

Capítulo 9

CLASIFICACIÓN USANDO REDES NEURONALES ARTIFICIALES

Luis René Marcial Castillo, Marcela Rivera Martínez, Lourdes Sandoval Solís
Benemérita Universidad Autónoma de Puebla
Puebla, Puebla, México

lmc@cs.buap.mx, mrivmar@hotmail.com, sandoval@cs.buap.mx

RESUMEN

La clasificación es una tarea muy importante en diversas aplicaciones de la salud y del medio ambiente. En este trabajo, se realiza la clasificación mediante el uso de tres modelos de redes neuronales artificiales: red neuronal multicapa, red neuronal en base radial y red neuronal probabilística. Se desarrollan las pruebas usando las bases de datos de uso libre: fertilidad, parkinson, cambio climático y plantas.

Palabras Clave: Clasificación, red neuronal multicapa, red neuronal en base radial, red neuronal probabilística, aprendizaje supervisado.

ABSTRACT:

The classification is an important task in many applications of health and the environment. In this paper, the classification is performed using three models of artificial neural networks: multilayer neural network, radial basis function neural network, and probabilistic neural network. The tests were developed using databases for free use: fertility, parkinson, climate change and plants.

Keywords: Classification, multilayer neural network, radial basis function neural network, probabilistic neural network, supervised learning

1. INTRODUCCIÓN

El problema de clasificación continúa siendo un problema muy importante como lo muestran varios trabajos recientes. En el área de la salud, para clasificar Arritmias cardiacas, que son alteraciones del ritmo del corazón, caracterizadas por una irregularidad de frecuencia y una desigualdad de sus contracciones (Khoshnoud y Ebrahimnezhad2013), para clasificar señales epilépticas que surgen a partir de un electroencefalograma conocidas también como señales de tipo EEG (Goshvarpour1, Ebrahimnezhad, y Goshvarpour,2013), para clasificar la calidad del líquido seminal (Gila, Girelab, De Juan, Gomez y Johnssonc, 2012) o para clasificar pacientes con la enfermedad de Parkinson tomando en cuenta parámetros sobre la calidad del habla (Erdogdu, 2013). En el área del medio ambiente, para clasificar fallos en los modelos climáticos que se han usado en la predicción (Lucas, Klein, Tannahill, Ivanova, Brandon, Domyancic y Zhang, 2013) o para clasificar diferentes tipos de

plantas (Mallah, Cope y Orwell, 2013). El presente trabajo, explica los modelos de redes neuronales y aplica las librerías de OCTAVE para resolver el problema de clasificación mediante redes neuronales con aprendizaje supervisado, las pruebas se realizan con problemas prueba(benchmark) de bases de datos de uso libre.

El trabajo está organizado de la manera siguiente. En la sección 2, se presentan y explican los modelos de redes neuronales artificiales: red neuronal multicapa, red neuronal en base radial y red neuronal probabilística, en la sección 3, se presentan los experimentos y resultados usando para ello cuatro problemas prueba, la sección 4 presenta las conclusiones y finalmente se listan las referencias.

2. REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales (RNA) intentan modelar las capacidades de procesamiento del sistema nervioso, por lo que se deben tener en cuenta las propiedades esenciales de las redes neuronales biológicas desde la perspectiva de su procesamiento. Esta información permite diseñar modelos abstractos de redes neuronales artificiales que pueden ser simulados y analizados (Rojas, 2007). Las RNA presentan una serie de características propias del cerebro: aprenden mediante el entrenamiento y resuelven problemas similares con los que nunca se han entrenado. Se puede definir formalmente a una red neuronal artificial como:

Definición 1. Una red neuronal es una tripleta (N, V, w) con dos conjuntos N, V y una función w , donde N es el conjunto de neuronas y V es un conjunto $\{i, j\}, i, j \in Z^+$ cuyos elementos son denominadas conexiones entre la neurona i y la neurona j . La función $w: V \rightarrow \mathcal{R}$ define los pesos, donde $w(i, j)$ es el peso entre la neurona i y la neurona j .

