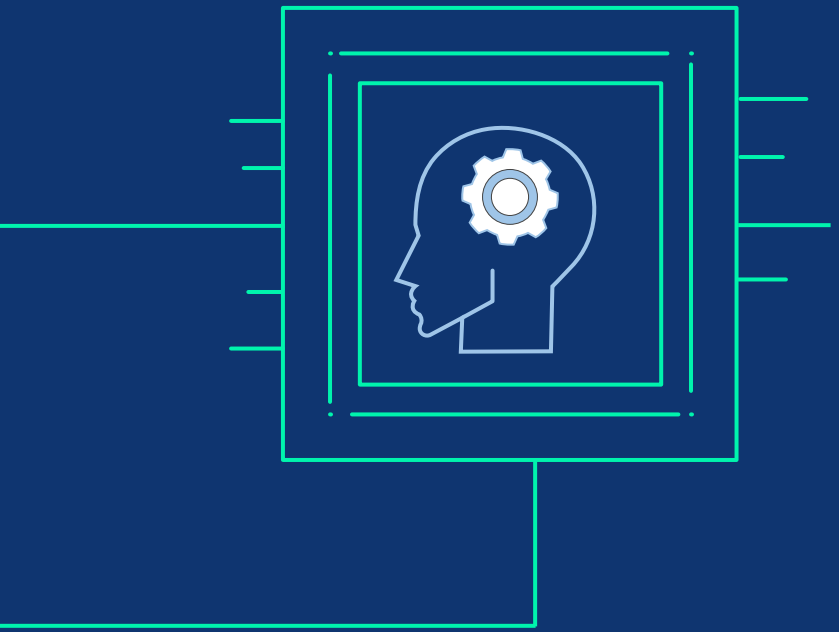
PERCEPTRON SIMPLE Y MULTICAPA

INTRODUCCIÓN

TABLE OF CONTENTS

01

PERCEPTRÓN SIMPLE

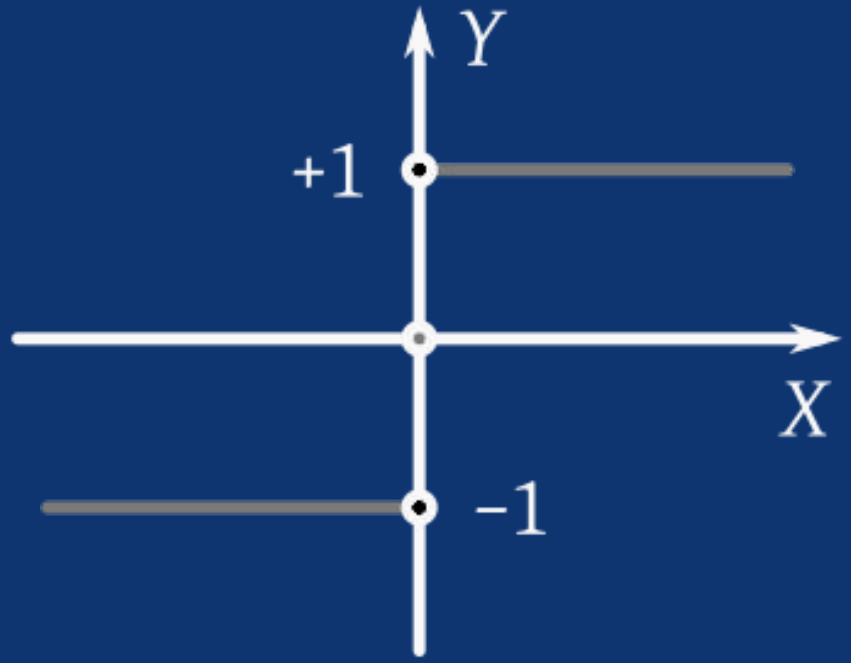


CAPACIDADES

El perceptrón simple tiene la capacidad de resolver problemas linealmente separables

EJERCICIO 1

FUNCION DE ACTIVACION SIGNO



AND

ES FÁCILMENTE SEPARABLE

Si representamos el problema en un gráfico, es fácil encontrar una recta que separe los dos grupos de puntos

AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1

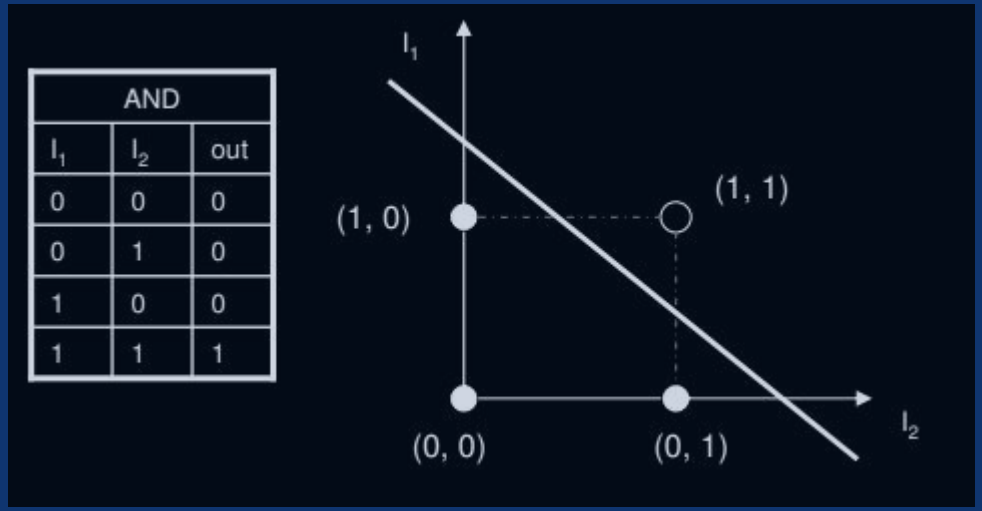


Gráfico 1

NUESTROS RESULTADOS

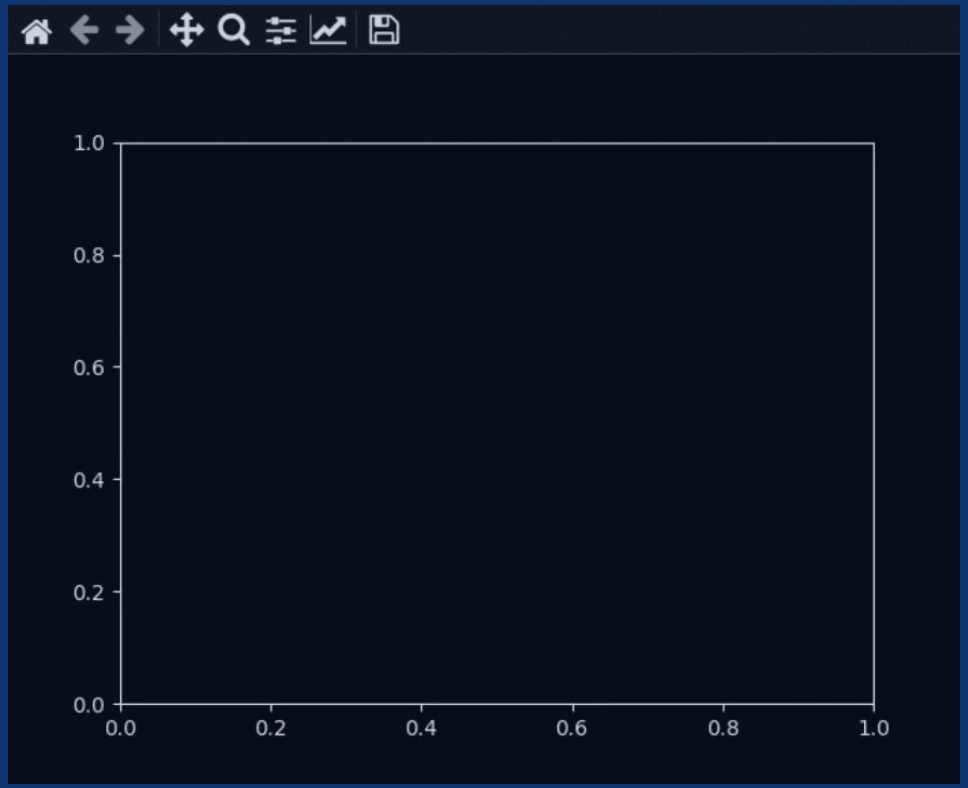


Gráfico 2

OR

OCURRE LO MISMO

En este problema también es fácil encontrar una recta que separe los dos grupos

OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1

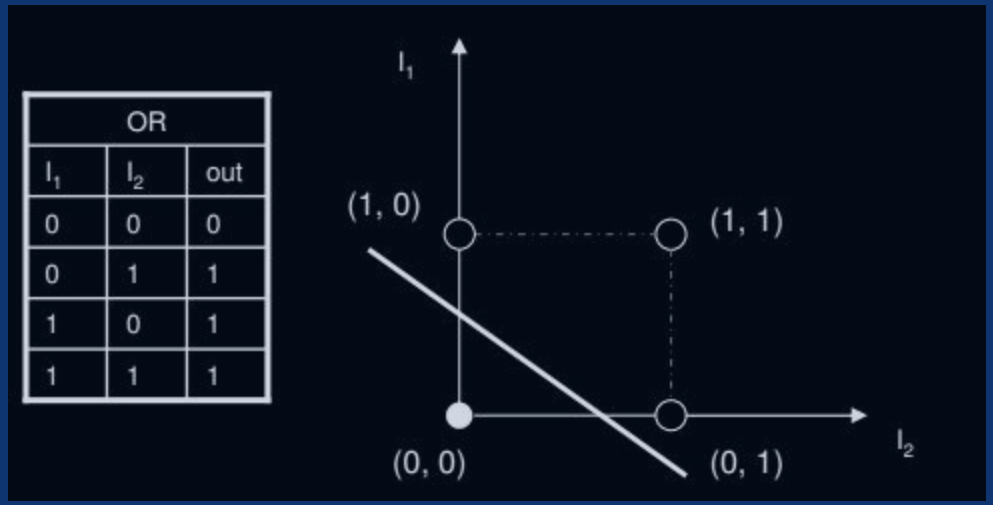


Gráfico 3

```
Finished training
{
    "configuration": {
        "weights": [
            0.030658292387169564,
            0.20764962503139014
        ],
        "bias": 0.21144625113954924
    },
    "training": {
        "params": {
            "epsilon": 0,
            "epochs": 100,
            "max_it_same_bias": 10000,
            "training_level": 0.01
        },
        "analysis": {
            "abs_error": 0.0,
            "mean_error": 0.0,
            "standard_deviation": 0.0
        }
    }
}
```


NUESTROS RESULTADOS

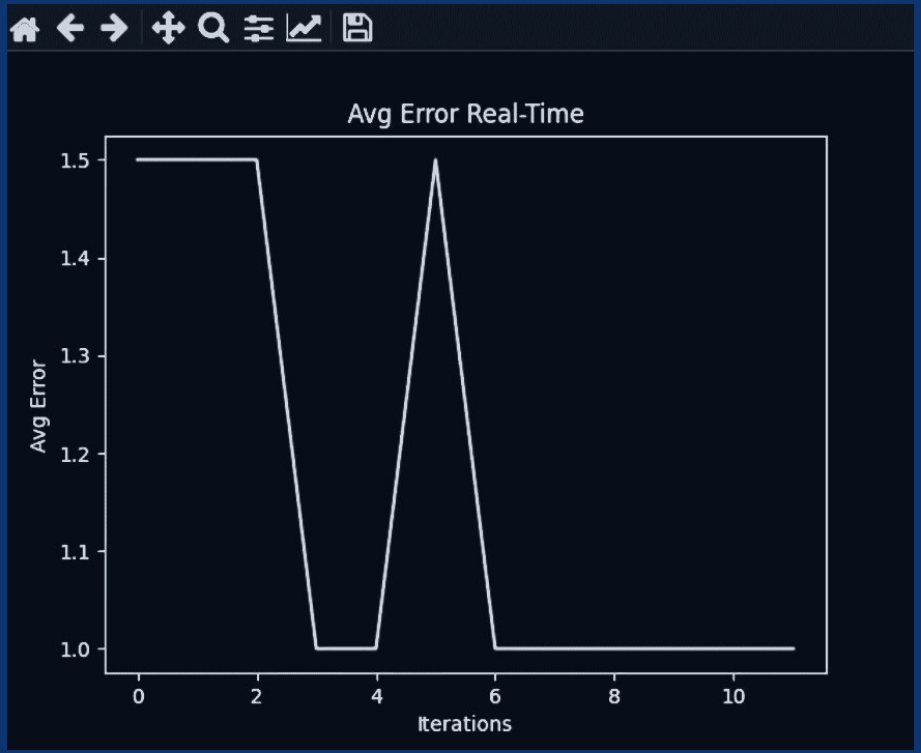


Gráfico 4

XOR

EL PROBLEMA CON XOR

En este caso podemos observar que es imposible encontrar una recta que separe los dos grupos. El problema no es linealmente separable.

