TP3 - Perceptrón Simple y Multicapa

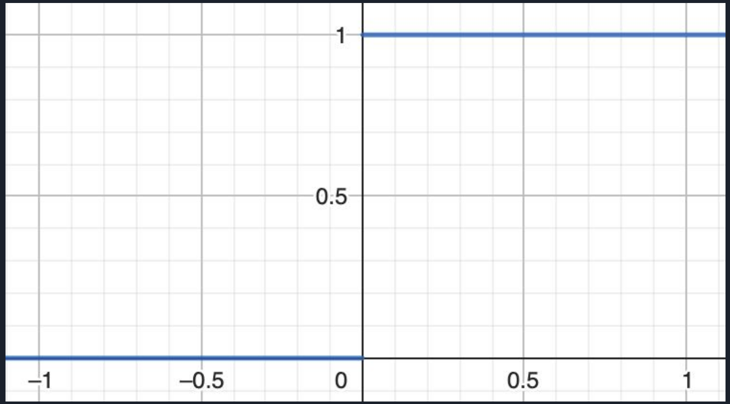
Grupo 1
Santiago José Hirsch
Matías Ignacio Luchetti
Santiago Tomás Medin
Mariano Agopian

Ejercicio 1

Perceptrón Simple

Parámetros

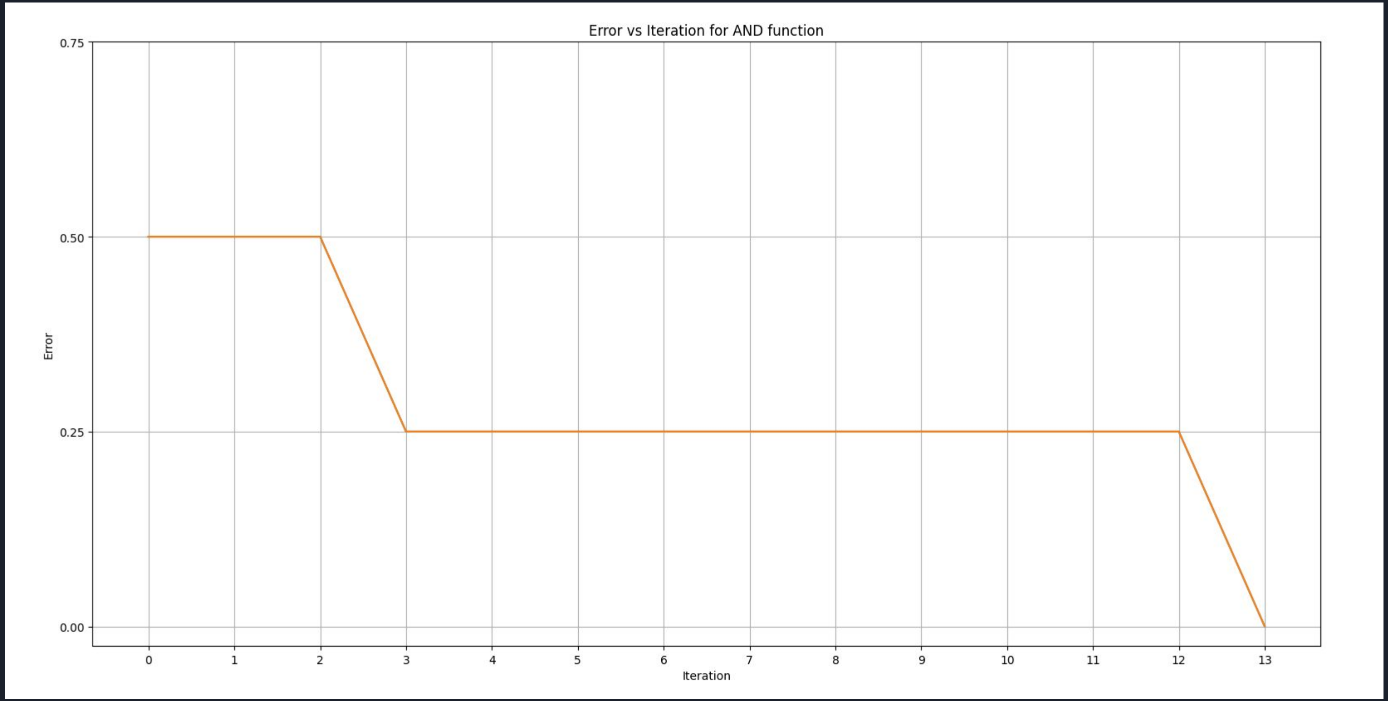
- Learning rate = 0.1
- Epochs = 200
- Epsilon = 0
- Función de activación = STEP



Problema AND

Parámetros

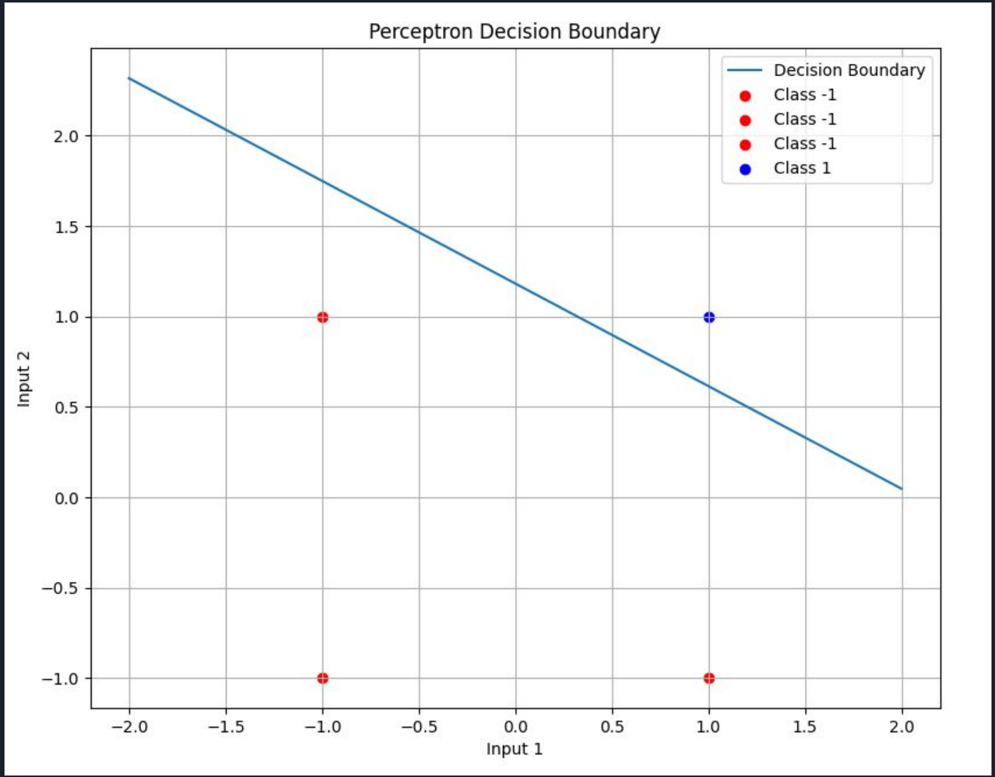
- Learning rate = 0.1
- Epochs = 200
- Epsilon = 0
- Función de activación = STEP



Decisión

Parámetros

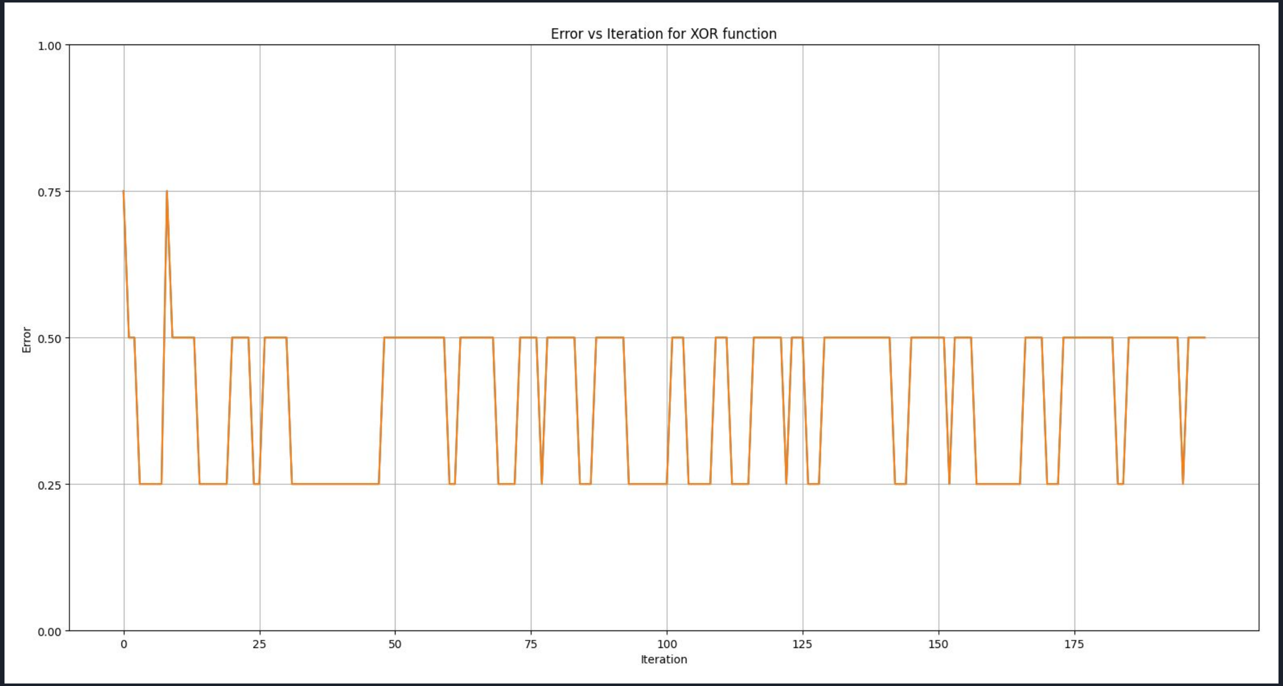
- Learning rate = 0.1
- Epochs = 200
- Epsilon = 0
- Función de activación = STEP



Problema XOR

Parámetros

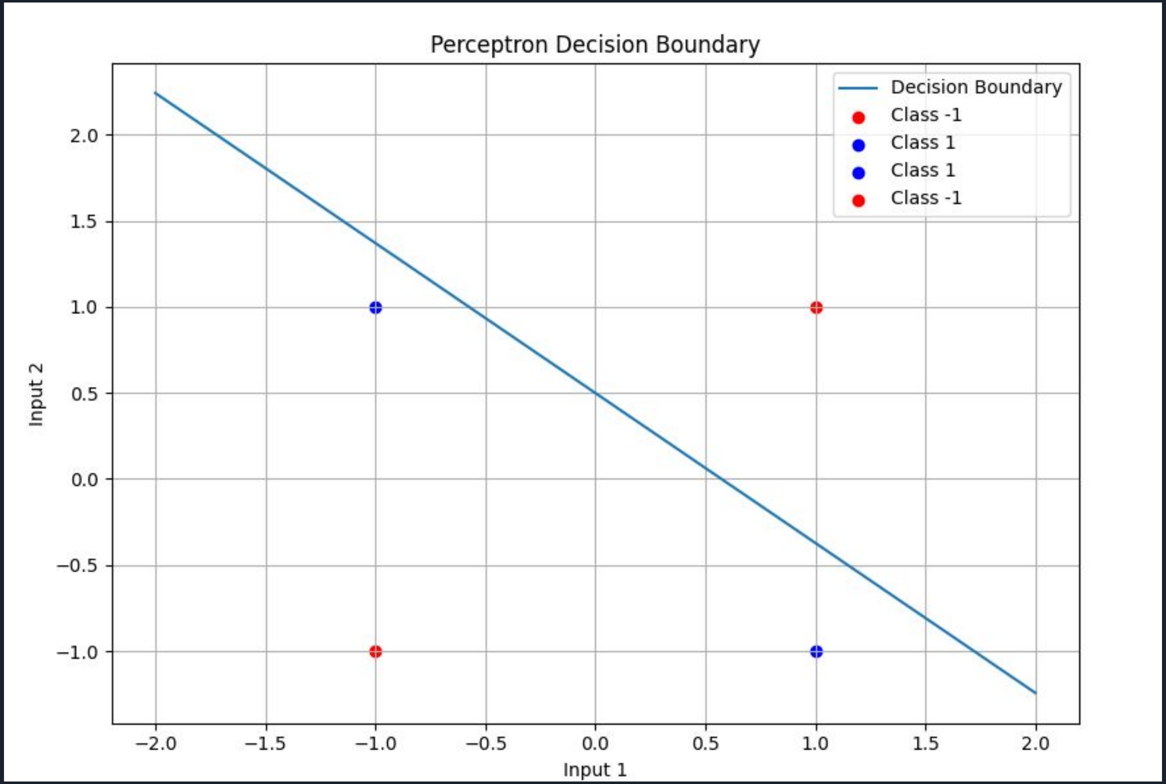
- Learning rate = 0.1
- Epochs = 200
- Epsilon = 0
- Función de activación = STEP



