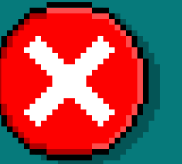
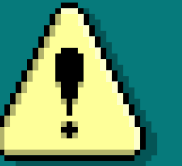
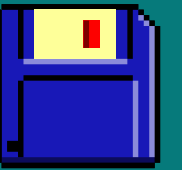


SIA



TP 3: Perceptrón Simple y Multicapa

GRUPO 1:

SOL VICTORIA ANSELMO

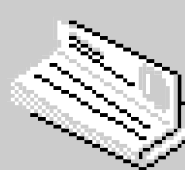
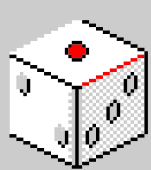
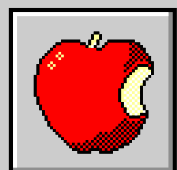
JULIÁN SASSO

AGUSTÍN MATTIUSI

CAMILA SIERRA PÉREZ

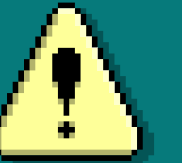
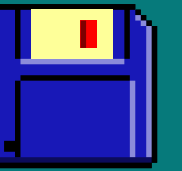
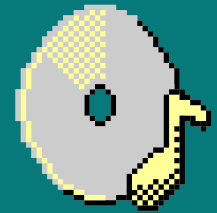
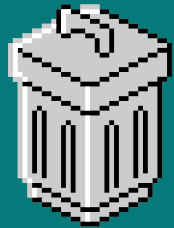
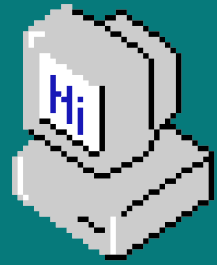
IAN JAMES ARNOTT

JUAN ADOLFO ROSAUER HERRMANN



11:11PM

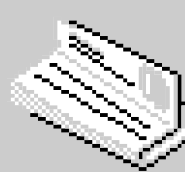
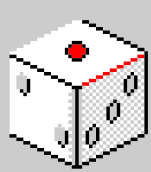
Ejercicio 1



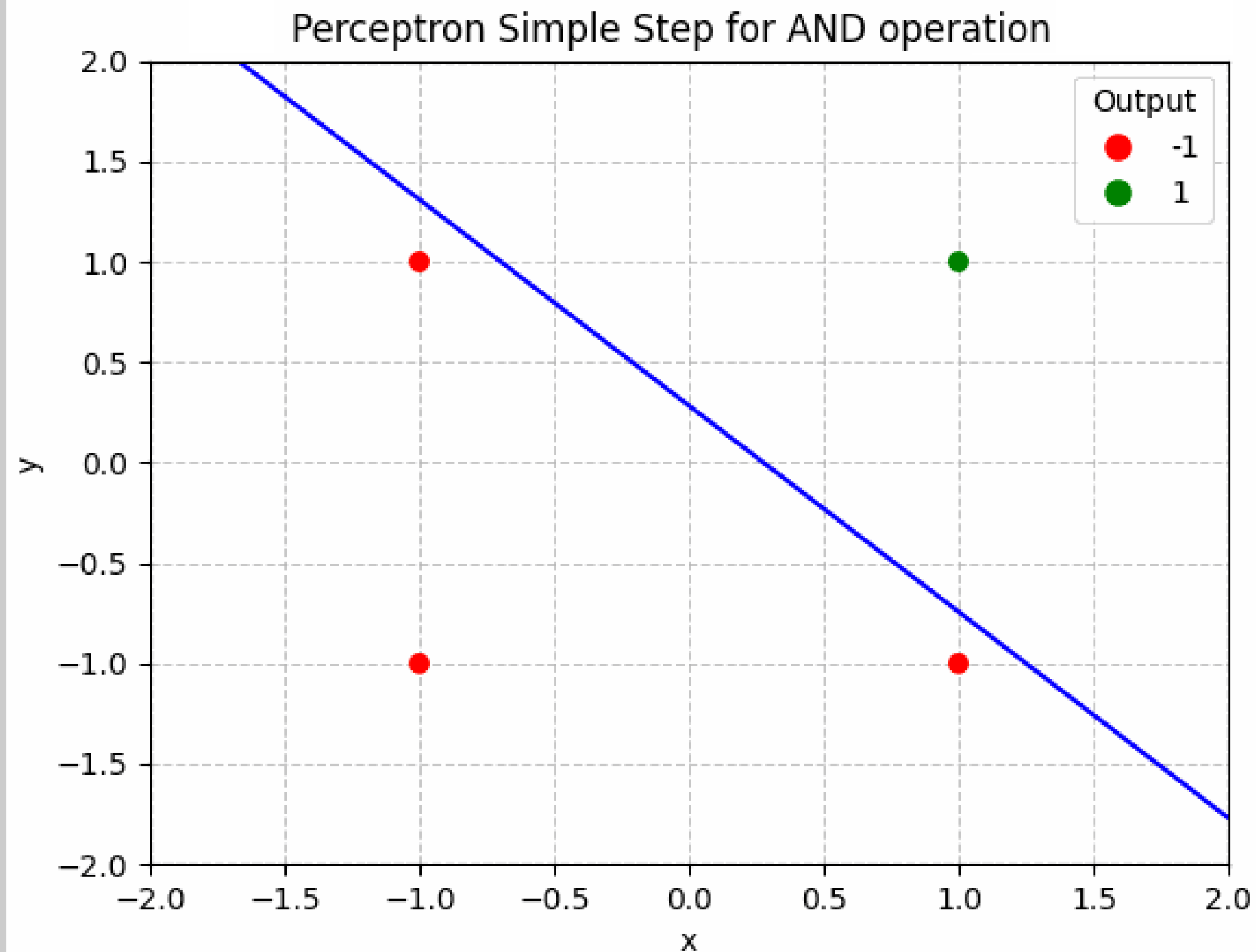
Perceptrón simple con función
de activación escalón

