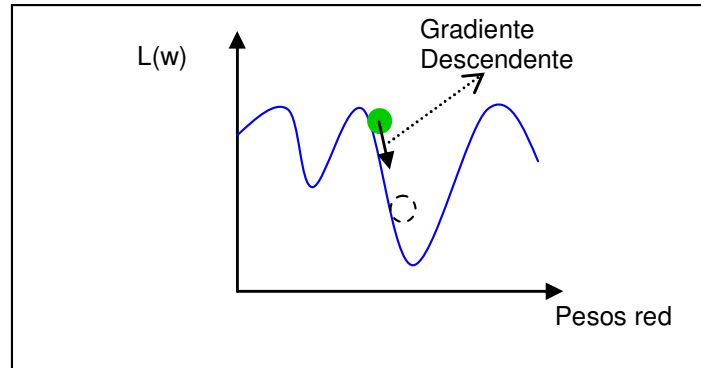


## Gradiente Descendente

Supongamos que nos encontramos en un sistema montañoso muy complejo en un lugar a gran altura y totalmente cubierto por neblina. Estamos perdidos, pero sabemos que en el valle hay una población donde podremos obtener ayuda, ¿qué estrategia utilizamos para llegar a dicha población? Muy seguramente estamos de acuerdo en que lo mejor es empezar a bajar y siempre mantener esta tendencia, por ejemplo, podríamos seguir el cauce de un río, pues esta corriente de agua nos garantiza que siempre caminaremos paulatinamente hacia un lugar más bajo.



**Fig. 1 Gradiente Descendente.**

Este es el principio del algoritmo utilizado en el proceso de aprendizaje de la red ADALINE, en donde, partiendo desde un punto aleatorio en la superficie de error, buscaremos un punto donde el error sea lo mínimo posible, siguiendo una trayectoria descendente, como se ilustra en la figura 1. A esta técnica de búsqueda la denominaremos Algoritmo de Gradiente Descendente.

La ecuación de actualización de pesos para una red neuronal la podemos definir en términos del peso en el instante  $t$  y de la variación del peso sináptico,

$$w(t+1) = w(t) + \Delta w(t) \quad (1)$$

donde,

- $w(t+1)$  : Valor actualizado del peso sináptico
- $w(t)$  : Valor actual del peso sináptico
- $\Delta w(t)$  : Variación del peso sináptico

Si variamos los pesos de la red en un factor proporcional al gradiente negativo del error de la red respecto a los pesos, logramos que los pesos así generados disminuyan la función de pérdida o costo que para este caso lo definimos en términos del error cuadrático promedio en la ecuación 2 y para una red ADALINE con una neurona de salida y con  $P$  patrones de entrenamiento.

$$L = MSE = \frac{1}{P} \sum_{p=1}^P (d_p - y_p)^2 \quad (2)$$

$p$  :  $p$ -ésimo patrón del conjunto de entrenamiento  
 $d_p$  : Valor deseado de salida para el  $p$ -ésimo patrón  
 $y_p$  : Valor de salida de la red ADALINE para el  $p$ -ésimo patrón

El objetivo de algoritmo es encontrar el conjunto de pesos que hagan que la función de pérdida  $L$  sea mínima; si hacemos la variación de los pesos proporcional al gradiente negativo, el cambio en el peso de la  $i$ -ésima conexión la representamos por la ecuación 3, donde  $\alpha$  es el factor de aprendizaje de la red.

$$\Delta w_i = -\alpha \frac{\partial L}{\partial w_i} \quad (3)$$

Si en la ecuación anterior aplicamos la regla de la cadena para obtener la derivada, queda en términos de la derivada del índice de desempeño respecto del error, la derivada del error respecto de la salida de la red ADALINE y la derivada de la salida de la red respecto del peso de la  $i$ -ésima conexión.

$$\Delta w_i = -\alpha \frac{\partial L}{\partial e} \cdot \frac{\partial e}{\partial y} \cdot \frac{\partial y}{\partial w_i} \quad (4)$$

Tras evaluar las derivadas parciales de la ecuación 4, obtenemos los siguientes valores,

$$\begin{aligned}
 \frac{\partial L}{\partial e} &= \frac{2}{P} (d_p - y_p) = \frac{2}{P} e_p \\
 \frac{\partial e}{\partial y} &= -1 \\
 \frac{\partial y}{\partial w_i} &= x_i
 \end{aligned} \quad (5)$$

Estos valores de las derivadas parciales los reemplazamos en la ecuación de actualización para obtener el valor final de la variación del peso de la  $i$ -ésima conexión.

$$\begin{aligned}
 \Delta w_i &= -\alpha \frac{2}{P} e_p (-1) x_i \\
 \Delta w_i &= \alpha \frac{2}{P} e_p x_i
 \end{aligned} \quad (6)$$

Retomando la ecuación 2.11 y 2.16, la expresión para la actualización del peso de la  $i$ -ésima conexión está dada por la ecuación 7.

$$w_i(t+1) = w_i(t) + \Delta w_i$$

$$w_i(t+1) = w_i(t) + \alpha \frac{2}{P} (d_p - y_p) x_i \quad (7)$$

$$w_i(t+1) = w_i(t) + \alpha \frac{2}{P} e_p x$$

Con base en las ecuaciones obtenidas, propongamos un algoritmo de aprendizaje para la red ADALINE, cuyos pasos los describimos a continuación.

### **Cuadro 2.2 Pasos para el entrenamiento de una red ADALINE**

#### **Paso 1**

Asignamos los valores iniciales de forma aleatoria a los pesos  $w_{ji}$  y el umbral  $b_j$

Asignamos un valor al parámetro de aprendizaje  $\alpha$

Definimos un valor para el error mínimo aceptado en el entrenamiento de la red

#### **Paso 2**

Mientras la condición de parada sea falsa, ejecutamos los pasos 3 al 7.

#### **Paso 3.**

Para cada patrón de entrenamiento  $\{\mathbf{x}_{pj}, \mathbf{y}_{pj}\}$  ejecutamos los pasos 4 al 5

#### **Paso 4.**

Calculamos la salida de la red ADALINE

$$y_{pj} = Neta_j = \sum_{i=1}^N w_{ji} x_{pi} + b_j \quad (8)$$

#### **Paso 5.**

Calculamos la función de pérdida ante el  $p$ -ésimo patrón. Este caso la función de pérdida es el error cuadrático promedio

$$L_p = MSE_p = \frac{1}{M} \sum_{j=1}^M (d_{pj} - y_{pj})^2 \quad (9)$$

**Paso 6.**

Actualizamos los pesos

$$w_i(t+1) = w_i(t) + \alpha \frac{2}{M} (d_{pj} - y_{pj}) x_i \quad (10)$$

$$w_i(t+1) = w_i(t) + \alpha \frac{2}{M} e_{pj} x_i \quad (11)$$

$$b_j(t+1) = b_j(t) + \alpha \frac{2}{M} e_{pj} \quad (12)$$

**Paso 7.**

Calculamos el error global de la red y si es menor que un valor mínimo detenemos el algoritmo. En caso contrario, retornamos al paso 2.