

# Perceptrón Simple y Multicapa

Sistemas de Inteligencia Artificial  
ITBA 2022 - 1C

## Integrantes:

- Serpe, Octavio (60076)
- Quesada, Francisco (60524)
- Arca, Gonzalo (60303)



# Introducción

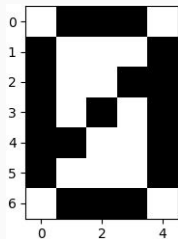
## Problemas



A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

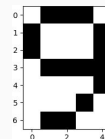


A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

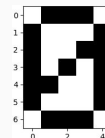


¿Par o impar?

$$F : \mathbb{R}^3 \rightarrow \mathbb{R}$$



¡Es un 9!



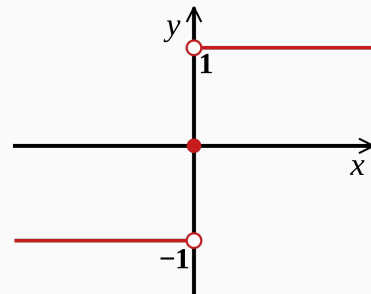
¡Es un 0!

# Implementación

## Funciones de activación

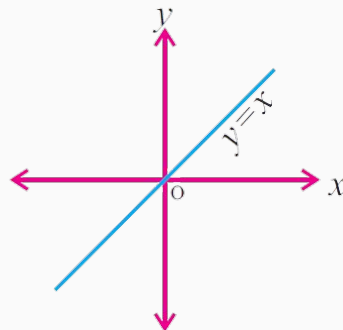
- **Escalón / signo**

$$g(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ -1 & \text{sino} \end{cases}$$



- **Identidad**

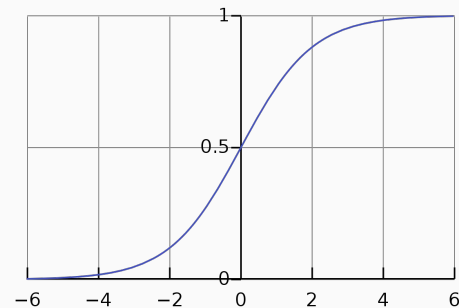
$$g(x) = x$$



## Funciones de activación

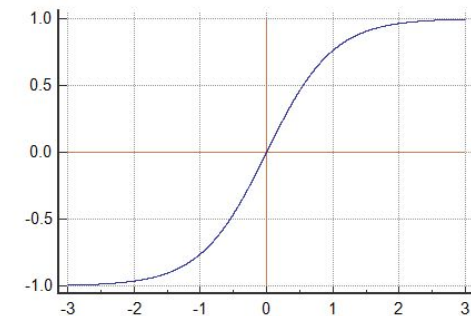
- **Logística**

$$g(x) = \frac{1}{1 + e^{-2\beta x}}$$



- **Tangente hiperbólica**

$$g(x) = \tanh(\beta x)$$



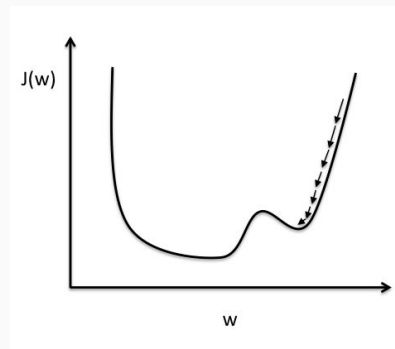
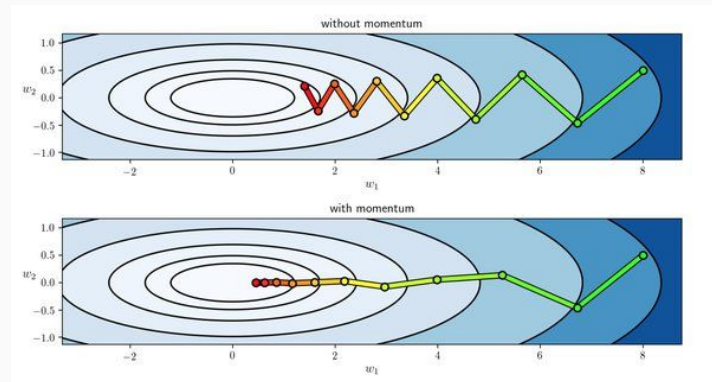
## Adicional

- Momentum**

$$\Delta w_{ij}(t) = -\eta \frac{\partial E}{\partial w_{ij}} + \alpha \Delta w_{ij}(t-1)$$

- Tasa de aprendizaje adaptativa**

$$\Delta \eta = \begin{cases} +a & \text{si } \Delta E < 0 \\ -b\eta & \text{si } \Delta E > 0 \\ 0 & \text{sino} \end{cases}$$



## Notación

$\eta \rightarrow$  Tasa de aprendizaje

$\beta \rightarrow$  Tasa de crecimiento de sigmoidea

$\alpha \rightarrow$  Tasa de momentum

$a \rightarrow$  Aceleración en  $\eta$  adaptativo

$b \rightarrow$  Desaceleración en  $\eta$  adaptativo

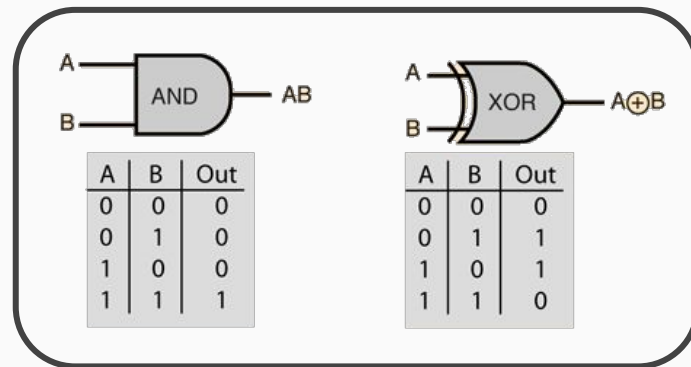
$k \rightarrow$  Cantidad de iteraciones consistentes en  $\eta$  adaptativo



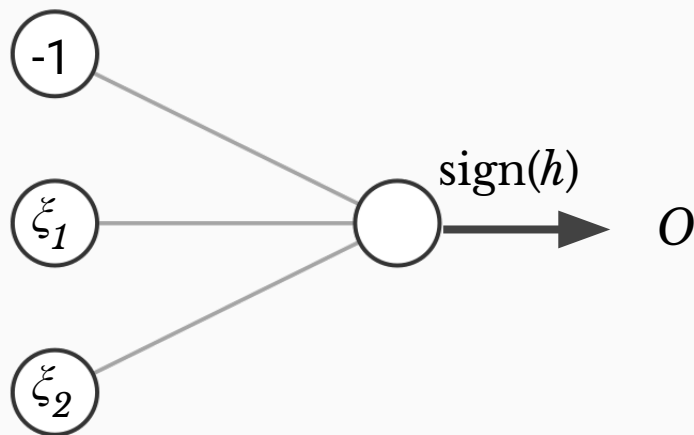
# Ejercicios

## Ejercicio #1

- Utilizar el algoritmo de un **perceptrón simple** con función de activación **escalón** para aprender problemas del tipo **AND** y **XOR**.
- **Analizar la capacidad** del perceptrón para cada problema.



## Ejercicio #1: Arquitectura



### Conjunto de entrenamiento AND

$Input = \{ (-1, -1); (-1, 1); (1, -1); (1, 1) \}$

$Output = \{ -1; -1; -1; 1 \}$

### Conjunto de entrenamiento XOR

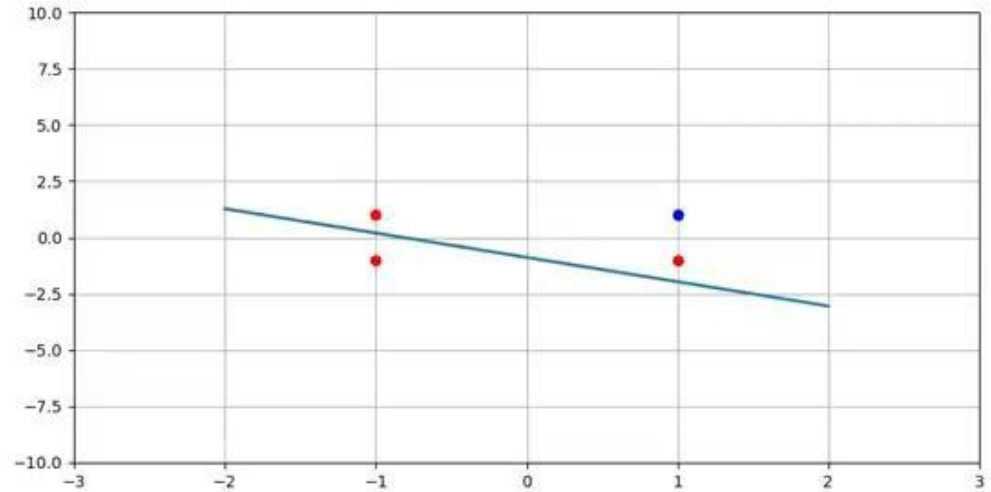
$Input = \{ (-1, -1); (-1, 1); (1, -1); (1, 1) \}$