Funciones a implementar

- Operación lógica AND
- Operación lógica XOR

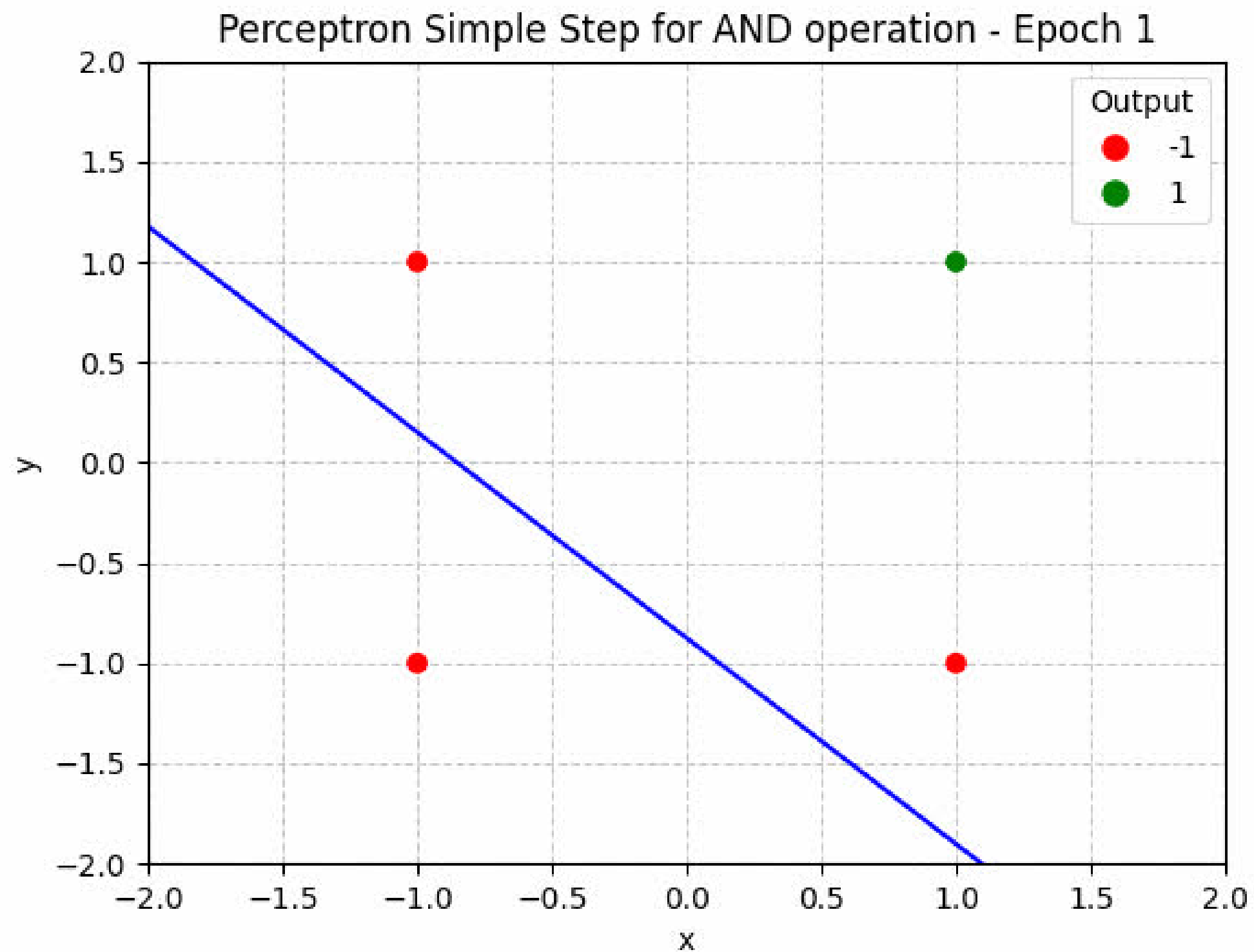


Operación AND



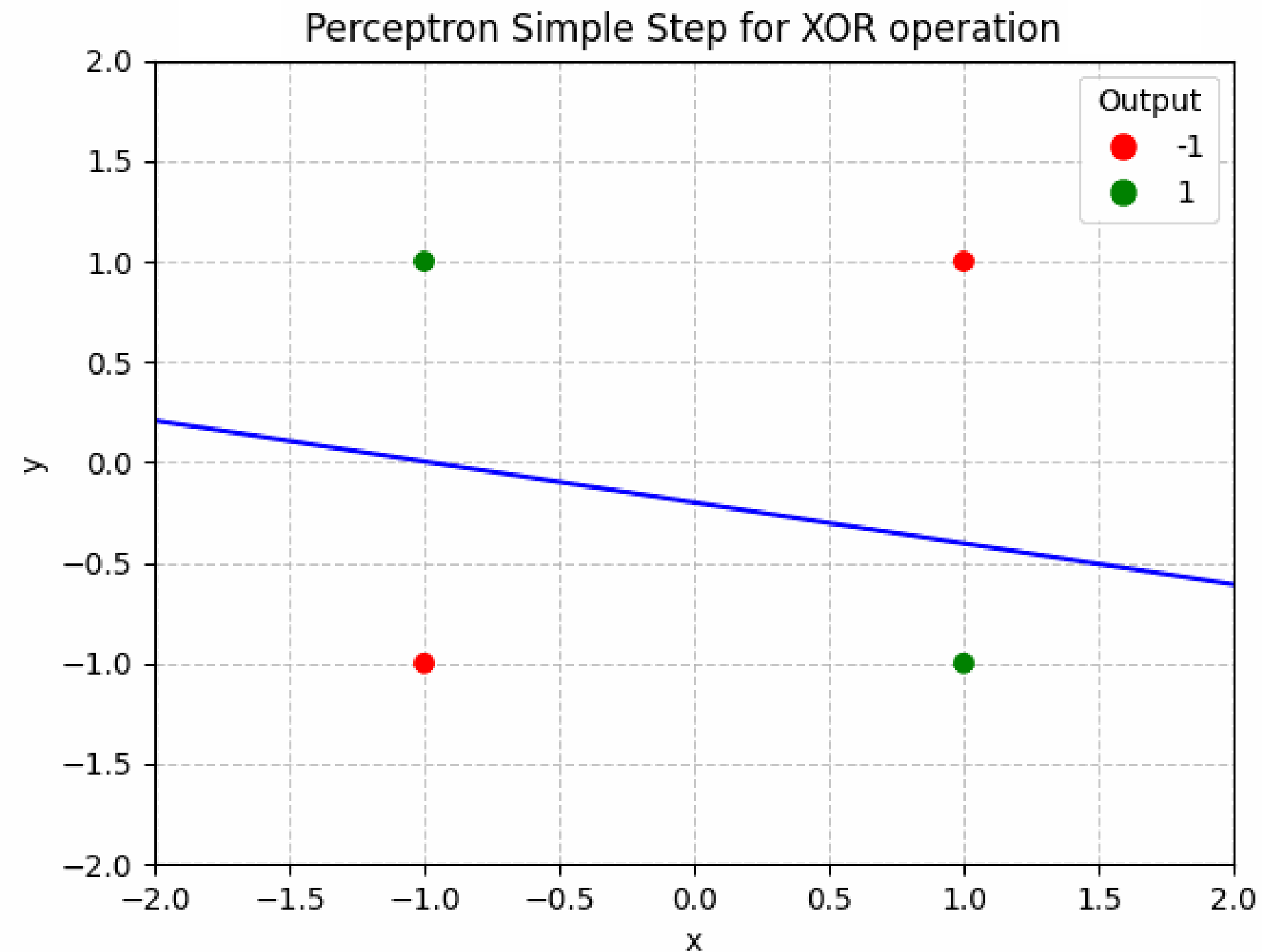
Cantidad de épocas: 0.1
Learning Rate: 500

Operación AND



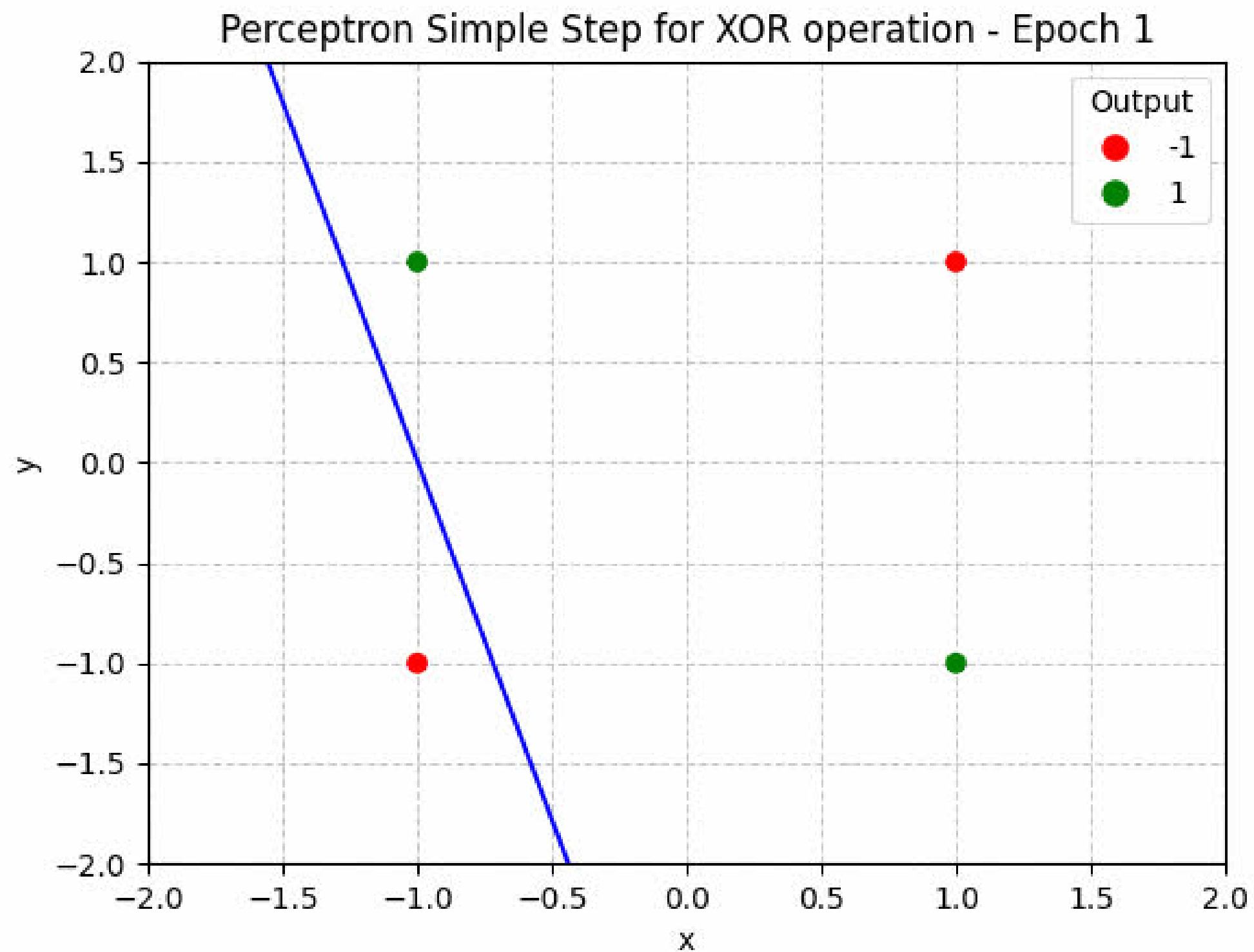
Cantidad de épocas: 0.1
Learning Rate: 500

Operación XOR



Cantidad de épocas: 0.1
Learning Rate: 500

Operación XOR



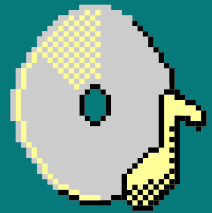
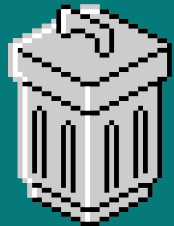
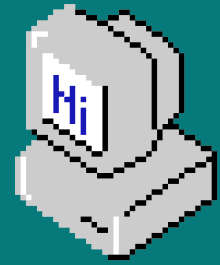
Cantidad de épocas: 0.1
Learning Rate: 500

Conclusiones

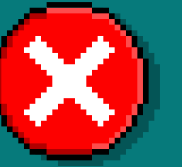
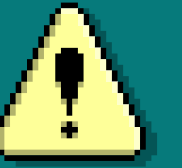
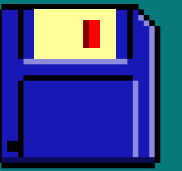
El perceptrón simple escalón:

- Encuentra un hiperplano que separa las dos clases en la operación “AND” ya que las coordenadas son **linealmente separables**.
- No encuentra un hiperplano que separe bien las dos clases en la operación “XOR” ya que los coordenadas **NO son linealmente separables**.