Cualquier modelo de red neuronal artificial contiene tres elementos esenciales (Rojas, 1996):

1. La **estructura de sus neuronas**, es decir, el contenido de la neurona.
2. La **topología de la red neuronal**, es decir, la forma en la que se realizan las conexiones de las neuronas.
3. El **algoritmo de aprendizaje**, es decir, el método utilizado para encontrar los valores óptimos de los pesos que se involucran en la red neuronal artificial.

A continuación se explican los modelos de redes neuronales artificiales usados en la clasificación de las bases de datos.

2.1 Red neuronal multicapa

Las redes multicapa son aquellas en las cuales el conjunto de neuronas N es subdividido en m subconjuntos N_1, N_2, \dots, N_m de tal manera que forman un grupo de capas simples en cascada, es decir, las salidas de la capa N_1 son conectadas como entrada a la capa N_2 , las salidas de la capa N_2 son las entradas de la capa N_3 , y así sucesivamente. Las entradas x_1, \dots, x_n del problema son conectadas a la capa N_1 y las salidas del problema t_1, \dots, t_s son las salidas de la capa N_m . El esquema de la red neuronal multicapa se muestra en la figura 1. No existe un resultado que nos diga para un problema específico cuántas capas ocultas colocar ni cuántas neuronas agregar por capa oculta, la cantidad se especifica a prueba y error. Los pesos o parámetros que deberán ajustarse para obtener la

salida deseada en el aprendizaje supervisado se representan por las matrices W_1, W_2, \dots, W_m . Por ejemplo, supóngase que se desea clasificar rostros humanos en base a su sexo (masculino o femenino), si la imagen digital que representa el rostro humano es de $64 \times 64 = 4096$ píxeles en tonos de gris, entonces la capa de entrada contendrá 4096 sitios de entrada y la capa de salida contendrá 2 neuronas si la salida para femenino se representa con [1 0] y para masculino con [0 1], si además se supone que existen 3 capas ocultas una con 20 neuronas, otra con 10 y la última con 5 neuronas, entonces W_1 contendrá $4096 \times 20 = 81920$ pesos, W_2 contendrá $20 \times 10 = 200$ pesos, W_3 contendrá $10 \times 5 = 50$ pesos y W_4 contendrá $5 \times 2 = 10$ pesos. Por lo tanto, la cantidad de pesos o parámetros que deberán ajustarse es $81920 + 200 + 50 + 10 = 82180$, es decir, se tendría que resolver un problema de optimización de 82180 variables.

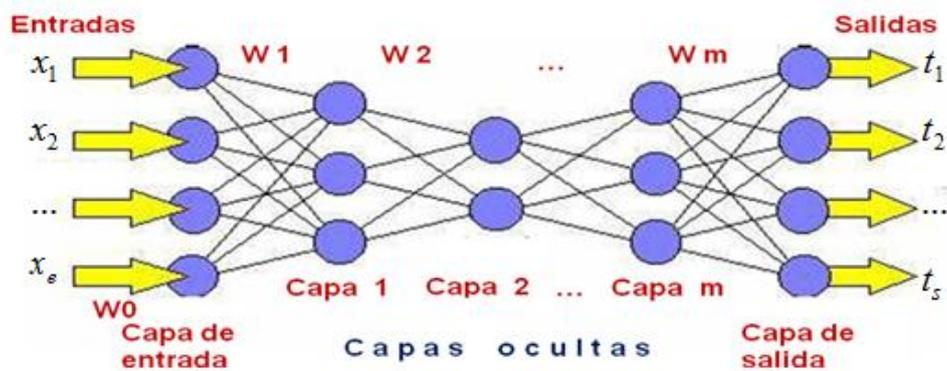


Figura 1. Red neuronal multicapa.

