	cls 3	4.894	0.11	92.50	
	cls 4	4.090	0.20	86.75	cls 5 (10.23)
	cls 5	0.375	0.11	77.11	cls 4 (11.47)
	cls 6	0.872	0.12	78.19	

Tabla C.21: Desempeño global del clasificador: Base de datos Satimage. Los valores entre paréntesis hacen referencia a la desviación estándar.

MLP	Opción 0	Opción 1	EW (Opción 0)	EW (Opción 3)
PC	82.26(0.31)	86.07(0.34)	83.66(0.34)	84.59(0.38)
g-mean	47.31(5.72)	83.34(0.95)	80.90(1.17)	84.70(0.36)
RBF	Opción 0	Opción 1	EW (Opción 0)	EW (Opción 3)
PC	83.28(0.94)	83.16(1.30)	82.50(1.19)	80.51(1.77)
g-mean	77.44(1.46)	80.95(1.20)	80.86(1.50)	80.25(1.75)
RBF+VF	Opción 0	Opción 1	EW (Opción 0)	EW (Opción 3)
PC	82.33(1.24)	84.48(0.77)	82.80(1.25)	82.20(1.47)
g-mean	71.96(7.04)	81.65(0.87)	80.32(2.11)	81.78(1.58)

Tabla C.22: Muestras por clase

Datos	1	2	3	4	5	6	7	8	9	10	11	Reducción total
Cayo	419	147	311	161	67	185	161	361	395	417	386	0%
Cayo + EW	374	147	175	161	67	185	161	341	342	213	200	21%
Feltwell	1488	1070	341	1411	814							0%
Feltwell + EW	1261	953	341	1253	707							12%
Satimage	1038	1072	479	961	415	470						0%
Satimage + EW	545	936	479	379	415	470						16%

Bibliografía

[Abu-Mostafa 1995] Y.S. Abu-Mostafa. *Hints.* Neural Computation, Vol. 7, No. 4, Pág. 639–671, 1995.

[Alejo 2006] R. Alejo, V. García, J.M. Sotoca, R.A. Mollineda y J.S. Sánchez. *Improving the classification accuracy of RBF and MLP neural networks trained with imbalanced samples.* En IDEAL, Volumen 4224, Pág. 464–471, Burgos, España, 2006.

[Alejo 2007] R. Alejo, V. García, J.M. Sotoca, R.A. Mollineda y J.S. Sánchez. *Improving the Performance of the RBF Neural Networks with Imbalanced Samples.* En IWANN, Pág. 162–169, San Sebastián, España, 2007. Springer Berlin / Heidelberg.

[Alejo 2008] R. Alejo, J.M. Sotoca y G. A. Casañ. *An Empirical Study for the Multiclass Imbalance Problem with Neural Networks.* En CIARP, Pág. 479–486, 2008.

[Alejo 2009] R. Alejo, J.M. Sotoca, R.M. Valdovinos y G.A. Casañ. *The Multi-Class Imbalance Problem: Cost Functions with Modular and Non-Modular Neural Networks.* En ISNN, Pág. 421–431, 2009.

[Amari 1996] S. Amari, N. Murata, K.-R. Müller, M. Finke y H. Yang. *Statistical Theory of Overtraining – Is Cross-Validation Asymptotically Effective?* En NIPS, Volumen 8, Pág. 176–182. The MIT Press, 1996.

[Anand 1993] R. Anand, K.G. Mehrotra, C.K. Mohan y S. Ranka. *An improved Algorithm for Neural Network Classification of Imbalanced Training Sets.* IEEE Transactions on Neural Networks, Vol. 4, Pág. 962–969, 1993.

[Anand 1995] R. Anand, K. Mehrotra, C.K. Mohan y S Ranka. *Efficient classification for multiclass problems using modular neural networks.* IEEE Transactions on Neural Networks, Vol. 6, No. 1, Pág. 117–124, 1995.