Ejercicio 2

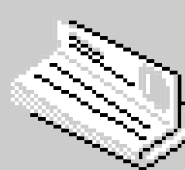
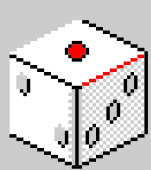
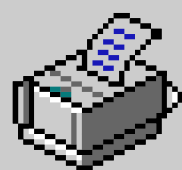
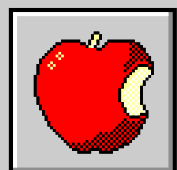


**Perceptrón simple lineal y
perceptrón simple no lineal**



Objetivos

- Evaluar capacidad de aprendizaje
- Evaluar capacidad de generalización

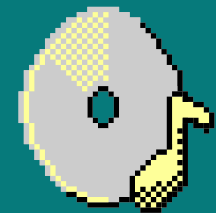
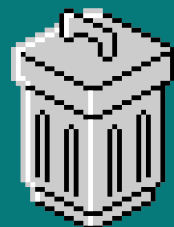
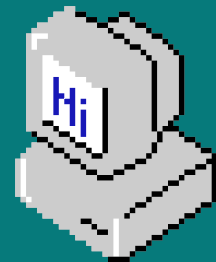


Consideraciones

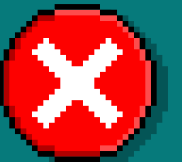
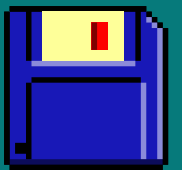
Para los perceptrones no lineales las imágenes de las funciones están acotadas:

- $(-1;1)$ para tanh
- $(0;1)$ para logística

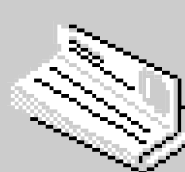
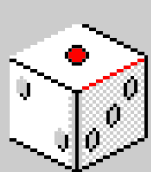
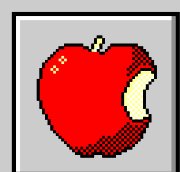
Como las salidas esperadas de nuestros perceptrones se encuentran en todos los reales, se normalizó el valor de θ con el metodo Min-Max Feature Scaling.