$Output = \{ -1; 1; 1; -1 \}$

# Ejercicio #1: AND

## Parámetros

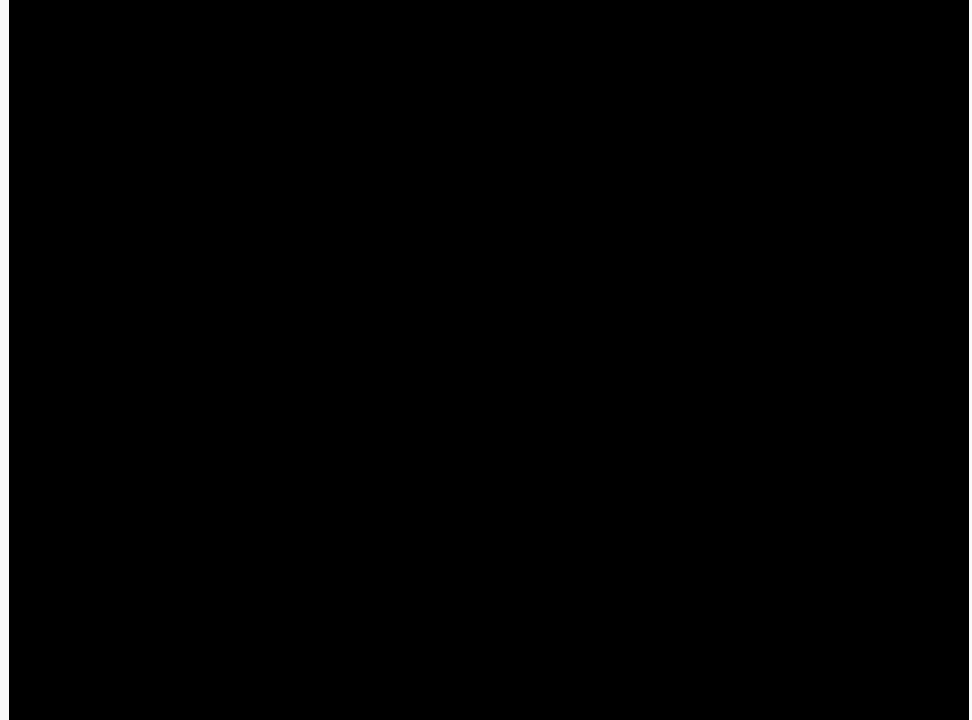
- $\eta = 0.1$
- *Epochs* = 100
- *Tolerancia* =  $1e-8$
- *Rango valores iniciales de pesos* =  $[-1,1]$
- *Umbral* = 1
- *Activación del umbral* = -1



# Ejercicio #1: XOR

## Parámetros

- $\eta = 0.1$
- *Epochs* = 100
- *Tolerancia* =  $1e-8$
- *Rango valores iniciales de pesos* =  $[-1,1]$
- Umbral = 1
- Activación del umbral = -1



## Ejercicio #2:

- Utilizar el algoritmo de un **perceptrón simple** con función de **activación lineal y no lineal** para aprender el problema brindado por la cátedra.
- Analizar la **capacidad de clasificación** del perceptrón.
- Analizar la **capacidad de generalización** del perceptrón simple no lineal.

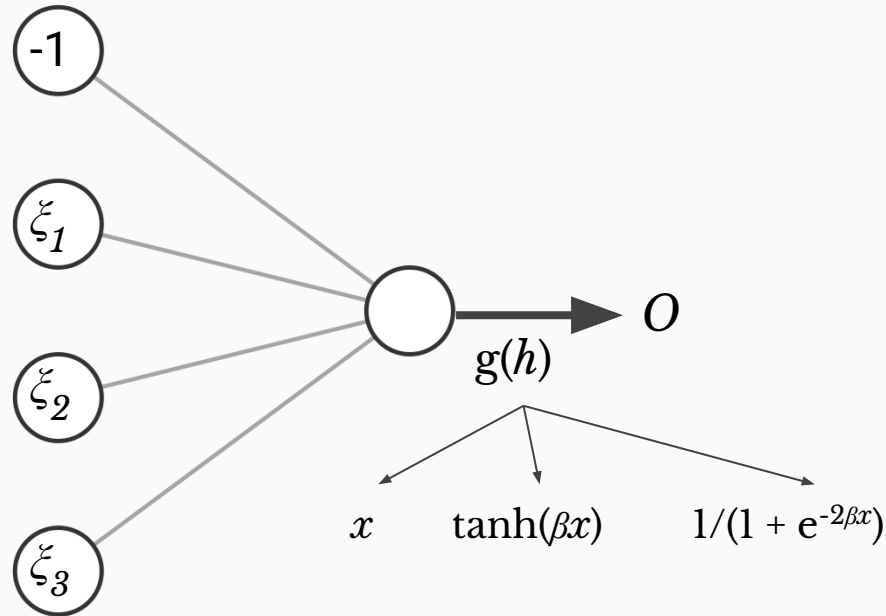
$$F : \mathbb{R}^3 \rightarrow \mathbb{R}$$

## Ejercicio #2:

- Elegir el **mejor conjunto de entrenamiento**.
- Evaluar la **máxima capacidad de generalización** del perceptrón simple no lineal para el conjunto de datos.

$$F : \mathbb{R}^3 \rightarrow \mathbb{R}$$

## Ejercicio #2: Arquitectura



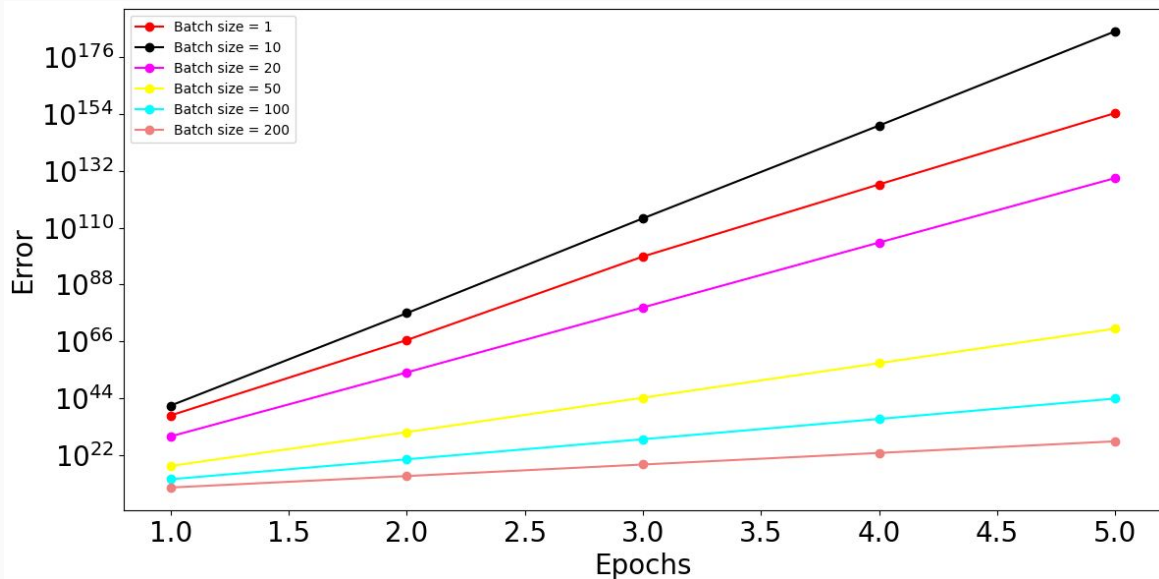


# Ejercicio #2

## Capacidad de Clasificación con perceptrón simple lineal

### Parámetros

- **Tamaños de lotes:**  
{1, 10, 20, 50, 100, 200}
- $\eta = 0.1$
- **Función:**  $g(x) = x$
- **Umbral** = 0.5
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 5



## Ejercicio #2

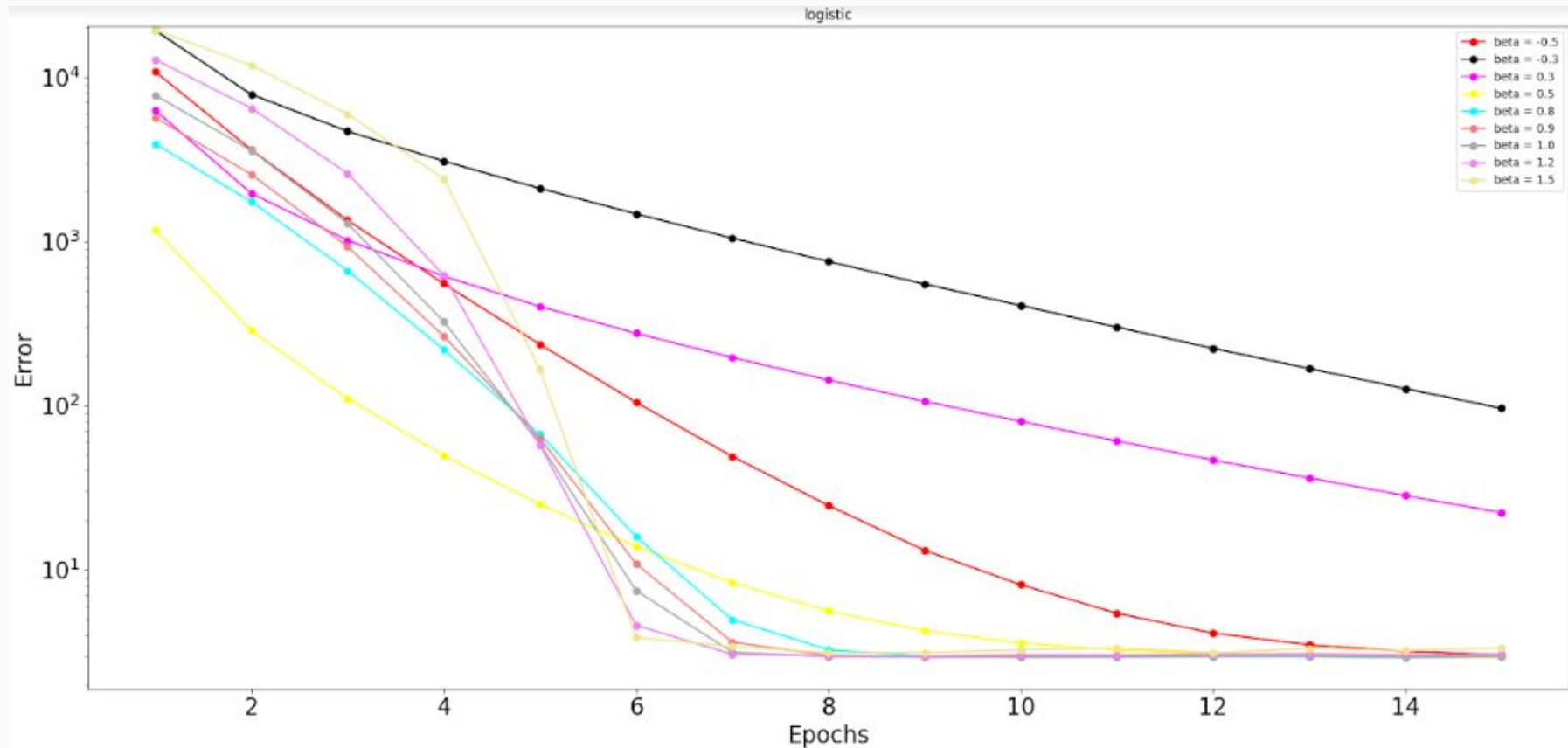
### Capacidad de Clasificación *con perceptrón simple lineal*

