

Perceptrón simple y multicapa

Santiago Terenziani
Eugenio Damm

Contenido

Ej. 01

Perceptrón simple

AND - XOR

Ej. 02

Perceptrón simple
lineal y no lineal

Conjuntos de
entrenamiento

Ej. 03

Perceptrón
multicapa

XOR - EVEN/ODD

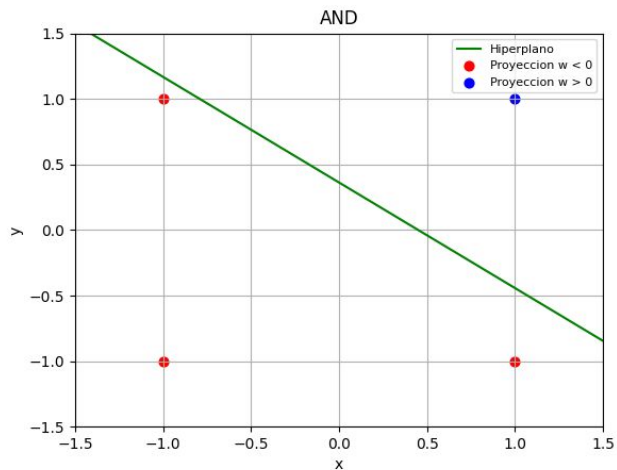


01.

Perceptrón simple

Perceptrón Simple

AND



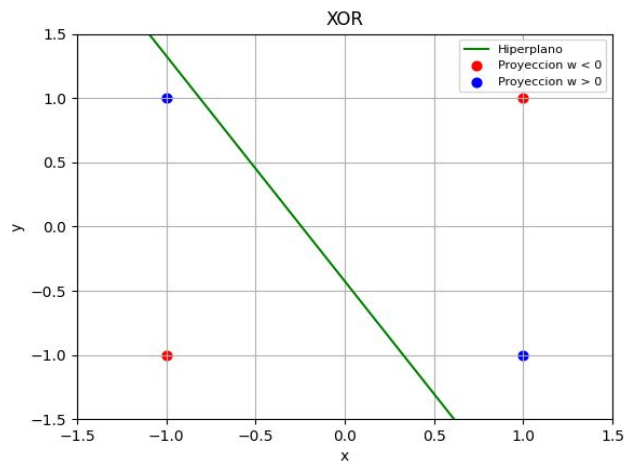
CONFIGURACIONES PROBADAS

	Aprendizaje inicial	Épocas límite	Aprendizaje Adaptativo	Aprendizaje final	Épocas para resolución
1	0,01	100	✗	0,01	17
2	0,01	100	✓	~0,0098	19
3	0,05	100	✓	0,05	7

* Todas las configuraciones se ejecutaron 10 veces, y se extrajo el promedio

Perceptrón Simple

XOR



CONFIGURACIONES PROBADAS

	Aprendizaje inicial	Épocas límite	Aprendizaje Adaptativo	Aprendizaje final	Épocas para resolución
1	0,01	100	✗	0,01	100
2	0,01	100	✓	~0,0091	100
3	0,05	100	✓	~0,0456	100

* Todas las configuraciones se ejecutaron 10 veces, y se extrajo el promedio



01.

¿Qué podemos decir acerca de los problemas que puede resolver el perceptrón simple escalón en relación a la resolución de los problemas que se le pidió que haga que el perceptrón aprenda?



02.

