



## SISTEMAS DE INTELIGENCIA ARTIFICIAL

INFORME PRELIMINAR N°3

---

# Red neuronal con aprendizaje supervisado

---

Autores:

Pablo Ballesty - 49359

Nicolás Magni - 48008

Guillermo Liss - 49282

## Resumen

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

## 1. Desarrollo

### 1.1. Decisiones de implementación

A diferencia de la entrega anterior, ahora se crean  $N$  redes con  $2 \leq N \leq 5$  para cada método. Luego de entrenar a las redes, los pesos resultantes del aprendizaje quedan persistidos en archivos de texto, que luego se utilizarán para evaluar cada método con la entrada que ingrese el usuario.

### 1.2. Funciones de transferencia

Se utilizan 2 funciones de transferencia. A continuación se definen las mismas

- **sigmoid**

$$g(x) = \tanh(x) \quad (1)$$

$$g'(x) = 1 - \tanh(x)^2 \quad (2)$$

- **lineal**

$$g(x) = x/4 \quad (3)$$

$$g'(x) = 1/4 \quad (4)$$

### 1.3. Entrenamiento

Para entrenar los pesos de las distintas redes, se utiliza *backpropagation* con condición de corte por mínimo error cuadrático aceptable. Al algoritmo de aprendizaje se le realizaron las siguientes mejoras:

- **$\eta$  adaptativo:** Dependiendo de la diferencia entre el error cuadrático de la última época corrida, con el error cuadrático de la anterior, se realizan pequeños cambios sobre  $\eta$ . Si esta diferencia es positiva se disminuye el valor de  $\eta$  en un porcentaje  $b$ , y si ocurren diez diferencias consecutivas negativas se aumenta el valor de  $\eta$  en un valor constante  $a$ . En este caso,  $b = 0.1$  y  $a = 0.01$ . El fin de estos cambios es que cuando el error aumenta, se realicen cambios pequeños sobre los pesos con el objeto de que el error deje de aumentar, mientras que cuando el error disminuye se busca realizar cambios más grandes con el objetivo de acelerar el aprendizaje.
- **Recomienzo:** el entrenamiento permite fijar una cantidad máxima de épocas a correr, y en caso de no llegar a un error cuadrático menor al esperado, se comienza nuevamente, tomando una nueva inicialización aleatoria. El objetivo de esta mejora es evitar que el algoritmo se quede siempre en un mínimo local.

## 2. Resultados

Para probar el comportamiento del algoritmo de aprendizaje se entrenaron distintas arquitecturas hasta cumplir un error cuadrático medio menor a  $err = 0.01$ , para las operaciones lógicas de paridad y simetría, con entradas de  $N = 5$  bits. Se variaron, para la arquitectura de cada operación lógica, el valor  $\alpha = \text{cantidad de neuronas en la capa oculta}$  y  $\eta$  el *learning rate*.

Se puede ver en el apéndice, en los cuadros A.1 y A.2 para las operaciones lógicas de simetría y paridad respectivamente, las comparaciones entre las distintas arquitecturas de redes neuronales.

Todos los resultados corresponden a arquitecturas que solo utilizan la función de transferencia *sigmoid*, ya que utilizando la función lineal no se logra entrenar a la red. Tampoco se utilizó la función *step*, ya que por la naturaleza del algoritmo *backpropagation*, ésta no aplica.

## 3. Conclusiones

Se pudo notar que luego de agregar la mejora de  $\eta$  adaptativo se logra entrenar a las redes con capa de entrada más grande, ya que sin esta mejora solían caer en mínimos locales.

Como se esperaba se lograron entrenar, para las operaciones lógicas de simetría y paridad, redes con arquitecturas  $a = [N \ 2 \ 1]$  y  $a = [N \ N \ 1]$  respectivamente, siendo  $N$  la cantidad de unidades de entrada, y donde cada elemento  $a_i$  es la cantidad de unidades en la capa  $i$ .

Se pudo ver también, la importancia que tiene una buena inicialización de los pesos de la red para lograr entrenarla de forma aceptable, es decir, cumplir el error esperado. Es por esto que en varios casos la mejora de *recomienzo* fue necesaria para alcanzar el objetivo planteado.

