Ensembles de redes neuronales construidos mediante algoritmos híbridos multiobjetivo para optimizar la precisión y la sensitividad

Juan C. Fernández[1], Mariano Carbonero[2], Pedro A. Gutiérrez[1], César Hervás[1]

Resumen—**En este trabajo se propone la construcción automática de un modelo de modelos o ensemble de redes neuronales para multiclasificación a partir del frente de Pareto obtenido mediante un algoritmo memético multiobjetivo evolutivo. Pretendemos que estos modelos tengan un alto porcentaje de patrones bien clasificados (precisión) no sólo sobre todo el conjunto de generalización, sino sobre cada una de las clases asociadas al problema (sensitividad). Como veremos a lo largo del trabajo, cuando estos dos objetivos se quieren optimizar a la vez entran en conflicto, sobre todo cuando la precisión o número de clases son altos, o cuando las clases del problema se encuentran no balanceadas en un alto grado. Para resolver este problema de forma automática utilizamos un algoritmo evolutivo memético basado en el concepto de dominancia de Pareto. En concreto utilizaremos una variante memética del algoritmo NSGA2 al que denominaremos Memetic Pareto NSGA2 (MPNSGA2). Para implantar esta metodología híbrida utilizaremos como algoritmo de búsqueda local una mejora del algoritmo RProp (*Resilient Backpropagation*). Una vez construido el frente de Pareto utilizamos diferentes estrategias automáticas de construcción de un modelo de modelos del frente. El algoritmo lo aplicamos a seis problemas obtenidos del repositorio de la UCI.**

Palabras clave-- **Algoritmo memético, Algoritmos Multiobjetivo Evolutivos, Multiclasificación, Precisión, Redes Neuronales, Sensitividad.**

I. INTRODUCCIÓN

La resolución de problemas de clasificación es una de las áreas de mayor interés en la actualidad en el campo del modelado de sistemas dinámicos reales en diferentes áreas de investigación. Asociado a este análisis se han desarrollado en los últimos años diferentes modelos de redes neuronales artificiales para regresión y clasificación, pero las redes de tipo perceptrón multicapa (MLP) basadas en la utilización de unidades de base sigmoide, US, son las más utilizadas.

Las Redes Neuronales Artificiales (RNA) [1,2], constituyen una de las áreas de la ingeniería del conocimiento que más se han desarrollado en los últimos años, aplicándose con éxito a problemas reales como el reconocimiento de patrones [3], diagnóstico médico, modelado de datos financieros [4], predicción meteorológica [5], etc.

El diseño de algoritmos de aprendizaje para RNA se puede dividir básicamente en dos grupos: algoritmos de búsqueda local y algoritmos de búsqueda global. Al primer grupo pertenecen los métodos basados en gradiente como el algoritmo de retropropagación del error (BP), donde la arquitectura del modelo de red se impone desde el inicio y donde hace falta la experiencia de un experto junto con tediosos procesos de prueba y error. Al segundo grupo pertenecen los algoritmos evolutivos (AE), donde el modelo de red cambia a lo largo del proceso evolutivo optimizando tanto la arquitectura como los valores de sus pesos, individual o conjuntamente [6,7]. Estos métodos y técnicas se han desarrollado para encontrar mejores aproximaciones en la evolución de los modelos de red, estando pendientes tanto del diseño de la arquitectura de la red como de los valores de las conexiones para mejorar su capacidad de generalización, además tienen una mejor capacidad para escapar de los óptimos locales y una mayor y más robusta adaptación a los cambios de entorno que los métodos basados en gradiente.

Existen diferentes métodos para optimizar dos o más objetivos aunque los más populares son los métodos que utilizan término de regularización mediante combinación lineal de objetivos y los métodos basados en dominancia de Pareto. Los primeros suelen presentar desventajas como la generación de una única solución de tipo Pareto a la vez [8] y suponen la convexidad de la frontera de Pareto. La utilización de RNA junto con algoritmos de entrenamiento evolutivos utilizando el concepto de dominancia de Pareto se conoce como Multiobjective Evolutionary Artificial Neural Networks [9], MOEANN, y es una técnica que se está utilizando con éxito en los últimos años para resolver tareas de clasificación y que tiene su principal exponente en el trabajo H. Abbass [10]. La tarea principal de estos algoritmos en su uso con RNA es el diseño de modelos con una alta precisión C (número de patrones bien clasificados) y con una pequeña complejidad estructural que permita simplicidad e interpretabilidad (la raíz cuadrada de la suma de los valores absolutos de los pesos de la red, el número de neuronas ocultas o el número de

[1] Departamento de Informática y Análisis Numérico de la Universidad de Córdoba, campus de Rabanales, edificio Albert Einstein, 3ª planta, 14071, Córdoba, España. E-mail: fernandezcaballero@gmail.com, i02gupep@uco.es, chevas@uco.es.

[2] Facultad de ciencias económicas y empresariales, ETEA, escritor Castilla Aguayo 4, 14005, Córdoba, España. E-mail: mariano@etea.com.

enlaces de la red [11,12]).

