ENTRENAMIENTO DE RED NEURONAL ADALINE POR MEDIO DEL MÉTODO DEL GRADIENTE DESCENDENTE

Camilo A. Hurtado E.
Maestría en Matemáticas Aplicadas
Universidad Sergio Arboleda
Bogotá, Colombia
camilo.hurtado01@correo.usa.edu.co

John Alexander Tami Buitrago
Maestría en Matemáticas Aplicadas
Universidad Sergio Arboleda
Bogotá, Colombia
john.tami01@correo.usa.edu.co

April 21, 2020

ABSTRACT

El problema de aprendizaje o entrenamiento de una red neuronal se establece a partir de la minimización de una función de error asociada. En este trabajo se estudia en específico la RNA tipo Adaline, y se analiza el algoritmo de aprendizaje Widrow-Hoff y Widrow-Hoff iterativo, pasando por un estudio teórico y práctico, incluyendo ejemplos y pseudocódigo.

***K*eywords** Red neuronal Adaline · gradiente descendente · Optimización

1 Introducción

Las RNAs -Redes Neuronales Artificiales- son artefactos matemáticos y de programación que sirven para clasificar, reconocer o controlar señales, que pueden ser datos, imágenes, sonidos. Su estructura y funcionamiento está basado en la estructura y funcionamiento de las redes neuronales biológicas, principalmente del ser humano.

El funcionamiento de las redes neuronales está basado en dos parámetros, que son los pesos sinápticos y la polarización, y en una función de activación. Los pesos sinápticos y la polarización simulan los neurotransmisores de la neuronal real, y son los que afectan directamente la señal de entrada a la neurona. La función de activación es la expresión matemática que define la salida de la neurona.

Las redes neuronales deben *entrenarse* -ajustar sus pesos sinápticos y polarización- para que ejecuten con precisión la tarea que deseemos. Entre los muchos métodos que existen para entrenar tales redes, se encuentra uno muy conocido, denominado *descenso por gradiente*, el cual será explicado en este informe.

A continuación, el trabajo contiene en el capítulo 2 los objetivos que se buscan con su desarrollo, en el capítulo 3 la manera cómo se investigó y desarrolló este tema, en el capítulo 4 las bases teóricas sobre una red neuronal y el método de gradiente descendente, en el capítulo 5 la aplicación práctica de este método, en el capítulo 6 las conclusiones y finalmente en la última sección las referencias bibliográficas de donde se obtuvo esta información.

2 Objetivos

2.1 Objetivo General

Entrenar una RNA monocapa tipo Adaline por medio del método del gradiente descendente.

2.2 Objetivos específicos

- Aprender y recordar conceptos sobre RNAs en general.
- Aprender las características de una RNA Adaline

- Aprender el método de Widrow-Hoff estándar e iterativo
- Aplicar el método de Widrow-Hoff iterativo, basado en el método matemático de gradiente descendente, al entrenamiento de RNA
- Explicar el pseudocódigo para aplicar el método de Widrow-Hoff iterativo al entrenamiento de una RNA

3 Metodología

Para desarrollar este trabajo, se realizó una búsqueda de referencias bibliográficas relacionadas con las RNAs y sus métodos de entrenamiento. Tal búsqueda incluyó artículos científicos, sitios web especializados y videos.

El proceso incluyó un análisis detallado de la estructura de una RNA y del método de gradiente descendente, así como de la explicación matemática para poder aplicar este último al entrenamiento de una RNA.

Finalmente, se muestra un pseudocódigo que explica la aplicación del método a cualquier lenguaje de programación.

4 Marco teórico

En esta sección se van a explicar los conceptos teóricos básicos y necesarios para entender el funcionamiento de una RNA y su proceso de entrenamiento por medio del método de gradiente descendente. En la subsección 4.1 se explican las bases matemáticas del método del gradiente descendente, en la subsección 4.2 se explican las características y bases matemáticas de la RNA tipo Adaline y en la subsección 4.3 se explica cómo se entrena una RNA Adaline a partir del método de gradiente descendente.

Cuando se está diseñando una RNA, se deben considerar los siguientes aspectos [1]:

- Parámetros: ¿Qué tipo o estructura de red es la más adecuada?
- Optimización: ¿Qué algoritmo de aprendizaje -entrenamiento- usamos para obtener los pesos?
- Generalización: ¿Cómo logramos que la RNA entregue resultados adecuados usando datos diferentes a los usados para su entrenamiento?
- Invarianza: ¿Cómo conseguimos que la RNA soporte cambios comunes en los datos?

