



Inteligencia Artificial

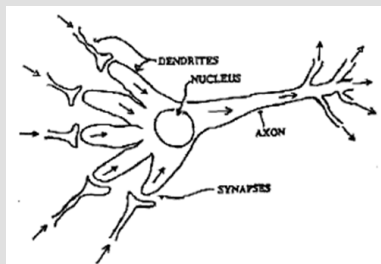


Redes neuronales artificiales.

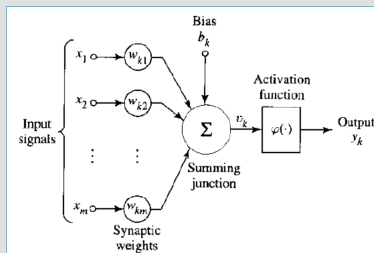
Una red neuronal es una gran cantidad de unidades simples de cómputo interconectadas entre si, que tratan de asemejarse al cerebro en dos aspectos:

1. El conocimiento es adquirido por la red desde el ambiente a través de un proceso de aprendizaje
2. El conocimiento adquirido es almacenado en las conexiones entre neuronas

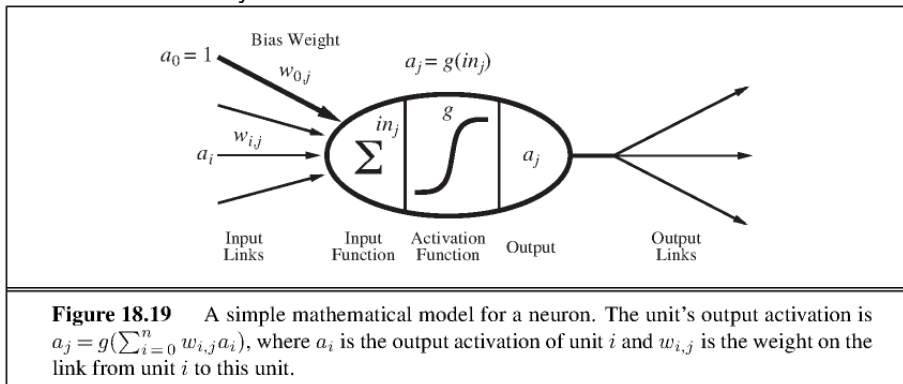
Neurona biológica



Neurona artificial



El modelo de McCulloch y Pitts data de 1943.



Estructura de una red neuronal

- Una red está compuesta por **unidades** (o neuronas) conectadas por **links**.
- Un link tiene asociado un peso y es usado para propagar la activación desde una neurona a otra.
- La activación de una neurona a_j se calcula:

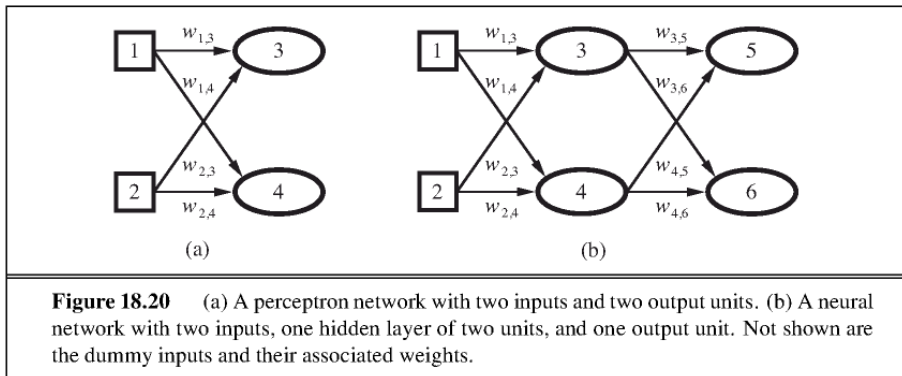
$$a_j = g(in_j) = g\left(\sum_{i=0}^n w_{i,j}a_i\right)$$

Donde:

- g es la función de activación
- w es el peso sináptico
- a_i es la entrada relacionada

- Si g es la función threshold, la neurona se denomina **perceptron**
- Hay redes **propagación hacia adelante** y **recurrentes**.

- Las redes **feed-forward** se pueden organizar en **capas** o **layers**.
- Si la red es multicapa, las capas que no se conectan con la salida se denominan **ocultas**.



Redes de una capa.

