Fisiología básica de una neurona

Diego Milone

Inteligencia Computacional
Departamento de Informática

FICH-UNL

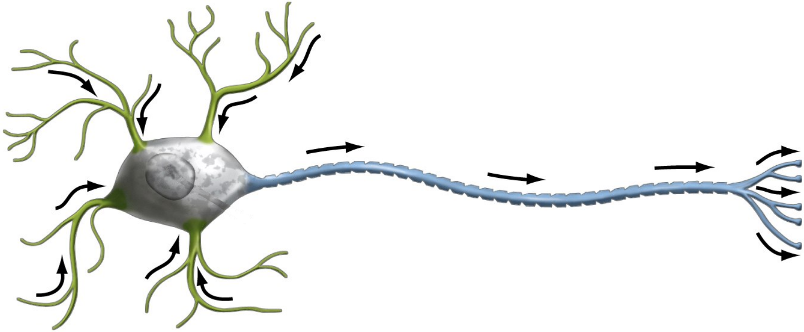
La inspiración biológica en redes neuronales

- La corteza cerebral $\ldots 10^{11} \ldots$
- Redes de neuronas
  - No linealidad, Paralelismo
  - Aprendizaje, Generalización
  - Adaptabilidad, Robustez

La inspiración biológica en redes neuronales

- La corteza cerebral $\ldots 10^{11} \ldots$
- Redes de neuronas
  - No linealidad, Paralelismo
  - Aprendizaje, Generalización
  - Adaptabilidad, Robustez
- La neurona biológica: soma, dendritas, axón...
- Fisiología de la neurona:
  - sinapsis, neurotransmisores
  - despolarización, comportamiento todo/nada
  - propagación del impulso
  - refuerzo de las sinapsis... aprendizaje
- Modelo simplificado de neurona

Neurona biológica



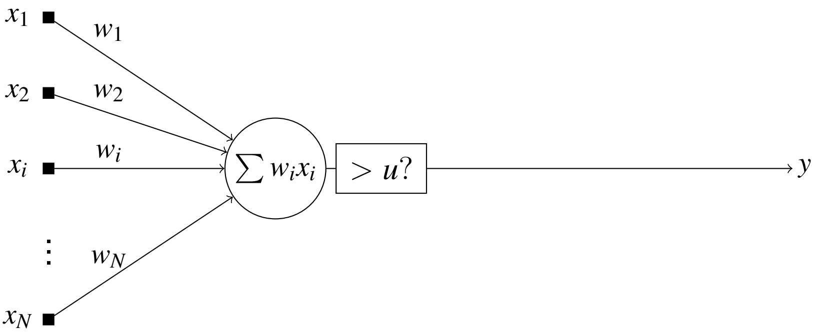
Modelo simplificado de neurona

Diego Milone

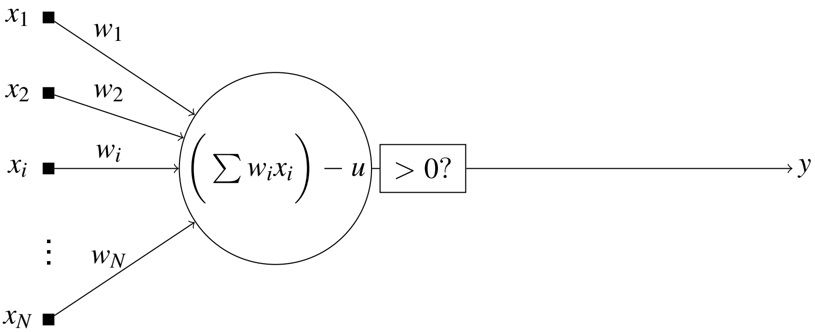
Inteligencia Computacional
Departamento de Informática

