Redes Neuronales

Perceptrón y Adeline

Facultad de Ingeniería. Universidad del Valle

Septiembre de 2018

Contenido

Universidad del Valle

Contenido

Perceptrón

Definición

- Fue introducido por Rossenblat a finales de los año 50
- Se inspira en los procesos de aprendizaje de los animales (ejemplo la visión), en los cuales la información va atravesando diferentes capas de neuronas
- Es un modelo unidireccional, compuesto por dos capas de neuronas, una de entrada y otra de salida
- La operación de este tipo puede darse con n neuronas de entrada y m de salida

Perceptrón

Definición

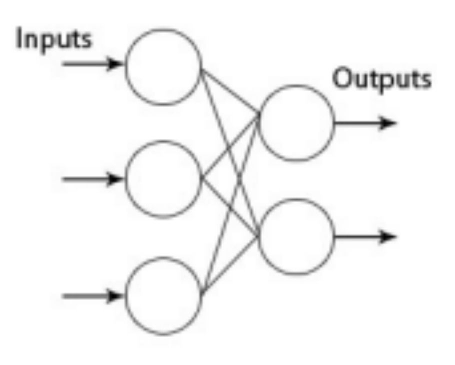


Figura 1: Modelo de Perceptrón, tomado de
http://neuroph.sourceforge.net/tutorials/Perceptron.html

Perceptrón

Definición

- Las neuronas de entrada no realizan ningún computo
- Se consideran señales discretas 0 o 1
- La operación para n neuronas de entrada y m de salida puede considerarse así:

$$y_i = H(\sum_{j=1}^{n} w_{ij}x_j - \Theta_i), \forall i, 1 \leq i \leq m$$

umbral

Donde H(x) es la función escalón.

Perceptrón

Definición

- El Perceptrón permite clasificar dos conjuntos linealmente separables en un plano o hiperplano
- La respuesta de la neurona es 1 si pertenece a la clase o 0 si no pertenece

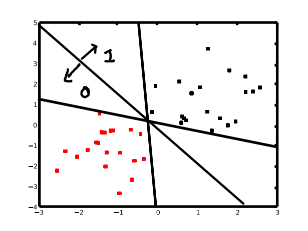
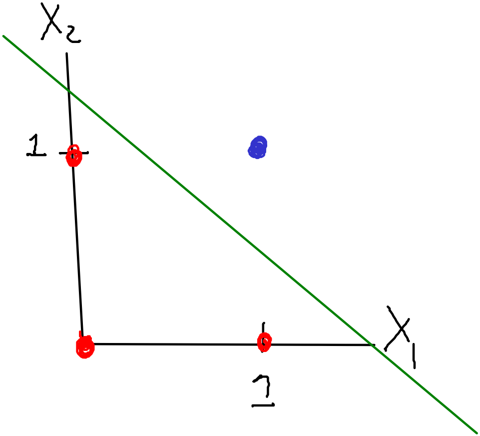


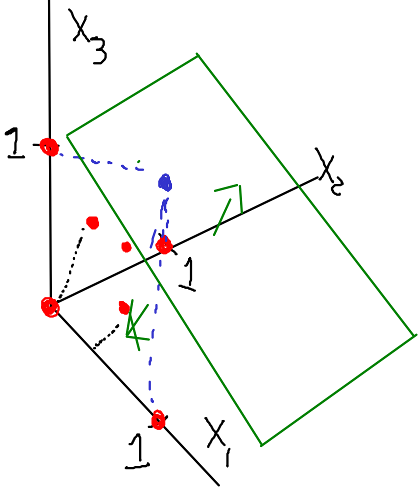
Figura 2: Conjunto linealmente separable, tomado de `https://en.wikipedia.org/`