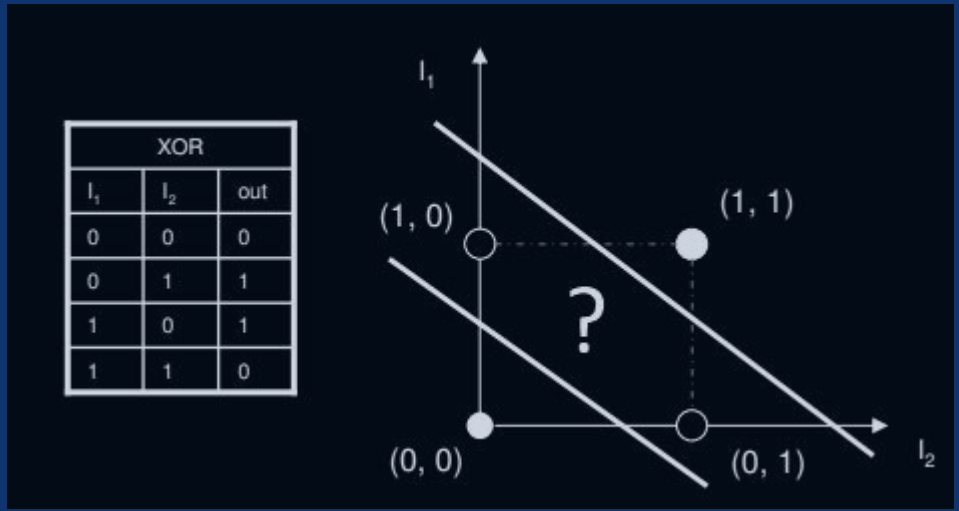


Gráfico 5

```
EPOCHS (100) passed
Finished training
{
    "configuration": {
        "weights": [
            -0.012657189466690372,
            0.3571898059710032 5
        ],
        "bias": -0.3477976524130717
    },
    "training": {
        "params": {
            "epsilon": 0,
            "epochs": 100,
            "max_it_same_bias": 10000,
            "training_level": 0.01
        },
        "analysis": {
            "abs_error": 2.0,
            "mean_error": 0.5,
            "standard_deviation": 1.0
        }
    }
}
Model saved: 'saves\ej1_xor.json'
```

NUESTROS RESULTADOS

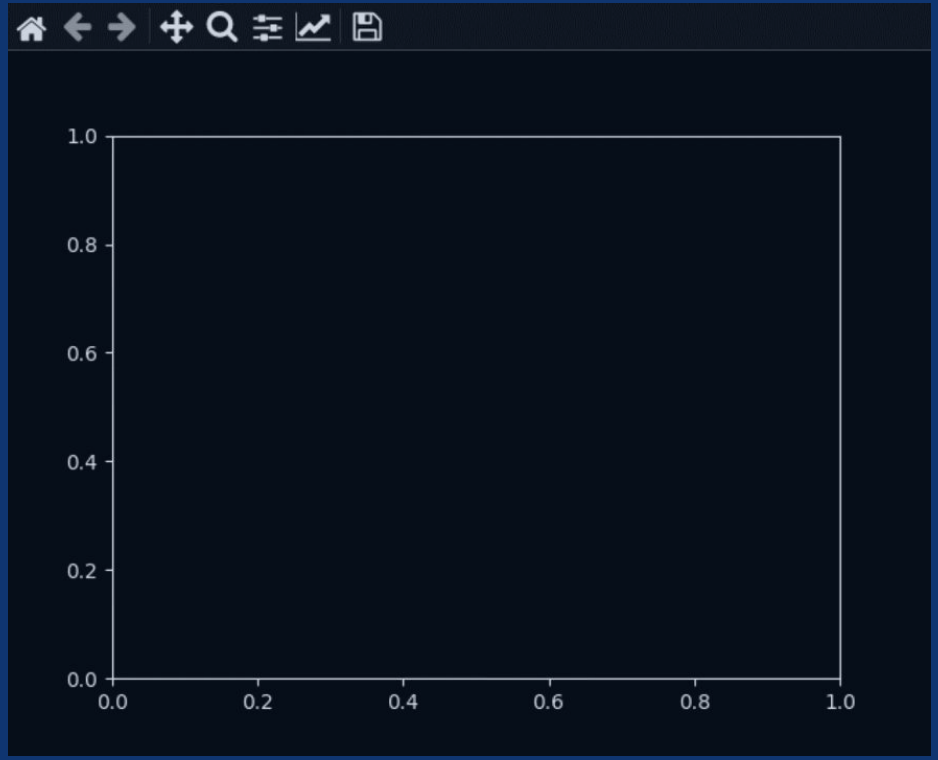


Gráfico 6

EJERCICIO 2

FUNCIÓN DE ACTIVACIÓN LINEAL

```
EPOCHS (100) passed
Finished training
{
    "configuration": {
        "weights": [
            0.13780726543951027,
            0.14999224888818915,
            0.15148767319430662
        ],
        "bias": -0.15538499427485805
    },
    "training": {
        "params": {
            "epsilon": 0.001,
            "epochs": 100,
            "max_it_same_bias": 10000,
            "training_level": 0.01
        },
        "analysis": {
            "abs_error": 31.739686356799456,
            "mean_error": 0.15869843178399728,
            "standard_deviation": 0.20640333611211661
        }
    }
}
Model saved: 'saves\ej2_lineal.json'
```


