REDES NEURONALES

2021

Clase 15 Parte 2

Facultad de Matemática, Astronomía, Física y Computación
Universidad Nacional de Córdoba

Martes 5 de octubre 2021

http://www.famaf.unc.edu.ar/~ftamarit/redes2021

https://www.famaf.unc.edu.ar/course/view.php?id=798

EL APRENDIZAJE SUPERVIZADO

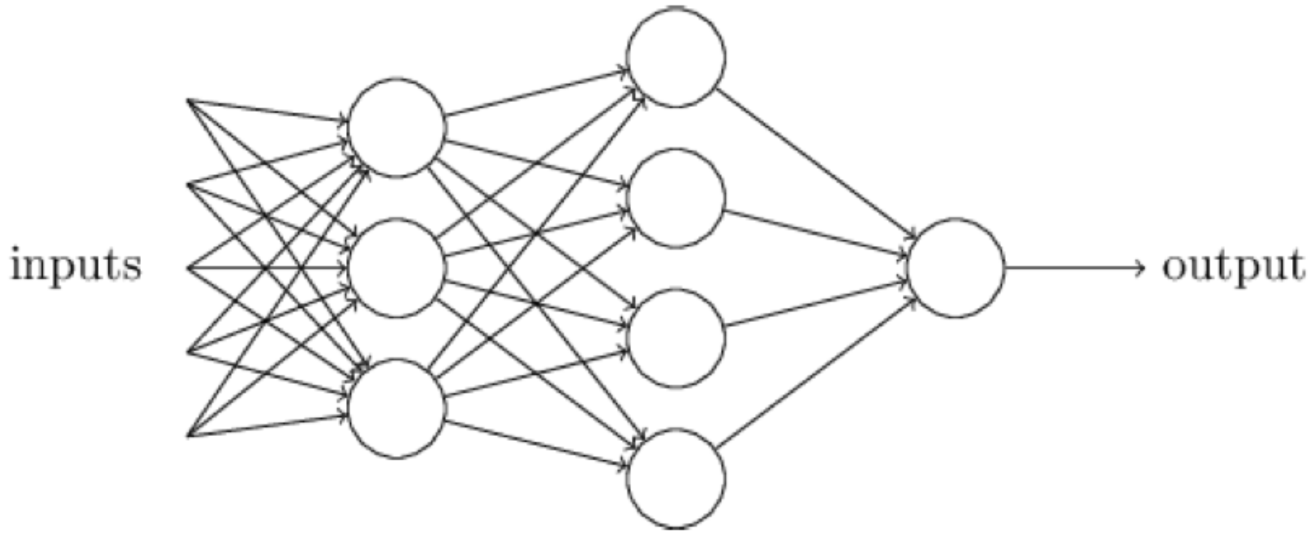
El aprendizaje automático es una técnica del aprendizaje de máquina por medio de la cual podemos estimar el valor de una función arbitraria sin conocer su expresión funcional y a partir de un conjunto de valores conocidos de la función llamado conjunto de entrenamiento.

