

RED NEURONAL BACKPROPAGATION: HERRAMIENTAS GRAFICAS INTERACTIVAS, IMPLEMENTACION COMPUTACIONAL Y APLICACIONES

Juan Fernando Espinosa*
Ingeniero de Sistemas

Jairo Pertuz Campo**
Físico

RESUMEN

En nuestro trabajo analizamos y describimos este método de entrenamiento, conocido como backpropagation (de errores) ó la regla delta generalizada. Se trata simplemente de un método descendente de gradiente, con el cual podemos minimizar el error cuadrático total de la salida calculada por la red. El backpropagation puede entrenar redes multicapas feedforward, con funciones de transferencia diferenciables para ejecutar funciones de aproximación, asociación de patrones, y clasificación de patrones. (Pueden entrenarse también otros tipos de redes, aunque la red multicapa es la más comúnmente usada.) El término backpropagation se refiere al proceso por el cual pueden calcularse las derivadas de error de la red, con respecto a los pesos de la red y biases. Este proceso puede usarse con varias estrategias de optimización diferentes.

El proceso de entrenamiento de una red backpropagation involucra tres fases: la feedforward de los patrones de entrenamiento de entrada, el cálculo y la propagación inversa del error asociado, y el ajuste de los pesos. Después del entrenamiento, la aplicación de la red incluye solamente las computaciones de la fase feedforward. Aún si el entrenamiento es bajo, una red entrenada puede producir su salida muy rápidamente. Se han desarrollado numerosas variaciones de backpropagation , con el propósito de mejorar la velocidad del proceso de entrenamiento.

Como tratamiento alternativo computacional, proponemos el empleo de las herramientas gráficas interactivas, a saber, la interfaz gráfica de usuario (GUI) que acompaña al MATLAB y la herramienta para redes neuronales (nntool), con las cuales hacemos las implementaciones computacionales para las compuertas lógicas AND y XOR. Finalmente, presentamos un bosquejo descriptivo, correspondiente a la operación financiera de créditos en sus

<http://matlab.udes.edu.co>

http://es.geocities.com/matlab_colombia/diamatlabnov3.html

<http://www.compelect.com.co/FormularioDiaMATLAB.html>

diferentes modalidades, en la cual mostramos la presencia de la red feedforward multicapa, las funciones de transferencia, el algoritmo backpropagation y los parámetros de entrenamiento tratados, en lo que atañe al proceso de diseño, entrenamiento y simulación de la red neuronal, con la cual fue posible hacer el proceso de clasificación y obtención de los puntajes, como resultado de la solicitud de los respectivos créditos.

Palabras claves: Red neuronal, entradas, pesos, bias, función de transferencia, salidas, sumación, sigmoide, algoritmo de entrenamiento, feedforward, red multicapa, salida deseada, gradiente descendente, entrenamiento por lote, tan-sigmoide, radbas, pesos de la entrada, los pesos de las capas, momentum, mínimo local, backpropagation, GUI, nntool.

I. INTRODUCCIÓN

El Backpropagation fue creado mediante la generalización de la regla de aprendizaje Widrow-Hoff para redes multicapas y las funciones de transferencia diferenciables no lineales. Los vectores de entrada y los correspondientes vectores deseados se usan para entrenar una red hasta cuando ella pueda aproximar una función, asociando los vectores de entrada con los vectores de salida específicos, ó clasificar los vectores de entrada de una manera apropiada tal como nosotros la definimos. Las redes con biases, una capa sigmoide, y una capa de salida lineal son competentes para aproximar cualquier función con un número finito de discontinuidades.

El backpropagation estándar es un algoritmo de gradiente descendente, como lo es la regla de aprendizaje Widrow-Hoff, en la cual los pesos de la red son movidos a lo largo del negativo del gradiente de la función de ejecución. El término backpropagation se refiere a la manera como el gradiente es calculado para redes multicapa no lineales. Existe un cierto número de variaciones en el algoritmo básico, las cuales están basadas en otras técnicas de optimización, tales como el gradiente conjugado y los métodos de Newton. Con el Toolbox de Redes Neuronales podemos implementar estas variaciones.

Las redes backpropagation entrenadas de manera apropiada, se orientan a dar respuestas razonables cuando se les presentan entradas que aún no han sido consideradas. Típicamente, una nueva entrada conduce a una salida similar a la salida correcta para vectores de entrada que se han empleado en el entrenamiento, los cuales a su vez son similares a las nuevas entradas que están siendo presentadas. Esta generalización con propiedad hace posible entrenar una red sobre un conjunto representativo de las parejas entrada / salida deseada, y se obtienen buenos resultados sin entrenar la red sobre todas las posibles parejas de entrada / salida. Existen dos características del Toolbox de Redes Neuronales las cuales están diseñadas para mejorar la generalización y regularización de la red y la detención remota.