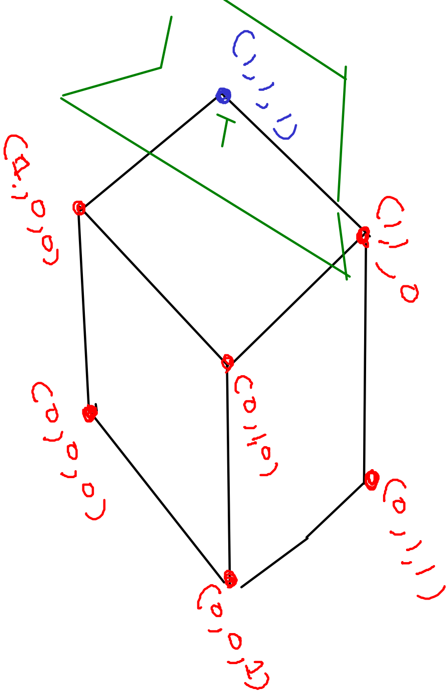
AND

X_1	X_2	S
0	0	0
0	1	0
1	0	0
1	1	1



X_1	X_2	X_3	S
0	0	0	0
			0
⋮			0
			⋮
1	1	1	1




(1,1,1)
(1,0,0)
(1,1,0)
(0,1,0)
(0,0,0)
(0,1,1)
(0,0,1)
T

Perceptrón

Ejemplo

- Sea una neurona tipo perceptron con entrada x_1 y x_2
- Entonces la operación se define como:

$$y = H(w_1 x_1 + \overset{w}{x}_2 x_2 - \Theta)$$

Perceptrón

Ejemplo

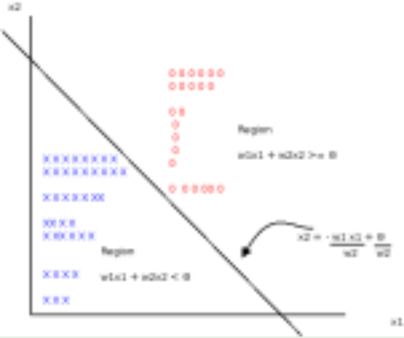


Figura 3: Regiones de decisión del plano, tomado de [Brio and Molina, 2005]

Perceptrón

Definición

Como se puede ver se divide el plano en dos regiones. Como se puede ver se requiere que el problema a solucionar tenga **solución lineal**

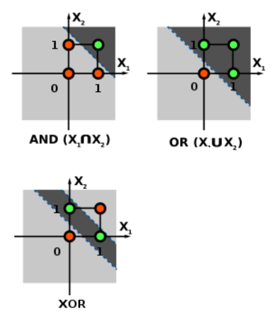


Figura 4: Casos de compuertas https://en.wikipedia.org/

Perceptrón

Algoritmo de aprendizaje

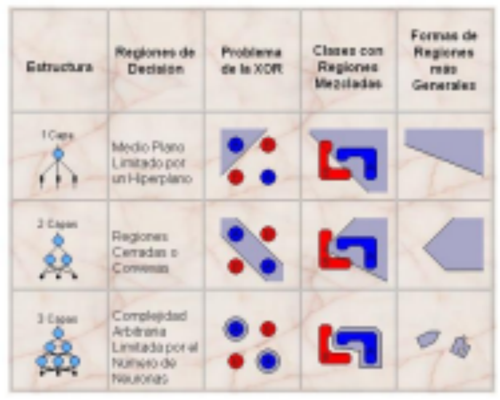


Figura 5: Regiones de decisión perceptron [Lippmann, 1988]

Perceptrón

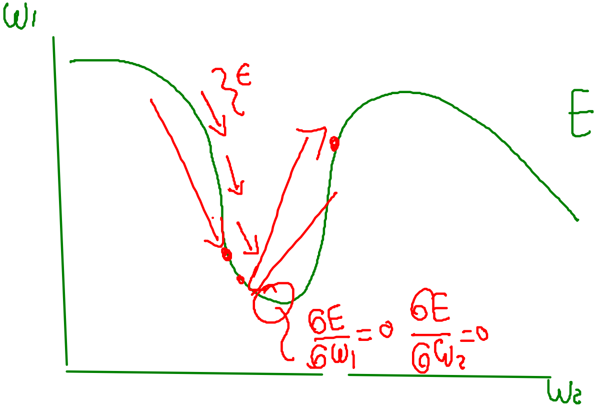
Algoritmo de aprendizaje

Vamos a trabajar el perceptrón de una capa

- Se basa en la corrección de errores
- Vamos a introducir una tasa de aprendizaje ϵ: Indica el ritmo de aprendizaje
- Dados unos patrones x^u, salidas obtenidas y^u y salidas deseadas t^u
- Los pesos iniciales son aleatorios entre -1 y 1. Se utiliza la función de activación signo y entradas $\{-1, 1\}$.
- Se examina cada patrón y aplicamos la relación de cambio:

$$\Delta w_{ij}^u(t) = \epsilon.(t_i^u - y_i^u)x_j^u$$

A esto se le conoce como **regla del perceptrón**



w_1
ϵ
E
$\frac{\partial E}{\partial w_1} = 0$
$\frac{\partial E}{\partial w_2} = 0$
w_2

