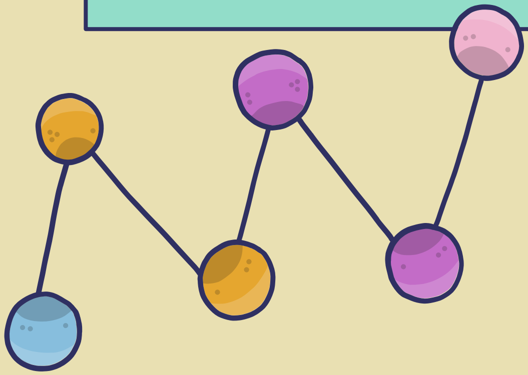
# Perceptrones

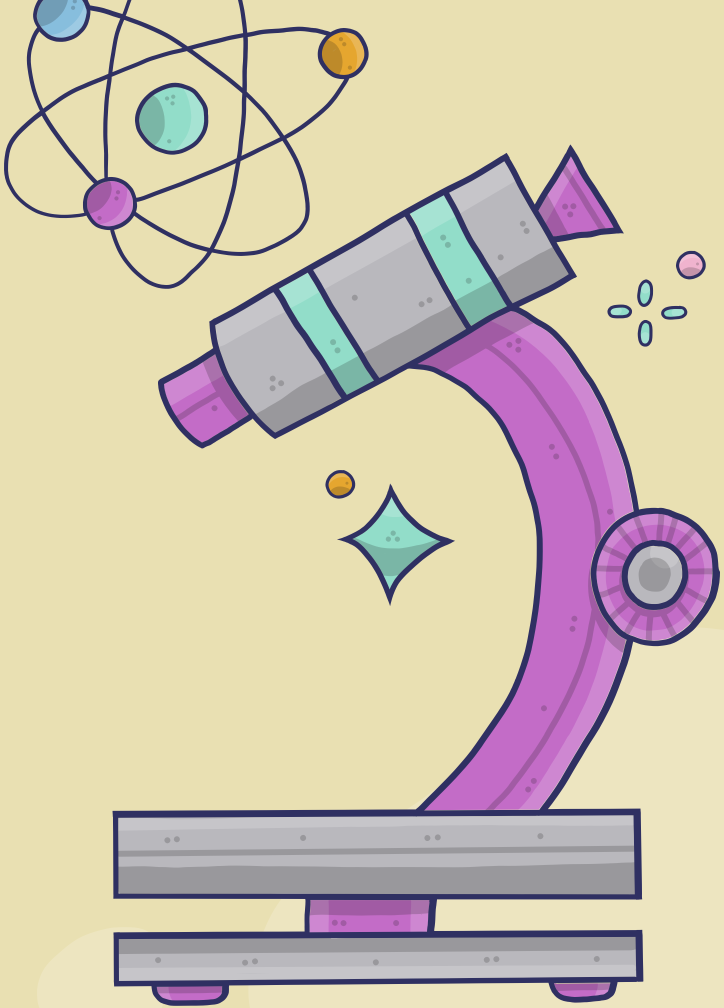
Grupo 4

Alejando Rolandelli, Benjamin Delasoie, Facundo Zimbimbakis, Malena Vásquez, Currie Santiago Larroude Álvarez

