Redes Neuronales Artificiales y Distribuciones no Balanceadas

R. Alejo [1,2], V. García[1], F. García[2] y M. G. de la Rosa[2]

[1] Dept. Llenguatges i Sistemes Informátics, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana (España)
[2] Centro Universitario Atlacomulco, Universidad Autónoma del Estado de México
Carretera Toluca-Atlacomulco Km. 60 (México)

Resumen. En investigaciones recientes, el problema de la distribución no balanceada de las clases, ha sido reconocido como un factor crítico en el desempeño del clasificador. Es común, que la comunidad científica se centre en problemas de dos clases y pocos trabajos aborden este problema en dominios de múltiples clases. En este trabajo, se estudian alternativas distintas para reducir la influencia del desbalance de la distribución de las clases cuando se trabaja con bases de datos de múltiples clases y redes neuronales entrenadas con el algoritmo back-propagation. Estas estrategias se basan en la modificación del algoritmo de aprendizaje para mejorar la capacidad de generalización de la red y acelerar su proceso de convergencia. Además se incluye una comparación entre el desempeño de Redes Modulares y No Modulares entrenadas con distribuciones no balanceadas.

1 Introducción

La entrada a una Red Neuronal Artificial (RNA) 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. De manera formal se puede definir como $ME = ME_1 \cup ME_2 \cup ... \cup ME_m$, donde $ME_i = (\mathbf{x}_j, \varphi(\mathbf{x}_j)), j = 1, ..., n_i, \mathbf{x}_j = [x_1, x_2, ..., x_d]^T$ es el vector de características que identifica una situación particular; $\varphi(x_j)$ la clase a la que corresponde y n_i el número de patrones de la clase i.

Generalmente, los métodos de aprendizaje supervisado, como las RNA, están diseñados para trabajar con MEs razonablemente balanceadas [1], es decir, donde la diferencia en el número de patrones de las distintas clases no es considerable. Sin embargo, existen numerosas aplicaciones donde la desproporción en el número de patrones entre clases es significativa [2]. Por ejemplo, en la detección de fraudes en llamadas telefónicas [3], o en transacciones con tarjetas de crédito [4].

M.A Cruz-Chávez, J.C. Zavala-Díaz (Eds): CICos 2008, ISBN:978-607-00-0165-9. pp. 77-88, 2008

Una ME no balanceada, es aquella donde la diferencia en el número de patrones de las distintas clases es considerable, es decir, si para alguna ME_i se cumple que $\| \text{ME}_i \| << \| \text{ME}_j \| \; i \neq j; \; i, j = 1, ..., m,$ donde m es el número total de clases en la ME .

Recientemente, el problema del desbalance de la *ME* , ha sido considerado un problema crítico en la minería de datos y el aprendizaje automático [5].
Numerosos estudios han sido dirigidos a mejorar el desempeño del clasificador cuando se entrena con *MEs* no balanceadas. En el contexto de las RNA del tipo Perceptron Múlticapa (PM), entrenadas 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, de forma que la convergencia de la clase minoritaria es muy lenta [6, 7].

En [6], se analiza al algoritmo *back-propagation* y se propone su modificación para acelerar el proceso de convergencia de la red. La idea esta centrada en el cálculo del vector gradiente y su dirección, de tal forma, que permita que el error pueda decrecer en dirección de ambas clases y se evite que la clase minoritaria sea ignorada. El estudio esta limitado a problemas de dos clases. Más adelante en [8] se extiende este trabajo a problemas de múltiples clases y redes modulares. En este caso, el problema de múltiples clases es descompuesto a problemas de dos clases y cada subproblema (de dos clases) es resulto por una red idéntica a la de la propuesta inicial de [6]. Posteriormente las salidas de las diferentes RNA (expertos) son combinadas linealmente para producir una salida global de la red (ver Fig. 1).

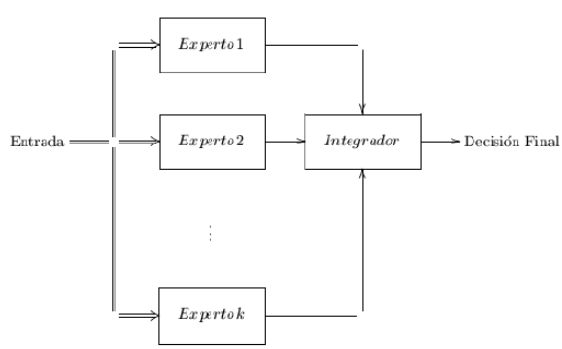

Fig. 1. Esquema simplificado de una red de módulos o red modular. Cada experto corresponde a una RNA. Los expertos son entrenados con los mismos datos, pero pueden diferir entre sí en las condiciones iniciales del proceso de entrenamiento.