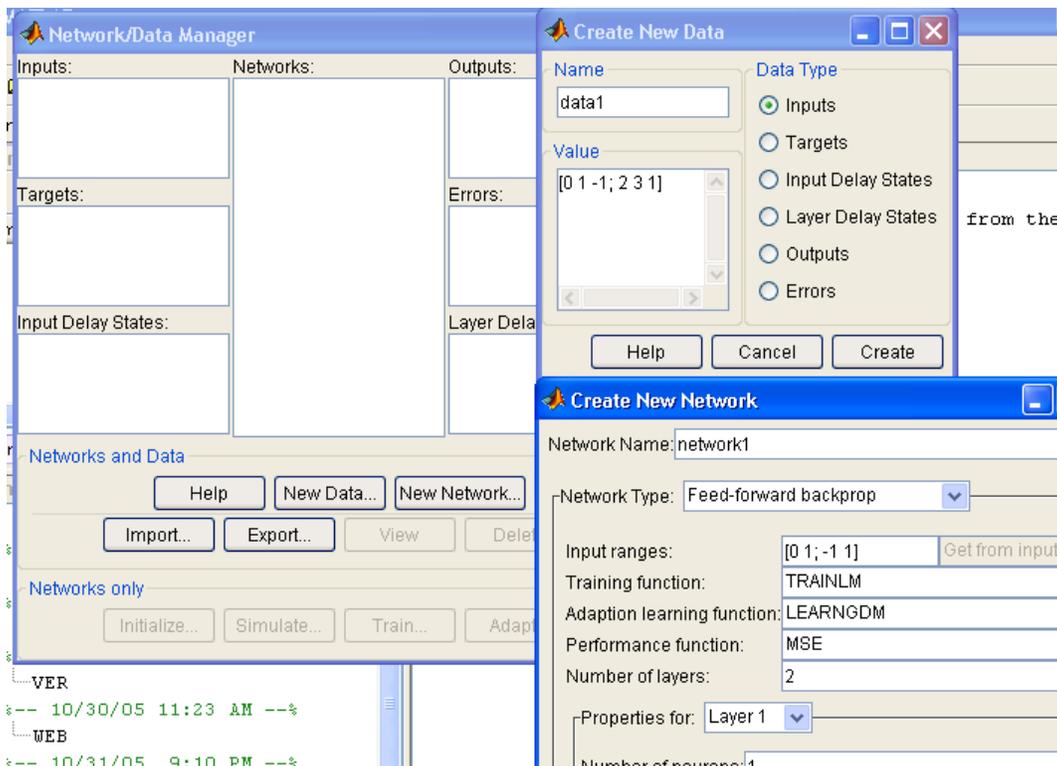
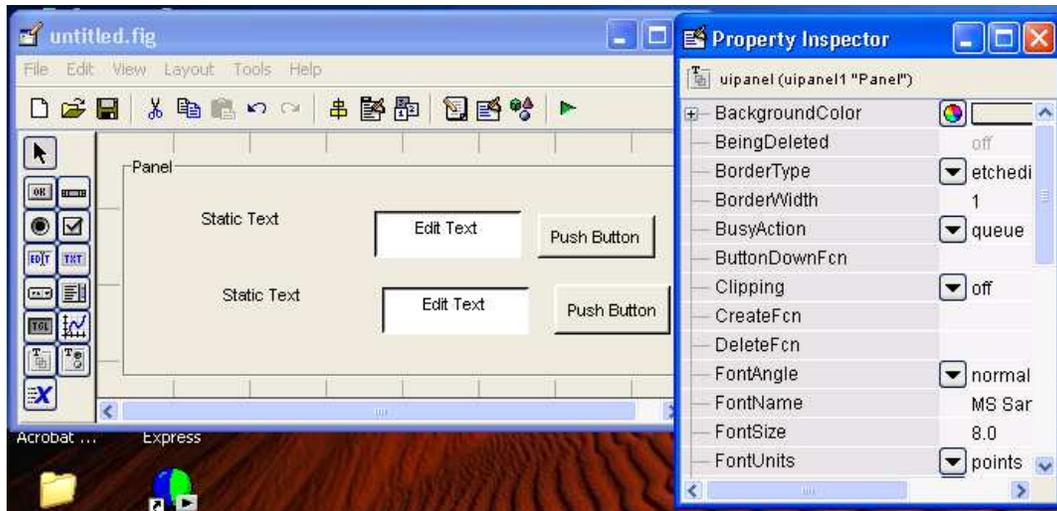
<http://matlab.udes.edu.co>

http://es.geocities.com/matlab_colombia/diamatlabnov3.html

<http://www.compelect.com.co/FormularioDiaMATLAB.html>

II. HERRAMIENTAS INTERACTIVAS

Interfaz gráfica de usuario (GUI) y herramienta para redes neuronales (nntool): recursos gráficos para el diseño, entrenamiento y simulación de redes neuronales.