# 01

## Simple

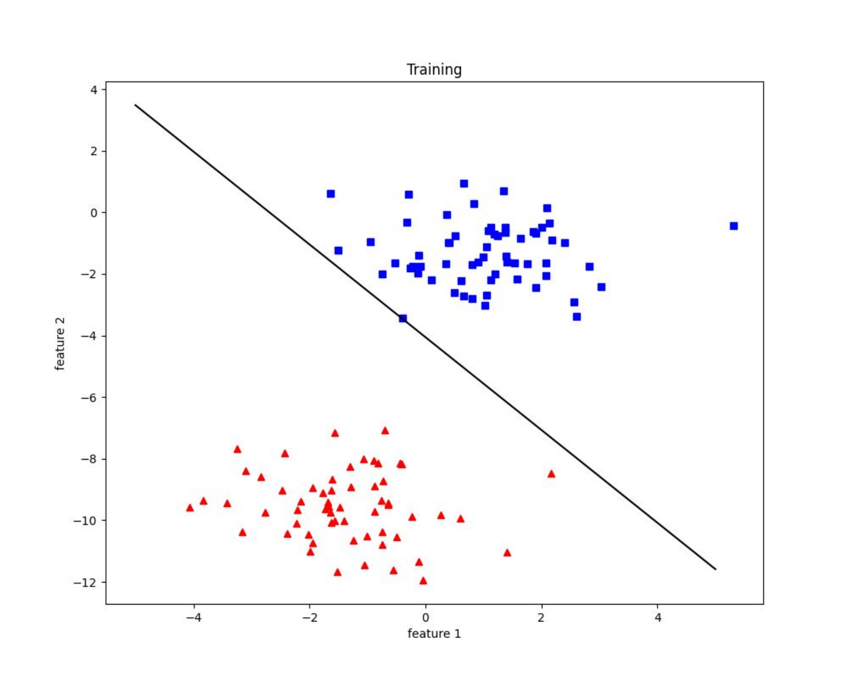


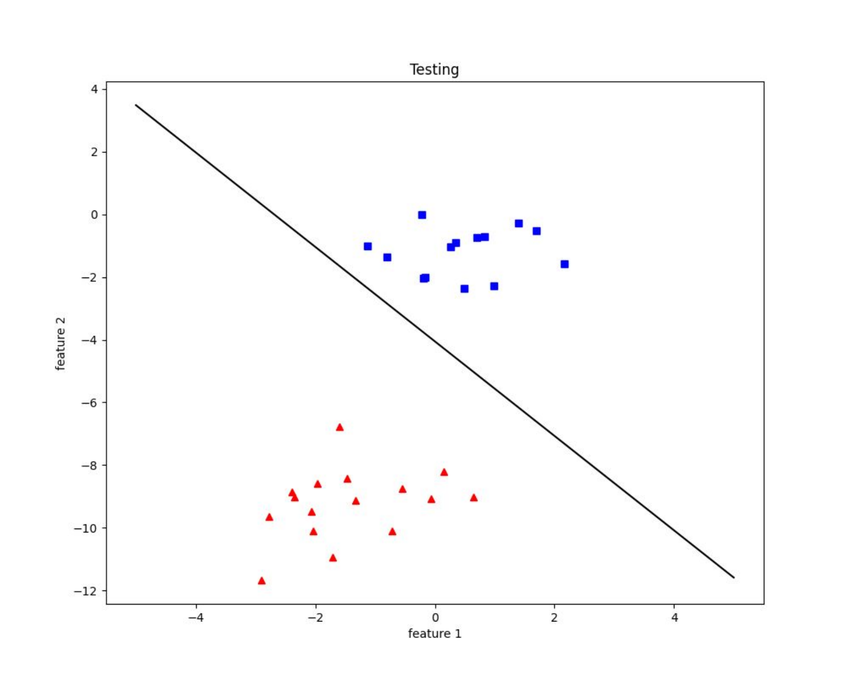


Función de Activación

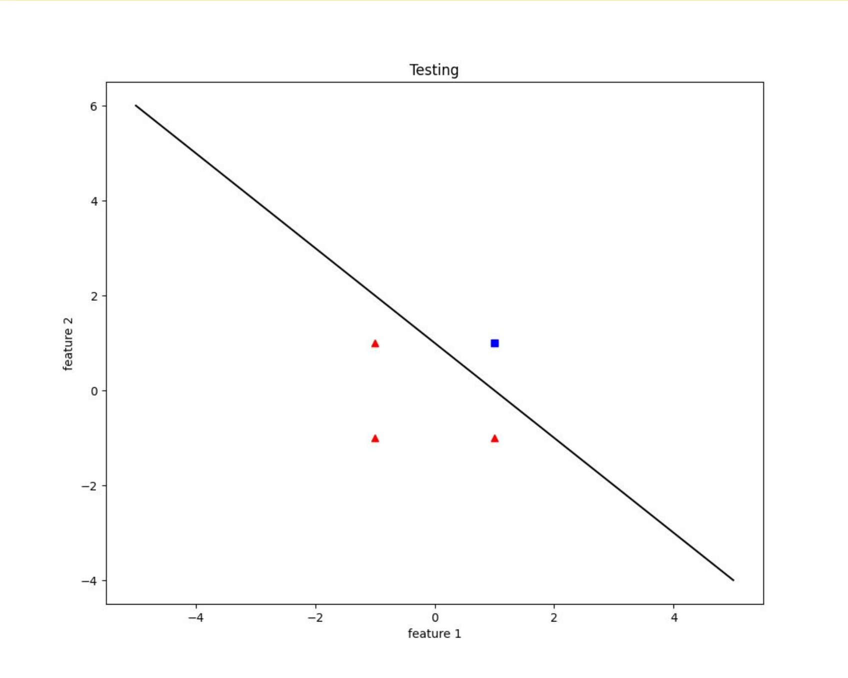
# Escalón

# Dataset: Random

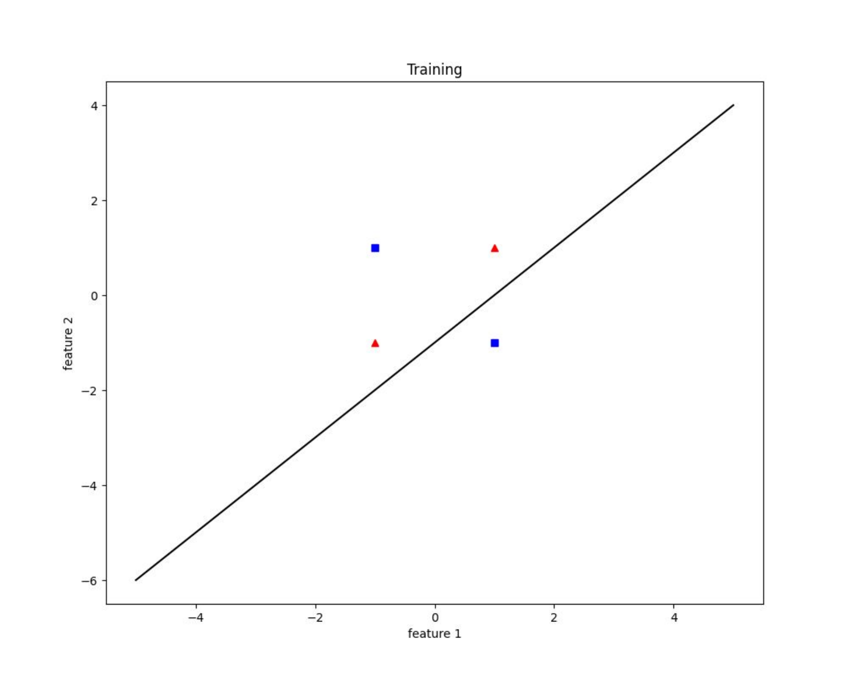


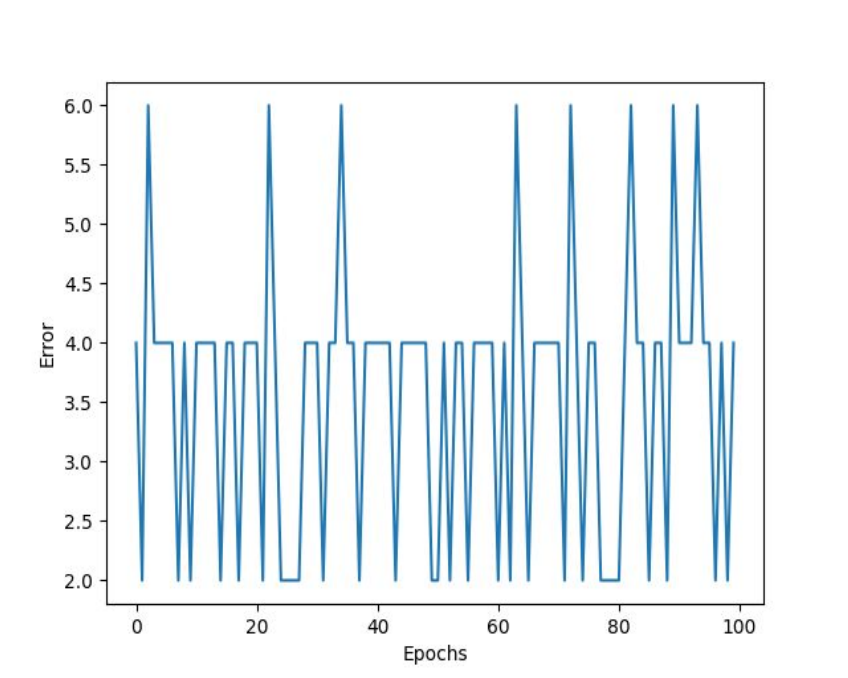
# Dataset: AND



# Dataset: XOR



Training
feature 2
feature 1
4
2
0
−2
−4
−6
−4
−2
0
2
4

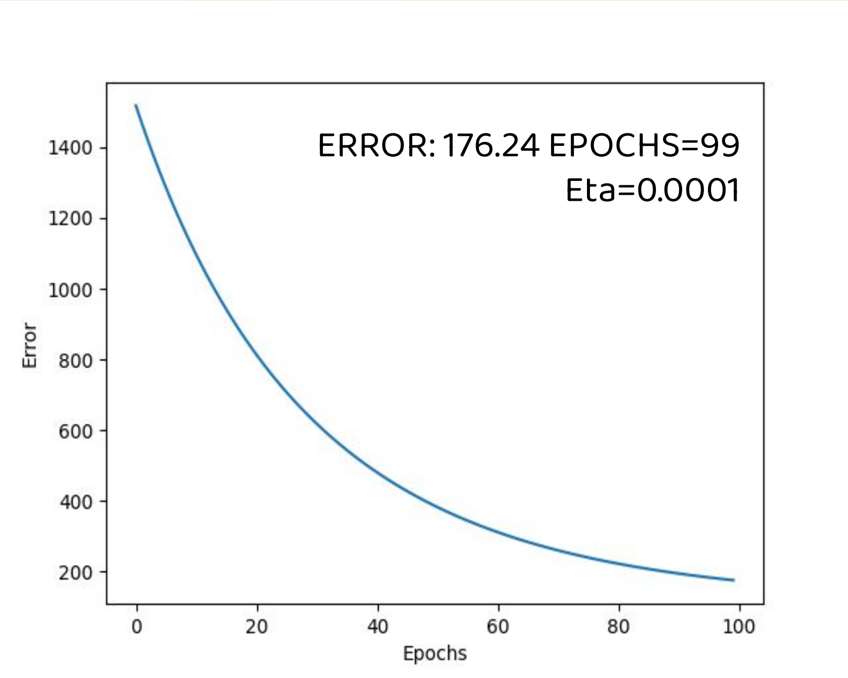
Error
Epochs
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
0
20
40
60
80
100

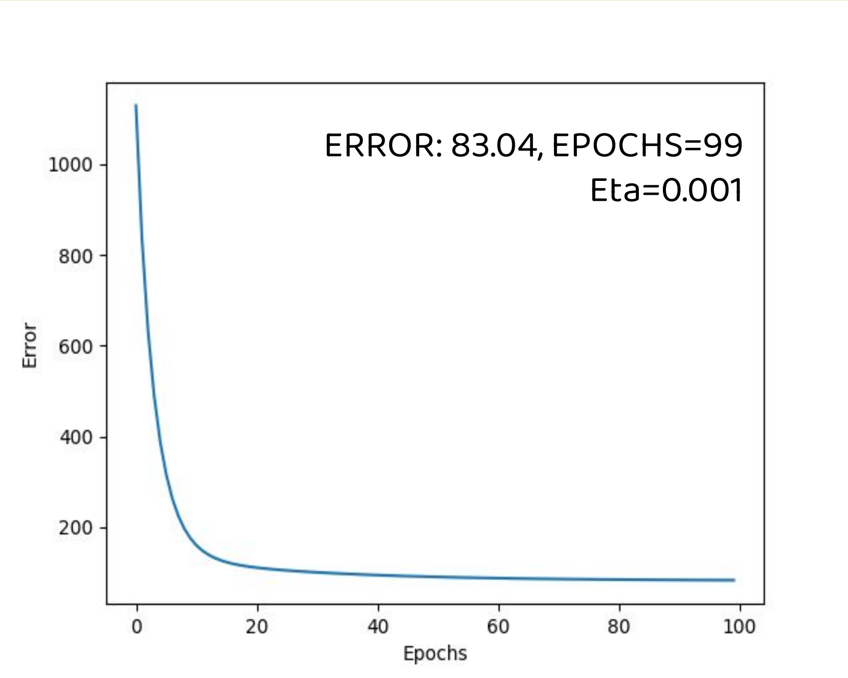
# CONCLUSIONES

- El perceptrón simple solo puede aprender problemas que son linealmente separables
- AND es linealmente separable así que lo aprende bien, con el XOR sucede lo opuesto.
- Para resolver el XOR se necesitan dos rectas, no existe un hiperplano.

# Lineal

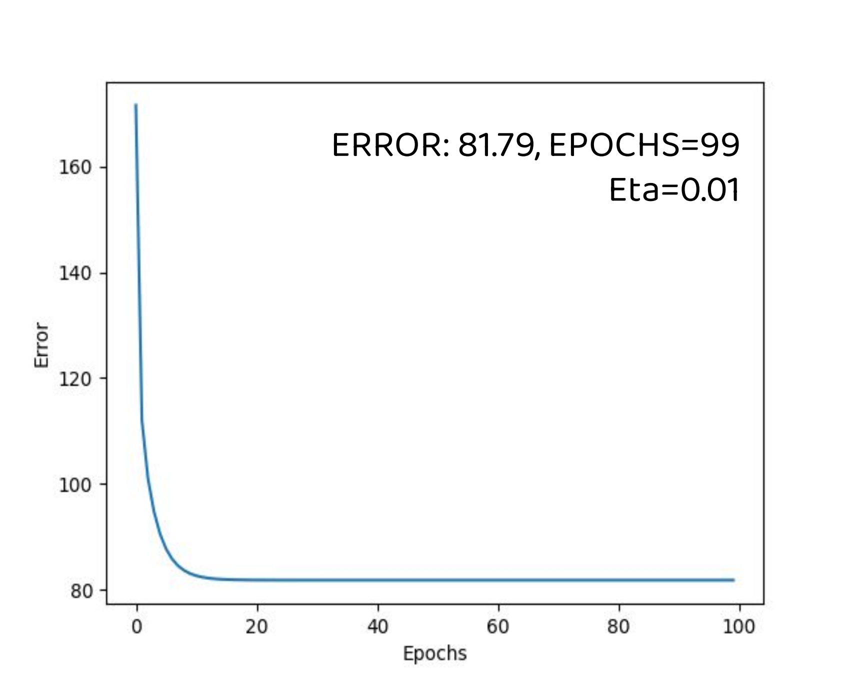
# Variación de Eta (1/4)

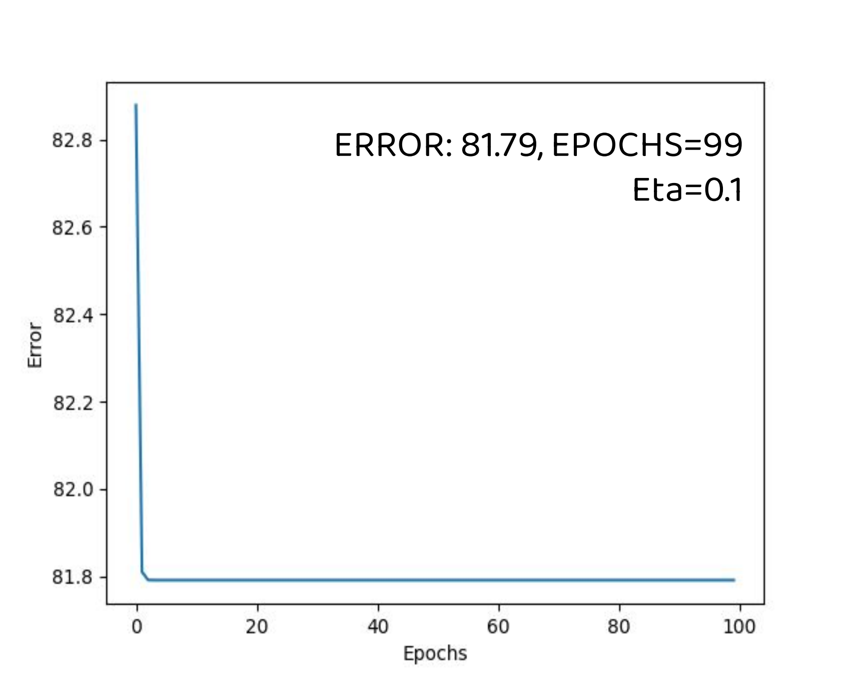



MAX EPOCHS = 100

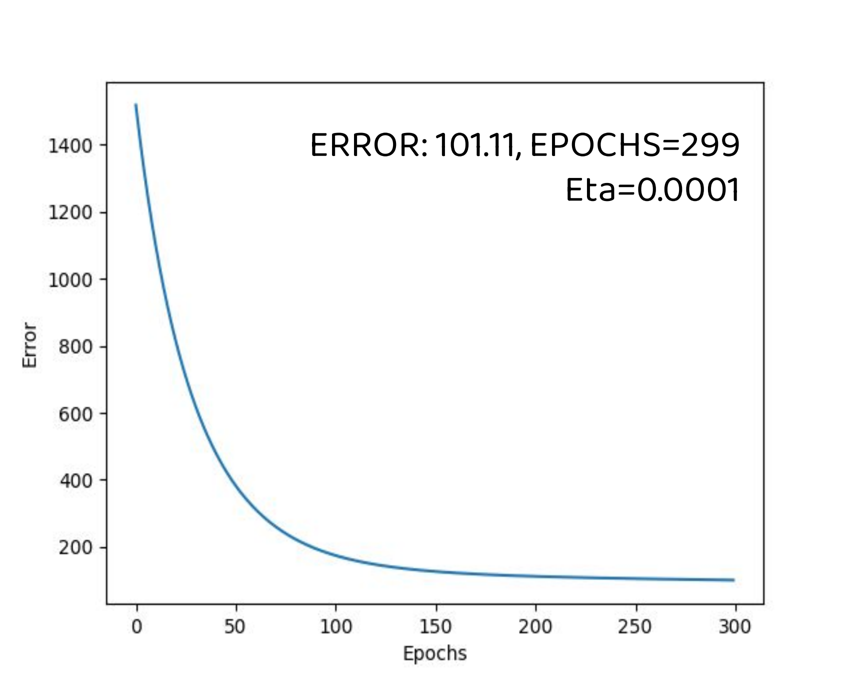
# Variación de Eta (2/4)