Por otra parte, en [7] se expone una alternativa para modificar el algoritmo *back-propagation* y acelerar su convergencia. Esta modificación consiste básicamente en incluir una función de costo en el algoritmo de entrenamiento y disminuir su valor a partir de una estrategia heurística, de tal forma que se reduzca su impacto en la

probabilidad de la distribución de los datos. La principal ventaja respecto a [8] es que no es necesario descomponer el problema en subproblemas de dos clases.

Las estrategias más estudiadas para enfrentar el problema del desbalance en la *ME* , han sido las técnicas de sub-muestreo (*under-sampling*) o sobre-muestreo (*over-sampling*) [9]. En el caso particular de las RNAPM, las técnicas de *under-sampling* o de *over-sampling* han mostrado notables mejoras en el desempeño del clasificador [2]. No obstante, al eliminar patrones de la *ME* se puede perder información relevante o si se incrementa el tamaño de la clase minoritaria, puede introducirse ruido en la misma y el tiempo de entrenamiento es incrementado. Además, la distribución de los datos puede verse afectada [10]. En algunos trabajos [5], el problema del desbalance en la *ME* , se plantea como un problema de costo sensitivo (*cost-sensitive*), es decir, un enfoque donde el precio de cometer un error de clasificación debe ser distinto para cada clase [11]. El principal inconveniente de este enfoque esta en la necesidad de contar con información a priori sobre el problema en cuestión, es decir, de antemano se debe cuantificar el costo de cometer cada error.

En este trabajo, se presenta un estudio empírico comparativo de diferentes propuestas diseñadas para tratar el problema del desbalance de los datos en dominios de múltiples clases y RNA entrenadas con el algoritmo *back-propagation* en "batch-mode". Además, se analiza la idea de simplificar los problemas de múltiples clases transformándolos a subproblemas de dos clases (Redes Modulares).

2 Redes Neuronales

Es muy popular, en la actualidad, el empleo de las RNA, en particular los modelos RNAPM y RNAFBR (RNA de Funciones de Base Radial), para tareas de reconocimiento de patrones, minería de datos y aprendizaje automático.

Son ejemplos claros de redes de propagación hacia adelante (*feedforward*) con capas no lineales. Ambas redes, son consideradas aproximadores universales [12] y pueden ser entrenadas con métodos similares de descenso por gradiente (por ejemplo, con el algoritmo *back-propagation*) [13]. Sin embargo, ambas redes presentan importantes diferencias [14]. Las RNAFBR tienen una sola capa oculta y las RNAPM pueden tener una o más. Generalmente, en las RNAPM los nodos ocultos y los de salida tienen el mismo modelo neuronal, mientras que en las RNAFBR, el modelo neuronal de la capa oculta y la de salida es distinto. Las RNAPM generan una aproximación global de la relación no lineal entrada-salida en tanto que en las RNAFBR esa relación es local. La principal diferencia entre las RNAFBR y las RNAPM esta en la función de activación de los nodos ocultos. En las RNAFBR depende de la distancia entre los vectores de entrada y los centros de la red, mientras que en la RNAPM depende del producto del vector de entrada y el vector de pesos.

El modelo RNARVFLN, corresponde a una particularización del modelo RNAFBR. A este último de le agrega una capa adicional representada por el vector aleatorio FLN (Functional Link Net) [15], con la finalidad de incrementar la capacidad de aprendizaje del modelo neuronal.

3 Análisis del error en clases no balanceadas

Estudios empíricos realizados al algoritmo *back-propagation* [6, 8], 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 RNA, es decir, la mayor parte de las aportaciones al MSE están dadas por la clase mayoritaria. En consecuencia, el entrenamiento de la red es dominado por esta última.

Consideremos una ME de dos clases (m = 2) con N patrones de entrenamiento, $N = \sum_i^m n_i$ tal que n_i sea el número de patrones de la clase i .Supongamos que el MSE por clase puede ser expresado como

$$E_i(U) = \frac{1}{N} \sum_{n=1}^{n_i} \sum_{p=1}^{L} (y_p^n - F_p^n)^2, \quad (\mathbf{1})$$

de tal forma el MSE global puede ser expresado como

$$E(U) = \sum_{n=1}^{n_i} E(U)_i = E_1(U) + E_2(U) \; . \quad (\mathbf{2})$$

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 [6].

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. 2) por parte de las clases, se puede considerar la inclusión de una función de costo (γ) al algoritmo, que compense el desequilibrio de las clases de la *ME* como se muestra a continuación

$$E(U) = \sum_{m=1}^{M} \Psi(m) E(U)_m \; = \; \Psi(1) E_1(U) + \Psi(2) E_2(U) \; , \quad (\mathbf{3})$$

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

En este trabajo se estudian las siguientes funciones de costo:

– Opción 0: $\Psi(m) = 1$, es decir, el algoritmo *back-propagation* sin ninguna modificación.