La elección de estos dos objetivos no siempre es la adecuada cuando queremos obtener un nivel aceptable de buena clasificación en todas las clases. De esta forma el objetivo de este trabajo es proponer una nueva aproximación para la obtención de multiclasificadores basados en modelos de redes neuronales donde se optimice tanto la precisión global como el nivel de buena clasificación en todas las clases; así, definiremos una nueva medida bidimensional del rendimiento del clasificador. En concreto consideraremos el mínimo de las sensitividades, S, esto es, el mínimo del porcentaje de patrones bien clasificados pertenecientes a una clase con respecto al número de patrones pertenecientes a la citada clase y por otra parte la precisión, C. Basándonos en los resultados obtenidos en [13], utilizar la sensitividad proporciona una interesante perspectiva para construir modelos en el reconocimiento de patrones.

De esta manera presentaremos en las siguientes secciones un algoritmo multiobjetivo evolutivo memético basado en dominancia de Pareto para el diseño de modelos de red (arquitectura y pesos) introduciendo el algoritmo de búsqueda local iRprop+ [14]. Una vez obtenido el frente de Pareto utilizaremos, sobre seis bases de datos [15], tres metodologías de construcción de un modelo de modelos o ensemble, a fin de extraer la información que aportan todos los modelos del citado frente.

Analizaremos, de esta manera, empíricamente el rendimiento del algoritmo justificando la estrategia MOEANN utilizada. El resto del trabajo se organiza como sigue. En la siguiente Sección proponemos las medidas de precisión y de sensitividad. En la Sección tercera hacemos un breve estado del arte de otros algoritmos del tipo MOEANN. En la Sección cuarta proponemos y describimos el algoritmo multiobjetivo utilizado en este trabajo, MPNSGA2. En la Sección 5 planteamos nuestro diseño experimental para terminar en la Sección 6 comentando las conclusiones extraídas de este trabajo.

II. PRECISIÓN Y SENSITIVIDAD EN CLASIFICACIÓN

A. *Precisión*

La comunidad que investiga en las áreas de estadística y de aprendizaje de máquinas ha utilizado tradicionalmente la métrica C, para medir el rendimiento de un clasificador obviando por lo general el porcentaje de buena clasificación por grupo. Sin embargo esta forma de presentar sólo el nivel de clasificación global ha sido criticado por diferentes autores, sobre todo en problemas multiclase y/o no balanceados [16,17]; puesto que la medida de la precisión no puede capturar los diferentes aspectos de un clasificador sobre todo si comparamos varios de ellos. Suponiendo que el coste de una mala clasificación es igual y que no tenemos preferencia en la clasificación de una determinada clase, entenderemos que un buen clasificador es aquel que presenta un alto nivel de precisión a la vez que un considerable nivel de buena clasificación para cada clase. En problemas reales estos dos objetivos están en conflicto puesto que para alcanzar un alto nivel de precisión por lo general es necesario sacrificar el nivel de una o varias clases, sobre todo en problemas no balanceados o con múltiples clases [18].

B. *Sensitividad contra precisión*

Para un problema de clasificación con Q clases y N patrones de entrenamiento o generalización definimos el clasificador g mediante una matriz $Q \times Q$ de contingencia o de confusión $M(g)$, de la forma:

$$M(g) = \left\{ n_{ij}; \sum_{i,j=1}^{Q} n_{ij} = N \right\}$$

donde n_{ij} representa el número de veces que los patrones son predichos por g para estar en la clase j cuando en realidad pertenecen a la clase i. La diagonal de la matriz se corresponde con los patrones correctamente clasificados y los elementos de fuera de la diagonal principal con los errores de clasificación. Si denotamos el número de patrones de la clase i por:

$$f_i = \sum_{j=1}^{Q} n_{ij}, \; i = 1, \ldots, Q$$

podemos, en función de esta matriz, definir dos medidas escalares para poder considerar los elementos de la matriz de confusión desde diferentes puntos de vista. Sea $S_i = n_{ii} / f_i$ el número de patrones correctamente predichos en la clase i con respecto al número total de patrones de la citada (lo que se define como sensitividad de la clase i). Así, la sensitividad para la clase i es un estimador de la probabilidad de predecir correctamente la clasificación de un patrón de dicha clase.

A partir de las sensitividades de cada clase definimos la sensitividad S del clasificador como el mínimo de las sensitividades de cada una de las clases:

$$S = mín \{ S_i; i = 1, \ldots, Q \} \quad (1)$$

Por otra parte el porcentaje de patrones correctamente clasificados o precisión, C, se define como:

$$C = (1/N) \sum_{j=1}^{Q} n_{jj} \quad (2)$$

De esta manera consideraremos la medida bidimensional (S,C) asociada con el clasificador g para evaluar estas dos características, el rendimiento global y el rendimiento en cada una de las clases.

La selección de S como una medida

complementaria a C se justifica si consideramos que C es una media ponderada de las sensitividades de las clases, esto es:

$$C = \sum_{i=1}^{Q} \frac{f_i}{N} S_i$$

con pesos que dependen de la base de datos, de forma tal que ambas medidas nos proporcionen una visión computacional e intuitiva de las sensitividades de cada una de las Q clases.