Batch size	Epoch				
	1	2	3	4	5
1	2.09E+37	3.19E+66	6.92E+98	5.82E+126	2.42E+154
10	1.36E+41	8.36E+76	4.14E+113	3.54E+149	1.11E+186
20	1.60E+29	9.42E+53	1.36E+79	1.78E+104	1.61E+129
50	5.84E+17	7.15E+30	1.51E+44	3.61E+57	8.56E+70
100	4.08E+12	2.09E+20	1.26E+28	9.05E+35	7.09E+43
200	2.54E+09	6.96E+13	2.03E+18	6.17E+22	1.96E+27



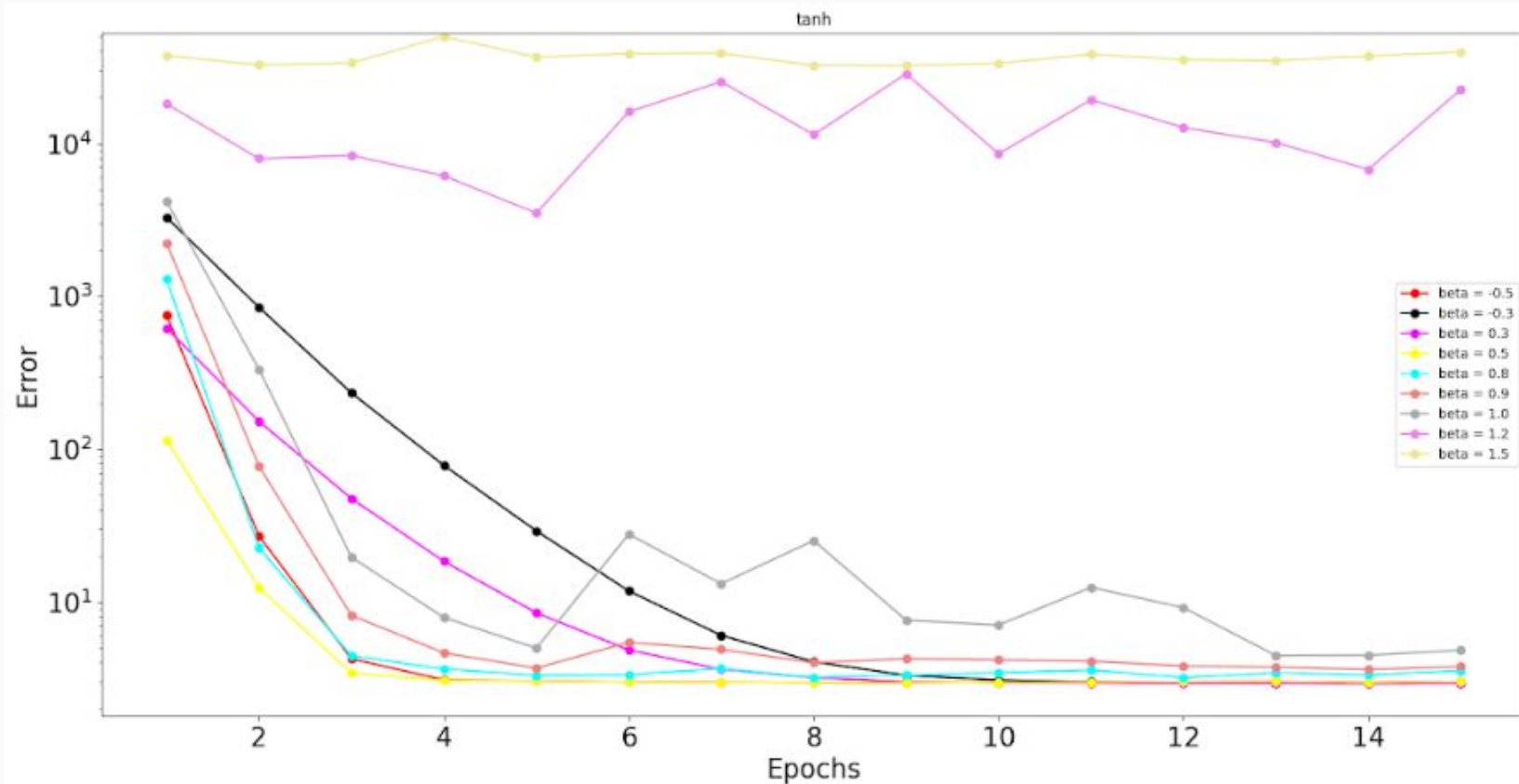
# Perceptrón No Lineal: Logística vs. Tanh

Logística:  $\beta$  óptimo



# Perceptrón No Lineal: Logística vs. Tanh

Tanh:  $\beta$  óptimo

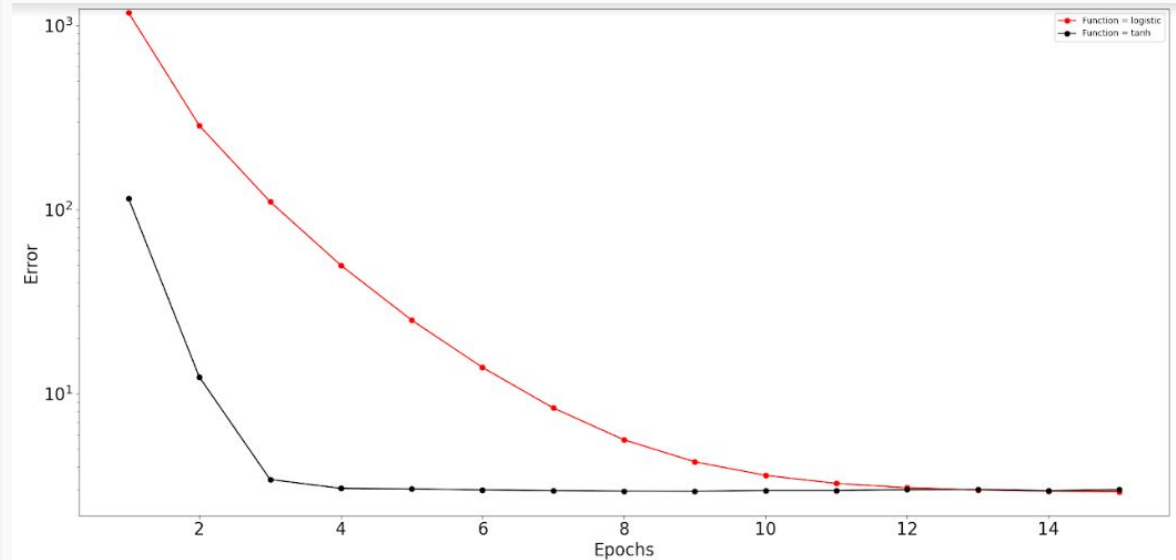


# Ejercicio #2

## Capacidad de Clasificación variando sigmoideas

### Parámetros

- **Incremental**
- $\eta = 0.1$
- **Funciones:**  
 $g_1(x) = 1/(1 + e^{-2\beta x})$   
 $g_2(x) = \tanh(\beta x)$
- $\beta \tanh = \beta \text{ logística} = 0.5$
- **Umbral** = 0.5
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 5

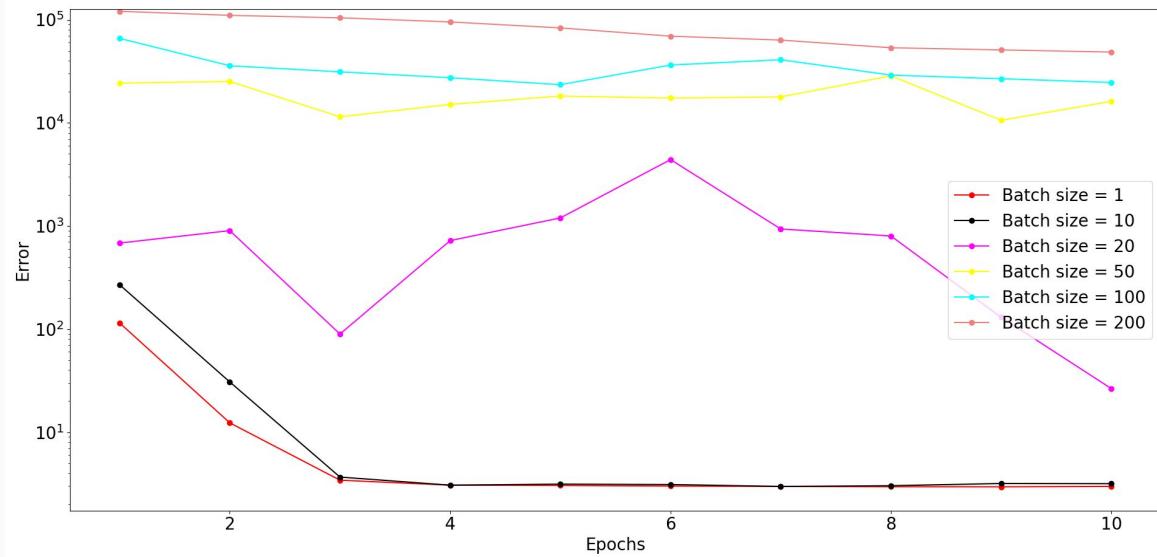


# Ejercicio #2

## Capacidad de Clasificación No lineal variando lotes

### Parámetros

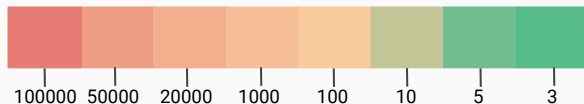
- **Tamaños de lotes:**  
 $\{1, 10, 20, 50, 100, 200\}$
- $\eta = 0.1$
- **Función:**  $g(x) = \tanh(\beta x)$
- $\beta = 0.5$
- **Umbral** = 0.5
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 5



# Ejercicio #2

*Capacidad de Clasificación*  
*No lineal variando lotes*

	Epoch									
Batch size	1	2	3	4	5	6	7	8	9	10
1	114.63	12.31	3.42	3.06	3.04	3.00	2.97	2.96	2.95	2.98
10	268.99	30.69	3.67	3.06	3.13	3.10	2.97	3.02	3.18	3.16
20	682.64	901.00	89.48	721.31	1,196.27	4,401.18	934.96	799.86	130.25	26.46
50	24,283.83	25,199.91	11,457.04	15,107.05	18,193.70	17,400.72	17,862.68	28,581.21	10,597.40	16,167.67
100	65,961.99	35,798.38	31,246.43	27,396.36	23,458.36	36,389.17	40,965.63	29,054.26	26,788.76	24,650.40
200	120,834.97	110,406.69	104,426.95	95,244.62	83,384.30	69,326.13	63,538.57	53,542.20	51,004.79	48,623.22

