

Trabajo Práctico N° 3

Perceptrón Simple y Multicapa

Grupo 1

- Augusto Henestrosa
- Francisco Choi
- Nicolás de la Torre

Algoritmos a desarrollar

- Perceptrón Simple
- Perceptrón Simple Lineal
- Perceptrón Simple No Lineal
- Perceptrón Multicapa

Desarrollo

Python 3.8.0



Marco Teórico

¿Qué es una Red Neuronal?

- Una **red neuronal** es un modelo que emula el modo en que el cerebro procesa la información.
- Consiste en subunidades representadas como neuronas que se conectan mediante pesos sinápticos
- Requiere entrenamiento

Perceptrón Simple Escalón

$$O = \theta(\sum w_i \varepsilon_i - \text{umbral})$$

$$\theta(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{en otro caso} \end{cases}$$

$$w_i^{\text{nuevo}} = w_i^{\text{viejo}} + \Delta w_i$$

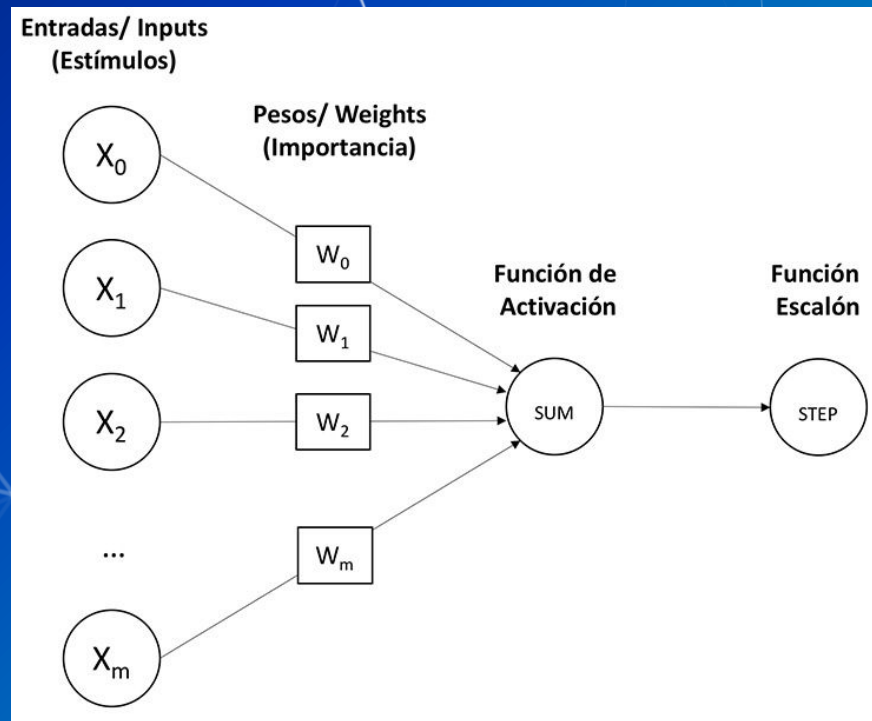
$$\Delta w_i = \eta(\zeta^u - O^u)\varepsilon_i^u$$

$w \rightarrow$ Peso sináptico

$\varepsilon \rightarrow$ Es el input ingresado para entrenar el sistema.

$O \rightarrow$ Es el estado de activación de la neurona.

$\zeta \rightarrow$ Es la salida esperada de cada estado de activación.



Perceptrón Simple Lineal y No Lineal

$O^u = g(\sum w_i \varepsilon_i^u) \rightarrow g(x)$ puede ser lineal o no lineal

$$E(w) = \frac{1}{2} \sum (\zeta^u - O^u)^2$$

$$w_i^{\text{nuevo}} = w_i^{\text{viejo}} + \Delta w_i$$

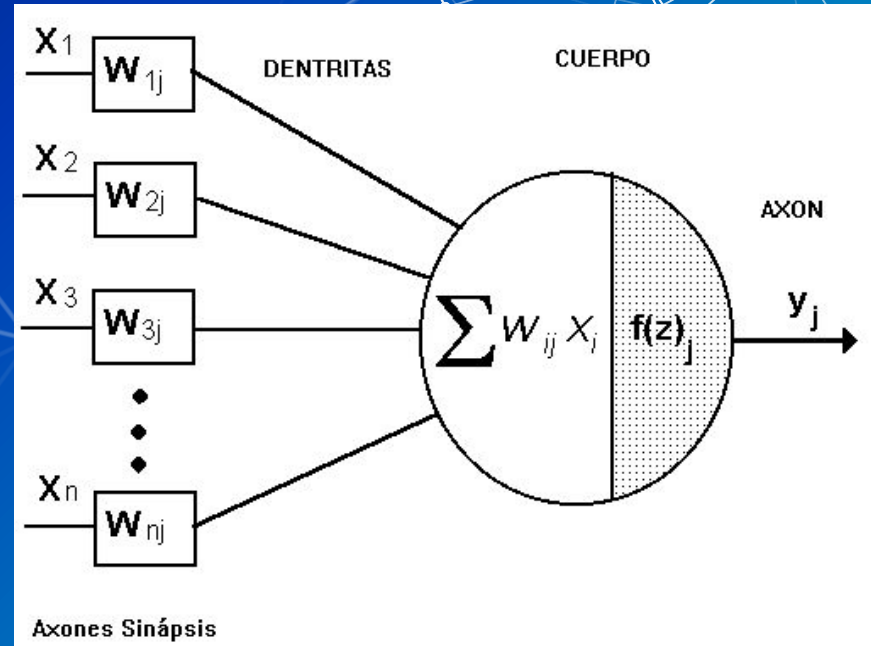
$$\Delta w_i = \eta (\zeta^u - O^u) \varepsilon_i^u$$

$w \rightarrow$ Peso sináptico

$\varepsilon \rightarrow$ Es el input ingresado para entrenar el sistema.

