

# Perceptrón

Eduardo Espinosa Avila

UNAM

21 de junio de 2018

- 1 Neuronas artificiales y el modelo de McCulloch-Pitts
- 2 Perceptrón de Rosenblatt
  - Función escalón
  - Perceptrón de aprendizaje
  - Ejemplo
  - Ejercicio
  - Problemas con el perceptrón
- 3 Bibliografía

## 1 Neuronas artificiales y el modelo de McCulloch-Pitts

## 2 Perceptrón de Rosenblatt

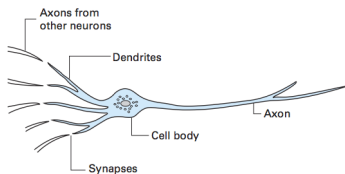
- Función escalón
- Perceptrón de aprendizaje
- Ejemplo
- Ejercicio
- Problemas con el perceptrón

## 3 Bibliografía

# El modelo de McCulloch-Pitts

La idea original del *perceptrón* se remonta al trabajo de Warren McCulloch y Walter Pitts, en 1943 observaron una analogía entre las neuronas biológicas y compuertas lógicas con salida binaria.

Una neurona puede verse como una subunidad de una red neuronal en un cerebro biológico. Las señales de magnitud variable entran por las dendritas; estas señales se acumulan en el cuerpo de la célula y, si el acumulado sobrepasa el umbral establecido, se genera una señal de salida que se entrega al axón.



## 1 Neuronas artificiales y el modelo de McCulloch-Pitts

## 2 Perceptrón de Rosenblatt

- Función escalón
- Perceptrón de aprendizaje
- Ejemplo
- Ejercicio
- Problemas con el perceptrón

## 3 Bibliografía

# El modelo de Rosenblatt

Posteriormente, en 1957 Frank Rosenblatt publicó el primer algoritmo para un perceptrón de aprendizaje. La idea básica es definir un algoritmo que aprende los valores de los pesos  $w$  que serán multiplicados con las características de entrada para poder decidir si la neurona dispara o no.

El perceptrón es un clasificador de aprendizaje *supervisado*, *determinista* y *a posteriori*.

# Función escalón de Heaviside I

Antes de describir a detalle el algoritmo de aprendizaje, definiremos algunos conceptos auxiliares.

Primero, llamaremos a las clases positiva y negativa para nuestra clasificación binaria como 1 y  $-1$  respectivamente. A continuación, definimos una función de activación  $g(z)$ :

$$f(x) = \begin{cases} 1, & \text{si } z \geq \theta \\ -1, & \text{en otro caso} \end{cases}$$

y

$$z = w_1x_1 + \cdots + w_nx_n = \sum_{i=1}^n w_i x_i$$

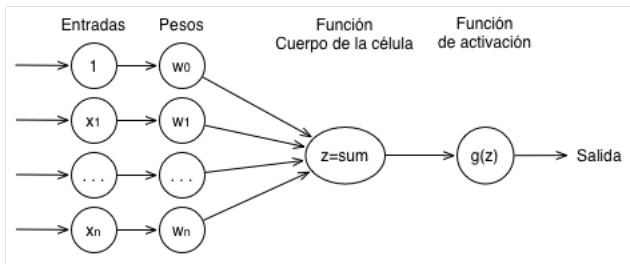
Se conoce como "función escalón unitario de Heaviside".

## Función escalón de Heaviside II

Además, es necesario añadir  $w_0 = \theta$  y  $x_0 = 1$ . De este modo, el parámetro de la función escalón se transforma en:

$$z = w_0x_0 + w_1x_1 + \cdots + w_nx_n = \sum_{i=0}^n w_ix_i$$

Y representa el siguiente modelo de neurona artificial:





# La regla del perceptrón de aprendizaje I

Este es el primer algoritmo de aprendizaje de máquina: el perceptrón *aprenderá* los pesos para cada señal de entrada para poder dibujar un límite de decisión que nos permita discriminar entre dos clases linealmente separables.

La regla del perceptrón de Rosenblatt es bastante simple y puede resumirse en los siguientes pasos:

## Regla de aprendizaje

inicializar los pesos a 0

para cada muestra de entrenamiento:

    calcular el valor de salida

    actualizar pesos

# La regla del perceptrón de aprendizaje II

La salida se calcula con la función escalón definida previamente y la actualización del peso se obtiene como  $w_j = w_j + \Delta w_j$ .

El valor para actualizar los pesos se obtiene mediante la regla de aprendizaje:

$$\Delta w_j = \eta (\text{resultado}_i - \text{salida}_i) x_{ij}$$

Donde  $\eta$  es la tasa (razón) de aprendizaje (una constante entre 0,0 y 1,0);  $\text{resultado}_i$  es la clase a la que pertenece la muestra y  $\text{salida}_i$  es la salida que predice el perceptrón en el paso actual.

