

Comparación del Desempeño de Funciones de Activación en Redes Feedforward para aproximar Funciones de Datos con y sin Ruido

Comparison of the Activation Functions Performance in Neural Networks Feedforward to approximate Data Functions with and without Noise

Luis Llano, MSc.(c)¹, Andrés Hoyos, Est.², Francisco Arias, Est.², Juan Velásquez, PhD.(c)³

¹Interconexión Eléctrica S.A. E.S.P. (ISA), Colombia

²GIDIA: Grupo de Investigación y Desarrollo en Inteligencia Artificial

³Grupo de Finanzas Computacionales

Escuela de Ingeniería de Sistemas, Facultad de Minas

Universidad Nacional de Colombia Sede Medellín

lellano@isa.com.co, afhoyos@unalmed.edu.co, fjarias@unalmed.edu.co, jdvelasq@unalmed.edu.co

Recibido para revisión 26 de Marzo de 2007, aceptado 15 de Junio de 2007, versión final 31 de julio de 2007

Resumen—Las redes neuronales en muchos casos han dado buenos resultados en la aproximación de funciones en muchas aplicaciones, aun así existen muchos problemas que no se han podido resolver. La selección de funciones de activación se realiza de acuerdo con el problema y a criterio del investigador, en ocasiones por ensayo y error. Comúnmente, la función de activación logística ha sido la más frecuentemente usada trayendo buenos resultados. En la literatura no existe un criterio estándar para la selección de estas funciones de activación en las redes neuronales, ni tampoco existe una exhaustiva investigación en este tema. Es por esta razón que el objetivo principal de este artículo es obtener un criterio de selección para tres funciones de activación en una red neuronal feedforward con una capa oculta, comparando su desempeño con múltiples neuronas, para aproximar las funciones objetivo propuestas en [4] las cuales fueron diseñadas para evaluar la capacidad de regresión de modelos de redes neuronales.

Palabras Clave—Aproximación de Funciones, Función de Activación, Red Neuronal Feedforward.

Abstract—The neural networks in many cases have given good results in the approach of functions in many applications, even so exist many problems that have not been able to solve. The selection of activation functions is made in agreement with the problem and to criterion of the investigator, sometimes by test and error. Commonly, the function of logistic activation has been more frequently used bringing good results. In Literature a

standard criterion for the selection of these functions of activation in the neuronal networks does not exist, nor exists either an exhaustive investigation in this subject. It is therefore

that the primary target of this I articulate is to obtain a criterion of selection for three functions of activation in a neural network feedforward with an hidden layer, comparing his performance with multiple neurons, to approximate the propose functions objective in [4] which were designed to evaluate the capacity of regression of models of neuronal networks.

Keywords—Approach Functions, Activation Function, Feedforward Neural Network.

I. INTRODUCCIÓN

Las redes neuronales perceptrón multicapa han sido ampliamente utilizadas debido a su simplicidad y buenos resultados, aunque en ocasiones fallan en proveer una adecuada solución, debido a una mala arquitectura, insuficiente número de neuronas o simplemente un número insuficientes de ciclos de entrenamiento [7].

El desempeño del entrenamiento de una red neuronal depende del algoritmo de aprendizaje utilizado, del número de capas ocultas, del número de neuronas en cada capa oculta, de la conectividad o arquitectura de la red y también del tipo de

función de activación que se elija para cada neurona.

Las actividades de entrenamiento y generalización, también conocidas como de aprendizaje y de prueba (o validación), respectivamente, son fundamentales en la definición de una red neuronal. En la actividad de entrenamiento, el algoritmo de aprendizaje aplica un procedimiento definido que modifica los pesos o ponderaciones de interconexión de la red, ajustándolos de tal forma que cada vez se obtenga un mejor resultado de la red con respecto a los datos de prueba. De acuerdo con [2] se pueden ajustar los pesos o sesgos, o quizás la misma red neuronal los ajuste para alcanzar el fin deseado.

Estos algoritmos pueden ser supervisados o no supervisados, dentro de los supervisados se encuentra el algoritmo de retropropagación (backpropagation) o del gradiente descendente, creado como generalización del algoritmo para perceptrón continuo cuando se tienen múltiples capas (redes neuronales perceptrón multicapa) y popularmente utilizado para la solución de problemas de clasificación y pronóstico [1], así como para aplicaciones de reconocimiento de patrones, procesamiento de señales, comprensión de datos y control automático, con ejemplos bien conocidos como el NETtalk, para pronunciar textos en inglés y ALVINN, para dirigir vehículos autónomos [5]. Este método es a menudo demasiado lento para problemas prácticos [2] por lo cual se han propuesto variantes de éste de mayor desempeño que pueden converger desde diez a cientos de veces más rápido utilizando técnicas heurísticas, tal es el caso del método elástico de retropropagación (resilient backpropagation o Rprop), de muy rápida convergencia para problemas de reconocimiento de patrones [2]. En esta etapa, la selección de los pesos iniciales en la solución de problemas prácticos puede ser crucial [1], característica de la que dispone el algoritmo Rprop.

La arquitectura más comúnmente utilizada con el algoritmo de aprendizaje de retropropagación es la red neuronal con alimentación adelantada (feedforward) [2], la cual tiene en contrapartida la red neuronal de función de base radial (radial basis networks) que quizás requiere más neuronas que la de alimentación adelantada pero que a menudo puede ser diseñada en una fracción del tiempo que toma entrenar la otra [2].