$O \rightarrow$ Es el estado de activación de la neurona.

$\zeta \rightarrow$ Es la salida esperada de cada estado de activación.



Perceptrón Multicapa

$$h_j^\mu = \sum w_{ik} \varepsilon_i^\mu$$

$$V_j^\mu = g(\sum w_{ik} \varepsilon_i^\mu)$$

$$O_j^\mu = g(\sum W_{ij} g(\sum w_{jk} \varepsilon_k^\mu))$$

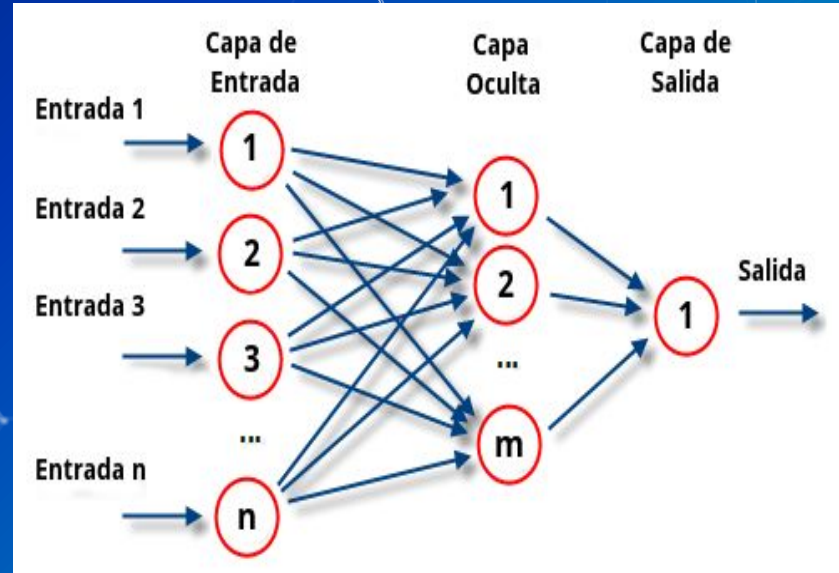
$$w_{ij}^{\text{nuevo}} = w_{ij}^{\text{viejo}} + \Delta w_{ij}$$

$$\delta_j^\mu = g'(h_j^\mu) \sum W_{ij} \delta_i^\mu$$

$$\Delta w_{jk} = \eta \sum \delta_i^\mu V_i^\mu$$

$$E(w) = \frac{1}{2} \sum (\zeta_i^\mu - O_i^\mu)^2$$

$$E(w) = \sum_{i\mu} \left(\frac{1}{2} (1 + \zeta_i^\mu) \log \frac{1 + \zeta_i^\mu}{1 + O_i^\mu} + \frac{1}{2} (1 - \zeta_i^\mu) \log \frac{1 - \zeta_i^\mu}{1 - O_i^\mu} \right)$$



$w \rightarrow$ Conexiones de la capa oculta.

$W \rightarrow$ Conexiones de la capa de salida.

$V \rightarrow$ Es el estado de activación de la capa oculta.

$h \rightarrow$ Es el estado de excitación.

Perceptrón Simple Escalonado

AND y XOR Lógico

Perceptrón simple escalonado:

- 2 entradas
- 1 salida
- Taza de aprendizaje: 0.1

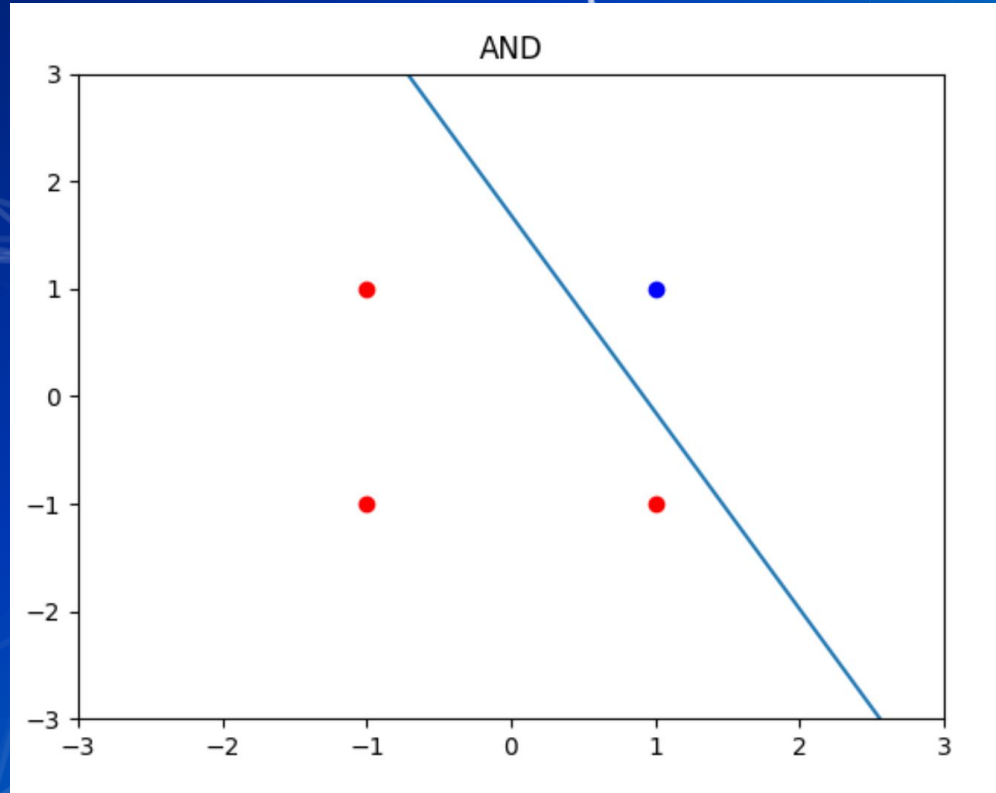
AND

```
1 AND 1  
Expected: 1  
Actual: 1
```

```
1 AND -1  
Expected: -1  
Actual: -1
```

```
-1 AND 1  
Expected: -1  
Actual: -1
```

```
-1 AND -1  
Expected: -1  
Actual: -1
```