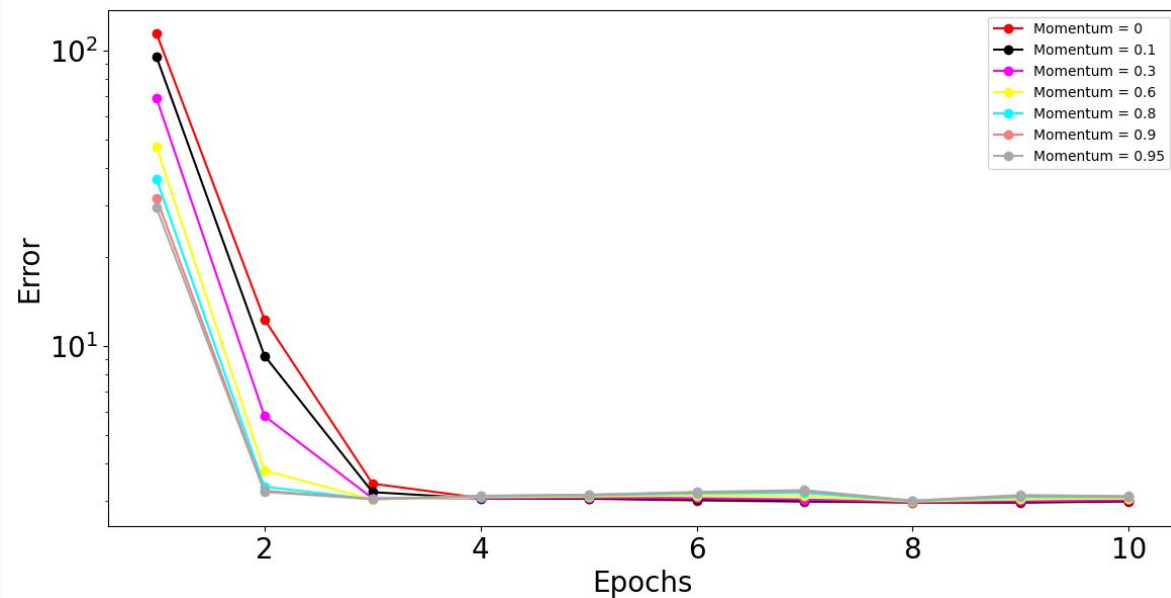


# Ejercicio #2

Capacidad de Clasificación  
*No lineal variando momentum*

## Parámetros

- *Incremental*
- $\alpha \in \{0, 0.1, 0.3, 0.6, 0.8, 0.9, 0.95\}$  (momentum)
- $\eta = 0.1$
- **Función:**  $g(x) = \tanh(\beta x)$
- $\beta = 0.5$
- **Umbral** = 0.5
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 5



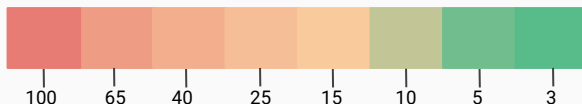


# Ejercicio #2

*Capacidad de Clasificación*

***No lineal variando momentum***

	Epoch									
Momentum	1	2	3	4	5	6	7	8	9	10
0	114.63	12.31	3.42	3.06	3.04	3.00	2.97	2.96	2.95	2.98
0.1	95.44	9.28	3.20	3.05	3.05	3.02	2.99	2.96	2.96	2.99
0.3	69.15	5.80	3.05	3.05	3.07	3.06	3.02	2.97	2.98	3.01
0.6	47.21	3.79	3.02	3.08	3.10	3.12	3.10	2.98	3.03	3.05
0.8	36.86	3.34	3.03	3.10	3.12	3.17	3.18	2.99	3.08	3.08
0.9	31.70	3.24	3.03	3.11	3.13	3.20	3.23	2.99	3.11	3.10
0.95	29.45	3.21	3.04	3.11	3.14	3.21	3.25	3.00	3.13	3.11

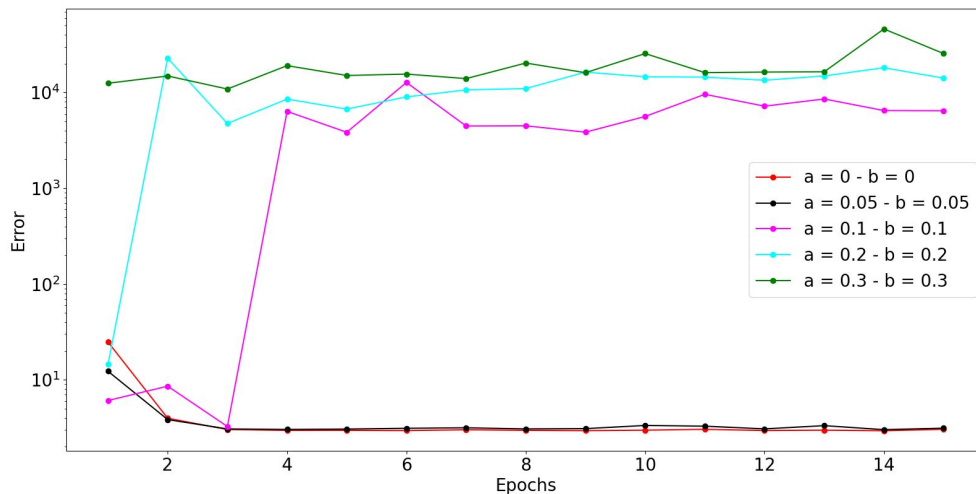


## Ejercicio #2

### Capacidad de Clasificación Variando $\eta$ adaptativo

#### Parámetros

- **Incremental**
- $(a, b) \in \{(0, 0); (0.05, 0.05); (0.1, 0.1); (0.2, 0.2); (0.3, 0.3)\}$
- $k = 10$
- $\eta$  inicial = 0.1,  $\eta_{max} = 1$ ,  $\eta_{min} = 0.0001$
- **Función:**  $g(x) = \tanh(\beta x)$
- $\beta = 0.5$
- **Umbral** = 0.5
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 5

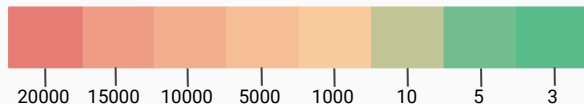


# Ejercicio #2

## Capacidad de Clasificación

### Variando $\eta$ adaptativo

	Epoch									
Adaptive (a, b)	1	2	3	4	5	6	7	8	9	10
(0, 0)	24.77	3.96	3.02	2.96	2.96	2.94	3.00	2.96	2.94	2.97
(0.05, 0.05)	12.27	3.84	3.06	3.03	3.05	3.11	3.15	3.07	3.09	3.33
(0.1, 0.1)	6.07	8.57	3.25	6,363.44	3,850.49	12,772.77	4,476.47	4,498.34	3,858.37	5,631.44
(0.2, 0.2)	14.52	22,857.44	4,755.13	8,553.04	6,726.58	9,007.26	10,669.65	11,010.72	16,363.91	14,608.70
(0.3, 0.3)	12,520.42	14,915.76	10,901.02	19,123.00	15,098.49	15,612.99	14,003.14	20,364.48	16,140.83	25,576.87

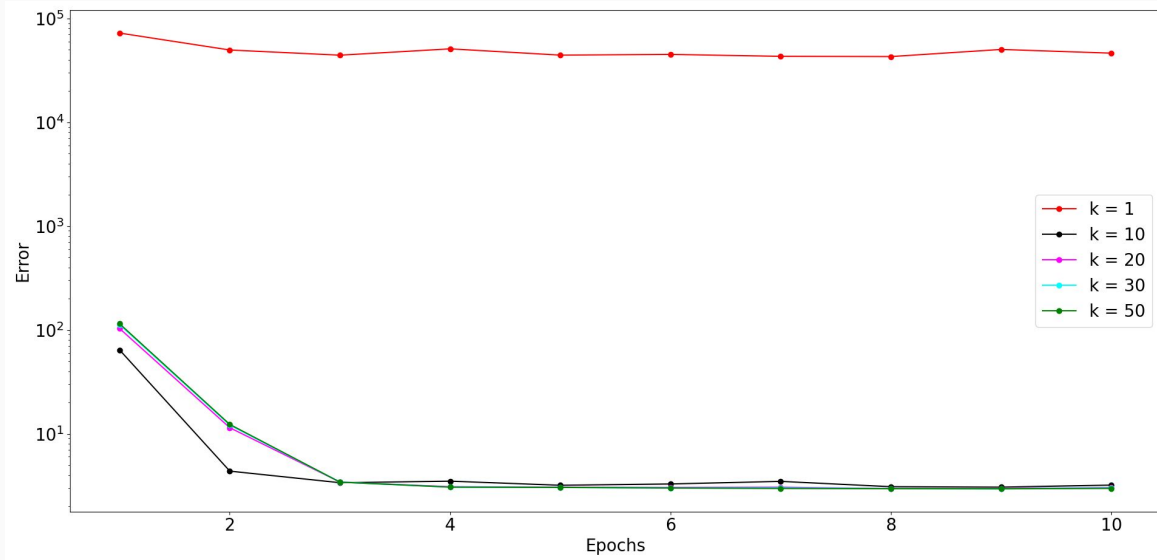


# Ejercicio #2

Capacidad de Clasificación  
**No lineal variando  $k$  en  $\eta$  adaptativo**

## Parámetros

- $(a, b) = (0.05, 0.05)$
- $k \in \{1, 10, 20, 30, 50\}$
- $\eta_{\text{inicial}} = 0.1$ ,  $\eta_{\text{max}} = 1$ ,  $\eta_{\text{min}} = 0.0001$
- **Función:**  $g(x) = \tanh(\beta x)$
- $\beta = 0.5$
- **Umbral** = 0.5
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 5

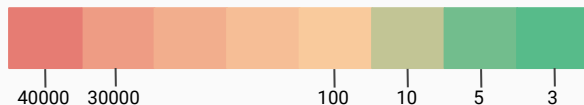


# Ejercicio #2

Capacidad de Clasificación

No lineal variando  $k$  en  $\eta$  adaptativo

	Epoch									
k	1	2	3	4	5	6	7	8	9	10
1	43,836.15	39,483.13	31,729.84	41,707.99	32,805.37	39,693.09	33,984.25	39,047.89	37,623.97	41,751.32
10	64.16	4.37	3.38	3.49	3.19	3.29	3.48	3.09	3.06	3.20
20	103.18	11.44	3.43	3.09	3.06	3.04	3.06	2.97	2.96	3.03
30	112.48	12.11	3.43	3.06	3.04	3.01	3.01	2.96	2.95	3.01
50	114.63	12.31	3.42	3.06	3.04	3.00	2.97	2.96	2.95	2.98