PERCEPTRÓN SIMPLE LINEAL

Análisis de la capacidad de aprendizaje

	MEAN ERROR
TRAINING	0.01586984

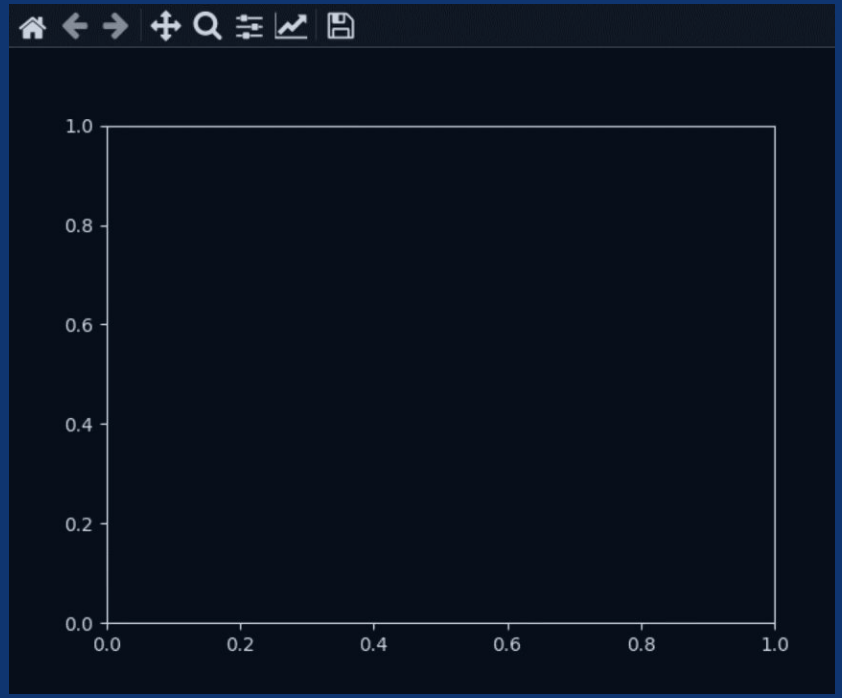
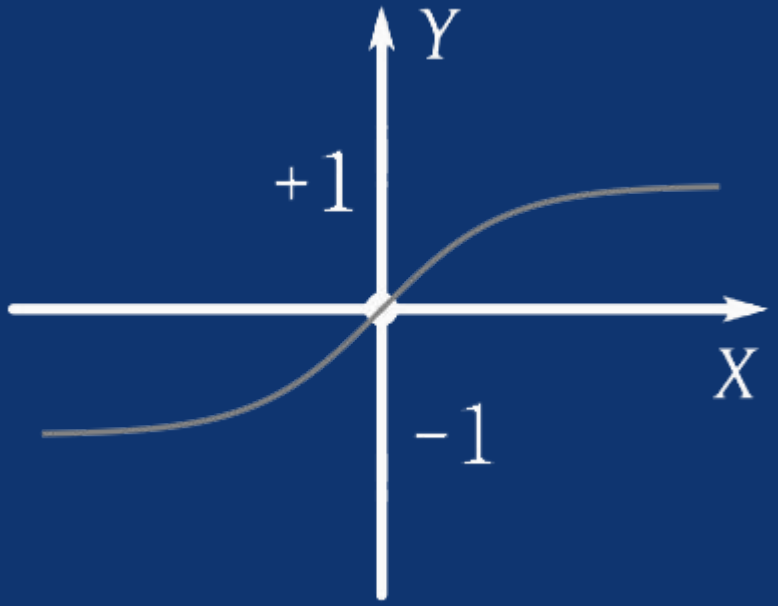
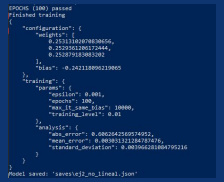


Gráfico 7

FUNCIÓN DE ACTIVACIÓN TANH





PERCEPTRÓN SIMPLE NO LINEAL

Análisis de la capacidad de aprendizaje

	MEAN ERROR
TRAINING	0.00303132

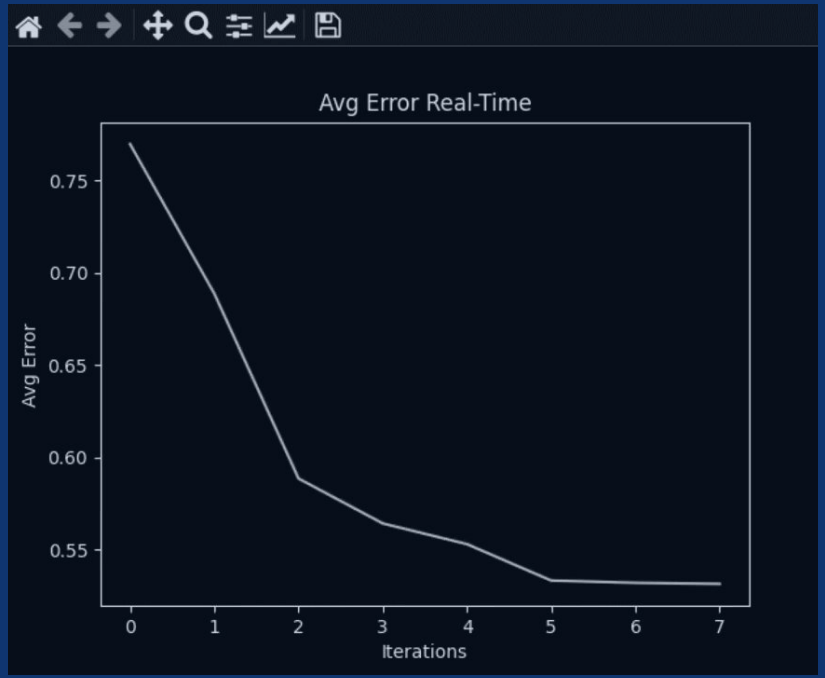
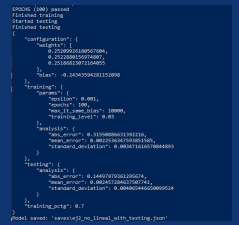