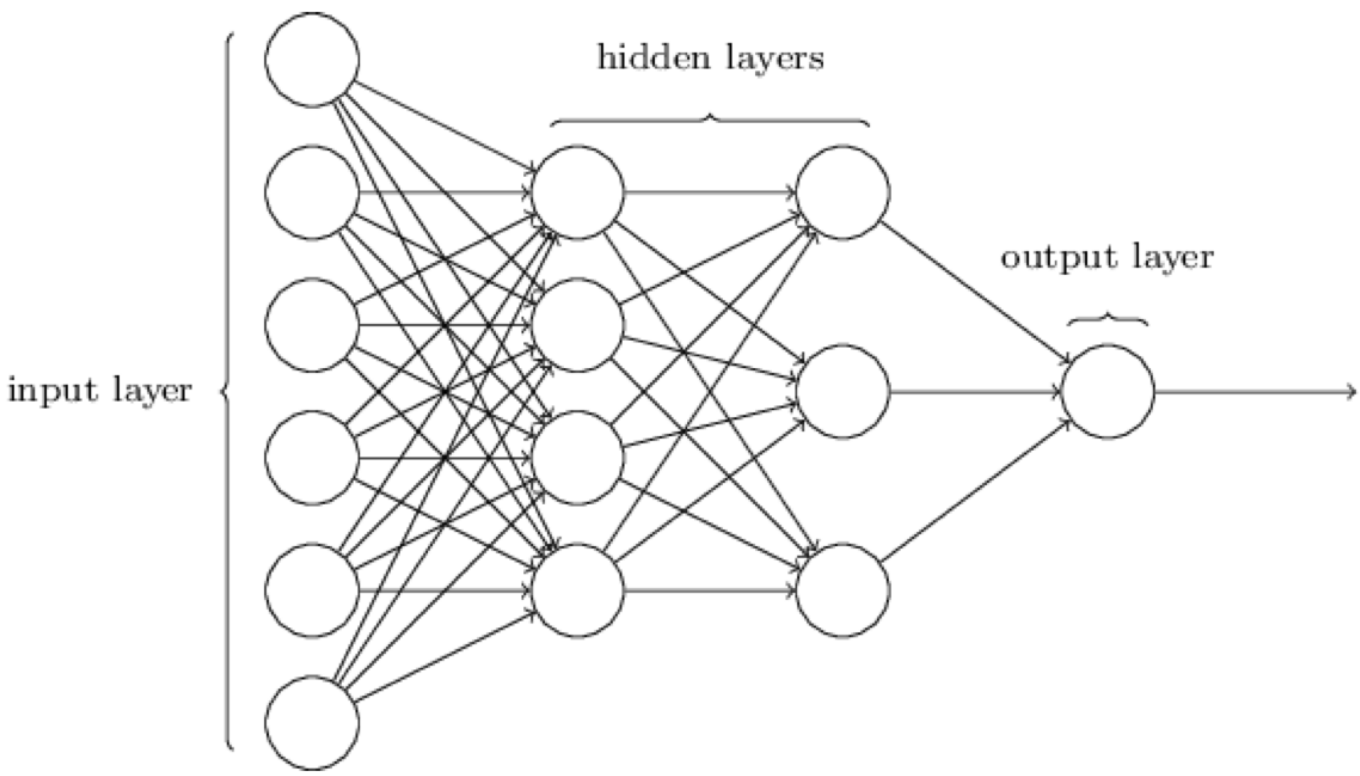
Redes feed-forward

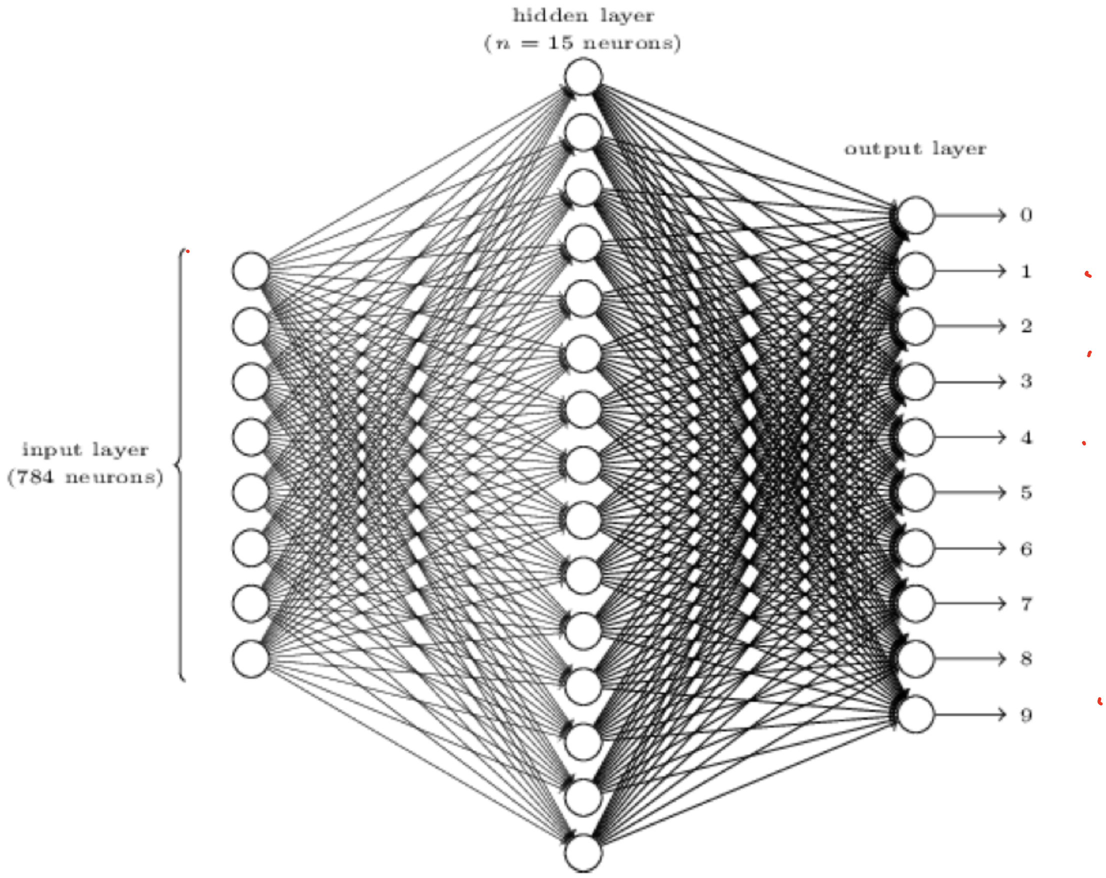
En una red neuronal feed forward las neuronas se organizan en capas sucesivas. La información entra en la llamada capa de entrada, que no es una capa de neuronas. De ahí viaja a la primera capa. Una vez procesada la información por las neuronas de la primera capa, estas se la envían a las neuronas de la segunda capa, hasta llegar a la última capa o capa de neuronas de salida.