Perceptrón simple lineal y no lineal

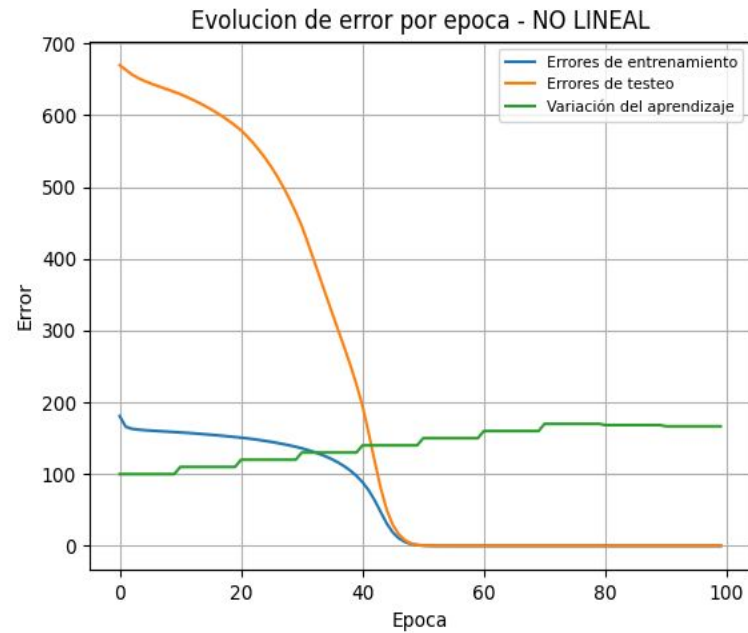
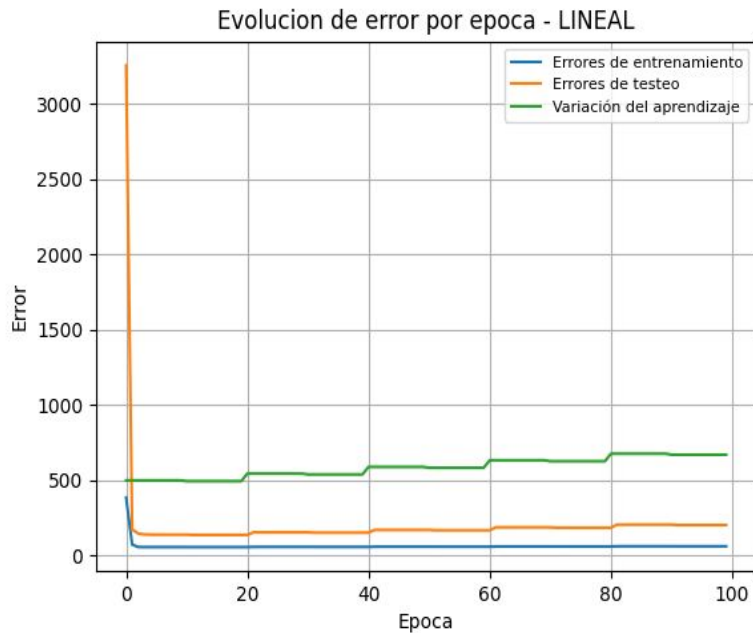
Perceptrón Simple Lineal y No Lineal

CONFIGURACIONES PRBADAS (entrenamiento)

	Aprendizaje inicial	Épocas límite	Aprendizaje Adaptativo	Beta para NL	Aprendizaje final L	Aprendizaje final NL
1	0,01	100	✗	0,5	0,01	0,01
2	0,01	100	✓	0,5	~0,0129	~0,0173
3	0,05	100	✓	0,5	~0,0456	~0,053
4	0,01	100	✓	1	~0,0134	~0,0147

* Todas las configuraciones se ejecutaron 10 veces, y se extrajo el promedio

Perceptrón Simple Lineal y No Lineal



02.

¿Cómo podríamos escoger el mejor conjunto de entrenamiento?

¿Cómo podríamos evaluar la máxima capacidad de generalización del perceptrón para este conjunto de datos?



03.