Son estos aspectos los que deben evaluarse al resolver un problema práctico, y algunos de los cuales se consideran a lo largo del trabajo.

4.1 Método de optimización de gradiente descendente

Se trata de algoritmo de aprendizaje en muchas técnicas de machine learning [2]. En muchos casos entrega mejores resultados que el método de los mínimos cuadrados.

Consiste en un método iterativo de optimización numérica que permite encontrar valores mínimos de funciones convexas y diferenciables en todo su dominio. [3]. La función de coste más usada para este algoritmo es el error cuadrático medio, aunque se pueden usar otras como la entropía cruzada o diseñar alguna otra personalizada. El valor de la función de error depende de los parámetros de la RNA: sus pesos sinápticos y sus bias -o polarizaciones.

Para hallar el valor mínimo de una función convexa a partir de un punto p de su dominio, se toman pasos proporcionales al sentido opuesto del gradiente de la función en el punto p. Si se toman los pasos en el mismo sentido del gradiente, se halla el máximo de la función.

Hay muchas versiones de este algoritmo, con mejoras realizadas en muchos de sus aspectos, como en la elección del ratio de aprendizaje, y reciben nombres como *gradiente descendente estocástico*, *momentum*, *AdaGrad*, *RMSProp*, *Adam*, entre otros [2]

4.2 Red neuronal Adaline

Su nombre significa **Ada**ptive **Li**near **Ne**twork. Para su entrenamiento, suele usarse el algoritmo de Widrow-Hoff, cuyo objetivo es encontrar la mejor separación posible entre los grupos, hallando la mejor frontera de decisión [4].

La principal característica de la red Adaline, y que la diferencia de otras configuraciones de RNAs, es que su función de activación es *lineal* y no *escalón* como en el perceptrón [5].

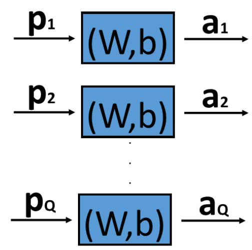

Figure 1: Red neuronal Adaline

La situación consiste en que se tienen ejemplos con los cuales se desea entrenar la red monocapa tipo Adaline (se trata por tanto de entrenamiento supervisado). Entrenar significa hallar los pesos sinápticos y la polarización de las neuronas para que se comporten como deseamos.

El método de Widrow-Hoff, a diferencia del algoritmo de aprendizaje del perceptrón, trata de hallar la mejor frontera de decisión que separa las clases presentes en los datos de entrada.

Para deducir el método matemáticamente, se supone la existencia de una única neurona en la RNA, a la cual van entrando todos los patrones, uno a uno, desde el p_1 hasta el p_Q, como se observa en la figura 1. Lo que se desea es minimizar el error e, definido como la respuesta que deseamos obtener, t, menos la salida entregada por la neurona a, es decir $e_i = t_i - a_i$, donde i representa un patrón específico.

Se procede a minimizar el error cuadrático, e^2 para eliminar posibles errores negativos.

Widrow y Hoff propusieron que una manera de minimizar los errores cuadráticos para cada uno de los ejemplos era minimizando el error cuadrático medio, es decir:

$$E[e^2] = \frac{1}{Q}\sum_{q=1}^{Q} e_q^2 \tag{1}$$

El cual en este caso no es más que el promedio de la suma de los errores cuadráticos de todos las salidas entregadas por la neurona con respecto a las salidas deseadas, cuando tienen como entrada los patrones p_i

Por tanto, el objetivo del método Widrow-Hoff es *hallar los valores de **(w,b)** que minimizan $E[e^2]$*

Se procede como sigue:

$$E[e^2] = E[(t-a)^2] \tag{2}$$
$$E[e^2] = E[(t-\mathbf{x^T z})^2] \tag{3}$$

Donde

$$a = [w_1 \ldots w_R \; b]\begin{bmatrix} p_1 \\ . \\ . \\ . \\ p_R \\ 1 \end{bmatrix} = \mathbf{x^T z} \tag{4}$$

Luego:

$$E[e^2] = E[t^2 - 2t\mathbf{x^T z} + (\mathbf{x^T z})(\mathbf{z^T x})] \tag{5}$$
$$E[e^2] = E[t^2] - 2E[t\mathbf{z^T}]\mathbf{x} + \mathbf{x^T}E[\mathbf{zz^T}]\mathbf{x} \tag{6}$$
$$E[e(\mathbf{x})^2] = c - 2\mathbf{h^T x} + \mathbf{x^T R x} \tag{7}$$