FICH-UNL

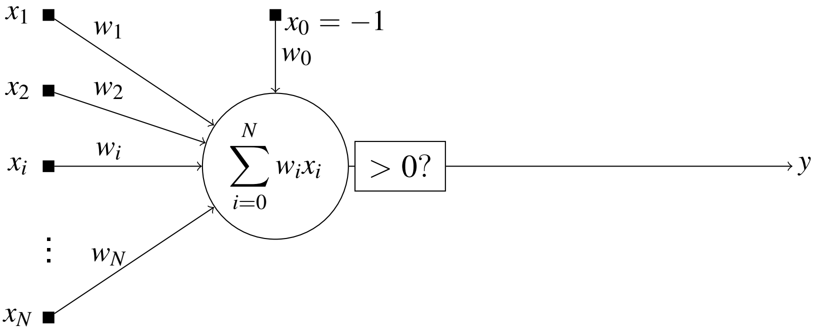
Modelo de neurona



Modelo de neurona



Modelo de neurona



Perceptrón simple

Perceptrón simple

- Modelo matemático del perceptrón simple
  - Producto interno y umbral: $y = \phi(v - u)$

Perceptrón simple

- Modelo matemático del perceptrón simple
  - Producto interno y umbral: $y = \phi(v - u) = \phi\left(\sum_{i=1}^{N} w_i x_i - u\right)$

Perceptrón simple

- Modelo matemático del perceptrón simple
  - Producto interno y umbral: $y = \phi(v - u) = \phi\left(\sum_{i=1}^{N} w_i x_i - u\right)$
  - Entrada extendida: $x_0 = -1, w_0 = u$

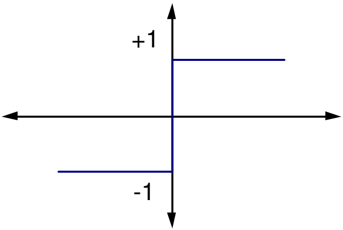
Perceptrón simple

- Modelo matemático del perceptrón simple
  - Producto interno y umbral: $y = \phi(v - u) = \phi\left(\sum_{i=1}^{N} w_i x_i - u\right)$
  - Entrada extendida: $x_0 = -1, w_0 = u$

$$y = \phi\left(\sum_{i=0}^{N} w_i x_i\right) = \phi\left(\langle \mathbf{w}, \mathbf{x} \rangle\right)$$

Perceptrón simple

- Modelo matemático del perceptrón simple
  - Producto interno y umbral: $y = \phi(v - u) = \phi\left(\sum_{i=1}^{N} w_i x_i - u\right)$
  - Entrada extendida: $x_0 = -1, w_0 = u$
  $$y = \phi\left(\sum_{i=0}^{N} w_i x_i\right) = \phi\left(\langle \mathbf{w}, \mathbf{x} \rangle\right)$$
  - Funciones de activación $\phi(z)$



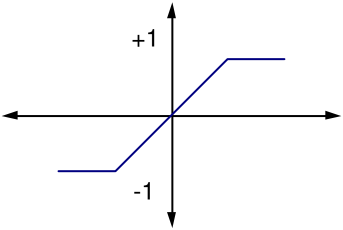
$$\operatorname{sgn}(z) = \begin{cases} -1 & \text{si } z < 0 \\ +1 & \text{si } z \geq 0 \end{cases}$$

Perceptrón simple

- Modelo matemático del perceptrón simple
  - Producto interno y umbral: $y = \phi(v - u) = \phi\left(\sum_{i=1}^{N} w_i x_i - u\right)$
  - Entrada extendida: $x_0 = -1, w_0 = u$

$$y = \phi\left(\sum_{i=0}^{N} w_i x_i\right) = \phi\left(\langle \mathbf{w}, \mathbf{x}\rangle\right)$$