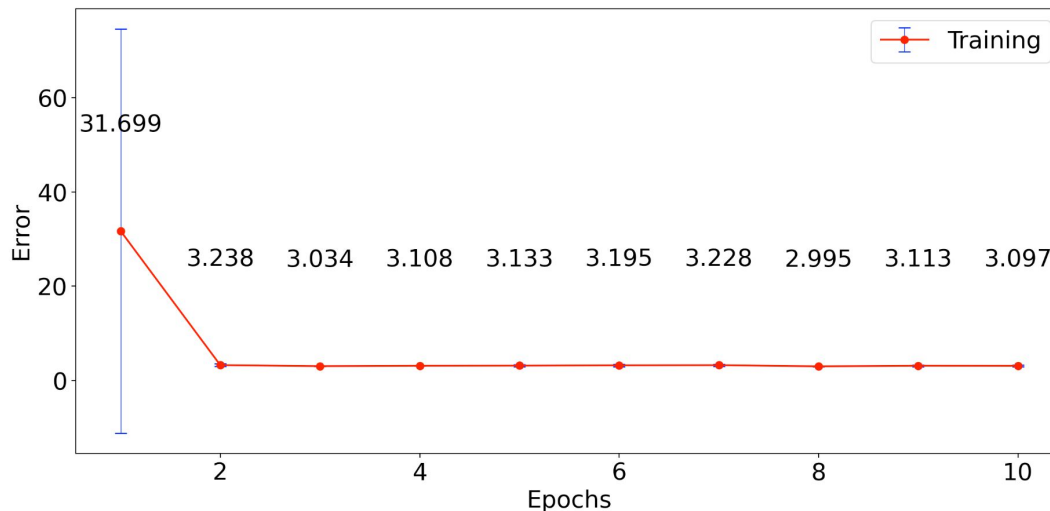


# Ejercicio #2

Capacidad de Clasificación  
*No lineal "óptimo"*

## Parámetros

- *Incremental*
- $\eta = 0.1$
- $\alpha = 0.9$
- **Función:**  $g(x) = \tanh(\beta x)$
- $\beta = 0.5$
- **Umbral** = 0.5
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 5

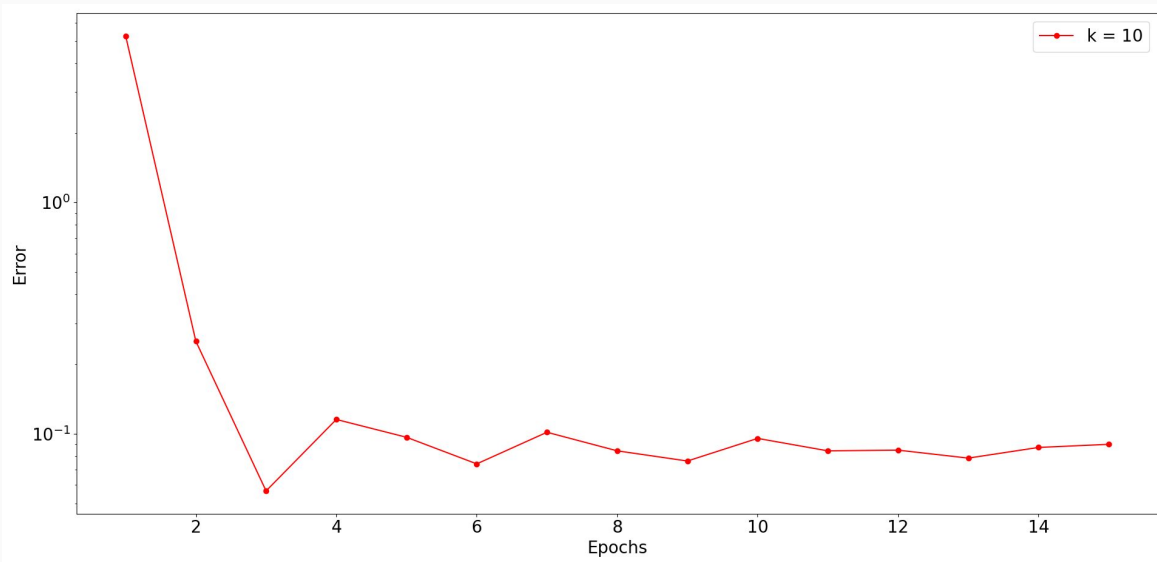


# Ejercicio #2: Perceptrón simple no lineal

## Capacidad de Generalización

### Parámetros

- *Incremental*
- $\eta = 0.1$
- **Función:**  $g(x) = \tanh(\beta x)$
- $\beta = 0.5$
- **Umbral** = 0.5
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 5



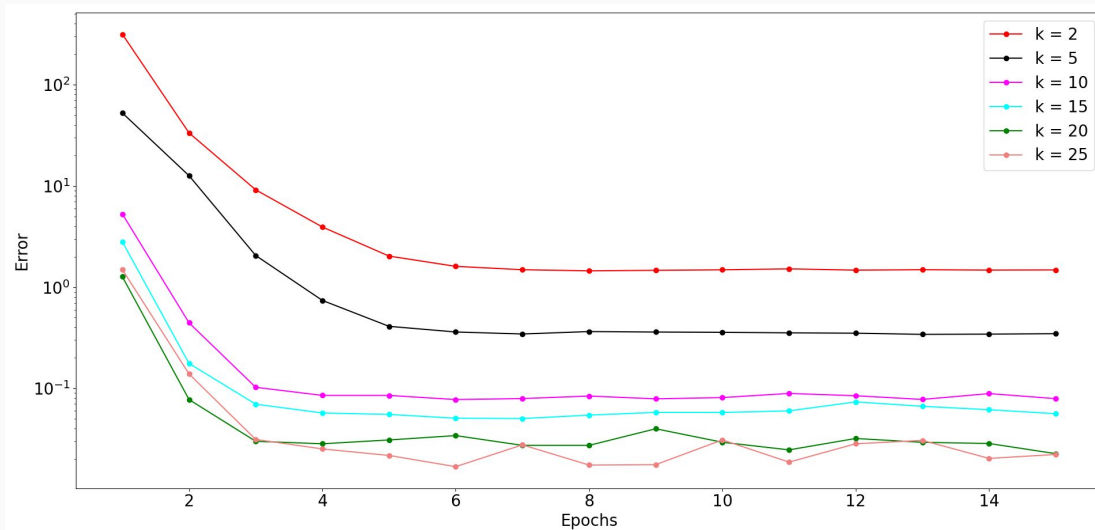
Error promedio: 0.303

# Ejercicio #2: Perceptrón simple no lineal

Máxima Capacidad de Generalización

## Parámetros

- *Incremental*
- $\eta = 0.1$
- Función:  $g(x) = \tanh(\beta x)$
- $\beta = 0.5$
- Umbral = 0.5
- Activación de umbral = -1
- Repeticiones de muestreo = 5

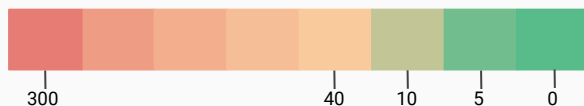




# Ejercicio #2

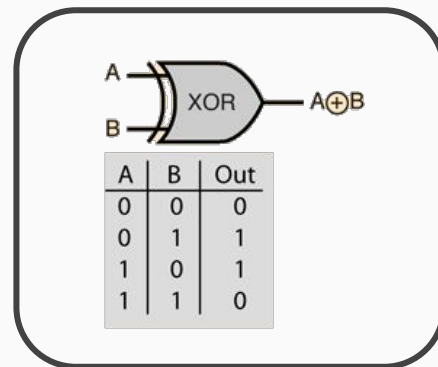
## Máxima Capacidad de Generalización

	Epochs									
k	1	2	3	4	5	6	7	8	9	10
2	315.60	33.45	9.16	3.93	2.03	1.61	1.49	1.45	1.47	1.48
5	52.64	12.63	2.06	0.74	0.41	0.36	0.34	0.36	0.36	0.36
10	5.28	0.44	0.10	0.09	0.09	0.08	0.08	0.08	0.08	0.08
15	2.81	0.18	0.07	0.06	0.06	0.05	0.05	0.05	0.06	0.06
20	1.27	0.08	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.03
25	1.49	0.14	0.03	0.03	0.02	0.02	0.03	0.02	0.02	0.03



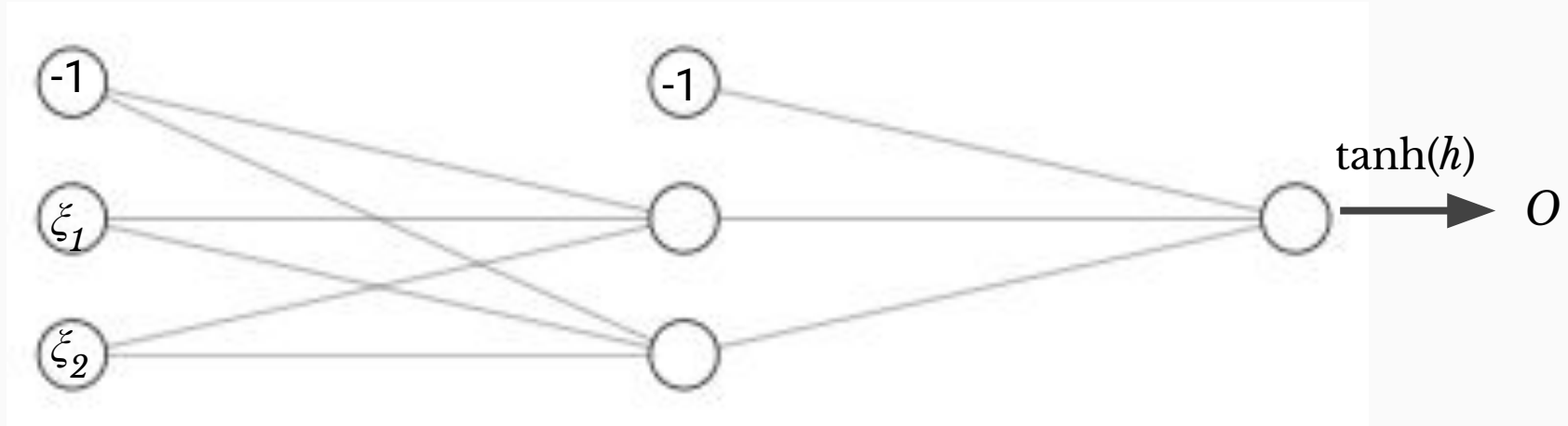
## Ejercicio #3.1: XOR

- Utilizar el algoritmo de un **perceptrón multicapa** para resolver el problema del **XOR** ejercicio #1.
- Analizar la **capacidad de clasificación** del perceptrón.