Si al construir la función que mide el error se tiene que ésta es continuamente diferenciable, entonces, se puede usar algún método de búsqueda lineal para encontrar los pesos óptimos. En caso de que la función de error no sea diferenciable se puede utilizar alguna meta heurística para ajustar los pesos.

Para encontrar los pesos óptimos de una red neuronal multicapa Rumelhart, Hilton y Williams propusieron en 1986 el algoritmo conocido como retro propagación hacia atrás (backpropagation). El algoritmo de aprendizaje backpropagation conlleva una fase de propagación hacia adelante y otra fase de propagación hacia atrás. El algoritmo es el siguiente (Rojas, 1996):

Considérese una red neuronal artificial con una sola entrada x y una función de red $f(x)$, la derivada $f'(x)$ se calcula en dos fases:

Propagación hacia adelante. En esta fase, la entrada x es alimentada a la red, se evalúan y almacenan los valores obtenidos de las funciones primitivas y sus derivadas en cada uno de los nodos, el valor de la función se almacena en la parte derecha del nodo y el valor de la derivada se almacena en la parte izquierda.

Propagación hacia atrás. La constante unose alimenta en la unidad de salida y la red es alimentada en orden inverso. La información en un nodo se suma y el resultado se multiplica por el valor almacenado en la parte izquierda de cada unidad. El resultado se transmite a la izquierda de la unidad. Finalmente, el resultado coleccionado en la unidad de entrada, es la derivada de la función de la red neuronal con respecto a la variable x .

Ejemplo 1. Supóngase que se desea clasificar la función booleana de 2 variables $f(x,y)$ conocida como la función *xor*, ilustrada en el cuadro 1.

Cuadro 1. Función booleana *xor* de dos variables.

x	y	$f(x,y)=xor(x,y)$
0	0	0
0	1	1
1	0	1
1	1	0

La arquitectura usada es la que se muestra en la figura 2.

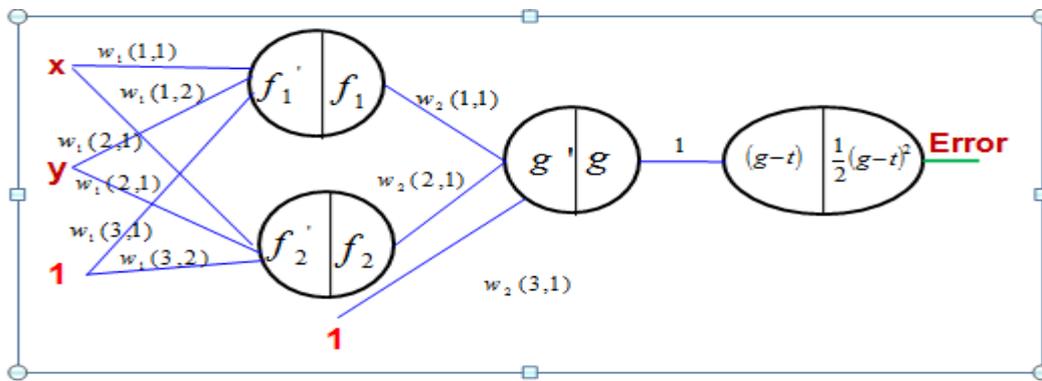


Figura 2. Red neuronal multicapa para el problema *xor*.

Para este problema deberán ajustarse 9 parámetros. Si se usa como función de activación la función sigmoide dada por:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

Cuya derivada se expresa en términos de sí misma, es decir,

$$f'(x) = \frac{e^{-x}}{1 + e^{-x}} = \frac{1}{1 + e^{-x}} \left(1 - \frac{1}{1 + e^{-x}} \right) = f(x) \left(1 - f(x) \right) \quad (2)$$