Gráfico 8



PERCEPTRÓN SIMPLE NO LINEAL

Análisis de capacidad de Generalización usando 0.7 para entrenar

	MEAN ERROR
TRAINING	0.00225363
TESTING	0.00245728

Gráfico 9

UTILIZACIÓN DEL MOMENTUM

Caso no lineal con sin momentum

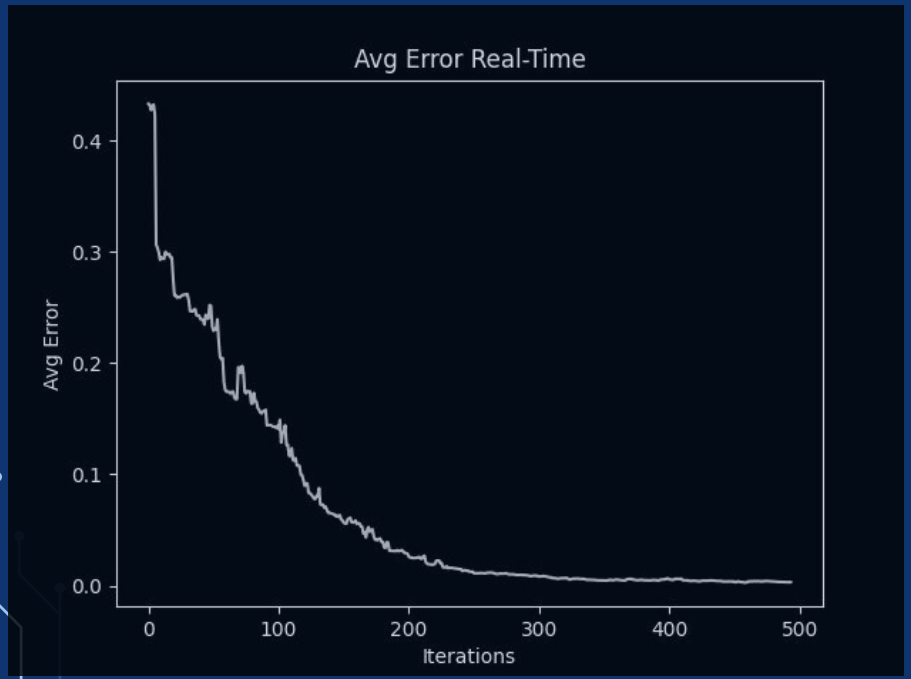


Gráfico 10

Caso no lineal con momentum

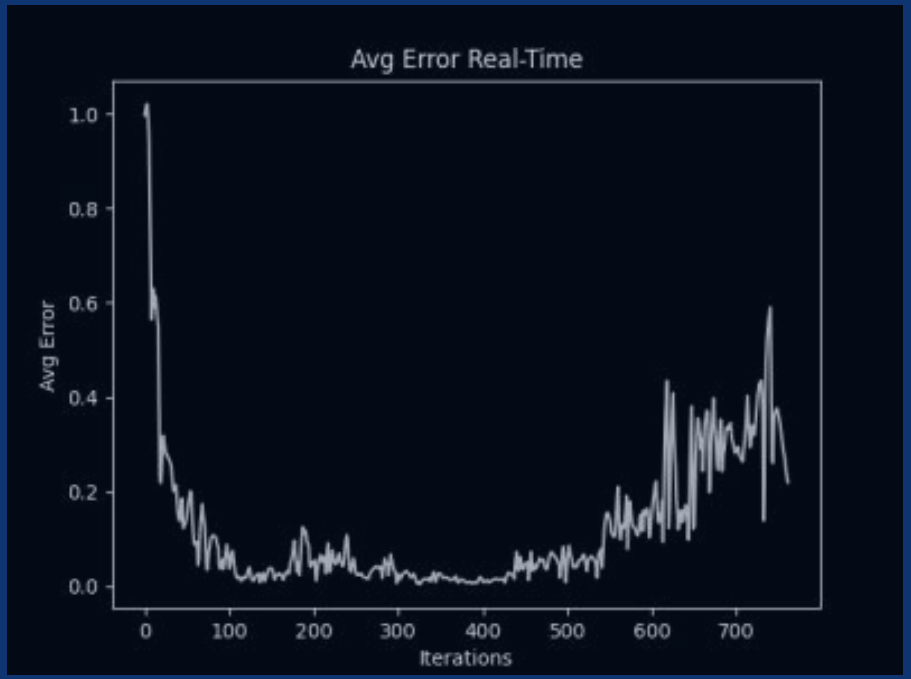


Gráfico 11

VALIDACIÓN CRUZADA

Resultados que generaron los mejores training avg error

	TESTING BLOCK IDX	TRAINING AVG	TESTING AVG
0,6	1	0,0018	0,0026
0,65	1	0,0018	0,0025
0,7	2	0,0022	0,0028
0,75	2	0,0021	0,0031
0,8	3	0,0021	0,0029
0,85	4	0,0022	0,0034
0,9	9	0,0022	0,0035
0,95	4	0,0022	0,0027

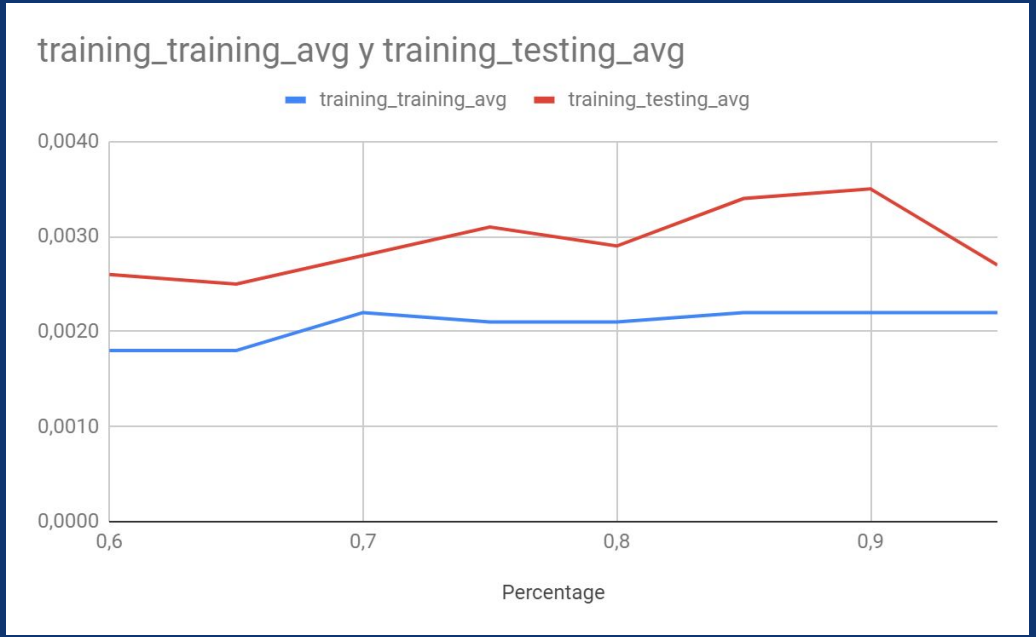


Gráfico 12

VALIDACIÓN CRUZADA

Resultados que generaron los mejores testing avg error

	TESTING BLOCK IDX	TRAINING AVG	TESTING AVG
0,6	0	0,0032	0,0025
0,65	0	0,0029	0,0022
0,7	0	0,0029	0,0023
0,75	1	0,0024	0,0022
0,8	1	0,0025	0,0022
0,85	5	0,0025	0,0017
0,9	8	0,0025	0,0011
0,95	1	0,0023	0,0008