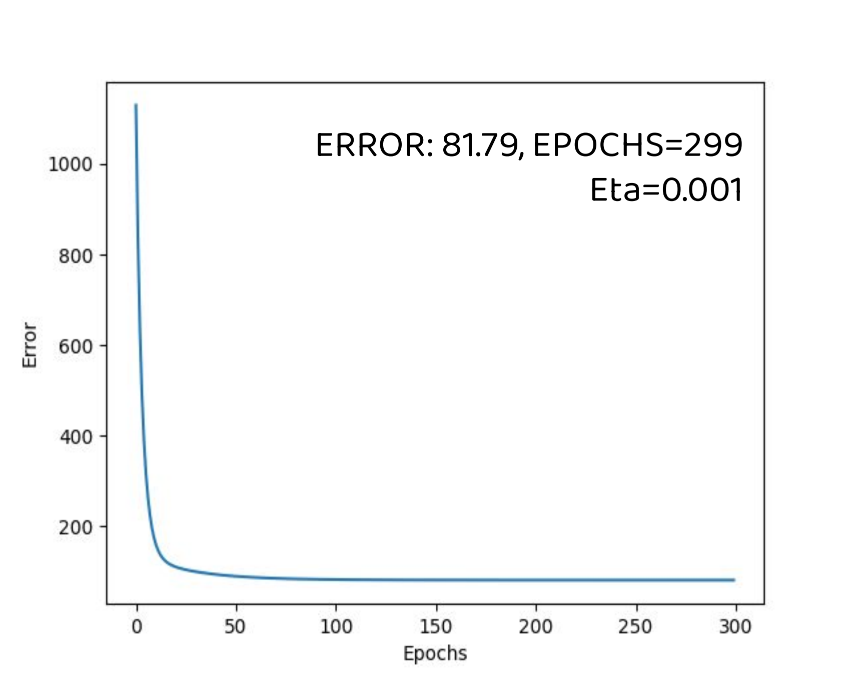



MAX EPOCHS = 100

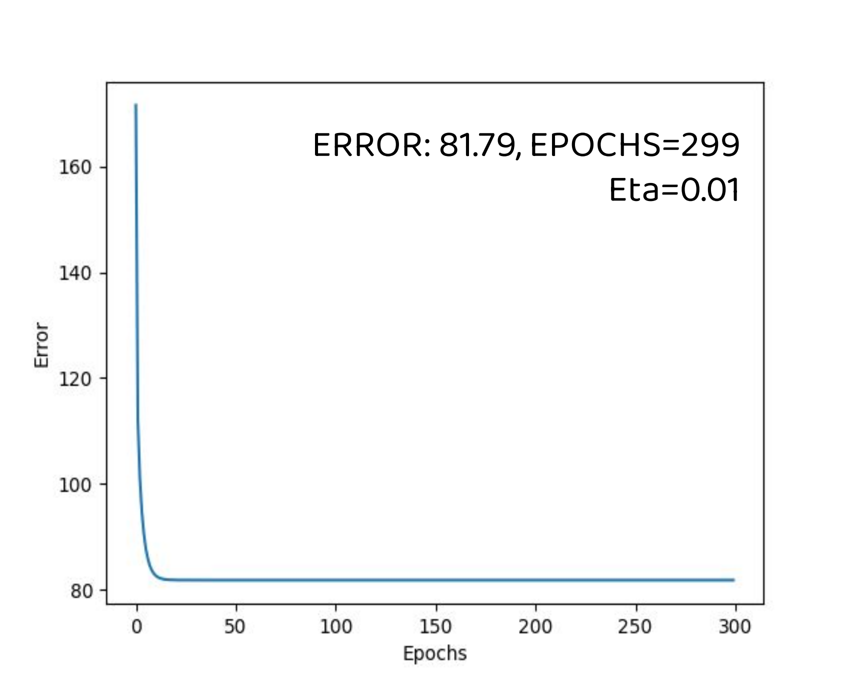
# Variación de Eta (3/4)

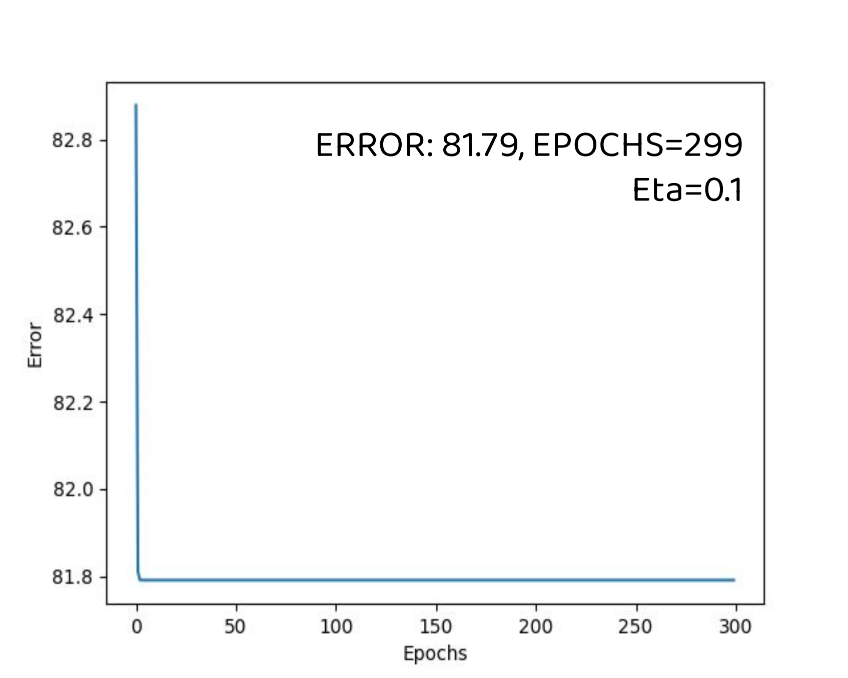



MAX EPOCHS = 300

# Variación de Eta (4/4)




MAX EPOCHS = 300

# RESUMEN DE DATOS

| MAX EPOCHS = 100 | ETA | | | |
|---|---|---|---|---|
| | **0.0001** | **0.001** | **0.01** | **0.1** |
| **Error** | 176.24 | 83.04 | 81.79 | 81.79 |

AVG ERROR = 105.72

| MAX EPOCHS = 300 | ETA | | | |
|---|---|---|---|---|
| | **0.0001** | **0.001** | **0.01** | **0.1** |
| **Error** | 101.11 | 81.79 | 81.79 | 81.79 |

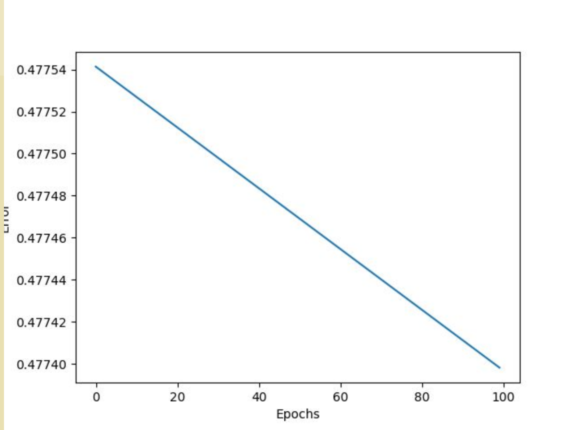
AVG ERROR = 86.62

# CONCLUSIONES

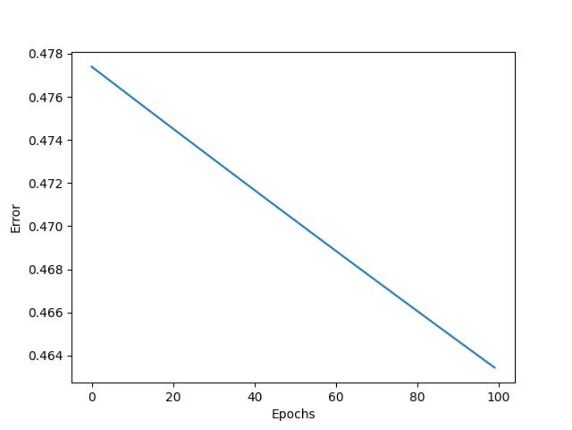
- El perceptrón lineal no llega a aprender lo suficiente con un eta pequeño en pocas épocas
- Cuanto mayor sea el eta, más rápido se llega a estancar en el error mínimo
- El dataset no es un problema linealmente independiente

# No Lineal

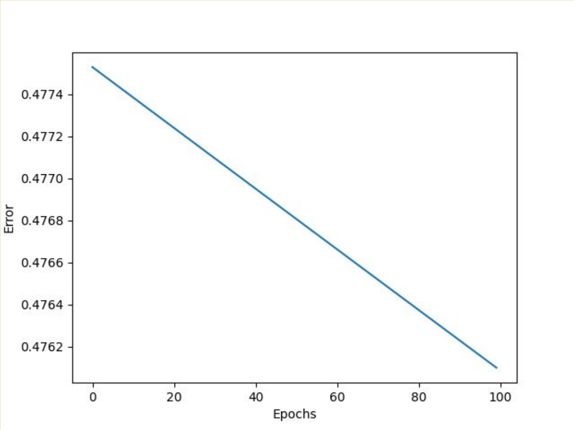
# Variación de Eta (beta = 0.01)



