

# Trabajo Práctico N°3

## Perceptrón Simple y Multicapa

Grupo 7:

- ❖ Valentin Ye Li
- ❖ Nicolás Birsá
- ❖ Andres Podgorny
- ❖ Ouss Slaitane
- ❖ Hugo Lichtenberger

# Ej1 | Perceptron Simple

Activación de escalón

# Problemas

## Clasificación binaria

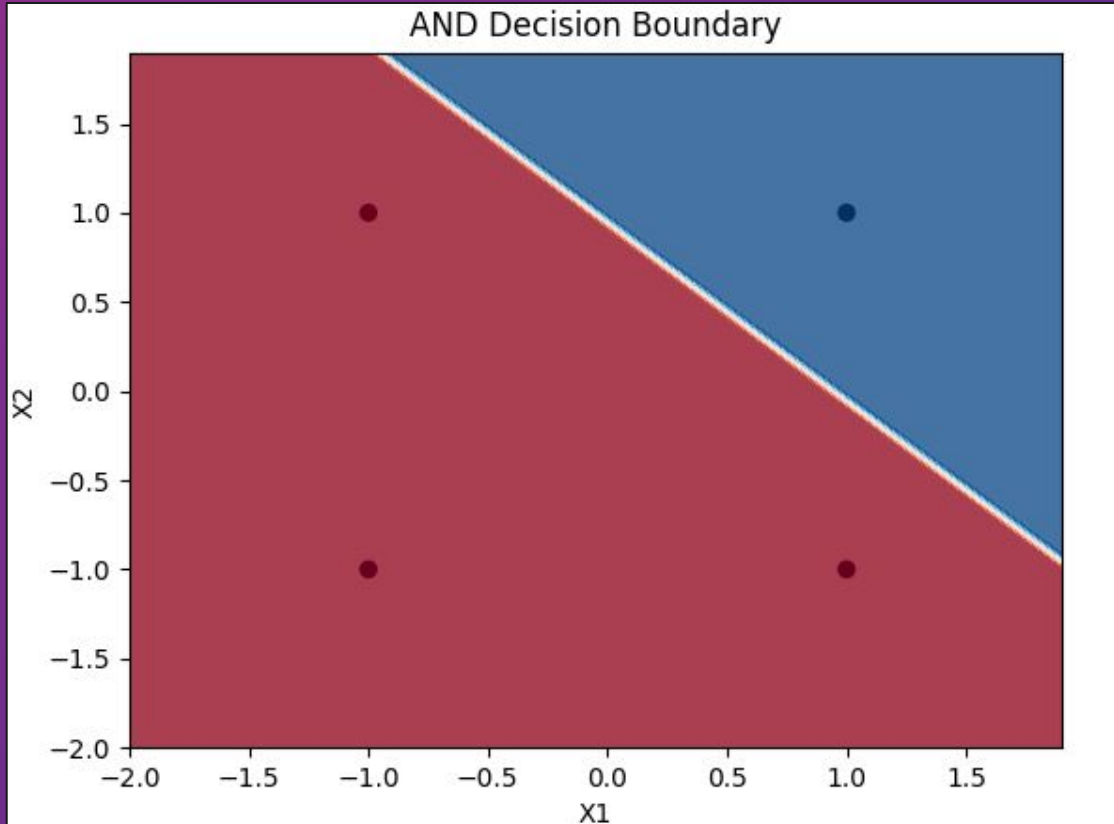
### AND

-1	1	-1
1	-1	-1
-1	-1	-1
1	1	1

### XOR

-1	1	1
1	-1	1
-1	-1	-1
1	1	-1

# AND

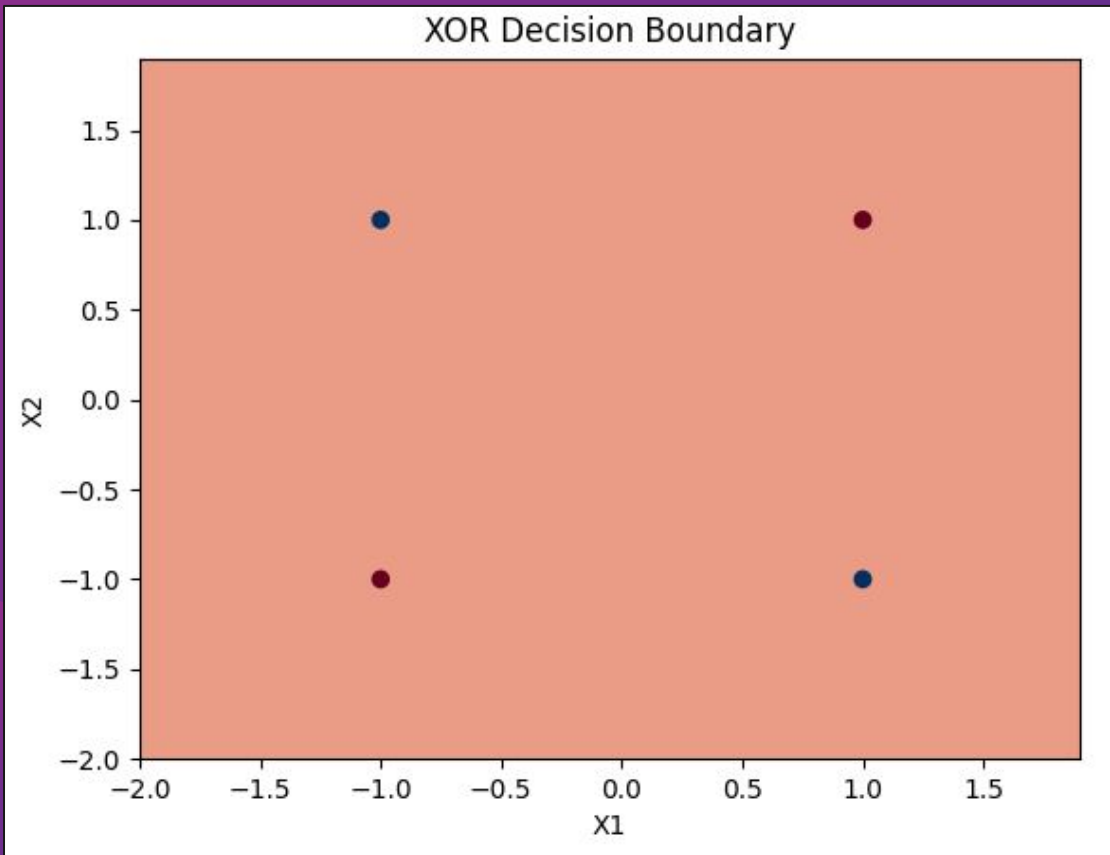


Epochs: 1000  
Learning rate: 0.1  
Accepted error: 0.05

AND Weights:  $[0.2 \ 0.2]$   
AND Bias:  $-0.2$   
AND Predictions:  $[-1 \ -1 \ -1 \ 1]$

Existe una línea blanca que clasifica correctamente a los puntos pedidos (zona roja vs zona azul).