Respecto al número de capas, éstas deben ser mayores si el problema es no lineal y complejo, pero en general un problema podrá representarse bastante bien con una o dos capas ocultas [1] y [3]. El número de neuronas por capa normalmente se determina por ensayo y error, aunque existe el criterio de considerar al promedio entre el número de entradas y salidas como un valor referencial del número de neuronas en las capas ocultas. [1]

La función de activación cumple con el objetivo de limitar el rango de salida de la neurona y puede ser lineal o no lineal. Se selecciona de acuerdo con el problema y a criterio del investigador, en ocasiones por ensayo y error [1], también depende de la precisión y velocidad requerida y del algoritmo de aprendizaje escogido [7]. De acuerdo con el rango de las funciones de activación utilizadas es conveniente escalar o

transformar los datos de entrada para ajustarlos a dichos rangos. Comúnmente se utilizan: la función paso, para problemas de clasificación; la función lineal, en distintos tipos de redes frecuentemente en la capa de salida (funciones no lineales en la neurona de salida son comúnmente utilizadas en tareas de clasificación de patrones para restringir los valores de salida a rangos tales como $[0,1]$ o $[-1,1]$) [3]; la función logística, es la más comúnmente usada por sus características de derivación y se recomienda para problemas de predicción (aunque en el algoritmo de retropropagación la función más comúnmente utilizada es la logística, éste puede trabajar con cualquier otra función de activación que sea diferenciable [7]); la función tangente hiperbólica, es similar a la logística y se utiliza con frecuencia en redes multicapa, la función gaussiana utilizada en redes neuronales de función de base radial las cuales pueden aplicarse a problemas de predicción. [1] y [2].

Una buena función de activación debería cumplir: primero, que ella misma y su derivada sean fáciles de computar y segundo, que la función debe tener una amplia parte lineal para lograr velocidad de entrenamiento y convergencia en pocos ciclos. Precisamente la función logística (logistic sigmoidal) es usada comúnmente porque su derivada es una de las más fáciles de computar. [7] plantea que hay una completa clase de otras funciones sigmoidal que pueden ser implementadas para ampliar las posibilidades en el uso de las funciones de activación e indican que la función log-exp puede ser mejor que la logística, como resultado del trabajo de comparación visual de algunas posibles funciones de activación.

Este documento presenta los resultados de un experimento de simulación de comparación de varias funciones de activación para determinar si alguna es superior a las demás, buscando recoger información sobre el desempeño de las distintas funciones de activación para dar indicios en la selección de éstas. Se utilizaron cinco funciones objetivo (con ruido y sin ruido), tres funciones de activación, una red neuronal de alimentación adelantada variando el número de neuronas y con retención de los pesos y el algoritmo de aprendizaje elástico de retropropagación.

II. PROBLEMA A RESOLVER

La selección de funciones de activación se realiza de acuerdo con el problema a resolver y a criterio del investigador, en ocasiones por ensayo y error. Comúnmente, la función de activación logística ha sido la más frecuentemente usada trayendo buenos resultados. En la literatura no existe un criterio estándar para la selección de estas funciones de activación en las redes neuronales, ni tampoco existe una exhaustiva investigación en este tema. En otras investigaciones se trabajó con las funciones de activación Sigmoidal, Gaussian y Sinusoidal, y se concluyó que la función Sigmoidal no es efectiva cuando se tiene una sola capa oculta a diferencia de las demás y para mayor cantidad de capas su convergencia es lenta, por otra parte la

función Sinusoidal es mas útil para clasificar patrones con y sin ruido basado en la tasa de convergencia, error mínimo y sensibilidad del ruido [8].

Debido a esto, surgió la idea de hacer una comparación aproximando las funciones objetivo propuestas en [4], las cuales fueron diseñadas para evaluar la capacidad de regresión de modelos de redes neuronales, variando la función de activación en las neuronas de la capa oculta de la red neuronal, haciendo pruebas con diferente numero de neuronas de tal manera que sea posible obtener un criterio de selección entre tres funciones de activación para redes de tipo feedforward con diferentes funciones objetivo de diferente complejidad. pixels.

III. METODOLOGÍA Y MODELO DE LA RED NEURONAL UTILIZADO

La metodología utilizada partió de la selección de las funciones objetivo de prueba, lo cual se basó en las funciones propuestas en [4] para el análisis comparativo de dos tipos de métodos de aprendizaje conexionista para problemas de regresión de modelos libres, el “backpropagation” y el “projection pursuit” donde utilizaron en su protocolo de simulaciones cinco funciones no lineales, las cuales fueron escaladas tal que la desviación estándar es 1 para un gran número regular de puntos en [0,1]² y trasladadas para hacerlas en un rango no negativo.

Al adicionar neuronas en la capa oculta se conservaron los pesos obtenidos en el entrenamiento anterior para lograr mayor eficiencia. En todo caso se hizo el monitoreo para identificar casos de pérdida de generalización ante la posibilidad de tener problemas por memorización de la red neuronal (problema de “overfitting”).

Se utilizó la herramienta Neural Network Toolbox del programa MATLAB® para todas las simulaciones y la obtención de los resultados de comparación, que en todo caso fueron los errores de entrenamiento y generalización (o validación), obtenidos como la media cuadrática de la diferencia entre la salida objetivo y la salida de la red de los datos de entrenamiento (MSE) y su análogo para los datos de generalización.

En [6] se utilizaron estas mismas funciones en el estudio de la adición de nuevas neuronas ocultas en la construcción de algoritmos para redes multicapa “feedforward” y concluyeron en general, que el desempeño en la generalización de las redes usando diferentes funciones objetivos puede estar influenciado por tres factores: primero, no se puede garantizar siempre el mejor desempeño de la red por el análisis del mayor error residual posible cada vez que se adiciona una nueva neurona. Segundo, la precisión de la función objetivo está influenciada por el conjunto de datos de entrenamiento seleccionados que sea en realidad una muestra representativa del espacio de entrada y tercero, los problemas de optimización prácticos tienden al problema de óptimo local de la función, algunas funciones son más vulnerables a este problema seguramente por su forma compleja.

Los resultados del presente trabajo también pueden verse como un complemento al análisis mencionado, toda vez que también se evalúa el desempeño en la generalización de las redes con diferentes funciones objetivo (algunas iguales a las usadas en [6]) pero a partir de la variación de las funciones de activación.