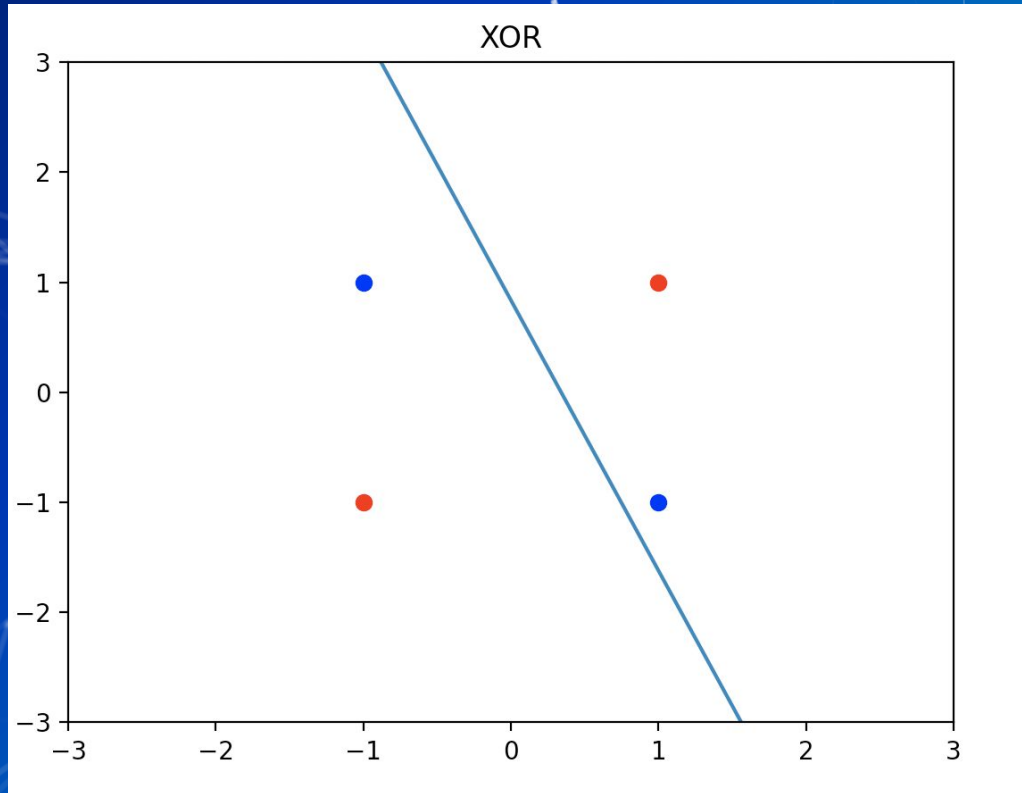
XOR

```
1 XOR 1  
Expected: -1  
Actual:1
```

```
1 XOR -1  
Expected: 1  
Actual:1
```

```
-1 XOR -1  
Expected: -1  
Actual:-1
```

```
-1 XOR 1  
Expected: 1  
Actual:1
```



Perceptrón Simple Lineal y No Lineal

Perceptrón Simple Lineal y No Lineal

Perceptron No Lineal:

- 3 entradas, 1 salida
- Características del perceptrón:
 - Función de activación: tanh

Perceptron Lineal:

- 3 entradas, 1 salida
- Características del perceptrón:
 - Función de activación: lineal

Testing Perceptrón No Lineal:

- 70% para entrenamiento
30% para prueba
- Épocas: 10, 50, 500, 1000, 10mil, 15mil
- Error mínimo: 0.001
- Taza de aprendizaje: 0.01, 0.03, 0.05

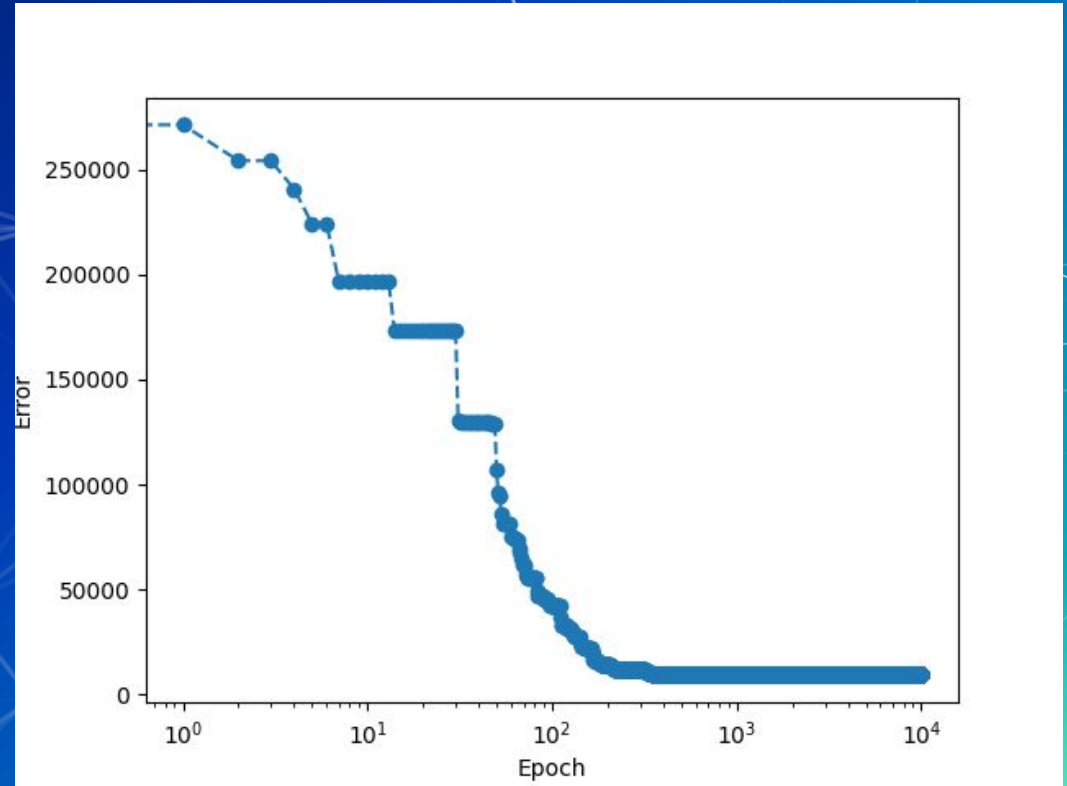
Testing Perceptrón Lineal:

