

Redes Neuronales Artificiales

Carleos Artime, C.; Corral Blanco, N.

13 de diciembre de 2017

- ▶ Son modelos matemáticos que intentan emular las siguientes características del cerebro.
 - ▶ Procesamiento paralelo: Actúan millones de neuronas
 - ▶ Memoria distribuida: la información está distribuida por toda la red y compartida por muchas neuronas
 - ▶ Adaptabilidad al entorno: Aprender a partir de casos particulares y generalizar conceptos
- ▶ Están formadas por un gran número de nodos (neuronas), interconectados entre sí y organizados en capas.
- ▶ Son capaces de establecer criterios de decisión por sí mismos, a partir de criterios generales de actuación.

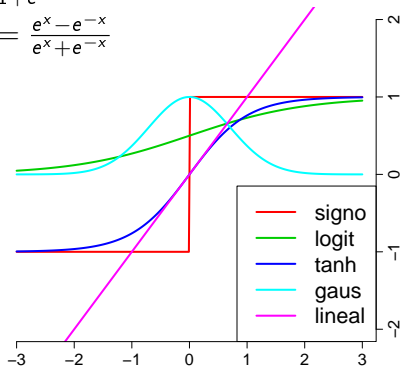
ELEMENTOS BÁSICOS DE UNA RED NEURONAL ARTIFICIAL

- ▶ Conjunto de neuronas
- ▶ Estructura de las conexiones entre las neuronas y sus ponderaciones
- ▶ Regla de propagación: cómo sintetiza cada neurona la información que recibe (p.ej. suma ponderada)
- ▶ Función de activación o transferencia (salida de la neurona)
- ▶ Regla de aprendizaje que utiliza la red

Redes Neuronales artificiales

Función de transferencia:

- ▶ Signo (umbral): $f(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}$
- ▶ Sigmoide (logística): $f(x) = \frac{1}{1+e^{-x}}$
- ▶ Tangente hiperbólica: $\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- ▶ Gaussiana: $f(x) = e^{-x^2}$
- ▶ Identidad: $f(x) = x$



Clasificación de las redes neuronales

- ▶ La disposición de las conexiones
 - ▶ Hacia adelante (feedforward)
La información se propaga exclusivamente en la dirección entrada→salida.
 - ▶ Hacia adelante y hacia atrás (feedforward/feedback)
La información fluye tanto hacia adelante como hacia atrás, pudiendo existir incluso conexiones laterales.
- ▶ La topología
 - ▶ Perceptrones multicapa (MLP Multi-Layer Perceptron)
 - ▶ Funciones de Base Radial (RBF Radial Base Functions)
- ▶ El tipo de aprendizaje
 - ▶ Supervisado
 - ▶ No supervisado

Tipos de aprendizaje supervisado

- ▶ Por corrección de error: Se buscan los pesos para hacer mínimo el error en las predicciones.
- ▶ Por refuerzo: Sólo se conoce si la salida obtenida en la red es acertada o errónea y se ajustan los pesos basándose en un mecanismo de probabilidades.
- ▶ Estocástico: Consiste en introducir cambios aleatorios en los pesos de la red para minimizar el error.

Redes según el aprendizaje No Supervisado

- ▶ Redes auto-organizadas no supervisadas
- ▶ Mapas auto-organizados de Kohonen (Self Organizing Maps. SOM)

Redes Neuronales artificiales

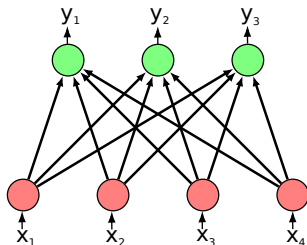
EL PERCEPTRÓN SIMPLE (Rosenblatt, 1962)

Tiene sólo una capa de entrada y otra de salida.

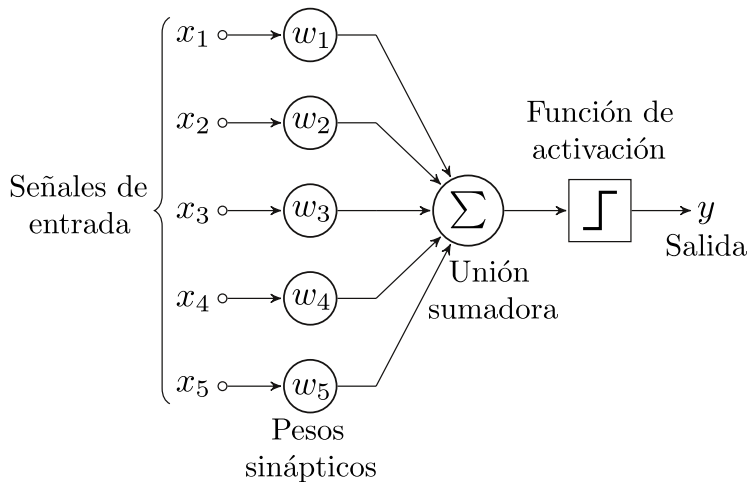
Problema linealmente separable \implies solución tras un número finito de iteraciones