  - Funciones de activación $\phi(z)$



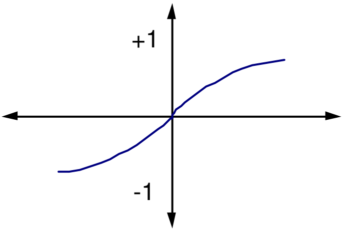
$$\mathrm{sln}(z) = \begin{cases} -1 & \text{si } z < -a \\ \alpha z & \text{si } -a < z < 0 \\ +1 & \text{si } z \geq a \end{cases}$$

Perceptrón simple

- Modelo matemático del perceptrón simple
  - Producto interno y umbral: $y = \phi(v - u) = \phi\left(\sum_{i=1}^{N} w_i x_i - u\right)$
  - Entrada extendida: $x_0 = -1, w_0 = u$

$$y = \phi\left(\sum_{i=0}^{N} w_i x_i\right) = \phi\left(\langle \mathbf{w}, \mathbf{x} \rangle\right)$$

  - Funciones de activación $\phi(z)$



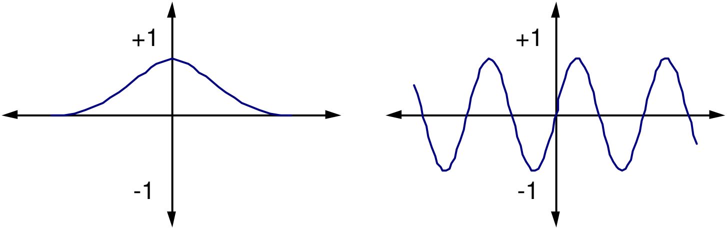
$$\mathrm{sig}(z) = \frac{1 - e^{-az}}{1 + e^{-az}}$$

Perceptrón simple

- Modelo matemático del perceptrón simple
  - Producto interno y umbral: $y = \phi(v - u) = \phi\left(\sum_{i=1}^{N} w_i x_i - u\right)$
  - Entrada extendida: $x_0 = -1, w_0 = u$

$$y = \phi\left(\sum_{i=0}^{N} w_i x_i\right) = \phi\left(\langle \mathbf{w}, \mathbf{x} \rangle\right)$$

  - Funciones de activación $\phi(z)$



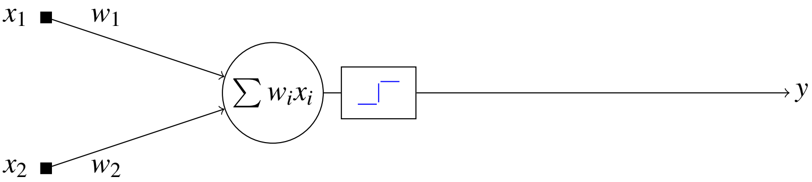
Un perceptrón simple con 2 entradas

Diego Milone

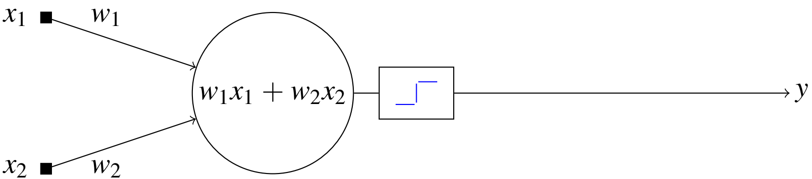
Inteligencia Computacional
Departamento de Informática

FICH-UNL

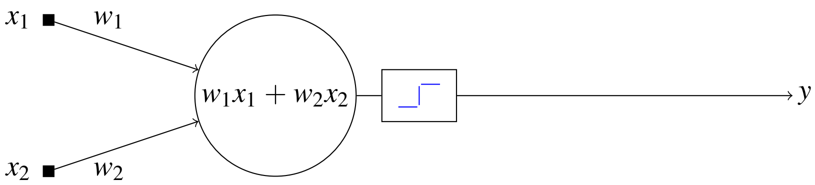
Perceptrón simple: ejemplo con 2 entradas



