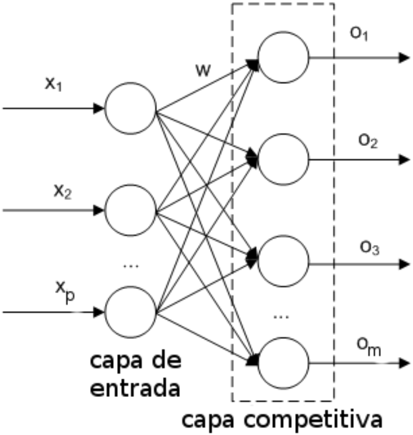
Redes Neuronales Artificiales Competitivas No Supervisadas

Carleos Artime, C.; Corral Blanco, N.

13 de mayo de 2021

Red Neuronal Competitiva No Supervisada

- Las neuronas de salida compiten entre ellas para activarse.
- Sólo se activa la que alcanza un mayor *potencial sináptico*.
- Las neuronas se especializan en detectar la presencia de ciertos patrones de entrada.



Los elementos básicos de estas redes son:

- La capa de entrada. Vector de entrada: $\vec{x} = (x_1, \ldots, x_p)$
- La capa de salida, que tiene naturaleza competitiva:
  - Sus neuronas son de naturaleza binaria.
  - Sólo pueden activarse una con cada patrón de entrada.
- Potencial sináptico de la neurona de salida k:

$$h_k = \sum_{r=1}^{p} w_{kr} x_r - \frac{1}{2} \sum_{r=1}^{p} w_{kr}^2 = -\frac{1}{2} \|\vec{w}_k - \vec{x}\|^2 + \text{constante}$$

- La neurona con mayor potencial sináptico es la que tiene la menor distancia euclídea entre su vector de pesos $\vec{w}_k$ y $\vec{x}$.
- La salida k vale: $o_k = \begin{cases} 1 & \text{si } h_k = \max\{h_1, \ldots, h_m\} \\ 0 & \text{en otro caso} \end{cases}$

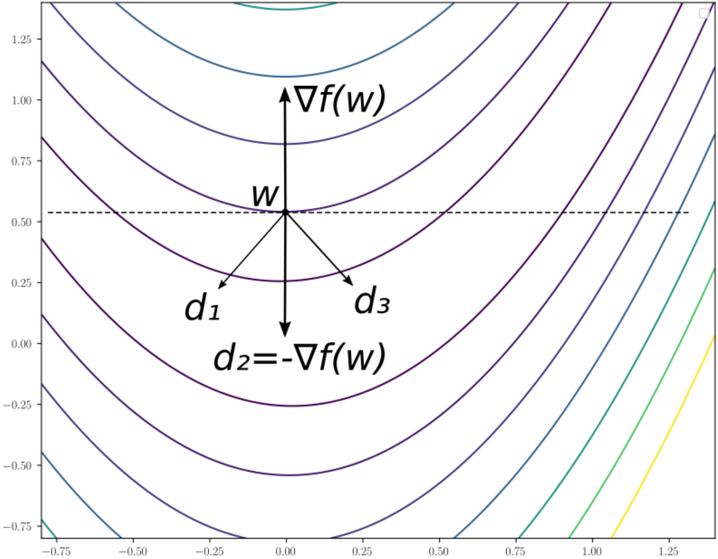
Entrenamiento de la red:

- Objetivo: Conseguir que los pesos sean una buena representación de los datos de entrenamiento.
- Procedimiento: Activar la neurona cuyos pesos sean los más parecidos al patrón de entrada.

Para ello se define el siguiente error:

$$E(\vec{x}) = \sum_{k=1}^{m} a_k \|\vec{x} - \vec{w}_k\|^2 = \sum_{k=1}^{m} a_k \sum_{r=1}^{p} (x_r - w_{kr})^2$$

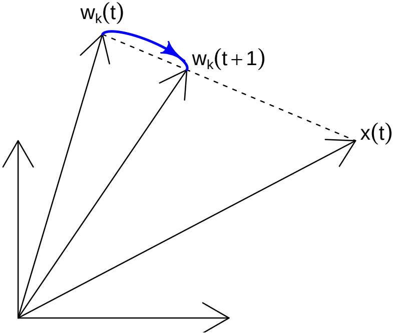
donde $a_k = \begin{cases} 1 & \text{si } \|\vec{x} - \vec{w}_k\|^2 = \min\limits_{i=1,\ldots,m} \|\vec{x} - \vec{w}_i\|^2 \\ 0 & \text{en el resto} \end{cases}$



$\nabla f(w)$
w
d_1
d_3
$d_2 = -\nabla f(w)$
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00 1.25

Los pesos sinápticos se estiman, de manera iterativa, usando el descenso del gradiente para minimizar el error

$$w_{kj}(t+1) = w_{kj}(t) - \eta_t \frac{\partial E}{\partial w_{kj}} = w_{kj}(t) + \eta_t \, a_k \sum_{r=1}^{p} (x_r - w_{kr})$$