De acuerdo con lo mencionado en la introducción, se encontró apropiado el siguiente modelo para la red neuronal: una red neuronal con dirección de alimentación adelantada (feedforward) con una sola capa oculta, variando el numero de neuronas en esta y la misma función de activación para cada neurona, con una neurona en la capa de salida para todos los casos con función de activación lineal. El algoritmo de aprendizaje utilizado para el análisis, según lo explicado en la introducción y debido a que se busca aproximar funciones objetivo, fue el algoritmo elástico de retropropagación (Rprop). No se realizó pruebas con otros algoritmos de aprendizaje, debido a que el objetivo del artículo es comprar el desempeño de la red neuronal según diferentes funciones de activaciones y bajo las mismas condiciones.

A. Funciones Objetivo

Las funciones objetivo pueden expresarse matemáticamente como sigue. En la figura 1 se muestra la representación 3-dimensional de estas funciones:

1) *F1: Función de Interacción Simple (figura 1a)*

$$g_{(1)}(x_1, x_2) = 10.391((x_1 - 0.4) * (x_2 - 0.6) + 0.36) \quad (1)$$

2) *F2: Función Radial (figura 1b)*

$$g_{(2)}(x_1, x_2) = 24.234 * r^2 * (r^2 (0.75 - r^2)) \quad (2)$$

donde $r^2 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$

3) *F3: Función Harmónica (figura 1c)*

$$g_{(3)}(x_1, x_2) = 42.659(0.1 + x_a(0.05 + x_a^4 - 10x_a^2x_b^2 + 5x_b^4)) \quad (3)$$

donde $x_a = x_1 - 0.5$ y $x_b = x_2 - 0.5$

4) *F4: Función Aditiva (figura 1d)*

$$g_{(4)}(x_1, x_2) = 1.3356(1.5(1-x_1) + e^{2x_1-1}\sin(3\pi(x_1-0.6)^2) + e^{3(x_2-0.5)}\sin(4\pi(x_2-0.9)^2)) \quad (4)$$

5) *F5: Función de Iteración Complicada (Figura 1e)*

$$g_{(5)}(x_1, x_2) = 1.9(1.35 + e^{x_1}\sin(13(x_1-0.6)^2)e^{x_2}\sin(7x_2)) \quad (5)$$

Al igual que en [4] para los datos de entrenamiento fueron generados aleatoriamente 255 pares de datos de entrada (x1, x2) en el rango [0,1] y con los que se obtuvieron los respectivos valores de las funciones objetivos g (i) sin ruido. Se generan 5 grupos de patrones adicionales para los datos con ruido sumando a cada uno de los grupos ya formados un número aleatorio obtenido de una distribución normal estándar y multiplicando por 0.25, esto es:

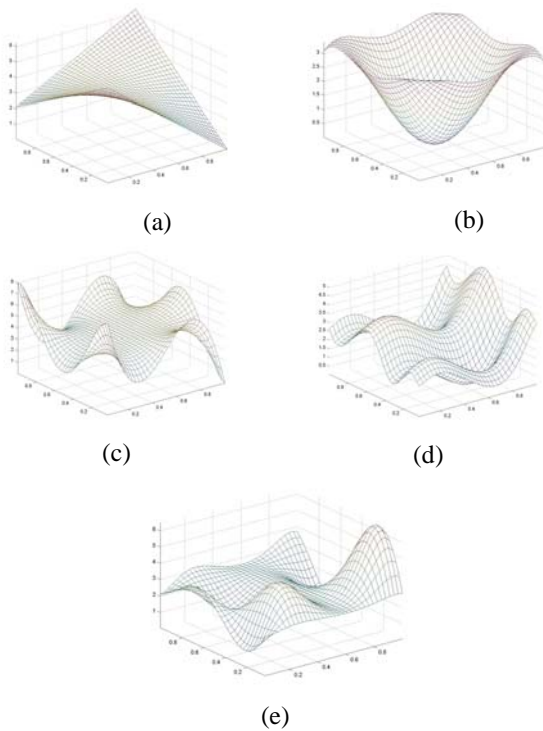


Figura 1. Gráficos 3D de las funciones objetivo (a) Función de Interacción Simple, (b) Función Radial, (c) Función Armónica, (e) Función Aditiva, (d) Función de Iteración Complicada.

$$f_{(i)} = g_{(i)} + 0.25 N[0,1] ; i = 1,2,3,4,5. \quad (6)$$

Los datos para la prueba de generalización de la red neuronal se realizaron con 10000 pares de datos de entrada (x1, x2) en el rango [0,1] igualmente espaciados.

Estos datos constituyeron los datos de entrada para el entrenamiento de la red neuronal y permanecieron invariables para todas las simulaciones.

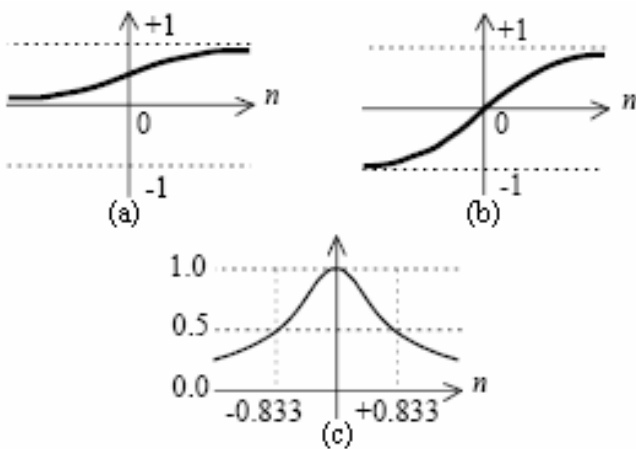


Figura 2. Funciones de activación (a) Función Logística, (b) Función tangente hiperbólica y (c) Función Gaussiana.