# XOR



Epochs: 1000  
Learning rate: 0.1  
Accepted error: 0.05

XOR Weights: [0.0.]  
XOR Bias: -0.4  
XOR Predictions: [-1 -1 -1 -1]

No existe una línea que pueda clasificar correctamente a los puntos pedidos.



# Conclusión

No es posible utilizar el perceptrón simple con activación de escalón para resolver problemas no linealmente separables, como es el caso del operador XOR.



# Ej2 | Perceptrón Simple


Lineal y No Lineal



# Selección de conjuntos de entrenamiento y testeo

- ❖ Lineal
- ❖ No lineal
  - Tanh
  - Logística

10 ejecuciones  
Epochs: 3000  
Learning rate: 0.00001  
Accepted error: 0.001  
Beta: 1







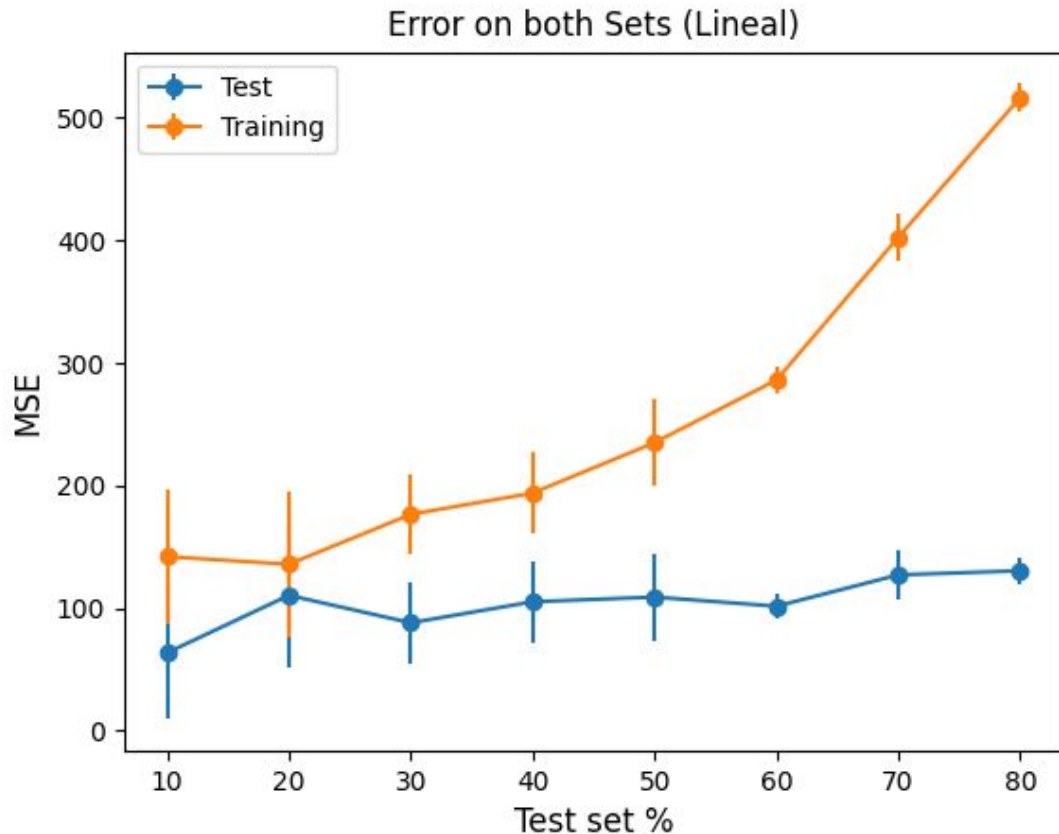
# A tener en cuenta

- ❖ Los MSE en los perceptrones no lineales están acotados por las funciones tanh y logística que tienen imagen  $[-1, 1]$  y  $[0, 1]$  respectivamente. Estas salidas al estar elevadas al cuadrado producen errores muy pequeños.

# Lineal

Se observa que el MSE es mínimo para ambos conjuntos cuando estos se divide en:

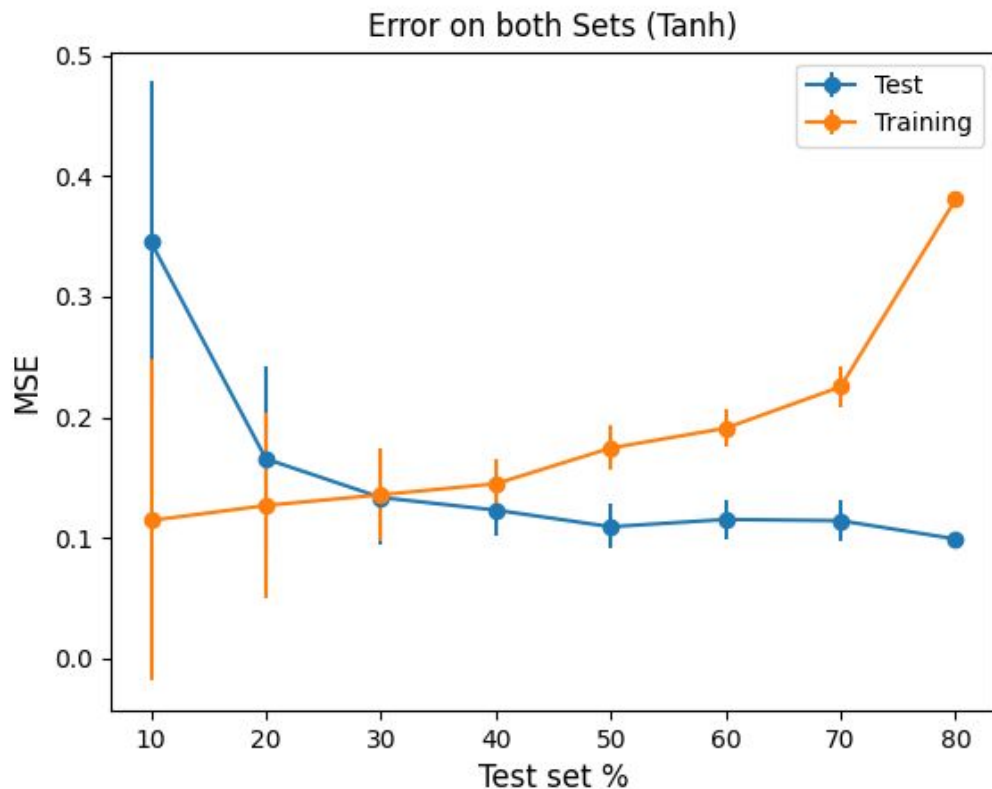
20% Test  
80% Training



# No lineal - Tangente hiperbólica

Se observa que el MSE es mínimo para ambos conjuntos cuando estos se divide en:

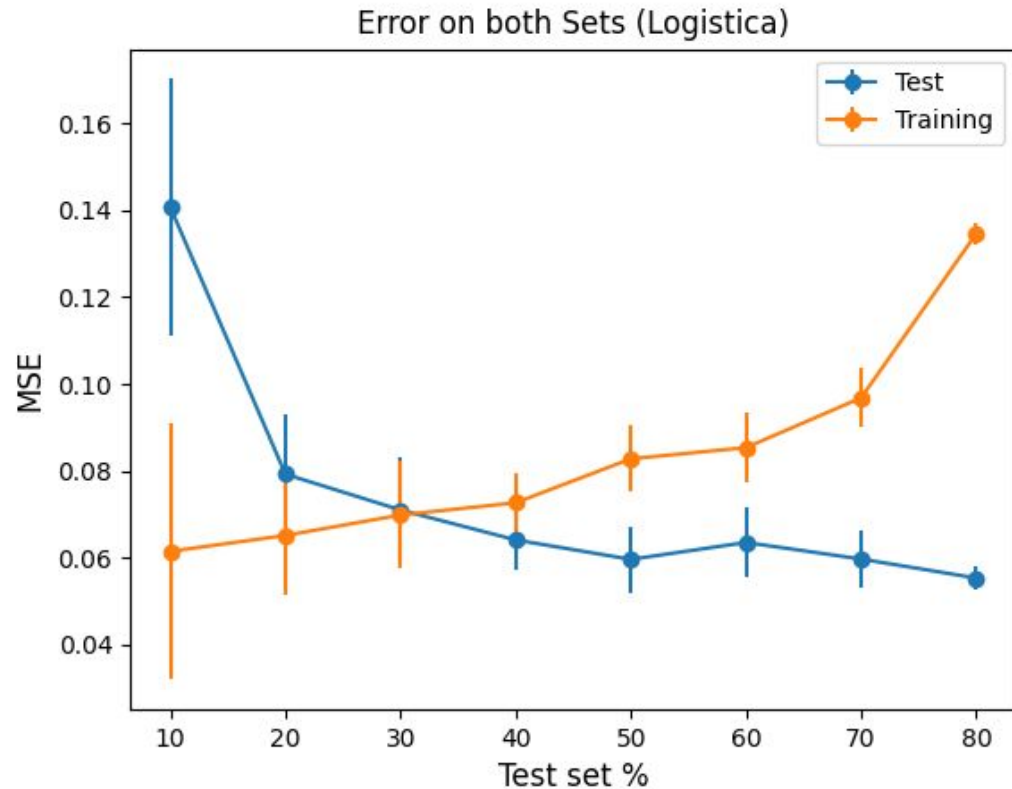
30% Test  
70% Training



# No lineal - Logística

Se observa que el MSE es mínimo para ambos conjuntos cuando estos se divide en:

30% Test  
70% Training

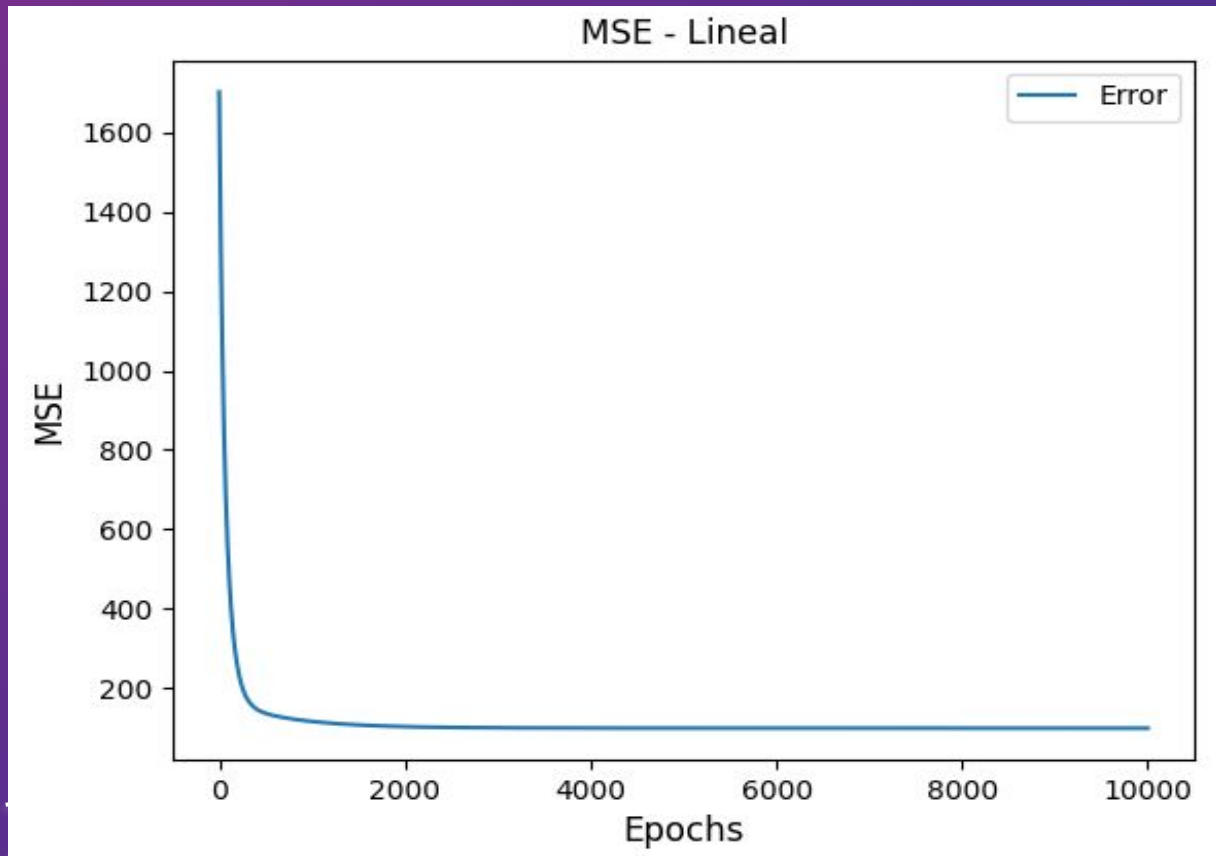


# Conclusiones

- ❖ Capacidad de generalización: Con un porcentaje de testeo del 30% (En caso del lineal es 20%) se puede observar que el MSE es mínimo en ambos conjuntos, esto puede producir que los modelos elegidos tengan mayor generalización
- ❖ Los extremos (10% de testeo y 90% de testeo) producen mínimos en uno de los conjuntos pero mayores en el otro. Esto se puede traducir en una mala generalización.
- ❖ Con una gran capacidad de generalización se puede lograr que el modelo sea menos susceptible al ruido, siempre y cuando se utilicen valores adecuados de training rate, beta y épocas.