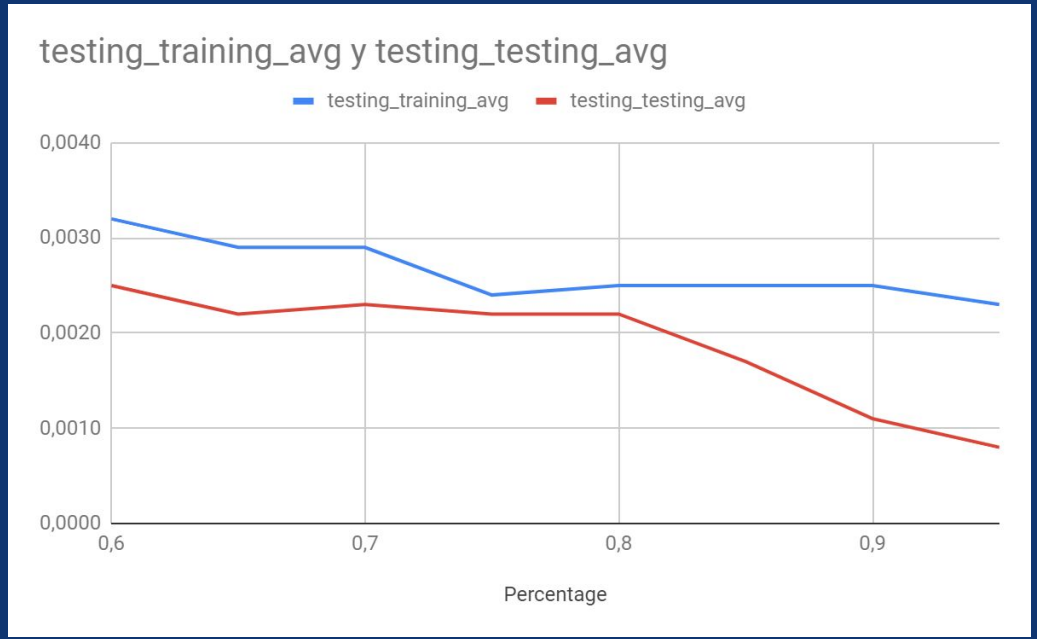
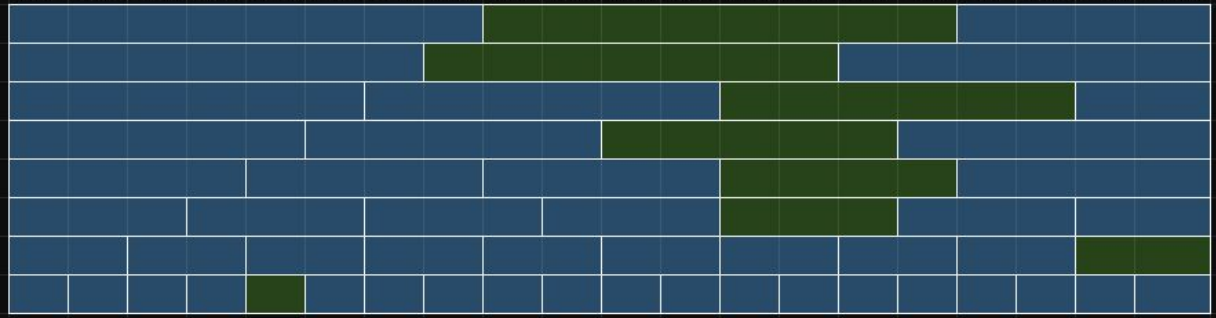


Gráfico 13

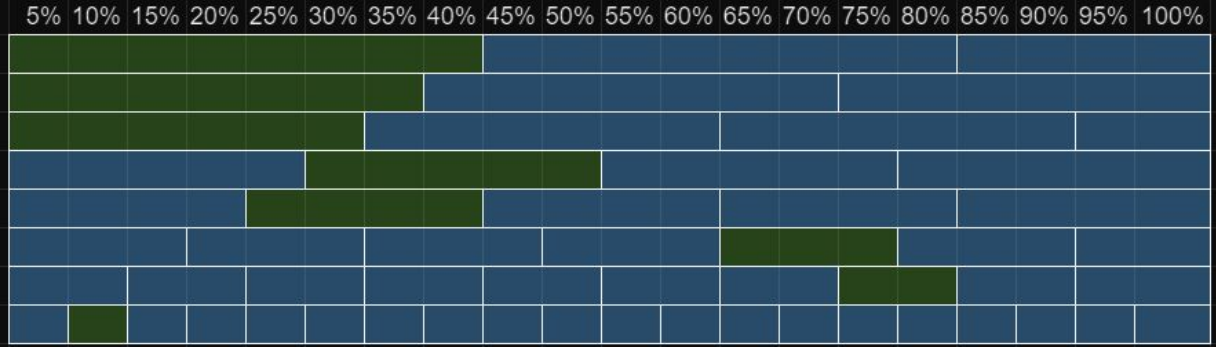
Con esto se entreno	
Con esto se testeo	

ANALIZAMOS

Mejor Training		
se usa 40% para testing	1	60%
se usa 35% para testing	1	65%
se usa 30% para testing	2	70%
se usa 25% para testing	2	75%
se usa 20% para testing	3	80%
se usa 15% para testing	4	85%
se usa 10% para testing	9	90%
se usa 5% para testing	4	95%



Mejor Testing		
se usa 40% para testing	0	60%
se usa 35% para testing	0	65%
se usa 30% para testing	0	70%
se usa 25% para testing	1	75%
se usa 20% para testing	1	80%
se usa 15% para testing	5	85%
se usa 10% para testing	8	90%
se usa 5% para testing	1	95%



02

PERCEPTRÓN MULTICAPA

CAPACIDADES

Con el agregado de más capas, ahora nuestra red no solo puede crear planos para separar sino que curvas, lo que aumenta el alcance de la red neuronal

EJERCICIO 3

XOR

VOLVIENDO AL XOR

Con el uso de una red neuronal, es posible crear una curva que separe los puntos en el xor

XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0

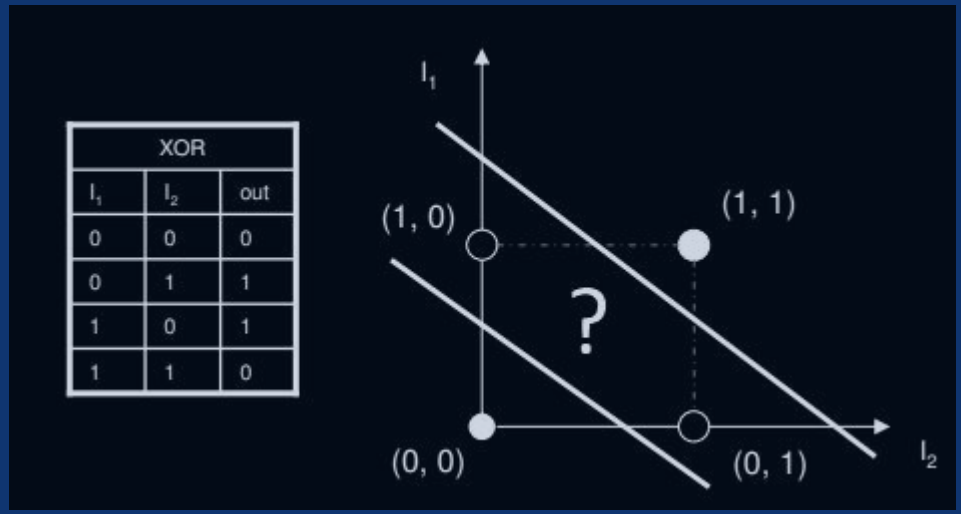
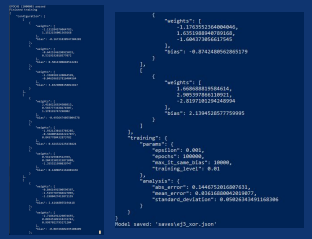
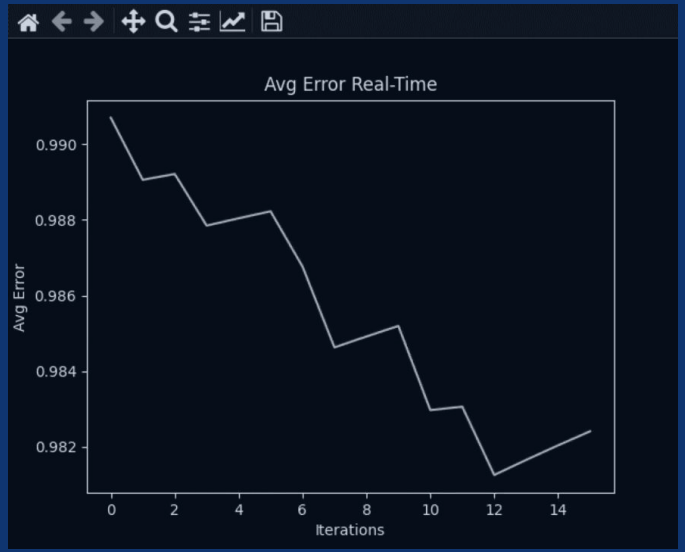


Gráfico 14



NUESTROS RESULTADOS

	MEAN ERROR
TRAINING	0.03616880



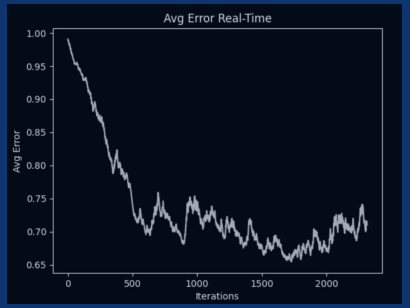
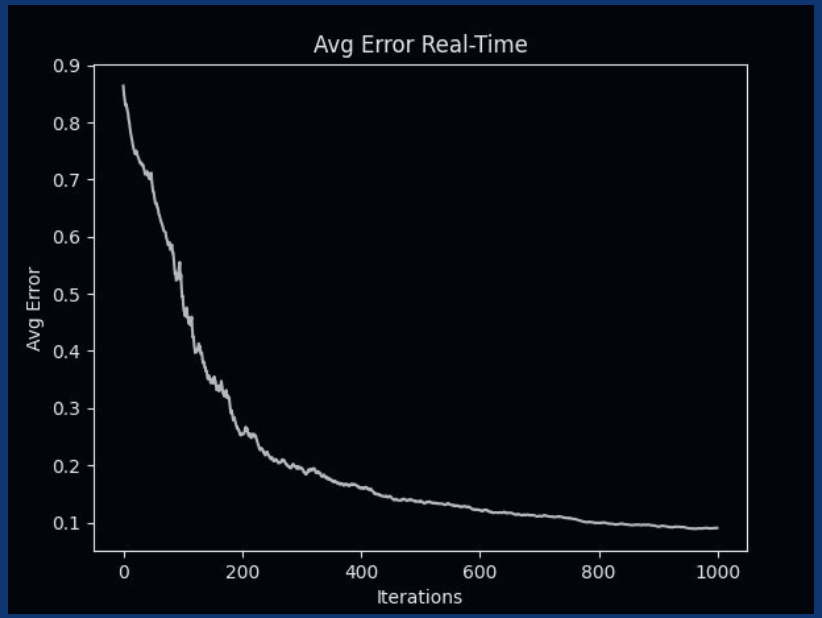


Gráfico 15

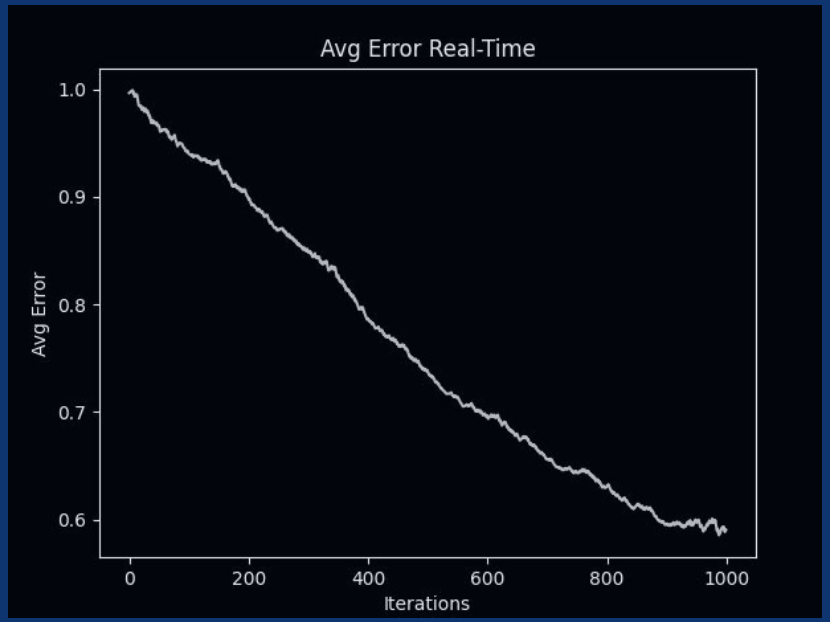
Ejemplo realizado con 100000 epochs y con una red con 3 hidden layers de 3 neuronas cada una

```
"training": {
    "params": {
        "epsilon": 0.001,
        "epochs": 250,
        "max_it_same_bias": 10000,
        "training_level": 0.01
    },
    "analysis": {
        "abs_error": 0.35494058971368525,
        "mean_error": 0.08873514742842131,
        "standard_deviation": 0.10159475641995036
    }
}
```


NUESTROS RESULTADOS



4 hidden layers de 8 neuronas cada una



2 hidden layers de 3 neuronas cada una

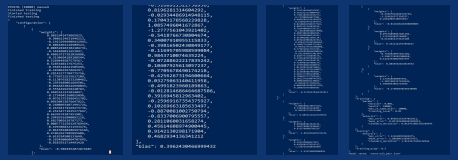
CLASIFICACIÓN POR PARIDAD

IDENTIFICACIÓN DE NÚMEROS PARES

En el caso de los números inicialmente partimos de la hipótesis de que no tiene porque necesariamente poder generalizar la paridad de un número dado la forma que tiene, esto se comprueba con los datos.