Decisión

Parámetros

- Learning rate = 0.1
- Epochs = 200
- Epsilon = 0
- Función de activación = STEP



Ejercicio 2

Lineal o no lineal, esa es la cuestión

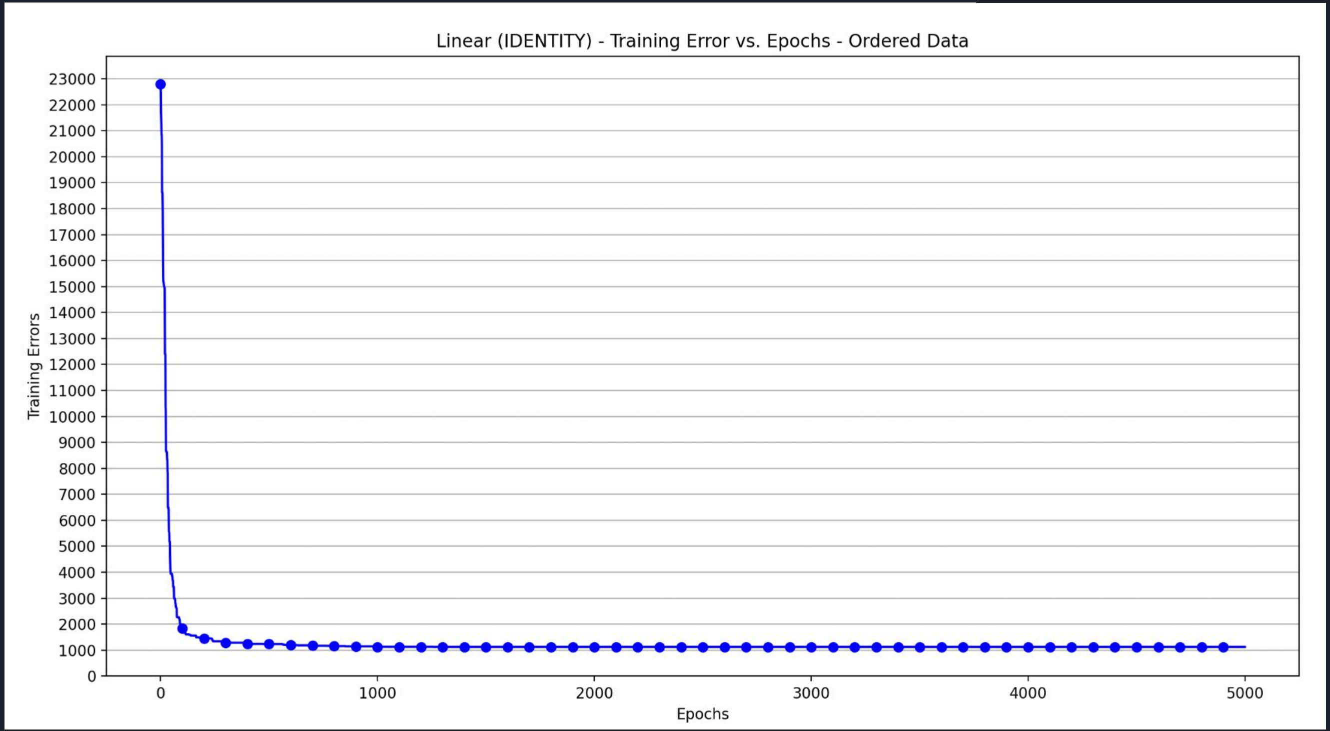
Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 5000
- Epsilon = 0
- Funcion de activacion (IDENTIDAD, SIGMOIDE, TANH)
- Beta = 1
- 100% training

Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 5000
- Epsilon = 0
- Funcion de activacion (IDENTIDAD, SIGMOIDE, TANH)
- Beta = 1
- 100% training

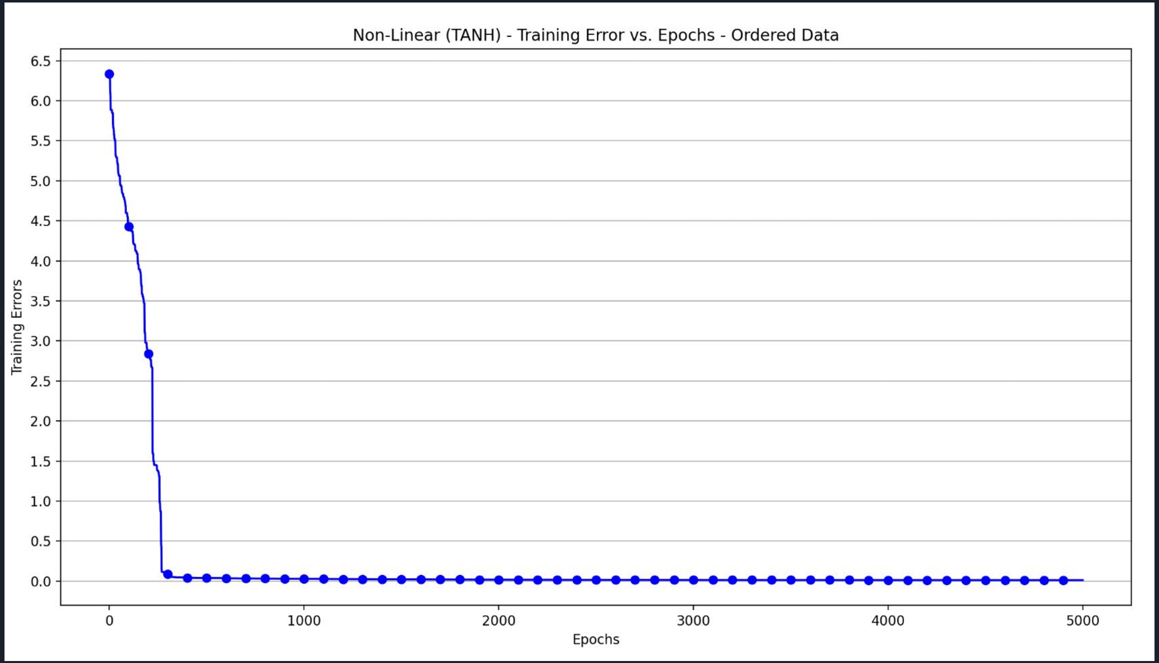
Lineal o no lineal



Lineal o no lineal

Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 5000
- Epsilon = 0
- Funcion de activacion (IDENTIDAD, SIGMOIDE, TANH)
- Beta = 1
- 100% training

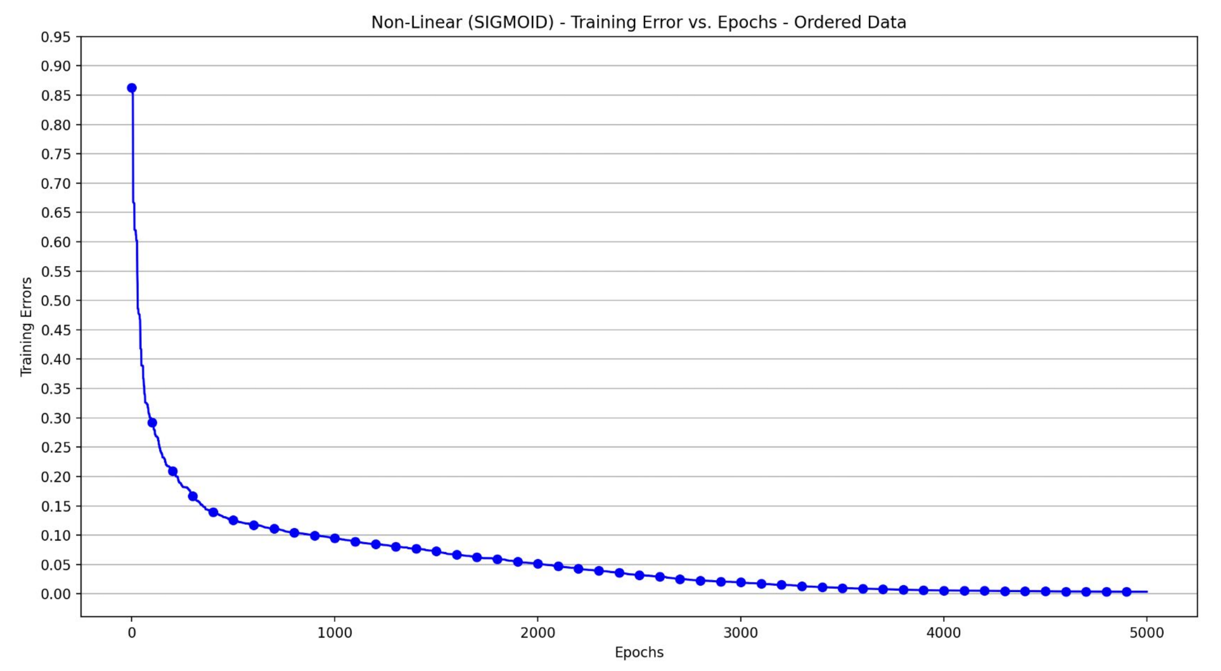


Mínimo: 0.0136

Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 5000
- Epsilon = 0
- Funcion de activacion (IDENTIDAD, SIGMOIDE, TANH)
- Beta = 1
- 100% training

Lineal o no lineal

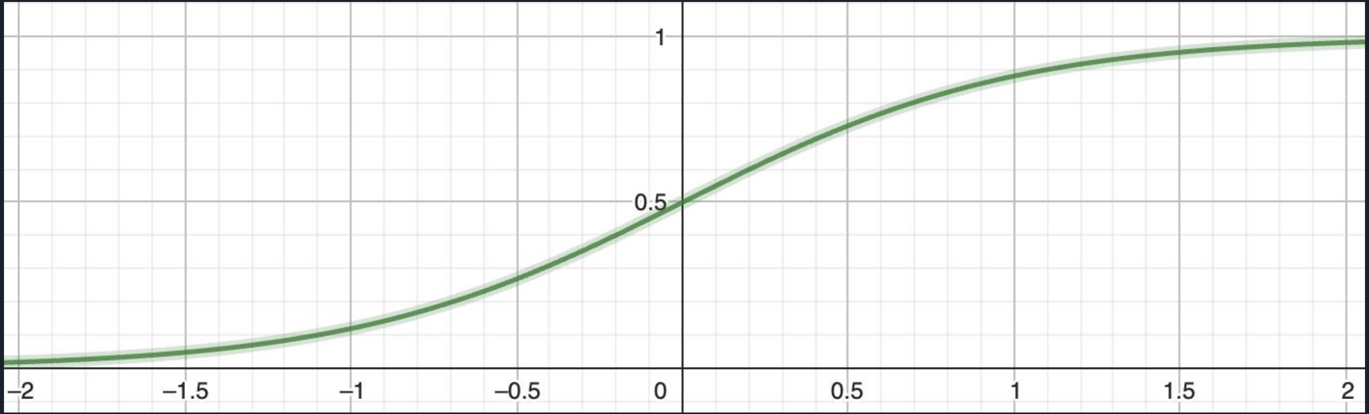


Mínimo: 0.00376

Lineal o no lineal - Función SIGMOIDE

$$\theta(h) = \frac{1}{1+exp^{-2\beta h}}$$

$$\theta'(h) = 2\beta\theta(h)(1 - \theta(h))$$



Datos ordenados vs datos mezclados

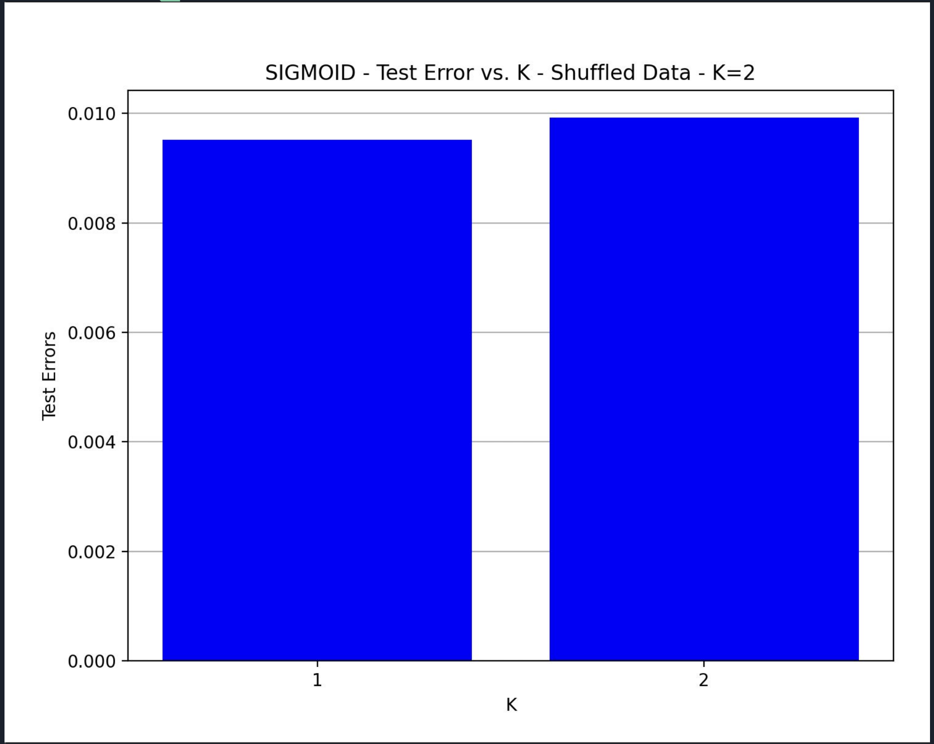
Parámetros:

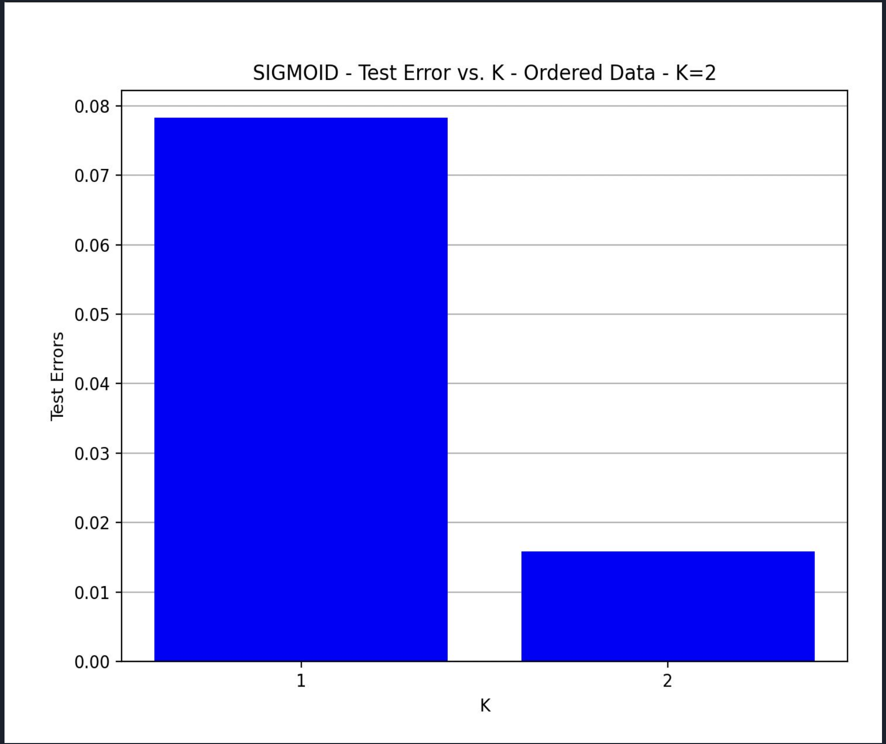
- Learning rate = 0.01
- Online
- Epochs = 5000
- Epsilon = 0.01
- Función de activación = sigmoide (datos interpolados entre 0 y 1)
- K-Fold (k entre 2 y 5)
- Beta = 1

Datos ordenados vs datos mezclados

Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 5000
- Epsilon = 0.01
- Función de activación = sigmoide
- K-Fold (k entre 2 y 5)
- Beta = 1

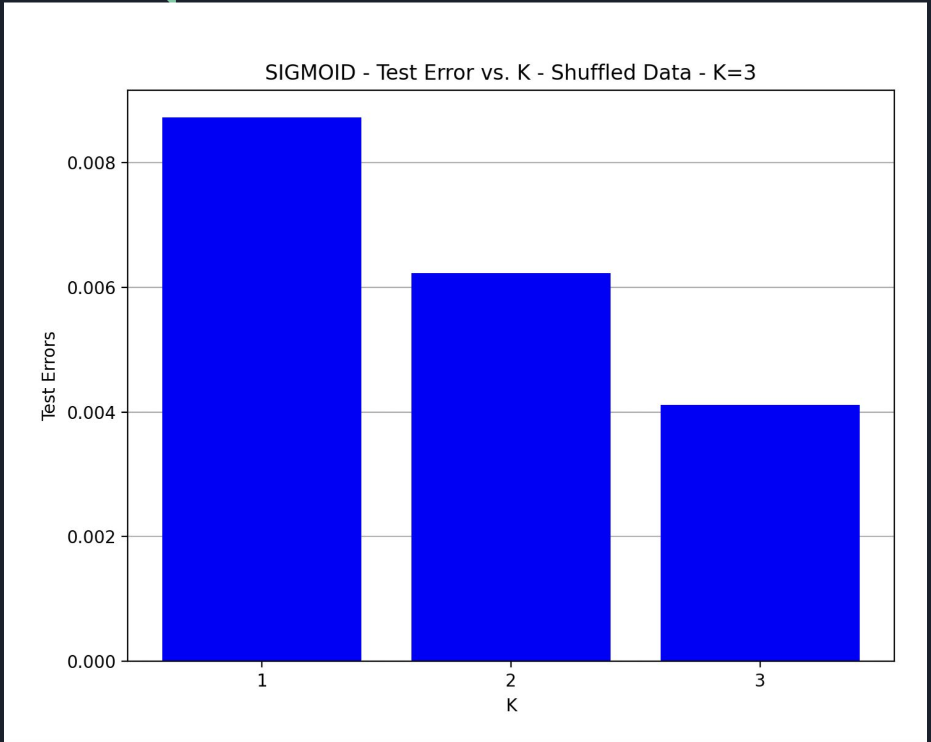


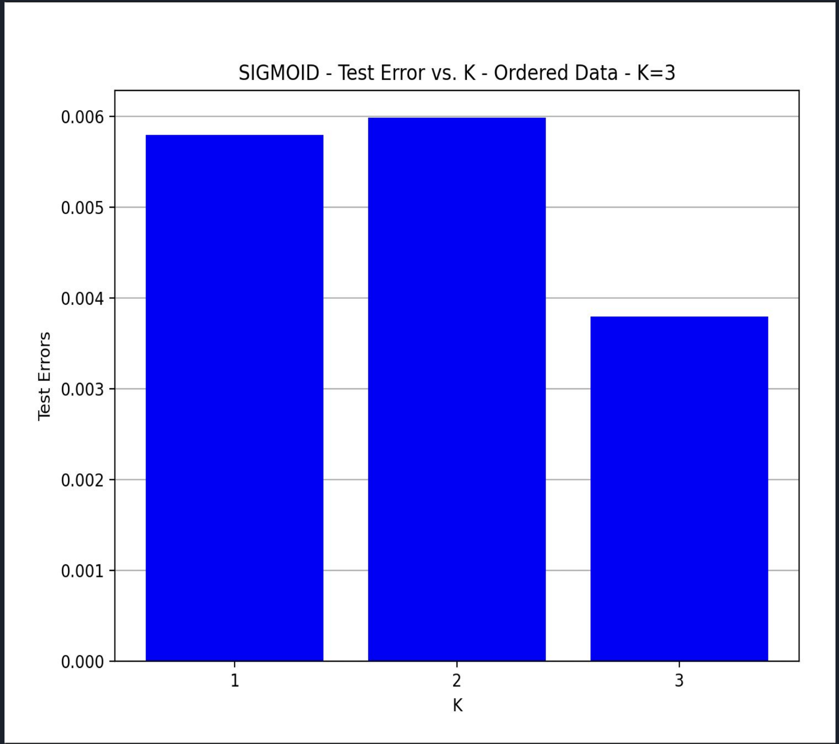


Datos ordenados vs datos mezclados

Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 5000
- Epsilon = 0.01
- Función de activación = sigmoide
- K-Fold (k entre 2 y 5)
- Beta = 1

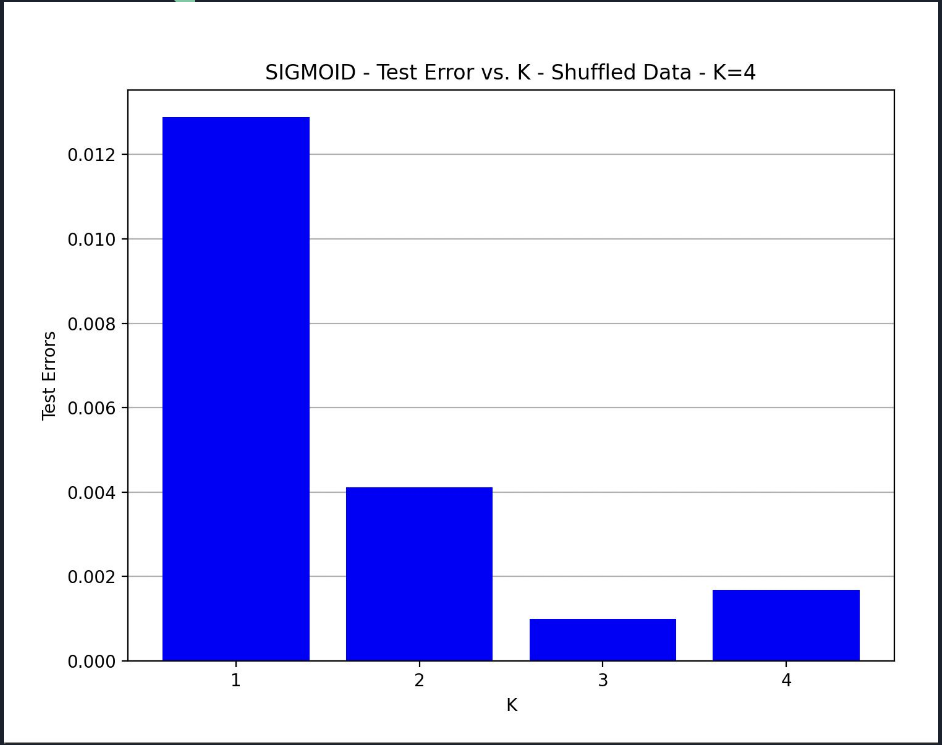


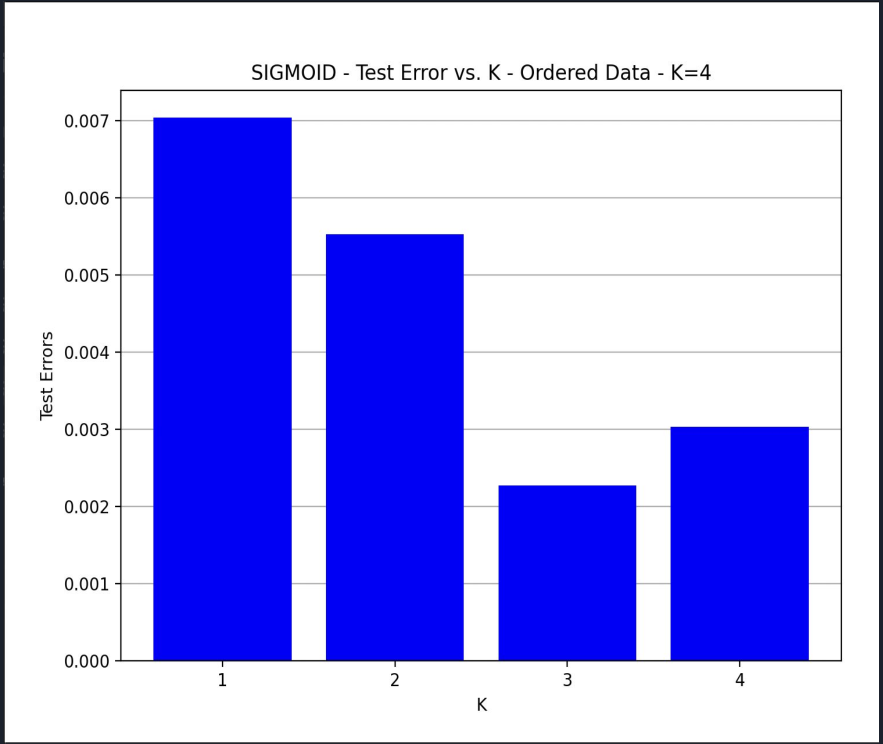


Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 5000
- Epsilon = 0.01
- Función de activación = sigmoide
- K-Fold (k entre 2 y 5)
- Beta = 1