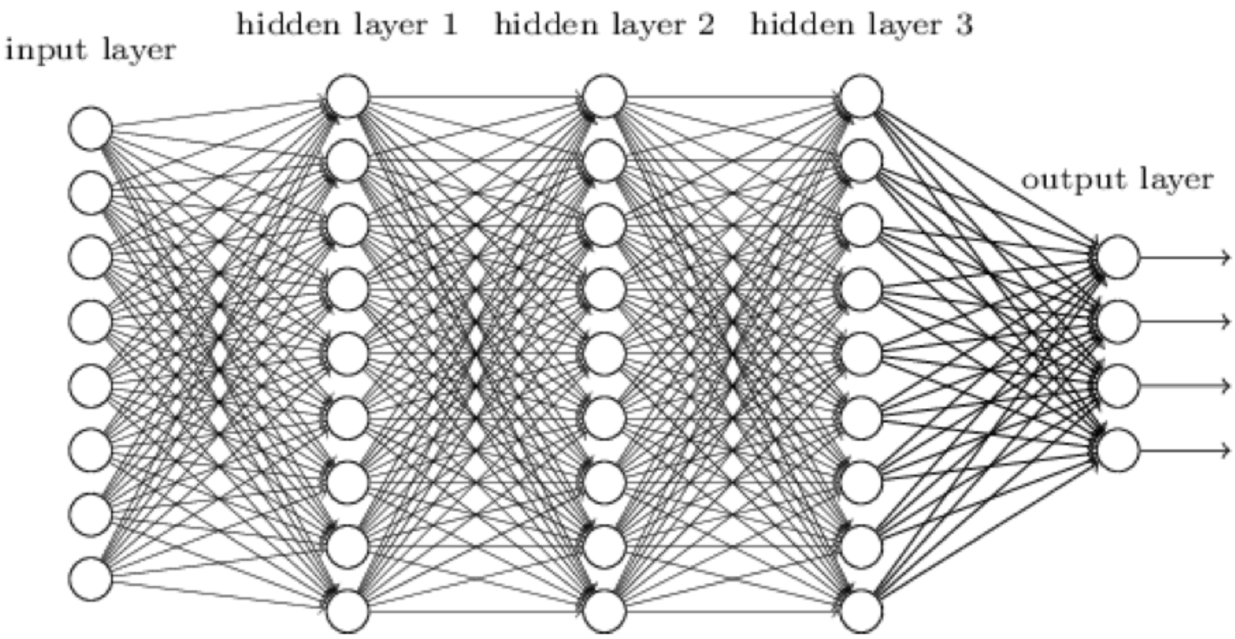
No puede haber intercambio de información dentro de una capa, ni hacia atrás ni saltearse capas hacia adelante. Esta arquitectura tan ordenada facilita la elaboración y la programación de los algoritmos.