SIA

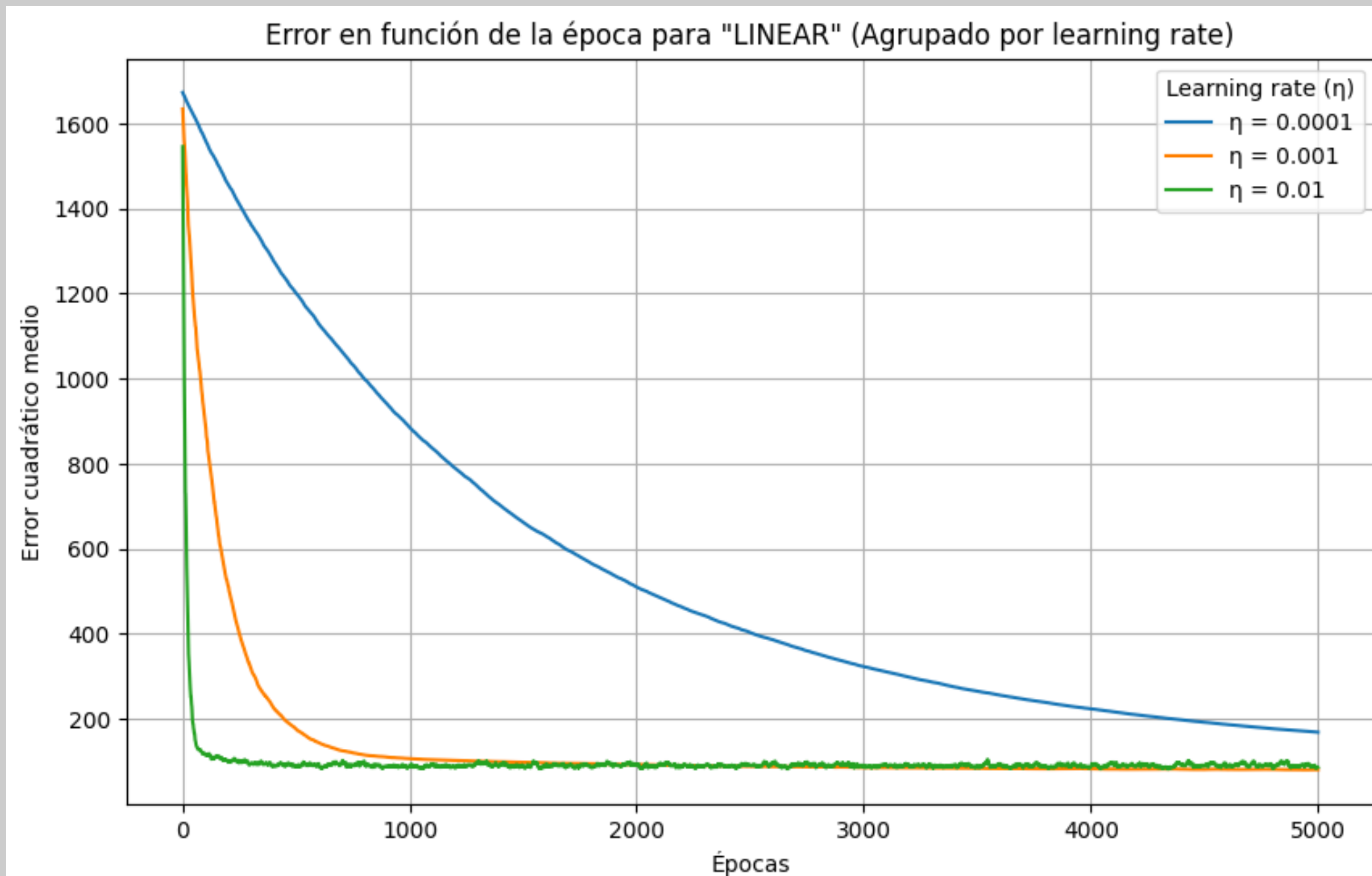


VEAMOS EL ERROR EN
FUNCIÓN DE LA ÉPOCA
AGRUPANDO POR
LEARNING RATE



11:11PM

Perceptrón Lineal



Training Percentage: 0.8

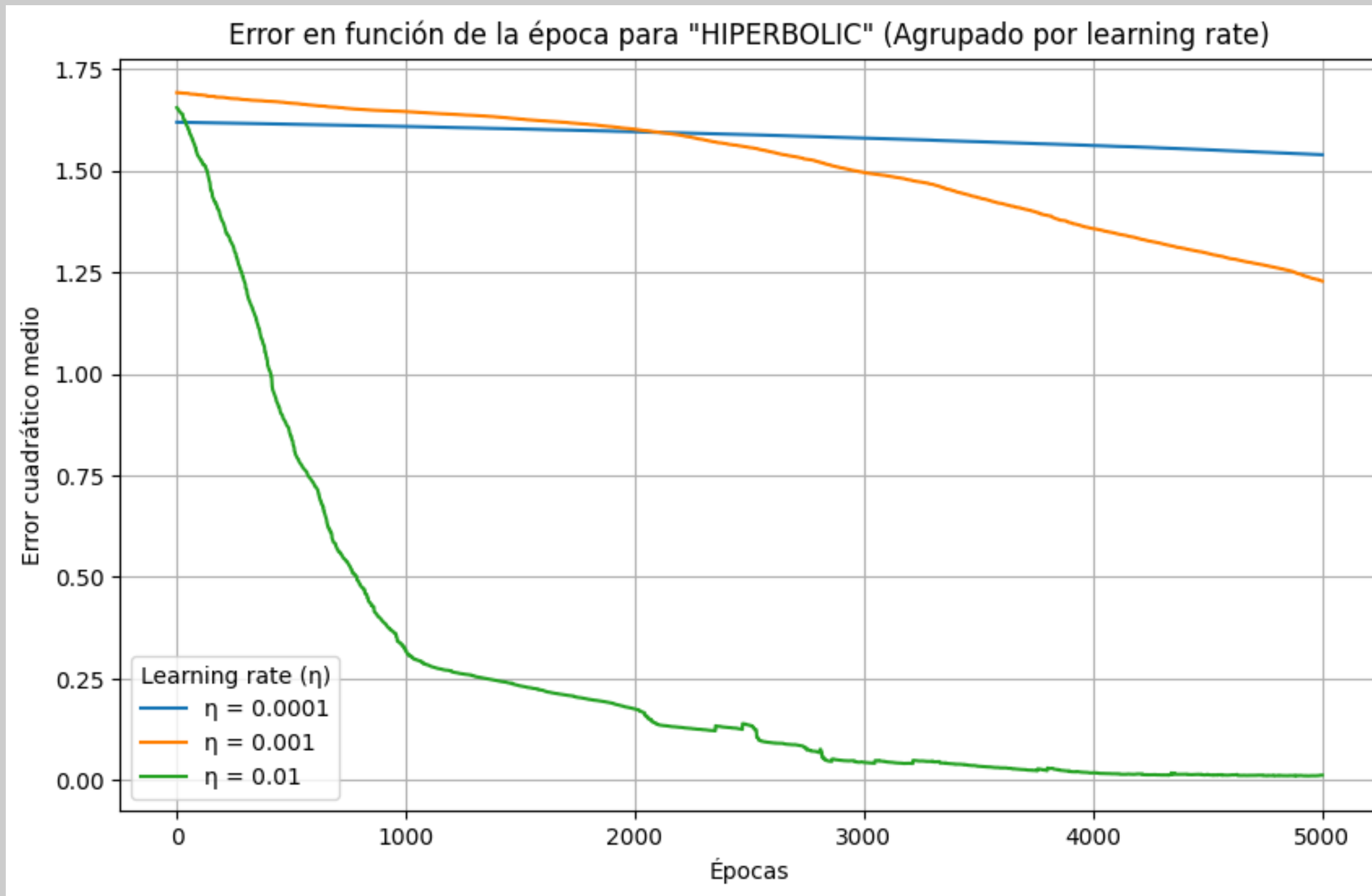
Epochs: 5000

Bias: 1

Beta: 1

Epsilon: 0.01

Perceptrón No Lineal



HYPERBOLIC

Training percentage: 0.8

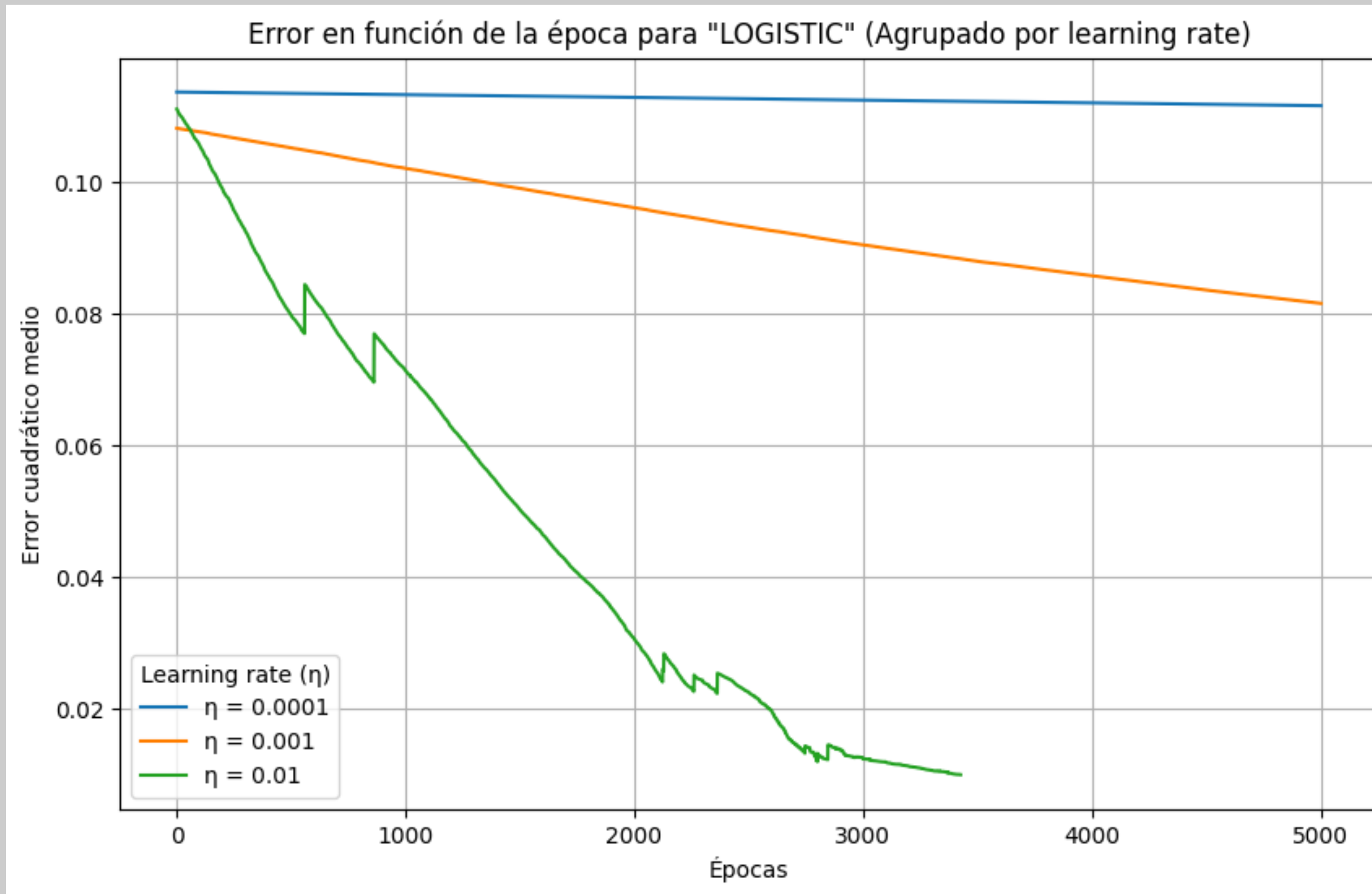
Epochs: 5000

Bias: 1

Beta: 1

Epsilon: 0.01

Perceptrón No Lineal



LOGISTIC

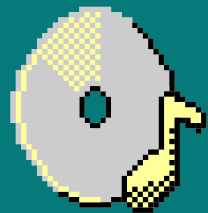
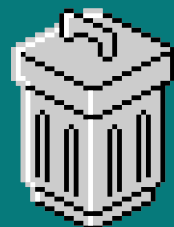
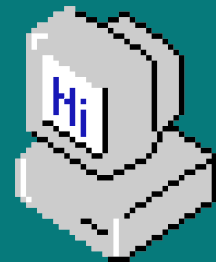
Training percentage: 0.8

Epochs: 5000

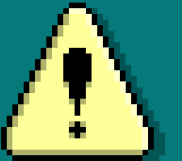
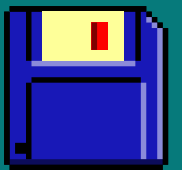
Bias: 1

Beta: 1

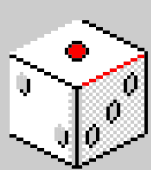
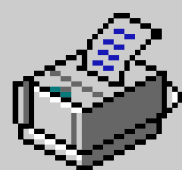
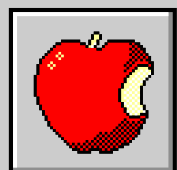
Epsilon: 0.01