## Ejercicio #3.1: XOR

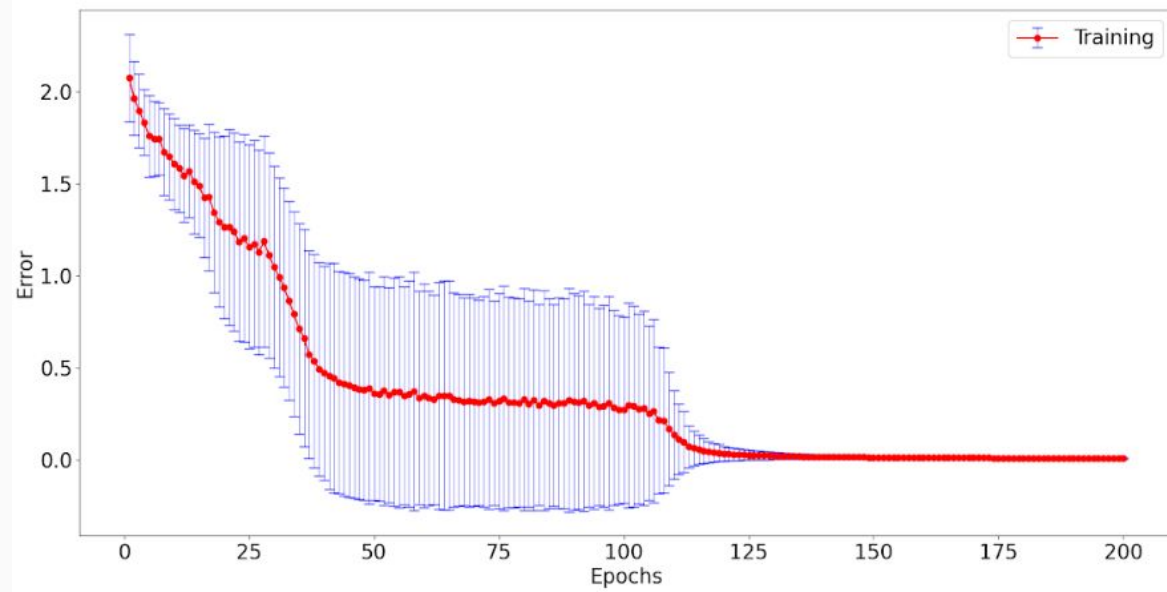
### Arquitectura



# Ejercicio #3.1: XOR

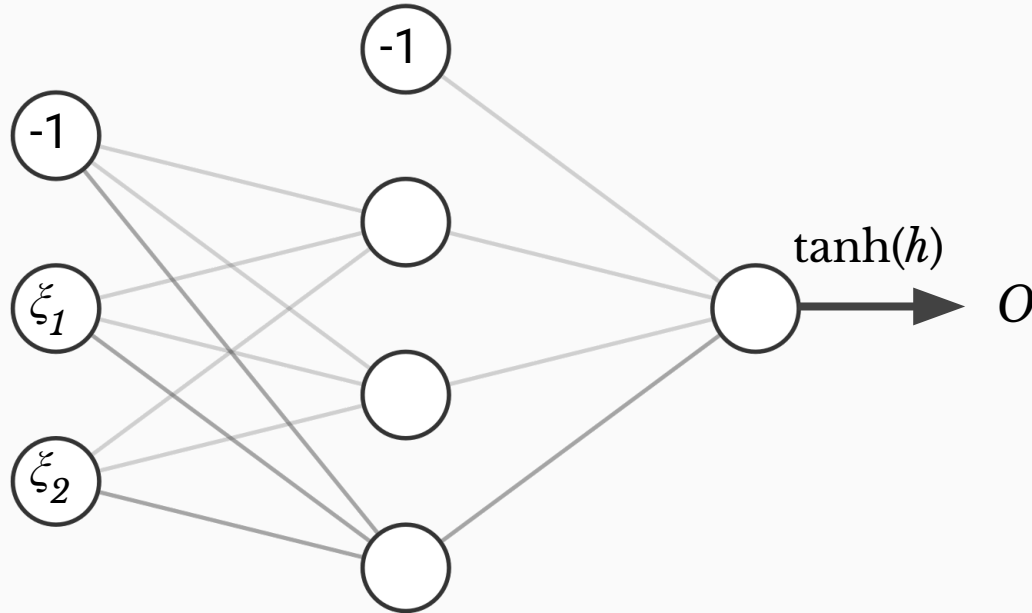
## Parámetros

- $\eta = 0.1$
- $\alpha = 0.9$  (momentum)
- **Epochs** = 200
- **Tolerancia** =  $1e-8$
- **Función:**  $g(x) = \tanh(\beta x)$
- $\beta = 0.5$
- **Cantidad de neuronas en la capa oculta** = 2
- **Rango valores iniciales de pesos** =  $[-1,1]$
- **Umbral** = 0.5



## Ejercicio #3.1: XOR

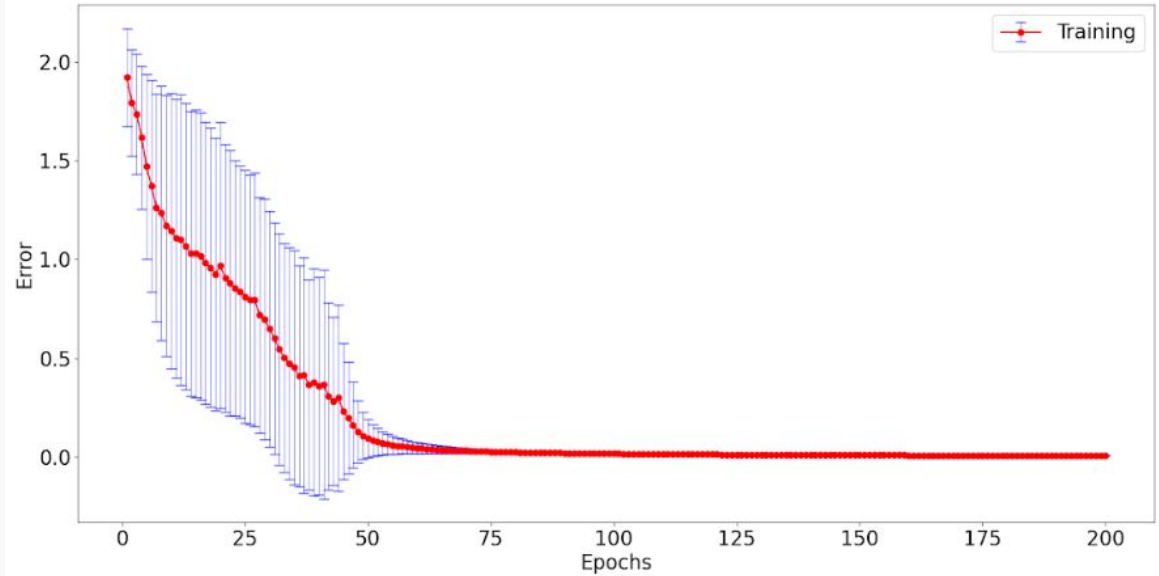
### Arquitectura



# Ejercicio #3.1: XOR

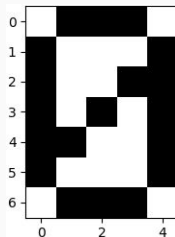
## Parámetros

- $\eta = 0.1$
- $\alpha = 0.9$  (momentum)
- **Epochs** = 200
- **Tolerancia** =  $1e-8$
- **Función:**  $g(x) = \tanh(\beta x)$
- $\beta = 0.5$
- **Cantidad de neuronas en la capa oculta** = 3
- **Rango valores iniciales de pesos** =  $[-1,1]$
- **Umbral** = 0.5



## Ejercicio #3.2.1: Paridad

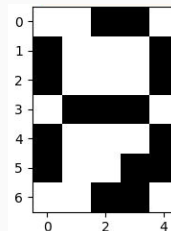
- Dado un **número** representado por una matriz de 5x7 píxeles **determinar** si el mismo es **par** utilizando un **perceptrón multicapa**.
- Analizar la **capacidad de generalización** del perceptrón.



¿Par o impar?

## Ejercicio #3.2.2: Paridad con ruido

- Dado un **número ligeramente modificado** representado por una matriz de 5x7 píxeles **determinar** si el mismo es **par** utilizando un **perceptrón multicapa**.
- Analizar la **capacidad de generalización** del perceptrón.
- Evaluar resultados.