– Opción 1: $\Psi(m) = n_{\max} / n_m$; donde $m = 1$, ...,M; M es el total de clases y n_{max} es el número de patrones de la clase mayoritaria.

– Opción 2: $\Psi(m) = N / n_m$; donde $m = 1$, ...,M y N es el número total de patrones.

– Opción 3: $\Psi(m) = \frac{\|\nabla E_{\max}(U)\|}{\|\nabla E_m(U)\|}$, donde $\| \nabla E_{\max}(U) \|$ corresponde a la clase mayoritaria. Esta función es una simplificación de la propuesta de [8].

Sin embargo, teóricamente es posible demostrar que incluir la función de coste $\Psi(m)$ altera la probabilidad de la distribución de los datos [10]. Por lo tanto, para reducir el impacto de la función de costo $\Psi(m)$, en la probabilidad de la distribución de los datos, se propone disminuir el valor de la función de costo gradualmente (**Opción 4**) [13]. La reducción de la función de costo se realiza de la siguiente forma

$$\Psi(m)^t = \begin{cases} \Psi(m)^{(t-1)}(1-\varepsilon) & \text{Si } \left[\Psi(m)^{(t-1)}\right] > 1 \\ 1 & \text{en otro caso} \end{cases} \quad (4)$$

donde t es la actual iteración y ε = 0, ..., 1. Así, en las primeras iteraciones el efecto del desbalance de la ME es disminuido y en las últimas la función de costo $\Psi(m)$ reduce su valor para evitar alterar la probabilidad de la distribución de los datos.

4 Resultados experimentales

Los experimentos fueron realizados con la base de datos de múltiples clases Cayo. Corresponde a una imagen de percepción remota, representa un cayo de una región cercana al golfo de México, y esta caracterizada por 4 atributos y 11 clases distribuidas de la siguiente forma: 838-293-624-322-133-369-324-722-789-833-772. Fue divida con el método H (50% entrenamiento y 50% prueba).

Observe que la base de datos Cayo presenta diferentes niveles de desbalance en la *ME*. Van desde un desbalance moderado hasta un desbalance severo en el mismo conjunto de datos. Por ejemplo, la clase 1 y 5 presentan desbalance severo entre sí (la clase mayoritaria tiene 838 patrones y la minoritaria solo tiene 133), mientras que las clases 8, 9 y 10 no muestran problemas de desbalance. Un desbalance moderado puede ubicarse en la clase 3.

Las RNA usadas en esta investigación, fueron entrenadas con el algoritmo *backpropagation* con procesamiento por lotes o "batch mode". Se implementaron en el lenguaje de programación Java como una simplificación de las que aparecen en el texto [12]. La razón de aprendizaje fue fijada en 0.0001 para RNAFBR y RNARVFLN, y en 0.9 para las RNAPM. En esta última solo se utilizó una capa oculta. El número de neuronas para la capa oculta se estableció en 16 para las redes no modulares. Los módulos que entrenaron problemas de dos clases siguieron la configuración anterior, con la excepción de que el número de neuronas estableció en 4.

El desempeño del clasificador generalmente es medido según el porcentaje de clasificación correcta. No obstante, existen situaciones donde no puede ser considerada una medida adecuada, principalmente cuando la *ME* esta desbalanceada [9]. Uno de los criterios de medida más ampliamente aceptados en problemas desbalance es la media geométrica (*g-mean*) [13].

El denominado coeficiente *Kappa*, es muy utilizado como parámetro de calidad. Su valor da una idea del porcentaje de acuerdo obtenido en la clasificación una vez se ha

eliminado la parte que se debería al azar. En este trabajo se utilizan estas medidas como criterios de evaluación.

4.1 Resultados experimentales con Redes No Modulares

Una visión sobre el comportamiento de los modelos de RNA no modulares sobre las distintas clases es presentada en la Tabla 1. La primera columna indica la clase a la que se hace referencia, mientras que en las siguientes se presentan los porcentajes de aciertos por clase, obtenidos al aplicar las estrategias Opción 0, 1, 2, 3 y 4. En la Tabla 2, se resumen algunos de los resultados globales generados al experimentar con la base de datos Cayo. La primera columna indica los criterios de evaluación aplicados y las columnas restantes presentan los valores obtenidos por las estrategias aplicadas. Los valores entre paréntesis representan la desviación estándar.

El conjunto de datos Cayo evidencia el bajo rendimiento del clasificador cuando es entrenado con bases de datos desbalanceadas. En la Tabla 2, se observa que el valor de g-mean es cero en los tres modelos de red. Sin embargo, la precisión global es considerable. Esto se debe a que las clases que no alcanza a identificar correctamente el clasificador, corresponden a clases minoritarias y por tanto repercuten en menor medida en el desempeño global del clasificador.

