



Perceptrón

Sistemas de Inteligencia Artificial
Trabajo Práctico 3

Integrantes:

- Ignacio Villanueva 59000
- Guido Barbieri 59567
- Ignacio Méndez 59058



01

Implementacion

02

Pruebas

03

Resultados

04

Conclusiones

01

Implementacion



Implementacion

Clases:

- SimplePerceptron
- Perceptron

Estas clases son capaces de recibir datasets de entrenamiento, y generar una clasificación.

Tienen una implementación muy genérica lo cual nos permite utilizar el mismo código para una gran variedad de problemas



02

Pruebas



Perceptron Simple Escalon

01

Funcion Logica
XOR

02

Funcion Logica
Y

Perceptron Simple

01

No Lineal

02

Lineal

Perceptron Multicapa

01

Funcion Logica
XOR

02

Clasificación
de Digitos

03

Resultados



Perceptron Simple Escalón



Perceptron Simple Escalón

01 XOR

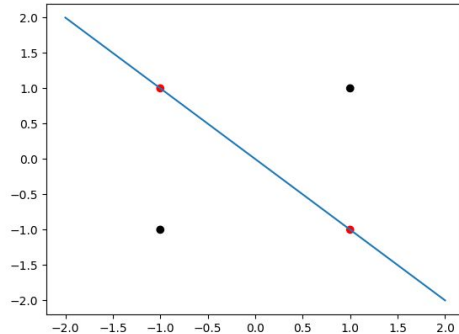
$$w_0 = 0$$

Learning Rate = 0.01

Limit = 5000

Error > 0

No linealmente
separable