Es claro que dos cantidades no pueden contemplar toda la información contenida por las $Q(Q-1)$ proporciones de mala clasificación contenidas en la matriz de confusión; sin embargo el par (S,C) intenta obviar las desventajas de considerar todas estas proporciones de mala clasificación, como por ejemplo, las dificultades para obtener una representación gráfica que nos permita ver el rendimiento de los clasificadores, el incremento en la dimensión del frente de Pareto con respecto al número de objetivos, asi como el coste computacional asociado con un problema de optimización multiobjetivo que tiene tantos objetivos. Así, el par (S,C) obtiene un punto medio entre el rendimiento de un clasificador basado en una medida escalar y aquellos multidimensionales basados en las $Q(Q-1)$ proporciones de mala clasificación. Es fácil probar que estas dos medidas verifican las inecuaciones $S \le C \le 1-(1-S)p^*$, siendo p^* el mínimo de las probabilidades estimadas a priori, lo que confiere a este valor una gran importancia en las relaciones entre ambas medidas.

Por tanto, cada clasificador se puede representar como un punto en la región sombreada de la Figura 1, donde podemos visualizar los resultados obtenidos independientemente del número de clases del problema.

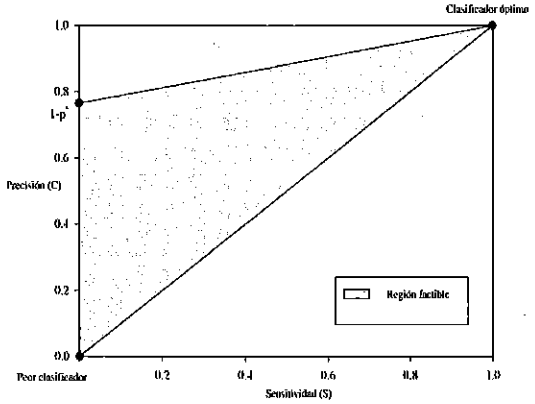

Fig. 1. Región factible en el espacio bidimensional (S, C) para un problema de clasificación dado.

III. ALGORITMOS MULTIOBJETIVO EVOLUTIVOS EN REDES NEURONALES ARTIFICIALES (MOEANN)

Los algoritmos de tipo Pareto como los (MOEANN) deben de proporcionar una distribución homogénea de la población lo largo del frente junto con una mejora de las soluciones a lo largo de las sucesivas generaciones del algoritmo evolutivo. De esta forma estas técnicas presentan un buen conjunto de soluciones que, cuando son evaluadas, producen vectores cuyas componentes representan la composición de la solución en el espacio de objetivos. Un decisor elegirá implícitamente una solución o un conjunto de soluciones aceptables seleccionando uno o varios de estos vectores. En este trabajo la selección del modelo se hará de forma automática utilizando diferentes propuestas de obtención de un modelo de modelos, donde consideraremos todos los modelos del último frente de Pareto.

Existe un limitado número de trabajos que utilizan MOEANN para entrenar una población de RNA utilizando técnicas multiobjetivo donde habitualmente, y a diferencia de este trabajo, se utilizan como objetivos la minimización del error cometido sobre el conjunto de entrenamiento y la complejidad de la red, siendo los principales exponentes H. Abbass [19] y Y. Jin [9].

Una mejora del AE tanto desde un punto de vista mono o multiobjetivo basado o no en la dominancia de Pareto es la incorporación de procedimientos de búsqueda local a lo largo de la evolución [20]. De esta manera mediante la combinación de un AE y de un procedimiento de búsqueda local, el algoritmo memético o híbrido [21], podrá llevar a cabo en primer lugar una búsqueda global dentro del espacio de soluciones, localizando modelos de redes cercanas al óptimo global, y en segundo lugar un procedimiento de búsqueda local que nos lleve rápidamente y eficientemente a la mejor solución.

En este trabajo proponemos un algoritmo memético multiobjetivo evolutivo para diseñar modelos de redes de tipo MLP para resolver problemas de multiclasificación [22]. En concreto, utilizaremos el algoritmo NSGA2 junto con una versión propia del algoritmo mejorado de búsqueda local Rprop (Resilient Barckpropagation) cuyo rendimiento en diversos problemas ha sido probado [14]. Con este algoritmo de búsqueda local mejoramos el frente de Pareto obtenido en la dirección del objetivo asociado a la optimización del error de clasificación.

IV. ALGORITMO MEMÉTICO BASADO EN DOMINANCIA DE PARETO

C. *Marco de trabajo del clasificador base*

En este trabajo utilizamos modelos de redes neuronales de tipo MLP con unidades de base sigmoide, de forma tal que modelen funciones discriminantes que expresen las relaciones existentes entre las características de entrada de

cada patrón y la probabilidad de pertenencia de ese patrón a una determinada clase.

El modelo funcional asociado a estas funciones de base es:

$$f_l(x,\theta_l) = \beta_0^l + \sum_{j=1}^{M} \beta_j^l \sigma_j^l(x), l = 1,2,...,J-1$$

donde $\boldsymbol{\theta} = (\theta_1,...,\theta_J)^T$ es la matriz transpuesta que contiene todos los pesos de la red, $\boldsymbol{\theta}_1 = (\beta_0^l, \beta_1^l,...,\beta_m^l, w_1,...,w_m)$ es el vector de pesos del nodo de salida l, $\mathbf{w}_j = (w_{1j},...,w_{kj})$ es el vector de pesos del nodo oculto j, J es el número de clases del problema, M el número de nodos o unidades de base sigmoide en capa oculta, $\mathbf{x}$ el patrón de entrada y

$$\sigma_j(\mathbf{x}) = \frac{1}{1+e^{-\left(w_0^j + \sum_{i=1}^{K} w_i^j x_i\right)}}$$