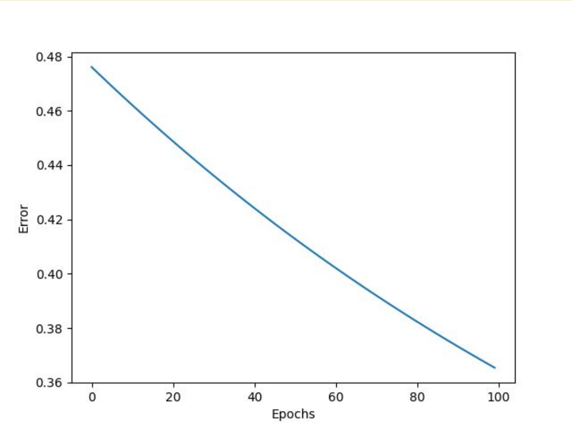

Eta = 0.0001
Error = 0.477




ETA = 0.01
Error = 0.463

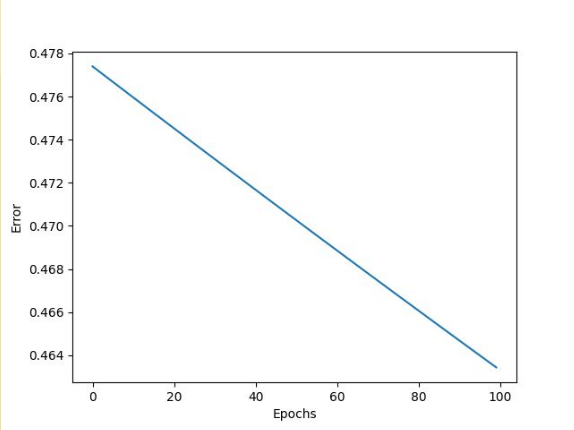



Eta = 0.001
Error: 0.476

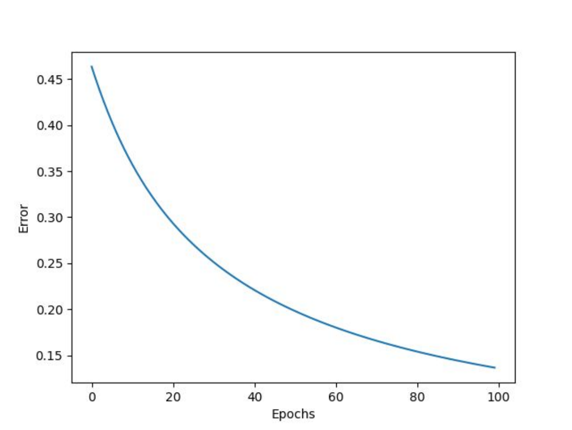



ETA = 0.01
Error = 0.365

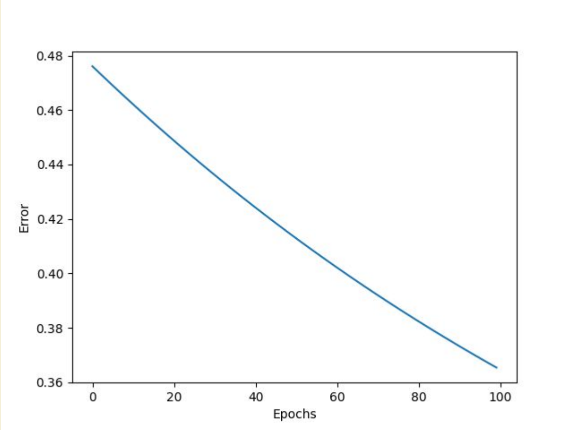
# Variación de Eta (beta = 0.1)



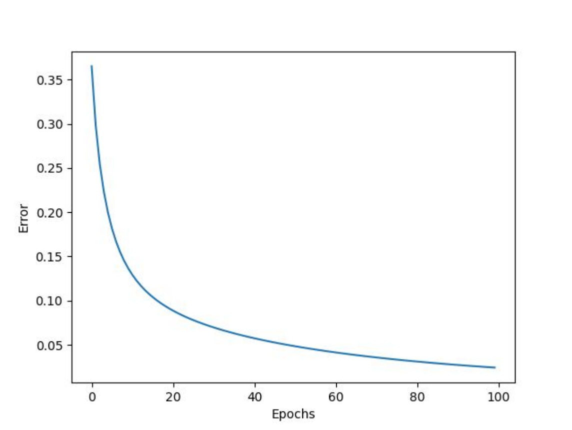

Eta = 0.0001
Error = 0.463




ETA = 0.01
Error = 0.136

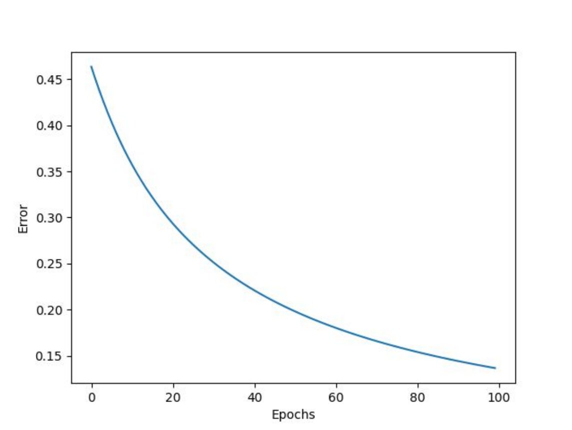



Eta = 0.001
Error: 0.365

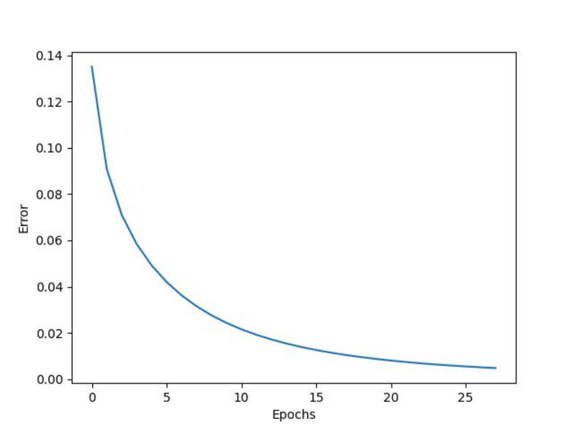



ETA = 0.1
Error = 0.0244

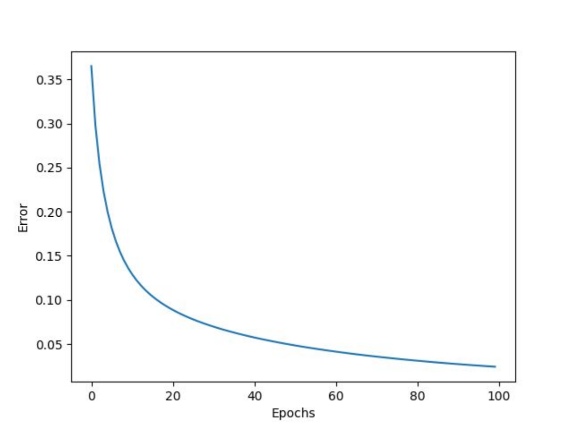
# Variación de Eta (beta = 1)



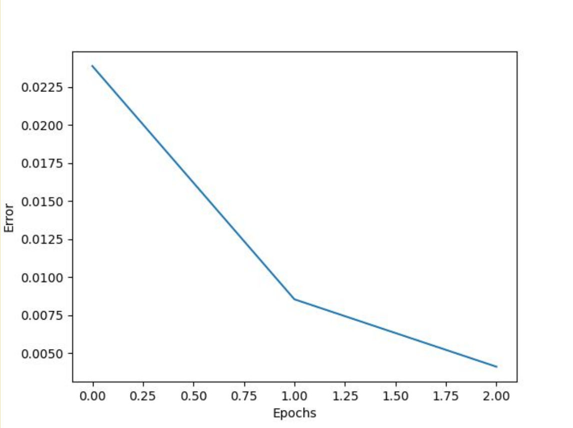

Eta = 0.0001
Error = 0.136




ETA = 0.01
Error = 0.00478

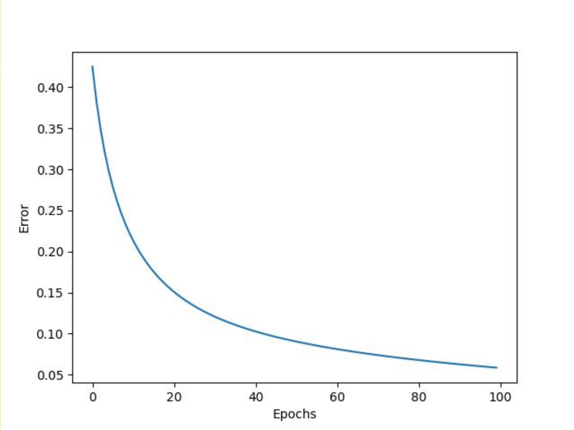



Eta = 0.001
Error: 0.0243

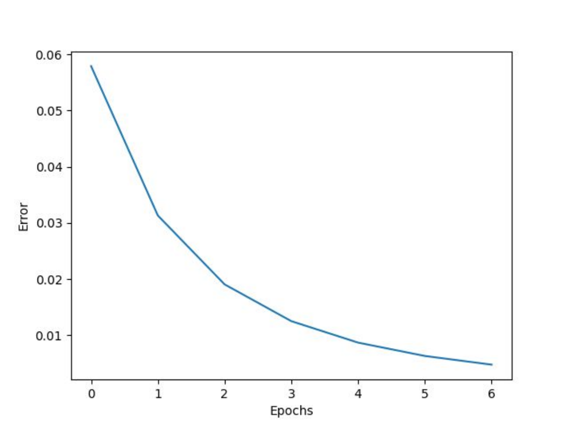



ETA = 0.1
Error = 0.00412

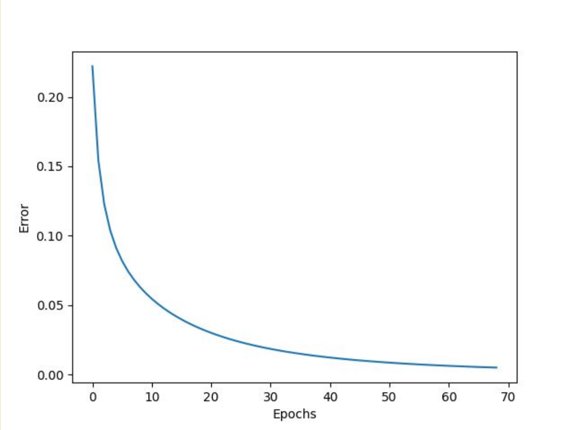
# Variación de Eta (beta = 2)




Eta = 0.0001
Error = 0.0585




ETA = 0.01
Error = 0.00475




Eta = 0.001
Error: 0.00493

ETA = 0.1
Error = 0.00629 (llega en una sola epoch)

# RESUMEN DE DATOS (1/2)

| MAX EPOCHS = 100<br>BETA = 0.01 | ETA | | | |
|---|---|---|---|---|
| | **0.0001** | **0.001** | **0.01** | **0.1** |
| **Error** | 0.477 | 0.476 | 0.463 | 0.365 |

AVG ERROR = 0.445

| MAX EPOCHS = 100<br>BETA = 0.1 | ETA | | | |
|---|---|---|---|---|
| | **0.0001** | **0.001** | **0.01** | **0.1** |
| **Error** | 0.463 | 0.365 | 0.136 | 0.0244 |