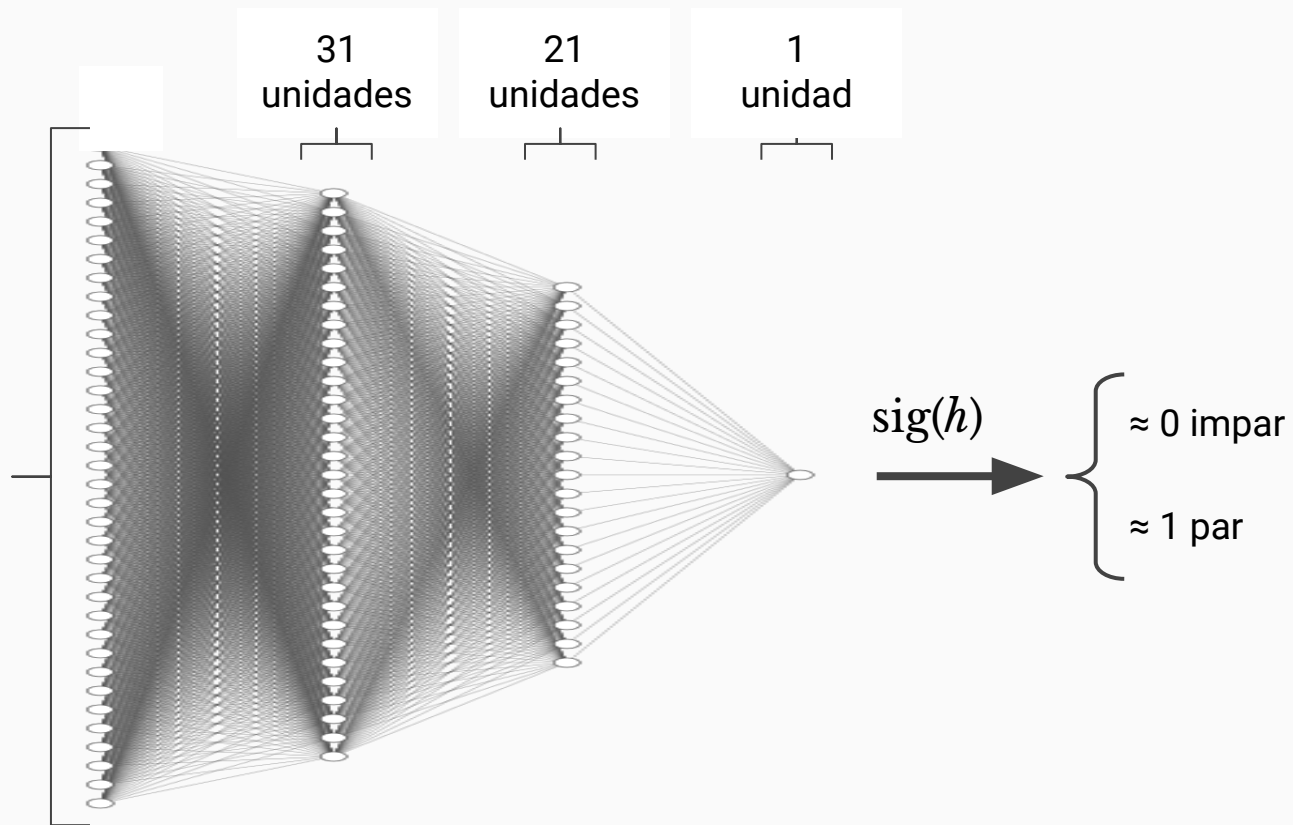


¿Par o impar?



## Ejercicio #3.2: Paridad Arquitectura

Imágenes de  $5 \times 7$  +  
umbral =  
36 unidades

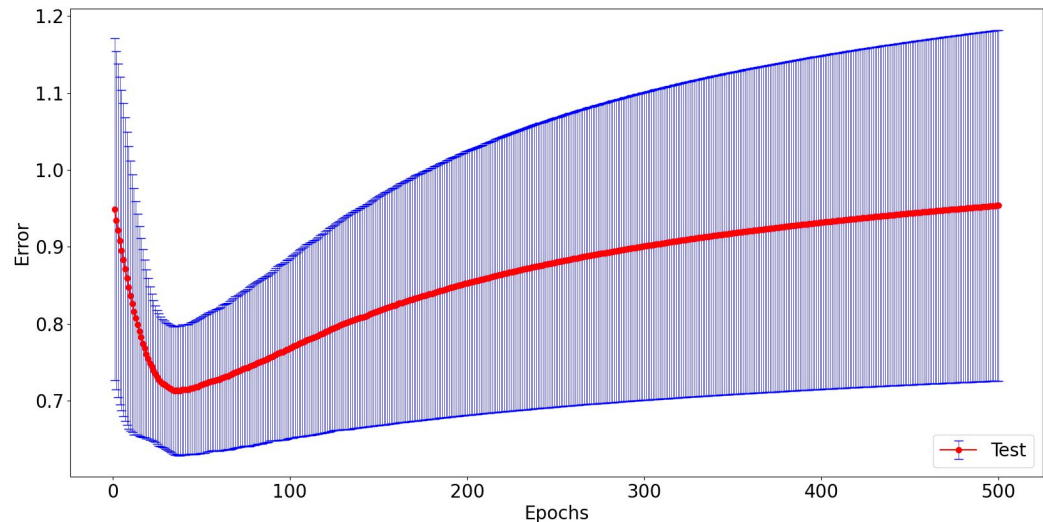


# Ejercicio #3.2: Paridad

Capacidad de Generalización  
(sin ruido)

## Parámetros

- $\eta = 0.1$
- $k = 5$
- **Epochs** = 500
- **Tolerancia** =  $1e-8$
- **Función**:  $g(x) = \text{logistic}(\beta x)$
- $\beta = 0.5$
- **Cantidad de neuronas en la capas ocultas** = [30, 20]
- **Rango valores iniciales de pesos** =  $[-1, 1]$
- **Umbral** = 1
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 10

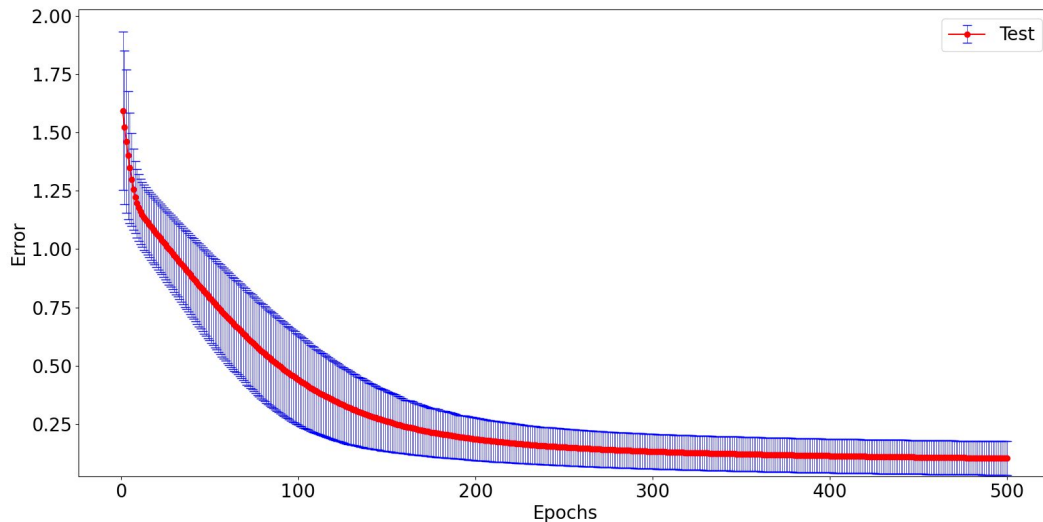


# Ejercicio #3.2: Paridad

Capacidad de Generalización  
(con ruido)

## Parámetros

- $\eta = 0.1$
- p. ruido = 0.1
- Umbral = 1
- Epochs = 500
- Tolerancia =  $1e-8$
- Función:  $g(x) = \text{logistic}(\beta x)$
- $\beta = 0.5$
- Cantidad de neuronas en la capas ocultas = [30, 20]
- Rango valores iniciales de pesos = [-1,1]
- k = 2 (10 con ruido para prueba)
- Activación de umbral = -1
- Repeticiones de muestreo = 10

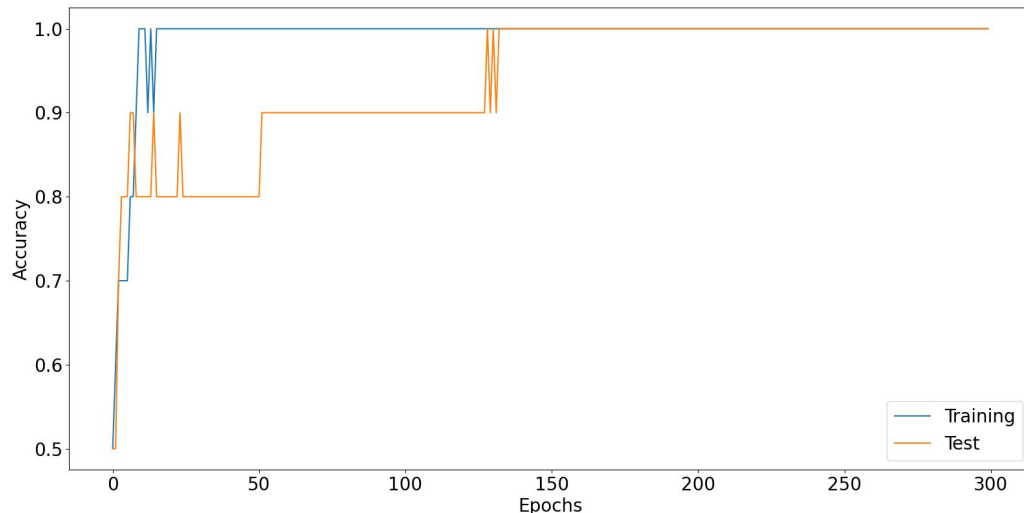


# Ejercicio #3.2: Paridad

## Métricas

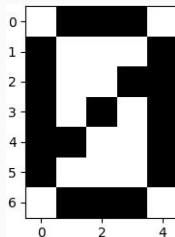
## Parámetros

- $\eta = 0.1$
- p. ruido = 0.1
- Umbral = 1
- **Epochs** = 300
- **Tolerancia** =  $1e-8$
- **Función**:  $g(x) = \text{logistic}(\beta x)$
- $\beta = 0.5$
- **Cantidad de neuronas en la capas ocultas** = [30, 20]
- **Rango valores iniciales de pesos** = [-1,1]
- $k = 2$  (10 con ruido para prueba)
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 10



## Ejercicio #3.3.1: Números

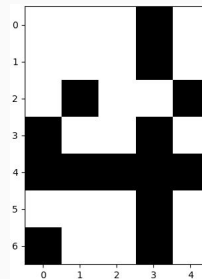
- Dado un **número** representado por una matriz de 5x7 píxeles **determinar qué número es**.
- Analizar la **capacidad de generalización** del perceptrón.



¿Qué número es?