- Todo el conjunto
- Épocas: 10mil
- Error mínimo: 0.001
- Taza de aprendizaje: 0.03

Perceptrón Simple Lineal

Perceptrón Simple Lineal

- Factor de Aprendizaje: 0.01
- Épocas: 10000
- Error mínimo: 9267.9205



Perceptrón Simple Lineal

- Factor de Aprendizaje: 0.01
- Épocas: 10000

```
~~~~~ Simple Linear Perceptron Results ~~~~~  
  
prediction: 74.7425 -- expected: 87.3174 -- error: [12.57486587]  
prediction: -4.7091 -- expected: 1.5257 -- error: [6.23481122]  
prediction: 45.4949 -- expected: 39.7859 -- error: [5.708984]  
prediction: 48.1484 -- expected: 45.5674 -- error: [2.58095655]  
prediction: 24.916 -- expected: 13.3589 -- error: [11.55705205]  
prediction: 63.5453 -- expected: 74.5119 -- error: [10.96663392]  
prediction: 4.1087 -- expected: 3.3358 -- error: [0.77293106]  
prediction: 8.7411 -- expected: 4.2974 -- error: [4.44370526]
```

Perceptrón Simple

No Lineal

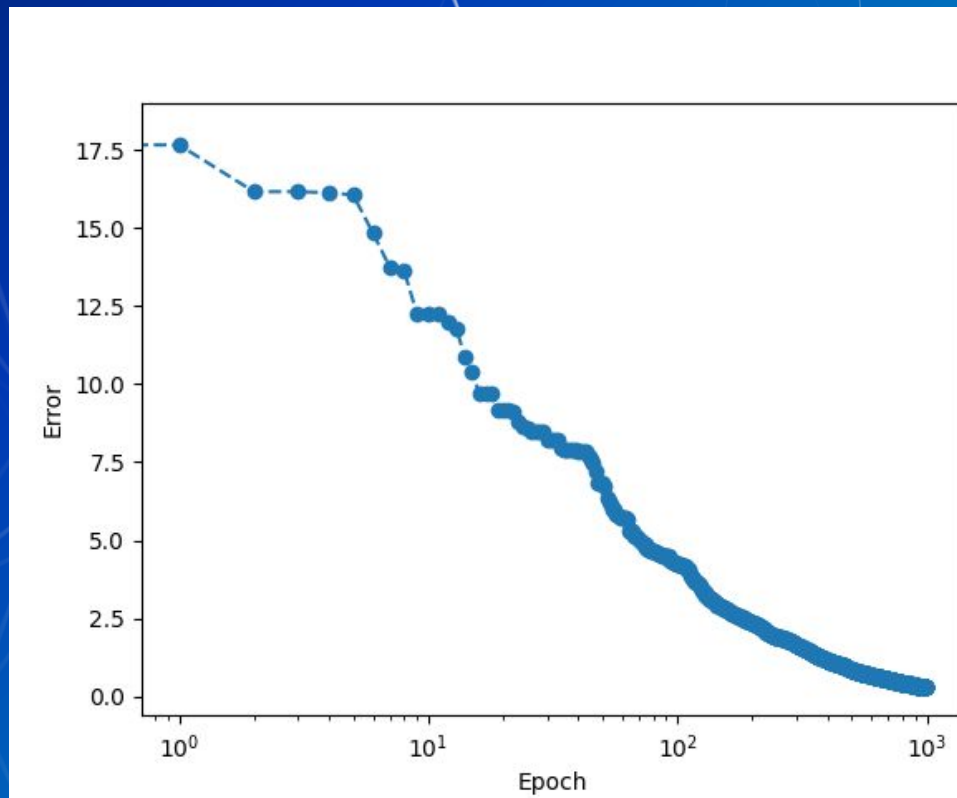
Perceptrón Simple No Lineal

- Épocas: 1000
- Factor de Aprendizaje: 0.05
- Error mínimo: 0.001
- Función de activación: tanh
- Conjunto de entrenamiento: 70%
- Conjunto de prueba: 30%

```
~~~~~ Simple Non Linear Perceptron Results ~~~~~  
  
prediction: -0.2158 -- expected: -0.21698599986294131 -- error: [0.00116858]  
prediction: 0.096 -- expected: 0.09438704553974908 -- error: [0.00158709]  
prediction: -0.3189 -- expected: -0.3201044870580363 -- error: [0.00124205]  
prediction: 0.9705 -- expected: 0.9815192547234843 -- error: [0.01101372]  
prediction: 0.9674 -- expected: 0.9783024077978306 -- error: [0.010895]  
prediction: -0.9808 -- expected: -0.9834239667189371 -- error: [0.0026537]  
prediction: -0.985 -- expected: -0.9877030181118963 -- error: [0.00272258]  
prediction: 0.9438 -- expected: 0.9537427086818586 -- error: [0.00998013]
```