Donde:

$$c := E[t^2] \tag{8}$$
$$\mathbf{h} := E[t\mathbf{z}] \tag{9}$$
$$\mathbf{R} := E[\mathbf{zz^T}] \tag{10}$$
$$\tag{11}$$

Esto significa que c es un escalar dado por la correlación entre las respuestas deseadas t de la neurona; h es un vector de correlación entre las respuestas deseadas t y los patrones de entrada z; R es una matriz de autocorrelación entre los patrones de entrada. R es simétrica, tiene valores propios reales y puede ser definida positiva -valores propios positivos- o semidefinida positiva -valores propios positivos o iguales a cero.

Ahora ya se puede hallar el gradiente de 7 con respecto a x:

$$\frac{dE[e^2]}{d\mathbf{x}}|_{x=x_m} = -2\mathbf{h} + 2\mathbf{R x_m} = \mathbf{0} \tag{12}$$
$$\mathbf{x_m} = \mathbf{R^{-1} h} \tag{13}$$

De acuerdo a lo anterior, 13 corresponde a la ecuación que sirve para entrenar una sola neurona.

Dependiendo de las características de R, se pueden encontrar los siguientes casos:

- Si R es invertible -es decir, es definida positiva-, entonces hay un único mínimo
- Si R **NO** es invertible -es decir, es semidefinida positiva-, entonces:
  - Pueden existir muchos mínimos, los cuales se hallan a través de la pseudoinversa de R_+
  - No existe mínimo

4.2.1 Aplicación práctica del método Widrow-Hoff para una única neurona

El ejemplo consiste en diseñar una RNA Adaline que realice la función AND, cuya tabla de verdad está en términos de 1 y -1 en la tabla 1:

p_1	p_2	AND	AND
0	0	0	-1
0	1	0	-1
1	0	0	-1
1	1	1	1

Table 1: Tabla de verdad de la función AND.

En primer lugar, se calcula la matriz R y luego el vector h:

$$\mathbf{R} := E[\mathbf{zz^T}] = \frac{1}{Q}\sum_{q=1}^{Q}\mathbf{z_q z_q^T} = \frac{1}{4}\left(\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}^T + \begin{bmatrix}0\\1\\1\end{bmatrix}\begin{bmatrix}0\\1\\1\end{bmatrix}^T + \begin{bmatrix}1\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\1\end{bmatrix}^T + \begin{bmatrix}1\\1\\1\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix}^T\right) \tag{14}$$

$$\mathbf{R} = \frac{1}{4}\begin{bmatrix}2 & 1 & 2\\1 & 2 & 2\\2 & 2 & 4\end{bmatrix} \tag{15}$$

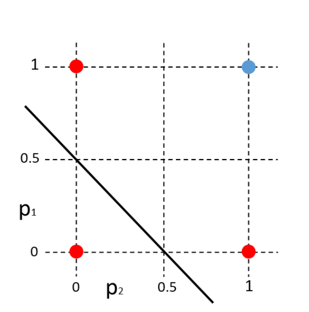

Figure 2: Espacio de entrada con patrones de entrenamiento y frontera de decisión

En donde $\mathbf{z}$ corresponde a los parámetros p_1 y p_2 con un 1 aumentado al final correspondiente al valor de la polarización, en donde:

$$\mathbf{z} = \begin{bmatrix} p_1 \\ . \\ . \\ . \\ p_R \\ 1 \end{bmatrix} \tag{16}$$

El vector h se calcula de la siguiente forma:

$$\mathbf{h_1} := E[t\mathbf{z}] = \frac{1}{Q}\sum_{q=1}^{Q} t_q \mathbf{z}_q = \frac{1}{4}\left(0\begin{bmatrix}0\\0\\1\end{bmatrix} + 0\begin{bmatrix}0\\1\\1\end{bmatrix} + 0\begin{bmatrix}1\\0\\1\end{bmatrix} + 1\begin{bmatrix}1\\1\\1\end{bmatrix}\right) = \frac{1}{4}\begin{bmatrix}1\\1\\1\end{bmatrix} \tag{17}$$