B. Funciones de Activación

Las funciones de activación usadas en cada neurona son:

1) Función Logística (figura 2a)

$$\text{logsig}(n) = 1 / (1 + e^{-n}) \quad (7)$$

2) Función Tangente Hiperbólica (figura 2b)

$$\text{tansig}(n) = 2 / (1 + e^{-2n}) - 1 \quad (8)$$

3) Función Gaussiana (figura 2c)

$$\text{radbas}(n) = e^{-n^2} \quad (9)$$

IV. ANÁLISIS DE RESULTADOS

El análisis se realizara inicialmente por función objetivo y posteriormente desde un punto de vista global, basado en la evaluación de la suma de los errores de entrenamiento y validación de las últimas cinco redes simuladas para cada función objetivo, variando la función de activación, suprimiendo así la inestabilidad de estas con menor número de neuronas.

En adelante las curvas mostradas en las gráficas y tablas se denotaran de la siguiente forma:

- Función Logística como logsig
- Función Tangente Hiperbólica como tansig
- Función Gaussiana como radbas

La tabla 1 presenta el análisis comparativo de los resultados de las diferentes simulaciones, mostradas en las figuras 3 a 20 El criterio de mejor o peor función de activación se obtiene del mínimo o máximo valor de la sumatoria de los errores de simulación de la red, con sus distintas cantidades de neuronas.

Tabla 1. Análisis comparativo funciones objetivo sin ruido.

| Función | Entrenamiento | | Validación | |
|---------|---------------|--------|------------|--------|
| | Mejor | Peor | Mejor | Peor |
| F1 | radbas | tansig | radbas | tansig |
| F2 | logsig | tansig | logsig | tansig |
| F3 | tansig | logsig | tansig | logsig |
| F4 | radbas | tansig | logsig | tansig |
| F5 | radbas | tansig | logsig | radbas |

Tabla 2. Análisis comparativo funciones objetivo con ruido.

| Función | Entrenamiento | | Validación | |
|---------|---------------|--------|------------|--------|
| | Mejor | Peor | Mejor | Peor |
| F1 | radbas | tansig | radbas | tansig |
| F2 | radbas | tansig | logsig | tansig |
| F3 | logsig | radbas | logsig | radbas |
| F4 | radbas | logsig | radbas | logsig |
| F5 | radbas | tansig | logsig | tansig |

Correlacionando la tabla 1 con las figuras 3 y 4 y de las tablas 1 y 2 se debe interpretar que radbas presenta mejor desempeño, debido a que posee el mínimo error de

entrenamiento y validación para la mayoría de los casos, y la función de activación tansig presenta el peor desempeño, como puede concluirse de estas mismas figuras. Conclusiones análogas se deben considerar para los demás funciones, con y sin ruido, de acuerdo con los datos de las tablas 1 y 2.

Algunos aspectos para resaltar en cada caso de las funciones sin ruido son los siguientes:

--En F1, para el error de validación, radbas presenta un comportamiento estable, indicando que la red converge a una buena solución independiente del número de neuronas (en este caso hasta 13 neuronas).

--En F2, después de 10 neuronas, el error de validación se comporta similar para las tres funciones de activación.

--La función tansig en F1 y F2, antes de 8 neuronas presenta un error de validación creciente, aún cuando el error de entrenamiento es decreciente. En F1 a partir de la décima neurona el error cae abruptamente y en F2, sucede a partir de la octava.

--En F3, inicialmente los errores de validación y entrenamiento son muy altos para las funciones logsig y radbas en comparación con tansig, pero a partir de la séptima neurona son similares.

--En F4, a pesar de que los errores de entrenamiento para logsig y radbas son similares, siendo levemente inferior radbas, logsig es mejor en la validación. Por su parte, tansig presenta un comportamiento muy inestable.

--En F4 y F5 los resultados de mejor función de activación son diferentes para el entrenamiento y validación, posiblemente por ser éstas las funciones objetivo más complejas.

--En F5, a partir de la novena neurona es tansig la que arroja los menores errores de entrenamiento y validación y su comportamiento es muy similar al de logsig.



Figura 3. Error de entrenamiento Función de Interacción Simple sin ruido

Algunos aspectos para resaltar en cada caso de las funciones con ruido de los datos mostrados en la tabla 2 son los siguientes:

--En F1 tansig presenta un error significativo con respecto a las otras dos funciones en la red neuronal de cuatro neuronas. A partir de 5 neuronas las tres funciones de activación presentan un comportamiento aceptable en entrenamiento y validación.

--En F2 radbas y logsig para todas las simulaciones variando las neuronas, presentan un comportamiento similar y constante en el error de validación. La función de activación tansig para esta función objetivo presenta un comportamiento totalmente diferente al que ha mostrado en las demás funciones, donde se resalta su comportamiento inaceptable a partir de la 8 neurona.

--En F3 el comportamiento en los errores de validación para las tres funciones de activación a partir de la séptima neurona es aceptable y similar. Con pocas neuronas radbas presenta el error de validación alto comparado con las demás funciones de activación.

--En F4 el error de validación para tansig y logsig es decreciente al aumentar el numero de neuronas, lo cual no sucede con radbas.

--En F5 tansig presenta el menor error de validación a pesar de tener el peor error de entrenamiento.