# La regla del perceptrón de aprendizaje III

Es importante notar que el vector de pesos se actualiza simultáneamente. Para el caso particular de un conjunto de datos de 2 dimensiones, la actualización se obtiene como:

$$\Delta w_0 = \eta (\text{resultado}_i - \text{salida}_i)$$

$$\Delta w_1 = \eta (\text{resultado}_i - \text{salida}_i) x_{i1}$$

$$\Delta w_2 = \eta (\text{resultado}_i - \text{salida}_i) x_{i2}$$

# NAND binaria

Aplicar el algoritmo de aprendizaje del perceptrón para una neurona artificial que calcule la función booleana NAND con 2 parámetros definida como:

$x_1$	$x_2$	<i>salida</i>
0	0	1
0	1	1
1	0	1
1	1	-1

con  $\theta = 0$ ,  $w_0 = 1$  y  $\eta = 0,1$

# NAND binaria

Aplicando la regla:  
inicializar los pesos a 0

$$w = [0 \ 0 \ 0]$$

para cada muestra de entrenamiento:  
calcular el valor de salida  
actualizar pesos

Como se trata de un conjunto bidimensional, se puede utilizar la regla:

$$\Delta w_0 = \eta (\text{resultado}_i - \text{salida}_i)$$

$$\Delta w_1 = \eta (\text{resultado}_i - \text{salida}_i) x_{i1}$$

$$\Delta w_2 = \eta (\text{resultado}_i - \text{salida}_i) x_{i2}$$

# NAND binaria

Entonces:

- $\Delta w_0 = \eta (\text{resultado}_1 - \text{salida}_1)$
- $\Delta w_0 = 0,1 * (1 - g(1 * 0 + 0 * 0 + 0 * 0))$
- $\Delta w_0 = 0,1 * (1 - 1)$
- $\Delta w_0 = 0$
- $\therefore$  actualización:

$$w = [0 \ 0 \ 0]$$

- Puede verificarse que para las muestras 2 y 3 sucede algo similar  $\Delta w_j = 0 \therefore w = [0 \ 0 \ 0]$

# NAND binaria

En cambio, para la última muestra:

- $\Delta w_3 = \eta (\text{resultado}_3 - \text{salida}_3)$
- $\Delta w_3 = 0,1 * (-1 - g(1 * 0 + 1 * 0 + 1 * 0))$
- $\Delta w_3 = 0,1 * (-1 - 1)$
- $\Delta w_3 = -0,2$
- $\therefore$  actualización:

$$w = [-0,2 \quad -0,2 \quad -0,2]$$

- **Como hubo cambio en los pesos, se debe repetir el proceso con todas las muestras, pero usando los pesos actualizados**

# NAND binaria

Se deja como ejercicio hacer las iteraciones necesarias para obtener los pesos finales, es decir, hasta que ya no exista cambio con respecto a los pesos iniciales (es mucho mejor escribir un programa que lo realice).

Pero se muestran los valores:

$$① \quad w = [-0,2 \quad -0,2 \quad -0,2]$$

$$② \quad w = [0,0 \quad -0,4 \quad -0,2]$$

$$③ \quad w = [0,2 \quad -0,4 \quad -0,2]$$

$$④ \quad w = [0,2 \quad -0,4 \quad -0,4]$$

$$⑤ \quad w = [0,4 \quad -0,4 \quad -0,2]$$

$$⑥ \quad w = [0,4 \quad -0,4 \quad -0,2]$$

∴ Pesos finales:  $w = [0,4 \quad -0,4 \quad -0,2]$  en 6 iteraciones.



# NAND terciaria

Un buen ejercicio (algo tedioso si se quiere hacer a mano), es aplicar el algoritmo de aprendizaje del perceptrón para una neurona que calcule la función booleana NAND con 3 parámetros tomando en cuenta las muestras:

0	0	0	1
0	1	0	1
1	0	1	1
1	1	0	1
1	1	1	-1

con  $\eta = 0,1$  y obtener los valores que predice la neurona para las combinaciones faltantes

## El principal problema del perceptrón

El mismo Rosenblatt demostró matemáticamente es que la regla de aprendizaje converge si las dos clases pueden separarse linealmente, pero no se puede garantizar su convergencia si no se cumple esta condición.

## 1 Neuronas artificiales y el modelo de McCulloch-Pitts

## 2 Perceptrón de Rosenblatt

- Función escalón
- Perceptrón de aprendizaje
- Ejemplo
- Ejercicio
- Problemas con el perceptrón

## 3 Bibliografía



Eduardo Espinosa Avila. *Notas de IA.*

<http://dicyg.fi-c.unam.mx:8080/lalo/ia>



Wikipedia. *Perceptron.*

<https://en.wikipedia.org/wiki/Perceptron>