	TRAINING	TESTING
ACCURACY	1	0.5
PRECISION	1	0.5
RECALL	1	1
F1 SCORE	1	0.666

Porcentaje de entrenamiento 0.5



NUESTROS RESULTADOS

	MEAN ERROR
TRAINING	0.01192630
TESTING	1.81139739

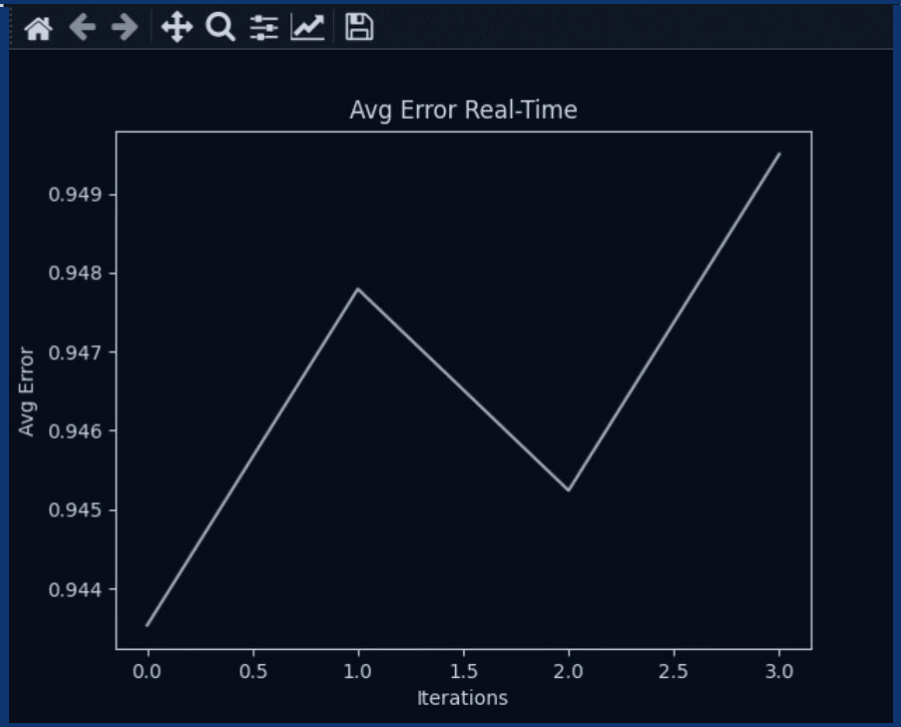


Gráfico 16

03

CONCLUSIONES

PATRONES

Si bien en un principio las redes neuronales parecen mágicas, necesitan poder identificar patrones reales en el material de entrenamiento si queremos que logren generalizar exitosamente

MOMENTUM

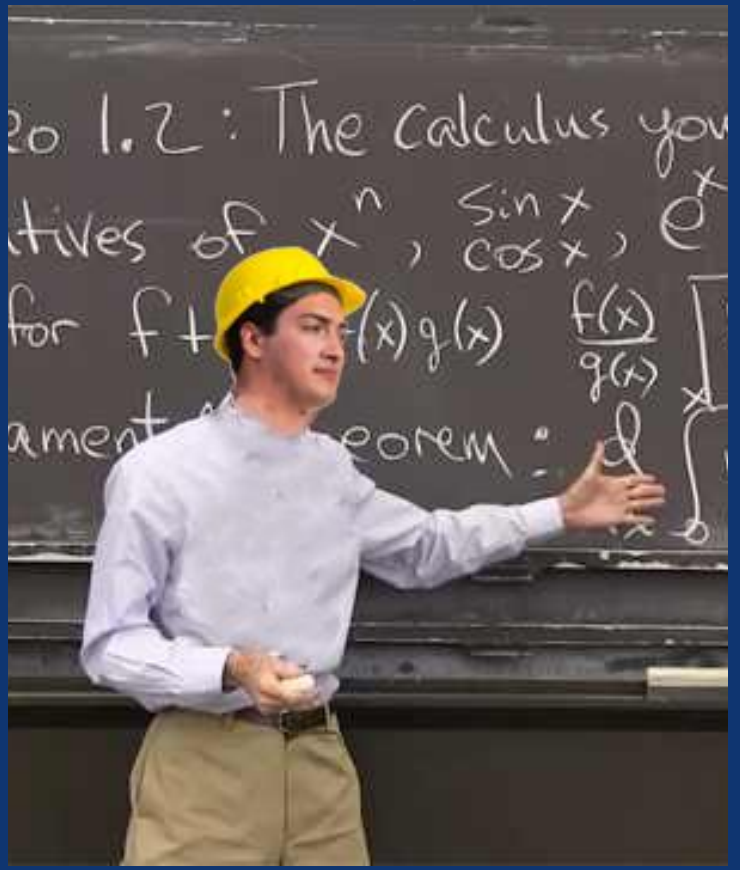
Si bien el aplicar un momentum me puede ayudar a alcanzar valores mínimos más rápido, una vez que llegamos a estos, observamos que los resultados pueden volverse erráticos y producir que el valor mínimo de error obtenido, empeore a lo largo de muchas iteraciones. Ya que al descender más rápido a valores pequeños acumula momentum de iteraciones anteriores, que probablemente descargue en alguna dirección no tan acertada o de forma exagerada.

CANTIDAD DE NEURONAS

En cuanto a la elección de las capas ocultas de una red, no encontramos un patrón para decidir cuántas neuronas usar en cada capa, o cuantas capas usar.
A mayor cantidad de neuronas, mayor procesamiento y mayor capacidad de identificar patrones complejos, pero no pudimos cuantificar esta variación.

CONJUNTOS REPRESENTATIVOS

Durante la validación cruzada del ejercicio 2, en la identificación de paridad, pudimos observar que había conjuntos que resultaban ser más representativos que otros, puesto que entrenar a la red con estos, permitía una mejor generalización sobre aquellos del set de test. También, es interesante ver que tanto para bloques de entrenamiento grandes como chicos, para training y testing pudimos obtener buenos resultados.



¡GRACIAS TOTALES!

Integrantes:

- Baiges, Matías Sebastián - 59076
- Bilevich, Andrés Leonardo - 59108
- Margossian, Gabriel Viken - 59130


slidesgo