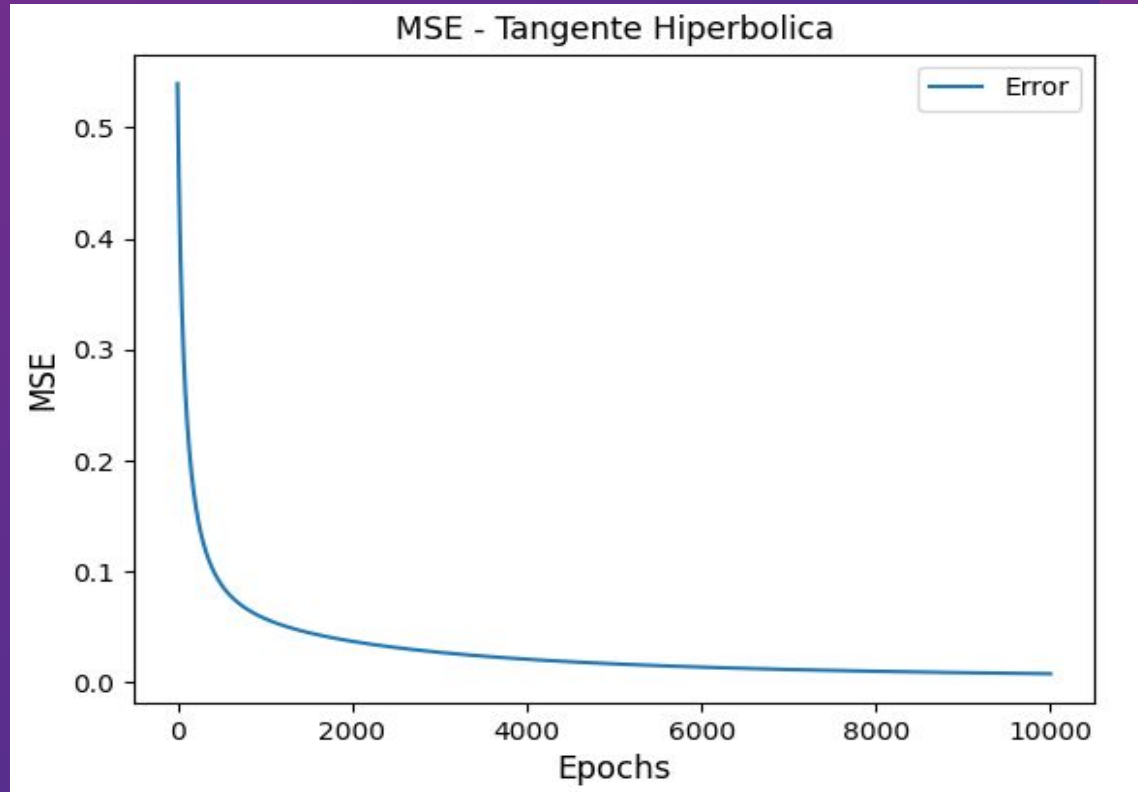
# Error cuadrático medio - Lineal

Test set: 20%  
Epochs: 10000  
Learning rate:  
0.0001  
Accepted error:  
0.001



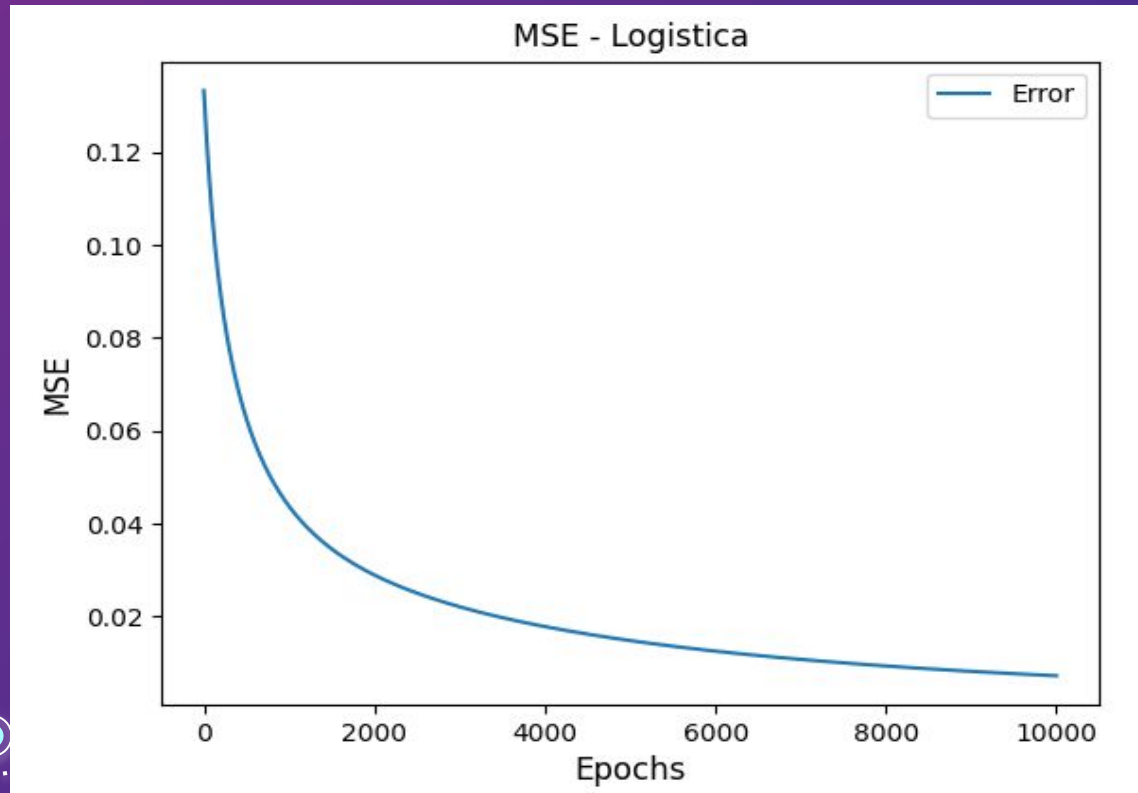
# Error cuadrático medio - No lineal Tanh

Test set: 30%  
Epochs: 10000  
Learning rate:  
0.0001  
Accepted error:  
0.001  
Beta: 1



