



## SISTEMAS DE INTELIGENCIA ARTIFICIAL

INFORME PRELIMINAR N°2

---

# Red neuronal con aprendizaje supervisado

---

Autores:

Pablo Ballesty - 49359

Nicolás Magni - 48008

Guillermo Liss - 49282

12 de abril de 2012

## Resumen

El objetivo del presente documento es detallar el diseño e implementación de una red neuronal utilizando aprendizaje supervisado para resolver las operaciones lógicas AND y OR para  $N$  bits con  $2 \leq N \leq 5$ .

## 1. Desarrollo

### 1.1. Decisiones de implementación

Para ambos problemas se decidió utilizar un perceptrón simple con 6 neuronas en su capa de entrada, y 1 neurona en su capa de salida.

Para la ejecución de uno de los métodos el usuario deberá especificar 3 datos los cuales definimos de la siguiente forma:

- **method** Método a ejecutar *AND* u *OR*.
- **uInput** Vector de tamaño  $N$  con  $2 \leq N \leq 5$ , con valores 0 o 1.
- **tfunction** Función de transferencia “*step*”, “*sigmoid*” o “*lineal*”.

Una vez ingresados estos datos, se normaliza *uInput* agregando un -1 en el primer lugar para el “*bias*”, se convierten los 0 a -1 y en caso que el tamaño de *uInput* sea menor que 5, se completa ingresando un valor que no altere el resultado, es decir, para *AND* 1 y para *OR* -1.

Por ejemplo si el usuario va a ejecutar *AND* con el vector (1,0), el vector de entrada normalizado será (-1,1,-1,1,1,1)

### 1.2. Funciones de transferencia

Se utilizan 3 funciones de transferencia. A continuación se definen las mismas.

$$step(x) = sign(x) \quad (1)$$

$$sigmoid(x) = tanh(1x) \quad (2)$$

$$lineal(x) = x/5 \quad (3)$$

### 1.3. Entrenamiento

Para ambos casos se probaron dos conjuntos de entrenamiento. Uno conformado por todos los 32 patrones posibles de longitud 5, y otro con 24 patrones.

El entrenamiento que se realiza es de tipo iterativo, y consta de los siguientes pasos:

1. Se define el error cuadrático mínimo que se acepta.
2. Se inicializan los pesos de forma aleatoria.
3. Se evalúan todos los patrones realizando de forma iterativa las modificaciones en los pesos.
4. Se calcula el error cuadrático, si éste no resulta por debajo del error mínimo que se acepta, se mezclan los patrones y se vuelve al paso 3.

Las modificaciones en los pesos en el paso 3, se realiza en el sentido de Hebb, de la siguiente manera

$$w_{ik}^{new} = w_{ik}^{old} + \Delta w_{ik}$$

siendo

$$\Delta w_{ik} = \begin{cases} \eta(S_i^\mu - o_i^\mu)\xi_k^\mu & \text{si tfunction es } step \text{ o } lineal \\ \eta(S_i^\mu - o_i^\mu)(1 - (o_i^\mu)^2)\xi_k^\mu & \text{si tfunction es } sigmoid \end{cases}$$

## 2. Resultados

Utilizando los dos conjuntos de entrenamiento antes mencionados, se entrenó a la red con tres valores distintos de  $\eta = 0.01, 0.03, 0.05$ , y con las tres funciones de activación, fijando un error mínimo de 0.01. Se puede ver en los siguientes cuadros los resultados de la cantidad de épocas necesarias para cada caso.

	Función	$\eta = 0.01$	$\eta = 0.03$	$\eta = 0.05$
AND	sigmoid	14751	4883	2910
	step	5	3	3
	lineal	6042	1990	1181
OR	sigmoid	14763	4870	2895
	step	9	2	1
	lineal	6037	1991	1181

	Función	$\eta = 0.01$	$\eta = 0.03$	$\eta = 0.05$
AND	sigmoid	8688	2875	1711
	step	8	4	3
	lineal	3570	1177	697
OR	sigmoid	8638	2843	1697
	step	6	4	2
	lineal	3565	1172	696

Se puede ver que, en general, para  $\eta = 0.05$  se necesitan menos épocas para entrenar la red que con  $\eta = 0.03$  y  $\eta = 0.01$ . Por otro lado se observa que para los casos en que se usaron menos patrones de entrenamiento, la cantidad de épocas necesarias baja (en algunos casos hasta a la mitad) con respecto a la red entrenada con todos los patrones posibles.

## 3. Conclusiones

Se pudo notar que utilizando un conjunto de entrenamiento con menos cantidad de patrones, pero que estos contengan los patrones representativos del método, pudo entrenarse a la red sin problemas, incluso en menos épocas.