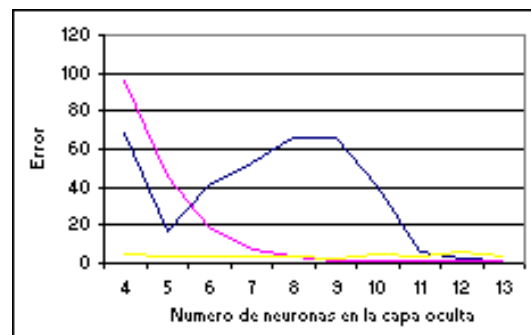


Figura 4. Error de validación Función de Interacción Simple sin ruido

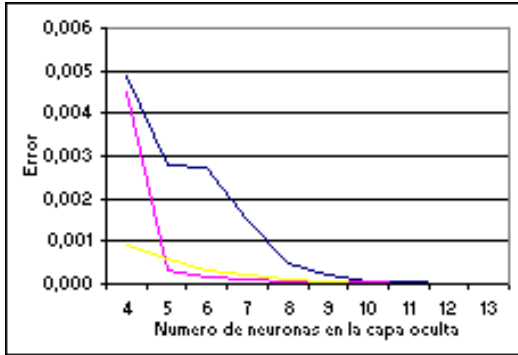


Figura 5. Error de entrenamiento Función Radial sin ruido

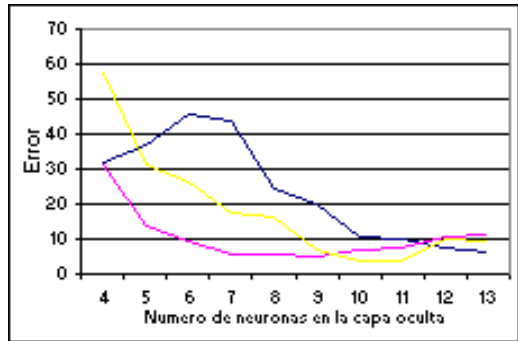


Figura 6. Error de validación Función Radial sin ruido

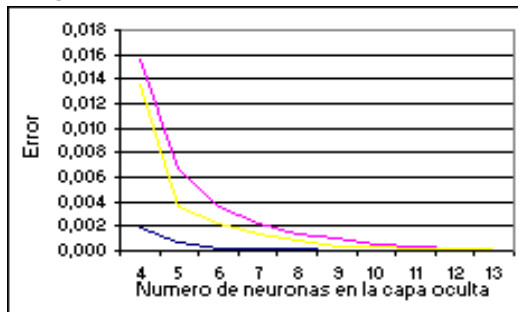


Figura 7 Error de entrenamiento Función Harmónica sin ruido

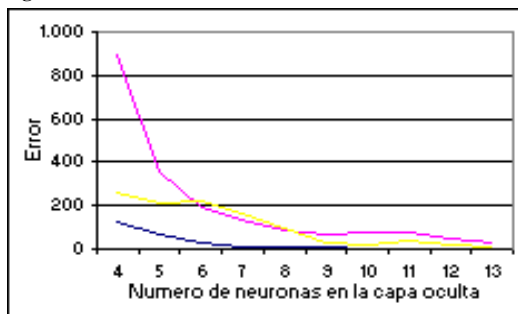


Figura 8. Error de validación Función Harmónica sin ruido

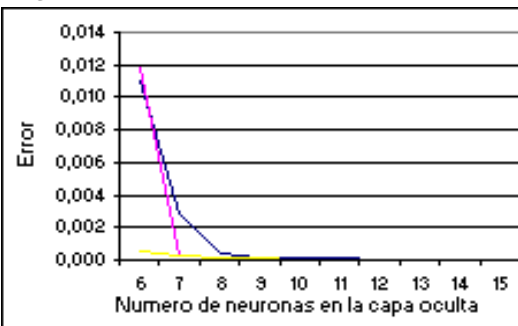


Figura 9. Error de entrenamiento Función Aditiva sin ruido

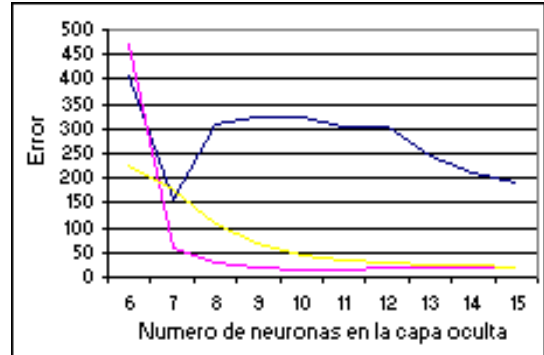


Figura 10. Error de validación Función Aditiva sin ruido

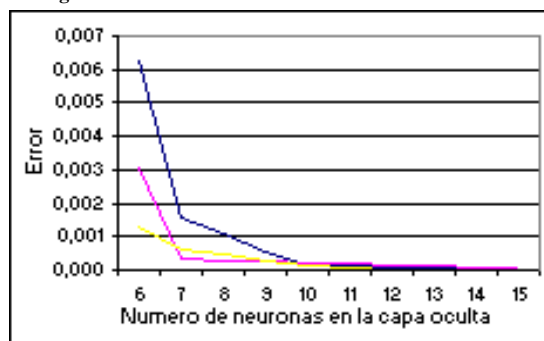


Figura 11. Error de entrenamiento Función de Iteración Complicada sin ruido

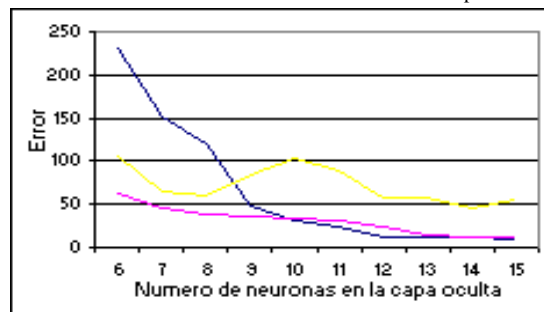


Figura 12. Error de validación Función de Iteración Complicada sin ruido

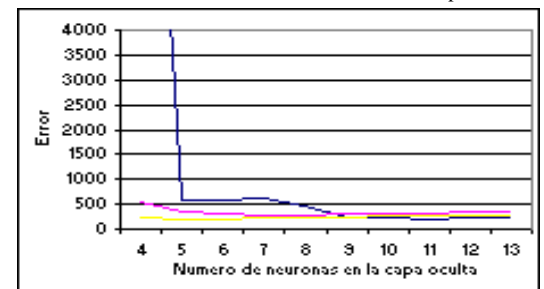


Figura 13. Error de validación Función de Interacción Simple con ruido

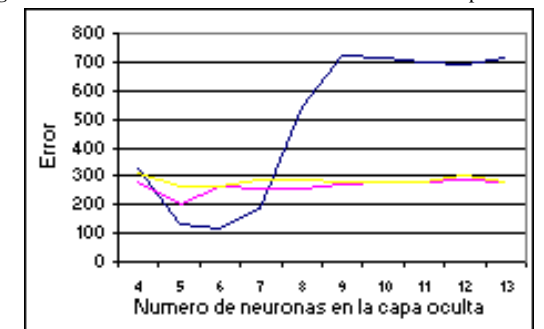


Figura 14. Error de validación Función Radial con ruido

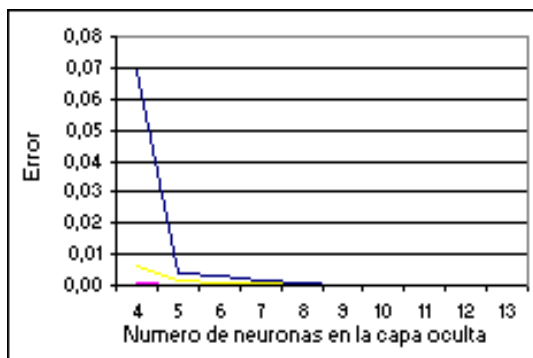


Figura 15. Error de entrenamiento Función Harmónica con ruido

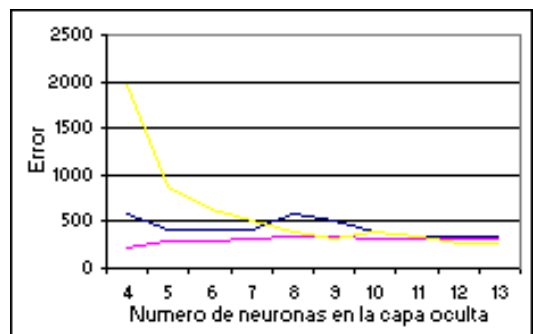