siendo K el número de características de cada patrón a clasificar. Para analizar los valores de salida de la red desde un punto de vista probabilístico consideraremos como función de activación la función dada por la expresión:

$$g_l(\mathbf{x},\boldsymbol{\theta}_l) = \frac{\exp f_l(\mathbf{x},\boldsymbol{\theta}_l)}{\sum_{l=1}^{J} \exp f_l(\mathbf{x},\boldsymbol{\theta}_l)}, l = 1,2,...,J$$

siendo $g_l(\mathbf{x},\boldsymbol{\theta}_l)$ la probabilidad de que el patrón $\mathbf{x}$ pertenezca a la clase J. Adoptamos la representación habitual "1-de-J" de las etiquetas de las clases codificando el vector $\mathbf{y} = (y^{(1)}, y^{(2)}..., y^{(J)})$ de forma tal que $y^{(l)} = 1$ si $\mathbf{x}$ se corresponde con un ejemplo perteneciente a la clase l e $y^{(l)} = 0$ en otro caso. De esta manera la regla optima de decisión $C(\mathbf{x})$ acerca de la pertenencia de un patrón $\mathbf{x}$ a la clase l, vendrá dada por la regla

$$C(\mathbf{x}) = \hat{l}, \text{ donde } \hat{l} = \arg\max_l g_l(\mathbf{x},\hat{\boldsymbol{\theta}}), \textit{ para } l = 1,2,...,J$$

De esta forma la probabilidad de pertenencia a una clase, en nuestro caso la última, no hace falta estimarla debido a la condición de normalización asociada a la axiomática del cálculo de probabilidades, así, suponemos que $f_J(\mathbf{x},\boldsymbol{\theta}_J) = 0$. Esta simplificación implica la eliminación de un nodo de la capa de salida de la red, siendo por tanto el número de nodos de esta capa $J-1$.

D. *Funciones de aptitud*

Para establecer una medida que determine la bondad de los modelos MLP consideraremos dos funciones de aptitud, el porcentaje de patrones bien clasificados sobre el conjunto de entrenamiento, C, definida en (2) y la función de entropía cruzada del error [1]. Si utilizamos el conjunto de datos de entrenamiento $D = \{(\mathbf{x}_n, \mathbf{y}_n);\ 1 \le n \le N\}$ entonces la función de entropía tiene la siguiente expresión para J clases:

$$l(\boldsymbol{\theta};g) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{J} y_n^{(l)} \log g_l(\mathbf{x}_n,\boldsymbol{\theta}_l)$$

La ventaja de utilizar como función de error $l(\boldsymbol{\theta};g)$ en lugar de $(1-C)$ esta en que la primera es una función continua, lo que nos permite un entrenamiento que converja hacia soluciones óptimas más robustamente. Por ello la primera medida de aptitud para maximizar C es una transformación estrictamente decreciente del error de entropía $l(\boldsymbol{\theta};g)$ dada por:

$$A(g) = \frac{1}{1+l(\theta;g)}$$

donde g es un modelo de funciones de base sigmoides dado por la función multivaluada

$$g(\mathbf{x},\boldsymbol{\theta}) = (g_1(\mathbf{x},\boldsymbol{\theta}_1),...,g_J(\mathbf{x},\boldsymbol{\theta}_J))$$

La segunda función objetivo a maximizar es la Sensitividad, S, del clasificador definida en (1).

E. *MPNSGA2 Algorithm*

En esta sección vamos a describir el algoritmo memético utilizado, al que llamaremos MPNSGA2 y el cual se basa en el original NSGA2 diseñado por K.Deb [23]. Este algoritmo obtiene un frente de Pareto con clasificadores que presentan un buen balanceo entre el nivel de clasificación global y el parcial de cada clase del problema.

Nuestra aproximación evoluciona a la vez la arquitectura y los valores de los pesos de las conexiones de la red. La red neuronal se representa mediante una aproximación orientada a objeto de forma tal que el algoritmo actúa directamente con el fenotipo.

No consideramos el operador de cruce debido a sus potenciales desventajas en la evolución de redes neuronales [24]. Como operadores de mutación utilizamos añadir/borrar neuronas y añadir/borrar enlaces, de forma similar a como se hace en [24]. Además de estos 4 mutadores estructurales se utiliza una mutación paramétrica sobre cada uno de los pesos de la red $w_{ji}(t+1) = w_{ji}(t) + \xi(t)$, donde $\xi(t) \in N(0,T(t))$ es una variable aleatoria normalmente distribuida con media 0 y varianza $T(t)$, siendo esta última una temperatura en descenso geométrico a través de la evolución. Otros detalles asociados a los operadores de mutación así como al procedimiento de generación de las redes tanto en la población inicial como a lo largo de la evolución se encuentran en los trabajos [25,26].

MPNSGA2 comienza generando una población inicial P_0 de tamaño N. Ordenamos los modelos de la población basándonos en el criterio de dominancia de Pareto, asignando a cada modelo una aptitud (o ranquin) igual a su nivel de no dominancia (siendo 1 el mejor nivel, 2 el siguiente mejor, y así sucesivamente). A continuación utilizamos una selección por torneo binario y operadores de mutación estructural y paramétrica

para crear una población de descendientes Q_0 de tamaño N.

En la Figura 2 se muestra el pseudo código del algoritmo MPNSGA2.