Las derivadas parciales con respecto a la función de error se evalúan como:

$$\frac{\partial Error(W1,W2)}{\partial w_2(1,1)} = (g-t)g'f_1 \quad (3)$$

$$\frac{\partial Error(W1,W2)}{\partial w_2(2,1)} = (g-t)g'f_2 \quad (4)$$

$$\frac{\partial Error(W1,W2)}{\partial w_2(3,1)} = (g-t)g'f_3 \quad (5)$$

$$\frac{\partial Error(W1,W2)}{\partial w_1(1,1)} = (g-t)g'w_2(1,1)f_1'x \quad (6)$$

$$\frac{\partial Error(W1,W2)}{\partial w_1(1,2)} = (g-t)g'w_2(2,1)f_2'x. \quad (7)$$

$$\frac{\partial Error(W1,W2)}{\partial w_1(2,1)} = (g-t)g'w_2(1,1)f_1'x. \quad (8)$$

$$\frac{\partial Error(W1,W2)}{\partial w_1(2,2)} = (g-t)g'w_2(2,1)f_2'y. \quad (9)$$

$$\frac{\partial Error(W1,W2)}{\partial w_1(3,1)} = (g-t)g'w_2(1,1)f_1'*1. \quad (10)$$

$$\frac{\partial Error(W1,W2)}{\partial w_1(3,2)} = (g-t)g'w_2(2,1)f_2'*1. \quad (11)$$

Para encontrar los pesos óptimos de los parámetros almacenados en forma

matricial por $W_1 = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \\ w_{3,1} & w_{3,2} \end{bmatrix}$ y $W_2 = \begin{bmatrix} w_{1,1} \\ w_{2,1} \\ w_{3,1} \end{bmatrix}$, mediante el método de máximo descenso

se siguen los pasos siguientes:

1. Generar una aproximación inicial de forma aleatoria para las marices W_1^0 y W_2^0 .
2. Calcular el gradiente $grad_0$ de W_1^0 y W_2^0 usando las ecuaciones (3)-(11) e inicializar k con el valor de 1.
3. Poner la dirección de descenso $d_k = -grad_k$.
4. Mientras $grad_k \neq 0$
 - a. Calcular el tamaño de paso α_k mediante una búsqueda lineal inexacta o bien asignarle un valor suficientemente pequeño.
 - b. Calcular $W_1^{k+1} = W_1^k + \alpha_k * d_k$ y $W_2^{k+1} = W_2^k + \alpha_k * d_k$.
 - c. Calcular el gradiente $grad_{k+1}$ usando las ecuaciones (3)- (11).
 - d. Poner $d_{k+1} = -grad_{k+1}$ y $k = k+1$.

La ejecución de la implementación computacional del algoritmo que no usa librería alguna para redes neuronales y realizada con la computadora especificada en la sección 3, entregó los resultados siguientes para los pesos óptimos:

$$W_1 = \begin{bmatrix} -5.7338 & -3.6772 \\ -5.6286 & -3.6610 \\ 2.0758 & 5.3744 \end{bmatrix}, \quad W_2 = \begin{bmatrix} -7.5248 \\ 7.1598 \\ -3.2273 \end{bmatrix}.$$

Tales valores de los pesos corresponden al esquema que se muestra en la figura 3, y que al evaluar la red neuronal se obtiene la función booleana $x \text{ or } y$ en dos variables.



Figura 3. Solución de la función booleana *xor* de dos variables.

La dirección de descenso también puede ser la dirección de gradientes conjugados para problemas de alta dimensión y la dirección de newton o cuasi newton si el problema es de baja dimensión.

2.2 Red neuronal en base radial

Las redes neuronales en funciones de base radial (FBR) surgen de la teoría de aproximación de funciones y fueron usadas para el reconocimiento de patrones (Tou y Gonzalez, 1974). Estas redes neuronales constan de dos capas, donde los nodos ocultos implementan un conjunto de funciones en base radial y los nodos de salida implementan funciones de sumatoria lineales, es decir, la entrada en estas redes es no lineal y la salida es lineal. Debido a la característica no lineal, las redes neuronales en base radial son hábiles para modelar mapeos complejos, cosa que el perceptrón multicapa puede solo realizar mediante múltiples capas intermedias. El entrenamiento para estas redes es dividido en dos estados: primero se determinan los pesos que conectan las entradas con la capa oculta y después se determinan los pesos de la capa oculta a la capa de salida, el entrenamiento en estas redes es muy rápido.