En los modelos RNAPM y RNAFBR, la clase menos representada (clase 5) es ignorada en el proceso de entrenamiento y consecuentemente la precisión de esa clase durante la fase de clasificación es de 0.0 (ver Tabla 1). Esto se traduce en valores de *g-mean* iguales a cero.

En el caso de las RNARVFLN, la clase 5 presenta niveles considerables de error, sin embargo, la clase que origina que el valor de *g-mean* sea cero es la clase 4 (ver Tabla 1), que también esta incluida en el grupo de clases consideradas minoritarias (clases 2, 4, 5, 6 y 7).

Así mismo, si se analizan los resultados de la Tabla 1 se observa que las clases 2, 4, 5 y 6 (incluso la clase 7 en el modelo RNAFBR) presentan bajos niveles de clasificación correcta y coincide con el hecho de que se trata de clases minoritarias.

Obsérvese en la Tabla 2, que en los tres modelos de red todas las estrategias logran resultados importantes al incrementarse el valor de la *g-mean* en al menos 70%. De igual manera estas estrategias logran incrementar la precisión global del clasificador en al menos 2% (en algunos casos se incrementa hasta en 8% aproximadamente, vea el modelo RNAPM, Opción 2). La confiabilidad de la clasificación (coeficiente kappa) también es incrementada.

Las mejoras en el clasificador son consecuencia de que la precisión de las clases minoritarias es incrementada cuando las funciones de costo son aplicadas en la fase de entrenamiento y consecuentemente en la fase de clasificación son identificadas correctamente. En la Tabla 2, se observa que en todos los casos las precisiones de las clases 2, 4, 5, 6 y 7 son incrementadas (o no son afectadas considerablemente, vea la clase 7 con el modelo RNAPM Opción 4) después de que se aplicaron las funciones de costo.

Es importante comentar que en algunos casos las precisiones de las clases mayoritarias se ven afectadas por las estrategias aplicadas (véase en la Tabla 1: los modelos RNAFBR y RNARVFLN, las clases 10 y 11; en las RNAPM la clase 10 con

las Opciones 2, 3 y 4; en RNAFBR la clase 11 con las Opciones 1 y 4; y en RNARVFLN, la clases 3, 8 y 10).

Se observa que en la base de datos Cayo (Tabla 2), el modelo RNAPM presenta un mejor desempeño que los modelos RNAFBR y RNARVFLN, además de mostrar una mayor tolerancia al desbalance de la ME .

Los mejores resultados se obtuvieron con las estrategias Opción 2 y 3, mientras los peores fueron generados por la estrategia Opción 4. Sin embargo, los resultados de Opción 4 en relación a Opción 0 siempre fueron mejores.

4.2 Resultados experimentales con Redes Modulares

La idea de descomponer un problema de múltiples clases a subproblemas de dos clases fue propuesta y evaluada en [8] con el modelo RNAPM. En este trabajo, se extiende a los modelos RNAFBR y RNARVFLN, y a bases de datos de múltiples clases no balanceadas. Además de evaluarse diferentes funciones de costo (Opción 1, 2, 3 y 4) diseñadas para compensar la desproporción de las clases en la ME .

Es necesario reiterar, que el problema de múltiples clases es descompuesto a subproblemas de dos clases y cada subproblema es resulto por una RNA independiente. Posteriormente las salidas de las diferentes RNA (expertos) son combinadas linealmente para producir una salida global de la red (ver Fig. 1).

La Tabla 4 muestra los resultados globales obtenidos en la fase de experimentación con redes modulares. La primera columna indica los criterios de evaluación aplicados y las columnas restantes presentan los valores producidos por las estrategias aplicadas. En la Tabla 3 se ilustra el porcentaje de acierto por clase. La primera columna muestra la clase a la que se hace referencia, mientras que las siguientes a las estrategias evaluadas. Los valores entre paréntesis representan la desviación estándar.

En la Tabla 3, se observa que las clases en las que el clasificador presenta deficiencias en su desempeño, corresponden a las clases 2, 4, 5 y 6 (además de la clase 7 en el modelo RNAPM), las cuales se pueden considerar minoritarias.

Es notable el incremento en la precisión del clasificador sobre esas clases cuando son aplicadas las estrategias Opción 1, 2, 3 y 4. La Opción 1 incrementa en el peor de los casos al menos un 10% (clase 2, modelo RNARVFLN) mientras que en el mejor caso llega a aumentar la precisión de la clase desfavorecida hasta en 94% (clase 4, modelo RNAPM). Un comportamiento similar se observa en el resto de las Opciones, aunque existen algunas excepciones (RNARVFLN, clase 6). Sin embargo, en ningún modelo u Opción se llega a reducir la precisión de las clases consideradas minoritarias (excepto en RNARVFLN, clase 6).