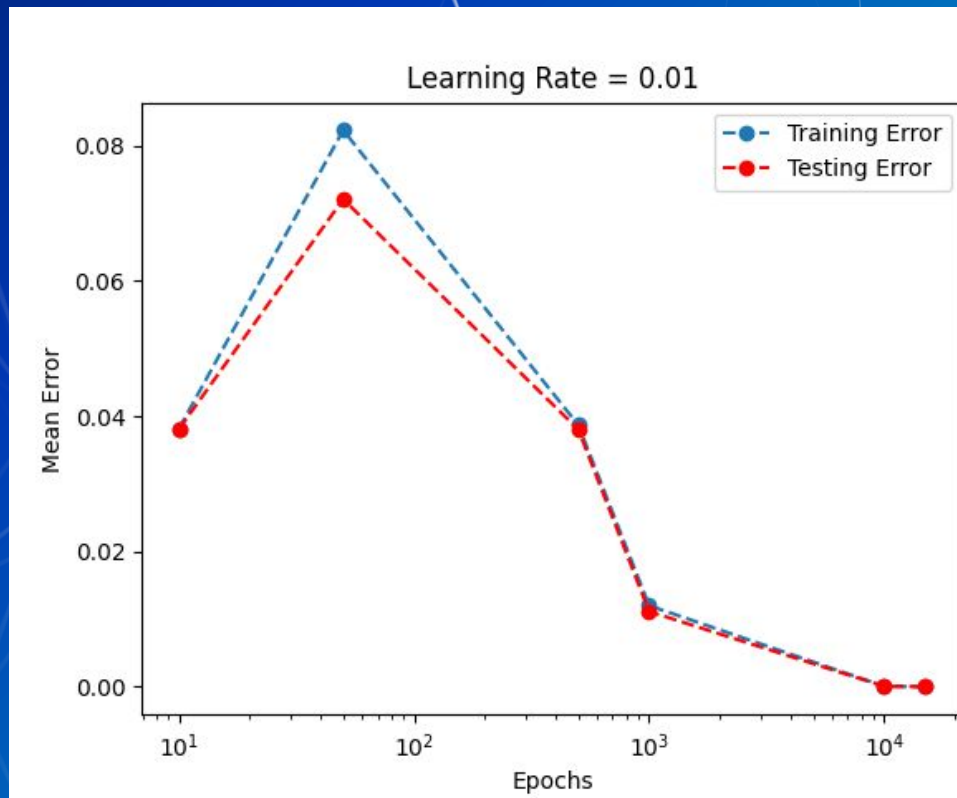
Perceptrón Simple No Lineal

- Factor de Aprendizaje: 0.01
- Épocas: 1000
- Función de activación: tanh
- Conjunto de entrenamiento: 70%
- Conjunto de prueba: 30%
- Error mínimo: 0.001



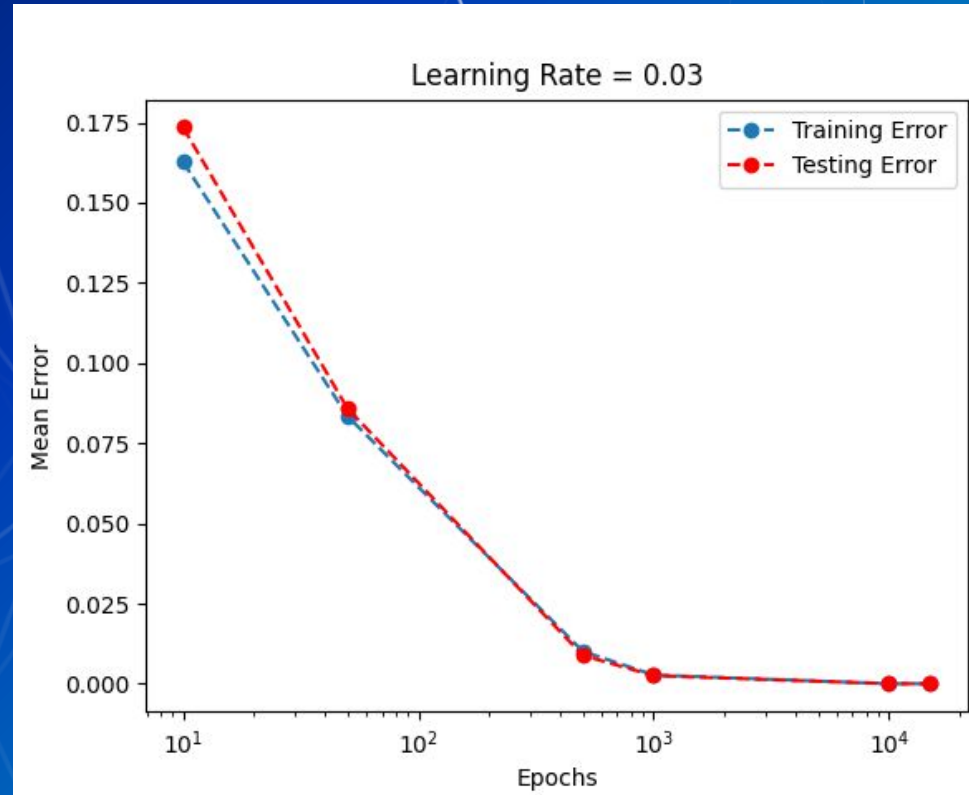
Perceptrón Simple No Lineal

- Factor de Aprendizaje: 0.01
- Función de activación: tanh
- Conjunto de entrenamiento: 70%
- Conjunto de prueba: 30%
- Error mínimo: 0.001