Perceptrón multicapa

Perceptrón Multicapa

XOR

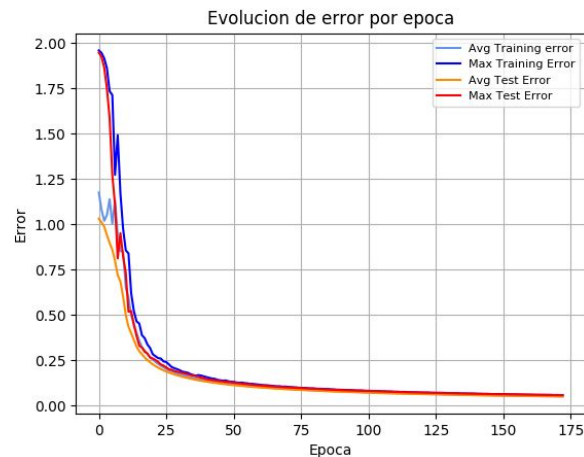
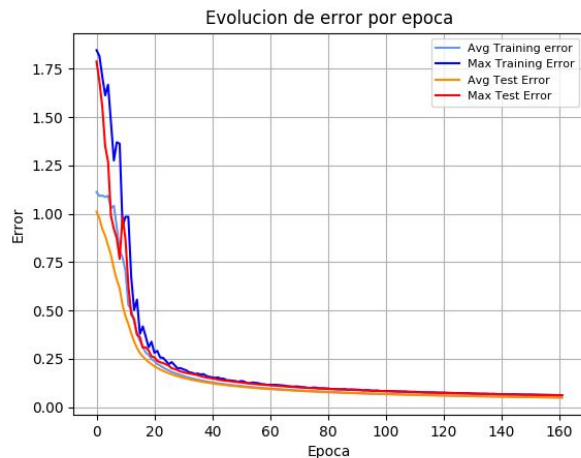
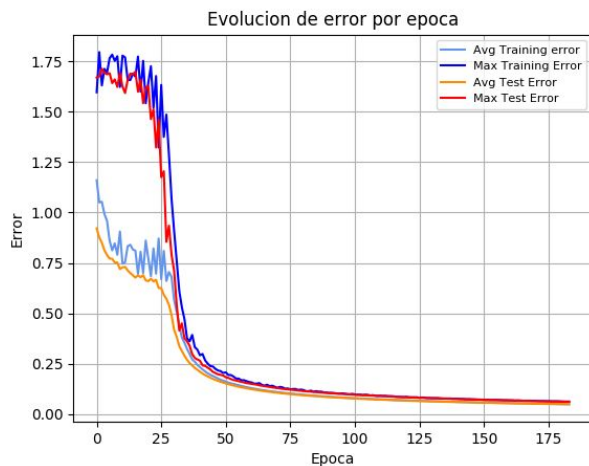
CONFIGURACIONES PROBADAS

	Aprendizaje inicial	Épocas límite	Cota de error	Aprendizaje Adaptativo	Aprendizaje final	Accuracy	Épocas para resolución
1	0,01	1000	0.1	✗	0.01	100%	353
2	0,01	1000	0.1	✓	0.026	100%	270
3	0.01	1000	0.05	✓	0.045	100%	438
4	0.1	1000	0.05	✓	0.11	100%	115

* Todas las configuraciones se ejecutaron 10 veces, y se extrajo el promedio. El margen de accuracy fue del 0.15 y la Beta de la función $\tanh(x)$ fue 2

Perceptrón Multicapa - Error

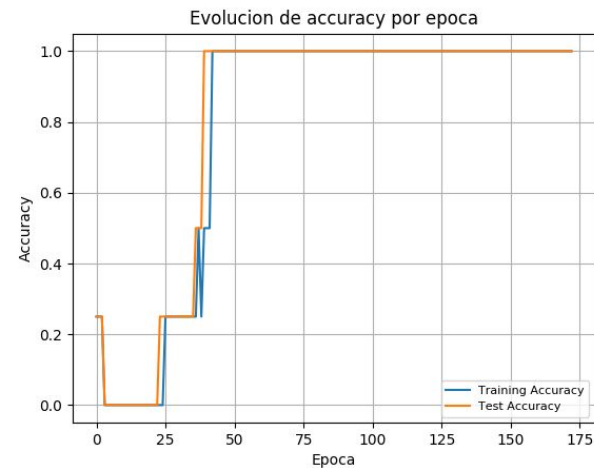
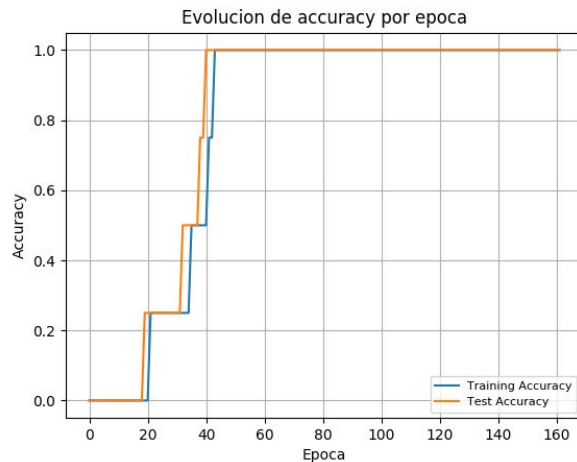
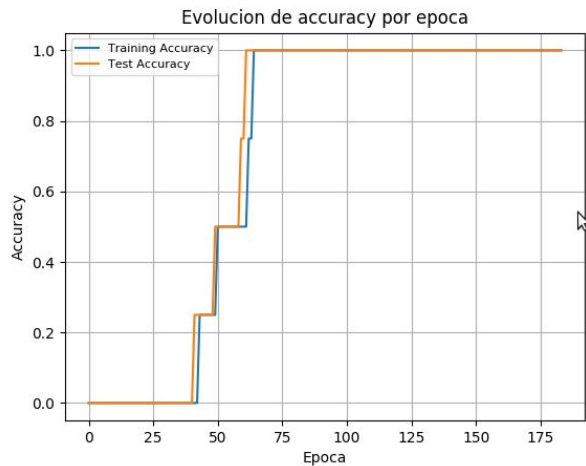
XOR