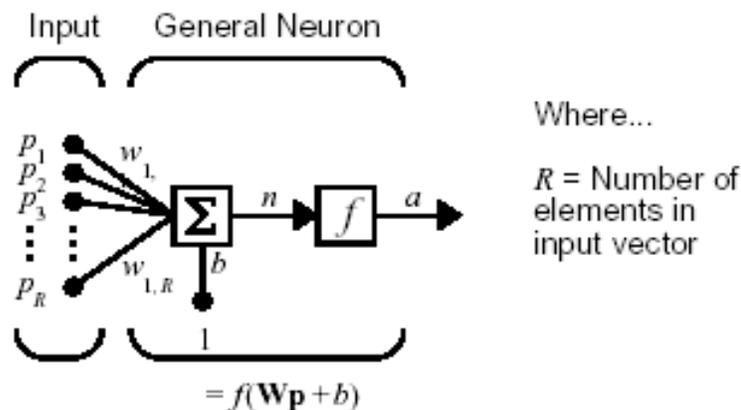


III. ARQUITECTURA

La arquitectura de la red más comúnmente utilizada conjuntamente con el algoritmo backpropagation es la red feedforward multicapa. Las rutinas en el Toolbox de Redes Neuronales pueden emplearse para entrenar más redes generales.

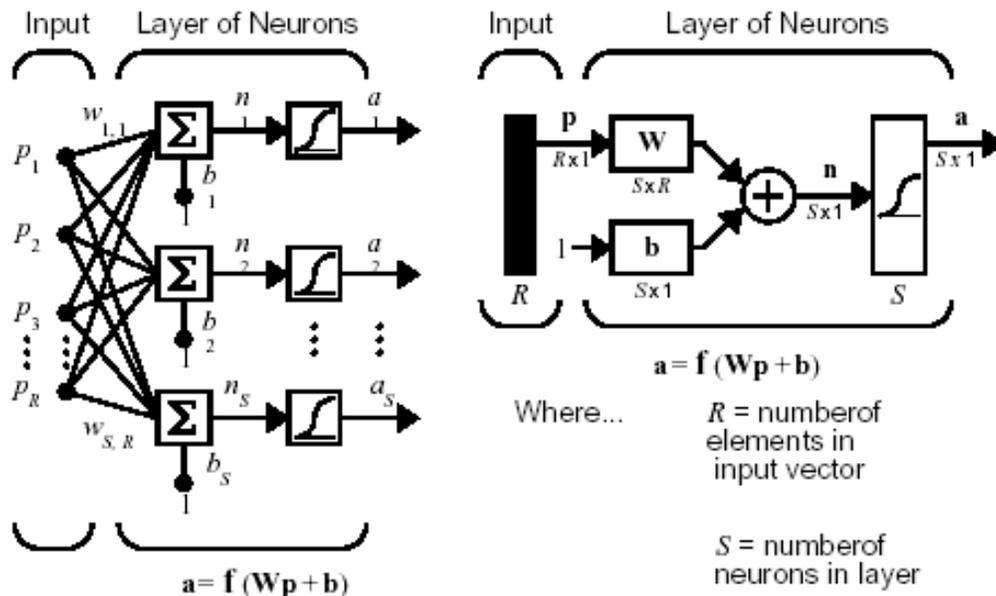
Modelo Neuronal (tansig, logsig, purelin)

Una neurona elemental con R entradas aparece abajo. Cada una de las entradas es ponderada con un w apropiado. La suma de las entradas ponderadas y el bias forman la entrada a la función de transferencia f . Las neuronas pueden usar cualquier función de transferencia diferenciable f para generar su salida.



Red Feedforward

Una red de una sola capa de S neuronas **logsig** que tienen R entradas, la mostramos abajo en forma detallada sobre la izquierda y con un diagrama de capa sobre la derecha.