SIA

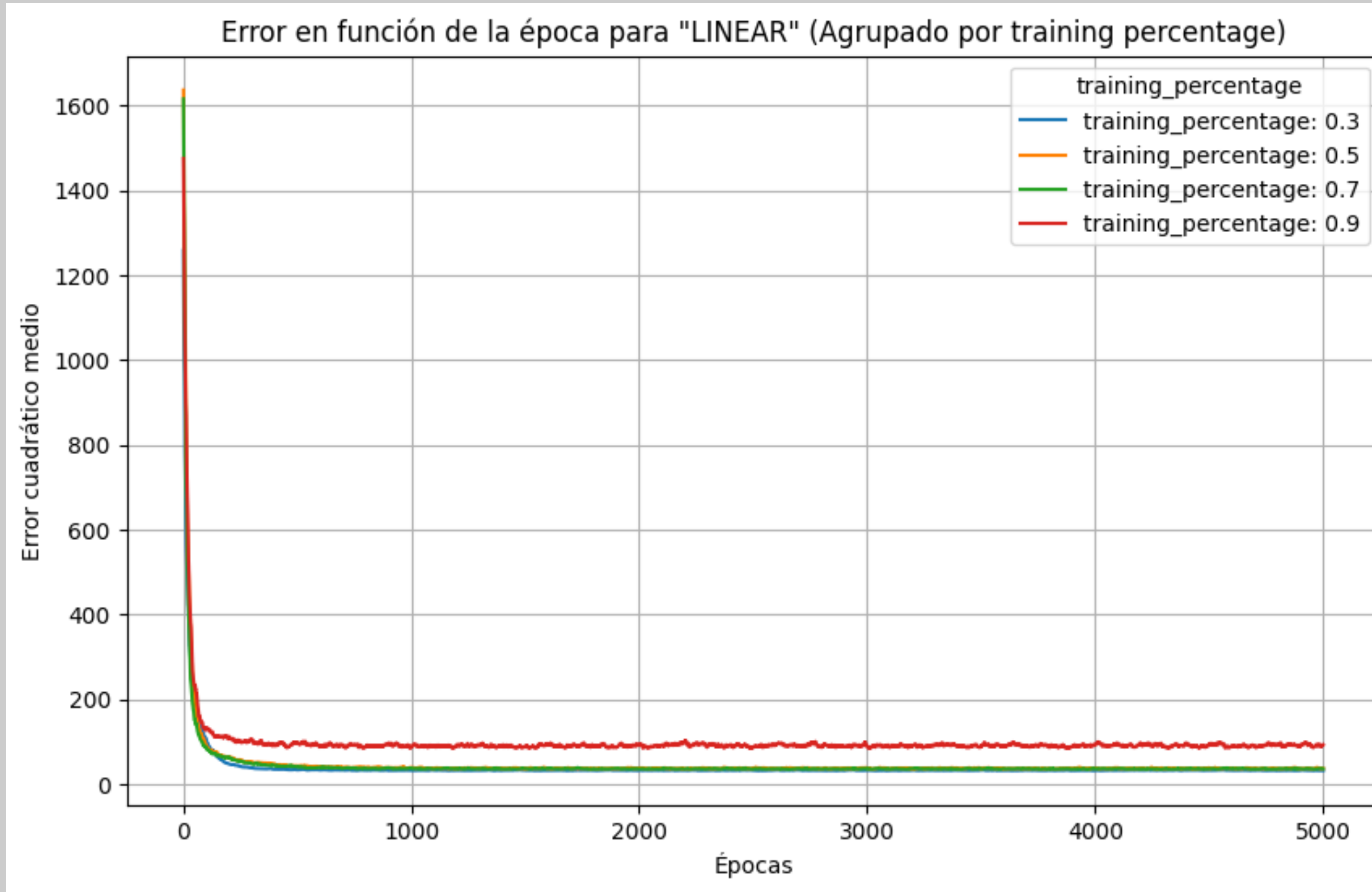


VEAMOS EL ERROR EN
FUNCIÓN DE LA ÉPOCA
AGRUPANDO POR
TRAINING PERCENTAGE



11:11PM

Perceptrón Lineal



Learning Rate: 0.01

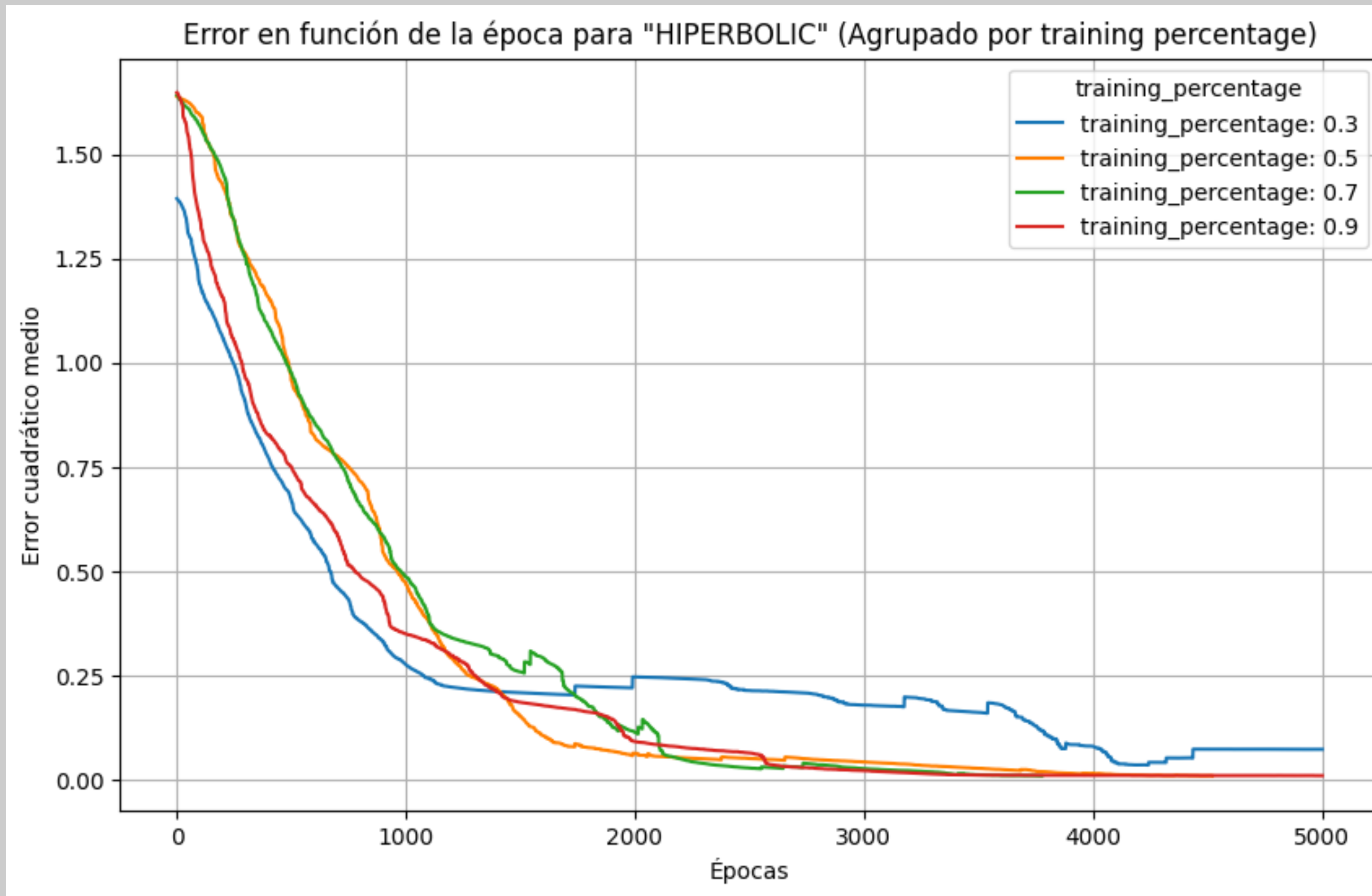
Epochs: 5000

Bias: 1

Beta: 1

Epsilon: 0.01

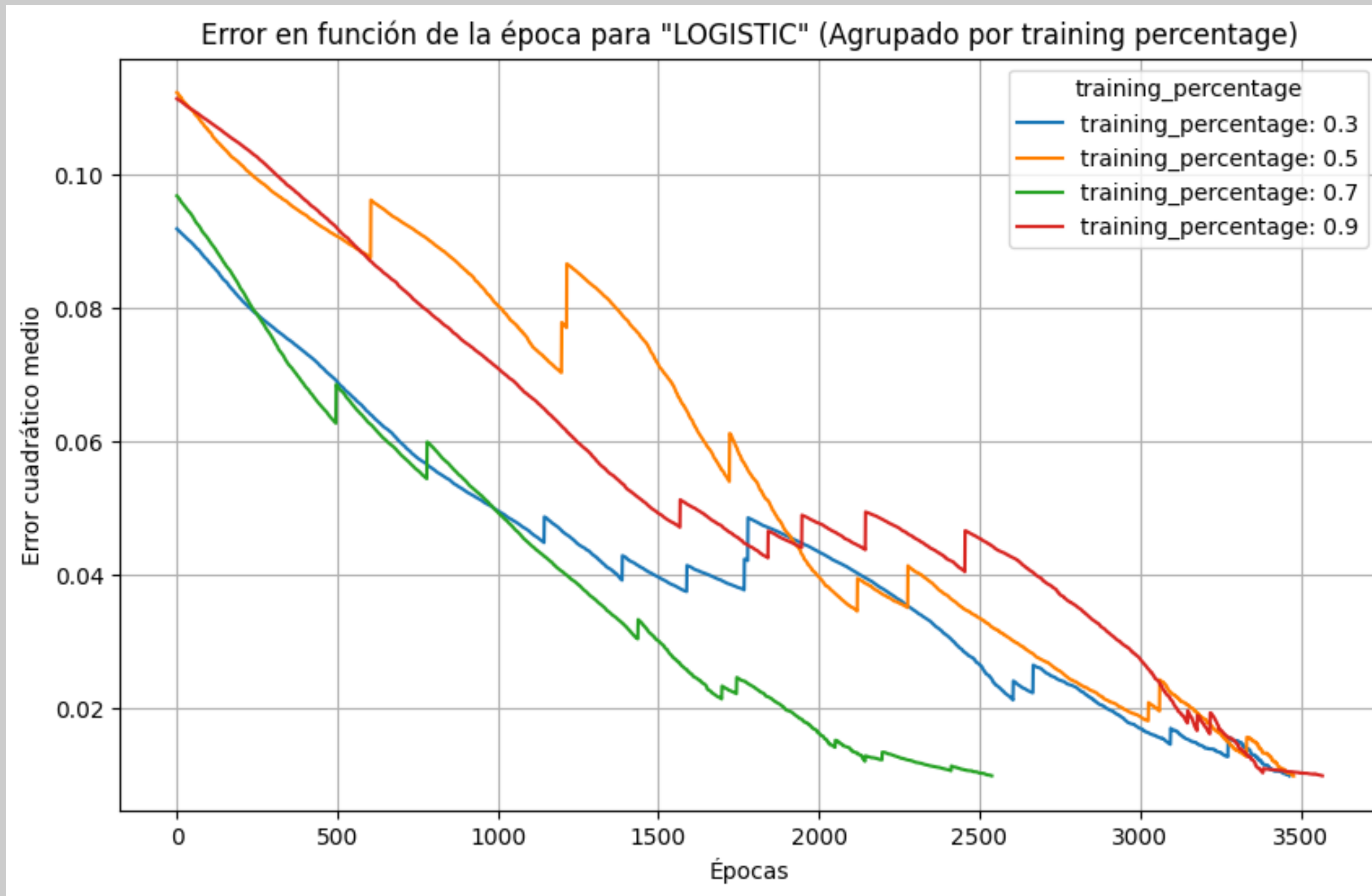
Perceptrón No Lineal



HYPERBOLIC

Learning Rate: 0.01
Epochs: 5000
Bias: 1
Beta: 1
Epsilon: 0.01

Perceptrón No Lineal



LOGISTIC

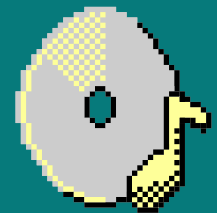
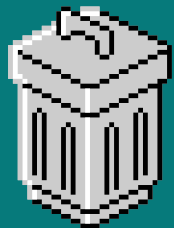
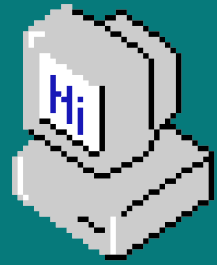
Learning Rate: 0.01

Epochs: 5000

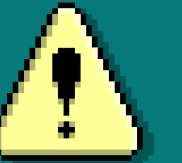
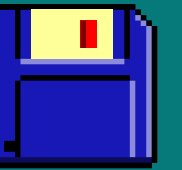
Bias: 1

Beta: 1

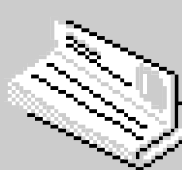
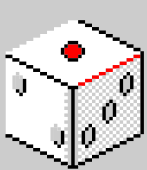
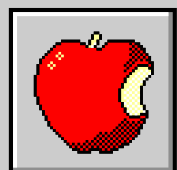
Epsilon: 0.01