Por fuera de las redes feed-forward, la otra arquitectura popular es la llamada arquitectura de redes recurrentes.


inputs
output

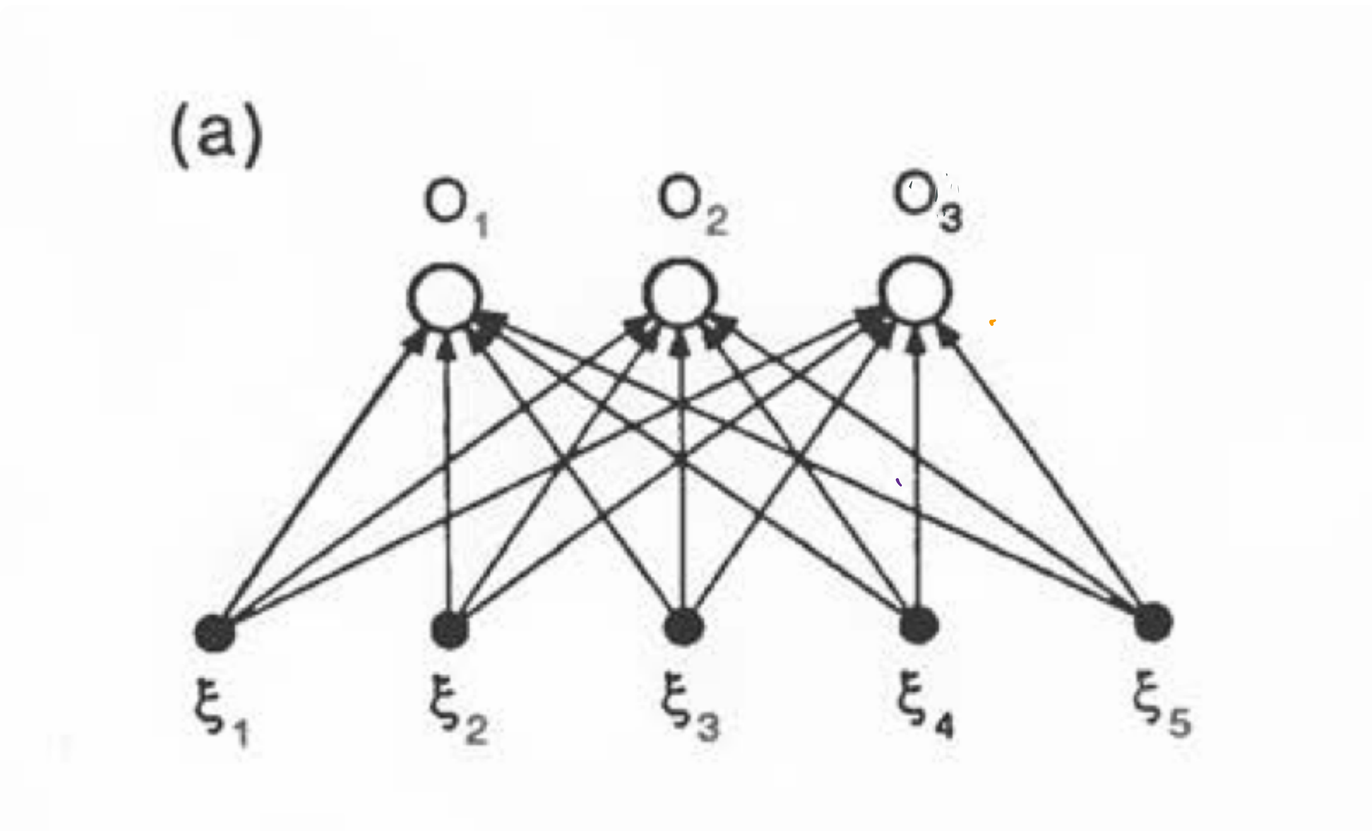

hidden layers
output layer
input layer


hidden layer
(n = 15 neurons)
output layer
input layer
(784 neurons)
0
1
2
3
4
5
6
7
8
9