Perceptrón

Algoritmo de aprendizaje

Vamos a trabajar el perceptrón de una capa

- Se basa en la corrección de errores
- Vamos a introducir una tasa de aprendizaje ϵ: Indica el ritmo de aprendizaje
- Dados unos patrones x^u, salidas obtenidas y^u y salidas deseadas t^u

Perceptrón

Algoritmo de aprendizaje

Vamos a trabajar el perceptrón de una capa

- Los pesos iniciales son aleatorios entre -1 y 1. Se utiliza la función de activación signo y entradas $\{-1, 1\}$.
- Se examina cada patrón y aplicamos la relación de cambio:

$$\boxed{\Delta w_{ij}^{u}(t) = \epsilon.(t_i^{u} - y_i^{u})x_j^{u}}$$

A esto se le conoce como **regla del perceptrón**

Universidad del Valle

Perceptrón

Algoritmo de aprendizaje

La función signo (sgn) se define así:

$$sgn(n) = \begin{cases} 1 & si \quad n > 0 \\ 0 & si \quad n = 0 \\ -1 & si \quad n < 0 \end{cases}$$

Perceptrón

Algoritmo de aprendizaje

Para comprender el Perceptrón se mostrará en una forma gráfica

$$y_i^u(t) = sgn(\sum_{j=1}^{n} w_{ij}x_j^u - \Theta_i) = sgn(||w_i||.||x^u||cos(\phi))$$

Perceptrón

Algoritmo de aprendizaje

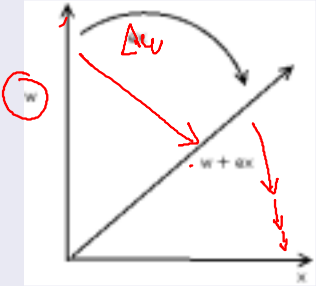


Figura 6: Aplicación regla perceptron [Brio and Molina, 2005]

Perceptrón

Algoritmo de aprendizaje

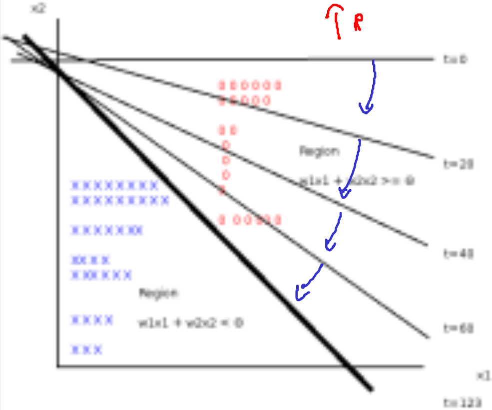


Figura 7: Aplicación iterativa de las reglas de decision [Brio and Molina, 2005]

Perceptrón

Algoritmo de aprendizaje

La idea es etiquetas con -1 y 1 dos regiones en el espacio

1. Inicializar los pesos aleatoriamente entre [-1 y 1]
2. Para el estado t. Calcular:

$$y^u(k) = sgn(\sum_{j=1}^{n}(w_j(t)x_j)$$

3. Corregir pasos sinápticos (Si $t_j^u \neq y_j^u$)

$$w_j(t+1) = w_j(t) + \epsilon[t_j^u - y_j^u]x^u, j = 1, 2, .., n$$

4. Para si no se han modificado los pesos en los últimos p patrones o se ha llegado a un número de iteraciones especificado.