Los pesos sinápticos se estiman, de manera iterativa, usando el descenso del gradiente para minimizar el error

$$w_{kj}(t+1) = w_{kj}(t) - \eta_t \frac{\partial E}{\partial w_{kj}} = w_{kj}(t) + \eta_t \, a_k \sum_{r=1}^{p} (x_r - w_{kr})$$

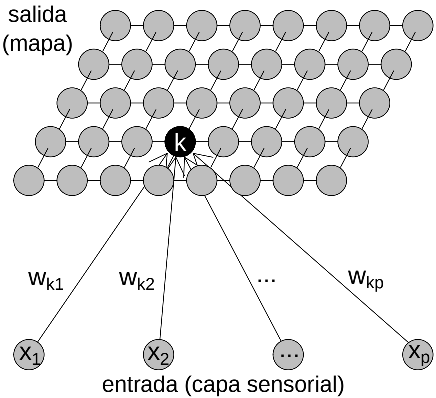
En cada iteración sólo cambian los pesos de la neurona ganadora.

La tasa de aprendizaje debe disminuir con cada iteración del proceso de entrenamiento. Algunos criterios habituales son:

- $\eta_t = \eta_0 e^{-t/\lambda}$
- $\eta_t = 1 - \frac{t}{T}$ con $T =$ número máximo de iteraciones

Mapas de Kohonen (Self Organizing Map, SOM)

- Son redes de aprendizaje no supervisado.
- Tienen naturaleza competitiva.
- Se construyen teniendo en cuenta las posiciones que ocupan las neuronas.



Mapas de Kohonen

Los elementos básicos de una red de Kohonen son:

- La capa de salida está formada por una rejilla rectangular, de neuronas binarias, de tamaño $m_1 \times m_2$. La rejilla también puede ser hexagonal.
- Cada neurona k de la capa de salida tiene asociado un vector de posición en el plano $\vec{p}_k = (p_{k1}, p_{k2})$
- La proximidad entre las neuronas de la capa de salida. La función más usual es

$$T_{g,k} = exp\left(-\frac{d^2(\vec{p}_g, \vec{p}_k)}{2\sigma^2}\right)$$

donde

$$d(p_g, p_k) = \left(|p_{g1} - p_{k1}|^\alpha + |p_{g2} - p_{k2}|^\alpha\right)^{\frac{1}{\alpha}}$$

  - $\alpha = 1 \Longrightarrow d$ es la distancia de Manhattan
  - $\alpha = 2 \Longrightarrow d$ es la distancia euclídea

Mapas de Kohonen

Etapas del algoritmo de una red de Kohonen:

1. Calcular el valor de activación de cada neurona de salida

$$h_k(\vec{x}) = \exp(-d^2(\vec{x}, \vec{w}_k))$$

   Neurona ganadora: vector de pesos más cercano al patrón de entrada.

2. Medir la proximidad de las neuronas respecto a la ganadora g:

$$T_{g,k}(t) = \exp\left(-\frac{d^2(\vec{p}_g, \vec{p}_k)}{2\sigma_t^2}\right)$$

   Influye el radio de vecindad, σ_t, que va decreciendo con cada iteración. Un criterio habitual es $\sigma_t = \sigma_0 \exp(-t/\tau)$.

3. Los pesos se modifican teniendo en cuenta la cercanía a la ganadora

$$w_{kr}(t+1) = w_{kr}(t) + \eta_t T_{g,k}(t)(x_r - w_{kr})$$

Mapas de Kohonen

En el entrenamiento de una red de Kohonen hay dos etapas.

1. Auto-organización: Organización topológica de los vectores de pesos. Inicialmente se consideran una tasa de aprendizaje y radio de vecindad altos. Por ejemplo, $\eta_0 = 1$ y $\sigma_0 =$ diámetro del mapa.
2. Convergencia: Está dirigida a obtener valores estables de los pesos sinápticos. Se suele emplear una tasa de aprendizaje pequeña (0,05) y un radio igual a 1 (distancia mínima entre intersecciones de la rejilla).

En implementaciones simplificadas como la de `class` de R, se usa

$$T_{g,k}(t) = \begin{cases} 1 & d(\vec{p}_g, \vec{p}_k) \leq \sigma_t \\ 0 & d(\vec{p}_g, \vec{p}_k) > \sigma_t \end{cases}$$

Entonces, para $\sigma_t = \sigma$ fijo,

$$\lim_{t \to \infty} \vec{w}_k = \bar{x}_{k,\sigma} = \frac{\sum_g T_{g,k} \sum V_g}{\sum_g T_{g,k} \cdot \# V_g}$$

donde $V_g = \{\vec{x}_i \mid g = \arg\min_j \|\vec{x}_i - \vec{w}_j\|\}$ es el conjunto de patrones de entrada asociados a la neurona de salida g.
Par tanto, los pesos de k tienden a la media de los patrones asociados a las neuronas de salida próximas a k.
Se repite el cálculo para una secuencia $\sigma_1 > \sigma_2 > \cdots > \sigma_n = 0$