* Configuraciones: Tasa de Aprendizaje = 0.05, Aprendizaje Adaptativo Activado, Beta = 2, Épocas máximas = 1000, Cota de Error = 0.05, Margen de Clasificación = 0.15

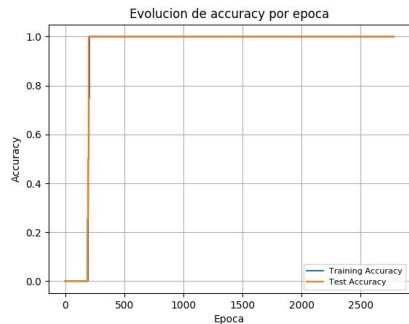
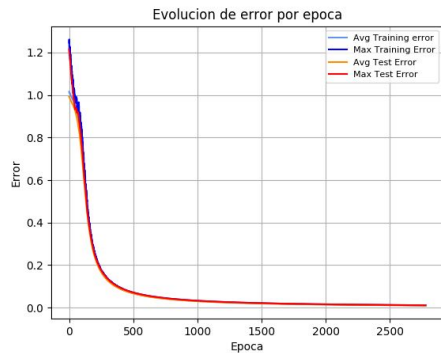
Perceptrón Multicapa - Accuracy

XOR

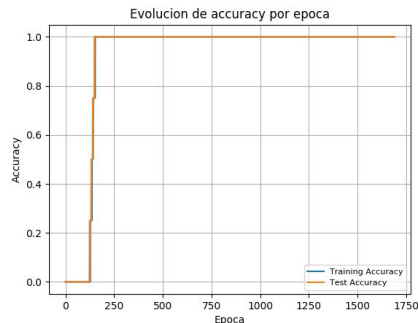
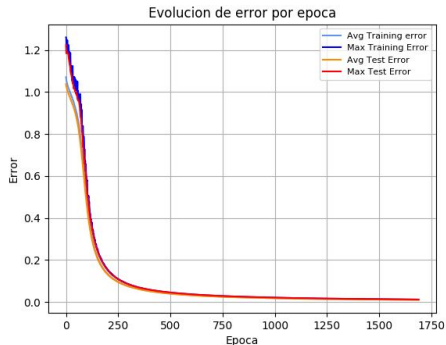


* Configuraciones: Tasa de Aprendizaje = 0.05, Aprendizaje Adaptativo Activado, Beta = 2, Épocas máximas = 1000, Cota de Error = 0.05, Margen de Clasificación = 0.15

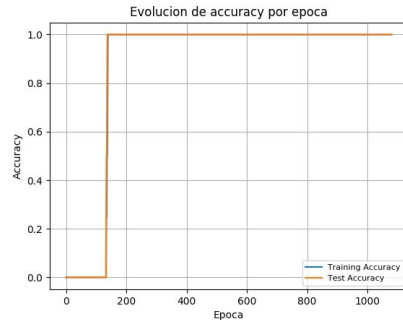
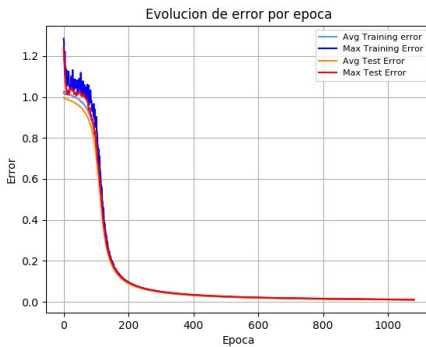
Perceptrón Multicapa - Cantidad de Capas XOR