SIA

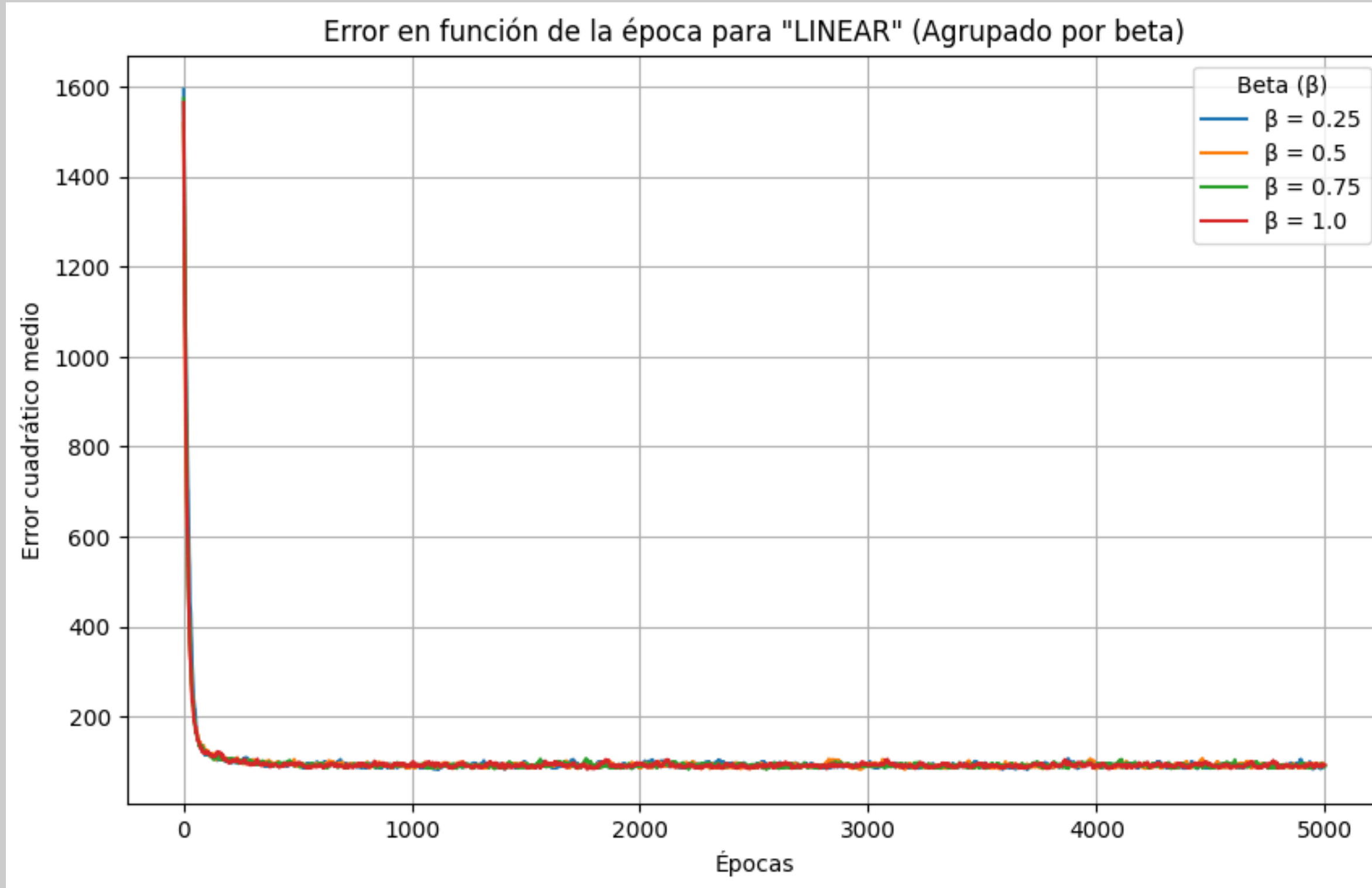


VEAMOS EL ERROR EN
FUNCIÓN DE LA ÉPOCA
AGRUPANDO POR BETA



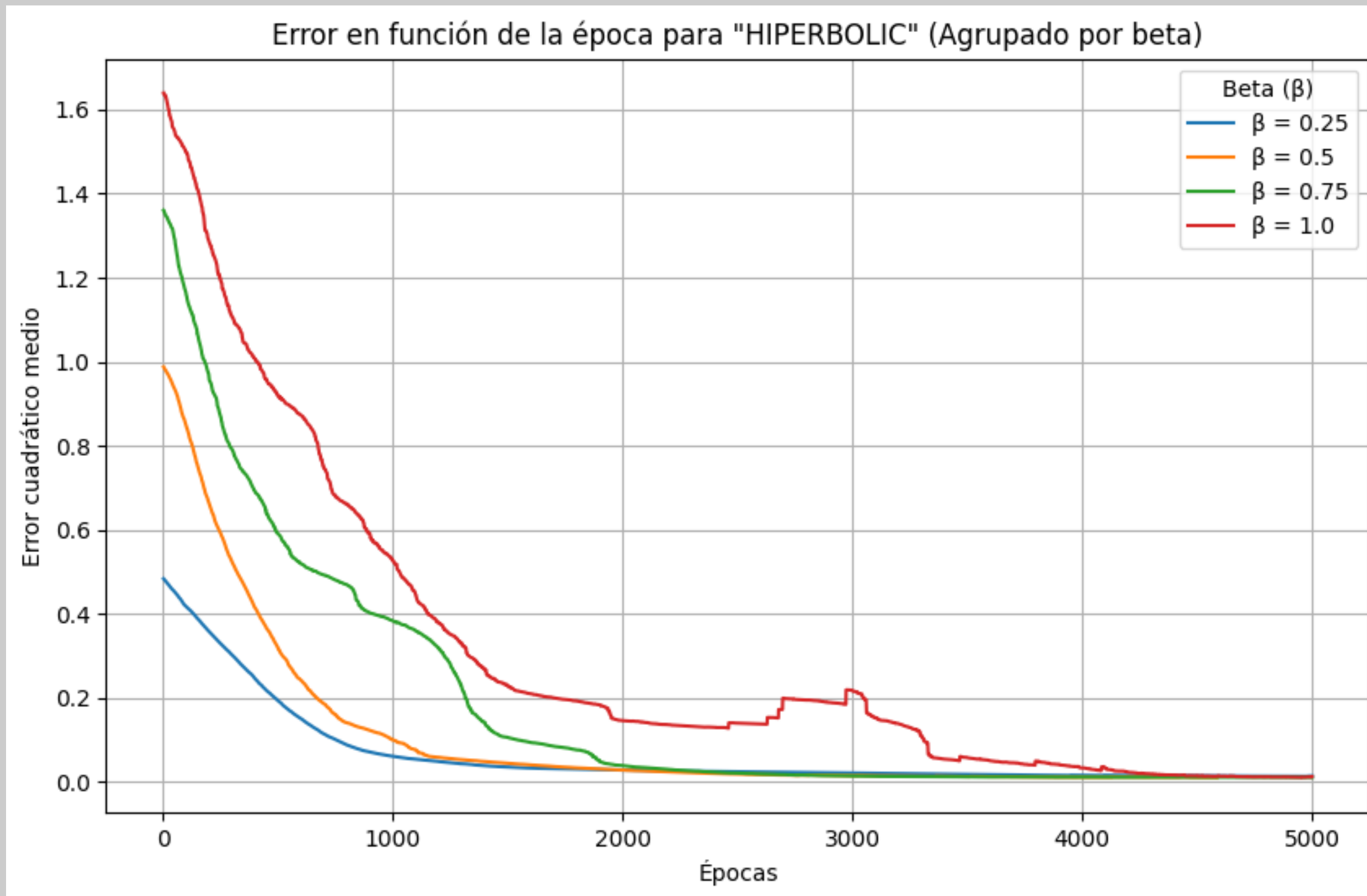
11:11PM

Perceptrón Lineal



Learning Rate: 0.01
Training percentage: 0.8
Epochs: 5000
Bias: 1
Epsilon: 0.01

Perceptrón No Lineal



HYPERBOLIC

Learning Rate: 0.01

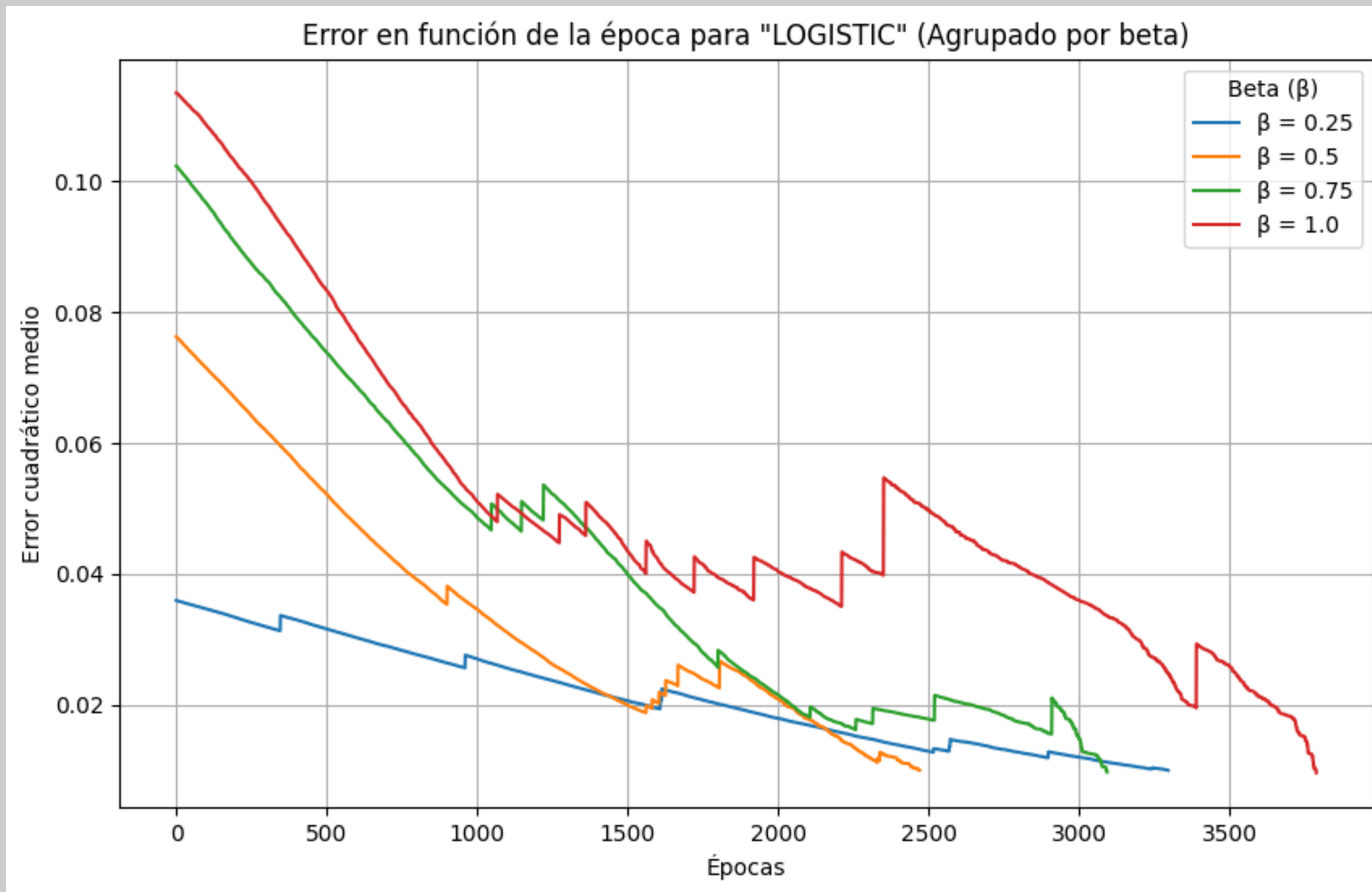
Training percentage: 0.8

Epochs: 5000

Bias: 1

Epsilon: 0.01

Perceptrón No Lineal



LOGISTIC

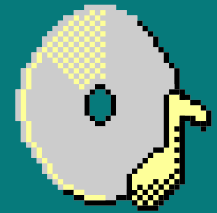
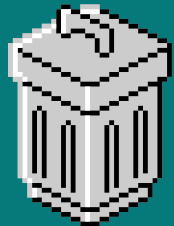
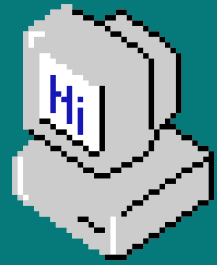
Learning Rate: 0.01