AVG ERROR = 0.247

# RESUMEN DE DATOS (2/2)

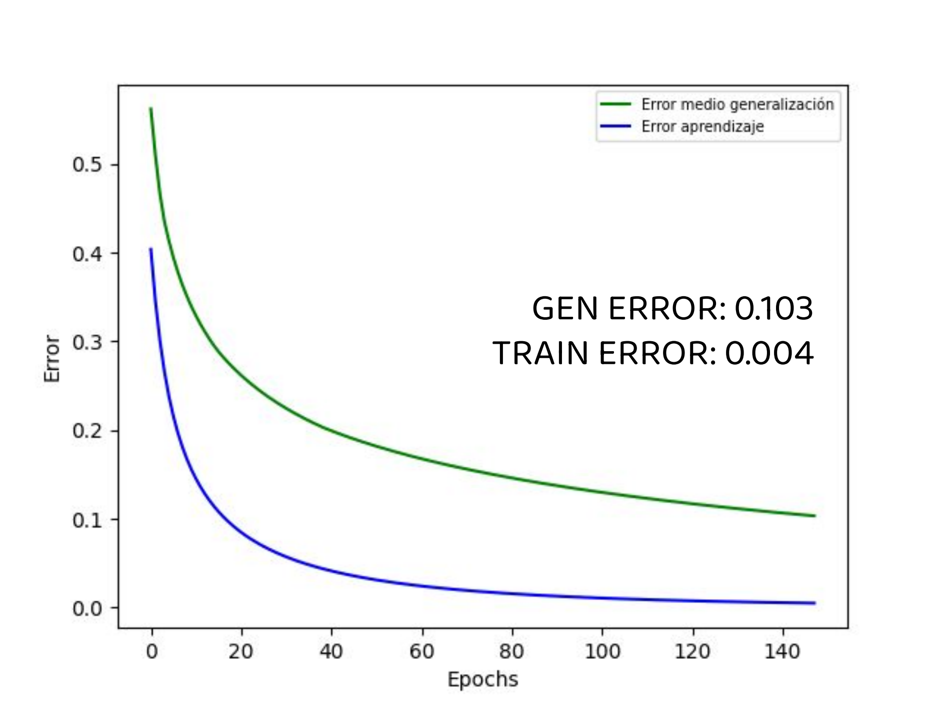
| MAX EPOCHS = 100 BETA=1 | ETA | | | |
|---|---|---|---|---|
| | **0.0001** | **0.001** | **0.01** | **0.1** |
| **Error** | 0.136 | 0.0243 | 0.00478 | 0.00412 |

AVG ERROR = 0.169

| MAX EPOCHS = 100 BETA=2 | ETA | | | |
|---|---|---|---|---|
| | **0.0001** | **0.001** | **0.01** | **0.1** |
| **Error** | 0.0585 | 0.00493 | 0.00475 | 0.00629 |

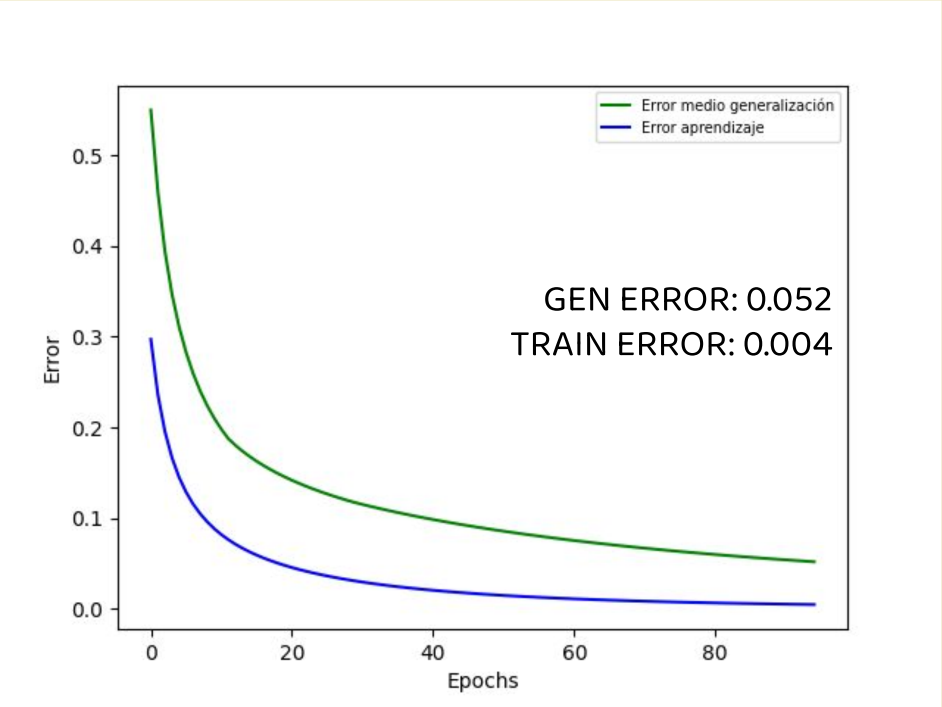
AVG ERROR = 0.0227

# Capacidad de Generalización (1/3)
# 25% aprendizaje, 75% testeo



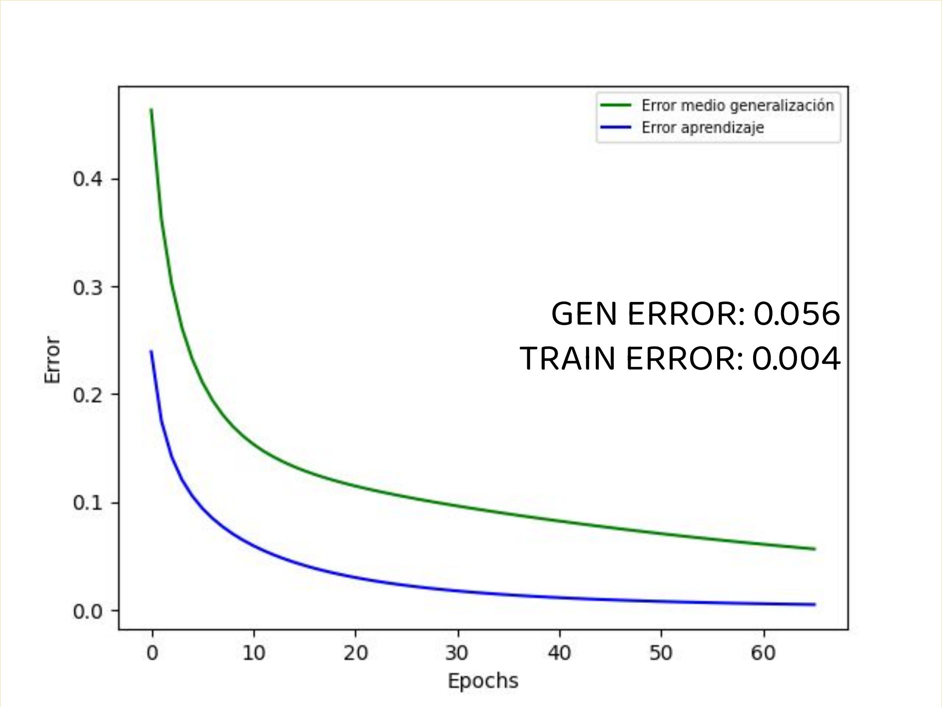

MAX EPOCHS = 1000
Beta = 1
Eta = 0.005
Error min = 0.005

# Capacidad de Generalización (2/3)
# 50% aprendizaje, 50% testeo




MAX EPOCHS = 1000
Beta = 1
Eta = 0.005
Error min = 0.005

# Capacidad de Generalización (3/3)
# 75% aprendizaje, 25% testeo




MAX EPOCHS = 1000
Beta = 1
Eta = 0.005
Error min = 0.005

# CONCLUSIONES

Con pocos datos de aprendizaje le cuesta generalizar

Con el 50% de datos utilizados para aprender , la generalización mejora

Con un exceso de datos para aprender, se pierde un poco esa capacidad para generalizar

# Selección de datos para entrenar

Siempre el objetivo es que el training set sea un subset representativo de la problemática que se está trabajando. Formarlo azarosamente puede causar que el modelo no aprenda.

- Problemas de Clasificación: se busca que en el training set estén presentes todas las clases del conjunto de salidas. (Ej. imágenes de perros y gatos)

- Problemas Lineales: de la misma manera, las etiquetas del training set deberían ser lo más diversas posibles para que el modelo gane capacidad de generalización.