## A. Tabla de resultados

### A.1. Simetría

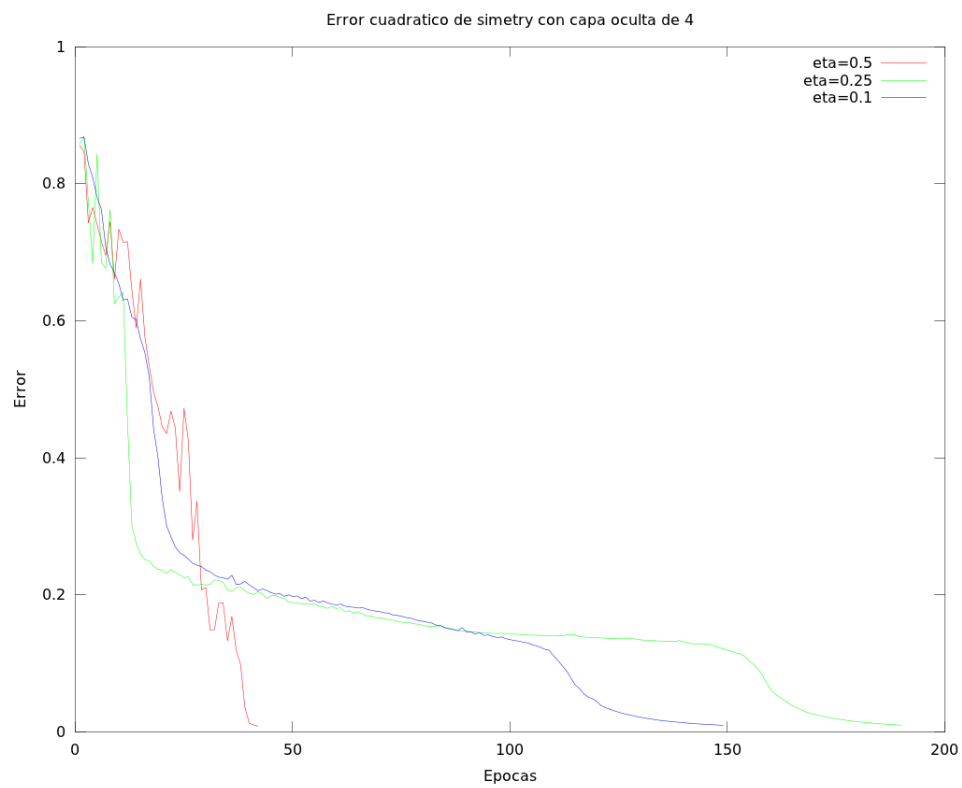
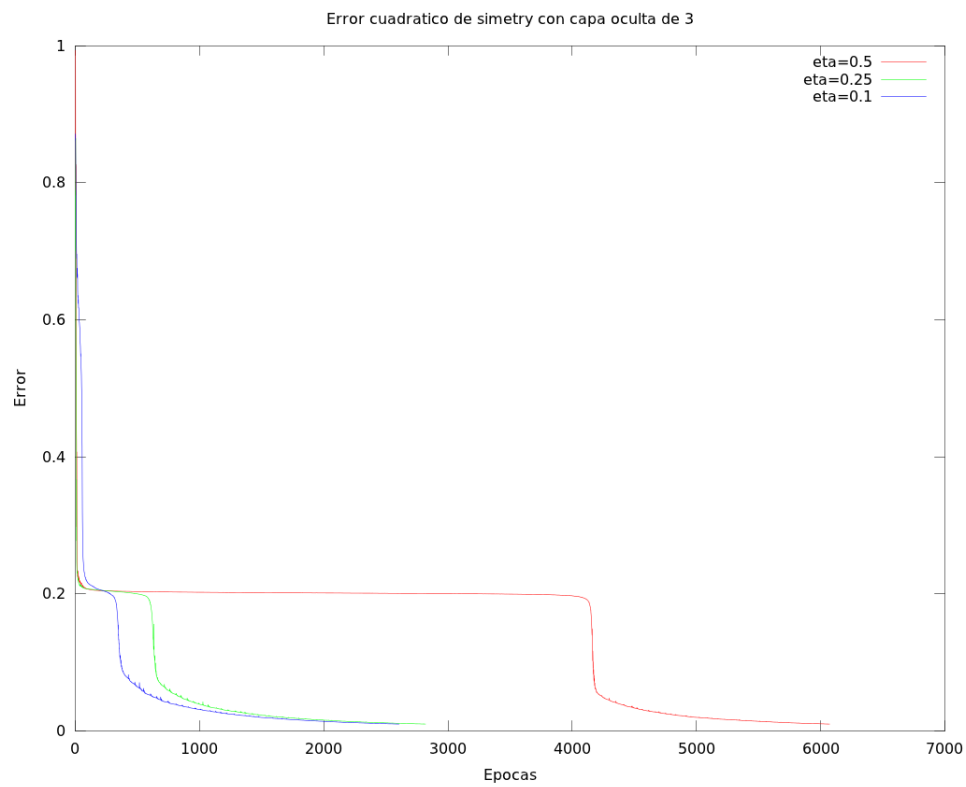
$\alpha$	$\eta$	# épocas
2	0.5	21
	0.25	4628
	0.1	$\infty$
3	0.5	6072
	0.25	2819
	0.1	2606
4	0.5	42
	0.25	190
	0.1	149

### A.2. Paridad

$\alpha$	$\eta$	# épocas
5	0.5	92
	0.25	263
	0.1	206
6	0.5	10835
	0.25	7686
	0.1	3353
7	0.5	118
	0.25	17011
	0.1	3272

## B. Gráficos

### B.1. Simetría



## B.2. Paridad