Perceptrón simple: ejemplo con 2 entradas

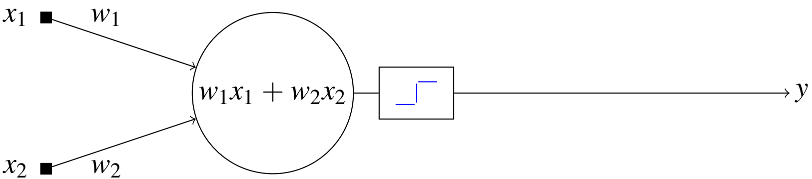


Perceptrón simple: ejemplo con 2 entradas



$$y = \mathrm{sgn}(w_1 x_1 + w_2 x_2)$$

Perceptrón simple: ejemplo con 2 entradas

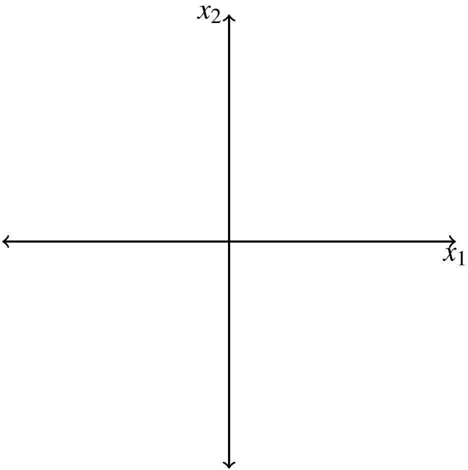


$$y = \mathrm{sgn}(w_1 x_1 + w_2 x_2)$$

$$w_1 x_1 + w_2 x_2 > 0?$$

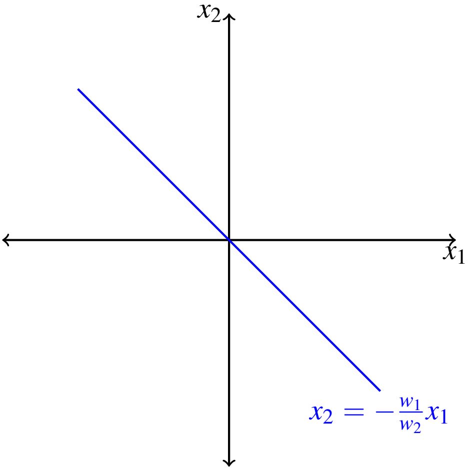
Perceptrón simple: ejemplo con 2 entradas

$$w_1x_1 + w_2x_2 = 0$$



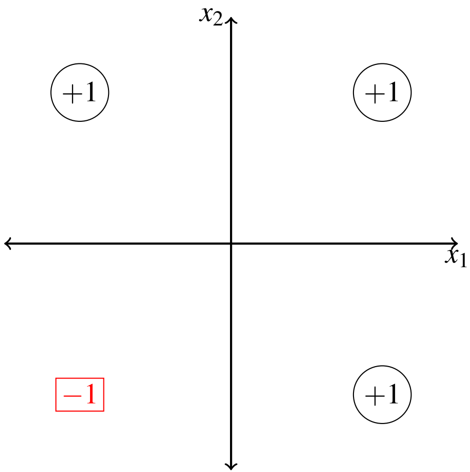
Perceptrón simple: ejemplo con 2 entradas

$$w_1 x_1 + w_2 x_2 = 0$$



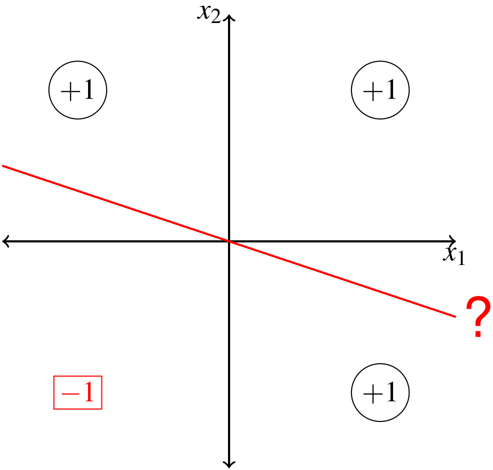
Perceptrón simple: ejemplo con 2 entradas