1. t=0 y generar una población aleatoria P(t) de tamaño N, donde cada individuo presenta una estructura en capa oculta con unidades de base sigmoides.
2. Evaluar los individuos P(t) en función de la entropía y la sensitividad.
3. Utilizar el ordenamiento rápido de no dominados para obtener una lista F con los frentes de la población P(t).
4. Asignar a cada individuo un valor o ranquin igual a su nivel de dominancia y un valor de distancia de desplazamiento para el caso en que exista un empate en el proceso de selección.
5. Usar un selector por torneos binario para seleccionar N individuos de P, acordando a su ranquin y a su distancia de desplazamiento.
6. Hacer una mutación (una de las 5 posibles seleccionada de manera aleatoria) a cada uno de los individuos seleccionados, para así generar una población hija Q(t) de tamaño N.
7. Evaluar los individuos Q(t) en función de la entropía y la sensitividad.
8. Mientras que el criterio de parada no se cumpla hacer
   a) f = 1
   b) R(t) = P(t) U Q(t).
   c) F = ordenamiento_rapido_no_dominados(R(t)).
   d) Si (número de generaciones es igual a 2/7 o 4/7 o 6/7 del número total de generaciones)
      e) Aplicar iRprop+ a los individuos del primer frente de Pareto F^1 de F, y evaluar los individuos F^1 en función de la entropía y la sensitividad, $R(t) = R'(t)$.
      f) F = ordenamiento_rapido_no_dominados(R'(t))
   g) fin si
   h) Mientras el tamaño de la población P(t+1) sea < N hacer
      I. Calcular la distancia de desplazamiento para el frente F^f
      II. $P(t+1) = P(t+1) \cup F^f$
      III. f = f+1
   i) fin mientras
   j) Ordenar la población P(t+1) de acuerdo a su ranquin y a su valor de distancia de desplazamiento y seleccionar los primeros N individuos. La nueva población P(t+1) de tamaño N está ahora completada.
   k) Usar selector por torneos binario teniendo en cuenta los valores de ranquin y distancia de desplazamiento para obtener N individuos desde P(t+1).
   l) Hacer una mutación (una de las 5 posibles seleccionada de manera aleatoria) a cada uno de los individuos seleccionados por torneo binario y generar una nueva población Q(t+1) de tamaño N.
   m) Evaluar los individuos Q(t+1) en base a la entropía y la sensitividad.
   n) t = t+1
9. fin mientras

Fig. 2. Pseudocódigo de MPNSGA2

F. El algoritmo iRprop+

El algoritmo *Improved Resilient Back-propagation* (iRprop+) es una mejora del original Rprop [27], siendo este un procedimiento basado en gradiente, difiriendo de las técnicas clásicas de propagación hacia atrás del error en que las derivadas parciales de la función error sólo son usadas para determinar el sentido en que deben ser corregidos los pesos de la red, pero no las magnitudes de los ajustes.

En este trabajo usamos dicho algoritmo cuando combinamos la población de los padres y de los descendientes en MPNSGA2. Entonces, solo los individuos del primer frente de Pareto (obtenido mediante una ordenación rápida de los no dominados) de la población combinada se optimizan mediante el algoritmo iRprop+, reduciendo el coste computacional porque no aplicamos la búsqueda local a todos los individuos mutados de la población de descendientes. Este procedimiento de búsqueda local se aplica sólo al principio (2/7 del nº de generaciones), en medio (4/7) y al final (6/7) del proceso evolutivo; mientras que en otros trabajos se usa la búsqueda local en cada generación [9]. El citado algoritmo ha sido modificado en cuanto a la función gradiente que guía el proceso de búsqueda, adaptándolo a la entropía como función de error utilizada.

G. Ensembles

Es un hecho teórica y empíricamente comprobado que la combinación (*ensemble*) de los resultados obtenidos por distintos clasificadores puede, si éstos, y la forma en que se combinen son adecuadamente elegidos, mejorar los resultados que cada uno de ellos proporciona. El éxito de un *ensemble* depende, en primera instancia, de dos factores relativos a los clasificadores que lo integran: precisión y diversidad. Tanto la calidad individual de cada clasificador como las diferencias entre ellos, en lo que a la manera en que el éxito se consigue, son condiciones necesarias para el eficaz funcionamiento del *ensemble* en el que se integren.

Para cada uno de los problemas tratados hemos considerado la incorporación en un *ensemble* de todos los clasificadores que forman el frente en el espacio (S,C). Esta elección implica la primera de las condiciones por la misma definición de frente de Pareto, pues cada uno de sus elementos integrantes es no dominado lo que supone un elevado grado de éxito entre los clasificadores considerados.

Por lo que a la diversidad se refiere, hemos asignado a cada clasificador sus coordenadas (S,C) y considerado como medida de la diversidad entre dos clasificadores g_1 y g_2 la distancia euclídea entre estas asignaciones:

$$d(g_1,g_2)=\sqrt{(C_1-C_2)^2+(S_1-S_2)^2}$$

El hecho de que tanto C como S tomen valores en el intervalo $[0,1]$ conlleva pesos similares para ambas medidas lo que, en principio, hace innecesario, al menos por razones de escala, la ponderación de los objetivos.

