Entrenamiento Perceptrón

Perceptron Training.

Autor 1: Kevin Leandro Bernal León

*Facultad de Ingenierías. IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e: kevinleon@utp.edu.co

***Resumen*— Como sabemos, un algoritmo de aprendizaje es un método adaptativo por el que una red de PE se auto modifica para implementar el comportamiento deseado. Esto se hace presentando algunos ejemplos de la función entrada-salida a la red. Se presenta un ejemplo y se ejecuta una acción correctiva iterativamente hasta que la red aprende a producir la respuesta deseada.**

***Palabras clave—* Aprendizaje, entrenamiento, inferencia, inteligencia artificial, neurona, patrones, perceptrón, pesos, red neuronal, umbral.**

***Abstract*— As we know, a learning algorithm is an adaptive method by which a PE network is automatically modified to implement the desired behavior. This is done by presenting some examples of the red input-output function. An example is presented and a corrective action is executed iteratively until the red one learns to produce the desired response.**

***Key Word* —Learning, training, inference, artificial intelligence, neuron, patterns, perceptron, weights, red neuronal, threshold.**

1. INTRODUCCIÓN

La primera red neuronal conocida, fue desarrollada en 1943 por WarrenMcCulloch y Walter Pitts; esta consistía en una suma de las señales de entrada, multiplicadas por unos valores de pesos escogidos aleatoriamente. La entrada es comparada con un patrón preestablecido para determinar la salida de la red. Si en la comparación, la suma de las entradas multiplicadas por los pesos es mayor o igual que el patrón preestablecido la salida de la red es uno (1), en caso contrario la salida es cero (0). Al inicio del desarrollo de los sistemas de inteligencia artificial, se encontró gran similitud entre su comportamiento y el de los sistemas biológicos y en principio se creyó que este modelo podía computar cualquier función aritmética o lógica.

El Perceptrón era inicialmente un dispositivo de aprendizaje, en su configuración inicial no estaba en capacidad de distinguir patrones de entrada muy complejos, sin embargo, mediante un proceso de aprendizaje era capaz de adquirir esta capacidad. En esencia, el entrenamiento implicaba un proceso de refuerzo mediante el cual la salida de las unidades A se incrementaba o se decrementaba dependiendo de si las unidades A contribuían o no a las respuestas correctas del Perceptrón para una entrada dada.

1. CONTENIDO

El aprendizaje es el proceso por medio del cual los parámetros libres de una red neuronal son adaptados a través de un proceso de estimulación por el ambiente en el cual la red se encuentra inmersa. El tipo de aprendizaje es determinado por la manera en el cual el cambio de parámetros tiene lugar. De manera que en el proceso de aprendizaje el perceptrón modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen en el proceso de aprendizaje son la destrucción, modificación y creación de conexiones entre las neuronas.

Hay dos vías:

NO ADAPTATIVOS: se determina de antemano cuál será el valor de los pesos.

ADAPTATIVOS: no existe una forma para determinar de antemano los pesos, por los que se necesita un proceso iterativo.

Wi(t) = Wi(t-1) + Δ Wi

TIPOS DE APRENDIZAJE:

* Aprendizaje supervisado.
* Aprendizaje no supervisado.
* Aprendizaje por reforzamiento.

Reglas de aprendizaje del perceptrón:

* Regla delta del mínimo error cuadrado.
* Regla delta generalizada o algoritmo de retropropagación de error.

Características del perceptrón.

* Entradas reales.
* Aprendizaje supervisado.
* El espacio debe ser linealmente separable.

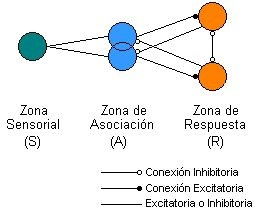


Figura 1. Esquema general de un Perceptrón.

**1) Aprendizaje del perceptrón**. Se presenta un ejemplo y se ejecuta una acción correctiva iterativamente hasta que la red aprende a producir la respuesta deseada. El conjunto de entrenamiento es el conjunto de los ejemplos de los que la red va a aprender. Usamos la siguiente notación: El vector de entrada al perceptrón es x = (x1, ..., xn). Si los pesos son los valores reales w1, ..., wn y el umbral es U, diremos que w = (w1, ..., wn, wn+1) con wn+1= – U es el vector extendido de Teorema de convergencia. Si las clases son linealmente separables, el algoritmo del perceptrón converge auna solución correcta en un número finito de pasos para cualquier elección inicial de pesos del perceptrón y que (x1, ..., xn, 1) es el vector extendido de entradas (se añade la entrada de tendencia o bias con valor 1 fijo). La computación de perceptrón puede expresarse mediante un producto escalar, ya que ∑i, wi, xi ≥ U es equivalente a w • x ≥ 0 donde w y x son los vectores extendidos de pesos y entradas, respectivamente.