Pero este vector h también puede calcularse tomando como salida de la operación AND los valores -1 y 1:

$$\mathbf{h_2} = \frac{1}{4}\left(-1\begin{bmatrix}0\\0\\1\end{bmatrix} - 1\begin{bmatrix}0\\1\\1\end{bmatrix} - 1\begin{bmatrix}1\\0\\1\end{bmatrix} + 1\begin{bmatrix}1\\1\\1\end{bmatrix}\right) = \frac{1}{4}\begin{bmatrix}0\\0\\2\end{bmatrix} \tag{18}$$

Aplicando la ecuación 13, se tiene:

$$\mathbf{x_1} = \mathbf{R}^{-1}\mathbf{h_1} = \begin{bmatrix}0.5\\0.5\\-0.25\end{bmatrix} \tag{19}$$

Sin embargo, estos pesos sinápticos y polarización no clasifican bien los patrones, como se observa en la figura 2

Ahora, se aplica la ecuación 13 pero usando el vector h_2

$$\mathbf{x_1} = \mathbf{R}^{-1}\mathbf{h_2} = \begin{bmatrix}1\\1\\-1.5\end{bmatrix} \tag{20}$$

Y la frontera de decisión que se obtiene se observa en la figura 3

Esta frontera de decisión sí clasifica correctamente los patrones. Por tanto, cuando se usa el algoritmo de Widrow-Hoff se deben codificar las salidas de las neuronas con -1 y 1 y no con 0 y 1

4.2.2 Método Widrow-Hoff para una red monocapa

Se considera una red monocapa de neuronas tipo Adaline, como se observa en la figura 4, con S neuronas.

Para cada neurona se puede escribir la ecuación correspondiente al error cuadrático medio:

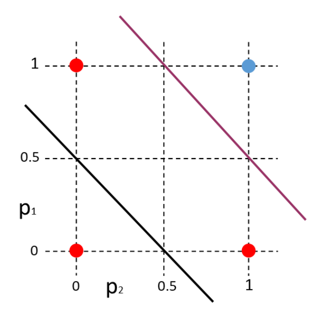

Figure 3: Espacio de entrada con patrones de entrenamiento y dos fronteras de decisión

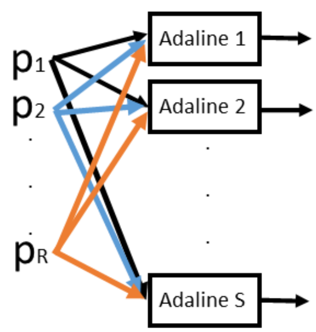

Figure 4: Red neuronal Adaline en versión explícita

$$E[e_1^2] = c_1 - 2\mathbf{h_1^T x_1} + \mathbf{x_1^T R x_1} \tag{21}$$
$$E[e_2^2] = c_2 - 2\mathbf{h_2^T x_2} + \mathbf{x_2^T R x_2} \tag{22}$$
$$...E[e_S^2] = c_S - 2\mathbf{h_S^T x_S} + \mathbf{x_S^T R x_S} \tag{23}$$

y se obtiene la ecuación que minimiza el vector $\mathbf{x}$ para cada una de las ecuaciones anteriores:

$$\mathbf{x_{m1}} = \mathbf{R}^{-1}\mathbf{h_1} \tag{24}$$
$$\mathbf{x_{m2}} = \mathbf{R}^{-1}\mathbf{h_2} \tag{25}$$
$$...\mathbf{x_{mS}} = \mathbf{R}^{-1}\mathbf{h_S} \tag{26}$$

Estas ecuaciones para $\mathbf{x_m}$ se compactan de la siguiente manera:

$$\mathbf{X_m} = \mathbf{R}^{-1}\mathbf{H} \tag{27}$$

Donde:

$$\mathbf{X_m} = [\mathbf{x_{m1}} \dots \mathbf{x_{ms}}] \tag{28}$$

$$\mathbf{H} = [\mathbf{h_1} \dots \mathbf{h_S}] \tag{29}$$

$$\mathbf{R} = \frac{1}{Q}\mathbf{Z}\mathbf{Z^T} \tag{30}$$

$$\mathbf{Z} = [\mathbf{z_1} \dots \mathbf{z}_Q] \tag{31}$$

$$\mathbf{H} = \frac{1}{Q}\mathbf{Z}\mathbf{T^T} \tag{32}$$

$$\mathbf{T} = [\mathbf{t}_1 \dots \mathbf{t_Q}] \tag{33}$$

$$\mathbf{x_i} = \begin{bmatrix} W_{i,1} \\ . \\ . \\ . \\ W_{i,R} \\ b_i \end{bmatrix} \tag{34}$$

$$\mathbf{z_q} = \begin{bmatrix} p_{1,q} \\ . \\ . \\ . \\ p_{R,q} \\ 1 \end{bmatrix} \tag{35}$$

4.3 Aplicación del método de gradiente descendente al entrenamiento de una red neuronal Adaline

La siguiente información fue tomada de [6].