02 Y

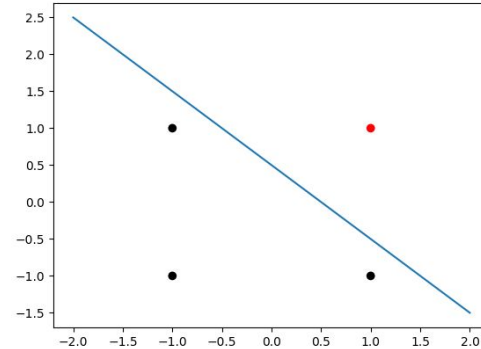
$$w_0 = 0.1$$

Learning Rate = 0.01

Limit = 5000

Error = 0

Linealmente
separable

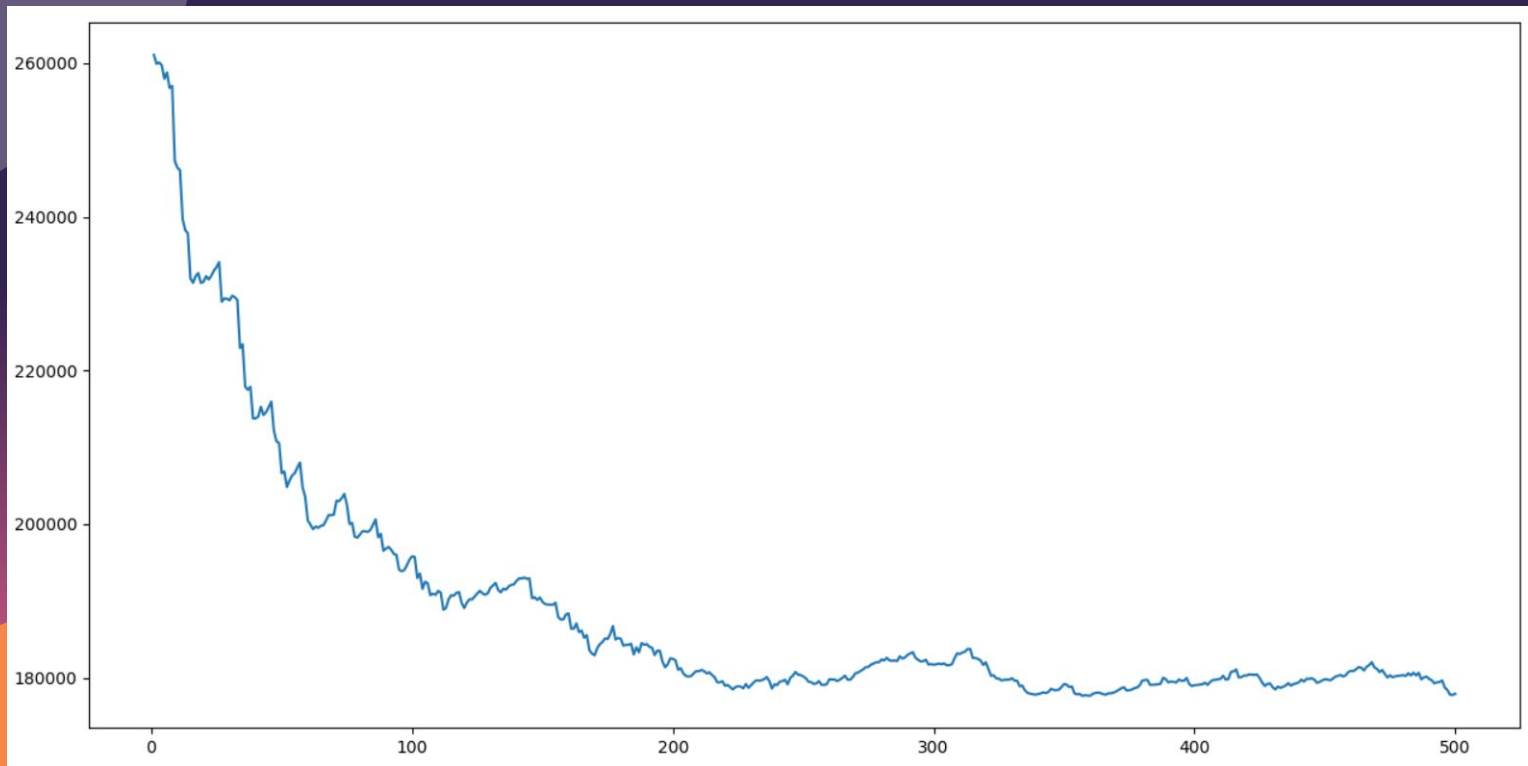


Perceptron Simple Lineal



Perceptron Simple Lineal

Learning rate = 0.1
Epoch = 2000
K = 20

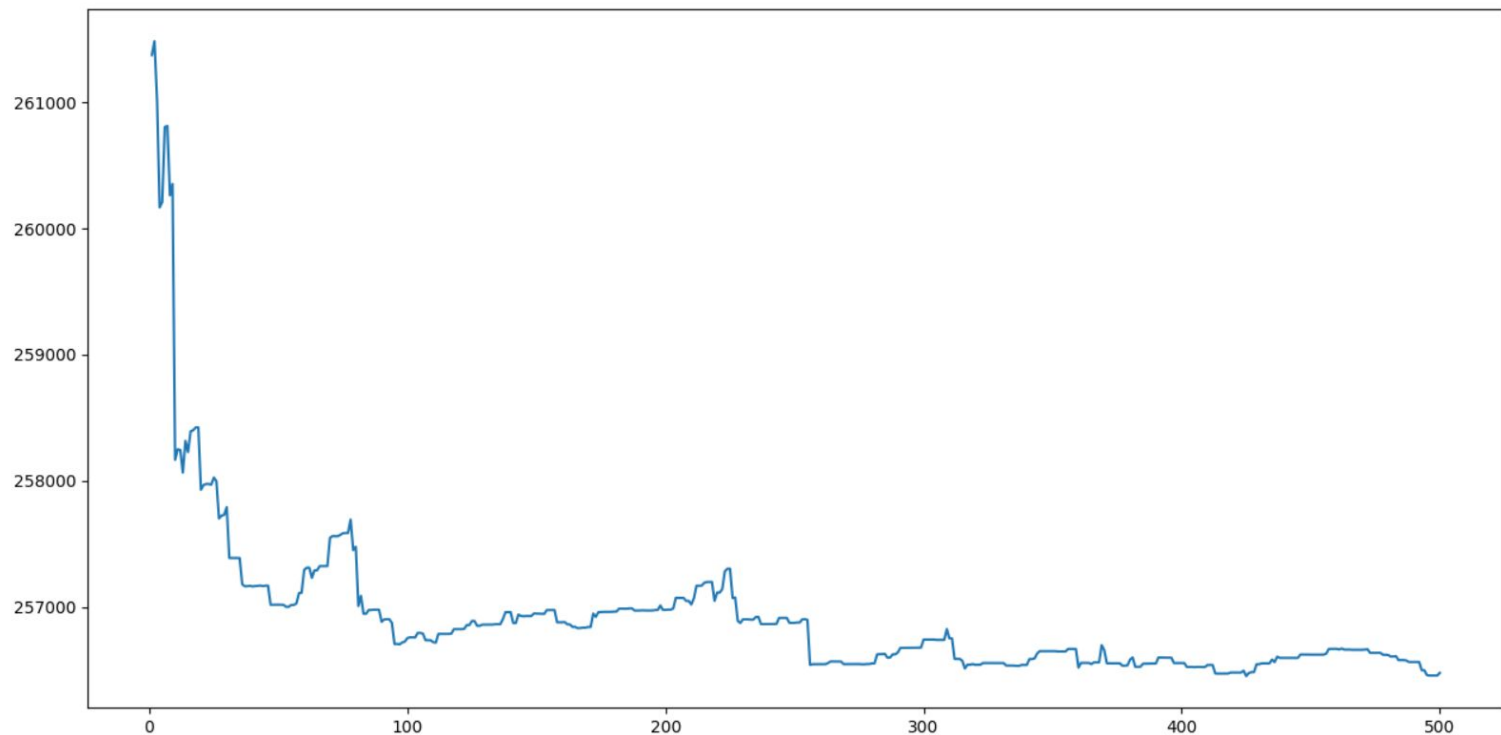


Perceptron
Simple
No Lineal



Perceptron No Lineal

Learning rate = 0.1
Epoch = 2000
K = 20



Perceptron Lineal y No Lineal

01

Lineal

Error train: 2794.3
Error test: 2918.2

02

No Lineal

Error train: 2778.8
Error test: 3212.3

¿Mejor conjunto de entrenamiento?

- 200 entradas
- Con $K=10$, se divide en 10 grupos de 20 entradas.
Lo mejor fue usar al 2do grupo (entradas 20-40) para testear
- Con $K=5$, se divide en 5 grupos de 40 entradas.
Lo mejor fue usar al 2do grupo (entradas 40-80) para testear
- Con $K=2$, se divide en 2 grupos de 100 entradas.
Lo mejor fue usar el 1er grupo (entradas 0-100) para testear