A diferencia de lo que ocurrió con la base de datos Cayo y redes no modulares, las clases consideradas mayoritarias fueron afectadas en menor medida por las estrategias. Las clases que fueron más afectadas son la clase 10 con el modelo RNAPM y RNARVFLN y la clase 3 con el clasificador RNAFBR (ver Tabla 3).

Estos resultados se tradujeron en mejoras en el desempeño global del clasificador (ver Tabla 4). Se observa en la Tabla 4, que todas las estrategias mejoraron o no perjudicaron la precisión global del clasificador en los tres modelos de RNA evaluados.

Tabla 1. Presición por clase obtenida al experimentar con la base de datos Cayo.

RNAPM

Clase	Op.0	Op.1	Op.2	Op.3	Op.4
1	90.1(0.16)	85.45(1.35)	91.17(2.02)	88.55(0.33)	89.26(0.34)
2	43(11.38)	51.54(0.23)	82.93(1.05)	51.2(0.25)	51.2(0.25)
3	98.4(0.01)	92.95(0.03)	75.96(0.11)	93.75(0.20)	88.47(1.76)
4	17.4(7.9)	95.96(0.44)	95.65(0.9)	93.48(0.44)	82.3(1.32)
5	0.0(0.0)	90.26(5.2)	92.49(2.06)	69.18(0.74)	60.18(4.9)
6	48.5(1.7)	61.3(0.23)	60.98(0.15)	60.16(0.54)	62.6(0.14)
7	96(2.2)	96.30(0.83)	96.91(0.91)	95.37(0.40)	93.22(0.81)
8	99.6(0.19)	97.37(0.20)	96.68(0.78)	94.74(0.78)	96.68(1.18)
9	87.6(0.02)	87.58(0.02)	87.58(0.02)	87.58(0.02)	87.07(0.69)
10	82.59(1.6)	80.08(0.04)	75.99(2.42)	78.87(1.74)	63.02(5.50)
11	79.4(3.1)	88.60(1.47)	94.30(0.37)	90.16(0.73)	72.67(23.63)

RNARBF

Clase	Op.0	Op. 1	Op. 2	Op. 3	Op. 4
1	89.50(1.01)	81.27(0.50)	83.89(0.17)	87.71(0.17)	88.55(0.33)
2	47.46(5.54)	49.84(2.17)	51.20(0.25)	34.87(21.40)	51.20(0.25)
3	98.72(0.01)	86.36(6.18)	94.56(2.24)	96.16(0.02)	80.61(1.68)
4	1.56(2.20)	85.41(3.08)	73.61(18.89)	64.29(6.58)	74.54(2.64)
5	0.00(0.00)	85.00(6.22)	83.46(0.18)	48.88(1.58)	54.86(6.83)
6	55.56(2.14)	58.53(2.84)	61.25(0.53)	58.01(4.05)	62.33(2.06)
7	51.46(51.96)	93.51(2.23)	95.69(1.70)	93.53(3.00)	93.50(5.73)
8	99.31(0.20)	95.57(0.39)	97.09(0.20)	90.58(3.92)	95.02(6.27)
9	87.58(0.02)	86.44(1.58)	87.58(0.02)	87.58(0.02)	87.08(0.02)
10	72.14(16.34)	72.16(9.80)	67.25(16.07)	72.02(10.41)	70.59(2.93)
11	89.38(7.33)	68.01(40.11)	86.92(2.38)	93.92(2.02)	65.93(0.55)

RNARVFLN

Clase	Op. 0	Op. 1	Op. 2	Op. 3	Op. 4
1	89.03(1.35)	81.87(1.01)	83.89(3.87)	85.20(2.02)	85.21(4.05)
2	13.27(18.76)	50.17(0.24)	36.23(22.37)	50.18(1.69)	51.20(0.25)
3	98.72(0.46)	88.00(5.15)	90.88(5.63)	94.14(6.09)	81.25(3.49)
4	0.00(0.00)	51.24(30.31)	77.95(12.74)	45.03(44.36)	47.52(39.97)
5	23.44(24.71)	85.00(6.22)	84.23(3.02)	60.19(7.02)	73.66(5.59)
6	58.27(0.61)	62.87(1.29)	62.87(1.29)	54.20(2.09)	58.26(4.76)
7	79.53(22.88)	97.53(0.90)	86.04(16.28)	95.37(0.40)	96.30(0.04)
8	99.45(0.00)	94.88(0.19)	93.35(9.01)	97.65(1.76)	92.25(9.79)
9	87.58(0.02)	87.58(0.02)	87.58(0.02)	87.58(0.02)	87.07(0.74)
10	75.86(10.73)	71.91(1.41)	62.06(10.25)	76.59(4.29)	68.31(1.64)
11	65.80(30.77)	64.38(10.08)	85.88(18.13)	92.23(6.60)	68.40(15.03)

En RNAPM las Opciones 1, 2, 3 y 4 incrementaron la precisión global en aproximadamente 10%. Mientras que en RNAFBR, se aumento entre 1% y 3%. El modelo RNARVFLN en la Opción 1 y 4 mantuvo el porcentaje de aciertos totales en 83% y en la Opción 2 y 3 solo alcanzo a mejorar en aproximadamente 1%.