La convergencia del algoritmo de aprendizaje del perceptrón se basa en que cada perceptrón realiza la comprobación w • x > 0, ó w • x ≥ 0, pero son equivalentes cuando el conjunto de entrenamiento es finito, lo cual siempre es cierto en problemas prácticos. Una forma habitual de comenzar el algoritmo de entrenamiento es inicializando aleatoriamente los pesos de la red y mejorar los parámetros iniciales, comprobando a cada paso, si puede lograrse una separación mejor del conjunto de entrenamiento. Cada vector de pesos w define un hiperplano que separa los puntos con salida 1 (w • x ≥ 0) de los puntos con salida 0 (w • x ≤ 0).

**2) Interpretación gráfica o geométrica del perceptrón.**

La ecuación de evaluación de las salidas representa un plano (un hiperplano en dimensión n) que divide el espacio en dos partes. El entrenamiento consiste en buscar el plano que hace la división adecuada, para que los puntos cuya salida es 1 queden todos en la misma región, y los que tienen salida esperada 0, que den todos en la otra región. Sean P y N dos conjuntos finitos de puntos en Rn que queremos separar linealmente. Se desea encontrar un vector w tal que su hiperplano asociado separe los dos conjuntos (salida 1 para P y 0 para N).

El error de un perceptrón con un vector de pesos w es el número de puntos incorrectamente clasificados. El algoritmo de entrenamiento debe minimizar este error E(w). Una de las posibles estrategias es la de usar un algoritmo voraz local que calcule el error del perceptrón para un vector de pesos dado, buscando una dirección en el espacio de pesos en la que moverse y actualizándolos, seleccionando para ello nuevos valores de acuerdo con la dirección de búsqueda.

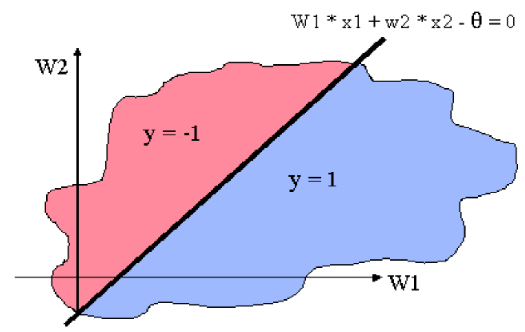


Figura 2. Interpretación gráfica del perceptrón.

**3) Algoritmo de Aprendizaje.** Realmente, se ha llamado Perceptrón a una RNA donde todas las entradas están conectadas a varias salidas, que se calculan como se ha indicado antes. Entonces la separabilidad lineal hay que mirarla en cada salida. El algoritmo de entrenamiento del perceptrón es supervisado y usa n vectores que se pueden clasificar en dos conjuntos P y N en el espacio extendido n+ 1-dimensional. Se considera el umbral como una entrada de tendencia con valor fijo 1.

Buscamos un vector w capaz de separar absolutamente ambos conjuntos. El algoritmo es (para cada salida):

**Iniciar**: w₀ aleatoriamente; t ← 0

**Repetir** **Mientras** queden vectores mal clasificados

seleccionar aleatoriamente un vector x

si x P (salida esperada 1) ywt•x< 0 (salida calculada 0) entonces

[Hay un error]

wt+1 <- wt + x; t <- t+1

fin si

si x 𝛜 N (salida esperada 0) y wt•x 0 (salida calculada 1) entonces

[Hay un error]

wt+1 <- wt – x; t <- t+1

fin si

fin repite

En general:

El valor correspondiente a la aplicación del objeto P a la red constituye la clasificación o salida, de la red para P. Este valor puede ser igual a T (la clasificación real de P) o diferente. si son iguales significa que la red a brindado el valor correcto para P, de lo contrario la red se ha equivocado.

Los posibles casos son los siguientes:

1. A = T la salida de la RED es igual a la clasificación de P por tanto funcionó correctamente y no hay que hacer cambios en los pesos.
2. T = 1 y A= 0
3. T = 0 y A=1

En los dos últimos casos la red se a equivocado por tanto sería necesario modificar los pesos (APRENDER).

Las acciones para el aprendizaje en cada caso serían:

1. t= a W N = W A
2. t=1, a=0 W N = W A + p
3. t=0, a=1 W N = W A – p

Simplificamos los casos haciendo: e = t-a

1. e = 0 W N = W A
2. e = 1 W N = W A + p
3. e = -1 W N = W A – p

En general:

W N = W A + e \* p

Bias N = Bias A + e

REFERENCIAS.

[1] Sistemas Expertos. ECURED. (2018) Disponible en: <https://www.ecured.cu/Sistemas_expertos>

[2] Regla del aprendizaje del perceptrón. https://es.slideshare.net/aerdna07/regla-de-aprendizaje-del-p erceptrn-simple

[3] El perceptrón Simple. http://www.lcc.uma.es/~munozp/documentos/modelos\_com putacionales/temas/Tema4MC-05.pdf

[5]http://sisbib.unmsm.edu.pe/bibvirtualdata/tesis/basic/carl os\_sm/conclu.pdf

[6] Redes de una Capa. ftp://decsai.ugr.es/pub/usuarios/castro/Actividades/Redes-Neuronales/Apuntes/Apuntes%20Javier%20Rodriguez%20Blazquez/Redes%20de%20una%20capa.pdf