CONCLUSIÓN: las primeras entradas del dataset son mejores para testear

Perceptron Multicapa



Funcion Logica XOR

★ Problema no linealmente separable

Escalón



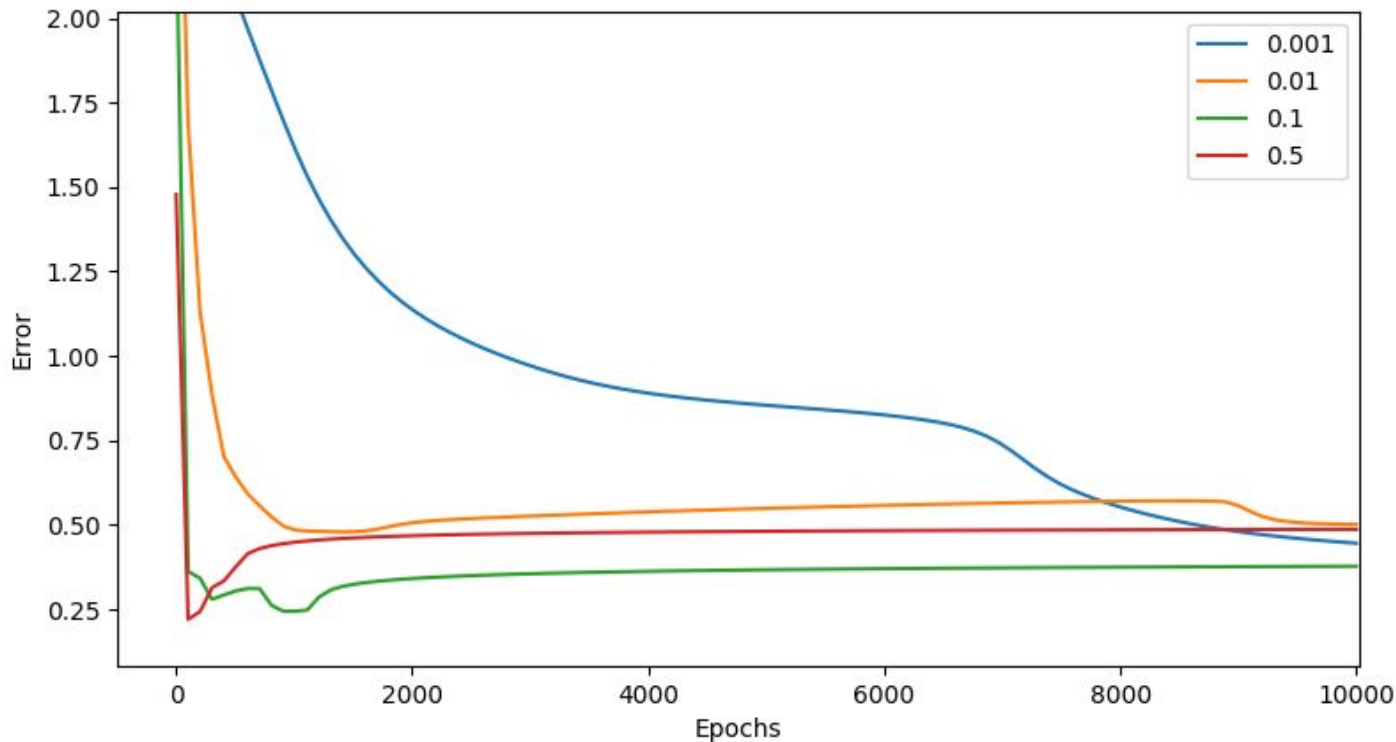
No llegaba a la solución

Multicapa

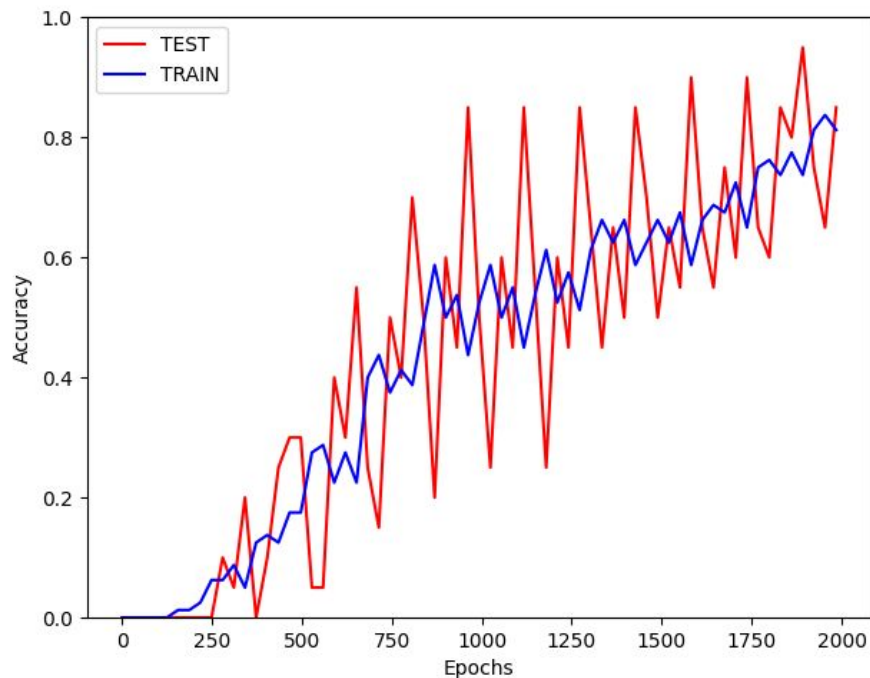
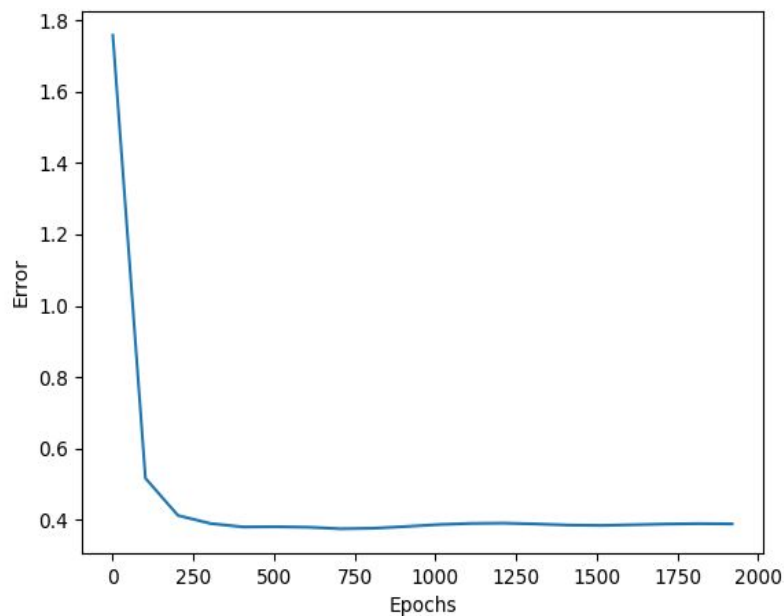


Si llega a la solución

Learning Rate y Hidden Layer



Clasificación de Dígitos



Clasificación de Dígitos

0% Noise



```
0 is 0 with 19.67% certainty
1 is 1 with 20.89% certainty
2 is 2 with 20.79% certainty
3 is 3 with 17.32% certainty
4 is 4 with 21.46% certainty
5 is 5 with 20.97% certainty
6 is 6 with 19.67% certainty
7 is 7 with 10.11% certainty
8 is 8 with 16.50% certainty
9 is 9 with 20.35% certainty
```

20% de Certeza Aprox.