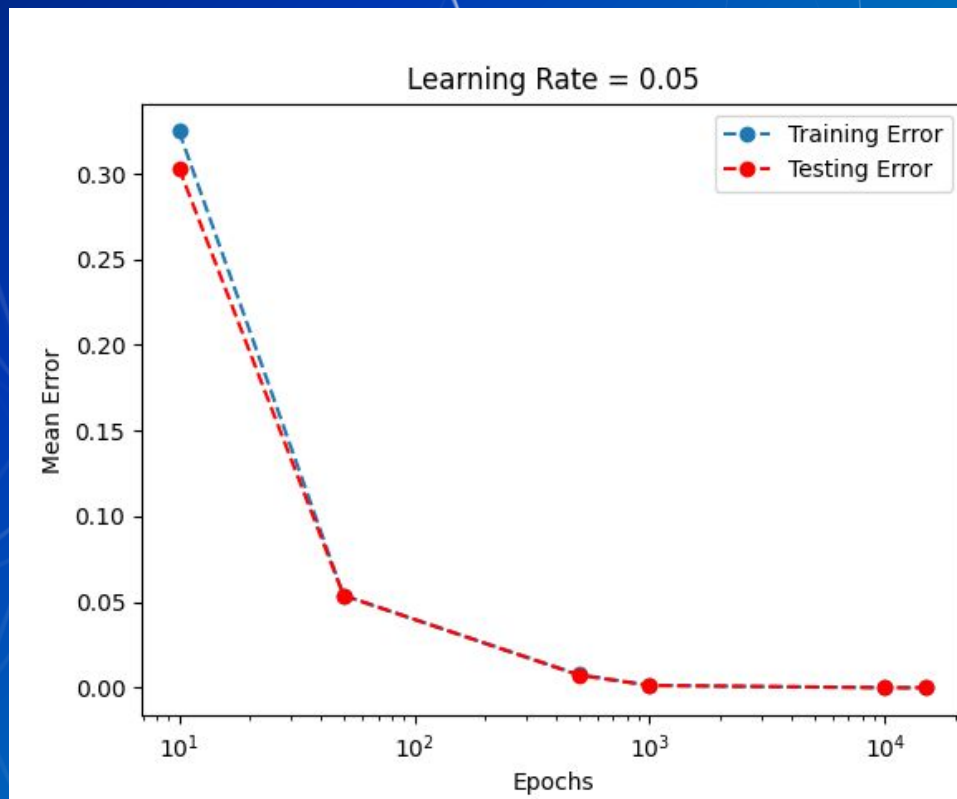
Perceptrón Simple No Lineal

- Factor de Aprendizaje: 0.03
- Función de activación: tanh
- Conjunto de entrenamiento: 70%
- Conjunto de prueba: 30%
- Error mínimo: 0.001



Perceptrón Simple No Lineal

- Factor de Aprendizaje: 0.05
- Función de activación: tanh
- Conjunto de entrenamiento: 70%
- Conjunto de prueba: 30%
- Error mínimo: 0.001



Perceptrón Multicapa

Ejercicio 3: XOR

Perceptrón Multicapa para XOR

Perceptron:

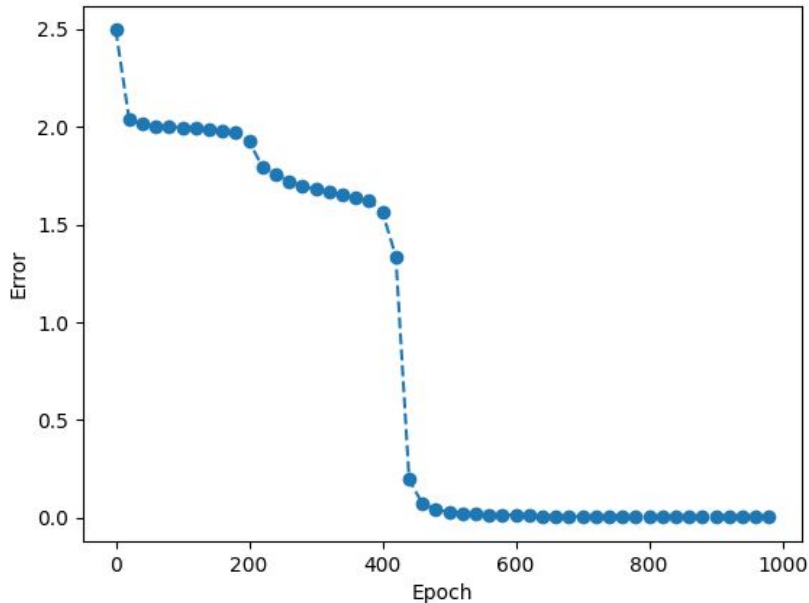
- 2 entradas, 1 salida
- 1 capa oculta con dos neuronas
- Características del perceptrón:
 - Taza de aprendizaje: 0.1
 - Función de activación: tanh
 - Función de costo: Cuadrático
- Características adicionales:
 - Momentum: NO
 - Adaptativo: NO

Testing:

- Todo el conjunto
- Iteraciones: 1000
- Error mínimo: 0.001

Perceptrón Multicapa para XOR

Training



Testing

```
----- XOR RESULTS -----  
-1 XOR -1  
Expected: 1  
Actual:0.9710559862309444  
Classified: 1  
  
1 XOR 1  
Expected: 1  
Actual:0.9711458954745029  
Classified: 1  
  
-1 XOR -1  
Expected: -1  
Actual:-0.9611423281112923  
Classified: -1  
  
1 XOR 1  
Expected: -1  
Actual:-0.9597026746795568  
Classified: -1
```

Ejercicio 3: Paridad

Perceptron Multicapa para Paridad

Entrada

0 1 1 1 0
1 0 0 0 1
1 0 0 1 1
1 0 1 0 1
1 1 0 0 1
1 0 0 0 1
0 1 1 1 0



Salida

1

Perceptron Multicapa para Paridad

Perceptron:

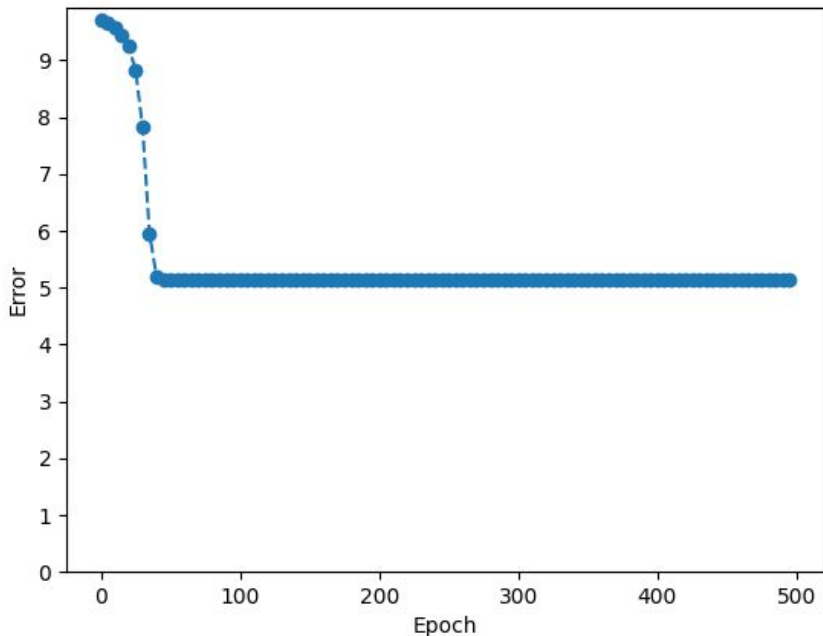
- 35 entradas, 1 salida
- 3 capas ocultas con:
 - 8, 4 y 2 neuronas c/u
- Características del perceptrón:
 - Taza de aprendizaje: 0.01
 - Función de activación: tanh
 - Función de costo: Entrópico
- Características adicionales:
 - Momentum: 0.9
 - Adaptativo: Si
 - $a = 10^{-4}$, $b = 10^{-5}$

Testing:

- Separando el conjunto de entrenamiento del de prueba:
 - Training: [0, 1, 4, 5, 7, 8]
 - Testing: [2,3,5,9]
- Iteraciones: 1000
- Error mínimo: 0.001

Training: Primer Approach

- Error: Cuadrático
- Iteraciones: 500
- Error final: 0.038
- Resultados similares utilizando momentum y adaptativo.
- Mínimos locales, muy lento para converger



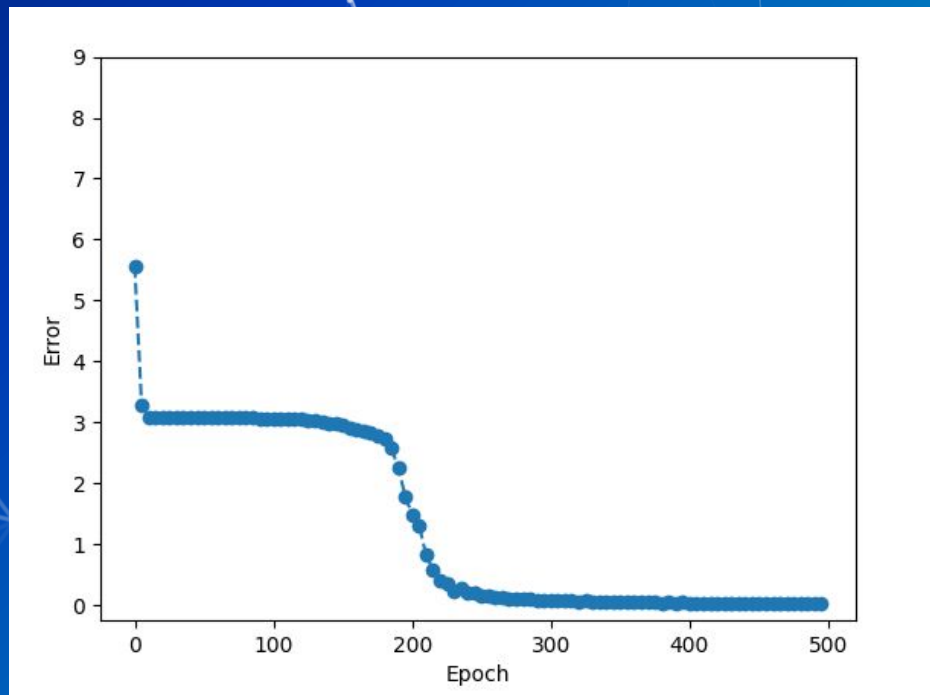
NO SIRVE.

Training: con costo entrópico

- Error: entrópico
- Iteraciones: 500
- Error final: 0.021

SIRVE, Se puede mejorar?