Perceptrón

Algoritmo de aprendizaje

Miremos la compuerta AND

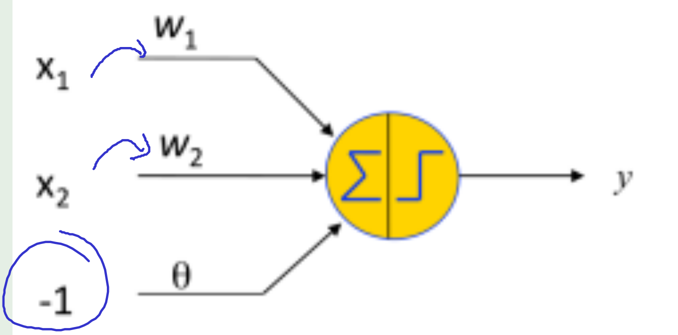


Figura 8: Perceptrón compuesta AND

Perceptrón

Algoritmo de aprendizaje

Miremos la compuerta AND

Entrada	Salida $t^u(k)$
(-1,-1)	-1
(-1,1)	-1
(1,-1)	-1
(1,1)	1

Cuadro 1: Función AND con lógica función signo

Perceptrón

Algoritmo de aprendizaje

1. Inicialización de pesos. Elegimos $\epsilon = 0{,}5$

$$w_1 = 0{,}4,\ w_2 = -0{,}2,\ \Theta = 0{,}6$$

$-0.4 + 0.2 - 0.6$

2. Con t = 1, patrón (-1,-1), $y^u = sgn(\underline{-0{,}8}) = \underline{-1}$. Esta bien ya que esperamos -1. $p = 1$
3. Para t = 1, patrón (-1, 1), $y^u = sgn(\underline{-1{,}2}) = \underline{-1}$. Esta bien. $p = 2$

Algoritmo de aprendizaje

4. Para t = 1, patrón (1,-1), $y^u = sgn(0) = 0$ Esta mal, ya que esperamos -1. Actualizamos pesos

$$
\begin{aligned}
w_1(2) &= w_1(1) + \epsilon[t^u(1) - y^u(1)]x_1 = 0{,}4 + 0{,}5[-1-0]\overset{x_1}{(1)} \\
w_1(2) &= -0{,}1 \\
w_2(2) &= -0{,}2 + 0{,}5[-1-0](-1) = 0{,}3 \\
\Theta(2) &= 0{,}6 + 0{,}5[-1-0](-1) = 1{,}1
\end{aligned}
$$

5. Ya que actualizamos, ahora $t = 2$, y revisamos.

Perceptrón

Algoritmo de aprendizaje

$$w_1 = -0{,}1, w_2 = 0{,}3, \Theta = 1{,}1$$

6. Para t = 2; patrón (-1,-1), $y^u = sgn(-1{,}1) = -1$. Correcto $p = 1$
7. Para t = 2; patrón (-1,1), $y^u = sgn(-0{,}7) = -1$. Correcto $p = 2$
8. Para t = 2; patrón (1,-1), $y^u = sgn(-1{,}5) = -1$. Correcto $p = 3$

Universidad del Valle

Perceptrón

Algoritmo de aprendizaje

$$w_1 = -0{,}1, w_2 = 0{,}3, \Theta = 1{,}1$$

9 Para t = 2; patrón (1,1), $y^u = sgn(-0{,}9) = -1$. Incorrecto

$$w_1(3) = -0{,}1 + 0{,}5[1 - (-1)](1) = 0{,}9$$
$$w_2(3) = 0{,}3 + 0{,}5[1 - (-1)](1) = 0{,}7$$
$$\Theta(3) = 1{,}1 + 0{,}5[1 - (-1)](-1) = 0{,}1$$

10 Ya que actualizamos, ahora $t = 3$, y revisamos.

Algoritmo de aprendizaje

$$w_1 = 0{,}9, w_2 = 0{,}7, \Theta = 0{,}1$$

11. Para t = 3; patrón (-1,-1), $y^u = sgn(-1{,}7) = -1$. Correcto
    $p = 1$
12. Para t = 3; patrón (-1,1), $y^u = sgn(-0{,}3) = -1$. Correcto
    $p = 2$
13. Para t = 3; patrón (1,-1), $y^u = sgn(0{,}1) = 1$. Incorrecto

Universidad del Valle

Perceptrón

Algoritmo de aprendizaje

$$w_1 = 0{,}9, w_2 = 0{,}7, \Theta = 0{,}1$$

14 Para t = 3; patrón (1,-1), $y^u = sgn(0{,}1) = 1$. Incorrecto

$$\begin{aligned} w_1(4) &= 0{,}9 + 0{,}5[-1 - 1](1) = -0{,}1 \\ w_2(4) &= 0{,}7 + 0{,}5[-1 - 1](-1) = 1{,}7 \\ \Theta(4) &= 0{,}1 + 0{,}5[-1 - 1](-1) = 1{,}1 \end{aligned}$$