input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

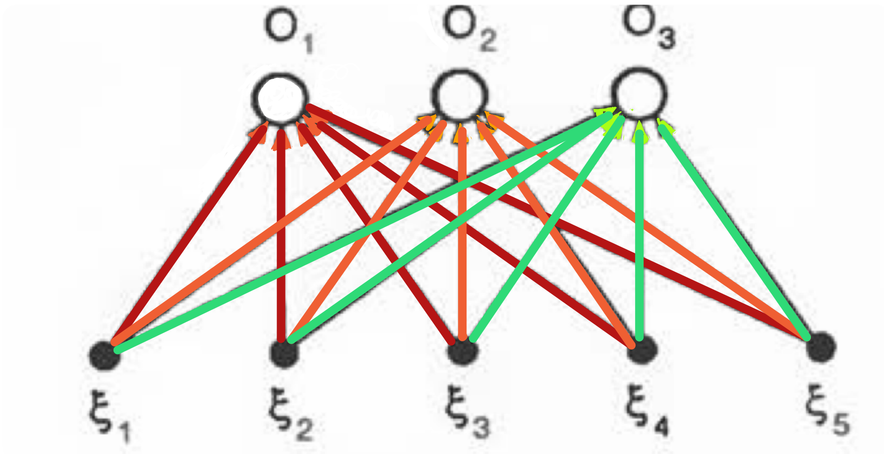
El PERCEPTRON SIMPLE

Consideremos una red muy simple como la de la figura siguiente, con cinco entradas y tres neuronas en su única capa, que es a la vez la capa de salida. Noten que la capa de entrada no es una capa de neuronas pues no procesan nada, solo presentan.



Llamaremos siempre ξ_i a la entradas (features)

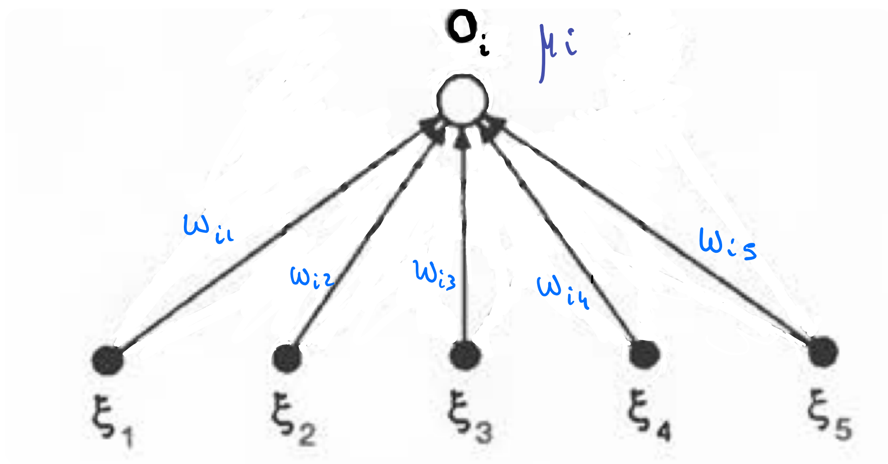
Llamaremos siempre O_i a los estados de las neuronas de salida



Tenemos tres redes, separadas por colores, cada una independiente de las otras, y cada una con una única salida.

Esto nos permite desacoplar toda red feed-forward de una única capa en tantos PERCEPTRONES simples como neuronas en la capa de salida tengamos.

En nuestro caso lo desacoplamos en tres perceptrones simple.



Neurona de salida $\quad O_i$

Entradas o features $\quad \xi_k \quad k = 1, 2, \ldots, N$

El conjunto de variables de entrada forman un vector en

$$\bar{\xi} = (\xi_1, \xi_2, \xi_3, \ldots, \xi_N) \in \mathbb{R}^N$$

Las variables de entrada ξ_i pueden tomar valores discretos o continuos, acotados o no.