Ejemplo 2. Supóngase que se desea clasificar mediante una red neuronal en base radial a la función booleana de 2 variables $f(x,y)$ conocida como la función *xor*, ilustrada en el cuadro 1. La estructura de la red neuronal en base radial para este problema se ilustra en la figura 4. La función de activación en la capa oculta es la función Gaussiana dada por la expresión

$$f(x) = e^{-\frac{x^2}{2\sigma^2}}. \quad (12)$$

En la capa de salida se usa la función lineal dada por la expresión

$$g(x) = x. \quad (13)$$

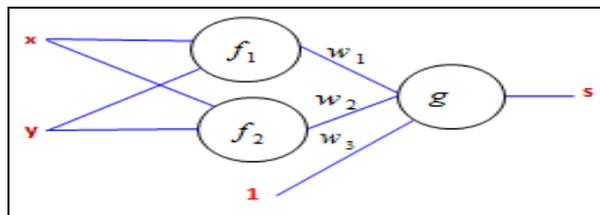


Figura 4. Topología usada de la red neuronal en base radial para el *xor* de dos variables.

Para encontrar los pesos óptimos en la red neuronal en base radial para la arquitectura de la figura 4, se deben seguir los pasos siguientes:

Paso 1. Encontrar los centros de los datos. Dos centros para el caso de 2 neuronas en la capa oculta, la forma más sencilla es utilizar el algoritmo k medias.

Paso 2. Para cada nodo de la capa oculta encontrar la distancia euclidiana entre los datos y el centro asignado al nodo.

Paso 3. Para cada nodo de la capa oculta aplicar la función Gaussiana.

Paso 4. Encontrar w resolviendo el sistema lineal $Gw=s$, donde s es el vector de salidas que dicta el entrenamiento y G es la matriz que se debe obtener en el paso 3.

En caso de no encontrar la solución del problema se deberá incrementar la cantidad de nodos en la capa oculta.

2.3 Red neuronal probabilística

Una red neuronal probabilista es la implementación de un algoritmo estadístico denominado análisis de discriminante en el cual las operaciones son organizadas en una red con cuatro capas (Chen y Tsai, 2014): capa de entrada, capa de patrones, capa de sumatoria y capa de salida. La arquitectura básica de una red neuronal probabilística se ilustra en la figura 5. La capa de entrada contendrá los valores de los atributos que se presentan a la red en cada entrada, la capa de patrones calculará la distancia entre el vector de entrada y la salida esperada produciendo un vector cuyos elementos mostrarán la cercanía entre la entrada y la salida, la capa de sumatoria sumará las contribuciones para cada clase de entradas con el fin de producir un vector de salida que contendrá las probabilidades y finalmente la capa de salida contendrá una cantidad de neuronas igual a la cantidad de clases que se desea clasificar.

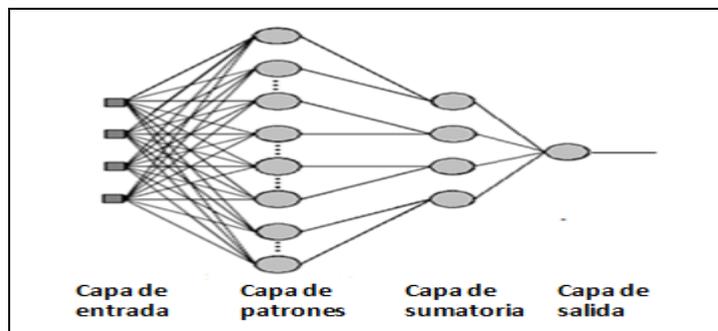


Figura 5. Topología de una red neuronal probabilística.