Definida así la diversidad entre cada par de algoritmos, la del frente $FP=\{g_1,\ldots,g_T\}$ formado por T modelos se ha obtenido como media aritmética de las diferencias entre cada par de ellos:

$$D=\frac{2}{T(T-1)}\sum_{i=1}^{T-1}\sum_{j=i+1}^{T} d(g_i,g_j)$$

Naturalmente el cálculo de la media presupone que la distribución de las distancias calculadas hacen de ésta una medida representativa de las mismas. De no darse esta premisa, otros estadísticos de posición más robustos, como la mediana, podrían ser empleados.

Esta misma observación puede hacerse también para la combinación en un único resultado final de las medidas de diversidad correspondientes a los frentes de Pareto de distintas iteraciones de un mismo algoritmo.

El segundo de los aspectos a los que se ha hecho referencia anteriormente es la forma en que los resultados proporcionados por los distintos modelos se combinan para determinar la del *ensemble*. De entre las distintas opciones que habitualmente se manejan hemos elegido para su comparación tres:

- *Majority Voting:* Donde cada patrón es asignado a la clase más votada por sus clasificadores integrantes. Siendo $C(\mathbf{x},g_j)$ la clase a que el clasificador g_j asigna al patrón $\mathbf{x}$ la asignación efectuada por el *ensemble*, $C(\mathbf{x})$ vendrá dada por la moda de las asignaciones individuales:

$$C(\mathbf{x}) = Mo\{C(\mathbf{x},g_j)\}$$

- *Simple Averaging:* Se calcula la media aritmética, para cada patrón, de las probabilidades de asignación a cada una de las Q clases para cada uno de los T modelos en FP. La asignación se hará a aquella clase para la que resulte una mayor probabilidad media. Siendo $P(\mathbf{x},g_j,C_k)$ la probabilidad estimada por el clasificador g_j de pertenencia a la clase C_k del patrón $\mathbf{x}$, la clase asignada será:

$$C(\mathbf{x}) = \arg\max_k \sum_{j=1}^{T} P(\mathbf{x},g_j,C_k)$$

expresión en la que se ha prescindido, por ser constante para todos los elementos, de la división por T, el número total de elementos promediados.

- *Winner-Takes-All:* Se asigna cada patrón a la clase a la que lo asigna el clasificador que presenta la mayor probabilidad de asignación:

$$C(\mathbf{x}) = \arg_k \max_{j,k} \{P(\mathbf{x},g_j,C_k)\}$$

V. EXPERIMENTACIÓN

Para poder analizar el rendimiento de los tres métodos de ensembles propuestos hemos considerado 6 bases de datos obtenidas de la UCI [15], cuyas características se pueden ver en la TABLA I.

El propósito de clasificar estos 6 problemas es comprobar la capacidad de precisión que tienen los modelos de red que conforman cada ensemble, obteniendo el máximo compromiso posible entre la precisión y la capacidad de clasificación en cada una de las clases que componen un determinado problema. En la TABLA II podemos observar los resultados obtenidos para cada uno de los métodos propuesto. En la Figura 3 podemos ver gráficamente los resultados obtenidos para cada conjunto de datos. En las gráficas de entrenamiento mostramos los frentes de Pareto obtenidos, siendo la entropía y la sensitividad los objetivos que guían al algoritmo. En las gráficas de generalización mostramos los valores de precisión C y de sensitividad S sobre el conjunto de test asociados a los individuos existentes en entrenamiento. Estos valores, obviamente, no forman un frente de Pareto en generalización al no haber una correspondencia directa entre Entropía y C. Con las gráficas de generalización se observa de una manera rápida y sencilla como los objetivos intentan confluir hacia el óptimo (1,1), siendo esto extremadamente complejo debido a la disminución del espacio de búsqueda cuando los objetivos llegan a un determinado nivel en el que son contrapuestos entre sí.

VI. CONCLUSIONES

El principal objetivo de este trabajo es tratar problemas multiclasificación considerando conjuntamente las medidas S y C, y determinar, mediante el uso de *ensembles*, una solución que involucre a las distintas soluciones no dominadas que el manejo de dos objetivos necesariamente conlleva.

Las pruebas efectuadas con tres métodos de combinación (Majority Voting, Simple Averaging y Winner-Takes-All) parecen señalar una cierta superioridad de estos dos últimos métodos, ambos basados en el manejo directo de la función de asignación y no en su resultado. Y de entre ellas, la combinación mediante promedio parece una selección menos apropiada que la basada en los valores dominantes, resultado previsiblemente relacionado con la distribución de probabilidad de los valores promedidados.

Estos ensembles resultan ciertamente interesantes en problemas de medicina y microbiología, donde un balanceo en la clasificación de cada clase es primordial.

AGRADECIMIENTOS

Este trabajo ha sido financiado en parte por el proyecto TIN2005-08386-C05-02 de la Comisión Interministerial de Ciencia y Tecnología y fondos FEDER y el proyecto de excelencia de la Junta de Andalucía P08-TIC-3745. La investigación de P.A. Gutiérrez y J.C. Fernández ha sido financiada respectivamente por el programa predoctoral FPU (referencia AP-2006-01746) y el programa predoctoral FPI (referencia BES-2006-12543) del Ministerio de Educación y Ciencia.

REFERENCIAS

[1] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[2] S. Haykin, *Neural Networks: A comprehensive Foundation*, Prentice Hall, 2nd edition, 1998.

[3] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Reading, 1989.