Training percentage: 0.8

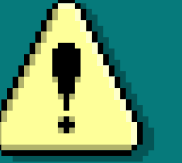
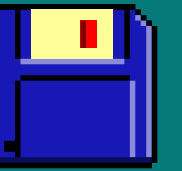
Epochs: 5000

Bias: 1

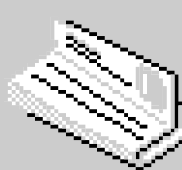
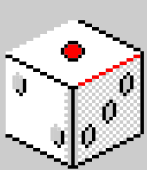
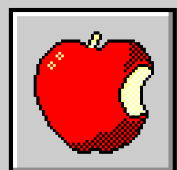
Epsilon: 0.01



SIA

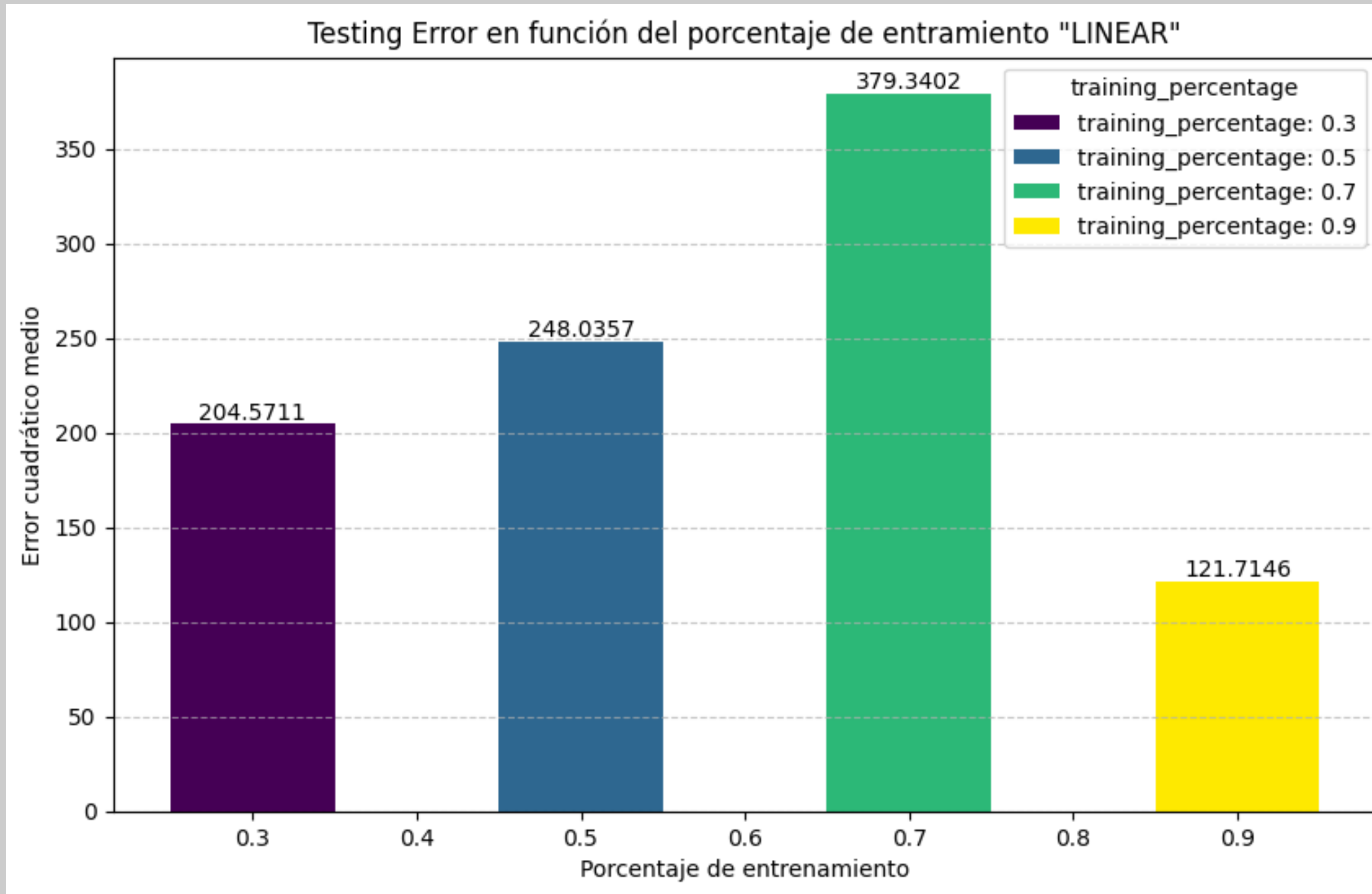


VEAMOS EL ERROR DE
TESTEO EN FUNCIÓN DE
LA ÉPOCA AGRUPANDO POR
TRAINING PERCENTAGE



11:11PM

Perceptrón Lineal



Learning Rate: 0.01

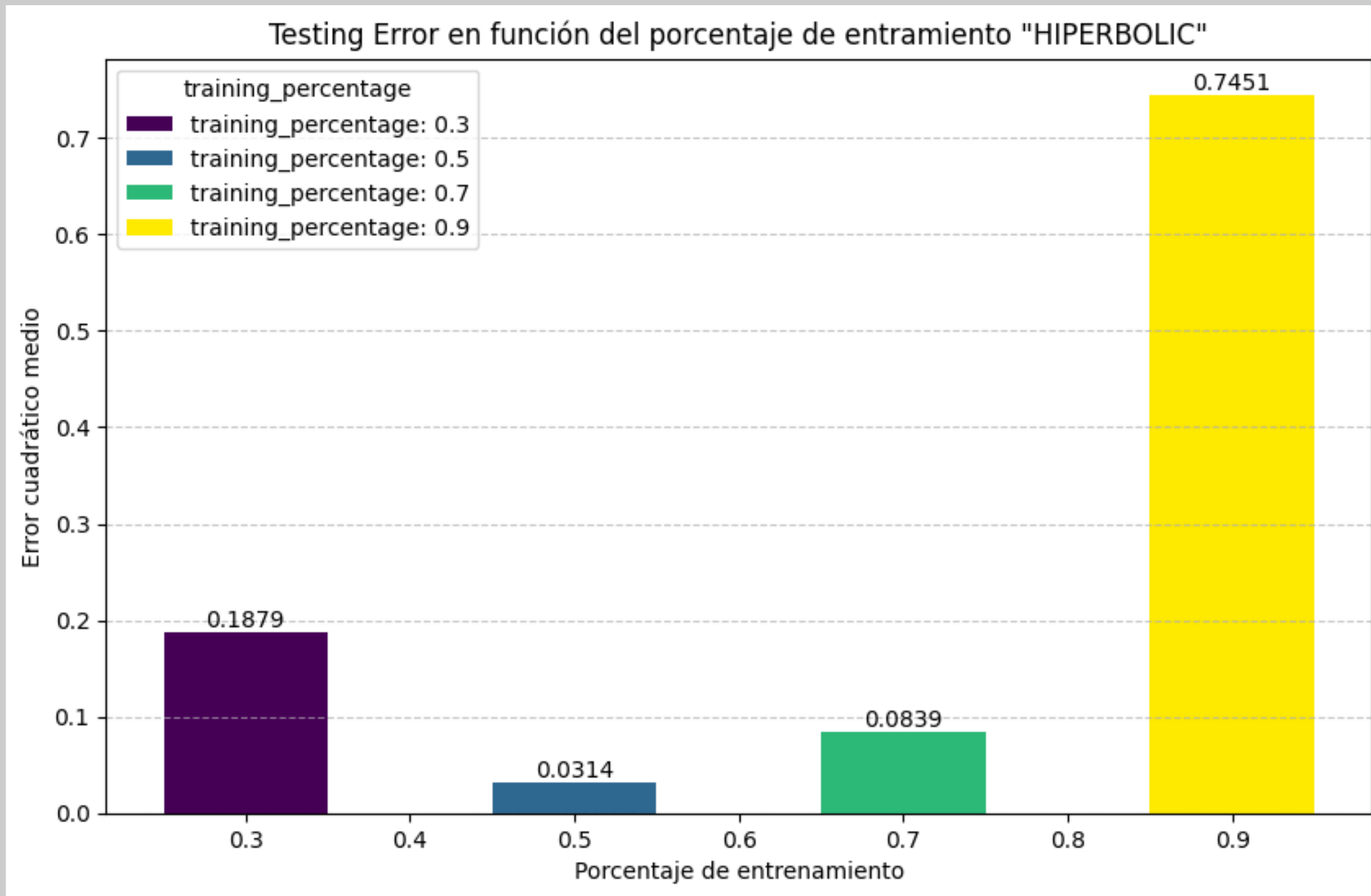
Epochs: 5000

Bias: 1

Beta: 1

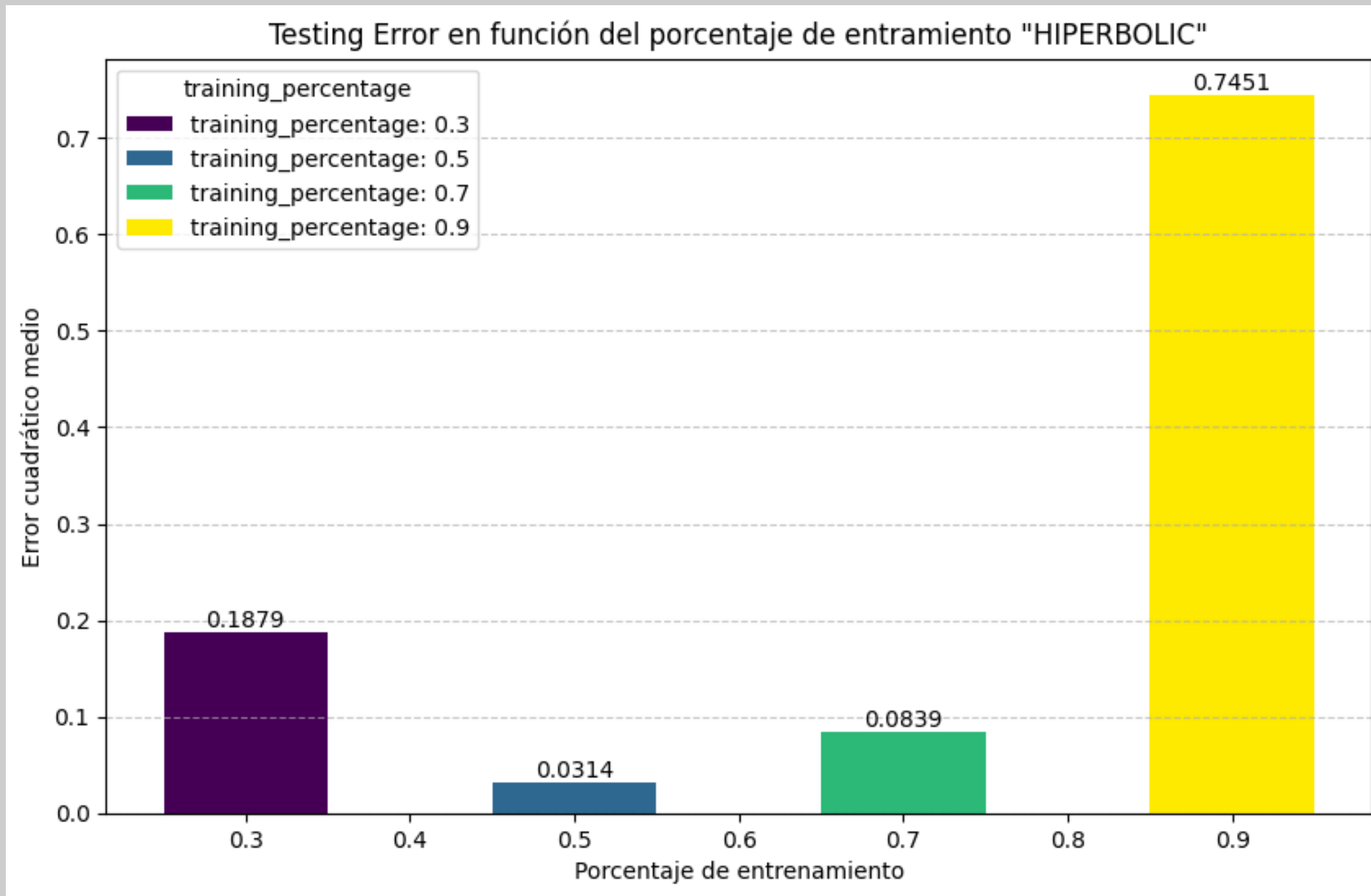
Epsilon: 0.01

Perceptrón No Lineal



HYPERBOLIC
Learning Rate: 0.01
Epochs: 5000
Bias: 1
Beta: 1
Epsilon: 0.01

Perceptrón No Lineal



LOGISTIC

Learning Rate: 0.01

Epochs: 5000

Bias: 1

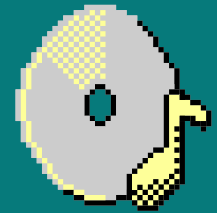
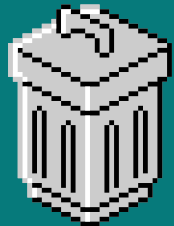