3. EXPERIMENTOS Y RESULTADOS

Se realizó la clasificación de cuatro problemas prueba dos del área de la salud y dos del medio ambiente. Para la clasificación se utilizaron las librerías de OCTAVE probando con redes multicapa, redes en base radial y redes probabilísticas. Todas las corridas se realizaron en una computadora Intel Core i7 con 2.20 GHz y 8.0GB de memoria RAM.

El nombre de los problemas prueba, así como la cantidad de instancias, atributos y clases de cada uno de ellos se muestran en el cuadro 2.

Cuadro 2. Problemas benchmark.

Nombre	Instancias	Atributos	Clases
Fertilidad	100	10	2
Parkinson	1040	26	2
Modelo climático	540	18	2
Plantas	1600	64	100

El problema prueba llamado *Fertilidad* (Gila, Girelab, De Juan, Gomez y Johnssonc, 2012), se diseñó en el año 2010 (Gil y Girela, 2013). Esta base de datos ofrece los resultados de muestras de líquido seminal obtenidos de 100 voluntarios que fueron analizadas bajo la norma del año 2010 de la Organización Mundial de Salud, los atributos son asociados a datos socio-demográficos, factores ambientales, estado de salud y hábitos de vida. El problema prueba *Parkinson* (Erdogdu, Isenkul, Sakar, Sertbas, Gurgen, Delil, Apaydin y Kursun, 2013), se diseñó en el año 2013. Esta base de datos consta de datos de entrenamiento de individuos que tienen o no el mal de Parkinson, en donde se toman múltiples tipos de grabaciones de sonido incluyendo las vocales, números, palabras y oraciones cortas. El problema prueba *Modelo climático* (Lucas, Klein, Tannahill, Ivanova, Brandon, Domyancic y Zhang 2013) apareció en el año 2013 (Lucas, 2013); los atributos son asociados a accidentes de simulación de los modelos climáticos cuando se usa el modelo estadístico hiper cubo latino. Finalmente, el problema prueba *Plantas* (Mallah, Cope y Orwell, 2013; Mallah y Orwell 2013) apareció en el año 2012 (Mallah, C. 2012), en esta base de datos los atributos están asociados a la forma, textura o margen de una planta.

Para el caso de la red neuronal multicapa se decidió a prueba y error, la cantidad de capas ocultas a utilizar y cuántas neuronas ubicar en cada capa oculta. Se observó que la arquitectura que daba un menor error cuadrático medio (MSE) fue cuando se tenían tres capas ocultas en todos los casos. Para las bases de datos *Fertilidad* y *Modelo climático*, la primera capa oculta contiene 10 neuronas, la segunda y tercera contienen 5 neuronas. Para las base de datos *Plantas* y *Parkinson*, las tres capas ocultas contienen 64 neuronas.

Todos los modelos de redes neuronales se entrenaron con el 50% de los datos y se probó su predicción en el 50% restante.

Para la base de datos *Fertilidad*, el cuadro 3 muestra los resultados del entrenamiento para cada tipo de red neuronal: cantidad de neuronas utilizadas, tiempo de entrenamiento, porcentaje de aciertos y error cuadrático medio (MSE). En el cuadro 4, se muestran los resultados de la predicción: porcentaje de aciertos, tiempo de ejecución, así como el MSE.

Cuadro 3. Resultados del entrenamiento para el problema prueba *Fertilidad*.

Tipo de red neuronal	Neuronas	Tiempo	Porcentaje de aciertos	MSE
Multicapa	21	22.690seg.	94	0.0507
Radial	80	0.236seg.	100	0.0
Probabilística	11	0.040seg.	100	0.0

Cuadro 4. Resultados de la predicción para el problema prueba *Fertilidad*.