15 Ya que actualizamos, ahora $t = 4$, y revisamos.

Universidad del Valle

Algoritmo de aprendizaje

$$w_1 = -0{,}1, w_2 = 1{,}7, \Theta = 1{,}1$$

16 Para t = 4; patrón (-1,-1), $y^u = sgn(-2{,}7) = -1$. Correcto $p = 1$

17 Para t = 4; patrón (-1,1), $y^u = sgn(0{,}7) = -1$. Incorrecto

$$\begin{aligned} w_1(5) &= -0{,}1 + 0{,}5[-1 - 1](-1) = 0{,}9 \\ w_2(5) &= 1{,}7 + 0{,}5[-1 - 1](1) = 0{,}7 \\ \Theta(5) &= 1{,}1 + 0{,}5[-1 - 1](-1) = 2{,}1 \end{aligned}$$

Ahora con $t = 5$

Perceptrón

Algoritmo de aprendizaje

$$w_1 = 0{,}9, w_2 = 0{,}7, \Theta = 2{,}1$$

18 Para t = 5; patrón (1,1), $y^u = sgn(-0{,}5) = -1$. Incorrecto, diferencia salidas 2.

$$w_1(6) = 0{,}9 + 0{,}5[2](1) = 1{,}9$$
$$w_2(6) = 0{,}7 + 0{,}5[2](1) = 1{,}7$$
$$\Theta(6) = 2{,}1 + 0{,}5[2](-1) = 1{,}1$$

Ahora con $t = 6$

Algoritmo de aprendizaje

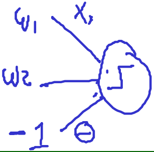
$$w_1 = 1{,}9, w_2 = 1{,}7, \Theta = 1{,}1$$

19. Para t = 6; patrón (-1,-1), $y^u = sgn(-4{,}7) = -1$. Correcto $p = 1$
20. Para t = 6; patrón (-1,1), $y^u = sgn(-1{,}3) = -1$. Correcto $p = 2$
21. Para t = 6; patrón (1,-1), $y^u = sgn(-0{,}9) = -1$. Correcto $p = 3$
22. Para t = 6; patrón (1,1), $y^u = sgn(2{,}5) = 1$. Correcto $p = 4$

Termina.

Perceptrón

$$Sgn(x) = \begin{cases} 1 & x>0 \\ 0 & x=0 \\ -1 & x<0 \end{cases}$$



$$W(t+1) = W(t) + \epsilon\,(t-y)\,X$$

Ejercicio

1. Aplica el algoritmo para la compuerta AND, con $w_1 = 0{,}7$, $w_2 = -1{,}8$, $\Theta = 1{,}4$, $\epsilon = 0{,}5$
2. Aplica el algoritmo para la compuerta OR, con $w_1 = 0{,}7$, $w_2 = -1{,}8$, $\Theta = 1{,}4$, $\epsilon = 0{,}5$

x_1	x_2	y
-1,	-1,	-1

	x_1	x_2	y
P_1	-1	-1	-1

$w_1 = 0.7 \qquad w_2 = -1.8 \qquad \theta = 1.4$

$\text{Sgn}(-0.7 + 1.8 - 1.4) = -1$ ✓

P_2, -1, 1, -1 $\qquad \text{Sgn}(-0.7 - 1.8 - 1.4) = -1$ ✓

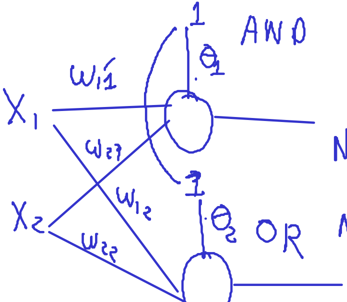
P_3, 1, -1, -1 $\qquad \text{Sgn}(0.7 + 1.8 - 1.4) = 1$ ✗

$$w_1 = 0.7 + 0.5(-2)(1) = -0.3$$

$$w_2 = -1.8 + 0.5(-2)(-1) = -0.8$$

$$\theta = 1.4 + 0.5(-2)(-1) = 2.4$$

$P_4 \rightarrow w_1 = 0.7, \quad w_2 = 0.2 \quad \theta = 1.4$



$$\begin{array}{c} N_1 \leftarrow \\ N_2 \leftarrow \end{array} \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