[Anderson 1977] J.A. Anderson, J.W. Silverstein, S.A. Ritz y R.S. Jones. *Distinctive features, categorical perception, and probability learning: Some applications of a neural model.* Psychological Review, Vol. 84, Pág. 413–451, 1977.

[Ariza 1996] F.J. Ariza, C. Pinilla y M.J. Borque. *Control de Calidad del Proceso de Clasificación de Imágenes de Satélite.* Mapping, No. 34, Pág. 74–86, 1996.

[Auda 1998] G. Auda y M. Kamel. *Modular Neural Network Classifiers: A Comparative Study.* Journal of Intelligent and Robotic Systems, Vol. 21, No. 2, Pág. 117–129, 1998.

[Barandela 2001] R. Barandela, E. Gasca y R. Alejo. *Corrección de la Muestra para el Aprendizaje del Perceptron Multicapa.* Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, Vol. 13, Pág. 2–9, 2001.

[Barandela 2003a] R. Barandela, J.S. Sánchez, V. García y F.J. Ferri. *Learning from Imbalanced Sets through Resampling and Weighting.* En IbPRIA, Pág. 80–88, 2003.

[Barandela 2003b] R. Barandela, J.S. Sánchez, V. García y E. Rangel. *Strategies for learning in class imbalance problems.* Pattern Recognition, Vol. 36, No. 3, Pág. 849–851, 2003.

[Barandela 2004] R. Barandela, R.M. Valdovinos, J.S. Sánchez y F.J. Ferri. *The Imbalanced Training Sample Problem: Under or over Sampling?* En SSPR/SPR, Pág. 806, 2004.

[Bárdossy 1998] A. Bárdossy. *Generating precipitation time series using simulated annealing.* Water Resources Research, Vol. 34, Pág. 1737–1744, 1998.

[Benoudjit 2003] N. Benoudjit y M. Verleysen. *On the Kernel Widths in Radial-Basis Function Networks.* Neural Processing Letters, Vol. 18, No. 2, Pág. 139–154, 2003.

[Bottou 1990] L. Bottou y P. Gallinari. *A framework for the cooperation of learning algorithms.* En NIPS-3, Pág. 781–788, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[Brunak 1990] S. Brunak y B. Lautrup. Neural networks: computers with intuition. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1990.

[Bruzzone 1997a] L. Bruzzone y S.B. Serpico. *Classification of imbalanced remote-sensing data by neural networks.* Pattern Recognition Letters, Vol. 18, Pág. 1323–1328, 1997.

[Bruzzone 1997b] L. Bruzzone y S.B. Serpico. *Training of neural networks for classification of imbalanced remote-sensing data.* En IGARSS, Volumen 3, Pág. 1202–1204, Singapore, Singapore, 1997.

[Castillo 1991] F. Castillo. *Incremental Neural Networks: A Survey.* Technical report, INPG Grenoble, France, 1991.

[Chan 1999] P.K. Chan, W. Fan, A.L. Prodromidis y S.J. Stolfo. *Distributed Data Mining in Credit Card Fraud Detection.* IEEE Intelligent Systems, Vol. 14, No. 6, Pág. 67–74, 1999.

[Chudler 2006] E.H. Chudler. *Brain Facts and Figures.* Web page, University of Washington Engineered Biomaterials, 2006. Disponible en http://faculty.washington.edu/chudler/facts.html.

[Cybenko 1989] G. Cybenko. *Approximation by superpositions of a sigmoidal function.* Mathematics of Control, Signals, and Systems (MCSS), Vol. 2, No. 4, Pág. 303–314, 1989.

[DARPA-USA 1988] DARPA-USA. Darpa neural network study. AFCEA Intl, 1988.

[Dash 2007] P.K. Dash, M. Nayak, M.R. Senapati y I.W.C. Lee. *Mining for similarities in time series data using wavelet-based feature vectors and neural networks.* Engineering Applications of Artificial Intelligence, Vol. 20, No. 2, Pág. 185–201, 2007.