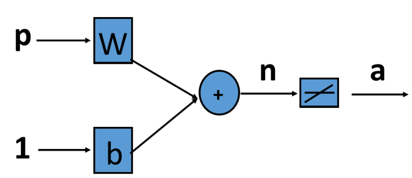

Figure 5: Red neuronal Adaline en versión compacta

El problema consiste en que, a partir de un conjunto de patrones de prueba que sirven como ejemplo, tales como $(p_1, t_1), ..., (p_q, t_q), ..., (p_Q, t_Q)$, queremos hallar los pesos sinápticos W y la polarización b de cada una de las neuronas de la red Adaline, como la de la figura 5 para hacer que responda como nosotros deseamos.

El método de Widrow-Hoff nos permite calcular los pesos sinápticos y la polarización de cada neurona a partir de tales patrones de ejemplo, minimizando el error cuadrático de todo el conjunto de patrones.

Este método tiene una desventaja, y es que requiere conocer desde un principio todos los ejemplos para poder computar los pesos sinápticos y la polarización de las neuronas. Esto significa que si en algún momento llega un nuevo ejemplo, es necesario recomputar todo nuevamente.

Un método que soluciona este inconveniente es una variación del método Widrow-Hoff, y se conoce como método Widrow-Hoff iterativo, el cual se obtiene a partir del método del gradiente descendente. Esta variación permite modificar los pesos sinápticos y la polarización conforme vayan llegando nuevos ejemplos

El método de Widrow-Hoff minimiza el MSE -Mean Squared Error, Error Cuadrático Medio -, de la red con respecto a todos los patrones de ejemplo. La variación del método a partir del gradiente descendente minimiza el MSE de la red pero de cada uno de los patrones por separado.

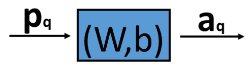

Figure 6: Patrón q entrando a la red neuronal

Esto significa que se analizará por ejemplo un patrón p_q que entra a la red Adaline, como se observa en la figura 6:

La red arroja una salida a_q, con la cual se puede calcular un vector de error $e_q = t_q - a_q$, que es la diferencia entre la respuesta deseada t_q y la respuesta de la red neuronal para el patrón p_q. El error cuadrático se define como $e_q^T e_q$, el cual será la función objetivo a minimizar. Este producto punto arroja la suma de los errores cuadráticos de cada neurona con respecto al patrón p_q

Lo que nos interesa es **minimizar el error cuadrático para el ejemplo q**. Precisamente en este punto es donde está la diferencia con el método de Widrow-Hoff original, pues en él se minimiza el MSE de **todos los patrones de ejemplo**

Por tanto, la función objetivo a minimizar es:

$$F(\mathbf{X}) = \mathbf{e}_q^T \mathbf{e}_q \tag{36}$$

$$\mathbf{X} = \begin{pmatrix} \mathbf{W}^T \\ \mathbf{b}^T \end{pmatrix} \tag{37}$$

En donde los errores de cada neurona están en función de los pesos sinápticos y la polarización de esa neurona, que están representados por **X**. La ecuación (36) es la suma de los errores cuadráticos de cada neurona con respecto a cada patrón de ejemplo (en este caso, el patrón p_q)

Usando el método de descenso por gradiente, podemos reescribir (36) como sigue:

$$F([\mathbf{x_1} \ldots \mathbf{x_S}]) = \left[\sum_{i=1}^{S} e_i^2(\mathbf{x}_i)\right]_q \tag{38}$$

$$e_i(\mathbf{x}_i) = t_i - \mathbf{x}_i^T \mathbf{z} \tag{39}$$

En donde la matriz **X** se escribe en forma de vectores columna **x**, cada uno de los cuales contiene los pesos sinápticos y la polarización de cada neurona, desde la 1 hasta la S. En la ecuación (39), el vector $\mathbf{z}$ contiene los patrones de entrada aumentado con un 1 al final.