En cuanto a valores de g-mean las RNAPM fueron las más beneficiadas al mejorar su valor en al menos 78%. El modelo RNAFBR y RNARVFLN en promedio: 20% y 6.5% respectivamente.

Tabla 2. Resultados globales obtenidos de la experimentación realizada con Cayo.

RNAPM	Op. 0	Op. 1	Op. 2	Op. 3	Op. 4
Presición G.	78.95(1.5)	85.71(0.04)	86.41(0)	85.2(0.5)	79.73(3.8)
g-mean	0(0)	82.86(0.3)	85.7(0.02)	80.6(0.3)	75.3(2.5)
Kappa	0.76(0.02)	0.84(0)	0.85(0)	0.83(0.01)	0.77(0.04)

RNAFBR	Op. 0	Op. 1	Op. 2	Op. 3	Op. 4
Presición G.	76.09(4)	79.3(2.5)	82.24(2.6)	81.35(0.1)	78.34(0.76)
g-mean	0(0)	76.27(1.9)	78.6(3.04)	71.3(4.3)	73.76(0.9)
Kappa	0.73(0.04)	0.77(0.03)	0.8(0.03)	0.79(0)	0.76(0.01)

RNARVFLN	Op. 0	Op. 1	Op. 2	Op. 3	Op. 4
Presición G.	74(3.26)	77.8(0.5)	79.7(2.5)	81.91(2.2)	76.4(1.6)
g-mean	0.0(0.0)	73.5(2.9)	74.25(6.5)	71.6(8.32)	70.33(4.4)
Kappa	0.71(0.04)	0.75(0)	0.77(0.03)	0.8(0.02)	0.74(0.02)

La confiabilidad en la clasificación se incremento en promedio de 0.12%, 0.03% y 0.012 para RNAPM, RNAFBR y RNARVFLN. Es notable que el clasificador RNAPM, fue el que obtuvo mayores beneficios al aplicar las Opciones 1, 2, 3 y 4.

Se observa, que a pesar de incrementar el desbalance entre clases, al transformar problemas de múltiples clases a subproblemas de dos clases, el desempeño global del clasificador no es perjudicado como habría de esperarse. En el conjunto de datos Cayo (Tabla 4) se aprecia una mayor tolerancia al desbalance de la *ME* en los modelos RNAFBR y RNARVFLN, que en el clasificador RNAPM (véase la Opción 0 de los modelos RNAFBR y RNARVFLN, respectivamente).

Es interesante notar que según los resultados presentados en esta sección es difícil identificar cual de las Opciones presenta un mejor rendimiento. Se evidencia que las Opciones 1, 2, 3 y 4 logran mejorar la precisión de las clases menos representadas (véanse la Tabla 3).

En algunas ocasiones las Opciones reducen el desempeño del clasificador sobre las clases mayoritarias, sin embargo, este deterioro no logra mermar de forma considerable el rendimiento global del clasificador.

En los modelos RNAFBR y RNARVFLN, se aprecian mejores resultados en precisión global, valores de g-mean y confiabilidad en la clasificación, cuando el problema es dividido en subproblemas de dos clases.

Tabla 3. Precisión por clase obtenida con Cayo (Redes Modulares).

RNAPM

Clase	Op.0	Op.1	Op.2	Op.3	Op.4
1	90.21(0.00)	87.35(0.68)	87.71(0.17)	89.03(1.35)	87.71(0.17)
2	2.05(0.01)	52.89(3.13)	54.25(5.05)	45.05(0.75)	48.80(2.18)
3	99.04(0.00)	90.70(2.31)	90.22(3.90)	93.11(0.25)	92.95(0.49)
4	0.00(0.00)	94.10(0.44)	93.79(0.88)	95.65(1.75)	96.95(1.32)
5	0.00(0.00)	80.47(4.04)	79.73(5.10)	67.64(5.66)	78.23(7.21)
6	38.21(3.30)	53.93(2.48)	53.66(2.86)	54.48(2.13)	48.25(2.48)
7	50.10(21.39)	99.39(0.01)	98.14(1.77)	93.23(4.31)	94.75(0.48)
8	99.59(0.19)	96.96(0.39)	96.82(0.59)	97.65(0.19)	97.09(0.20)
9	87.58(0.02)	87.33(0.02)	87.45(0.20)	87.58(0.02)	87.58(0.02)
10	83.07(0.88)	66.50(3.28)	67.94(4.64)	75.03(1.06)	74.67(1.23)
11	81.22(3.11)	91.45(2.56)	90.68(5.86)	85.62(0.55)	86.53(0.74)