Tipo de red neuronal	Tiempo	Porcentaje de aciertos	MSE
Multicapa	0.011seg.	84	0.16
Radial	0.103seg.	80	0.2
Probabilística	0.008seg.	78	0.22

Para la base de datos *Parkinson*, el cuadro 5 muestra los resultados del entrenamiento para cada tipo de red neuronal: cantidad de neuronas utilizadas, tiempo de entrenamiento, porcentaje de aciertos y error cuadrático medio (MSE) y el cuadro 6, muestra los resultados de la predicción.

Cuadro 5. Resultados del entrenamiento para el problema prueba *Parkinson*.

Tipo de red neuronal	Neuronas	Tiempo	Porcentaje de aciertos	MSE
Multicapa	21	6.30seg.	100	0.00000
Radial	500	36.34seg.	98.0769	0.00005
Probabilística	27	0.082seg.	100	0.00000

Cuadro 6. Resultados de la predicción para el problema prueba *Parkinson*.

Tipo de red neuronal	Tiempo	Porcentaje de aciertos	MSE
Multicapa	0.009seg.	100	0.0000
Radial	0.04seg.	68.07	0.28
Probabilística	0.07seg.	70.1923	0.2990

Para la base de datos *Modelo climático*, el cuadro 7 muestra los resultados del entrenamiento para cada tipo de red neuronal: cantidad de neuronas utilizadas, tiempo de entrenamiento, porcentaje de aciertos y error cuadrático medio (MSE). En el cuadro 8, se muestran los resultados de la predicción: porcentaje de aciertos, tiempo de ejecución, así como el MSE.

Cuadro 7. Resultados del entrenamiento para el problema prueba *Modelo climático*.

Tipo de red neuronal	Neuronas	Tiempo	Porcentaje de aciertos	MSE
Multicapa	21	4.7222seg.	96.66	0.0261
Radial	80	0.0753seg.	100	0.0
Probabilística	19	0.0479seg.	100	0.0

Cuadro 8. Resultados de la predicción para el problema prueba *Modelo climático*.

Tipo de red neuronal	Tiempo	Porcentaje de aciertos	MSE
Multicapa	0.0132seg.	92.59	0.0741
Radial	0.0174seg.	91.48	0.0852
Probabilística	0.0192seg.	85.18	0.1418

Para la base de datos de *Plantas*, el cuadro 9 muestra los resultados del entrenamiento para cada tipo de red neuronal: cantidad de neuronas utilizadas, tiempo de entrenamiento, porcentaje de aciertos y error cuadrático medio (MSE). En el cuadro 10, se muestran los resultados de la predicción: porcentaje de aciertos, tiempo de ejecución, así como el MSE.

Cuadro 9. Resultados del entrenamiento para el problema prueba *Plantas*.

Tipo de red neuronal	Neuronas	Tiempo	Porcentaje de aciertos	MSE
Multicapa	292	45 minutos	60.375	0.0066
Radial	400	28.79seg.	99.87	0.000025
Probabilística	65	0.2058seg.	84.1250	0.0032

Cuadro 10. Resultados de la predicción para el problema prueba *Plantas*.

Tipo de red neuronal	Tiempo	Porcentaje de aciertos	MSE
Multicapa	0.0192seg.	46.875	0.0086
Radial	0.094seg.	73.0	0.0040
Probabilística	0.1840seg.	67.53	0.00658

4. CONCLUSIONES

Las redes neuronales son técnicas excelentes que deben usarse en los problemas de clasificación debido a que sus porcentaje de clasificación son altos en la fase de entrenamiento para las bases de datos *Fertilidad*, *Parkinson* y *Modelo climático*.

La red neuronal multicapa es un modelo que tiene muchos parámetros que deben ajustarse a prueba y error, como por ejemplo, la cantidad de capas ocultas, el número de neuronas de cada capa oculta. Es difícil lograr que disminuya el error cuadrático medio, siendo por consiguiente muy tardado su entrenamiento.