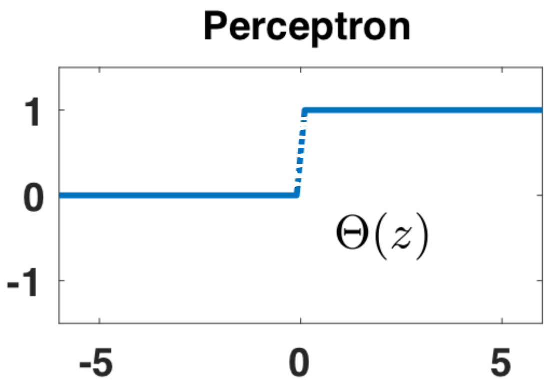
El rango de valores que puede tomar *O* depende exclusivamente de la función de activación *g(x)*.

Si *g(x)* es discreta nuestra red aprenderá a clasificar. Si es continua aprenderá a hacer una regresión. Volveremos sobre este punto muchas veces.

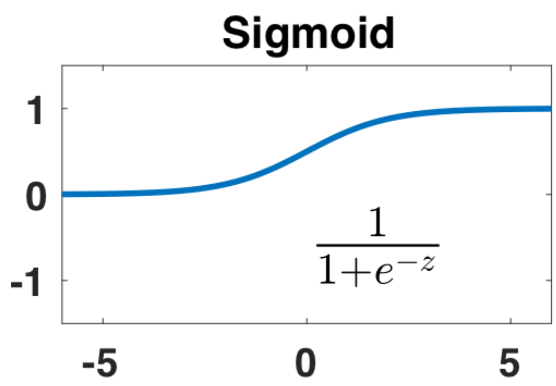
$$O_i = g\left(h_i - \mu_i\right)$$
$$= g\left(\sum_{k=1}^{N} w_{ik}\xi_k - \mu_i\right)$$

w_{ik} : eficacia sináptica entre entrada k y neurona i

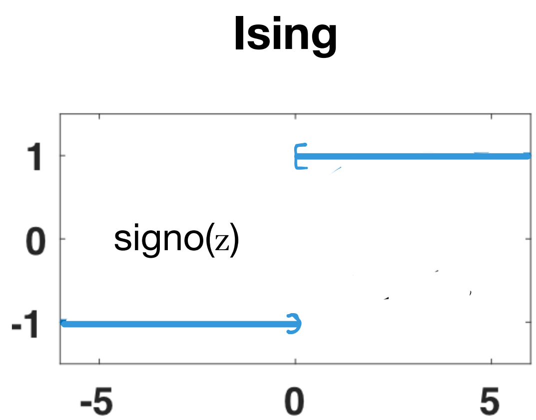
μ_i : umbral de activación de la neurona i