Caso del OR



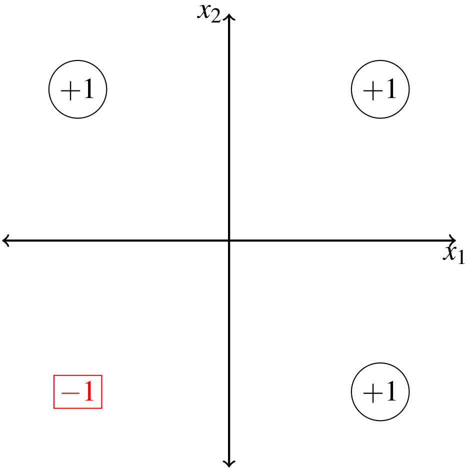
Perceptrón simple: ejemplo con 2 entradas

Caso del OR



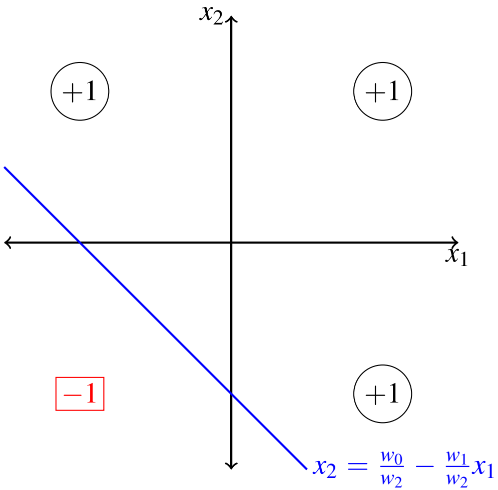
Perceptrón simple: necesidad del bias

$$w_1x_1 + w_2x_2 - w_0 = 0$$

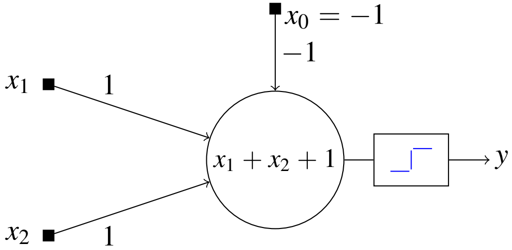


Perceptrón simple: necesidad del bias

$$w_1 x_1 + w_2 x_2 - w_0 = 0$$



Perceptrón simple: probamos el OR?



Perceptrón simple

Pero cómo...

¿no aprende solo a partir de los datos?

Algoritmos de aprendizaje: perceptrón simple

Diego Milone

Inteligencia Computacional
Departamento de Informática

FICH-UNL

Perceptrón simple: aprendizaje

Entrenamiento por corrección de error:

1. Inicialización al azar
2. Se muestran muchos ejemplos con las salidas esperadas y en cada caso:

Perceptrón simple: aprendizaje

Entrenamiento por corrección de error:

1. Inicialización al azar
2. Se muestran muchos ejemplos con las salidas esperadas y en cada caso:
   - **SI** la salida de la red es correcta
     no se hacen cambios: principio de mínima perturbación

Perceptrón simple: aprendizaje

Entrenamiento por corrección de error:

1. Inicialización al azar
2. Se muestran muchos ejemplos con las salidas esperadas y en cada caso:
    - **SI** la salida de la red es correcta
      no se hacen cambios: principio de mínima perturbación
    - **SI** la salida de la red es incorrecta
      penalización: se actualizan los w_i en el sentido opuesto al cual con el que contribuyeron a la salida incorrecta

Perceptrón simple: aprendizaje

Entrenamiento por corrección de error:

1. Inicialización al azar
2. Se muestran muchos ejemplos con las salidas esperadas y en cada caso se calcula: $y(n) = \phi(\langle \mathbf{w}(n), \mathbf{x}(n) \rangle)$
   - **SI** $y(n) = y_d(n) \Rightarrow$ *no se hacen cambios*
   - **SI** $y(n) \neq y_d(n)$ (supongamos $\mathbf{x}_i(n) > 0$)

Perceptrón simple: aprendizaje

Entrenamiento por corrección de error:

1. Inicialización al azar
2. Se muestran muchos ejemplos con las salidas esperadas y en cada caso se calcula: $y(n) = \phi(\langle \mathbf{w}(n), \mathbf{x}(n) \rangle)$
   - **SI** $y(n) = y_d(n) \Rightarrow$ *no se hacen cambios*
   - **SI** $y(n) \neq y_d(n)$ (supongamos $\mathbf{x}_i(n) > 0$)
     - **SI** $y(n) = +1$ mientras $y_d(n) = -1$
       $\Rightarrow \mathbf{w}(n+1) = \mathbf{w}(n) - \eta \mathbf{x}(n)$

Perceptrón simple: aprendizaje

Entrenamiento por corrección de error:

1. Inicialización al azar
2. Se muestran muchos ejemplos con las salidas esperadas y en cada caso se calcula: $y(n) = \phi(\langle \mathbf{w}(n), \mathbf{x}(n) \rangle)$
   - **SI** $y(n) = y_d(n) \Rightarrow$ *no se hacen cambios*
   - **SI** $y(n) \neq y_d(n)$ (supongamos $\mathbf{x}_i(n) > 0$)
     - **SI** $y(n) = +1$ mientras $y_d(n) = -1$ $\Rightarrow \mathbf{w}(n+1) = \mathbf{w}(n) - \eta \mathbf{x}(n)$
     - **SI** $y(n) = -1$ mientras $y_d(n) = +1$ $\Rightarrow \mathbf{w}(n+1) = \mathbf{w}(n) + \eta \mathbf{x}(n)$

Perceptrón simple: aprendizaje

Algoritmo del perceptrón simple:

1. Inicialización al azar: $\mathbf{w}(1)\in$[-0.5 0.5]
2. Para cada ejemplo de entrenamiento $\mathbf{x}(n)|y_d(n)$:
   - Se obtiene la salida:
     $$y(n) = \phi(\langle \mathbf{w}(n), \mathbf{x}(n)\rangle)$$
   - Se adaptan los pesos:
     $$\mathbf{w}(n+1) = ?$$
3. Volver a 2 hasta satisfacer algún criterio de finalización.

Perceptrón simple: aprendizaje

Algoritmo del perceptrón simple:

1. Inicialización al azar: $\mathbf{w}(1) \in$[-0.5 0.5]
2. Para cada ejemplo de entrenamiento $\mathbf{x}(n)|y_d(n)$:
   - Se obtiene la salida:
     $$y(n) = \phi(\langle \mathbf{w}(n), \mathbf{x}(n) \rangle)$$
   - Se adaptan los pesos:
     $$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\eta}{2}[y_d(n) - y(n)]\mathbf{x}(n)$$
3. Volver a 2 hasta satisfacer algún criterio de finalización.

Métodos de gradiente: perceptrón simple

Diego Milone

Inteligencia Computacional
Departamento de Informática

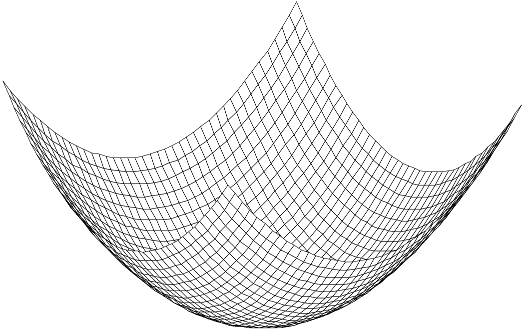
FICH-UNL

Entrenamiento por métodos de gradiente

- Concepto:
  Mover los pesos en la dirección en que se reduce el error, dirección que es opuesta a su gradiente con respecto a los pesos