La red neuronal en base radial es rápida en su entrenamiento. En su fase de entrenamiento clasifica adecuadamente para las cuatro bases de datos con al menos el 99.87% y predice con al menos el 73% de aciertos en datos que no se usan en su entrenamiento. Esto se debe a la naturaleza no lineal de su capa oculta.

La red neuronal probabilística es la más rápida, aunque resulta inferior en sus resultados cuando se compara a la red neuronal en base radial.

Se recomienda el uso de las redes neuronales en base radial, pues su funcionamiento es bastante bueno sin importar la cantidad de instancia, atributos o clases.

Para la base de datos *Plantas* la fase de prueba no proporciona porcentajes altos de clasificación correcta, la razón es que existen demasiadas clases (100) y la cantidad de instancias es bastante limitada.

REFERENCIAS

- [1] Chen, S.L., Tsai M.T. (2014), "The Operational Monitor of Air-conditioners by using Probabilistic Neural Network", *Applied Mechanics and Materials*, 543(547), p. 1149-1152.
- [2] Erdogdu S., Isenkul, M., Sakar, C., Sertbas, A., Gurgun, F., Delil, S., Apaydin, H., Kursun, O. (2013) "Collection and Analysis of a Parkinson Speech Dataset with Multiple Types of Sound Recordings", *IEEE Journal of Biomedical and Health Informatics*, vol. 17(4), pp. 828-834.
- [3] Erdogdu, S. (2013), Parkinson Speech Dataset
- [4] <http://archive.ics.uci.edu/ml/datasets/Parkinson>
- [5] Fecha consulta: 15 de Octubre de 2014, fecha actualización sitio web: 27Junio de 2014.
- [6] Gila, D., Girelab J., De Juan, J., Gomez M., Johnssonc M.(2012), "Predicting seminal quality with artificial intelligence methods", *Expert systems with applications*, 39 (16).
- [7] Gil, D y Girela L. (2013), Fertility Data Set,
- [8] <http://archive.ics.uci.edu/ml/datasets/Fertility>
- [9] Fecha consulta: 2 de Julio de 2014, fecha actualización sitio web: 27Junio de 2014.
- [10] Goshvarpour¹, A., Ebrahimnezhad H., Goshvarpour A. (2013), "Classification of Epileptic EEG Signals using Time-Delay Neural Networks and Probabilistic Neural Networks", *I.J. Information Engineering and Electronic Business*, 1(1), p. 59-67.
- [11] Khoshnoud, S., Ebrahimnezhad, H. (2013), "Classification of Arrhythmias Using Linear Predictive Coefficients and Probabilistic Neural Network", *Applied Medical Informatics* 33(3), p: 55 – 62.
- [12] Lucas, D., Klein, D., Tannahill, J., Ivanova, D., Brandon, S., Domyancic, D., and Zhang, Y. (2013), "Failure analysis of parameter-induced simulation crashes in climate models", *Geosci. Model Dev. Discuss*, 6 (6), p. 585-623.
- [13] Lucas, D. (2013), Climate Model Data Set
- [14] <http://archive.ics.uci.edu/ml/datasets/Climate+Model+Simulation+Crashes>
- [15] Fecha consulta: 2 de Julio de 2014, fecha actualización sitio web: 27Junio de 2014
- [16] Mallah, C., Orwel, J. (2013), "Probabilistic Classification from a K-Nearest-Neighbour Classifier", *Computational Research*, 1(1), p. 12564-12573.
- [17] Mallah, C., Cope, J., Orwell, J. (2013), "Plant leaf classification using probabilistic integration of shape, texture and margin features", *Signal Processing, Pattern Recognition and Applications*, 1(1): 1-9
- [18] Mallah, C. (2012), 100 Leaves Plant Species Data Set
- [19] <http://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set> último acceso 2 de Julio de 2014, actualización 27 de Junio de 2014.
- [20] Rojas, R. (1996), *Neural Networks a systemic introduction*, Springer-Verlag.
- [21] Tou, J.T., Gonzalez, R.C. (1974), *Pattern Recognition*, Addison-Wesley.