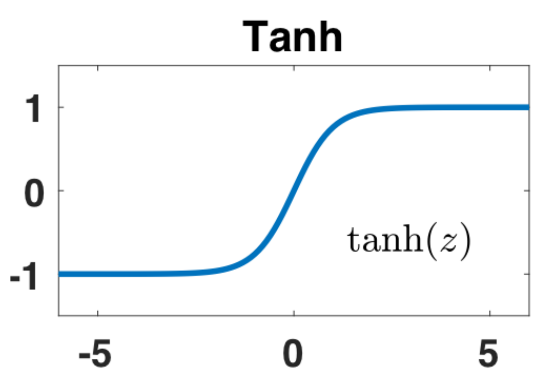
$O_i \in \{0, 1\}$



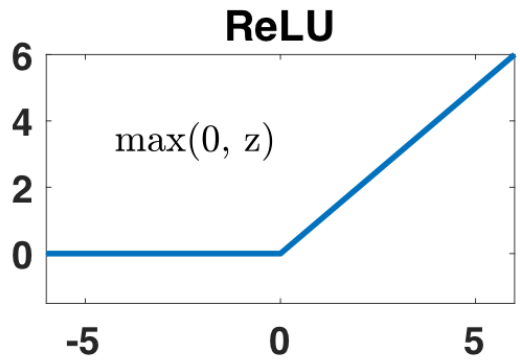
$0 < O_i < 1$



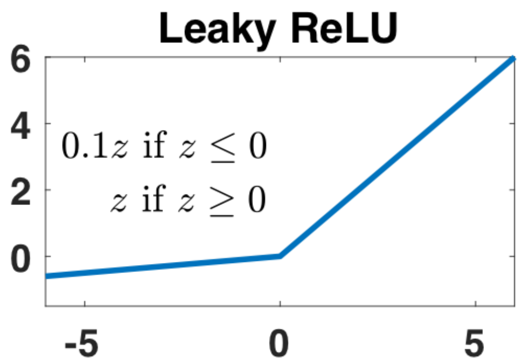
$O_i \in \{-1, +1\}$



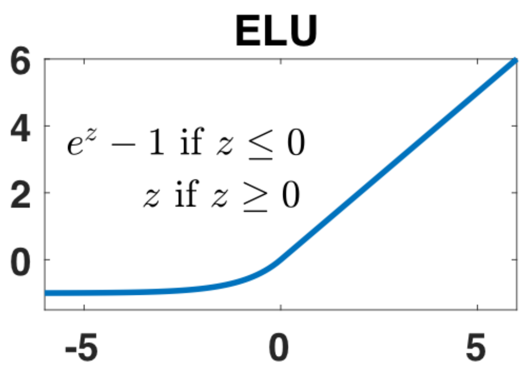
$-1 < O_i < +1$



$0 \le \theta_i < +\infty$



$-\infty < \theta_i < +\infty$



$-1 < \theta_i < +\infty$

Aprendizaje supervisado en un perceptron simple

Supongamos que deseamos modelar cierto proceso que asigna a vectores en un único valor real O_i. No tenemos ninguna información sobre la expresión matemática de dicha asignación, pero podemos imaginarla como en función real que come una N-upla de números reales y devuelve un número real. Llamemos f a esa relación; entonces

$$f(\xi) = O \qquad f : \mathbb{R}^N \longrightarrow \mathbb{R}$$

Supongamos que la única información de la cual disponemos es el valor que está función asigna a cierto conjunto de valores de entrada. Supongamos que son p los datos de entrada cuya solución conozco y los enumeramos con la letra griega α.

$$\xi^\alpha = (\xi_1^\alpha, \xi_2^\alpha, \ldots, \xi_N^\alpha) \xrightarrow[\text{Experto}]{} \zeta_i^\alpha \in \mathbb{R}$$

$$\alpha = 1, 2, 3, \ldots, p$$

$$\left\{ \xi^\alpha, \zeta_i^\alpha \right\}_{\alpha=1}^{p} \longleftarrow$$ Conjunto de entrenamiento

ζ_i^α representa el valor que debería tomar O_i cuando le presentamos $\vec{\xi}_i^\mu$. Pero la neurona de salida tomará el valor

$$O_i^\alpha = g\left(h_i^\alpha - \mu_i\right)$$
$$= g\left(\sum_{k=1}^{N} W_{ik}\, \xi_k^\alpha - \mu_i\right)$$

Nuestro desafío es ahora encontrar un conjunto posible de eficacias sinápticas W_{ik} y un valor del umbral μ_i que sean capaces de hacer que nuestra red asigne correctamente las entradas y las respectivas salidas para el conjunto de entrenamiento.

Si tenemos suerte y si el conjunto de entrenamiento es suficientemente grande y variado, esta red, que aprendió el conjunto de entrenamiento, será capaz de resolver bien incluso los casos que nunca vió. O sea, podrá asignar una salida para una entrada arbitraria.

Esto es equivalente al problema de ajustar los parámetros de una función modelo mediante cuadrados mínimos.

Observemos que podemos pensar a las eficacias sinápticas, que recordamos pueden tomar cualquier valor real, como formando un vector en $\mathbb{R}^N$.

$$\vec{w}_i = (w_{i1}, w_{i2}, \ldots, w_{iN}) \in \mathbb{R}^N$$

Ambos vectores, $\vec{\xi}$ y $\vec{w}_i$ viven en el mismo espacio.

¡Son vecinos!

Si tenemos N entradas o features y M neuronas de salida, podemos aprender una función arbitraria

$$f : \mathbb{R}^N \longrightarrow \mathbb{R}^M$$