[Ding 2004] S.Q. Ding y C. Xiang. *From multilayer perceptrons to radial basis function networks: a comparative study.* En IEEE CIS, Volumen 1, Pág. 69–74, Singapore, Singapore, 2004.

[Dorizzi 1993] B. Dorizzi, J.-M. Auger y P. Sebire. *Cooperation and modularity for classification through neuralnetwork techniques.* En SMC, Volumen 3, Pág. 469–474, 1993.

[Duda 2001] R.O. Duda, P.E. Hart y D.G. Stork. Pattern classification and scene analysis. Wiley, New York, 2 edición, 2001.

[Fahlman 1988] S.E. Fahlman. *Faster-Learning Variations on Back-Propagation: An Empirical Study.* En CMSS, Pág. 38–51, Carnegie Mellon University, Pittsburgh, PA, 1988.

[Fawcett 1997] T. Fawcett y F. Provost. *Adaptive Fraud Detection.* Data Mining and Knowledge Discovery, Vol. 1, No. 3, Pág. 291–316, 1997.

[Freeman 1991] J.A. Freeman y D.M. Skapura. Neural networks: algorithms, applications, and programming techniques. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1991.

[Freund 1995] Y. Freund. *Boosting a weak learning algorithm by majority.* Information and Computation, Vol. 121, No. 2, Pág. 256–285, 1995.

[Fu 2002] X. Fu, L. Wang, K.S. Chua y F. Chu. *Training RBF neural networks on unbalanced data.* En ICONIP, Volumen 2, Pág. 1016–1020, Singapore, Singapore, 2002.

[Fukushima 1979] K. Fukushima. *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.* Biological Cybernetics, Vol. 36, Pág. 193–202, 1979.

[Funahashi 1989] K. Funahashi. *On the approximate realization of continuous mappings by neural networks.* Neural Networks, Vol. 2, No. 3, Pág. 183–192, 1989.

[Ghodsi 2003] A. Ghodsi y D. Schuurmans. *Automatic basis selection techniques for RBF networks.* Neural Networks, Vol. 16, No. 5-6, Pág. 809–816, 2003.

[Golub 1996] G.H. Golub y C.F. Van Loan. Matrix computations. Johns Hopkins University Press, Baltimore, MD, USA, 3 edición, 1996.

[Gray 1984] R. Gray. *Vector quantization.* ASSP Magazine, IEEE Signal Processing Magazine, Vol. 1, No. 2, Pág. 4–29, 1984.

[Grossberg 1987] S. Grossberg. *Competitive learning: from interactive activation to adaptative resonance.* Cognitive Science, Vol. 11, Pág. 23–63, 1987.

[Guérin-Dugué 1995] A. Guérin-Dugué*et al.* *Deliverable R3-B1-P - Task B1: Databases.* Reporte Técnico 6891, Elena-NervesII "Enhanced Learning for Evolutive Neural Architecture", ESPRIT-Basic Research Project, Junio 1995. FTP: /pub/neural-nets/ELENA/Databases.ps.Z en ftp.dice.ucl.ac.be.

[Gúzman 2004] M. Gúzman y C. Vázquez. *Neurocomputación: Entrenamiento de redes neuronales.* Ciencia y Desarrollo, Vol. 30, No. 177, Pág. 26–31, 2004.

[Happel 1994] B.L.M. Happel y J.M.J. Murre. *Design and evolution of modular neural network architectures.* Neural Networks, Vol. 7, No. 6-7, Pág. 985–1004, 1994.

[Harpham 2004] C. Harpham, W. Dawson y R. Brown. *A review of genetic algorithms applied to training radial basis function networks.* Neural Computing and Applications, Vol. 13, No. 3, Pág. 193–201, 2004.

[Hashem 1997] S. Hashem. *Optimal linear combinations of neural networks.* Neural Networks, Vol. 10, No. 4, Pág. 599–614, 1997.

[Haykin 1999] S. Haykin. Neural networks. a comprehensive foundation. Pretince Hall, New Jersey, 2 edición, 1999.