Entrenamiento por métodos de gradiente

- Concepto:
  Mover los pesos en la dirección en que se reduce el error, dirección que es opuesta a su gradiente con respecto a los pesos
- Interpretación gráfica



Entrenamiento por métodos de gradiente

- Concepto:
  Mover los pesos en la dirección en que se reduce el error, dirección que es opuesta a su gradiente con respecto a los pesos
- Interpretación gráfica
- Ecuación básica:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_w \xi(\mathbf{w}(n))$$

Entrenamiento por métodos de gradiente

- Concepto:
  Mover los pesos en la dirección en que se reduce el error, dirección que es opuesta a su gradiente con respecto a los pesos
- Interpretación gráfica
- Ecuación básica:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_w \xi(\mathbf{w}(n))$$

- Aplicación:
  - Caso sencillo: perceptrón simple (least mean squares)
  - Caso general: perceptrón multicapa (back-propagation)

Perceptrón simple: método de gradiente (caso lineal)

Criterio del error instantáneo:

$$e^2(n) = [d(n) - y(n)]^2$$

Perceptrón simple: método de gradiente (caso lineal)

Criterio del error instantáneo:

$$e^2(n) = [d(n) - y(n)]^2 = [d(n) - \langle \mathbf{w}(n), \mathbf{x}(n) \rangle]^2$$

Perceptrón simple: método de gradiente (caso lineal)

Criterio del error instantáneo:

$$e^2(n) = [d(n) - y(n)]^2 = [d(n) - \langle \mathbf{w}(n), \mathbf{x}(n) \rangle]^2$$

$$\nabla_w e^2(n) = 2\left[d(n) - \langle \mathbf{w}(n), \mathbf{x}(n) \rangle\right](-\mathbf{x}(n))$$

Perceptrón simple: método de gradiente (caso lineal)

Criterio del error instantáneo:

$$e^2(n) = [d(n) - y(n)]^2 = [d(n) - \langle \mathbf{w}(n), \mathbf{x}(n) \rangle]^2$$

$$\nabla_w e^2(n) = 2\,[d(n) - \langle \mathbf{w}(n), \mathbf{x}(n) \rangle]\,(-\mathbf{x}(n))$$

$$\nabla_w e^2(n) = 2e(n)(-\mathbf{x}(n))$$

Perceptrón simple: método de gradiente (caso lineal)

Criterio del error instantáneo:

$$e^2(n) = [d(n) - y(n)]^2 = [d(n) - \langle \mathbf{w}(n), \mathbf{x}(n) \rangle]^2$$

$$\nabla_w e^2(n) = 2\,[d(n) - \langle \mathbf{w}(n), \mathbf{x}(n) \rangle]\,(-\mathbf{x}(n))$$

$$\nabla_w e^2(n) = 2e(n)(-\mathbf{x}(n))$$

reemplazando en: $\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_w \xi(\mathbf{w}(n))$

Perceptrón simple: método de gradiente (caso lineal)

Criterio del error instantáneo:

$$e^2(n) = [d(n) - y(n)]^2 = [d(n) - \langle \mathbf{w}(n), \mathbf{x}(n) \rangle]^2$$

$$\nabla_w e^2(n) = 2\,[d(n) - \langle \mathbf{w}(n), \mathbf{x}(n) \rangle]\,(-\mathbf{x}(n))$$

$$\nabla_w e^2(n) = 2e(n)(-\mathbf{x}(n))$$

reemplazando en: $\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_w \xi(\mathbf{w}(n))$

$$\boxed{\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n)}$$

Perceptrón simple: método de gradiente

Supongamos que:

$$\begin{array}{lll} w_0 = +1 & x_0 = -1 & d = -1 \\ w_1 = +1 & x_1 = +1 & \\ w_2 = +1 & x_2 = +1 & \end{array}$$