[4] K. Saito, N. Ueda, S. Katagiri, Y. Fukai, H. Fujimaru and M. Fujinawa, "Law discovery from financial data using neural networks," *Proceedings of the IEEE/IAFE/INFORMS 2000, conference on Computational Intelligence for Financial Engineering (CIFEr)*, vol. 209-212, 2000.

[5] A. Carpintero, "Weather forecasting with adaptive time-delay neural networks: A case study," *Proc. Proceedings of the International Conference on Neural Network Processing*, pp. 842-846, 1994.

[6] X. Yao, "Evolutionary Artificial Neural Networks," *International Journal of Intelligent Systems*, vol. 4, no. 5, pp. 203-222, 1993.

[7] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 694-713, 1997.
[8] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, LTD, 2004.
[9] Y. Jin and B. Sendhoff, "Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies," *IEEE Transaction on Systems, Man and Cybernetics, Part. C: Applications and reviews*, vol. 38, no. 3, pp. 397-415, 2008.
[10] H.A. Abbass, R. Sarker and C. Newton, " PDE: a Pareto-frontier differential evolution approach formulti-objective optimization problems," *Proc. Proceedings of the 2001 Congress on Evolutionary Computation*, 2001.
[11] Y. Jin, B. Sendhoff and E. Körner, "Evolutionary Multi-Objective Optimization for Simultaneous Generation of Signal-Type and Simbol-Type Representations," *Proc. Lectures Notes in Computer Science, EMO 2005*, Springer- Verlag, pp. 752-766, 2005.
[12] A.P. Braga, R.H.C. Takahashi, M.A. Costa and R.A. Teixeira, "Multi-objective Algorithms for Neural Networks Learning," *Studies in Computational Intelligence*, vol. 16, no. 151-171, 2006.
[13] F.J. Martínez-Estudillo, P.A. Gutiérrez, C. Hervás and J.C. Fernández, "Evolutionary Learning by a Sensitivity-Accuracy Approach for Multi-class Problems " *Proc. IEEE World Congress on Computational Intelligence, CEC 08*, pp. 1581-1588, 2008.
[14] C. Igel and M. Hüsken, "Empirical evaluation of the improved Rprop learning algorithms," *Neurocomputing*, vol. 50, no. 6, pp. 105-123, 2003.
[15] A. Asuncion and D.J. Newman, "UCI Maching Learning Repository", http://www.ics.uci.edu/~mlearn/MLRepository.html, Irvine, CA: University of California, School of Information and Computer Science, 2007.
[16] F. Provost and T. Fawcett, "Analysis and visualization of the classifier performance: comparison under imprecise class and cost distribution," *Proc. Proceedings of the Third International Conference on Knowledge Discovery (KDD97) and Data Mining*, AAAI Press, pp. 43-88, 1997.
[17] F. Provost and T. Fawcett, "Robust classification system for imprecise environments," *Proc. Proceedings of the Fifhteenth National Conference on Artificial Intelligence*, pp. 706-713, 1998.
[18] T.K. Ho and M. Basu, "Complexity Measures of Supervised Classification Problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 289 - 300, 2002.
[19] H. Abbass, "Speeding Up Backpropagation Using Multiobjective Evolutionary Algorithms," *Neural Computation*, vol. 15, pp. 2705-2726, 2003.
[20] C.R. Houck, J.A. Joines, M.G. Kay and J.R. Wilson, "Empirical investigation of the benefits of partial lamarckianism," *Evolutionary Computation*, vol. 5, pp. 31-60, 1997.
[21] W. Yan, Z.Z. Z and R. Hu, "Hybrid genetic/BP algorithm and its application for radar target classification," *Proc. Proceedings of the 1997 IEEE National Aerospace and Electronics Conference, NAECON*, IEEE Press, pp. 981-984, 1997.
[22] G. Ou and Y.L. Murphey, "Multi-class pattern classification using neural networks," *Pattern Recognition*, vol. 40, pp. 4-18, 2007.
[23] K. Deb, A. Pratab, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA2," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.
[24] P.J. Angeline, G.M. Saunders and J.B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 54-65, 1994.
[25] A.C. Martínez-Estudillo, C. Hervás-Martínez, F.J. Martínez-Estudillo and N. García, "Hybridation of evolutionary algorithms and local search by means of a clustering method," *IEEE Transaction on Systems, Man and Cybernetics, Part. B: Cybernetics*, vol. 36, no. 3, pp. 534-546, 2006.
[26] A.C. Martínez-Estudillo, F.J. Martínez-Estudillo and C. Hervás-Martínez, "Evolutionary Product Unit based Neural Networks for Regression," *Neural Networks*, vol. 19, no. 4, pp. 477-486, 2006.
[27] M. Riedmiller and H. Braun, "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm," *Proc. Neural Networks, IEEE International Conference*, pp. 586-591, 1993.

TABLA I

CARACTERÍSTICAS PARA LAS BASES DE DATOS OBTENIDAS DE LA UCI

DATASET	#PATTERNS	#TRAINING PATTERNS	#TEST PATTERNS	#INPUT VARIABLES	#CLASSES	#PATTERNS PER CLASS	P*
AUSTRALIANC	690	517	173	51	2	307-383	0.44
BALANCE	625	469	156	4	3	288-49-288	0.06
BREASTC	286	215	71	15	2	201-85	0.29
GERMAN	1000	750	250	61	2	700-300	0.30
IONOSPHERE	351	263	88	34	2	126-225	0.36
PIMA	768	576	192	8	2	500-268	0.34

TABLA II

RESUMEN DE RESULTADOS COMPARANDO EL RENDIMIENTO DE LAS TRES METODOLOGÍAS DE ENSEMBLE.