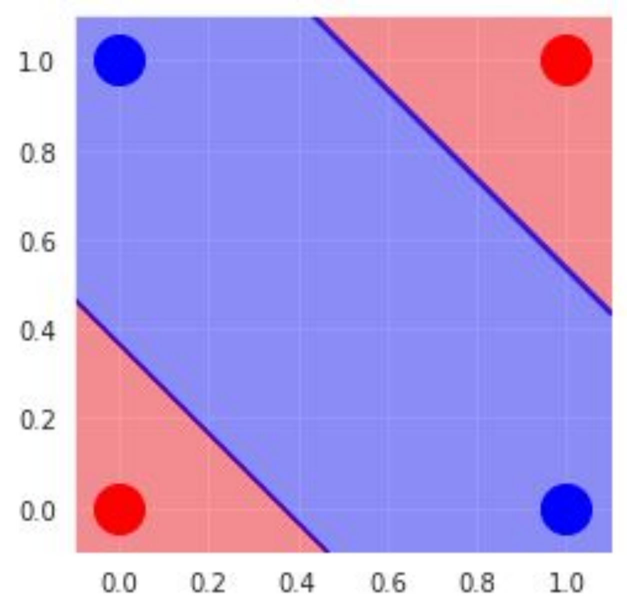
02

# Multicapa

Gradiente Descendiente

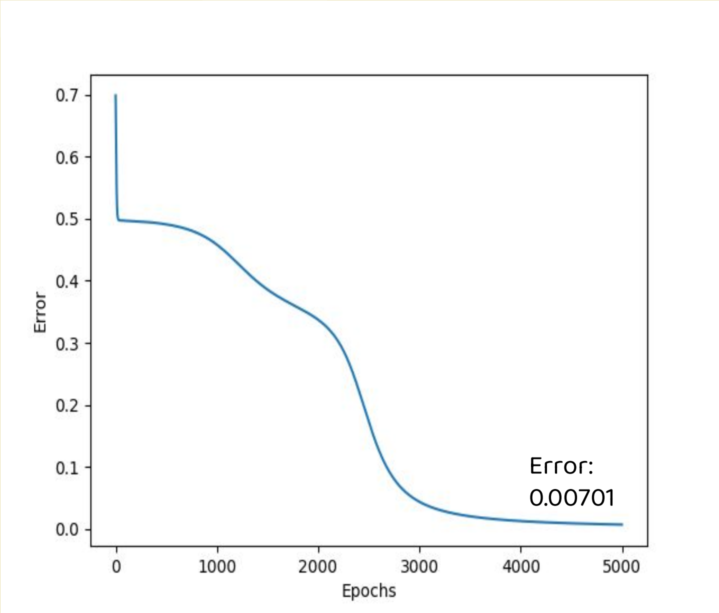
# XOR

# Sigmoide (1/2)

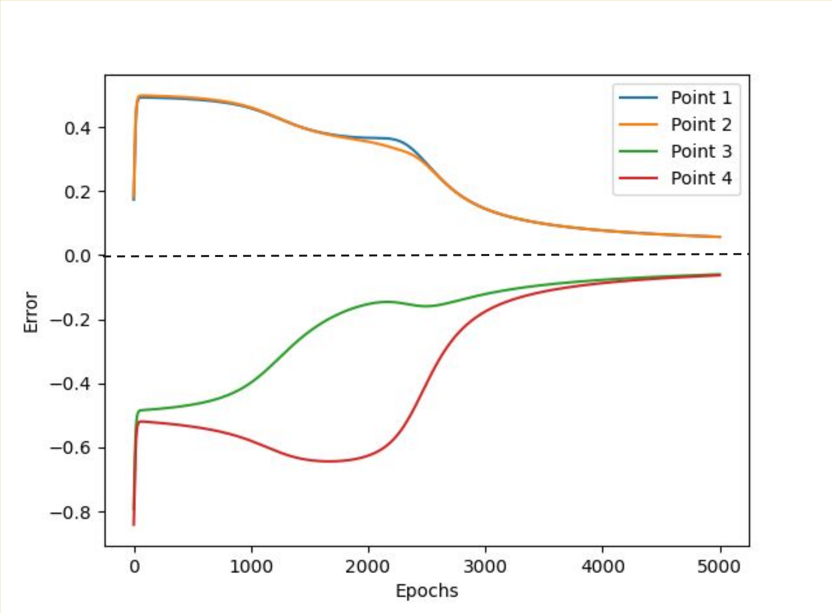



1 capa oculta con 5 nodos

# Sigmoide (2/2)

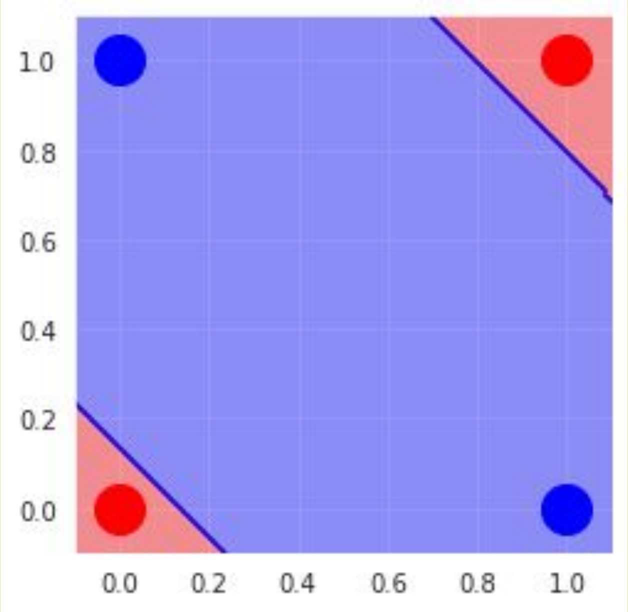


Error
Epochs
Error:
0.00701

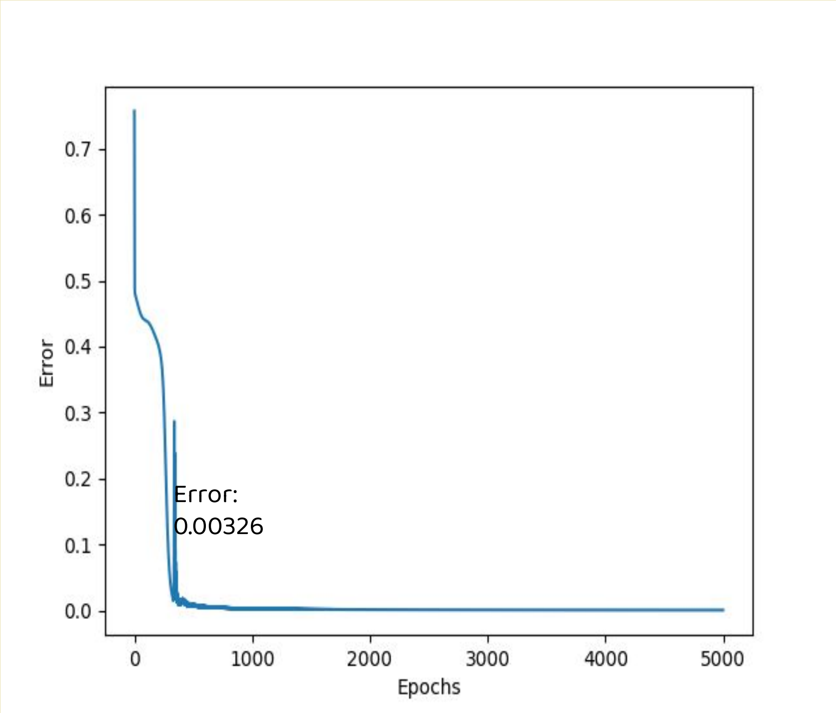
Point 1
Point 2
Point 3
Point 4
Error
Epochs

# Tanh (1/2)




1 capa oculta con 5 nodos

# Tanh (2/2)



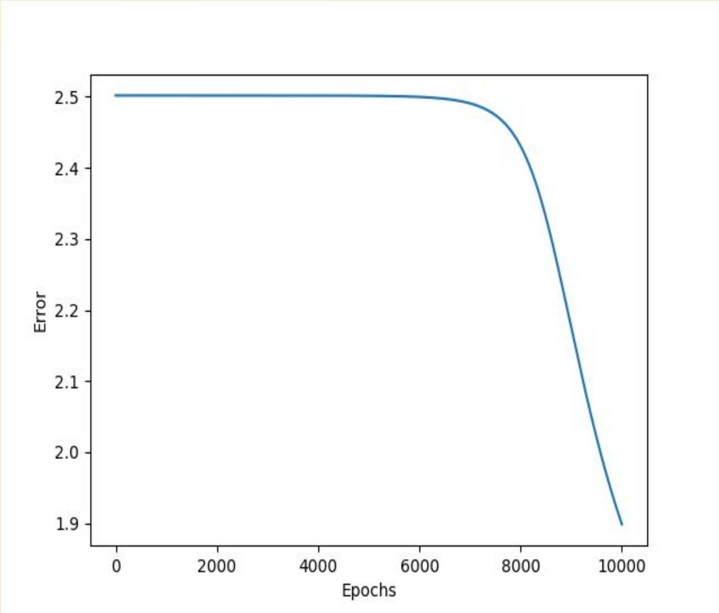
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Error
0
1000
2000
3000
4000
5000
Epochs
Error:
0.00326

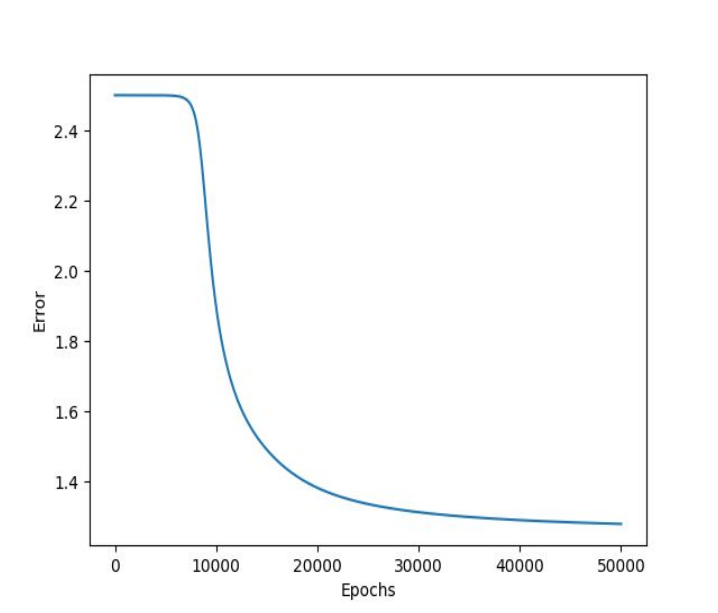
# CONCLUSIONES

Se puede resolver el XOR ya que el perceptrón multicapa permite representar dos rectas de separación.