# Error cuadrático medio - No lineal Logística

Test set: 30%  
Epochs: 10000  
Learning rate:  
0.0001  
Accepted error:  
0.001  
Beta: 1



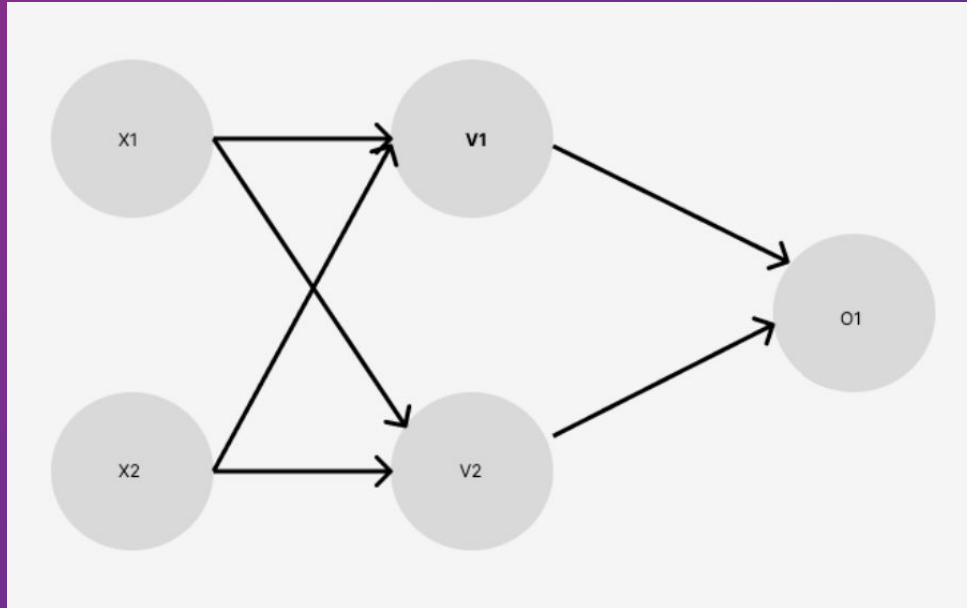


# Conclusiones

- La función de activación lineal tiene una convergencia más rápida que el resto de modelos. Pero esto no necesariamente quiere decir que se obtenga el mejor resultado.
- Entre las funciones no lineales, la logística presenta mejores resultados, ya que para que la tanh pueda obtener un valor de error mínimo cercano al de la logística requiere una cantidad de épocas muy superior.

# Ej3 | Perceptrón Multicapa

# Arquitectura

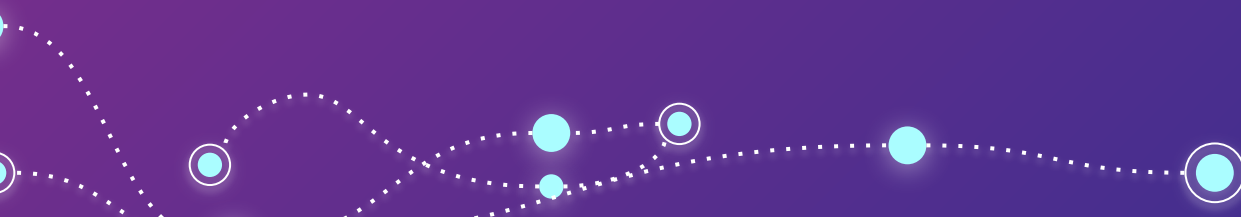


# XOR

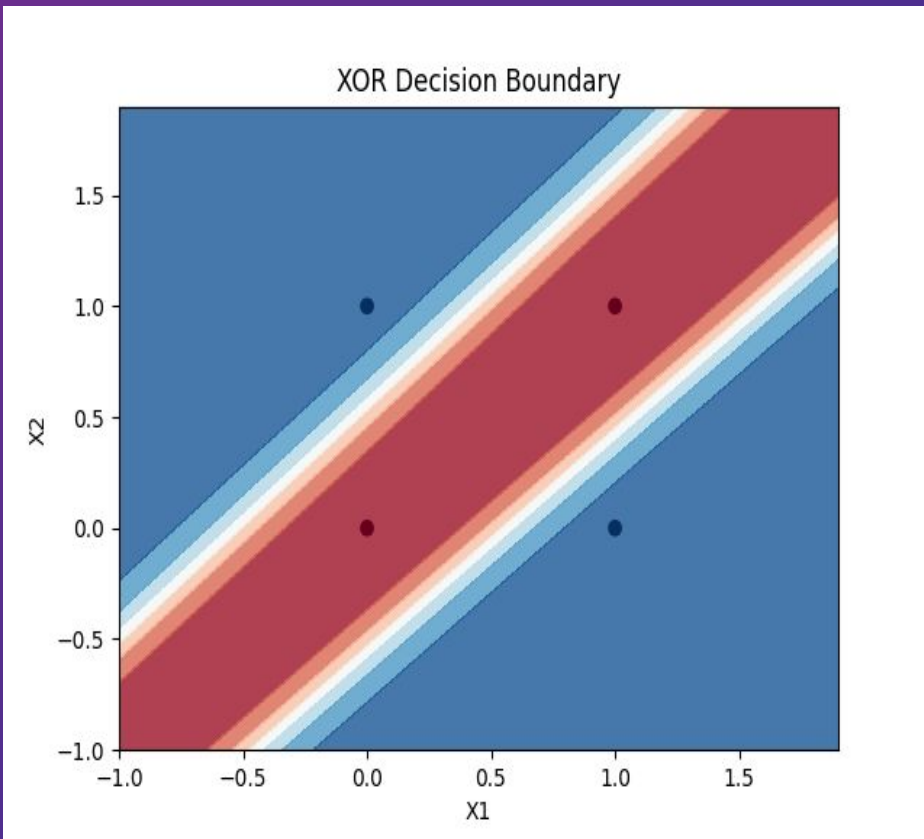
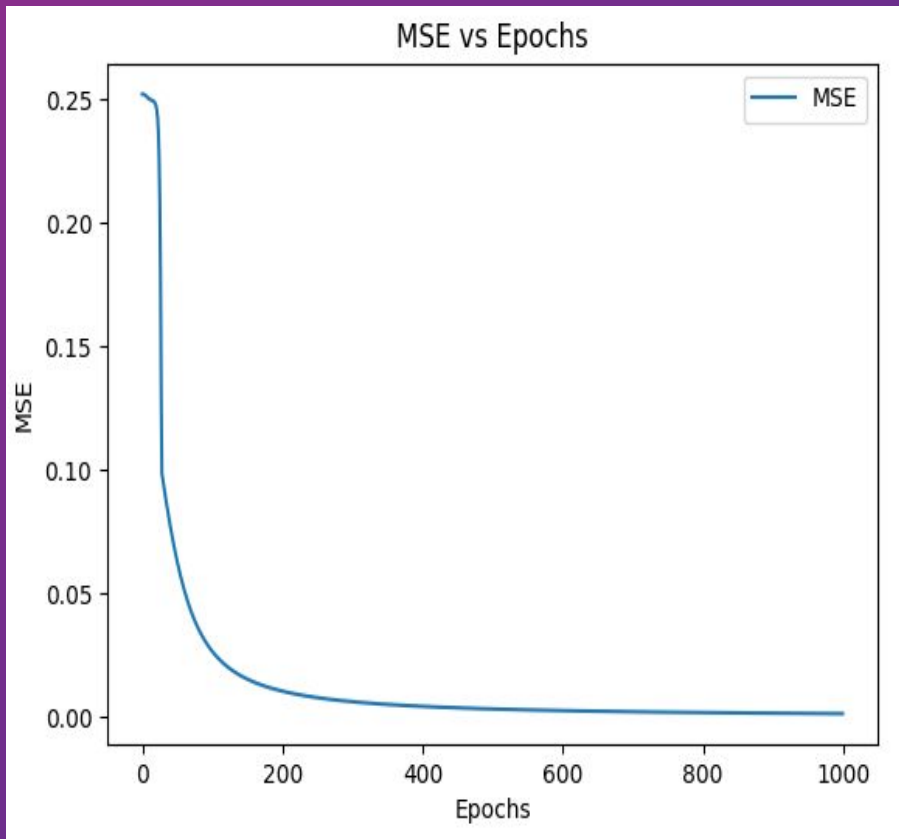
No es linealmente separable