RNARBF

Clase	Op.0	Op.1	Op.2	Op.3	Op.4
1	89.15(1.18)	88.79(1.35)	86.88(1.35)	91.53(6.58)	86.99(0.17)
2	51.20(0.25)	85.28(20.82)	84.01(20.68)	51.20(0.25)	80.17(14.58)
3	93.92(4.05)	61.33(20.80)	65.60(25.91)	91.99(1.85)	71.61(9.42)
4	50.62(57.53)	81.06(7.47)	91.93(5.27)	87.89(1.32)	92.55(1.76)
5	24.99(33.23)	78.97(4.03)	79.70(0.85)	71.44(1.82)	81.21(0.86)
6	41.77(17.02)	61.52(0.15)	52.58(0.97)	57.19(1.36)	57.73(4.44)
7	93.23(3.43)	96.30(0.04)	93.54(4.74)	95.38(1.27)	95.99(0.40)
8	98.48(1.76)	95.85(1.18)	96.40(1.17)	97.37(0.20)	96.82(0.19)
9	87.58(0.02)	87.58(0.02)	87.58(0.02)	87.20(0.51)	87.58(0.02)
10	69.02(7.86)	66.62(7.18)	70.59(0.56)	73.59(1.40)	72.27(2.26)
11	96.89(0.37)	92.36(2.75)	95.08(3.66)	86.92(3.85)	90.55(8.61)

RNARVFLN

Clase	Op.0	Op.1	Op.2	Op.3	Op.4
1	89.03(1.01)	90.58(8.94)	87.35(1.68)	94.51(1.68)	86.64(1.69)
2	50.17(1.20)	61.47(14.78)	69.64(0.25)	51.20(0.25)	71.27(28.14)
3	96.92(3.66)	82.55(6.94)	76.45(0.19)	91.83(0.19)	78.80(20.94)
4	74.85(6.58)	88.20(5.28)	95.03(7.03)	90.68(7.03)	87.58(1.75)
5	42.14(6.83)	73.66(5.59)	78.20(1.80)	68.43(1.80)	78.23(5.08)
6	56.91(1.32)	54.18(10.52)	58.54(0.61)	58.27(0.61)	60.16(2.06)
7	96.30(0.04)	96.60(1.34)	94.75(2.57)	95.07(2.57)	96.30(0.04)
8	98.34(1.17)	96.26(0.59)	97.51(0.59)	96.82(0.59)	96.54(0.20)
9	87.58(0.02)	87.58(0.02)	87.58(4.69)	84.28(4.69)	87.58(0.02)
10	79.11(6.48)	69.27(1.41)	67.83(2.93)	67.95(2.93)	71.55(4.63)
11	86.66(13.74)	94.30(5.50)	98.06(0.18)	97.28(0.18)	92.23(5.13)

5 Conclusiones y trabajos futuros

A partir de los resultados presentados se puede concluir lo siguiente: 1) Las RNA entrenadas con el algoritmo back-propagation con procesamiento por grupos, tienen la tendencia a ignorar a las clases menos representadas durante el proceso de entrenamiento y en consecuencia a mostrar desempeños deficientes sobre las clases minoritarias. Sin embargo, aunque se trate de clases muy minoritarias, existen

situaciones donde son clasificadas con un alto grado de acierto. Es probable que estas clases estén representadas con elementos que les permitan establecer más fácilmente las fronteras de decisión. 2) Las estrategias Opción 1, 2, 3 y 4 mejoran el desempeño de los modelos RNAPM, RNAFBR y RNARVFLN sobre las clases menos representadas en la *ME* , evitando que sean ignoradas en el proceso de entrenamiento y consecuentemente incrementado su rendimiento en la fase de clasificación. 3) En algunas situaciones las estrategias muestran una tendencia a afectar el rendimiento de las clases consideradas mayoritarias. No obstante, esta tendencia no logra mermar el desempeño global del clasificador. 4) Las redes modulares muestran un mejor rendimiento que las redes no modulares. Se observa, que a pesar de incrementar el desbalance entre clases, al transformar problemas de múltiples clases a subproblemas de dos clases, el desempeño global del clasificador no es perjudicado de forma considerable. 5) A partir de estos resultados es difícil identificar que estrategia logra el mejor desempeño del clasificador.

Tabla 4. Resultados obtenidos con Cayo (Redes Modulares).