AND

X_1	X_2	S_1	S_2
-1	-1	-1	-1
-1	1	-1	1
1	-1	-1	1
1	1	1	1

Contenido

Limitaciones del perceptrón

Limitaciones

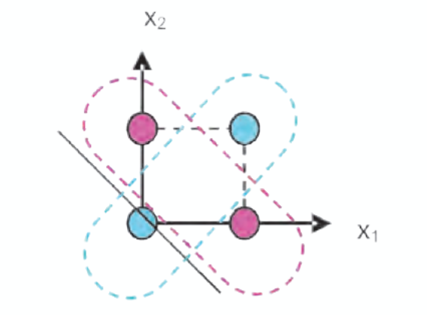
1. El perceptrón monocapa no puede trabajar con problemas que no sean linealmente separables
2. Para esto utilizamos el perceptrón multicapa, la cual cuenta con una capa de entrada, capas ocultas y una capa de salida

Limitaciones del perceptrón

Limitaciones

Para el caso de la función XOR, tenemos el siguiente problema:

Figura 9: Problema del perceptrón con compuerta XOR, tomado de [Eduardo and Jesus Alfonso, 2009]



Contenido

Adeline

Definición

- Fue introducido por Widrow [Widrow and Hoff, 1988], [Widrow and Winter, 1988] entre 1959 y 1988.
- Es de respuesta lineal a diferencia del perceptrón
- Puede trabajar con entradas continuas
- Se incorpora un elemento adicional llamado **bias** u **umbral** Θ. La cual se suma a la entrada (usualmente es -1)
- Utiliza mínimos cuadrados para el cálculo del error.
- Se tiene una tasa de aprendizaje ϵ

Adeline

Regla del gradiente descendiente

- Si las entradas tiene vectores de entrada ortogonales, se podrá llegar a asociaciones perfectas
- La salida de la neurona es $y = \sum_{j=1}^{n} x_j w_j + \Theta$. Ya que la función de activación es lineal
- El cambio se basa en el cálculo del gradiente descendiente para los patrones de entrada. En este caso el error cuadrático $Err = \frac{1}{2} \sum_{j=0}^{n} (t^u(j) - y^u(j))^2$

Adeline

Regla del gradiente descendiente

- Lo que se busca es modificar los valores de forma iterativa mediante la regla del descenso del gradiente:

$$\Delta_p w_j = -\epsilon \frac{\partial Err^P}{\partial w_j}$$

Universidad del Valle

Adeline

Regla del gradiente descendiente

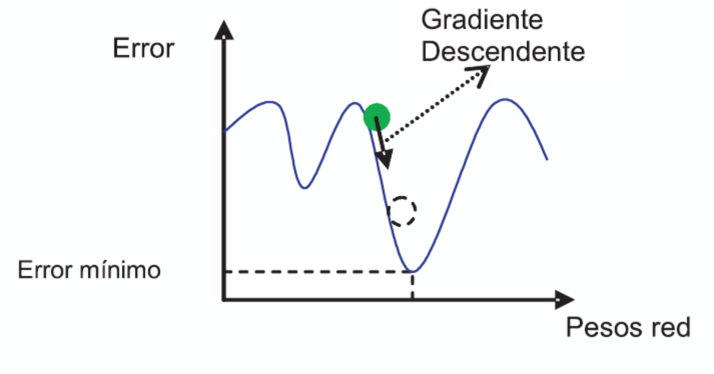


Figura 10: Regla del gradiente descendiente, tomado de [Eduardo and Jesus Alfonso, 2009]

Adeline

Regla del gradiente descendiente

- Tomando en cuenta que el error global cuadrático medio es:

$$E = \frac{1}{2}\sum_{u=1}^{m}(t^u - y^u)$$

- La regla del gradiente descendiente busca el mínimo global, aplicando la regla en un patrón p cualquiera de la siguiente forma:

$$\Delta w_j = -\epsilon \frac{\partial E^p}{\partial w_j}$$

Esta va en sentido contrario, ya que deseamos hacer la derivada del error igual a 0

Universidad del Valle

Adeline

Regla del gradiente descendiente

- Al aplicar la derivada obtenemos que:

$$\Delta w_j = \alpha(t^u - y^u) * x_j$$

Adeline

Algoritmo de aprendizaje

1. Inicialice los pesos aleatorios
2. Para cada patrón, actualice los pesos a razón de:

$$w_i(t+1) = w(t) + \epsilon * \Delta w_j$$

3. Actualizamos los pesos y Θ
4. Puede detenerse cuando todos los patrones cumplen la salida deseada o bien se han cumplido cierto número de iteraciones.