Epochs: 1000  
Learning rate: 0.001  
Momentum: 0.9  
Inputs: 2  
Hidden Layers: 2  
Outputs: 1

$\zeta_1$	$\zeta_2$	$\zeta$ Esperado	O Obtenido
0	1	1	0.90557614
1	0	1	0.90551302
0	0	0	0.07672676
1	1	0	0.12715666



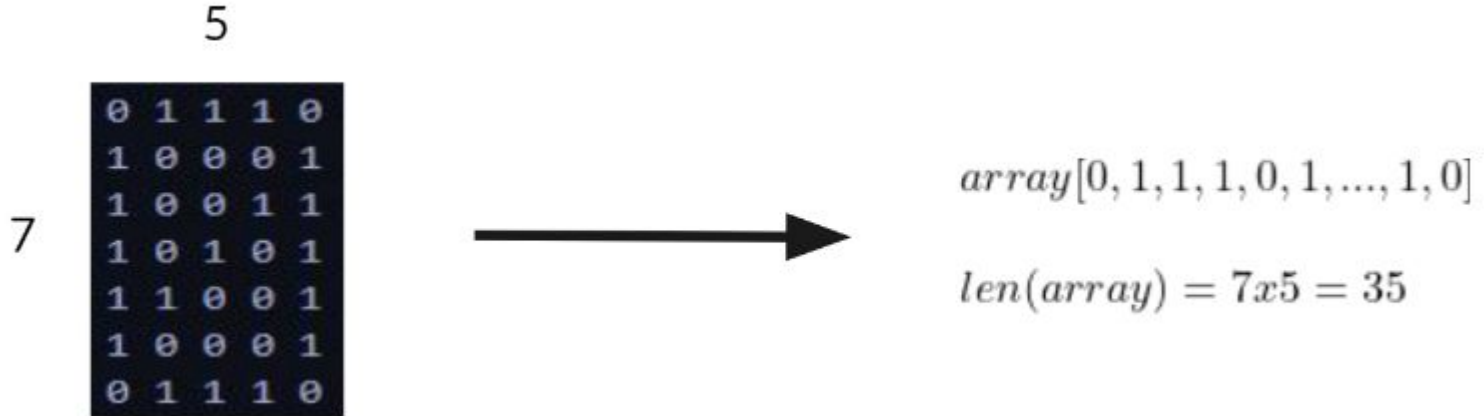
# XOR



# Ejercicios 3B y 3C

El perceptrón multicapa debe interpretar una matriz binaria como un dígito en ascii art, “encendiendo” una de las 10 salidas correspondientes.

Internamente se toma la matriz como un arreglo de 35 valores.



# Criterios de corte y Optimizaciones

Para terminar el algoritmo se debe o alcanzar el número máximo de épocas, o que las predicciones tengan un error MSE menor a cierto valor, de ahora en más **Convergence Threshold**.

También se implementó la optimización **Momentum**, aunque en los modelos estudiados no disminuyó los errores obtenidos, por lo que no se analizará en detalle.

# Evaluación de parámetros

Vamos a evaluar cómo afectan los siguientes parámetros en el error medio obtenido para los modelos 3B y 3C.

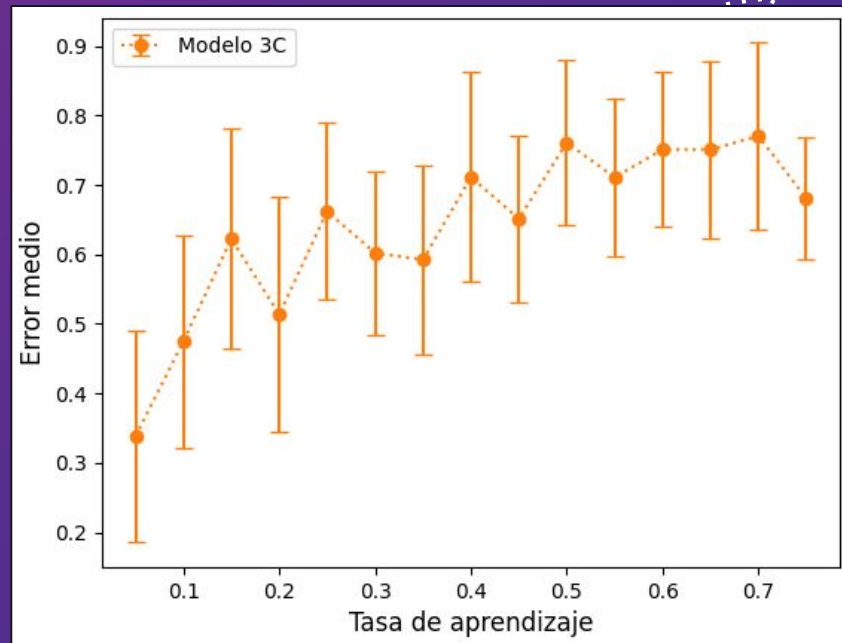
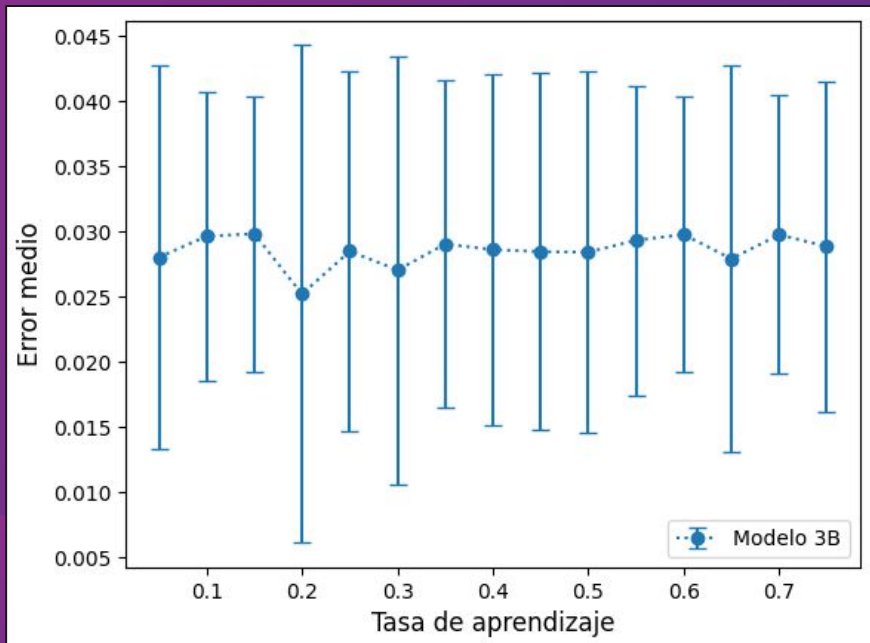
- Tasa de aprendizaje.
- Arquitectura. Cantidad de perceptrones por capa, cantidad de capas.
- Capacidad de generalización, separando en conjuntos de train y test.
- Susceptibilidad al ruido (solo para 3C).