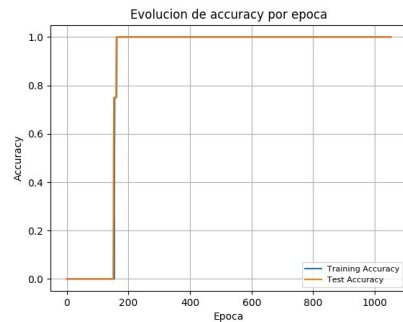
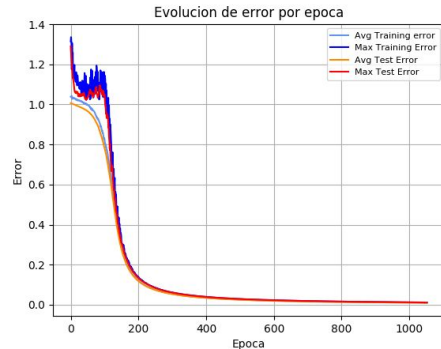
1 CAPA OCULTA
2900 épocas
Preciso a las 250



3 CAPAS OCULTAS
1650 épocas
Preciso a las 225



5 CAPAS OCULTAS
1150 épocas
Preciso a las 200



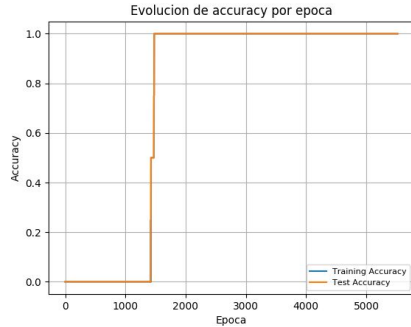
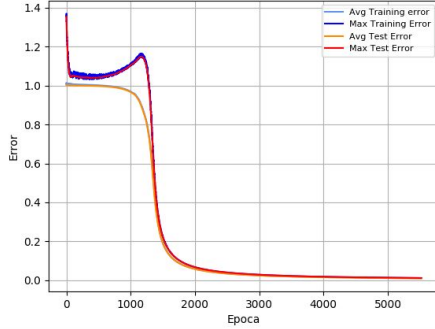
7 CAPAS OCULTAS
1100 épocas
Preciso a las 180

* Configuraciones: Tasa de Aprendizaje = 0.01, Aprendizaje Adaptativo Activado, Beta = 1, Épocas máximas = 10000, Cota de Error = 0.01, Márgen de Clasificación = 0.25
Nodos por capa = 10

Perceptrón Multicapa - Nodos por Capa

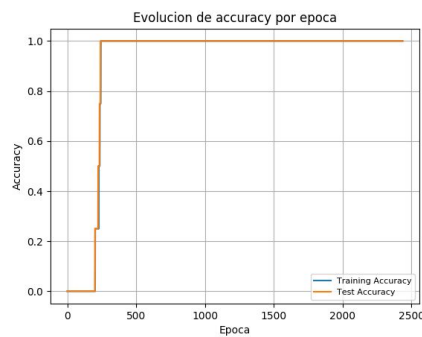
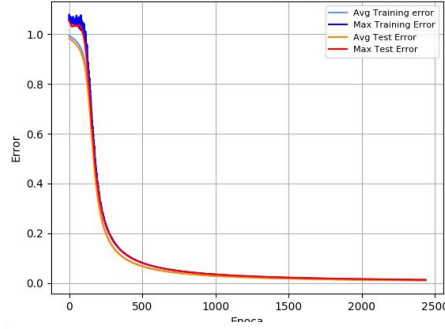
XOR

Evolucion de error por epoca



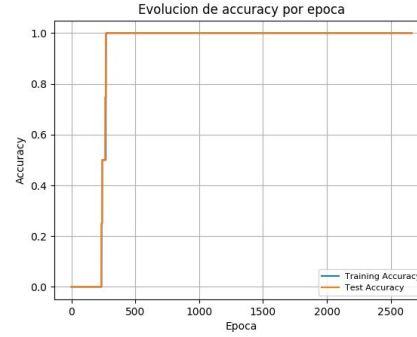
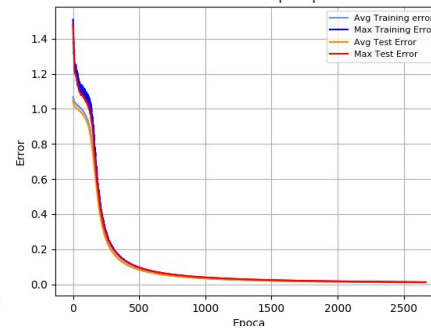
3 NODOS
3000 épocas
Preciso a las 1000