Las redes multicapas a menudo usan la función de transferencia log-sigmoide (logsig), de manera alternativa, la tan-sigmoid (tansig), y también, ocasionalmente, la función de transferencia purelin es usada en las redes backpropagation. Las tres funciones de transferencia descritas aquí son las más comúnmente usadas para backpropagation, pero otras funciones de transferencia diferenciables pueden ser creadas y usadas con backpropagation si así lo deseamos.

IV. ALGORITMOS

Existen muchas variaciones del algoritmo backpropagation, varios de los cuales trataremos. La implementación más simple de las reglas de aprendizaje del backpropagation, actualiza los pesos de la red y biases en la dirección en la cual la función ejecución decrece más rápidamente- el negativo del gradiente. Una iteración de este algoritmo se escribe así

$$x_{k+1} = x_k - \alpha_k g_k$$

donde x_k es un vector de pesos actuales y biases, g_k es el gradiente actual, y α_k es la velocidad de aprendizaje.

Entrenamiento rápido

Conviene mencionar dos algoritmos de entrenamiento backpropagation : el gradiente descendente, y el gradiente descendente con momentum. Estos dos métodos a menudo son demasiado lento para los problemas prácticos. Existen varios algoritmos de alta ejecución, los cuales pueden converger desde diez hasta cien veces más rápido que los algoritmos mencionados anteriormente.

Otra categoría de algoritmos rápidos usa técnicas de optimización numérica estándar. Resulta oportuno mencionar tres tipos de técnicas de optimización numéricas para el entrenamiento de las redes neuronales: **gradiente conjugado** (traincgf, traincgp, traincgb, trainscg), **quasi-Newton** (trainbfg, trainoss), y **Levenberg-Marquardt** (trainlm).

V. IMPLEMENTACIONES COMPUTACIONALES

Mediante el empleo de la **GUI** y la **nntool** implementamos las compuertas lógicas AND y XOR, mostrando de manera categórica la facilidad, eficiencia, e integración de los procesos de diseño, entrenamiento y simulación de la red neuronal feedforward multicapa con los algoritmos backpropagation.

VI. APLICACIÓN FINANCIERA

La aplicación considerada corresponde a la operación financiera de créditos : Tarjeta, Consumo e Hipotecario. Para tal efecto empleamos una red feedforward multicapa, con función de transferencia radbas, función de ejecución trainlm, y los parámetros de entrenamiento: show, velocidad de aprendizaje (lr), epochs y goal. Los valores de salida que la red ejecutó, se obtuvieron mediante la operación de clasificación y predicción (scoring)

VII COMENTARIOS FINALES

- La arquitectura de red que más armoniza con el algoritmo backpropagation , es la red feedforward multicapa.
- El algoritmo backpropagation puede mejorar su ejecución, si se le ajustan ciertas modificaciones.
- Las modificaciones involucran cambios al procedimiento para actualizar los pesos, y alternativas a las funciones de activación
- La interfaz gráfica de usuario (GUI) y la herramienta para redes neuronales (nntool), son herramientas que permiten diseñar y entrenar eficientemente redes neuronales.

- Las redes feedforward y el algoritmo backpropagation, ejecutan eficientemente aplicaciones académicas y empresariales.

BIBLIOGRAFIA

- Freeman, James A., Skapura, David M. Redes Neuronales. Algoritmos, aplicaciones y técnicas de programación. Addison-Wesley Iberoamericana, 1993
- Hilera, José R., Martínez, Víctor J. Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones. Addison-Wesley Iberoamericana, 1995.
- Fausett, Laurene. Fundamentals of neuronal networks: Architectures, Algorithms, and applications. ED. Prentice Hall, 1994.
- Manual de Matlab. The MATHWORKS Inc.
- Neural Networks TOOLBOX. User's Guide. The MATHWORKS Inc.

* **Juan Fernando Espinosa.**

Ingeniero De Sistemas De La Universidad De Medellín, Tecnólogo En Sistematización De Datos Del Politécnico Colombiano Jaime Isaza Cadavid, Investigador En Redes Neuronales.

Email. Ktaju@epm.net.co

DESCRialmente ponderadas

** **Jairo Pertuz Campo.**

Físico, Universidad de Antioquia. Estudios de Maestría, Universidad de Puerto Rico. Investigador en Redes Neuronales Artificiales. Instructor, asesor en MATLAB y sus herramientas especializadas. Asesor de investigación en el programa de Ingeniería Financiera, de la Universidad de Medellín.

E-Mails: jpertuz@udem.edu.co, pertuzjairo@yahoo.es

<http://matlab.udes.edu.co>

http://es.geocities.com/matlab_colombia/diamatlabnov3.html

<http://www.compelect.com.co/FormularioDiaMATLAB.html>