Perceptrón simple: método de gradiente

Supongamos que:
$$\begin{array}{lll} w_0 = +1 & x_0 = -1 & d = -1 \\ w_1 = +1 & x_1 = +1 & \\ w_2 = +1 & x_2 = +1 & \end{array}$$

$y = \mathrm{sgn}(-1 + 1 + 1) \rightarrow 1$

Perceptrón simple: método de gradiente

Supongamos que:
$$\begin{array}{lll} w_0 = +1 & x_0 = -1 & d = -1 \\ w_1 = +1 & x_1 = +1 & \\ w_2 = +1 & x_2 = +1 & \end{array}$$

$$y = \mathrm{sgn}(-1 + 1 + 1) \rightarrow 1$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n)$$

Perceptrón simple: método de gradiente

Supongamos que:
$$\begin{array}{lll} w_0 = +1 & x_0 = -1 & d = -1 \\ w_1 = +1 & x_1 = +1 & \\ w_2 = +1 & x_2 = +1 & \end{array}$$

$$y = \operatorname{sgn}(-1 + 1 + 1) \to 1$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n)$$

$$\mathbf{w}(n+1) = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} + 2\mu(-1-1)\begin{bmatrix} -1 \\ +1 \\ +1 \end{bmatrix}$$

Perceptrón simple: método de gradiente

Supongamos que:
$$\begin{array}{lll} w_0 = +1 & x_0 = -1 & d = -1 \\ w_1 = +1 & x_1 = +1 & \\ w_2 = +1 & x_2 = +1 & \end{array}$$

$$y = \mathrm{sgn}(-1 + 1 + 1) \rightarrow 1$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n)$$

$$\mathbf{w}(n+1) = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} + 2\mu(-1-1)\begin{bmatrix} -1 \\ +1 \\ +1 \end{bmatrix}$$

suponiendo que $\mu = \frac{1}{2}$

Perceptrón simple: método de gradiente

$$\mathbf{w}(n+1) = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} + \begin{bmatrix} +2 \\ -2 \\ -2 \end{bmatrix} = \begin{bmatrix} +3 \\ -1 \\ -1 \end{bmatrix}$$

Perceptrón simple: método de gradiente

$$\mathbf{w}(n+1) = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} + \begin{bmatrix} +2 \\ -2 \\ -2 \end{bmatrix} = \begin{bmatrix} +3 \\ -1 \\ -1 \end{bmatrix}$$

Qué sucede si volvemos a poner: $\mathbf{x}(n+1) = \begin{bmatrix} -1 \\ +1 \\ +1 \end{bmatrix}$?

Perceptrón simple: método de gradiente

$$\mathbf{w}(n+1) = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} + \begin{bmatrix} +2 \\ -2 \\ -2 \end{bmatrix} = \begin{bmatrix} +3 \\ -1 \\ -1 \end{bmatrix}$$

Qué sucede si volvemos a poner: $\mathbf{x}(n+1) = \begin{bmatrix} -1 \\ +1 \\ +1 \end{bmatrix}$?

$$y = \mathrm{sgn}(-3 - 1 - 1) \rightarrow -1$$

Perceptrón simple: método de gradiente

$$\mathbf{w}(n+1) = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} + \begin{bmatrix} +2 \\ -2 \\ -2 \end{bmatrix} = \begin{bmatrix} +3 \\ -1 \\ -1 \end{bmatrix}$$

Qué sucede si volvemos a poner: $\mathbf{x}(n+1) = \begin{bmatrix} -1 \\ +1 \\ +1 \end{bmatrix}$?

$$y = \mathrm{sgn}(-3 - 1 - 1) \rightarrow -1$$

$$\boxed{\text{ahora } e(n+1) = -1 - (-1) = 0}$$

El perceptón simple puede resolver este problema?