A su vez el error será el módulo de la diferencia entre el valor obtenido y el deseado para el modelo 3B, mientras que para el 3C será la diferencia entre el valor en la posición deseado y el “segundo máximo” obtenido.

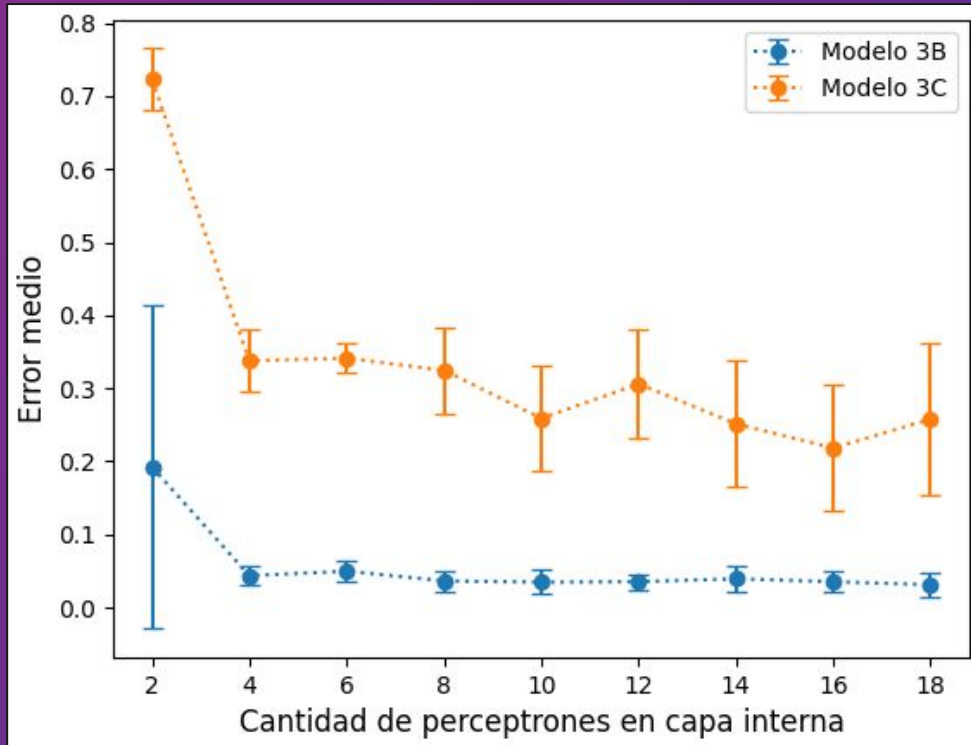
**Ejemplo 3C:** Si para un 3, la posición del 3 tiene un 0.8 y la “segunda máxima” tiene un 0.3, el error será de  $1-(0.8-0.3)=0.5$



# Tasa de aprendizaje



# Una capa



Epochs: 10000

Learning rate: 0.01

Momentum: 0.9

Beta: 0.1

Convergence threshold: 0.05

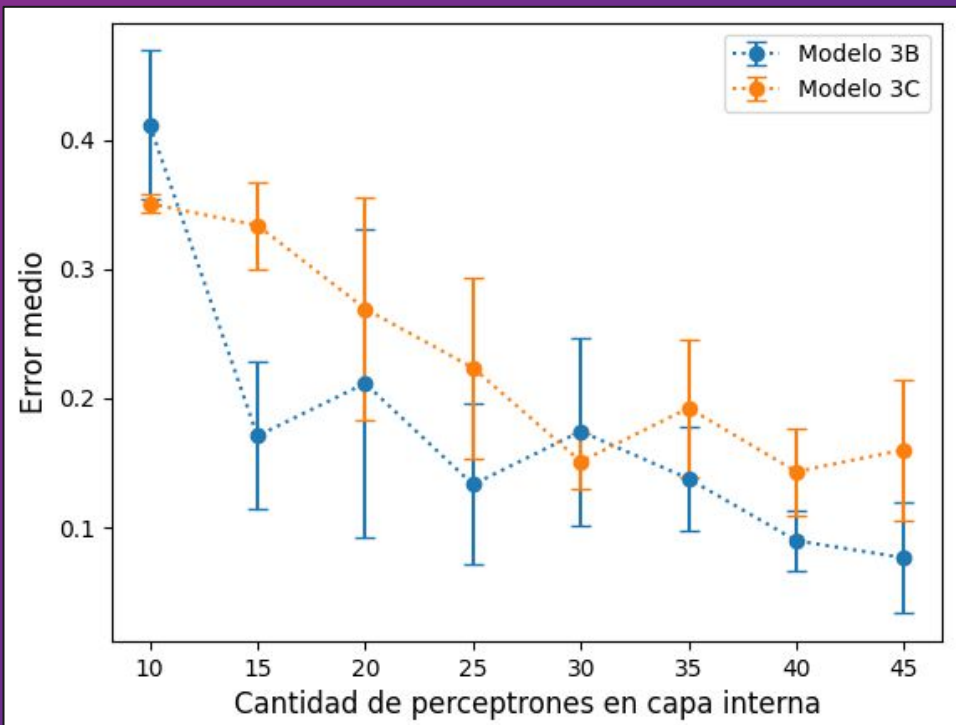
10 ejecuciones por valor

Al aumentar la cantidad de capas el error medio disminuye, pero a partir de cierto punto dicha disminución deja de ser notoria.