Para usar el método del gradiente descendente, debemos calcular el gradiente de F con respecto a la matriz **X**:

$$\frac{\mathrm{d}F}{\mathrm{d}\mathbf{X}} = \left[\frac{\mathrm{d}F}{\mathrm{d}\mathbf{x}_1} \ldots \frac{\mathrm{d}F}{\mathrm{d}\mathbf{x}_S}\right] \tag{40}$$

Este gradiente, calculado en la ecuación (40), es una matriz que contiene los gradientes respecto a $x_1, x_2, \ldots x_S$, donde cada x_i contiene los pesos sinápticos y la polarización de cada neurona.

$$\frac{\mathrm{d}F}{\mathrm{d}\mathbf{x_i}} = \frac{\mathrm{d}F}{\mathrm{d}e_i}\frac{\mathrm{d}e_i}{\mathrm{d}\mathbf{x}_i} \ = \ 2e_i\frac{\mathrm{d}e_i}{\mathrm{d}\mathbf{x}_i} = -2e_i\mathbf{z} \tag{41}$$

Reemplazando (41) en (40) obtenemos

$$\frac{\mathrm{d}F}{\mathrm{d}\mathbf{X}} = [-2e_i\mathbf{z} \ldots \ - 2e_s\mathbf{z}]_q = -2\mathbf{z}_q\mathbf{e}_q^T \tag{42}$$

La ecuación (42) representa el gradiente para cada neurona de acuerdo al patrón p_q

Usando la fórmula para el descenso por gradiente, obtenemos la siguiente ecuación:

$$\mathbf{X} = \mathbf{X} + 2\alpha \mathbf{z}_q \mathbf{e}_q \tag{43}$$

La ecuación (43) indica que en cada iteración se va a realizar una mejor apuesta acerca de dónde se encuentra el mínimo de la función. El factor α se conoce como el índice de aprendizaje.

La ecuación (43) se puede descomponer en términos de los pesos sinápticos y la polarización. Así que se reemplaza la ecuación (37) en la ecuación (43), y obtenemos:

$$\begin{pmatrix} \mathbf{W}^T \\ \mathbf{b}^T \end{pmatrix} = \begin{pmatrix} \mathbf{W}^T \\ \mathbf{b}^T \end{pmatrix} + 2\alpha \begin{pmatrix} \mathbf{p}^q \\ 1 \end{pmatrix} \mathbf{e}_q^T \tag{44}$$

Finalmente, la ecuación (44) se descompone en dos ecuaciones y se obtiene la transpuesta de cada una de ellas:

$$\mathbf{W} = \mathbf{W} + 2\alpha \mathbf{e}_q \mathbf{p_q^T} \tag{45}$$

$$\mathbf{b} = \mathbf{b} + 2\alpha \mathbf{e}_q \tag{46}$$

Las ecuaciones (10) y (46) son las que se usarán para actualizar los pesos sinápticos y las polarizaciones en cada iteración del aprendizaje de Widrow-Hoff

5 Aplicación práctica del método

El pseudocódigo del método Widrow-Hoff iterativo es el siguiente:

Inicialización aleatoria de $\mathbf{W}$ y $\mathbf{b}$

Inicialización del número total de neuronas, $Neuronas$
Inicialización del número total de patornes, $Patrones$
for i=1:$Neuronas$ **do**
  for j=1:$Patrones$ **do**
    $\mathbf{a}_q = \mathbf{W}\mathbf{p}_q + \mathbf{b}$
    $\mathbf{e}_q = \mathbf{t}_q - \mathbf{a}_q$
    $\mathbf{W} = \mathbf{W} + 2\alpha \mathbf{e}_q \mathbf{p}_q^T$
    $b = b + 2\alpha \mathbf{e}_q$
  end for
end for

6 Comentarios finales

El método iterativo de Widrow-Hoff basado en el método matemático del gradiente descendiente permite hallar de manera más eficiente los perámetros de la RNA, y además permite que dichos parámetros se vayan actualizando a medida que se tengan más parámetros de ejemplo.

References

[1] Fernando Berzal. Entrenamiento de redes neuronales. page 21.

[2] Jose Martinez Heras. Gradiente Descendiente para aprendizaje automático, March 2019.

[3] Alejandro Sánchez Yalí. El algoritmo del gradiente descendente.

[4] Redes Neuronales - 4.1 Introducción a la Red Adaline - Hackeando Tec - YouTube.

[5] Redes Neuronales - 4.2 Aprendizaje Widrow-Hoff para una sola neurona - Hackeando Tec - YouTube.

[6] Hackeando Tec. Redes Neuronales - 4.6 Aprendizaje Widrow-Hoff Iterativo - Hackeando Tec.