RNAPM	Op. 0	Op. 1	Op. 2	Op. 3	Op. 4
Precisión G.	73.30(1.81)	83.62(0.04)	83.67(0.44)	83.77(0.03)	83.74(0.08)
g-mean	0.00(0.00)	80.12(0.32)	80.14(0.07)	78.24(0.57)	79.04(0.48)
Kappa	0.70(0.02)	0.82(0.00)	0.82(0.00)	0.82(0.00)	0.82(0.00)

RNAFBR	Op. 0	Op. 1	Op. 2	Op. 3	Op. 4
Precisión G.	80.74(2.88)	81.96(0.65)	82.93(1.79)	84.13(0.68)	83.60(0.92)
g-mean	59.83(22.47)	80.05(0.98)	80.53(1.03)	79.49(0.67)	82.06(0.81)
Kappa	0.78(0.03)	0.80(0.01)	0.81(0.02)	0.82(0.01)	0.82(0.01)

RNARVFLN	Op. 0	Op. 1	Op. 2	Op. 3	Op. 4
Precisión G.	83.39(0.94)	83.74(0.73)	84.12(0.04)	84.77(0.40)	83.79(0.46)
g-mean	74.88(0.99)	79.79(0.21)	81.72(0.30)	79.68(0.09)	81.21(1.16)
Kappa	0.81(0.01)	0.82(0.01)	0.82(0.00)	0.83(0.00)	0.82(0.00)

En conclusión, las estrategias propuestas mejoran el rendimiento de las RNA sobre las clases menos representadas en el conjunto de datos de entrenamiento y de esta forma evitan que sean ignoradas en el proceso de aprendizaje. Esto trae como consecuencia inmediata el incremento del rendimiento del clasificador sobre las clases minoritarias y el desempeño en general. Sin embargo, es necesario complementar este estudio relacionando el problema del desbalance de la *ME* con la complejidad de los datos (solapamiento, ruido o identificación de las fronteras de decisión) y aplicar técnicas estadísticas para la identificación de la mejor alternativa.

Agradecimientos

Este trabajo ha sido parcialmente soportado por los proyectos de investigación DPI2006-15542-C04-03 del CICYT (España) , SEP-2003-C02-44225 del CONACYT (México), y GV/2007/105 de la Generalitat Valenciana.

Referencias

1. S. Kotsiantis and P. Pintelas. Mixture of expert agents for handling imbalanced data sets. Annals of Mathematics and Computing & Tele-Informatics, 1(1):46–55, 2003.
2. N. Japkowicz and S. Stephen. The class imbalance problem: a systematic study. Intelligent Data Analysis, 6:429–449, 2002.
3. T. Fawcett and F. Provost. Adaptive fraud detection. Data Min. Knowl. Discov., 1(3):291–316, 1997.
4. P.K. Chan, W. Fan, A.L. Prodromidis, and S.J. Stolfo. Distributed data mining in credit card fraud detection. IEEE Intelligent Systems, 14(6):67–74, 1999.
5. Z.-H. Zhou and X.-Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering., 18:63–77, 2006.
6. R. Anand, K.G. Mehrotra, C.K. Mohan, and S. Ranka. An improved algorithm for neural network classification of imbalanced training sets. IEEE Transactions on Neural Networks, 4:962–969, 1993.
7. L. Bruzzone and S.B. Serpico. Classification of imbalanced remote-sensing data by neural networks. Pattern Recognition Letters, 18:1323–1328, 1997.
8. R. Anand, K. Mehrotra, C.K. Mohan, and S Ranka. Efficient classification for multiclass problems using modular neural networks. Neural Networks, IEEE Transactions on, 6(1):117–124, 1995.
9. R. Alejo, V. García, 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, pages 464–471, 2006.
10. Steve Lawrence, I. Burns, A.D. Back, A.C. Tsoi, and C. Lee Giles. Neural network classification and unequal prior class probabilities. In G. Orr, K.-R. M¨uller, and R. Caruana, editors, Neural Networks: Tricks of the Trade, volume 1524 of Lecture Notes in Computer Science, pages 299–314. Springer Verlag, 1998.
11. Matjaz Kukar and Igor Kononenko. Cost-sensitive learning with neural networks. In 13th European Conference on Artificial Intelligence, pages 445–449, 1998.
12. C. Looney. Pattern Recognition Using Neuronal Networks - theory and algorithms for engineers and scientists. Oxford University Press, New York, 1 edition, 1997.
13. R. Alejo, V. García, J.M. Sotoca, R.A. Mollineda, and J.S. Sánchez. Improving the performance of the rbf neural networks with imbalanced samples. In 9th International Work-Conference on Artificial Neural Networks (IWANN'2007), pages 162–169, San Sebastián, Spain, 2007. Springer Berlin / Heidelberg.
14. S. Haykin. Neural Networks. A Comprehensive Foundation. Pretince Hall, New Jersey, 2 edition, 1999.
15. Y-H. Pao, G.H. Park, and D.J. Sobajic. Learning and generalization characteristics of the random vector functional-link net. Neurocomputing, 6(2):163–180, 1994.