[He 2009] H. He y E.A. Garcia. *Learning from Imbalanced Data.* IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 9, Pág. 1263–1284, 2009.

[Hebb 1949] D. Hebb. The organization of behavior - a neurophysiological theory. John Wiley & Sons, New York, USA, 1 edición, 1949.

[Hecht-Nielsen 1990] R. Hecht-Nielsen. Neurocomputing. Addison-Wesley, Massachusetts, 1990.

[Hilera 1995] V.J. Hilera J.R.and Martínez. Redes neuronales artificiales. fundamentos, modelos y aplicaciones. Ra-Ma, Madrid, España, 1 edición, 1995.

[Holland 1992] J.H. Holland. Adaptation in natural and artificial systems. MIT Press, Cambridge, MA, USA, 1992.

[Hopfield 1982] J.J. Hopfield. *Neural networks as physical systems with emergent collective computational abilities.* Proc. National Academy of Sciences of USA, Vol. 79, No. 8, Pág. 2554–2558, 1982.

[Horton] P. Horton y K. Nakai. *A Probabilistic Classification System for Predicting the Cellular Localization Sites of Proteins.*

[Hsu 1999] K.-l. Hsu, H. V. Gupta, X. Gao y S. Sorooshian. *Estimation of physical variables from multichannel remotely sensed imagery using a neural network: Application to rainfall estimation.* Water Resources Research, Vol. 35, Pág. 1605–1618, 1999.

[Hutchinson 1995] J.M. Hutchinson, A. Lo y T. Poggio. *A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks.* NBER Working Papers 4718, National Bureau of Economic Research, Inc, Febrero 1995. disponible en http://ideas.repec.org/p/nbr/nberwo/4718.html.

[Ian 2005] H.W. Ian y F. Eibe. Data mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, CA, 2 edición, 2005.

[Jacobs 1988] R.A. Jacobs. *Increased rates of convergence through learning rate adaptation.* Neural Networks, Vol. 1, No. 4, Pág. 295–308, 1988.

[Jacobs 1990] R.A. Jacobs y M.I. Jordan. *A Competitive Modular Connectionist Architecture.* En NIPS, Pág. 767–773, 1990.

[Jacobs 1991] R.A. Jacobs, M.I. Jordan, S.J. Nowlan y S.J. Hinton. *Adaptive mixtures of local experts.* Neural Computation, Vol. 3, Pág. 79–87, 1991.

[Jadid 1996] M.N. Jadid y D.R. Fairbairn. *Neural-network Applications in Predicting Moment-curvature Parameters from Experimental Data.* Engineering Applications of Artificial Intelligence, Vol. 9, No. 3, Pág. 309–319, 1996.

[Jain 1996] A.K. Jain, J. Mao y K.M. Mohiuddin. *Artificial Neural Networks: A Tutorial.* IEEE Computer, Vol. 29, No. 3, Pág. 31–44, 1996.

[Japkowicz 2002] N. Japkowicz y S. Stephen. *The class imbalance problem: a systematic study.* Intelligent Data Analysis, Vol. 6, Pág. 429–449, 2002.

[Jordan 1994] M.I. Jordan y R.A. Jacobs. *Hierarchical mixtures of experts and the EM algorithm.* Neural Computation, Vol. 6, No. 2, Pág. 181–214, 1994.

[Jordan 1995] M.I. Jordan y L. Xu. *Convergence results for the EM approach to mixtures of experts architectures.* Neural Networks, Vol. 8, No. 9, Pág. 1409–1431, 1995.

[Jutten 1995] C. Jutten y O. Fambon. *Pruning Methods: A Review.* En ESANN, Pág. 129–140, Brussels, Belgium, 1995. D facto publications. Invited paper.

[Kanellopoulos 1997] I. Kanellopoulos y G.G. Wilkinson. *Strategies and Best Practice for Neural-Network Image Classification.* International Journal of Remote Sensing, Vol. 18, No. 4, Pág. 711–725, March 1997.