Se dispone de N elementos de entrenamiento (\vec{x}, \vec{y}) , donde

- ▶ $\vec{x} = (x_1, \dots, x_p)$ son las variables predictoras
- ▶ $\vec{y} = (y_1, \dots, y_m)$ las variables objetivo



Redes Neuronales artificiales



El perceptrón simple trata de predecir las variable objetivo utilizando expresiones del tipo:

$$o_k(x) = g \left(\sum_{r=1}^p w_{rk} x_r + \theta_k \right) = g \left(\sum_{r=1}^{p+1} w_{rk} x_r \right)$$

Los pesos w_{rk} se estiman iterativamente usando la regla delta:

$$w_{rk}(t+1) = w_{rk}(t) + \nabla w_{rk} = w_{rk}(t) + \eta(y_k - o_k)x_r; \quad t = 1, \dots$$

El algoritmo consta de los siguientes pasos:

1. Se inicializan los pesos; en general valores aleatorios $\in [-1, 1]$
2. Se calculan las funciones objetivo $o_k = g \left(\sum_{r=1}^{p+1} w_{rk} x_r \right)$
3. Se modifican los pesos aplicando la regla delta
4. Si no se cumple el criterio de parada se vuelve al paso 2

EL PERCEPTRÓN MULTICAPA

Está formado por una capa de entrada, **una o varias capas ocultas** y una de salida.

En cada capa las neuronas reciben información de la capa previa, la elaboran y la envían a todas las neuronas de la capa siguiente.

Los patrones de entrenamiento de la red son (\vec{x}, \vec{y}) , donde

- ▶ $\vec{x} = (x_1, \dots, x_p)$ son las variables predictoras
- ▶ $\vec{y} = (y_1, \dots, y_m)$ son las variables objetivo

La estructura de la red es la siguiente:

1. La capa de entrada tiene como neuronas a las variables predictoras \vec{x}

2. La capa oculta está formada por L neuronas

- ▶ La información de entrada en la neurona j -ésima, u_j , vale:

$$u_j = \sum_{r=1}^p v_{rj} x_r + \alpha_j = \sum_{r=1}^{p+1} v_{rj} x_r$$

- ▶ La información s_j de salida de la neurona j -ésima de la capa oculta viene dada por la expresión: $s_j = g_1(u_j)$, $j = 1, \dots, L$

3. La capa de salida tiene m unidades de procesamiento

- ▶ La información de entrada en la neurona k -ésima, h_k , es:

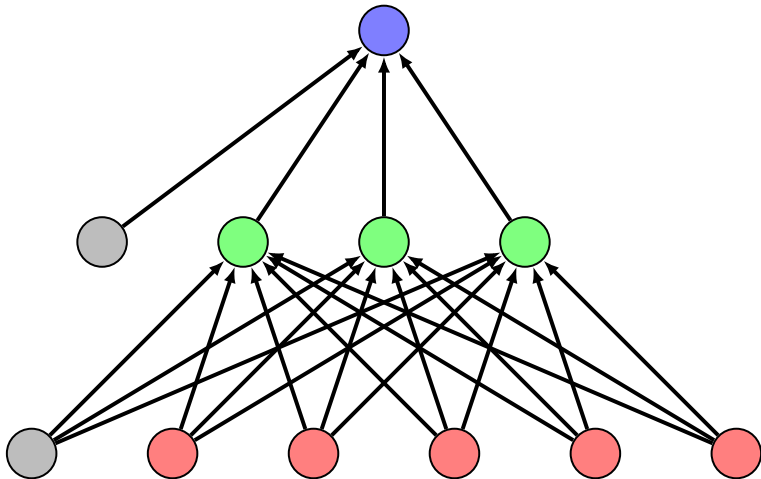
$$h_k = \sum_{j=1}^L w_{jk} s_j + \theta_k = \sum_{j=1}^{L+1} w_{jk} s_j$$

- ▶ La salida de la neurona k -ésima vale:

$$o_k = g_2(h_k)$$

Redes Neuronales artificiales

Estructura de un perceptron con una sola capa oculta



TEOREMA DE REPRESENTACIÓN UNIVERSAL

Sea $\varphi(\cdot)$ una función continua, no constante, acotada y monótona creciente, e I_p el hipercubo unidad, $[0, 1]^p \subset \mathbb{R}^p$, y $C(I_p)$ el espacio de funciones continuas de I_p en \mathbb{R} . Para cualquier función $f \in C(I_p)$ y $\varepsilon > 0$ arbitrario existe un entero m y vectores $\vec{v}, \vec{b} \in \mathbb{R}^m$, $\vec{w} \in \mathbb{R}^p$, tales que la función $F(\vec{x})$ definida por:

$$F(\vec{x}) = \sum_{i=1}^m v_i \varphi(\vec{w}'\vec{x} + b_i)$$

verifica que $|F(\vec{x}) - f(\vec{x})| < \varepsilon \quad \forall \vec{x}$

Para estimar los pesos sinápticos se van a utilizar el algoritmo recursivo de retropropagación y la regla delta generalizada. En cada etapa se intentará reducir el error de predicción:

$$E(\vec{x}) = \frac{1}{2} \sum_{k=1}^m (y_k - o_k(\vec{x}))^2$$

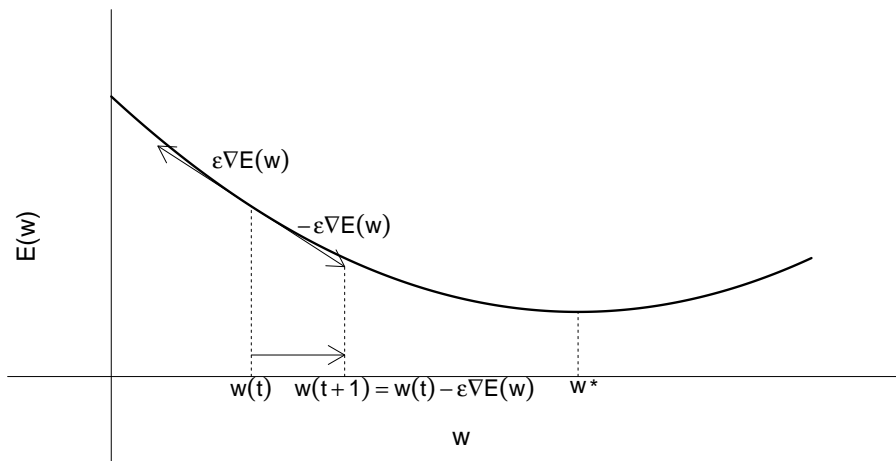
Este algoritmo se basa en el método del descenso por el gradiente. Ajusta los pesos asociados empezando por la capa de salida hasta llegar a la inicial.

Los pesos w_{rk} se estiman iterativamente usando la regla delta generalizada:

$$w_{rk}(t + 1) = w_{rk}(t) + \nabla w_{rk}$$

$$\nabla w_{rk} = -\eta \frac{\partial E}{\partial w_{rk}}$$

Redes Neuronales artificiales



Ajuste de los pesos en la capa de salida:

$$\nabla w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \frac{\partial E}{\partial h_k} \frac{\partial h_k}{\partial w_{jk}}$$

$$\nabla w_{jk} = \eta (y_k - o_k) g_2'(h_k) s_{jk}$$

Ajuste de los pesos en la capa oculta:

$$\begin{aligned} \frac{\partial E}{\partial v_{rj}} &= \frac{\partial}{\partial v_{rj}} \sum_{k=1}^m (y_k - g_2(h_k))^2 \\ &= \sum_{k=1}^m \frac{\partial}{\partial v_{rj}} \left[y_k - g_2 \left(\sum_{j=1}^L w_{jk} g_1(u_j) \right) \right]^2 = \sum_{k=1}^m \frac{\partial E_k}{\partial h_k} \frac{\partial h_k}{\partial s_j} \frac{\partial s_j}{\partial u_j} \frac{\partial u_j}{\partial v_{rj}} \\ \nabla v_{rj} &= \eta \sum_{k=1}^m (y_k - o_k) g_2'(h_k) w_{jk} g_1'(u_j) x_r \end{aligned}$$

Sugerencias para usar redes neuronales

- ▶ Buscar un compromiso entre la complejidad y la calidad de los resultados
- ▶ Utilizar siempre que sea posible un conjunto de entrenamiento y otro de validación. Si se dispone de conjunto de datos relativamente pequeño utilizar validación cruzada varias veces.
- ▶ ¿Cuántas capas ocultas utilizar? No hay nada concluyente, pero se pueden emplear los siguientes criterios:
 - ▶ Una sola capa será suficiente en muchos problemas y se debe probar con distinto número de neuronas analizando la evolución del error de estimación.
 - ▶ Si no se obtiene un resultado satisfactorio, emplear dos capas ocultas y diferentes números de neuronas en esas capas.
 - ▶ Los problemas que requieren más de dos capas ocultas suelen tener una gran complejidad, como por ejemplo la clasificación de imágenes.

Sugerencias para usar redes neuronales

- ▶ ¿Cuántas neuronas, L , se deben emplear en la capa oculta? Hay muchas reglas pero no son muy fiables. Si p es el número de variables de entrada y m el de salida
 - ▶ L debe estar entre m y p
 - ▶ $L = \frac{p+m}{2}$
 - ▶ $L = \sqrt{p m}$
 - ▶ $L \leq \frac{N}{\kappa \cdot (m+p)}$ donde $2 \leq \kappa \leq 10$
- ▶ Hacer el entrenamiento y “podar” los nodos que tengan los pesos muy próximos a cero. En ese caso conviene empezar con más neuronas de las que se puedan necesitar.