Datos ordenados vs datos mezclados

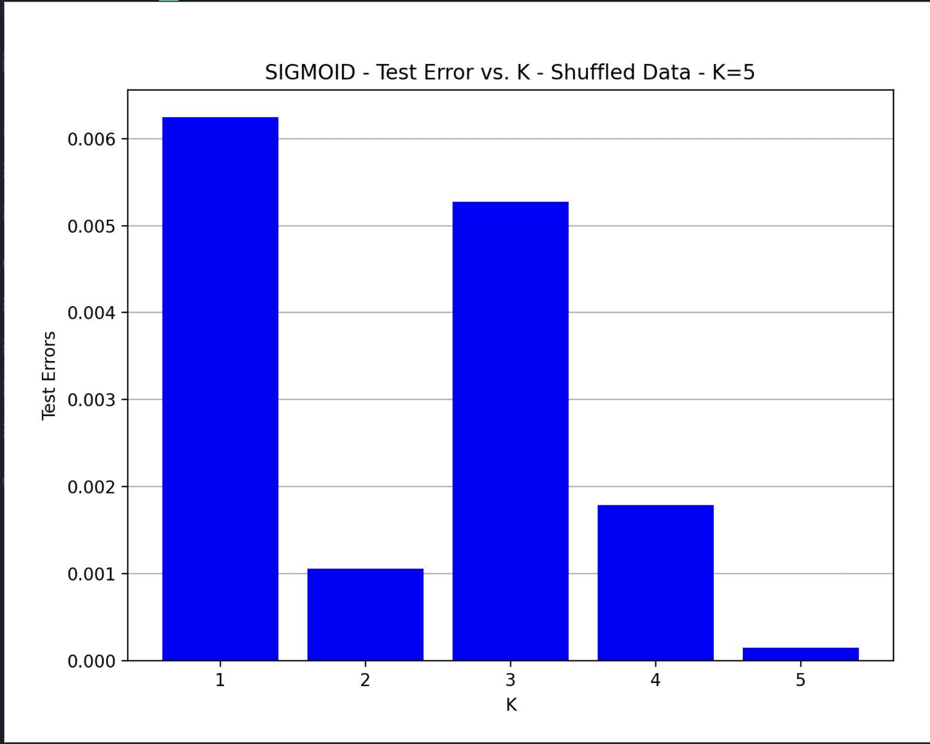


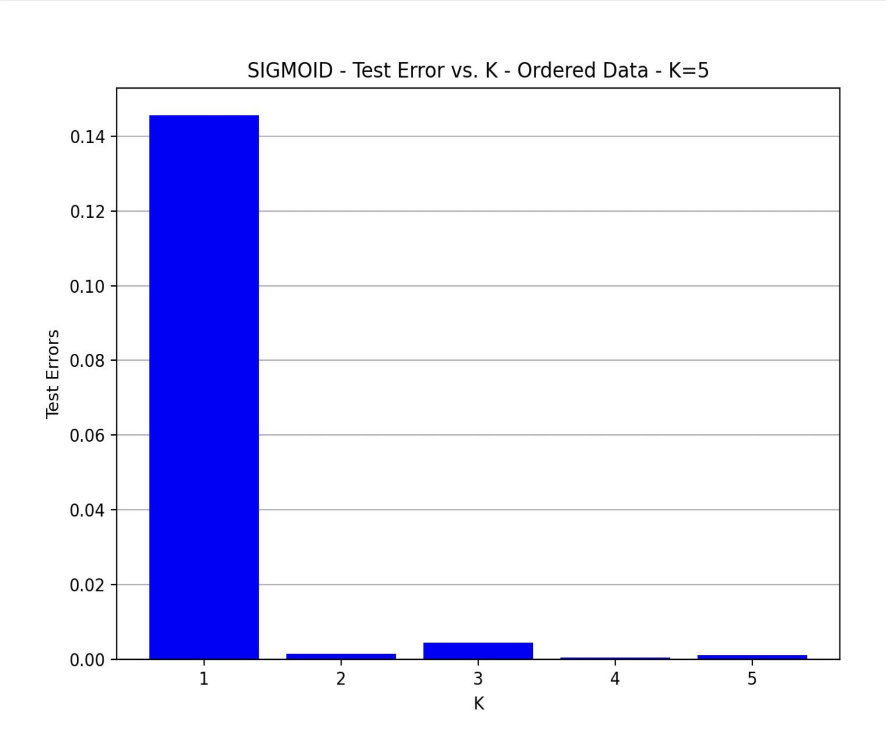


Datos ordenados vs datos mezclados

Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 5000
- Epsilon = 0.01
- Función de activación = sigmoide
- K-Fold (k entre 2 y 5)
- Beta = 1

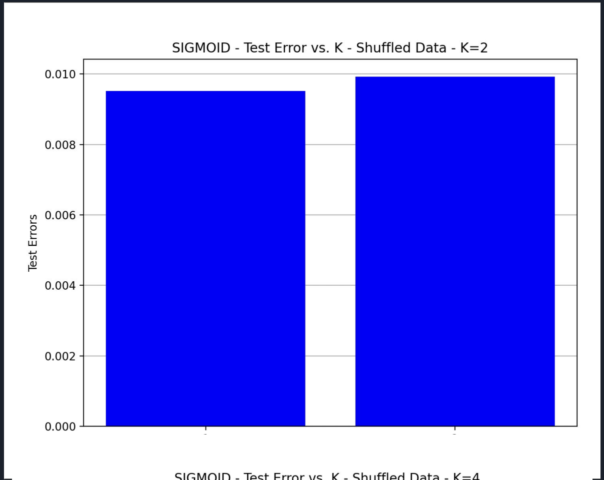


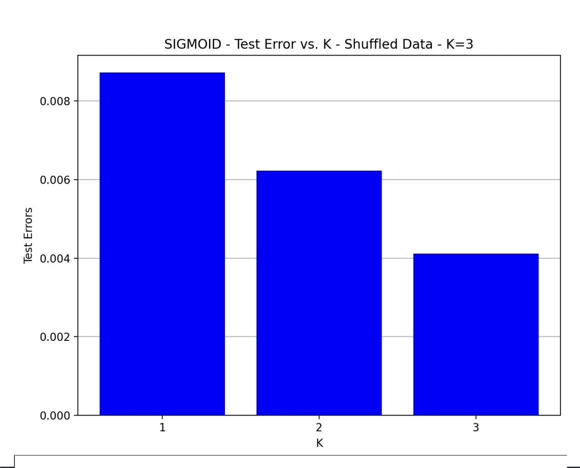


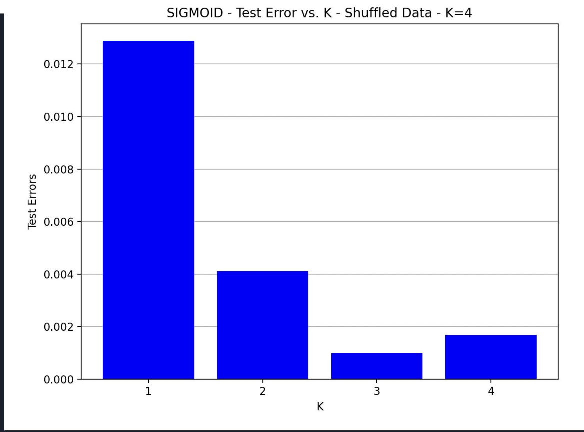
Datos ordenados vs datos mezclados

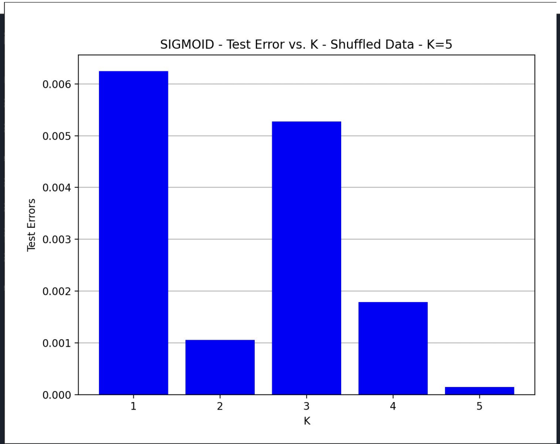
Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 5000
- Epsilon = 0.01
- Función de activación = sigmoide
- K-Fold (k entre 2 y 5)
- Beta = 1









Errores en función de las épocas

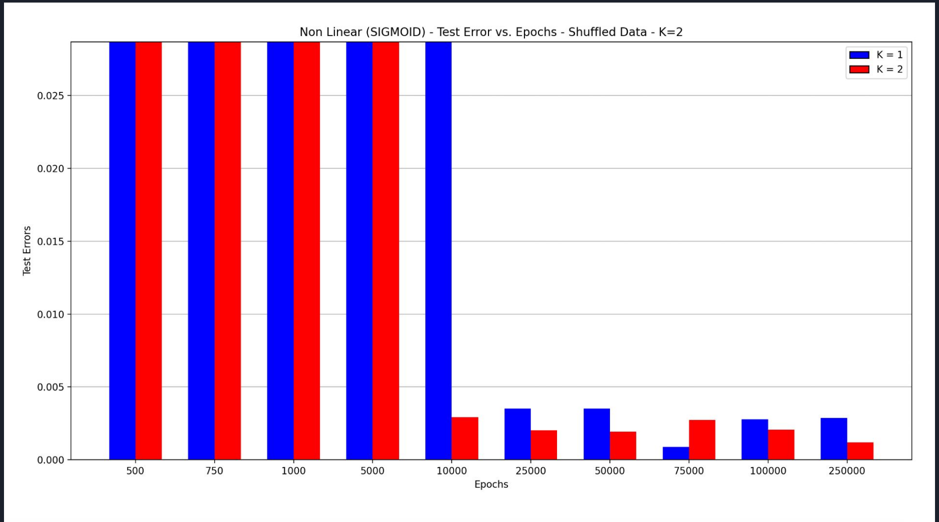
Parámetros:

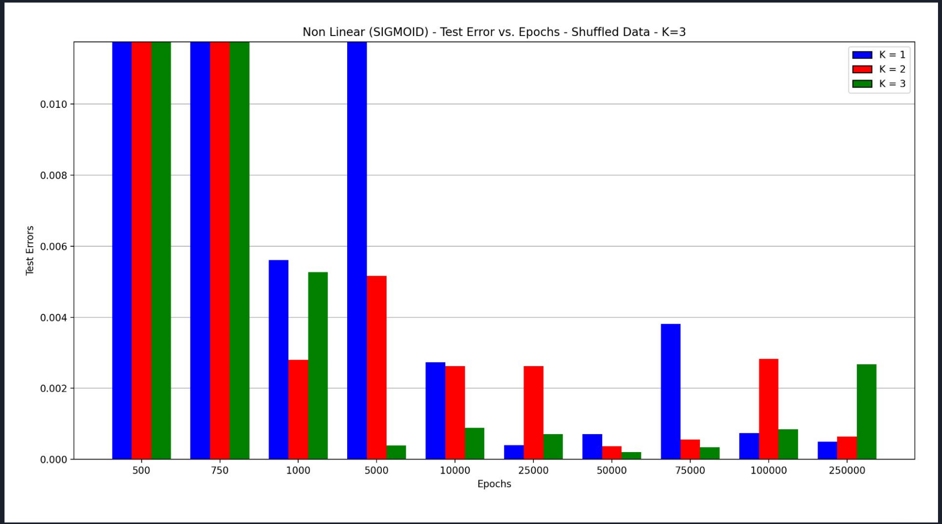
- Learning rate = 0.01
- Online
- Epochs (500, 750, 1000, 5000, 10000, 25000, 50000, 75000, 100000, 250000)
- Epsilon = 0
- Función de activación = sigmoide (datos interpolados entre 0 y 1)
- K-Fold (k entre 2 y 5)
- Beta = 1

Errores en función de las épocas

Parámetros:

- Learning rate = 0.01
- Online
- Epsilon = 0
- Función de activación = sigmoide
- K-Fold (k entre 2 y 5)
- Beta = 1

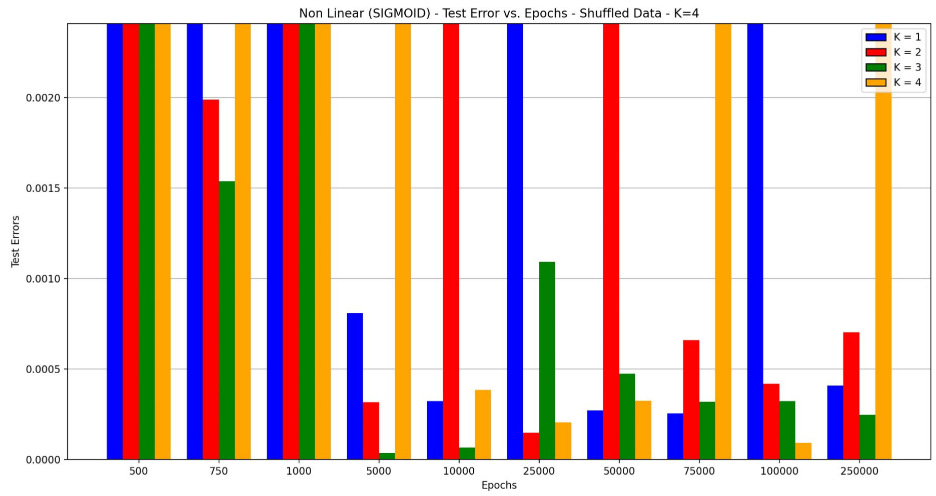


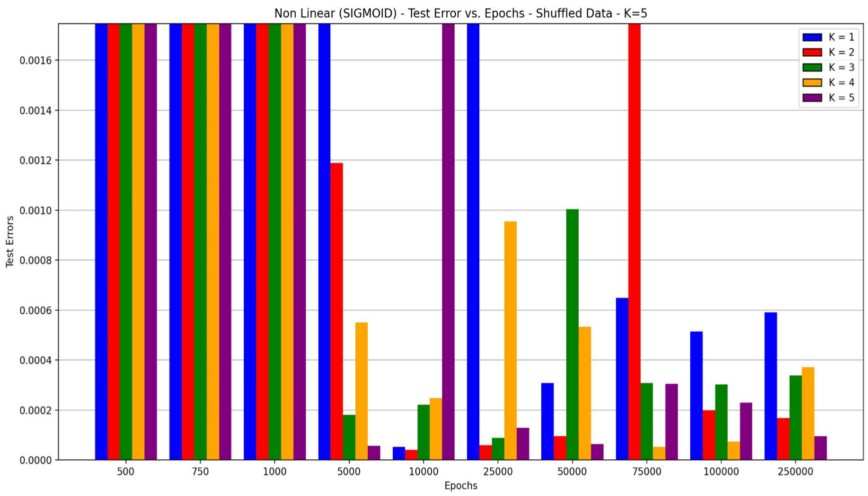


Errores en función de las épocas

Parámetros:

- Learning rate = 0.01
- Online
- Epsilon = 0
- Función de activación = sigmoide
- K-Fold (k entre 2 y 5)
- Beta = 1

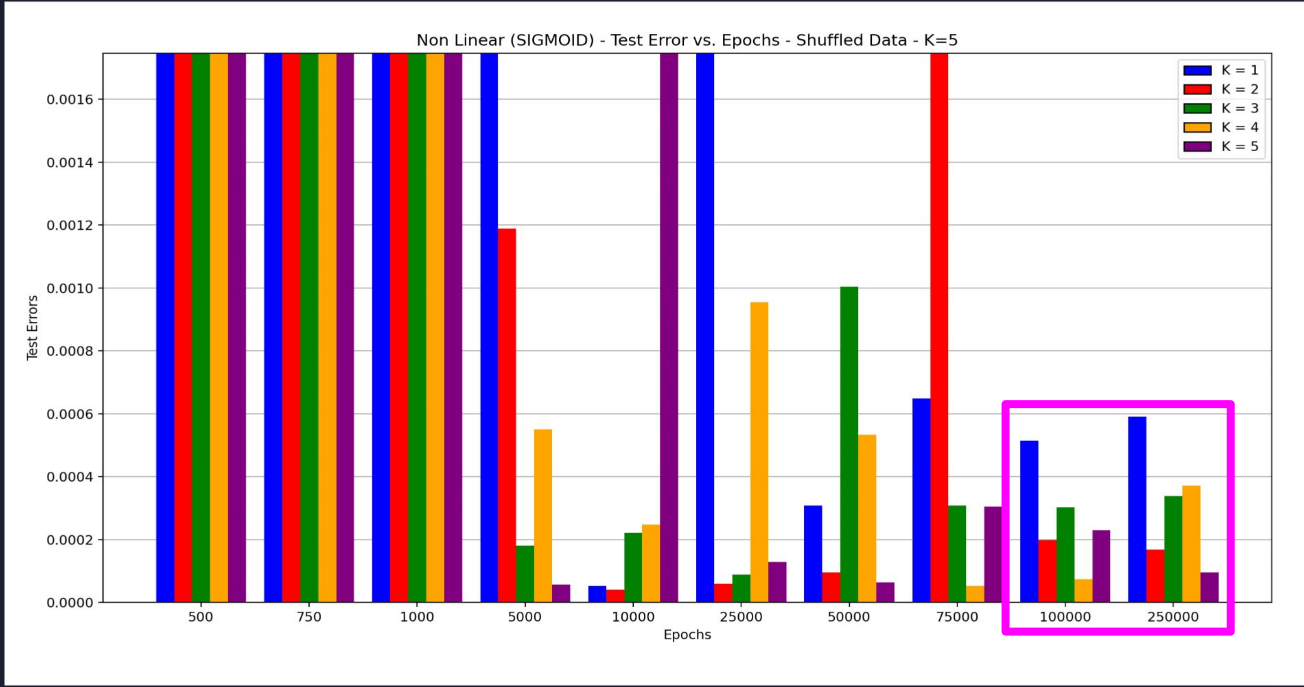




Errores en función de las épocas

Parámetros:

- Learning rate = 0.01
- Online
- Epsilon = 0
- Función de activación = sigmoide
- K-Fold (k entre 2 y 5)
- Beta = 1



Error promedio para 100000 épocas = 2.0352x10^-4

Error promedio para 250000 épocas = 1.5629x10^-3

Elección mejor beta

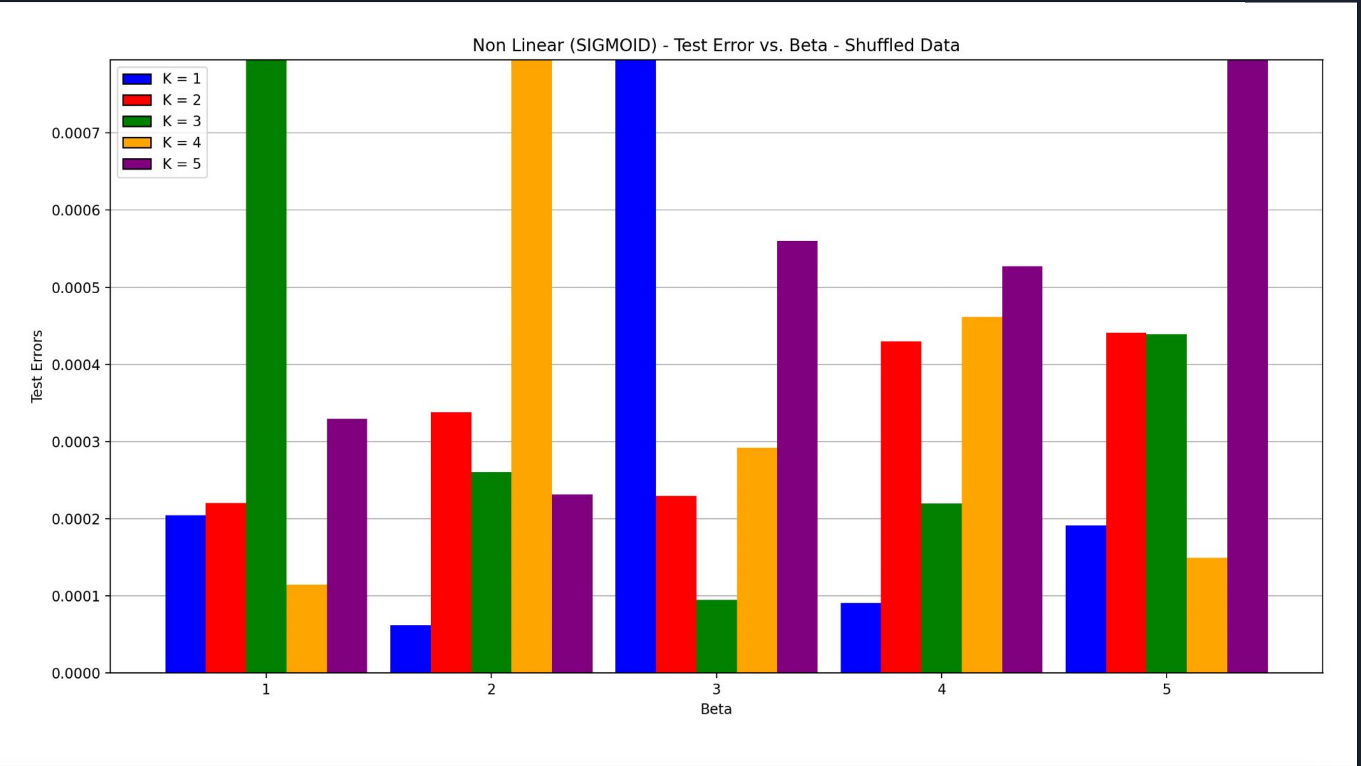
Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 100000
- Epsilon = 0
- Función de activación = sigmoide (datos interpolados entre 0 y 1)
- K-Fold: k = 5
- Beta (1, 2, 3, 4, 5)

Elección mejor beta

Parámetros:

- Learning rate = 0.01
- Online
- Epochs = 100000
- Epsilon = 0
- Función de activación = sigmoide
- K-Fold: k = 5
- Beta (1, 2, 3, 4, 5)

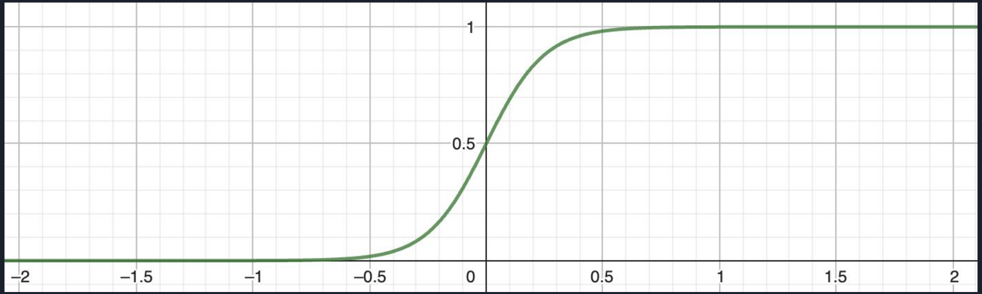


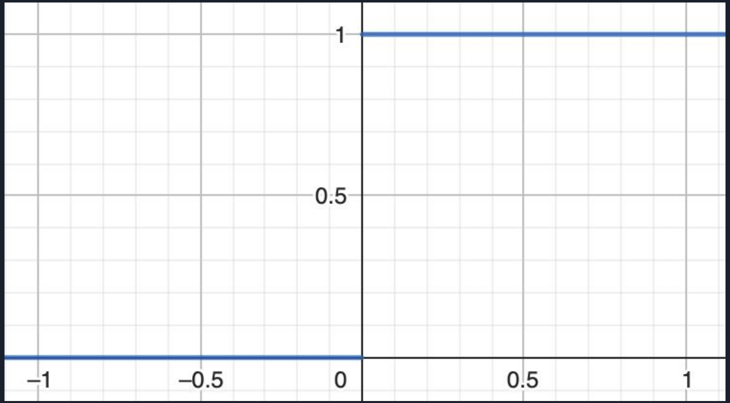
Elección mejor beta - Resultado

Beta = 4

$$\theta(h) = \frac{1}{1+exp^{-2\beta h}}$$

$$\theta'(h) = 2\beta\theta(h)(1 - \theta(h))$$





Elección mejor learning rate

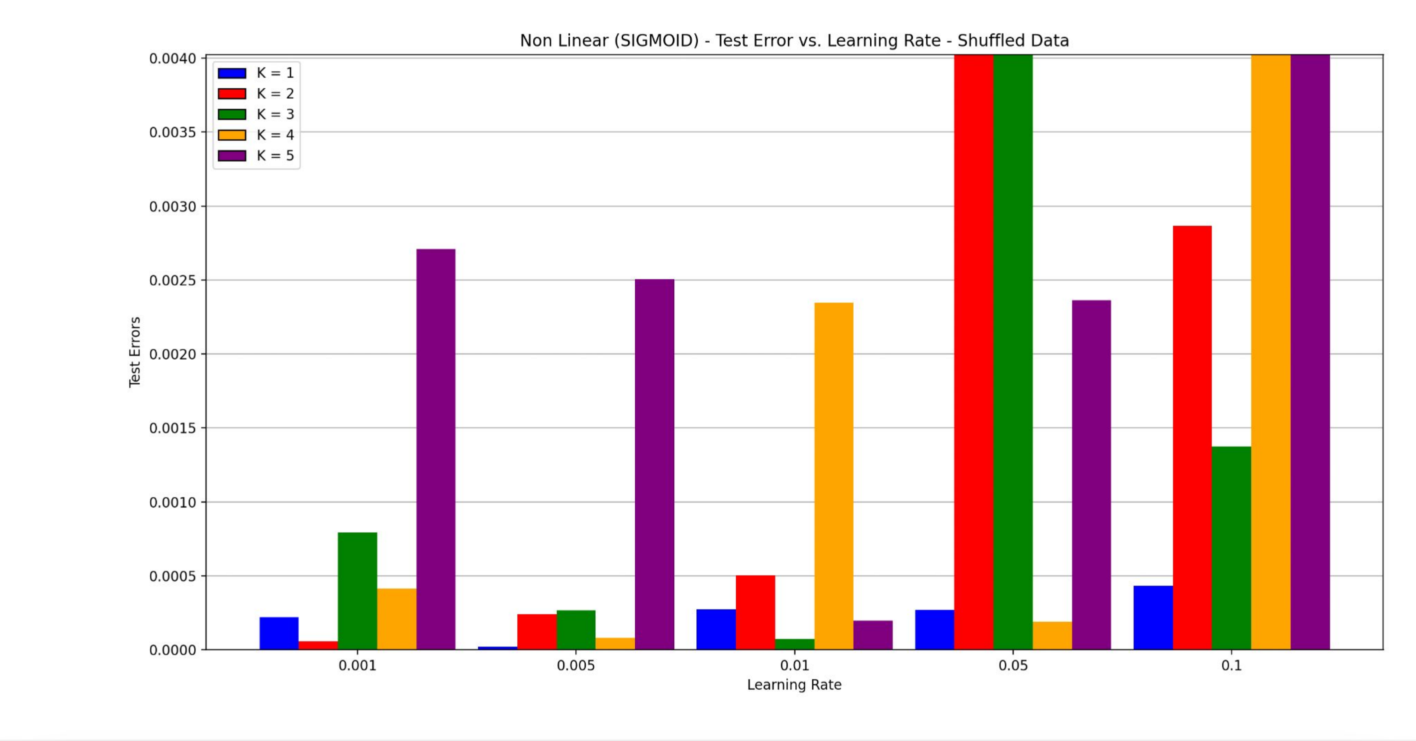
Parámetros:

- Learning rate (0.001, 0.005, 0.01, 0.05, 0.1)
- Online
- Epochs = 100000
- Epsilon = 0
- Función de activación = sigmoide (datos interpolados entre 0 y 1)
- K-Fold: k = 5
- Beta = 4

Elección mejor learning rate

Parámetros:

- Online
- Epochs = 100000
- Epsilon = 0
- Función de activación = sigmoide
- K-Fold: k = 5
- Beta = 4



Batch vs. Online

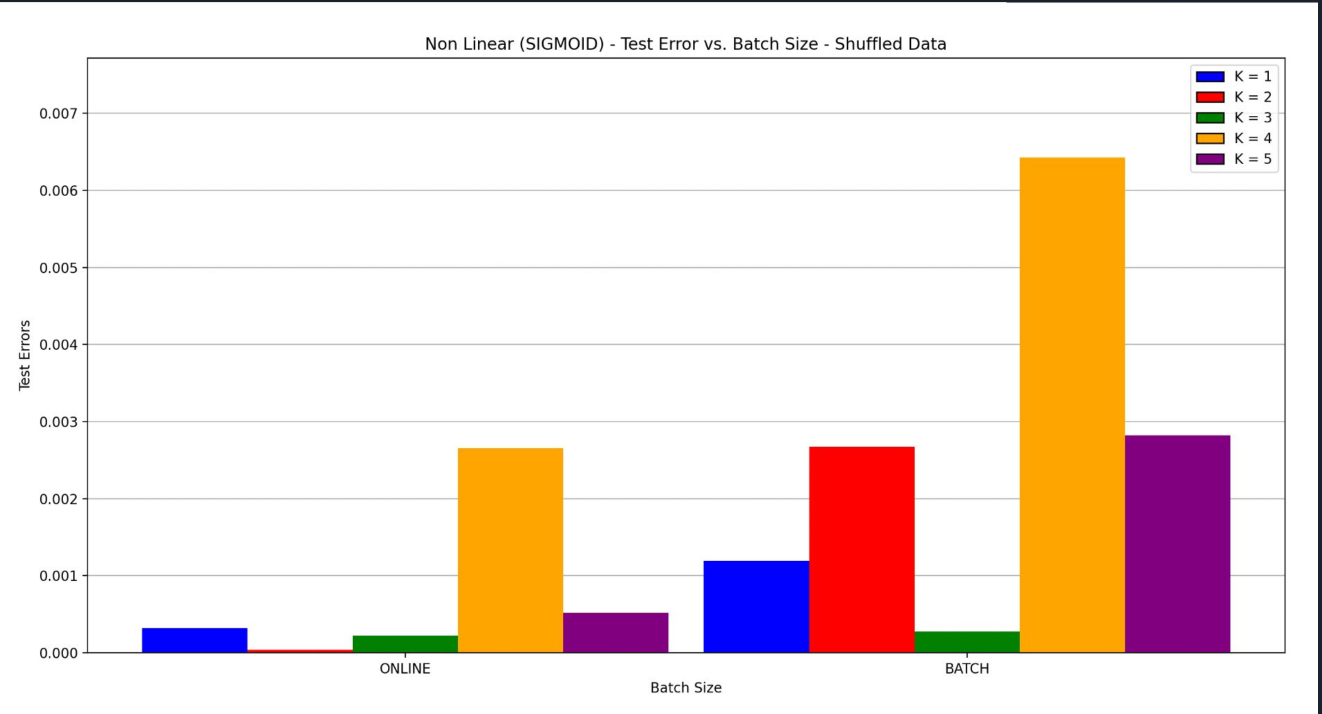
Parámetros:

- Learning rate = 0.01
- Online (batch = 1) - Batch (batch = len(training_set))
- Epochs = 100000
- Epsilon = 0
- Función de activación = sigmoide (datos interpolados entre 0 y 1)
- K-Fold: k = 5
- Beta = 4

Batch vs. Online

Parámetros:

- Learning rate = 0.01
- Online (batch = 1) - Batch (batch = len(training_set))
- Epochs = 100000
- Epsilon = 0
- Función de activación = sigmoide
- K-Fold: k = 5
- Beta = 4



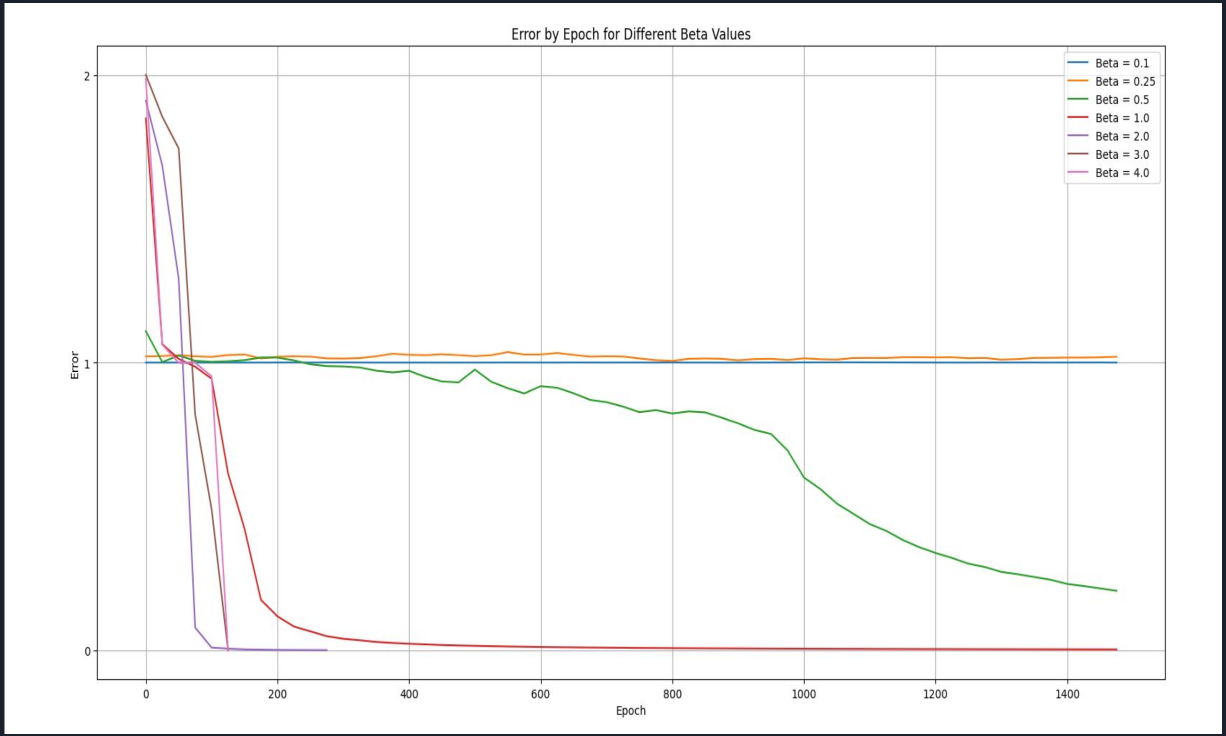
Conclusiones

- Capacidad de aprender
- Capacidad de generalización
- Mejor conjunto de entrenamiento
- Listado de los mejores parámetros para este problema:
    - Learning rate = 0.01
    - Online
    - Epochs = 100000
    - Epsilon = 0
    - Función de activación = sigmoide (datos interpolados entre 0 y 1)
    - K-Fold: k = 5
    - Beta = 4

Ejercicio 3

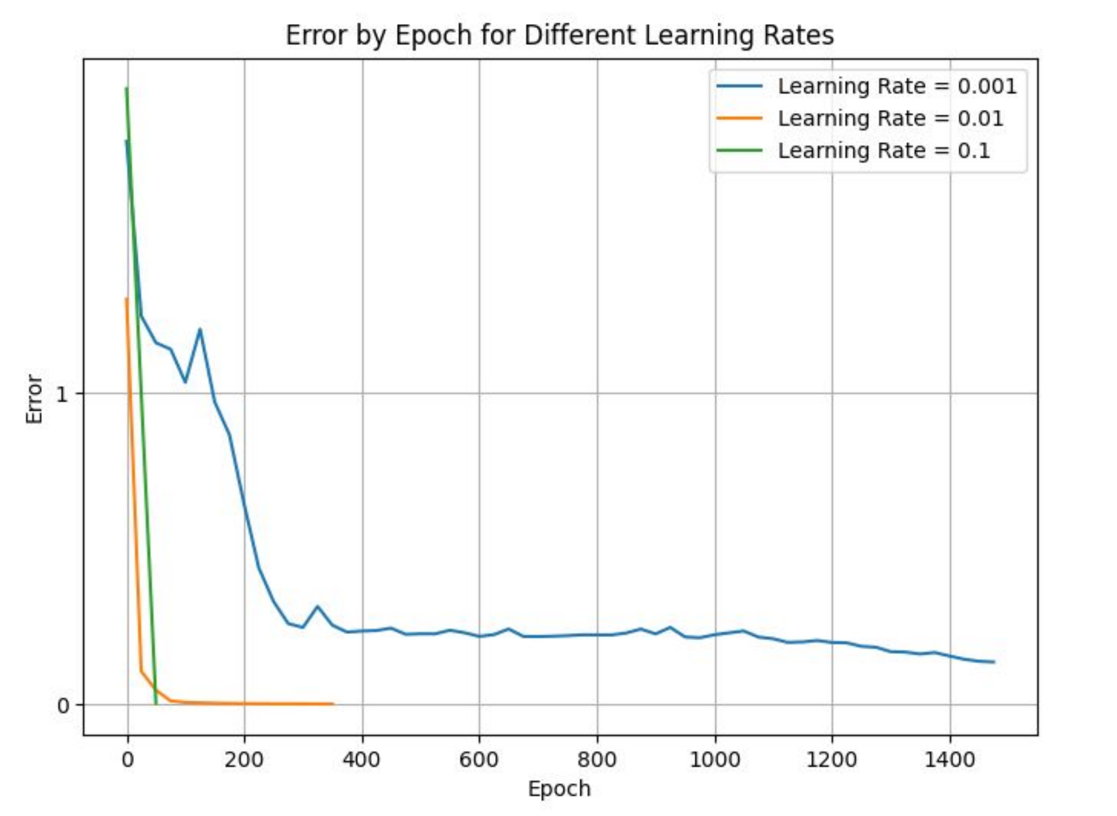
Ejercicio 3A

Elección de Beta



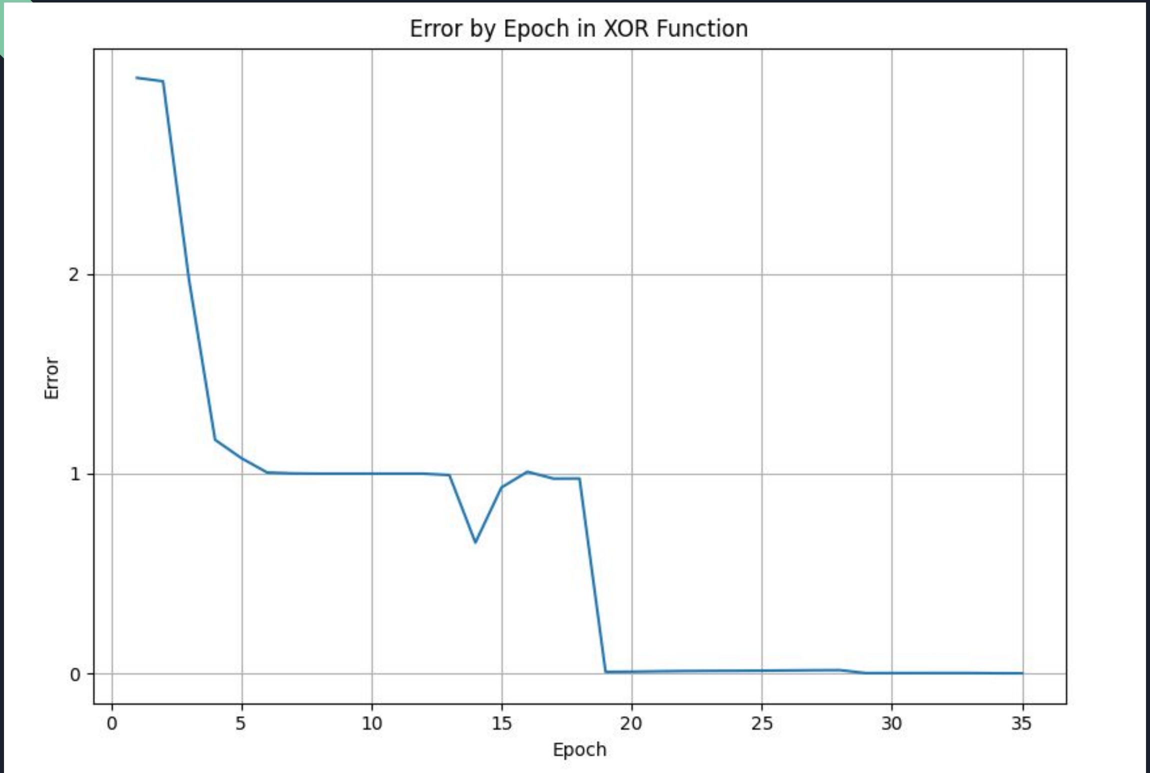
Se probó con lr = 0,1 y epsilon 0,001 ver con distintos betas como variaba el error (siempre 2-21 y tan e incremental)