Evolucion de error por epoca



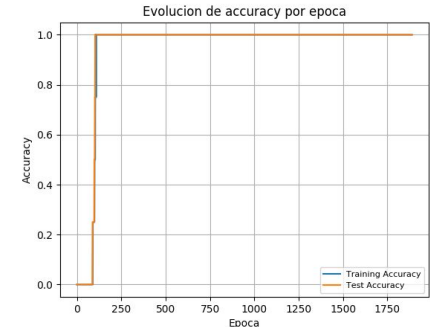
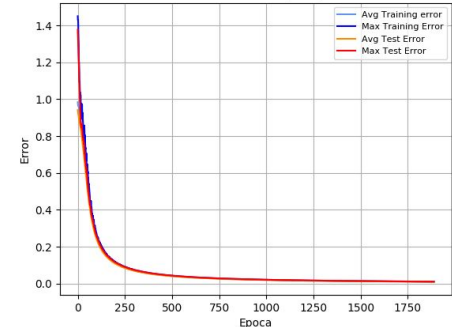
6 NODOS
2500 épocas
Preciso a las 400

Evolucion de error por epoca



10 NODOS
2250 épocas
Preciso a las 250

Evolucion de error por epoca



20 NODOS
1850 épocas
Preciso a las 125

* Configuraciones: Tasa de Aprendizaje = 0.01, Aprendizaje Adaptativo Activado, Beta = 1, Épocas máximas = 10000, Cota de Error = 0.01, Márgen de Clasificación = 0.25
Nodos por capa = 10

Perceptrón Multicapa

EVEN/ODD

CONFIGURACIONES PROBADAS

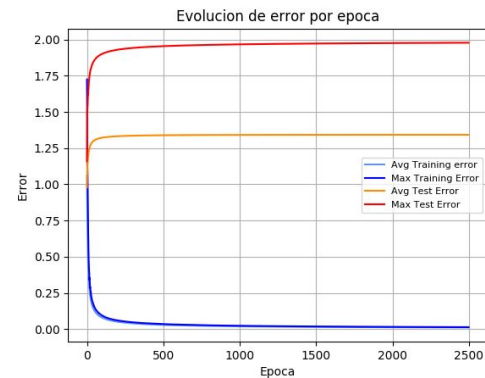
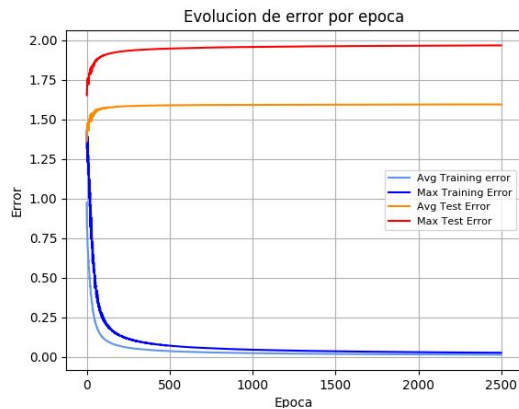
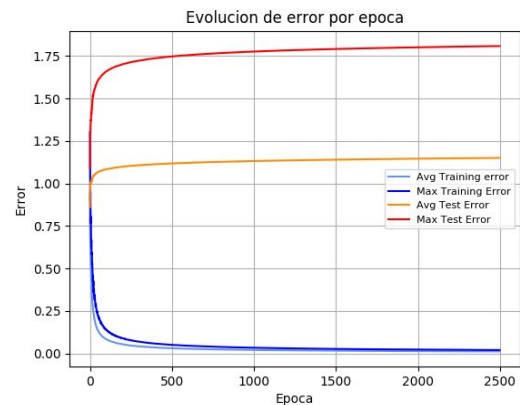
	Aprendizaje inicial	Épocas límite	Cota de error	Aprendizaje Adaptativo	Aprendizaje final	Accuracy	Épocas para resolución
1	0,01	1000	0.05	✗	0.01	32%	136 (rango 6-813)
2	0,01	1000	0.05	✓	0.014	26%	204 (rango 25-1000)
3	0.01	1000	0.01	✓	0.053	26%	545 (rango 284-1000)

