Red Neuronal Backpropagation.

Backpropagation neuronal network.

Sebastian Lopera Osorio, Luis David Restrepo Cadavid, Edwin Alexander Enciso
Sebastian.lopera@utp.edu.co, david95rc@utp.edu.co, edwinenciso@utp.edu.co

Resumen- Las redes neuronales artificiales son sistemas dinámicos en tanto que el aprendizaje requiere de un proceso iterativo en el que los pesos asociados a las conexiones sean modificados, una y otra vez, hasta que los pesos de la red no varíen, permaneciendo estables. Como ha sido descrito en esta sección, las expresiones computacionales con las que se calcula el cambio de los pesos son la solución de una o más ecuaciones diferenciales, tal es el caso de las redes neuronales perceptrón y backpropagation **[1]**

Palabras clave- **Red neuronal, perceptrón, inteligencia artificial, propagación hacia atrás.**

***Abstract*—Artificial neural networks are dynamic systems insofar as learning requires an iterative process in which the weights associated with the connections are modified, again and again, until the weights of the network do not vary, remaining stable. As it has been described in this section, the computational expressions with which the change of weights is calculated are the solution of one or more differential equations, such is the case of the neural networks perceptron and backpropagation.**

Keywords- **Neural network, perceptron, artificial intelligence, backpropagation.**

I. INTRODUCCION

El concepto de redes neuronales suele ser explicado fácilmente cuando se relaciona con el aprendizaje humano. Cuando un niño va a aprender a identificar los colores, primero debe mirar el color y luego con indicaciones, aprende las características de ese color, esto le permite identificar cada uno de ellos en el futuro sin tener que recibir nuevamente la información. Lo mismo sucede con las redes neuronales, pueden tomar decisiones una vez han sido entrenadas con información previa. Si se desea que identifique colores como se mencionaba anteriormente, se debe suministrar información que permita identificar cada color tal cual como lo haría el niño en su aprendizaje.
En resumen, una red neuronal es la representación computacional de las neuronas del cerebro humano, las cuales intervienen en las decisiones que tomamos a diario y son claves al momento de adquirir conocimientos. Para comprender mejor el funcionamiento de una red neuronal, este tutorial se enfoca en el análisis del perceptrón puesto que es el modelo más simple de una red neuronal. [3]

II. CONTENIDO

Red Neuronal Backpropagation

La propagación hacia atrás de errores o retropropagación (del inglés backpropagation) es un método de cálculo del gradiente utilizado en algoritmos de aprendizaje supervisado utilizados para entrenar redes neuronales artificiales. El método emplea un ciclo propagación – adaptación de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas siguientes de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas. [2]

En redes neuronales se busca ajustar los pesos de cada neurona de tal manera que se minimice el error. El algoritmo de backpropagation nos indica cuanto de culpa tiene cada neurona del error global cometido.[3]

La forma en que como se calcula la culpa que tiene cada neurona en el error es lo que da sentido al nombre de backpropagation, ya que primeramente calcula la culpa del error de cada neurona de la última capa y lo va propagando hacia atrás para ver cuanta culta tienen el resto.[3]

Se podría decir que pondera el reparto del error para cada una de las neuronas de la red.[3]

El algoritmo de backpropagation determina la culpa del error, calculando las derivadas parciales de la función de coste con respecto a cada una de las variables. [3]

Algoritmo para realizar una red neuronal Backpropagation

Se considera una etapa de funcionamiento donde se presenta, ante la red entrenada, un patrón de entrada y éste se transmite a través de las sucesivas capas de neuronas hasta obtener una salida y, después, una etapa de entrenamiento o aprendizaje donde se modifican los pesos de la red de manera que coincida la salida deseada por el usuario con la salida obtenida por la red. Etapa de funcionamiento Cuando se presenta un patrón p de entrada Xp : x p 1 , . . . , x p i , . . . , x p N , éste se transmite a través de los pesos wji desde la capa de entrada hacia la capa oculta. Las neuronas de esta capa intermedia transforman las señales recibidas mediante la aplicación de una función de activación proporcionando, de este modo, un valor de salida. Este se transmite a través de los pesos vkj hacia la capa de salida, donde aplicando la misma operación que en el caso anterior, las neuronas de esta última capa proporcionan la salida de la red. Este proceso se resume en lo siguiente: La entrada total o neta que recibe una neurona oculta j, net^p_j , es:[4]

Fecha de Recepción: (Letra Times New Roman de 8 puntos)
Fecha de Aceptación: Dejar en blanco

$$net_j^p = \sum_{i=1}^{N} w_{ji} x_i^p + \theta_j$$

Imagen 1 [4]