Figura 16. Error de validación Función Harmónica con ruido

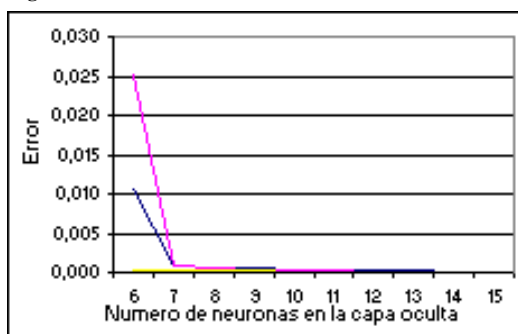


Figura 17. Error de entrenamiento Función Aditiva con ruido

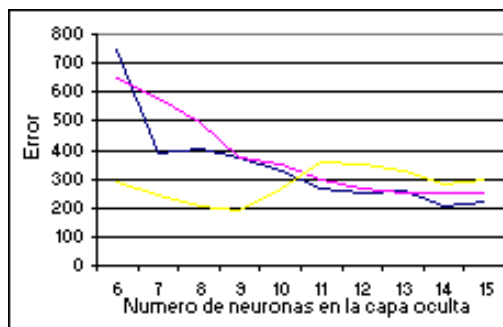


Figura 18. Error de validación Función Aditiva con ruido

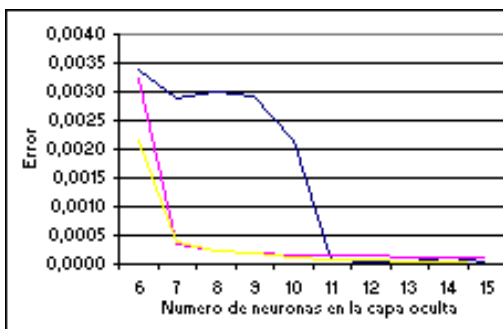


Figura 19. Error de entrenamiento Función de Iteración Complicada con ruido

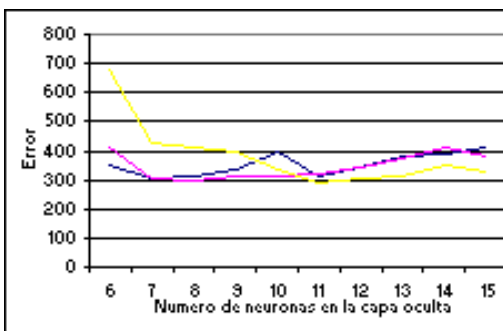


Figura 20. Error de validación Función de Iteración Complicada con ruido

V. ANÁLISIS GLOBAL PARA FUNCIONES

Del análisis de la tabla 3, 4 y las figuras 21, 22, 23 y 24 se tiene:

--En 4 de 5 funciones objetivo simuladas, radbas es la función de activación con menor error de entrenamiento, lo cual es congruente con el error de validación, que no es significativamente diferente al comportamiento del error global de la siguiente mejor función.

--El error de entrenamiento en tansig para las funciones

con ruido es significativamente superior a las otras funciones de activación.

--El error de validación en radbas y logsig oscila en un rango limitado, contrario para tansig donde su error en F2 (con ruido) y F4 (sin ruido) es desproporcionado.

Con base en lo anterior, aplicar una función de activación como radbas o logsig en la capa oculta de una red neuronal de tipo feedforward, es mejor opción.