METHOD/REFERENCE	DATASET											
	BALANCE		BREASTC		GERMAN		IONOSPHERE		AUSTRALIANC		PIMA	
	C(%)	*S*(%)	*C*(%)	*S*(%)	*C*(%)	*S*(%)	*C*(%)	*S*(%)	*C*(%)	*S*(%)	*C*(%)	*S*(%)
MV_ENSEMBLE	51.64	12.68	66.47	40.15	71.41	60.35	91.51	80.72	87.66	86.29	77.04	65.77
SA_ENSEMBLE	**88.93**	17.33	69.01	**41.11**	74.18	**61.28**	92.57	82.70	87.86	86.73	78.05	**66.51**
WT_ENSEMBLE	84.50	**64.44**	**69.90**	26.19	**74.62**	41.33	**92.84**	**84.79**	**88.38**	**87.40**	**78.29**	55.92

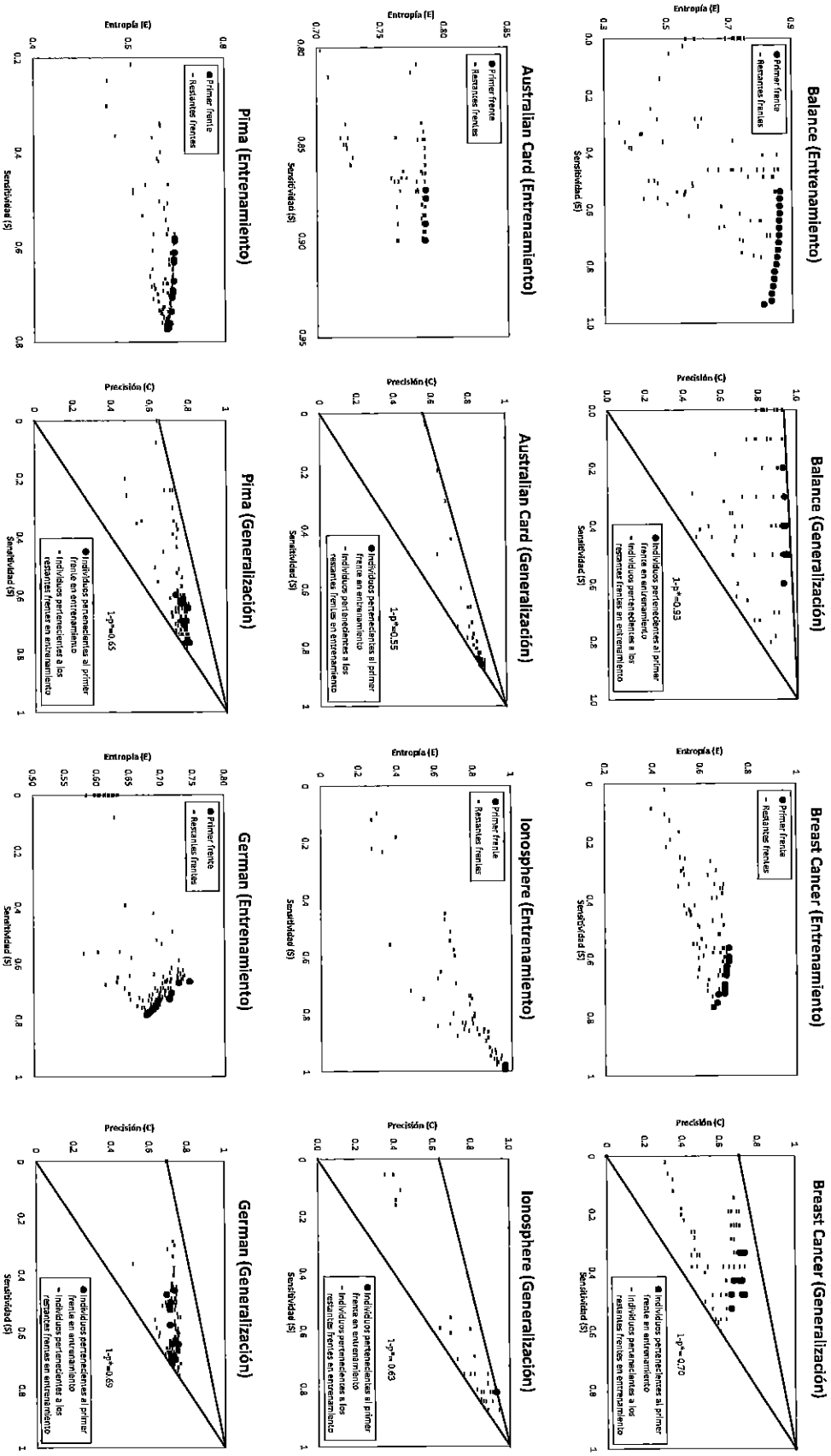

Fig. 3. Frente de Pareto en entrenamiento (E, S) y valores asociados en generalización (C, S) para Balance, Breast Cancer, Australian Card, Ionosphere, Pima y German en una ejecución específica.