En el 3B a partir de 4 perceptrones no tiene sentido aumentar la cantidad por capa.

# Varias capas internas

2 capas internas con misma cantidad de perceptrones



Epochs: 10000

Learning rate: 0.01

Momentum: 0.9

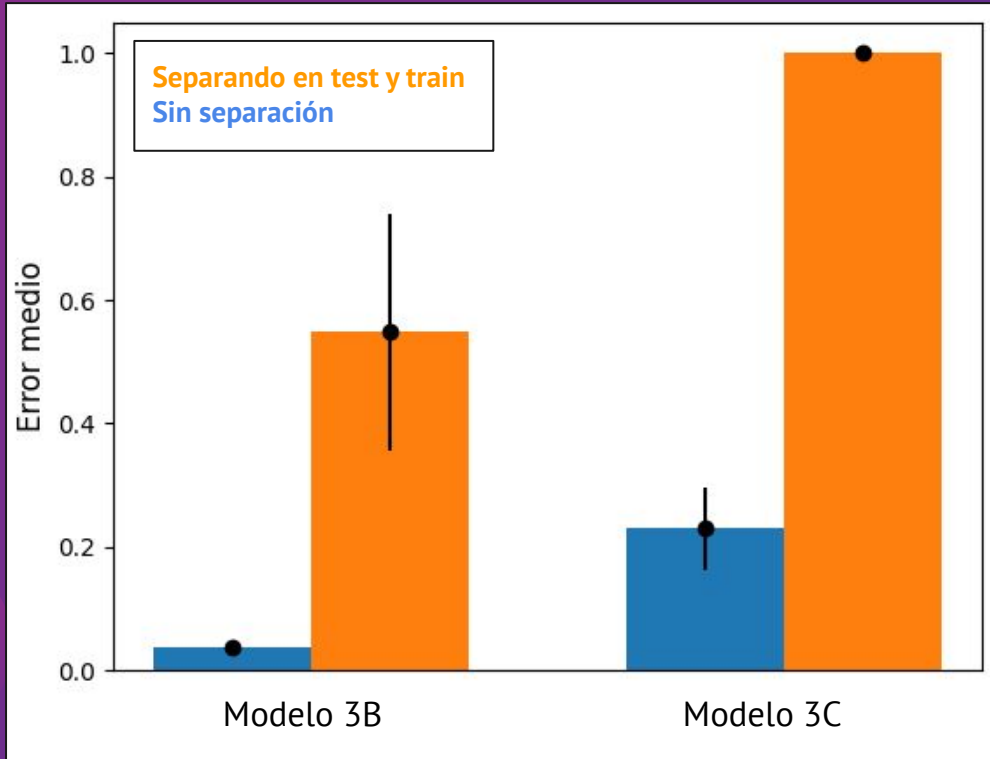
Beta: 0.1

Convergence threshold: 0.05

10 ejecuciones por valor

Al aumentar a 2 la cantidad de capas, el error medio disminuye a medida que aumentan los perceptrones por capa, pero sin “estancarse” a diferencia de una capa.

# Capacidad de Generalización



Epochs: 10000

Learning rate: 0.01

Momentum: 0.9

Beta: 0.1

Convergence threshold: 0.05

Hidden layer: 1, con 20 perceptrons

10 ejecuciones por valor

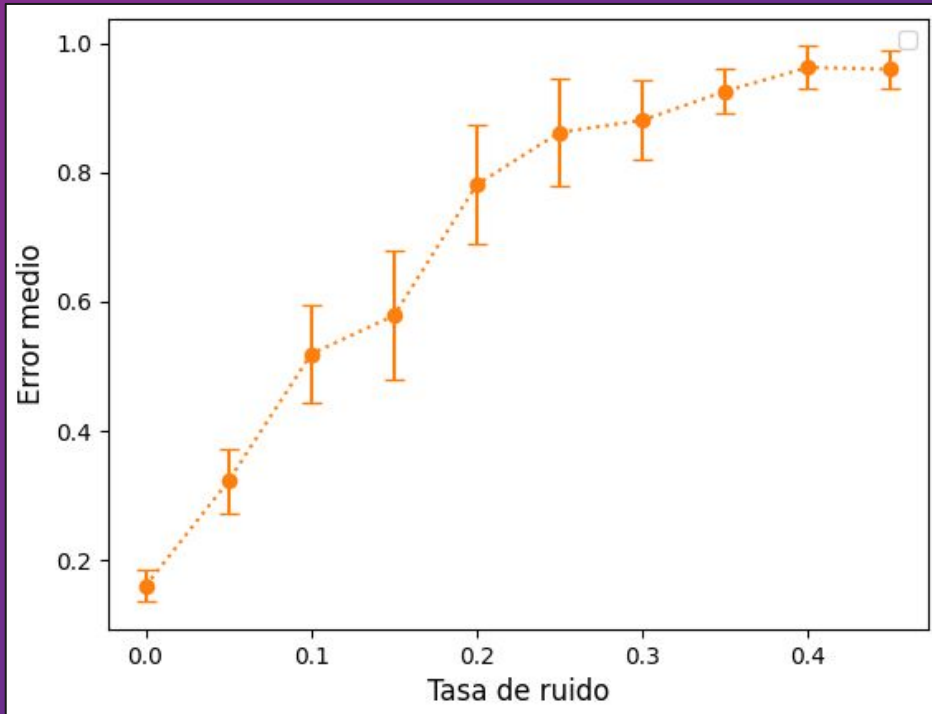
70% Train - 30% Test

Ambos modelos tienen poca capacidad de generalización.

El 3C no puede predecir correctamente ningún valor no conocido, y el 3B tiene un error menor, pero igualmente considerable ya que sus posibles valores son  $[0, 1]$

# Susceptibilidad al ruido

## Modelo 3C



Epochs: 10000

Learning rate: 0.01

Momentum: 0.9

Beta: 0.1

Convergence threshold: 0.05

Hidden layer: 1 with 20  
perceptrons

10 ejecuciones por valor

# Conclusiones

- Disminuir la tasa de aprendizaje a valores comunes (0.1) no necesariamente mejora el error medio
- Mayor cantidad de perceptrones y de capas internas minimiza el error medio
- Un modelo con baja capacidad de generalización es muy susceptible al ruido

**¡Gracias!**