Donde θ_j es el umbral de la neurona que se considera como un peso asociado a una neurona ficticia con valor de salida igual a 1. El valor de salida de la neurona oculta j, y $^{p}_{j}$, se obtiene aplicando una función f(·) sobre su entrada neta:[4]

$$y_j^p = f\left(net_j^p\right)$$

Imagen 2 [4]

De igual forma, la entrada neta que recibe una neurona de salida k, net$^{p}_{k}$, es:[4]

$$net_k^p = \sum_{j=1}^{H} v_{kj} y_j^p + \theta_k$$

Imagen 3 [4]

Por último, el valor de salida de la neurona de salida k, y $^{p}_{k}$, es:

$$y_k^p = f\left(net_k^p\right)$$

Imagen 4[4]

Etapa de aprendizaje
En la etapa de aprendizaje, el objetivo es hacer mínimo el error entre la salida obtenida por la red y la salida deseada por el usuario ante la presentación de un conjunto de patrones denominado grupo de entrenamiento.[4]

Así el aprendizaje en las redes backpropagation es de tipo supervisado. La función de error que se pretende minimizar para cada patrón p viene dada por:

$$E^p = \frac{1}{2} \sum_{k=1}^{M} (d_k^p - y_k^p)^2$$

Imagen 5[4]

Donde d$^{p}_{k}$ es la salida deseada para la neurona de salida k ante la presentación del patrón p. A partir de esta expresión se puede obtener una medida general de error mediante:

$$E = \sum_{p=1}^{P} E^p$$

La base del algoritmo backpropagation para la modificación de los pesos es la técnica conocida como gradiente decreciente. Como E_p es función de todos los pesos de la red, el gradiente de E_p es un vector igual a la derivada parcial de E_p respecto a cada uno de los pesos. El gradiente toma la dirección que determina el incremento más rápido en el error, mientras que la dirección opuesta, es decir, la dirección negativa, determina el decremento más rápido en el error. Por tanto, el error puede reducirse ajustando cada peso en la dirección:

$$-\sum_{p=1}^{P} \frac{\partial E^p}{\partial w_{ji}}$$

Un peligro que puede surgir al utilizar el método de gradiente decreciente es que el aprendizaje converja a un mínimo local. Sin embargo, el problema potencial de los mínimos locales se da en raras ocasiones en datos reales. A nivel práctico, la forma de modificar los pesos de forma iterativa consiste en aplicar la regla de la cadena a la expresión del gradiente y añadir una tasa de aprendizaje η. Así, en una neurona de salida:

$$\Delta v_{kj}(n+1) = -\eta \frac{\partial E^p}{\partial v_{kj}} = \eta \sum_{p=1}^{P} \delta_k^p y_j^p$$

donde

$$\delta_k^p = \left(d_k^p - y_k^p\right) f'\left(net_k^p\right)$$

y n indica la iteración. En una neurona oculta:

$$\Delta w_{ji}(n+1) = \eta \sum_{p=1}^{P} \delta_j^p x_i^p$$

donde

$$\delta_j^p = f\left(net_j^p\right) \sum_{k=1}^{M} \delta_k^p v_{kj}$$

Imagen 6[4]

Se puede observar que el error o valor delta asociado a una neurona oculta j, viene determinado por la suma de los errores que se cometen en las k neuronas de salida que reciben como entrada la salida de esa neurona oculta j. De ahí que el algoritmo también se denomine propagación del error hacia atrás. Para la modificación de los pesos, la actualización se realiza después de haber presentado todos los patrones de entrenamiento. Este es el modo habitual de proceder y se denomina aprendizaje por lotes o modo batch. Otra modalidad denominada aprendizaje en serie o modo on line consistente en actualizar los pesos tras la presentación de cada patrón de entrenamiento. Ha de hacerse en orden aleatorio. Para acelerar el proceso de convergencia de los pesos, Rumelhart et al. (1986) sugirieron añadir un factor momento, α, que tiene en cuenta la dirección del incremento tomada en la iteración anterior:

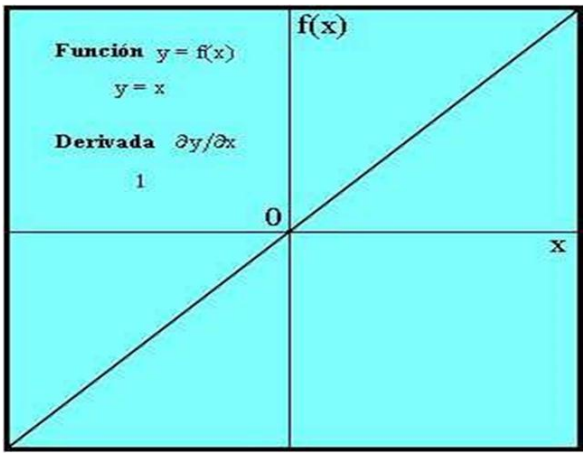
$$\Delta v_{kj}\left(n+1\right) = \eta\left(\sum_{p=1}^{P} \delta_k^p y_j^p\right) + \alpha \Delta v_{kj}\left(n\right)$$