[Kohonen 1977] T. Kohonen. Associative memory. A system-theoretical approach. Springer-Verlag, New York, 1977.

[Kohonen 1990] T. Kohonen. *The Self-Organizing Map.* En Proceedings of the IEEE, Pág. 1464–1480, 1990.

[Kotsiantis 2003] S. Kotsiantis y P. Pintelas. *Mixture of expert agents for handling imbalanced data sets.* Annals of Mathematics and Computing & TeleInformatics, Vol. 1, No. 1, Pág. 46–55, 2003.

[Kramer 1988] A.H. Kramer y A.L. Sangiovanni-Vincentelli. *Efficient Parallel Learning Algorithms for Neural Networks.* En NIPS, Pág. 40–48, 1988.

[Kubat 1997] M. Kubat y S. Matwin. *Addressing the Curse of Imbalanced Training Sets: One-Sided Selection.* En ICML, Pág. 179–186, Nashville, Tennessee, USA, 1997. Morgan Kaufmann.

[Kubat 1998] M. Kubat. *Decision Trees Can Initialize Radial-Basis Function Networks.* IEEE Transactions on Neural Networks, Vol. 9, No. 5, Pág. 813–821, September 1998.

[Kukar 1998] M. Kukar y I. Kononenko. *Cost-Sensitive Learning with Neural Networks.* En ECAI, Pág. 445–449, Brighton, UK, 1998. John Wiley and Sons.

[Kuncheva 2004] L.I. Kuncheva. Combining pattern classifiers: Methods and algorithms. Wiley-Interscience, July 2004.

[Lachtermacher 1995] G. Lachtermacher y J.D. Fuller. *Article Back propagation in time-series forecasting.* Journal of Forecasting, Vol. 14, No. 4, Pág. 381–393, 1995.

[Lawrence 1998] S. Lawrence, I. Burns, A.D. Back, A.C. Tsoi y C.L. Giles. *Neural Network Classification and Unequal Prior Class Probabilities.* En G. Orr, K.-R. Müller y R. Caruana, editores, Neural Networks: Tricks of the Trade, Volumen 1524 de *Lecture Notes in Computer Science*, Pág. 299–314. Springer Verlag, 1998.

[LeCun 1990] Y. LeCun, J. Denker, S. Solla, R. E. Howard y L. D. Jackel. *Optimal Brain Damage.* En NIPS-2, Volumen 2, Pág. 598–605, San Mateo, CA, 1990. Morgan Kauffman.

[LeCun 1993] Y. LeCun. *Efficient Learning and Second-order Methods.* En NIPS, Denver, CO., 1993.

[Ling 2007] C. X. Ling y V. S. Sheng. Cost-sensitive learning and the class imbalanced problem. 2007.

[Lippmann 1988] R.P. Lippmann. *An introduction to computing with neural nets.* SIGARCH Computer Architecture News, Vol. 16, No. 1, Pág. 7–25, 1988.

[Looney 1997] C. Looney. Pattern recognition using neuronal networks - theory and algorithms for engineers and scientists. Oxford University Press, New York, 1 edición, 1997.

[Looney 2002] C.G. Looney. *Radial basis functional link nets and fuzzy reasoning.* Neurocomputing, Vol. 48, No. 1-4, Pág. 489–509, 2002.

[Lowe 1989] D. Lowe. *Adaptive radial basis function non linearities, and the problem of generalisation.* En ICANN, Pág. 171–175, 1989.

[Lu 1998] Y. Lu, H. Guo y L. Feldkamp. *Robust neural learning from unbalanced data examples.* En IJCNN, Pág. 1816–1821, 1998.

[Martín 2001] B. Martín y A. Sanz. Redes neuronales y sistemas borrosos. Alfaomega-Rama, 2001.

[Masters 1993] T. Masters. Practical neural network recipes in c++. Academic Press Professional, Inc., San Diego, CA, USA, 1993.

[McCulloch 1943] W.S. McCulloch y W. Pitts. *A logical calculus of the ideas immanent in nervous activity.* Bulletin of Mathematical Biophysics, Vol. 5, Pág. 115–133, 1943.