## Ejercicio #3.3.2: Números con ruido

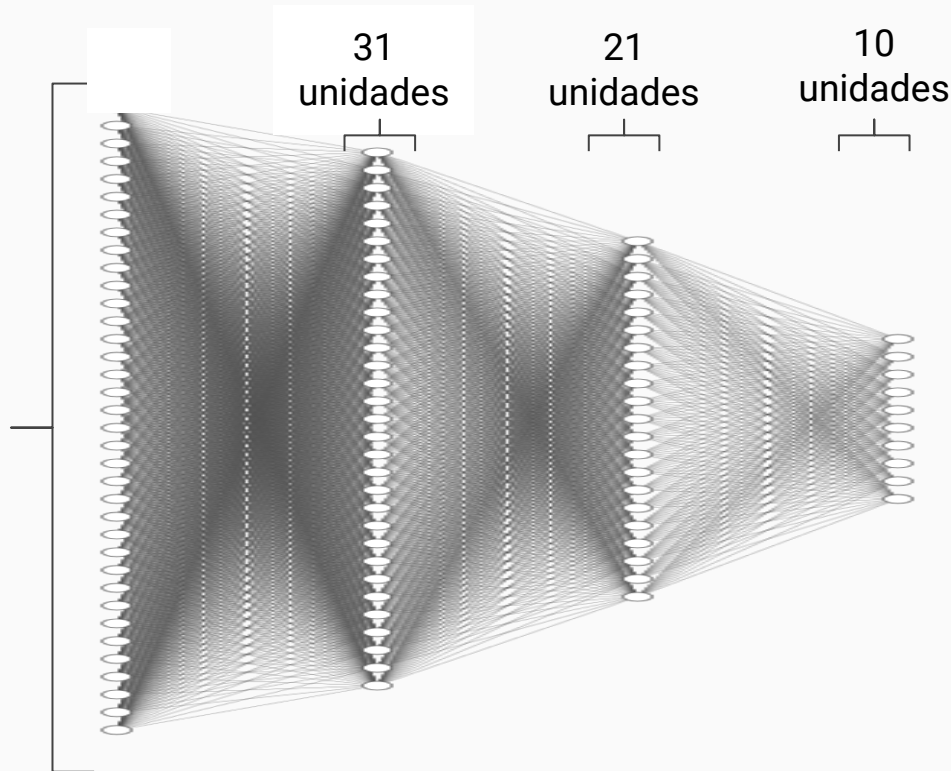
- Dado un **número ligeramente modificado** representado por una matriz de 5x7 píxeles **determinar qué número es.**
- Analizar la **capacidad de generalización** del perceptrón.
- Evaluar los resultados



¿Qué número es?

## Ejercicio #3.3: Números Arquitectura

Imágenes de 5x7  
+ umbral =  
36 unidades



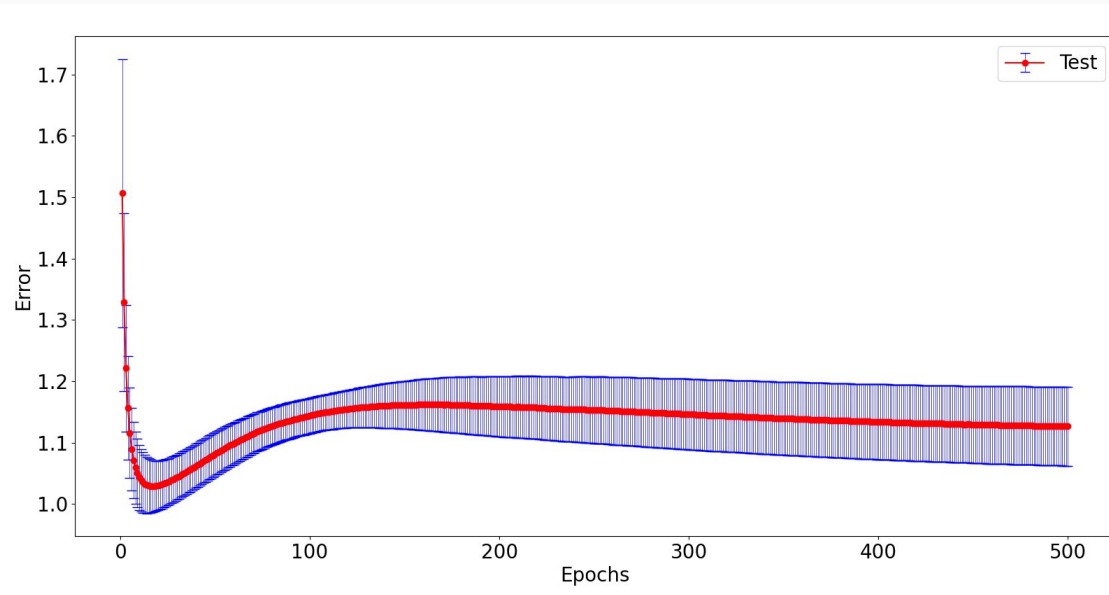
$$\left. \begin{array}{c} \text{ } \end{array} \right\} \rightarrow \arg \max_i \{O_i\}_{i=0}^{i=9}$$

# Ejercicio #3.3: Números

*Capacidad de Generalización  
(sin ruido)*

## Parámetros

- $\eta = 0.1$
- $k = 5$
- **Epochs** = 500
- **Tolerancia** =  $1e-8$
- **Función:**  $g(x) = \text{logistic}(\beta x)$
- $\beta = 0.5$
- **Cantidad de neuronas en la capas ocultas** = [30, 20]
- **Rango valores iniciales de pesos** = [-1,1]
- **Umbral** = 1
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 10



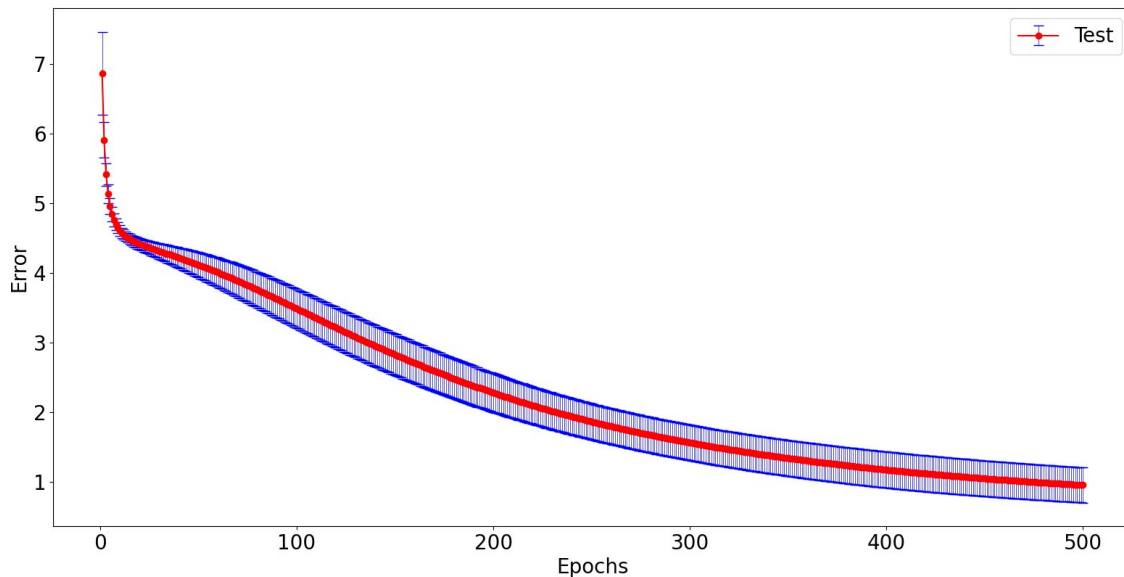


# Ejercicio #3.3: Números

## Capacidad de Generalización (con ruido)

### Parámetros

- $\eta = 0.1$
- Umbral = 1
- **Epochs** = 500
- **Tolerancia** =  $1e-8$
- **Función**:  $g(x) = \text{logistic}(\beta x)$
- $\beta = 0.5$
- **Cantidad de neuronas en la capas ocultas** = [30, 20]
- **Rango valores iniciales de pesos** = [-1,1]
- $k = 2$  (10 con ruido para prueba)
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 10

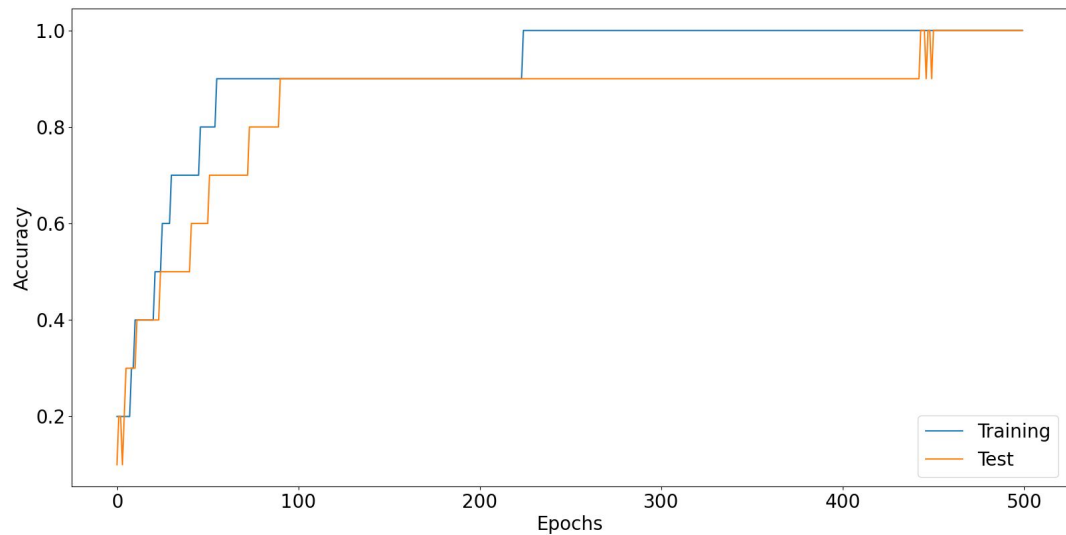


# Ejercicio #3.3: Números

## Métricas

## Parámetros

- $\eta = 0.1$
- p. ruido = 0.1
- Umbral = 1
- **Epochs** = 500
- **Tolerancia** =  $1e-8$
- **Función**:  $g(x) = \text{logistic}(\beta x)$
- $\beta = 0.5$
- **Cantidad de neuronas en la capas ocultas** = [30, 20]
- **Rango valores iniciales de pesos** = [-1,1]
- $k = 2$  (10 con ruido para prueba)
- **Activación de umbral** = -1
- **Repeticiones de muestreo** = 10



# Conclusiones

# Conclusiones

- El perceptrón simple no puede resolver todos los tipos de problemas.
- La elección del tipo de perceptrón simple puede llevar a una convergencia o una divergencia en el resultado.
- Un aprendizaje lento permite aprender problemas con mayor precisión, aunque conlleva un mayor costo computacional.
- En un problema de clasificación, el conjunto de entrenamiento determina las clases que la red puede clasificar.
- El aprendizaje en lotes mejora el rendimiento computacional, aunque puede impactar negativamente en el aprendizaje de la red.

**Gracias por su atención**