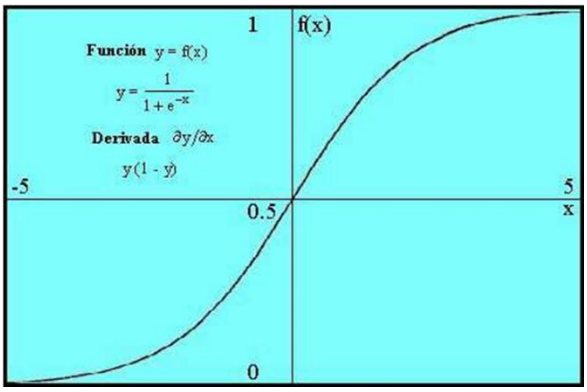
Imagen 6[4]

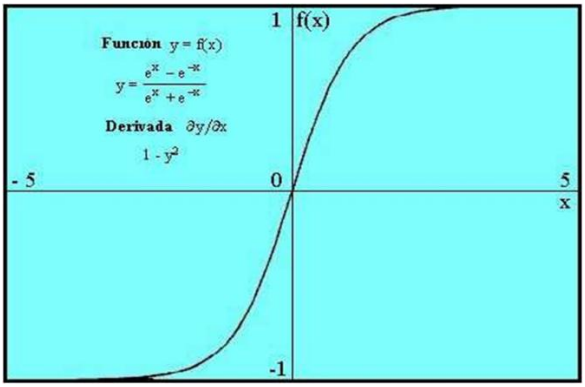
Fases en la aplicación de un perceptrón multicapa
Una red del tipo perceptrón multicapa intenta resolver dos tipos de problemas: — Problemas de predicción, que consisten en la estimación de una variable continúa de salida, a partir de la presentación de un conjunto de variables predictoras de entrada (discretas y/o continuas). — Problemas de clasificación, que consisten en la asignación de la categoría de pertenencia de un determinado patrón a partir de un conjunto de variables predictoras de entrada (discretas y/o continuas). [4]

Selección de las variables relevantes y preprocesamiento de los datos
Para obtener una aproximación funcional óptima, se deben elegir cuidadosamente las variables a emplear: se trata es de incluir en el modelo las variables predictoras que realmente predigan la variable dependiente o de salida, pero que a su vez no tengan relaciones entre sí ya que esto puede provocar un sobreajuste innecesario en el modelo. Las variables deben seguir una distribución normal o uniforme, y el rango de posibles valores debe ser aproximadamente el mismo y acotado dentro del intervalo de trabajo de la función de activación empleada en las capas ocultas y de salida de la red neuronal. Así, las variables de entrada y salida suelen acotarse en valores comprendidos entre 0 y 1 ó entre −1 y 1. Si la variable es discreta, se utiliza la codificación dummy. Por ejemplo, la variable sexo podría codificarse como: 0 = hombre, 1 = mujer; estando representada por una única neurona. La variable nivel social podría codificarse como: 100 = bajo, 010 = medio, 001 = alto; estando representada por tres neuronas. Por su parte, si la variable es de naturaleza continua, ésta se representa mediante una sola neurona, como, por ejemplo, el CI de un sujeto. [4]

Entrenamiento de la red neuronal
Elección de los pesos iniciales Se hace una asignación de pesos pequeños generados de forma aleatoria, en un rango de valores entre −0,5 y 0,5 o algo similar. Arquitectura de la red Respecto a la arquitectura de la red, se sabe que para la mayoría de problemas prácticos bastará con utilizar una sola capa oculta. El número de neuronas de la capa de entrada está determinado por el número de variables predictoras. Así, en los ejemplos anteriores, la variable sexo estaría representada por una neurona que recibiría los valores 0 ó 1. La variable status social estaría representada por tres neuronas. La variable puntuación en CI estaría representada por una neurona que recibiría la puntuación previamente acotada, por ejemplo, a valores entre 0 y 1. El número de neuronas de la capa de salida está determinado bajo el mismo esquema que en el caso anterior. Cuando intentamos discriminar entre dos categorías, bastará con utilizar una única neurona (por ejemplo, salida 1 para la categoría A, salida 0 para la categoría B). Si estamos ante un problema de estimación, tendremos una única neurona que dará como salida el valor de la variable a estimar. El número de neuronas ocultas determina la capacidad de aprendizaje de la red neuronal. Recordando el problema del sobreajuste, se debe usar el mínimo número de neuronas ocultas con las cuales la red rinda de forma adecuada. Esto se consigue evaluando el rendimiento de diferentes arquitecturas en función de los resultados obtenidos con el grupo de validación. Tasa de aprendizaje y factor momento El valor de la tasa de aprendizaje (η) controla el tamaño del cambio de los pesos en cada iteración. Se deben evitar dos extremos: un ritmo de aprendizaje demasiado pequeño puede ocasionar una disminución importante en la velocidad de convergencia y la posibilidad de acabar atrapado en un mínimo local; en cambio, un ritmo de aprendizaje demasiado grande puede conducir a inestabilidades en la función de error, lo cual evitará que se produzca la convergencia debido a que se darán saltos en torno al mínimo sin alcanzarlo. Por tanto, se recomienda elegir un ritmo de aprendizaje lo más grande posible sin que provoque grandes oscilaciones. En general, el valor de la tasa de aprendizaje suele estar comprendida entre 0.05 y 0.5. El factor momento (α) acelera la convergencia de los pesos. Suele tomar un valor próximo a 1 (por ejemplo, 0.9). Función de activación de las neuronas ocultas y de salida Hay dos formas básicas que cumplen esta condición: la función lineal (o identidad) y la función sigmoidal (logística o tangente hiperbólica).[4]