[Moody 1989] J. Moody y C. Darken. *Fast learning in networks of locally-tuned processing units.* Neural Computation, Vol. 1, No. 2, Pág. 281–294, 1989.

[Murphey 2004] Y. Murphey, H. Guo y L. Feldkamp. *Neural Learning from Unbalanced Data.* Applied Intelligence, Vol. 21, Pág. 117–128, 2004.

[Murre 1992] J.M.J. Murre, R.H. Phaf y G. Wolters. *CALM: Categorizing and Learning Module.* Neural Networks, Vol. 5, No. 1, Pág. 55–82, 1992.

[Nakai 1991] K. Nakai y M. Kanehisa. *Expert system for predicting protein localization sites in gram-negative bacteria.* Proteins: Structure, Function, and Bioinformatics, Vol. 11, No. 2, Pág. 95–110, 1991.

[Newman 1998] D.J. Newman, S. Hettich, C.L. Blake y C.J. Merz. *UCI Repository of Machine Learning Databases.* Repository, University of California, Irvine, Dept. of Information and Computer Sciences, 1998. Disponible en http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[Nigrin 1993] A. Nigrin. Neural networks for pattern recognition. MIT Press, Cambridge, MA, USA, 1993.

[Pao 1989] Y.-H. Pao. Adaptive pattern recognition and neural networks. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[Pao 1994] Y.-H. Pao, G.H. Park y D.J. Sobajic. *Learning and generalization characteristics of the random vector Functional-link net.* Neurocomputing, Vol. 6, No. 2, Pág. 163–180, 1994.

[Papik 1998] K. Papik, B. Molnar, R. Schaefer, Z. Dombovari, Z. Tulassay y J. Feher. *Application of neural networks in medicine - a review.* Diagnostics and Medical Technology, Vol. 4, No. 3, Pág. 538–546, 1998.

[Parker 1985] D.B. Parker. *Learning-logic.* Technical Report 47, Center for Comp. Research in Economics and Management Sci., MIT, Abril 1985.

[Powell 1987] M. J. D. Powell. *Radial basis functions for multivariable interpolation: a review.* Algorithms for approximation, Pág. 143–167, 1987.

[Reed 1993] R. Reed. *Pruning algorithms-a survey.* IEEE Transactions on Neural Networks, Vol. 4, No. 5, Pág. 740–747, 1993.

[Rojas 1996] R. Rojas. Neural networks: A systematic introduction. Springer-Verlag New York, Inc., New York, NY, USA, 1996.

[Roli 1996] F. Roli, S.B. Serpico y L. Bruzzone. *Classification of Multisensor Remote-Sensing Images by Multiple Structured Neural Networks.* En ICPR, Volumen 4, Pág. 180, Washington, DC, USA, 1996. IEEE Computer Society.

[Ronco 1995] E. Ronco y P. Gawthrop. *Modular neural networks: a state of the art.* Technical Report CSC-95026, Centre for System and Control. Faculty of mechanical Engineering, University of Glasgow, Uk., 1995.

[Rosenblatt 1958] F. Rosenblatt. *The percepteron: A probabilistic model for information storage and organization in the brain.* Psycological Review, Vol. 65, No. 6, Pág. 386–408, 1958.

[Rumelhart 1986] D.E. Rumelhart, G.E. Hinton y R.J. Williams. Learning internal representations by error propagation. MIT Press, Cambridge, MA, USA, 1986.

[Russell 1996] P. Russell S. y Norvig. Inteligencia artificial. Un enfoque moderno. Prentice-Hall, EUA, 1996.

[Russell 1999] G.C. Russell y G. Kass. Assessing the overall accuracy of remotely sensed data: Principles and practice. Lewis Publishers, New York, Washington, D.C., 1999.

[Saha 1989] A. Saha y J.D. Keeler. *Algorithms for Better Representation and Faster Learning in Radial Basis Function Networks.* En NIPS, Pág. 482–489, 1989.