* Todas las configuraciones se ejecutaron 10 veces, y se extrajo el promedio. El margen de accuracy fue del 0.15, el Beta de la función $\tanh(x)$ fue 2 y se usó un subconjunto de 7 elementos aleatorios. El perceptrón tiene una única capa oculta y 4 nodos en la misma (uno de ellos siendo el bias)

Perceptrón Multicapa - Error

EVEN/ODD

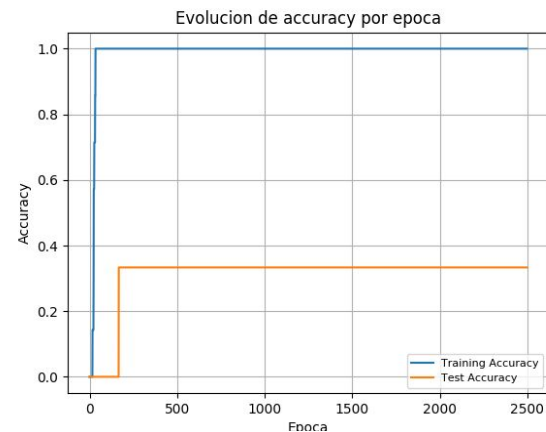
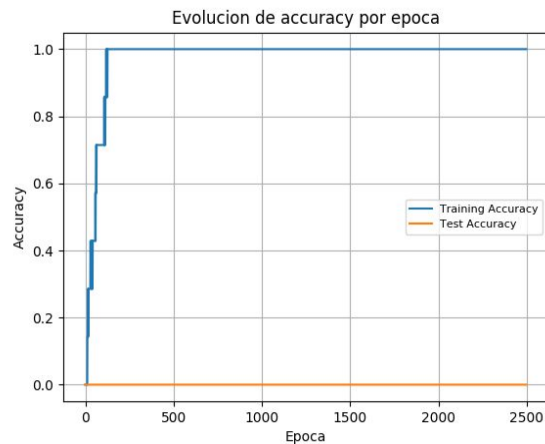
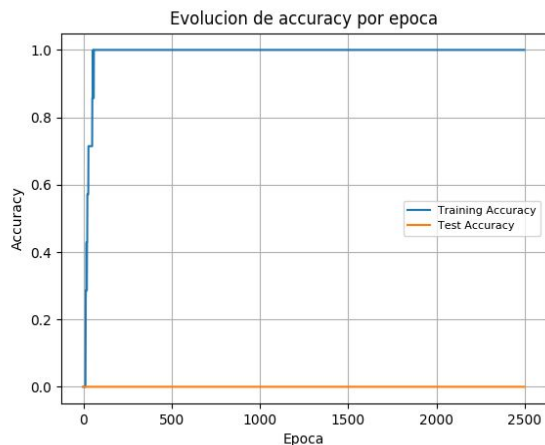
CONFIGURACIONES PROBADAS



* Configuraciones: Tasa de Aprendizaje = 0.05, Aprendizaje Adaptativo Desactivado, Beta = 2, Épocas máximas = 2500, Cota de Error = 0, Márgen de Clasificación = 0.2

Perceptrón Multicapa - Accuracy

EVEN/ODD



* Configuraciones: Tasa de Aprendizaje = 0.05, Aprendizaje Adaptativo Desactivado, Beta = 2, Épocas máximas = 2500, Cota de Error = 0, Margen de Clasificación = 0.2



03.

**¿Qué podríamos decir
acerca de la capacidad
para generalizar de la
red?**

Configuración y ejecución

Archivo input.json de configuración

```
{ } input.json ×
{ } input.json > ...
1 {
2   "PERCEPTRON": "SIMPLE",
3   "FUNCTION": "Ej2",
4   "LEARNING_RATE": 0.01,
5   "ADAPTIVE_LEARNING_RATE": "TRUE",
6   "BETA": 2,
7   "EPOCHS": 100,
8   "ERROR_TOLERANCE": 0,
9   "CLASSIFICATION_MARGIN": 0.15
10 }
```

Ejecutamos con el entorno de python que tengamos, ya sea 2 o 3

```
iuke@Eugenios-iMac-Pro: ~
→ src git:(master) x python3 main.py
```

```
iuke@Eugenios-iMac-Pro: ~
→ src git:(master) x python main.py
```

Demo

Conclusiones