Para aprovechar la capacidad de las NN de aprender relaciones complejas o no lineales entre variables, se recomienda la utilización de funciones no lineales al menos en las neuronas de la capa oculta. Por tanto, en general se utilizará una función sigmoidal (logística o tangente hiperbólica) como función de activación en las neuronas de la capa oculta.
La elección de la función de activación en las neuronas de la capa de salida dependerá del tipo de tarea impuesto.
En tareas de clasificación, las neuronas normalmente toman la función de activación sigmoidal. En cambio, en tareas de predicción o aproximación de una función, general- mente las neuronas toman la función de activación lineal.

Evaluación del rendimiento del modelo

Una vez seleccionado el modelo de red cuya configuración de parámetros ha obtenido la mejor ejecución ante el conjunto de validación, debemos evaluar la capacidad de generalización de la red de una forma completamente objetiva a partir de un tercer grupo de datos independiente, el conjunto de *test*.

Cuando se trata de la estimar una función, normalmente se utiliza la media cuadrática del error para evaluar la ejecución del modelo:

$$MC_{error} = \frac{\sum_{p=1}^{P}\sum_{k=1}^{M}(d_k^p - y_k^p)^2}{P \cdot M}$$

En problemas de clasificación de patrones es mejor la frecuencia de clasificaciones correctas e incorrectas. Se puede construir una tabla de confusión y calcular diferentes índices de asociación y acuerdo entre el criterio y la decisión tomada por la red neuronal. Interpretación de los pesos obtenidos Se trata de interpretar los pesos de la red neuronal. El método más popular es el análisis de sensibilidad. El análisis de sensibilidad está basado en la medición del efecto que se observa en una salida y_k debido al cambio que se produce en una entrada x_i . Cuanto mayor efecto se observe sobre la salida, mayor sensibilidad se puede deducir que presenta respecto a la entrada. Un método común consiste en fijar el valor de todas las variables de entrada a su valor medio e ir variando el valor de una de ellas a lo largo de todo su rango, registrando el valor de salida de la red

III. CONCLUSIONES:

•La realización del algoritmo backpropagation surgió para evitar en gran medida los errores obtenidos por los cálculos que se realizan en el entrenamiento de la red neuronal
•Se aplica matemática avanzada donde se busca darle mayor eficiencia a la red neuronal, debido a que la escalabilidad de una red puede generar una variación de los datos que deben ser corregidos mediante cálculo diferencial.

IV. REFERENCIAS

1. HTTPS://BOOKS.GOOGLE.COM.CO/BOOKS?ID=U4PWETEPMPQC&PG=PA418&DQ=RESUMEN+RED+NEURONAL+BACKPROPAGATION&HL=ES-419&SA=X&VED=0AHUKEWIN7TUN8P3KAHUFNKWKHVG-DAGQ6AEIKDAA#V=ONEPAGE&Q=RESUMEN%20RED%20NEURONAL%20BACKPROPAGATION&F=FALSEHTTPS://PLATZI.COM/TUTORIAL

ES/1157-IA-2017/2619-ENTRENAMIENTO-DEL-PERCEPTRON/

2. HTTPS://ES.WIKIPEDIA.ORG/WIKI/PROPAGACI%C3%B3N_HACIA_ATR%C3%A1S

3. HTTP://WWW.DIEGOCALVO.ES/BACKPROPAGATION-REDES-NEURONALES/

4. HTTP://HALWEB.UC3M.ES/ESP/PERSONAL/PERSONAS/JMMARIN/ESP/DM/TEMA3DM.PDF