Elección de Learning Rate



Se probó con beta 3 y epsilon 0,001 como variaba el error para distintos lr

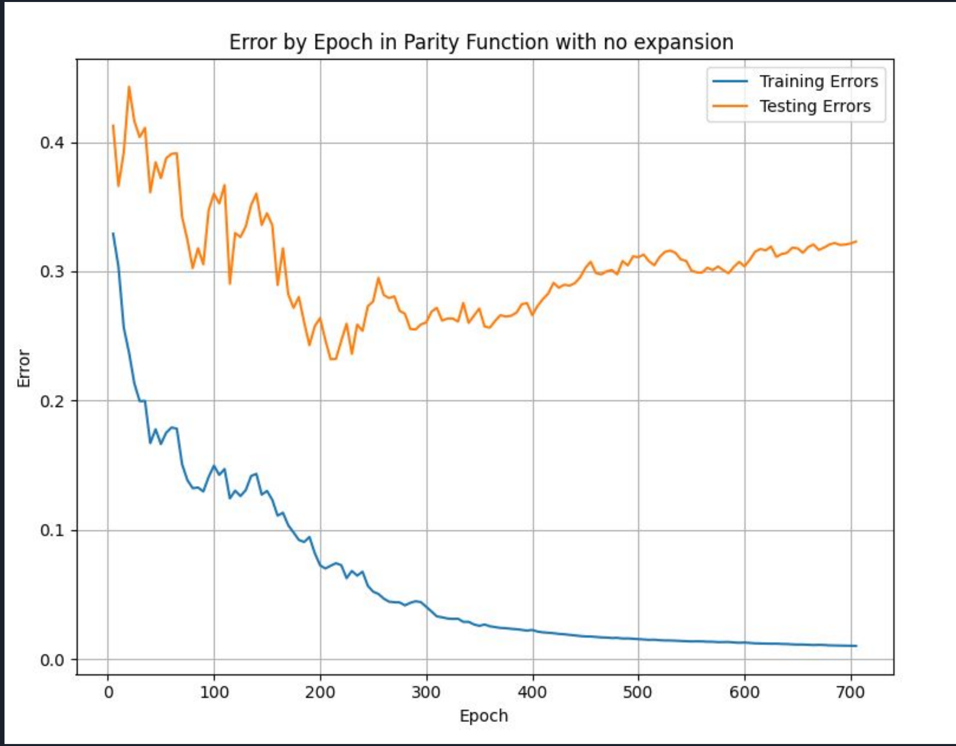
Error con las soluciones elegidas



Epochs: 35
Error: 0.00067
Input-Outputs:
[1,1] = -0.9498
[-1,1] = 0.9989
[1,-1] = 0.9987
[-1,-1] = -0.9869

EJERCICIO 3B

Errores en la función de paridad

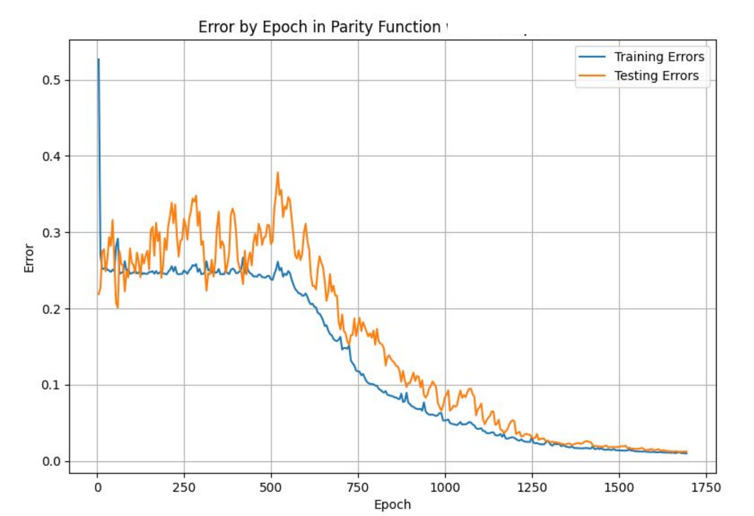


Split 70% train -30% test
LR = 0,1
Beta = 0.8
Epsilon = 0.01
Max Epochs = 10k
Método Online

Parity(6) = 0.8581
Parity(7) = 0.5
Parity(9) = 0.8413

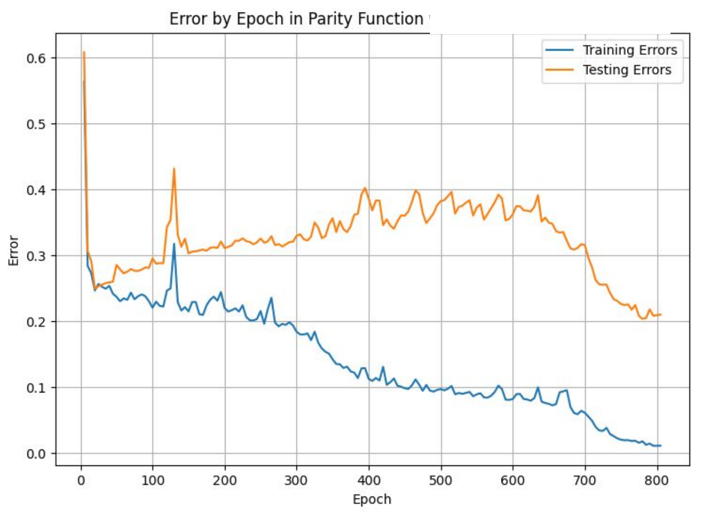
Errores en la función de paridad duplicando el training set

Sin ausencias en el testing set



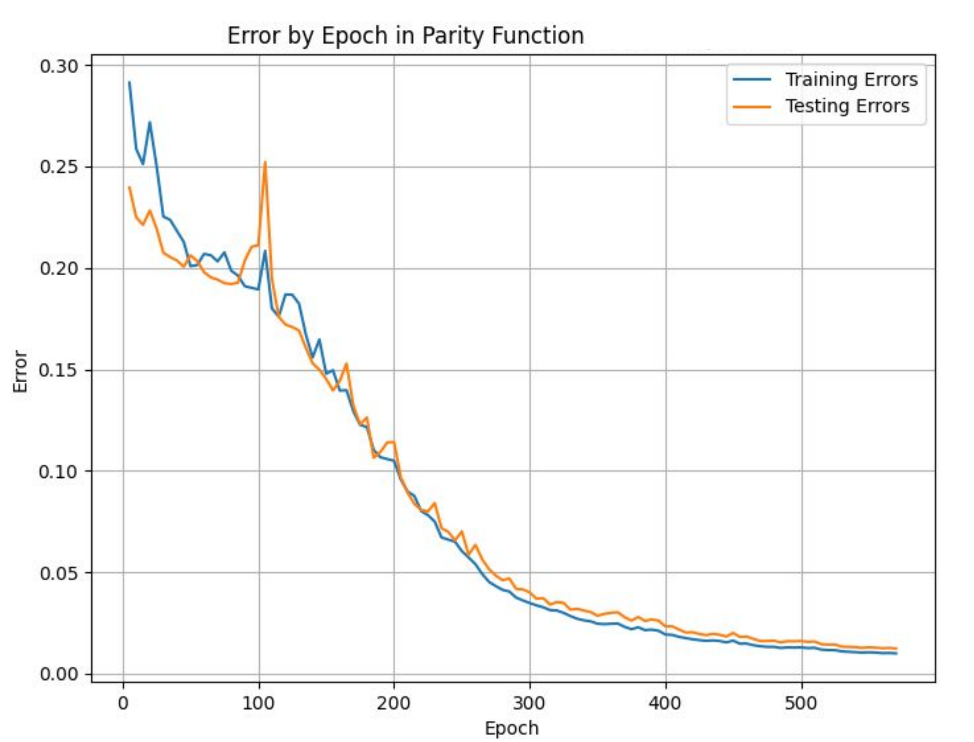
Final error: 0.0099
Final test error: 0.01246

Con ausencias en el testing set



Final error: 0.0099
Final test error: 0.2019

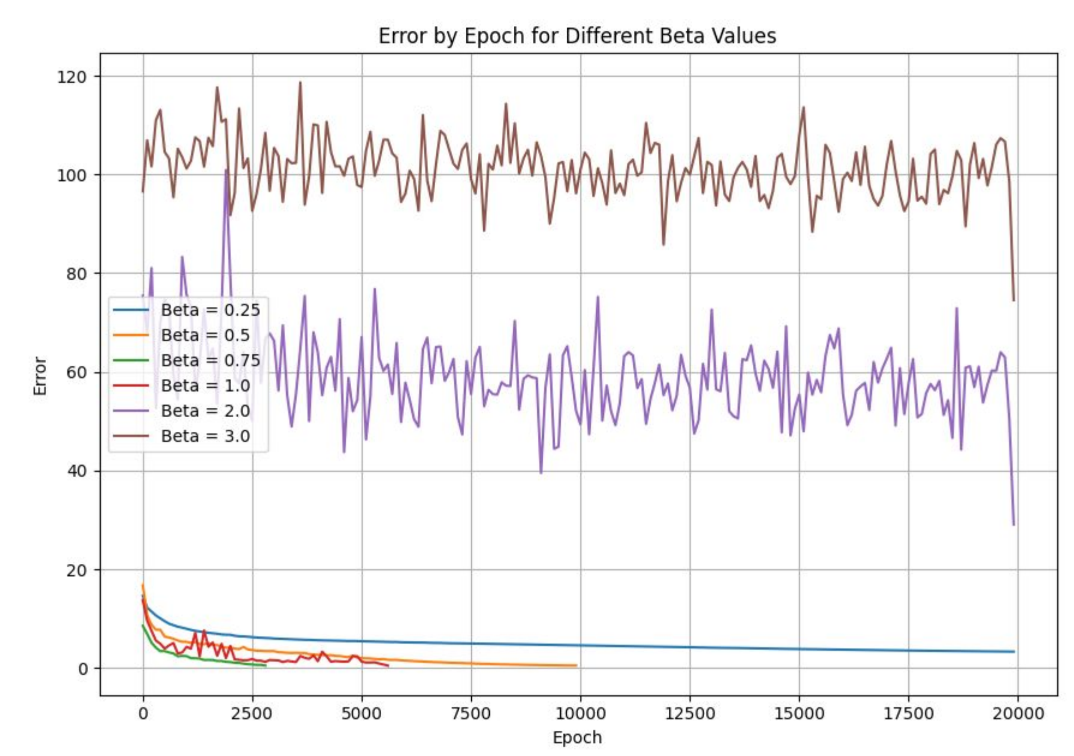
Si extendemos el dataset 10 veces más...