# Paridad

# Variación de Arquitectura (1/4)



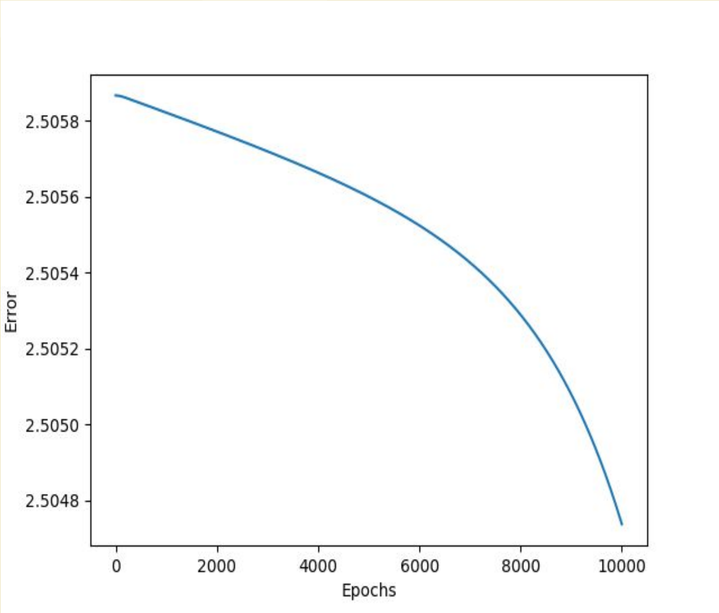



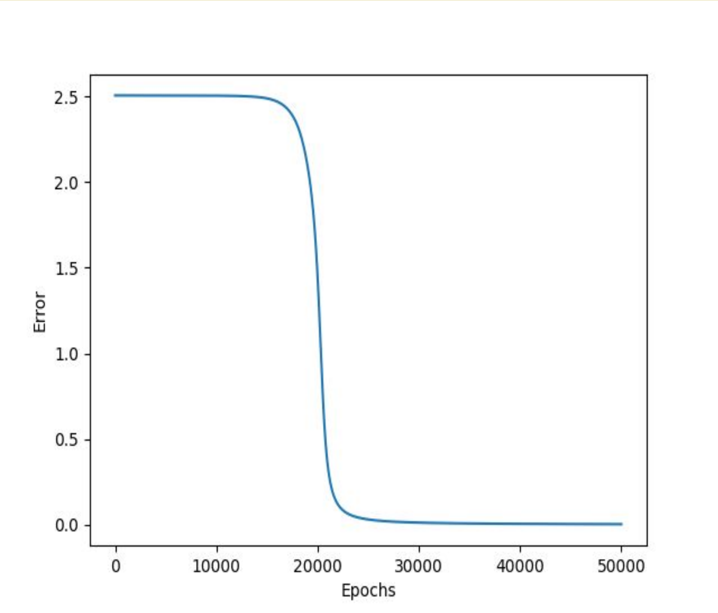

Error = 1.90

Error = 1.28

1 capa oculta con 5 neuronas
eta = 0.01

# Variación de Arquitectura (2/4)



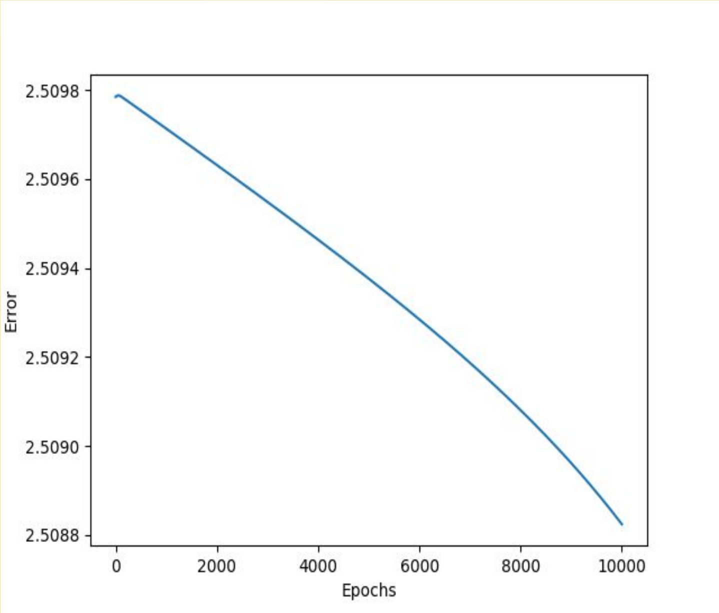



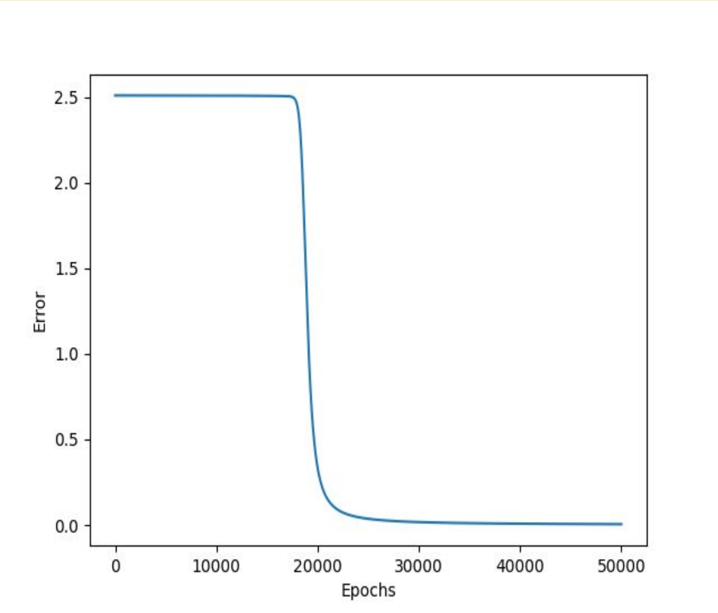

Error = 2.504

Error = 0.00319

1 capa oculta con 15 neuronas
eta = 0.01

# Variación de Arquitectura (3/4)




Error = 2.51

Error = 0.00493

1 capa oculta con 25 neuronas
eta = 0.01

# RESUMEN DE DATOS

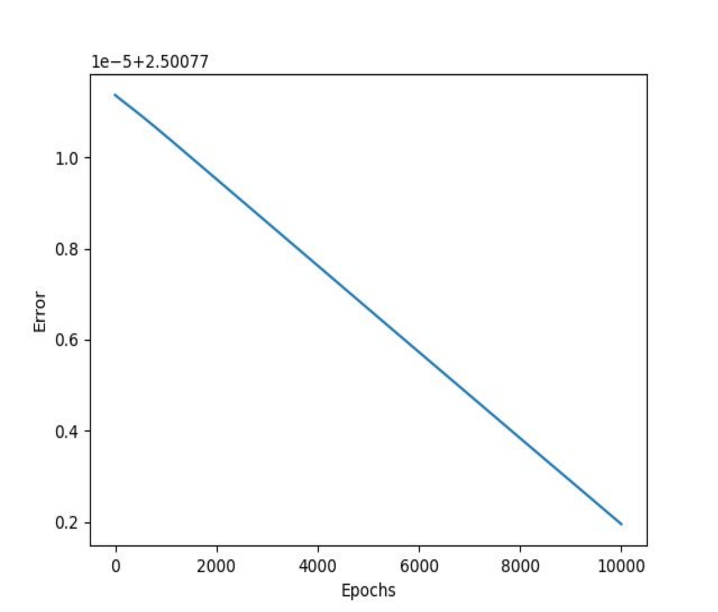
| MAX EPOCHS = 10000<br>ETA = 0.01 | Cantidad de Neuronas (En capa oculta) | | |
|---|---|---|---|
| | **5** | **15** | **25** |
| **Error** | 1.90 | 2.504 | 2.51 |

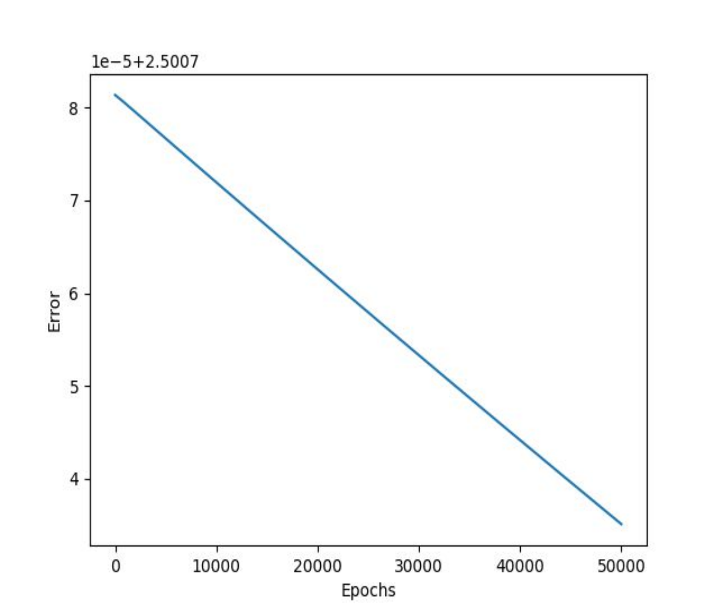
AVG ERROR = 0.445

| MAX EPOCHS = 50000<br>ETA = 0.01 | Cantidad de Neuronas (En capa oculta) | | |
|---|---|---|---|
| | **5** | **15** | **25** |
| **Error** | 1.28 | 0.00319 | 0.00493 |

AVG ERROR = 0.247

# Variación de Arquitectura (4/4)




Error = 0.25008

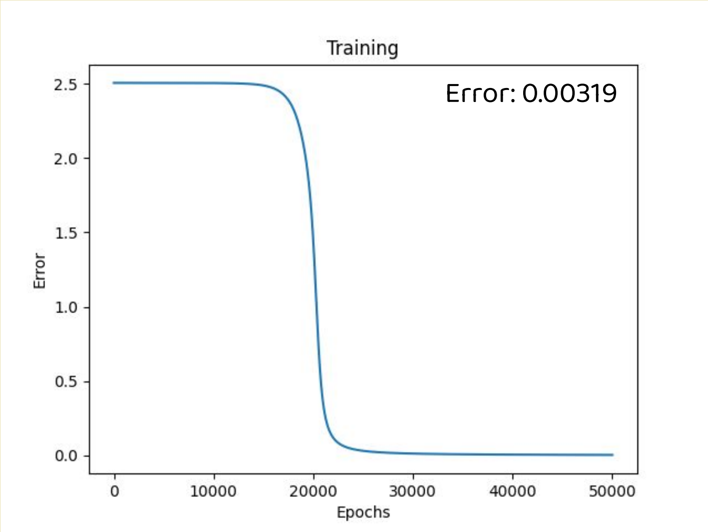
Error = 0.25008

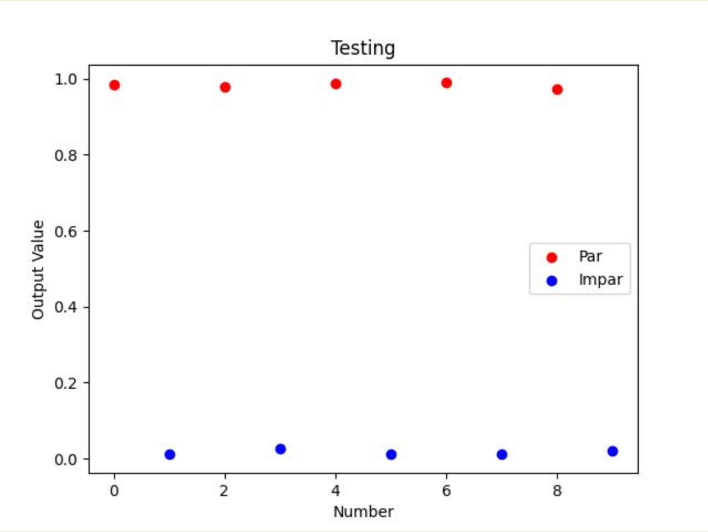
2 capas oculta con 5 neuronas
eta = 0.01

# Capacidad de Generalización (1/4)

El perceptrón define una función output y a cada clase se le asigna un color:

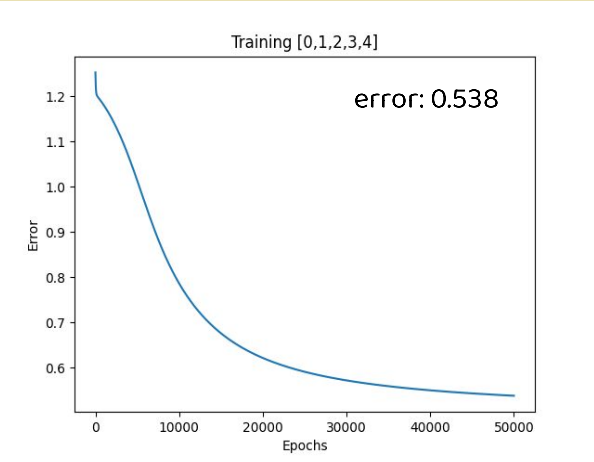
$$\begin{cases} output(x) \geq 0.5 & si\ x\ par \\ output(x) < 0.5 & si\ x\ impar \end{cases}$$