Tabla 3. Totales de funciones sin ruido

| | Entrenamiento | | | Validación | | |
|-------|---------------|------------|-------------|------------|----------|----------|
| | tansig | logsig | radbas | tansig | logsig | radbas |
| F1 | 0,1229418 | 0,0010450 | 0,0003315 | 17204,39 | 3428,34 | 2440,36 |
| F2 | 0,0225119 | 0,0041001 | 0,0011856 | 4836,36 | 2648 | 2827,26 |
| F3 | 0,07938 | 0,00123 | 0,00970 | 4300,28 | 3051,88 | 5972,84 |
| F4 | 0,013665 | 0,027490 | 0,001285 | 3449,55 | 3756,05 | 2798,65 |
| F5 | 0,014643368 | 0,00481239 | 0,0032946 | 3536,82 | 3464 | 3826,38 |
| Total | 0,253143553 | 0,03867695 | 0,015799709 | 33327,4 | 16348,27 | 17865,49 |

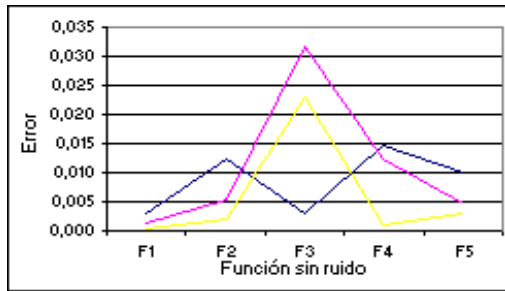


Figura 21. Error total de entrenamiento por función objetivo sin ruido.

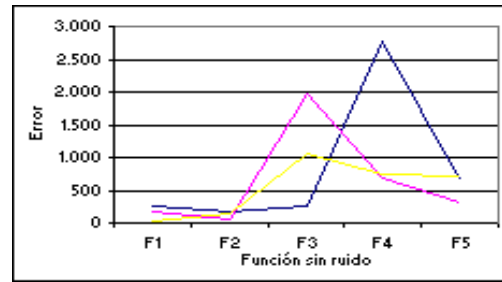


Figura 23. Error total de validación por función objetivo sin ruido.

Tabla 4. Totales de funciones con ruido

| | Entrenamiento | | | Validación | | |
|-------|---------------|------------|-------------|------------|---------|---------|
| | tansig | logsig | radbas | tansig | logsig | radbas |
| F1 | 0,0003974 | 0,0000852 | 0,0000098 | 1133,38 | 1641,86 | 1323,25 |
| F2 | 0,0008277 | 0,0001397 | 0,0000155 | 3537,08 | 1395,19 | 1419,99 |
| F3 | 0,00063 | 0,00009 | 0,00033 | 1913,04 | 1608,64 | 1582,65 |
| F4 | 0,000845 | 0,000344 | 0,000065 | 1200,84 | 1320,01 | 1608,14 |
| F5 | 0,000313 | 0,000632 | 0,000224 | 1836,84 | 1831,00 | 1584,79 |
| Total | 0,003017093 | 0,00129432 | 0,000645525 | 9621,18 | 7796,7 | 7518,82 |

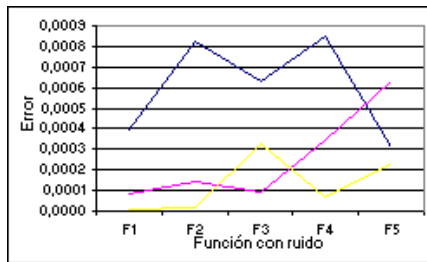


Figura 22. Error total de entrenamiento por función objetivo con ruido.

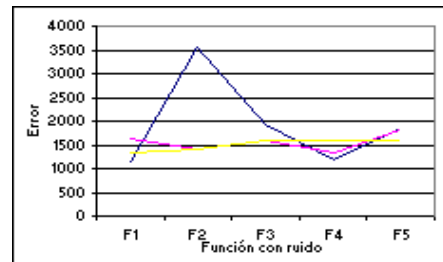


Figura 24. Error total de validación por función objetivo con ruido.

VI. CONCLUSIONES

De acuerdo con los resultados obtenidos

--Sin importar el número de neuronas, para aproximar las funciones simples (F1, F2) sin ruido las tres funciones de activación presentan buenos resultados. Para funciones complejas (F4, F5) sin ruido y redes con pocas neuronas logsig presenta mejor desempeño, en el caso de redes con muchas neuronas puede incluirse también a radbas. Para funciones como F3, sin ruido, tansig presenta mejor desempeño sin importar el número de neuronas.

--Para funciones simples o complejas con ruido, logsig y radbas son apropiadas, en las complejas también tansig. En funciones simples con ruido no se puede concluir sobre el desempeño de tansig

--En las funciones objetivo con ruido radbas permaneció relativamente constante al incrementar el número de neuronas, lo cual sugiere que esta función puede utilizarse con igual desempeño independientemente del número de neuronas.

--La magnitud de los errores de validación para las funciones con ruido es significativamente superior a los obtenidos para las funciones sin ruido, lo cual se ve de manifiesto con las tres funciones de activación estudiadas.

VII. TRABAJOS FUTUROS

Podría complementarse el análisis comparativo variando la tasa de aprendizaje y evaluando el comportamiento de los errores de entrenamiento y generalización, para establecer el efecto de este parámetro en el tiempo de entrenamiento y razón de caída del error.

Dado que la escogencia de los pesos iniciales puede ser crucial en el desempeño de la red neuronal y que es recomendable probar con diferentes conjuntos de valores iniciales para tratar de obtener buenos resultados. Sería conveniente realizar la comparación ampliando el criterio de selección de dichos pesos. Como se menciona en [1] generando los pesos iniciales aleatoriamente 50 veces y seleccionando el modelo que obtenga el menor promedio entre la suma de cuadrados de los errores de ajuste y predicción.

Comparaciones con un rango más amplio de funciones de activación y un mayor número de corridas de las redes neuronales por función objetivo, complementarían las perspectivas dadas en este artículo.

REFERENCIAS

[1] Collantes Duarte, J. V, Predicción con Redes Neuronales: Comparación con las Metodologías de Box y Jenkins. Magister Scientiae en Estadística Aplicada. Universidad de los Andes. Mérida, Venezuela. 2001, 25-45.