Split 70% train -30% test
LR = 0,1
Beta = 0.8
Epsilon = 0.01
Max Epochs = 10k
Método Online

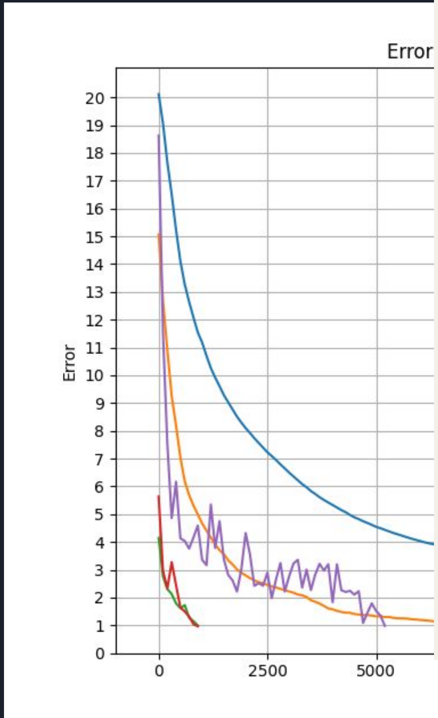
Ejercicio 3C

Betas

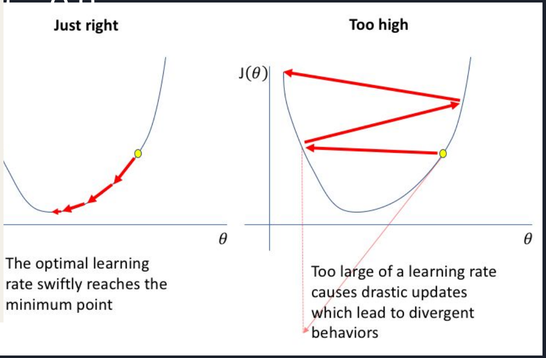


- Learning Rate = 0.1
- Online
- Epochs = 20000
- Epsilon = 0.5
- Neurons per Layer = [35, 10, 10]
- Expansion = 3
- Split = 0.8

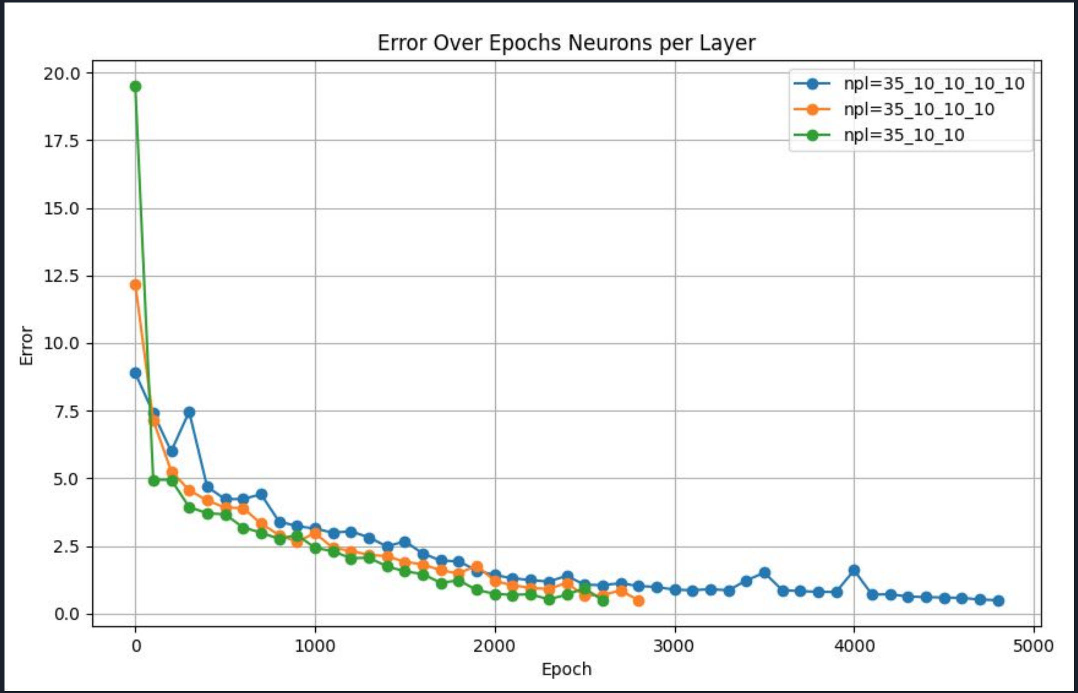
Elección mejor learning rate



- a = 0.75
- ine
- chs = 20000
- ilon = 0.5
- urons per Layer = [35, 10, 10]
- ansion = 3

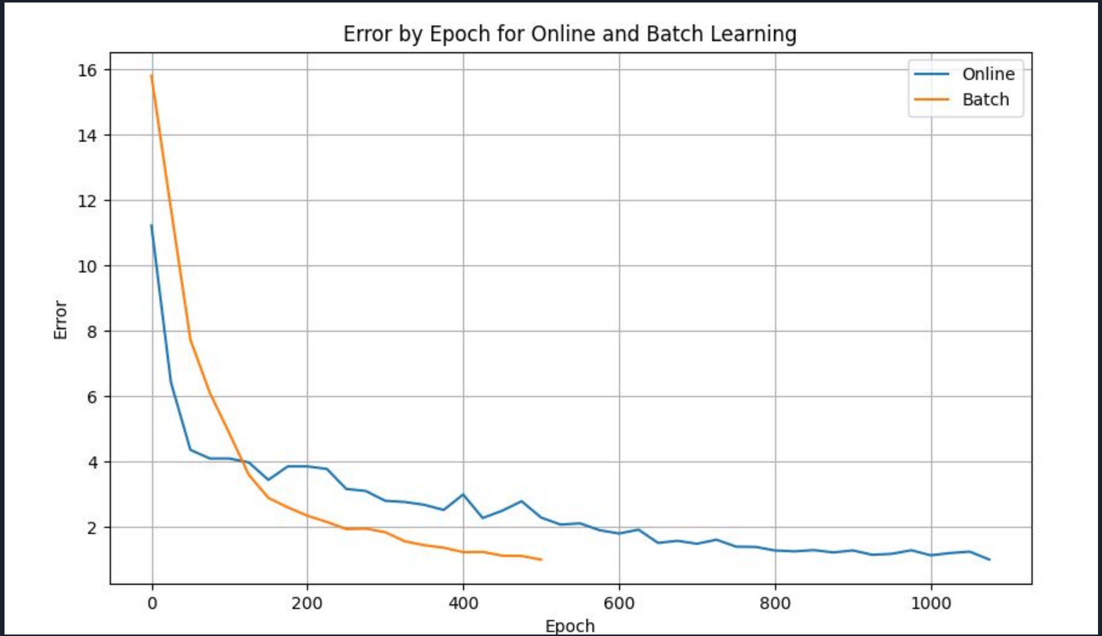


Elección estructura de la red



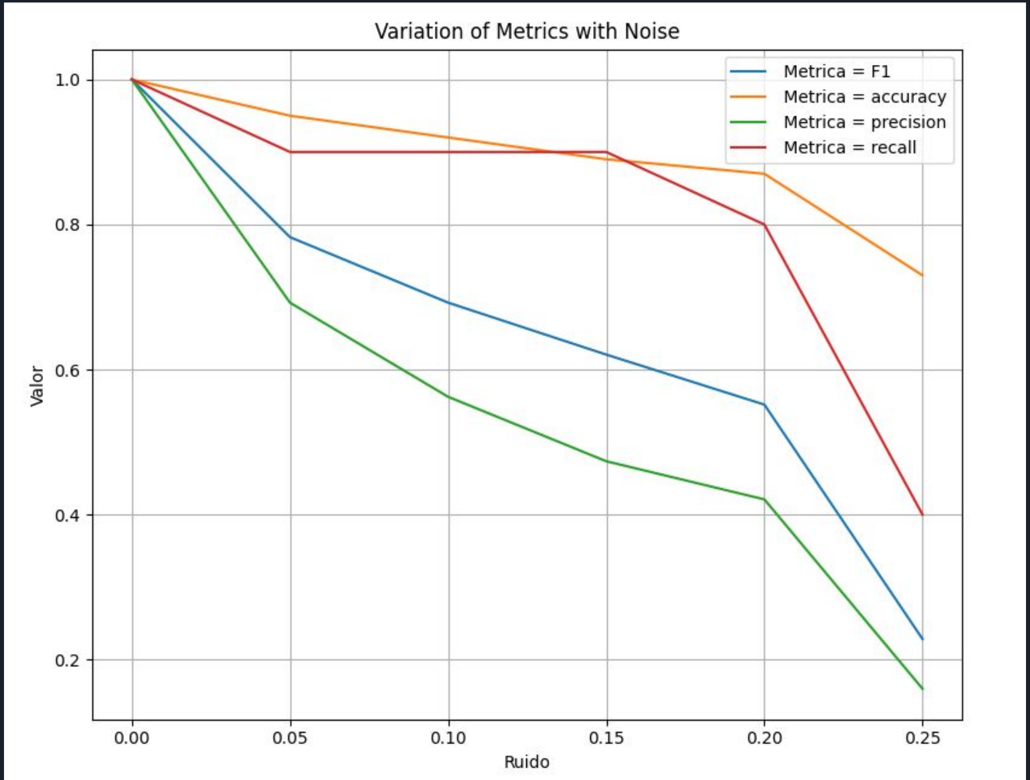
- Beta = 0.75
- Online
- Epochs = 20000
- Epsilon = 0.5
- Learning Rate = 0.1
- Expansion = 3
- Split = 0.8

Batch vs. Online



- Beta = 0.75
- Epochs = 20000
- Epsilon = 0.5
- Learning Rate = 0.1
- Expansion = 3
- Split = 0.8

Y si agregamos ruido?



- Beta = 0.75
- Batch
- Epochs = 20000
- Epsilon = 0.5
- Learning Rate = 0.1
- Expansion = 3
- Split = 0.8

Tampoco seamos malos…

```
[[0 1 0 1 0]
 [0 0 0 0 0]
 [0 0 1 0 0]
 [1 0 0 1 0]
 [1 0 0 1 0]
 [1 0 0 0 1]
 [1 1 0 1 1]]
output:
```

3

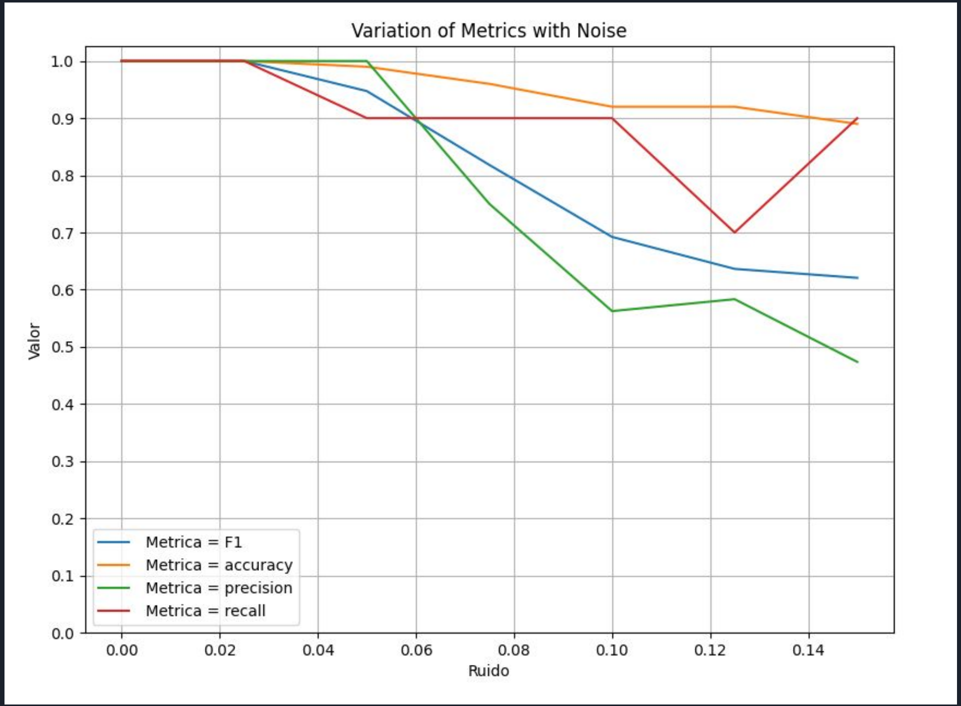
```
[[1 0 1 0 0]
 [0 1 0 0 0]
 [0 0 1 1 0]
 [0 0 0 0 0]
 [0 1 1 0 0]
 [0 0 0 0 0]
 [0 1 1 1 0]]
```

2

```
[[0 1 1 0 0]
 [1 0 0 0 1]
 [0 1 0 1 1]
 [1 0 0 0 1]
 [1 1 0 0 1]
 [0 0 0 0 1]
 [0 1 1 1 1]]
```

0

Achiquemos un poquito entonces...



- Beta = 0.75
- Batch
- Epochs = 20000
- Epsilon = 0.5
- Learning Rate = 0.1
- Expansion = 3
- Split = 0.8

Conclusiones

Ejercicio 1:

- Vemos que el perceptrón escalón es únicamente capaz de aprender funciones linealmente separables

Ejercicio 2:

- El conjunto de datos determina la funcion de activacion necesaria para resolver el problema
- El ordenamiento de los datos y la separación de los mismos son cruciales para la capacidad de generalización del perceptrón
- La cantidad límite de épocas es importante para evitar tener problemas de underfitting

Ejercicio 3:

- El perceptrón multicapa logra reconocer funciones no linealmente separables
- El learning rate debe ser adecuado para no tardar mucho más de lo necesario pero que también logre encontrar el mínimo global
- Es importante separar bien entre testing y training para que no haya underfitting ni overfitting