[Sánchez 2007] J.S. Sánchez, R.A. Mollineda y J.M. Sotoca. *An analysis of how training data complexity affects the nearest neighbor classifiers.* Pattern Analysis and Applications, Vol. 10, No. 3, Pág. 189–201, 2007.

[Schölkopf 1997] B. Schölkopf, Sung K.-K., Burges C.J.C., F. Girosi, P. Niyogi, T. Poggio y V. Vapnik. *Comparing support vector machines with Gaussian kernels to radial basis function classiers.* IEEE Transactions on Signal Processing, Vol. 45, No. 11, Pág. 2758–2765, 1997.

[Schwenker 2001] F. Schwenker, H.A. Kestler y G. Palm. *Three learning phases for radial-basis-function networks.* Neural Networks, Vol. 14, No. 4-5, Pág. 439–458, 2001.

[Serpico 1993] S.B. Serpico, F. Roli, P. Pellegretti y G. Vemazza. *Structured neural networks for the classification of multisensor remote-sensing images.* En IGARSS, Pág. 907–909, Tokyo, Japan, 1993.

[Upadhyaya 1992] B.R. Upadhyaya y E. Eryurek. *Application of Neural Networks for Sensor Validation and Plant Monitoring.* Nuclear Technology, Vol. 97, No. 2, Pág. 170–176, 1992.

[Uykan 2000] Z. Uykan, C. Guzelis, M.E. Celebi y H.N. Koivo. *Analysis of input-output clustering for determining centers of RBFN.* IEEE Transactions on Neural Networks, Vol. 11, No. 4, Pág. 851–858, 2000.

[Valdovinos 2006] R.M. Valdovinos. *Técnicas de Submuestreo, Toma de Decisiones y Análisis de Diversidad en Aprendizaje Supervisado con Sistemas Múltiples de Clasificación.* Tesis doctoral, Universitat Jaume I, Castellón, España, Septiembre 2006.

[Werbos 1974] P. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* Tesis doctoral, Harvard University, Cambridge, MA, 1974.

[Wettschereck 1992] D. Wettschereck y T.G. Dietterich. *Improving the performance of Radial Basis Function Networks by Learning Center Locations.* En NIPS 4, Volumen 4, Pág. 1133–1140. Morgan Kaufmann, San Mateo, CA, 12 1992.

[Widrow 1960] B. Widrow y M.E. Hoff. *Adaptative switching circuits.* IRE WESCON Convention Records, Pág. 96–104, 1960.

[Wilson 1972] D.L. Wilson. *Asymptotic properties of nearest neighbor rules using edited data.* IEEE Transactions on Systems, Man and Cybernetics, Vol. 2, No. 4, Pág. 408–420, 1972.

[Wilson 2000] D. Randall Wilson y Tony R. Martinez. *Reduction Techniques for Instance-BasedLearning Algorithms.* Machine Learning, Vol. 38, No. 3, Pág. 257–286, 2000.

[Xu 1995] L. Xu, M.I. Jordan y G.E. Hinton. *An Alternative Model for Mixtures of Experts.* En NIPS, Volumen 7, Pág. 633–640. The MIT Press, 1995.

[Xu 1998] L. Xu. *RBF nets, mixture experts, and Bayesian Ying-Yang learning.* Neurocomputing, Vol. 19, No. 1-3, Pág. 223–257, 1998.

[Yao 1993] X. Yao. *A review of evolutionary artificial neural networks.* International Journal of Intelligent Systems, Vol. 8, No. 4, Pág. 539–567, 1993.

[Zhou 2006] Z.-H. Zhou y X.-Y. Liu. *Training cost-sensitive neural networks with methods addressing the class imbalance problem.* IEEE Transactions on Knowledge and Data Engineering, Vol. 18, Pág. 63–77, 2006.

[Zurada 1992] J. Zurada. Introduction to artificial neural systems. West Publishing Co., St. Paul, MN, USA, 1992.

Índice alfabético