- Si hay m salidas, entonces hay m redes distintas, cada una con su propio entrenamiento
- Cada neurona es un clasificador lineal

| x_1 | x_2 | y_3 (carry) | y_4 (sum) |
|-------|-------|---------------|-------------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

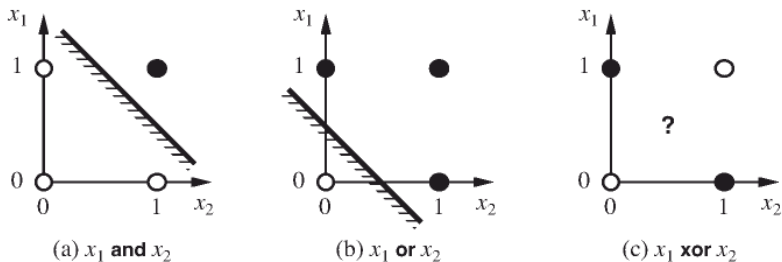
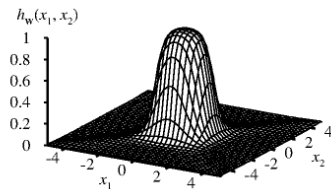
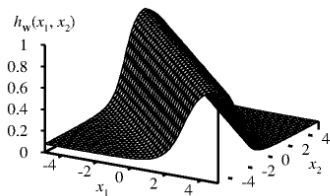


Figure 18.21 Linear separability in threshold perceptrons. Black dots indicate a point in the input space where the value of the function is 1, and white dots indicate a point where the value is 0. The perceptron returns 1 on the region on the non-shaded side of the line. In (c), no such line exists that correctly classifies the inputs.

Redes multicapa. MLP

Mediante este tipo de redes es posible representar cualquier tipo de función, incluso funciones no lineales.



Las salidas de una capa son entradas a otra capa.

$$\begin{aligned}
 a_5 &= g(w_{0,5} + w_{3,5} a_3 + w_{4,5} a_4) \\
 &= g(w_{0,5} + w_{3,5} g(w_{0,3} + w_{1,3} a_1 + w_{2,3} a_2) + w_{4,5} g(w_{0,4} + w_{1,4} a_1 + w_{2,4} a_2)) \\
 &= g(w_{0,5} + w_{3,5} g(w_{0,3} + w_{1,3} x_1 + w_{2,3} x_2) + w_{4,5} g(w_{0,4} + w_{1,4} x_1 + w_{2,4} x_2)).
 \end{aligned}$$

Entrenamiento del MLP. back propagation

El valor de las salidas dependen de **todos** los pesos de entrada

$$\frac{\partial}{\partial w} Loss(\mathbf{w}) = \frac{\partial}{\partial w} |\mathbf{y} - \mathbf{h}_w(\mathbf{x})|^2 = \frac{\partial}{\partial w} \sum_k (y_k - a_k)^2 = \sum_k \frac{\partial}{\partial w} (y_k - a_k)^2 \quad (18.10)$$

Otro inconveniente es que el error en las salidas debe **retro propagarse** a la capa oculta. La idea básica es dividirlo de acuerdo a los pesos en la capa oculta.

Regla para la capa de salida:

$$\begin{aligned} \frac{\partial Loss_k}{\partial w_{j,k}} &= -2(y_k - a_k) \frac{\partial a_k}{\partial w_{j,k}} = -2(y_k - a_k) \frac{\partial g(in_k)}{\partial w_{j,k}} \\ &= -2(y_k - a_k) g'(in_k) \frac{\partial in_k}{\partial w_{j,k}} = -2(y_k - a_k) g'(in_k) \frac{\partial}{\partial w_{j,k}} \left(\sum_j w_{j,k} a_j \right) \\ &= -2(y_k - a_k) g'(in_k) a_j = -a_j \Delta_k, \end{aligned}$$

Regla para la capa oculta:

$$\begin{aligned}
 \frac{\partial Loss_k}{\partial w_{i,j}} &= -2(y_k - a_k) \frac{\partial a_k}{\partial w_{i,j}} = -2(y_k - a_k) \frac{\partial g(in_k)}{\partial w_{i,j}} \\
 &= -2(y_k - a_k) g'(in_k) \frac{\partial in_k}{\partial w_{i,j}} = -2\Delta_k \frac{\partial}{\partial w_{i,j}} \left(\sum_j w_{j,k} a_j \right) \\
 &= -2\Delta_k w_{j,k} \frac{\partial a_j}{\partial w_{i,j}} = -2\Delta_k w_{j,k} \frac{\partial g(in_j)}{\partial w_{i,j}} \\
 &= -2\Delta_k w_{j,k} g'(in_j) \frac{\partial in_j}{\partial w_{i,j}} \\
 &= -2\Delta_k w_{j,k} g'(in_j) \frac{\partial}{\partial w_{i,j}} \left(\sum_i w_{i,j} a_i \right) \\
 &= -2\Delta_k w_{j,k} g'(in_j) a_i = -a_i \Delta_j,
 \end{aligned}$$

Datos adicionales

- Existen otros modelos de red además del MLP:
 - Radial Base Networks
 - Self Organized Maps (SOM)
 - Hopfield
- En problemas reales es necesaria una etapa de preprocesamiento

Bibliografía y enlaces útiles.

- Russell S., Norvig P.: Artificial Intelligence: A modern Approach. Third Edition. Chapter 18.
- Haykin, Simon: Neural Networks. A comprehensive foundation. Second Edition.