Clasificación de Dígitos

10% Noise

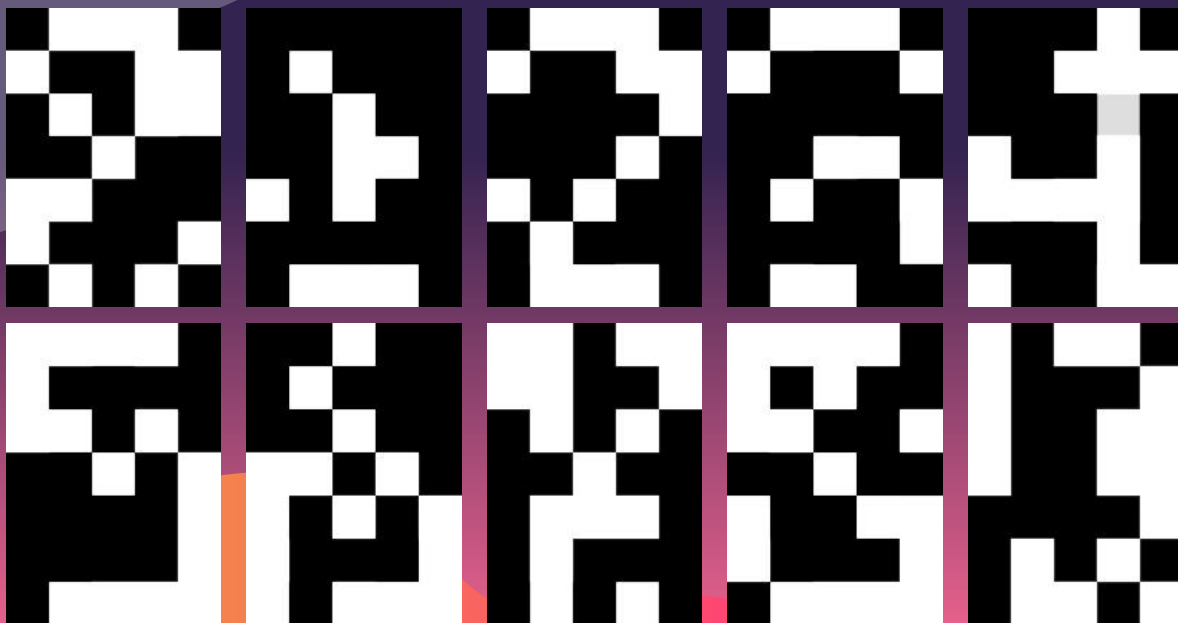


```
0 is 0 with 16.54% certainty
1 is 1 with 18.46% certainty
2 is 2 with 15.02% certainty
3 is 3 with 18.11% certainty
4 is 4 with 19.34% certainty
5 is 5 with 17.05% certainty
6 is 6 with 18.81% certainty
7 is 7 with 16.64% certainty
8 is 8 with 10.92% certainty
9 is 9 with 12.00% certainty
```

16% de Certeza Aprox.

Clasificación de Dígitos

20% Noise



```
0 is 8 with 11.00% certainty
1 is 1 with 19.17% certainty
2 is 2 with 18.50% certainty
3 is 3 with 10.74% certainty
4 is 4 with 20.50% certainty
5 is 5 with 18.92% certainty
6 is 6 with 13.97% certainty
7 is 9 with 11.85% certainty
8 is 3 with 12.70% certainty
9 is 9 with 9.90% certainty
```

13% de Certeza Aprox.

Clasificación de Dígitos

30% Noise



```
0 is 8 with 13.52% certainty
1 is 5 with 11.15% certainty
2 is 8 with 12.59% certainty
3 is 1 with 12.41% certainty
4 is 8 with 19.17% certainty
5 is 6 with 13.22% certainty
6 is 6 with 11.87% certainty
7 is 2 with 15.90% certainty
8 is 6 with 12.76% certainty
9 is 5 with 10.95% certainty
```

12% de Certeza Aprox.

Conclusiones

01

El perceptrón es una herramienta que permite resolver una variedad de problemas, pero cada uno requiere de la configuración correcta del mismo.

02

Learning Rate: cuanto más bajo, mas tardar en converger. Si es muy alto, puede diverger.

03

Los problemas que no son linealmente separables solo pueden ser resueltos por un perceptrón superior al escalón.

04

El perceptrón tiene la capacidad de predecir resultados pese a nunca haber entrenado con el mismo.