- [2] Demuth H., and Beale M. Neural Network Toolbox User's Guide For Use with MATLAB®. The MathWorks, Inc. 2003, 5-37.
 - [3] Hong-Choon., and Wong Y. A Comparative Study on the Multi-layered Perceptrons with One and Two Hidden Layers. TS5B-3, M2USIC, 2004.
 - [4] Hwang, J., Lay, S., Maechler, M., Martin, R. D., Schimert, J. Regression Modeling in Back-Propagation and Projection Pursuit Learning. IEEE. Transaction on Neural Networks, Vol 5. No. 3. May 1994.
 - [5] Jang, J. R., Sun, C., and Mizutani, E. Neuro-Fuzzy And Soft Computing. A Computational Approach to Learning and Machine Intelligence. Prentice Hall, USA, 1997, 234.
 - [6] Kwok, T., and Yeung, D. Objective Functions for Training New Hidden Units in Constructive Neural Networks. IEEE, Transactions on Neural Networks, Vol. 8, No. 5, Sep. 1997.
 - [7] Piekniowski, F., Rybicki, L. Visual Comparison of Performance for Different Activation Functions in MLP Networks. IEEE, 2004.
 - [8] Hara, K.; Nakayama, K. Comparison of activation functions in multilayer neural network for pattern classification. IEEE World Congress on Computational Intelligence, 1994. IEEE International Conference on Volume 5, Issue , 27 Jun-2 Jul 1994 Page(s):2997 - 3002 vol.5.
- financieras usando inteligencia computacional. Otros tópicos de investigación en los que trabaja incluyen Finanzas Computacionales, Compiladores, Neurocomputación.

Luis Everley Llano Zuleta. Ingeniero Electricista de la Universidad Nacional de Colombia (1997) con Diploma en Ingeniería del Mantenimiento de la Universidad de Antioquia (2004). Analista de Operación de 1997 a 2005 e Ingeniero de Operación desde 2005 a la fecha en Interconexión Eléctrica S.A. E.S.P. (ISA). Actualmente, es estudiante candidato a la Maestría en Ingeniería de Sistemas de la Universidad Nacional de Colombia - Sede Medellín (2007), línea de investigación en Inteligencia Artificial.

Andrés Felipe Hoyos Palacio. Estudiante de último semestre de Ingeniería de Sistemas e Informática de la Facultad de Minas de la Universidad Nacional de Colombia Sede Medellín. Integrante del GIDIA: Grupo de Investigación y Desarrollo en Inteligencia Artificial, Categoría A de Colciencias. Ha trabajado, durante el último año como monitor académico de investigación, en los dos proyectos del GIDIA titulados: "Modelo de Sistema Multi-Agente de Cursos Adaptativos Integrados con Ambientes Colaborativos de Aprendizaje" y "Herramientas informáticas para el diagnóstico automático de eventos en líneas de transmisión de energía eléctrica", este último en colaboración con GAUNAL (Grupo de Automática de la Universidad Nacional de Colombia), ISA y cofinanciado por Colciencias. Ha participado como ponente en varios eventos nacionales.

Francisco Javier Arias Sánchez. Estudiante de último semestre de Ingeniería de Sistemas e Informática de la Facultad de Minas de la Universidad Nacional de Colombia Sede Medellín. Integrante del GIDIA: Grupo de Investigación y Desarrollo en Inteligencia Artificial, Categoría A de Colciencias. Ha trabajado, durante los últimos dos años, como monitor académico de investigación, en los tres proyectos del GIDIA titulados: "Diseño e implementación de un sistema multi-agente que simule el comportamiento del mercado energético en Colombia", "Modelo de Sistema Multi-Agente de Cursos Adaptativos Integrados con Ambientes Colaborativos de Aprendizaje" y "Herramientas informáticas para el diagnóstico automático de eventos en líneas de transmisión de energía eléctrica", este último en colaboración con GAUNAL (Grupo de Automática de la Universidad Nacional de Colombia), ISA y cofinanciado por Colciencias.

Juan David Velásquez Henao. Profesor Asociado, Universidad Nacional de Colombia sede Medellín. Ingeniero Civil. Magíster en Ingeniería de Sistemas PhD(c) en Sistemas Energéticos. Integrante de GIDIA: Grupo de Investigación y Desarrollo en Inteligencia Artificial, Categoría A de Colciencias. El área de énfasis de su investigación es el modelado y predicción de series económicas y

Universidad Nacional de Colombia Sede Medellín Facultad de Minas



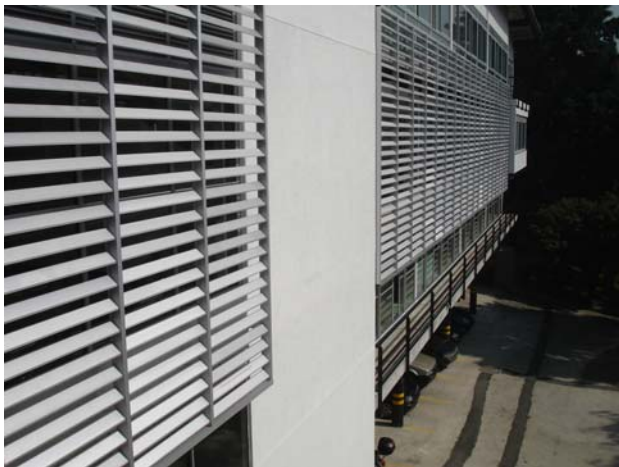
Escuela de Ingeniería de Sistemas Grupos de Investigación

Grupo de Investigación en Sistemas e Informática

Categoría A de Excelencia Colciencias
2004 - 2006 y 2000.

GIDIA: Grupo de Investigación y Desarrollo en Inteligencia Artificial

Categoría A de Excelencia Colciencias
2006 - 2009.



Centro de Excelencia en Complejidad Colciencias 2006

Escuela de Ingeniería de Sistemas
Dirección Postal:
Carrera 80 No. 65 - 223 Bloque M8A
Facultad de Minas. Medellín - Colombia
Tel: (574) 4255350 Fax: (574) 4255365
Email: esistema@unalmed.edu.co
<http://pisis.unalmed.edu.co/>



Grupo de Ingeniería de Software

Categoría C Colciencias 2006.

Grupo de Finanzas Computacionales

Categoría C Colciencias 2006.