- Tiene oscilaciones finales
- Podría converger más rapido

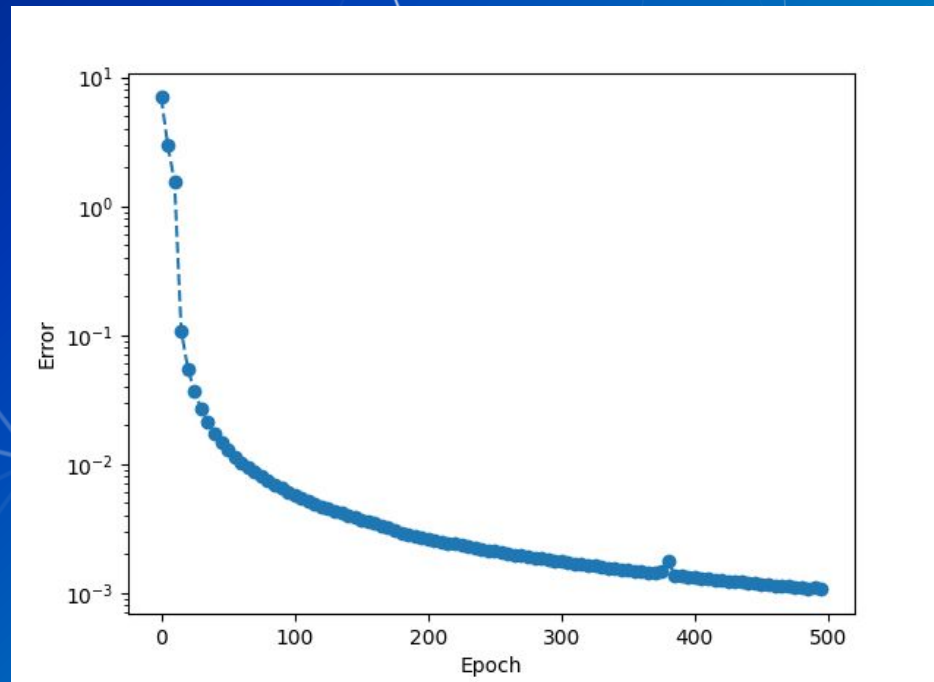


Training: entrópico + momentum

- Error: entrópico
- Iteraciones: 500
- Momentum de 0.9
- Error final: 0.001059
- Soluciona las perturbaciones
- Más rápido en regiones planas

SIRVE, Se puede mejorar?

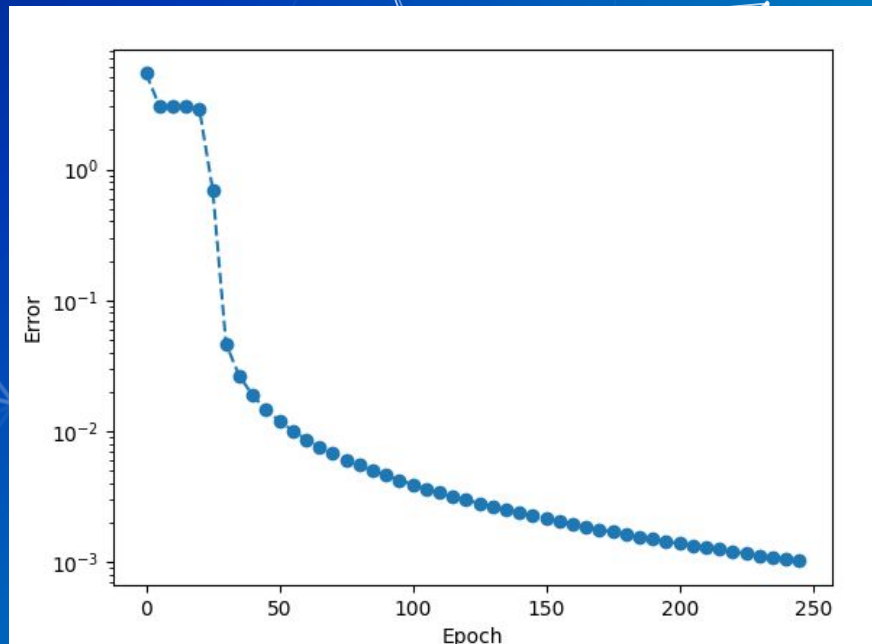
- Podría converger más rapido



Training: entrópico + momentum + adaptativo

- Error: entrópico
- Momentum de 0.9
- Adaptativo con
 - $a = 10^{-4}$, $b = 10^{-5}$
- Error final: 0.000993
- Iteraciones: 249

(Siempre se puede mejorar)



Testing: clasificaciones de paridad

----- Testing -----

```
1 1 1
1 1
  1
  1
  1
1 1 1 1 1 Expected: [1]
Actual: 0.9923700060550805
Classified: 1
```

```
1 1 1
1 1
  1
  1
  1
1 1 1 Expected: [-1]
Actual: 0.9927107126948421
Classified: 1
```

```
1 1 1 1 1
1
1 1 1 1
  1
  1
1 1
  1 1 Expected: [-1]
Actual: -0.9931169683240569
Classified: -1
```

```
1 1 1
1 1
1 1
1 1 1 1
  1
  1
1 1 Expected: [-1]
Actual: 0.9926608724945125
Classified: 1
```

Misma efectividad para:

- Train: [1,3,4,5,8,9]
Test: [0,2,6,7]
- Train: [0,2,4,5,7]
Test: [1,3,5,6,8]
- Otros

La red no generaliza bien

25-75%
efectividad



Conclusiones

Conclusiones particulares

- **Perceptrón simple escalón:**
 - ◆ **Permite resolver problemas linealmente separables únicamente**
- **Perceptrón simple lineal y no lineal:**
 - ◆ **Realizar una transformación no lineal sobre los datos permite tener una mayor efectividad**
- **Perceptron Multicapa:**
 - ◆ **Permite resolver problemas linealmente separables y no separables**
 - ◆ **No es perfecto: mínimos locales y convergencia lenta**
 - ◆ **Permite resolver una gran variedad de problemas**
 - ◆ **No es efectivo para inputs que no están en el conjunto de entrenamiento (para el ejemplo de paridad)**
 - **Baja capacidad de generalización**
 - **Necesita cierto parecido en los datos**
 - **La red no es adivina**

Conclusiones generales

- Es sumamente importante el conjunto de entrenamiento
- Es muy importante la función de costo, una función de costo inadecuada resulta en un mal entrenamiento.
- Error de entrenamiento 0 no asegura efectividad en el testing
- Es importante la división entre el conjunto de entrenamiento y el de testing

A person is shown from the chest up, wearing a VR headset. The image is heavily stylized with a blue color palette. A white, glowing network of lines and dots is overlaid on the person's face and the VR headset, suggesting a digital or networked environment. The text '¡Gracias!' is prominently displayed in the upper left corner.

¡Gracias!