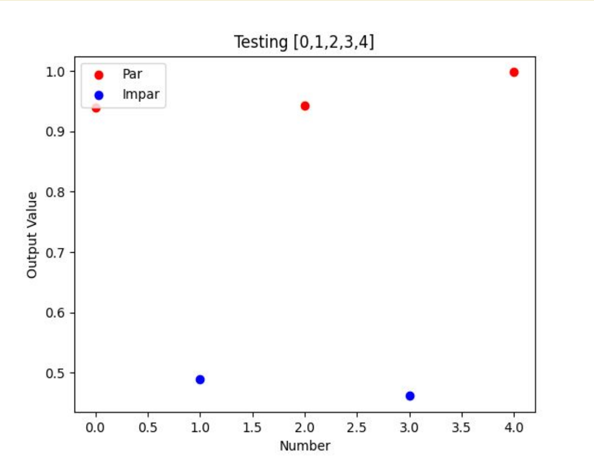


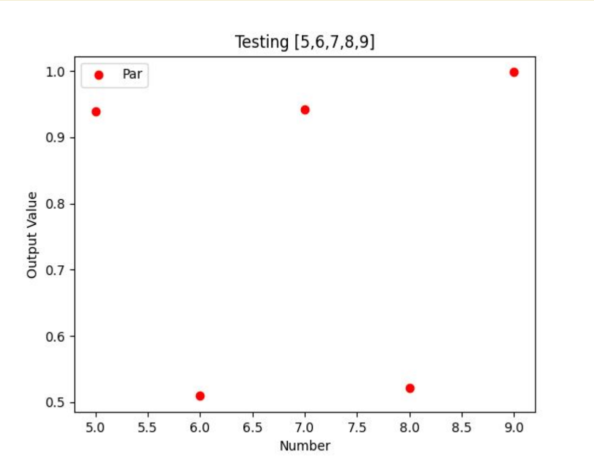




training y testing con el conjunto completo
[0,1,2,3,4,5,6,7,8,9]

# Capacidad de Generalización (2/4)

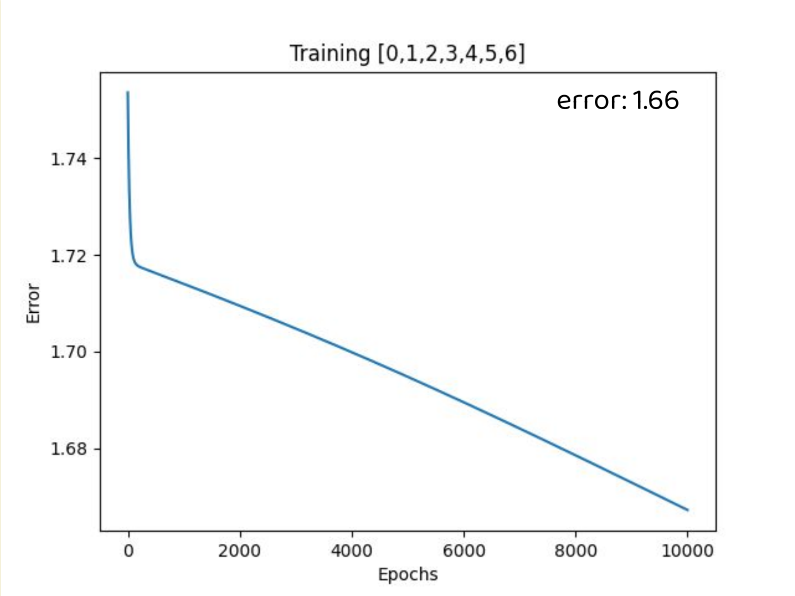


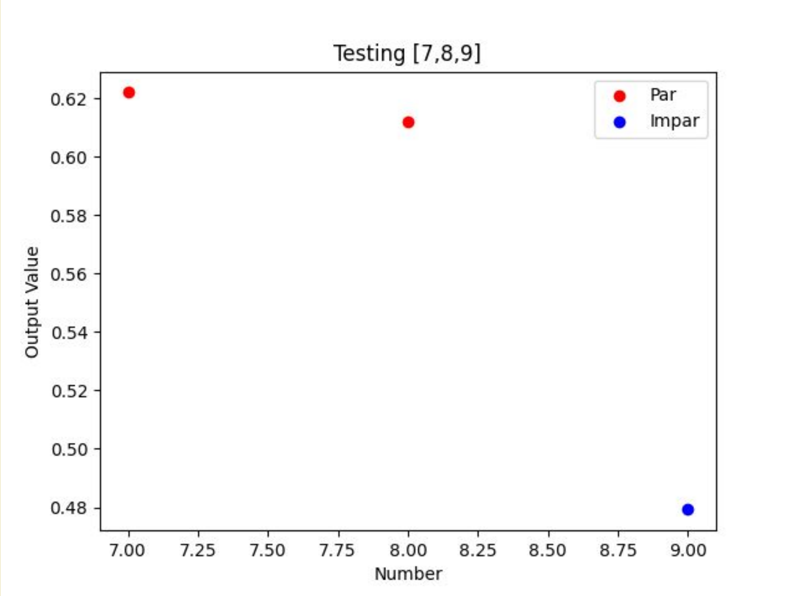




Se dividió el conjunto de inputs en 2: los primeros 5 números fueron usados para entrenamiento.

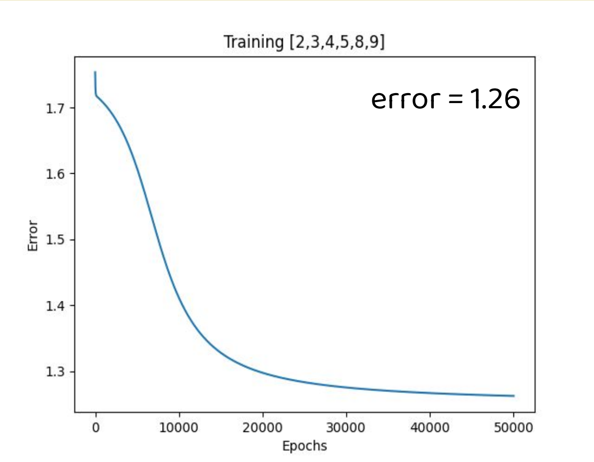
# Capacidad de Generalización (3/4)

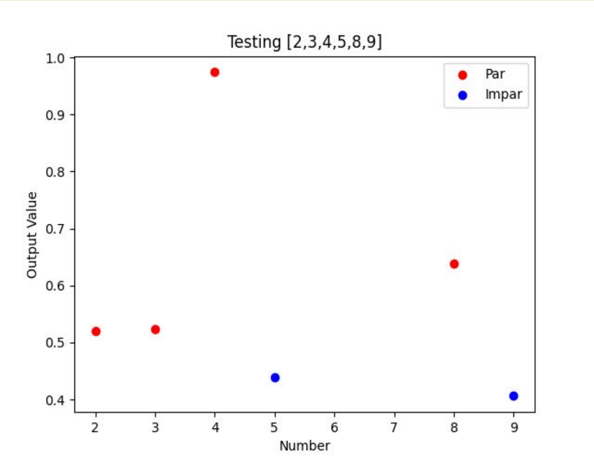


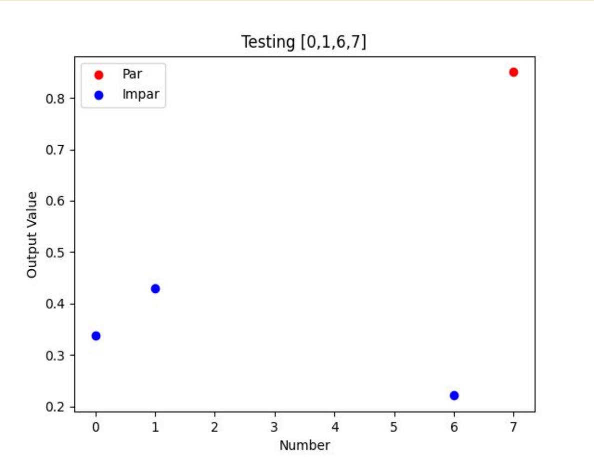




Se probaron diferentes configuraciones y tamaños al azar.

# Capacidad de Generalización (4/4)



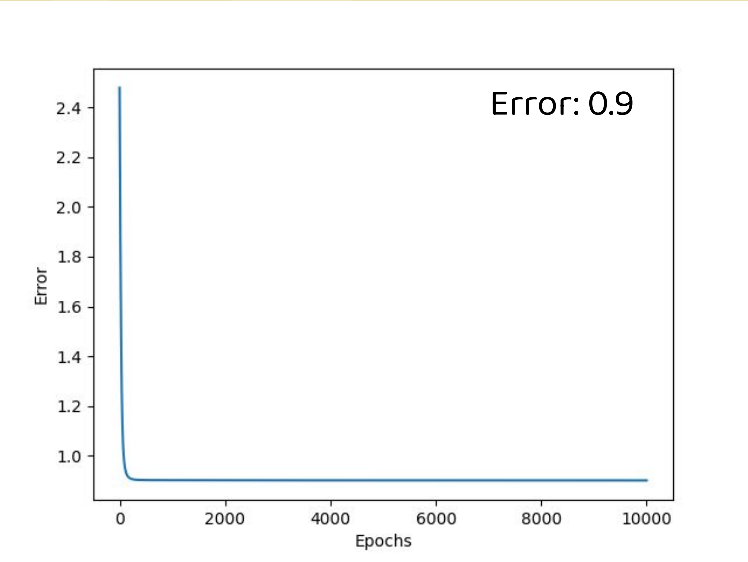



# CONCLUSIONES

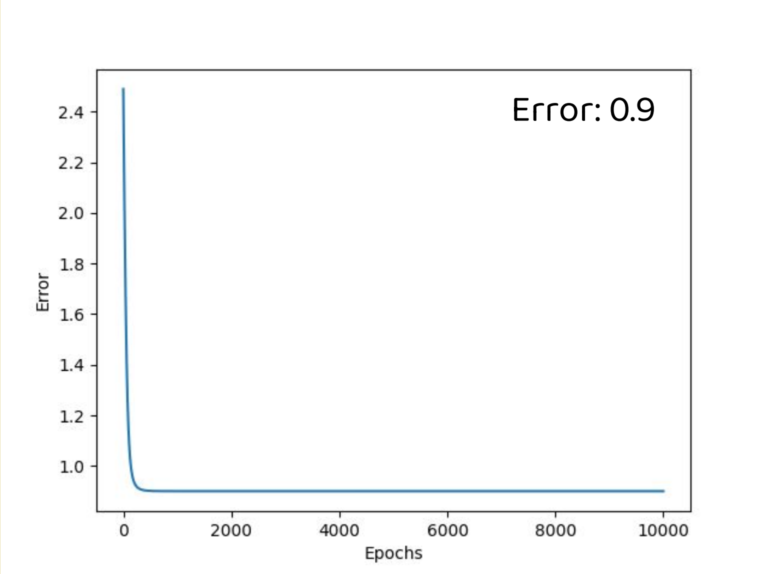
- La paridad de los números no es generalizable, al menos no con formato de imagen.
- Mayores epochs permiten mejor capacidad de aprendizaje.
- Cambiar la arquitectura no parece influir siempre de la misma forma. Agregar capas o neuronas tienen efectos diferentes dependiendo de otros factores

# Números

# Capacidad de Aprendizaje




5 capas, 10 neuronas




5 capas, 25 neuronas

# Gracias

**¿preguntas?**