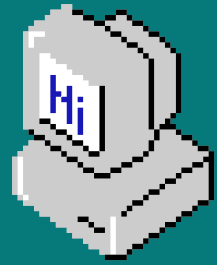
Beta: 1

Epsilon: 0.01

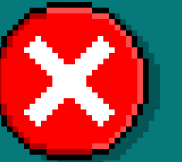
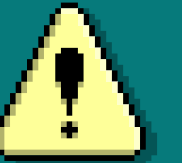
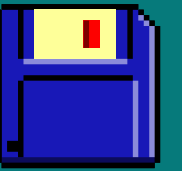
Conclusiones

- Para elegir el mejor conjunto de entrenamiento comparamos cada una de las distintas divisiones del conjunto de datos para elegir la mejor.
 - El mejor training percentage varía según el tipo de perceptron elegido.
- Para ver el efecto de la capacidad de generalización del perceptrón comparamos los errores del conjunto de testeo con los errores del conjunto de entrenamiento.
 - Con porcentajes muy bajos o muy altos se produce una pérdida de capacidad de generalización del perceptrón.

Ejercicio 3

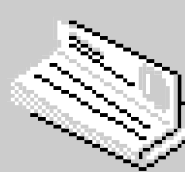
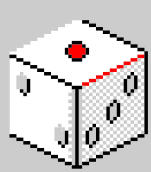


Perceptrón multicapa

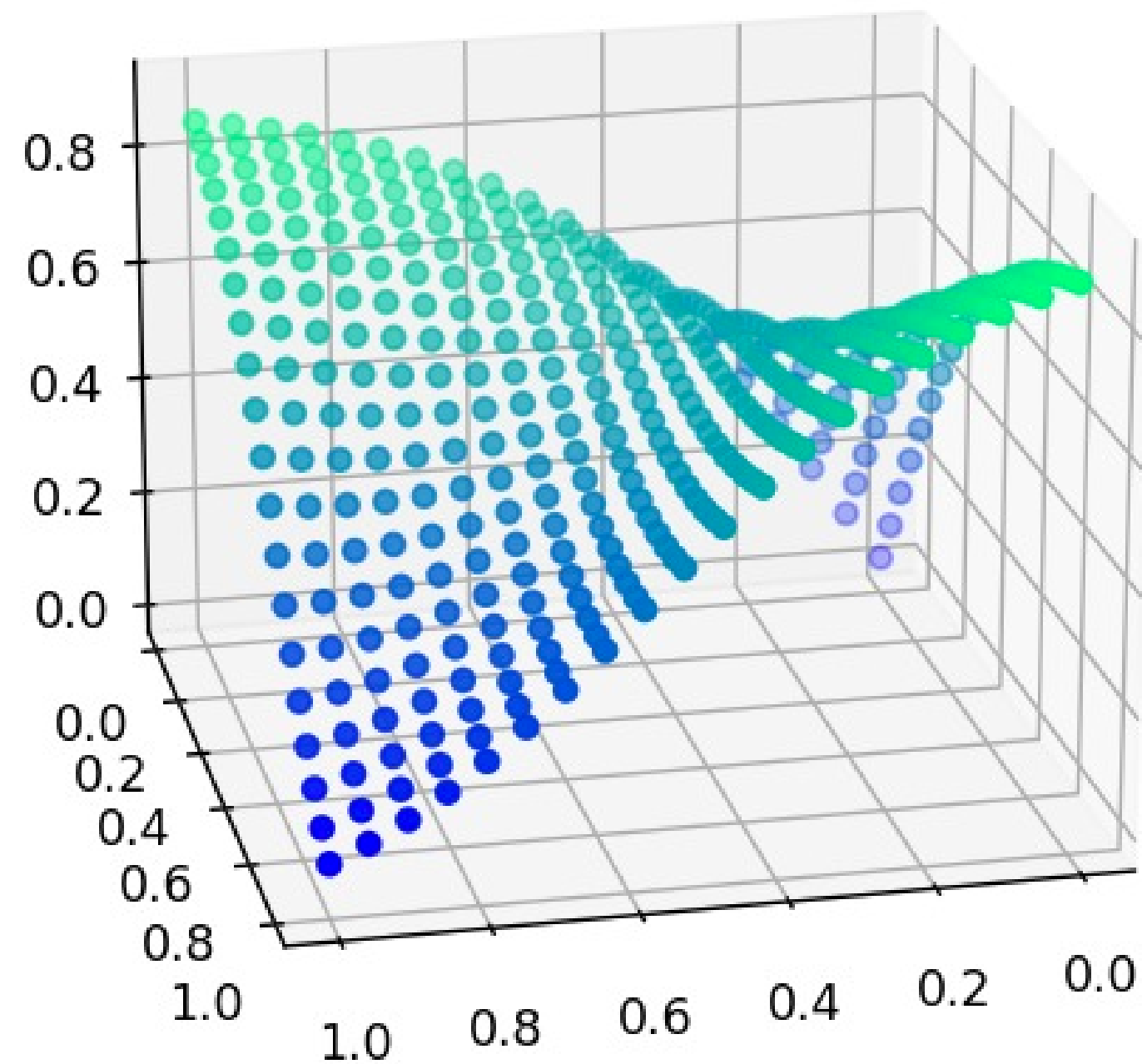


Funciones a implementar

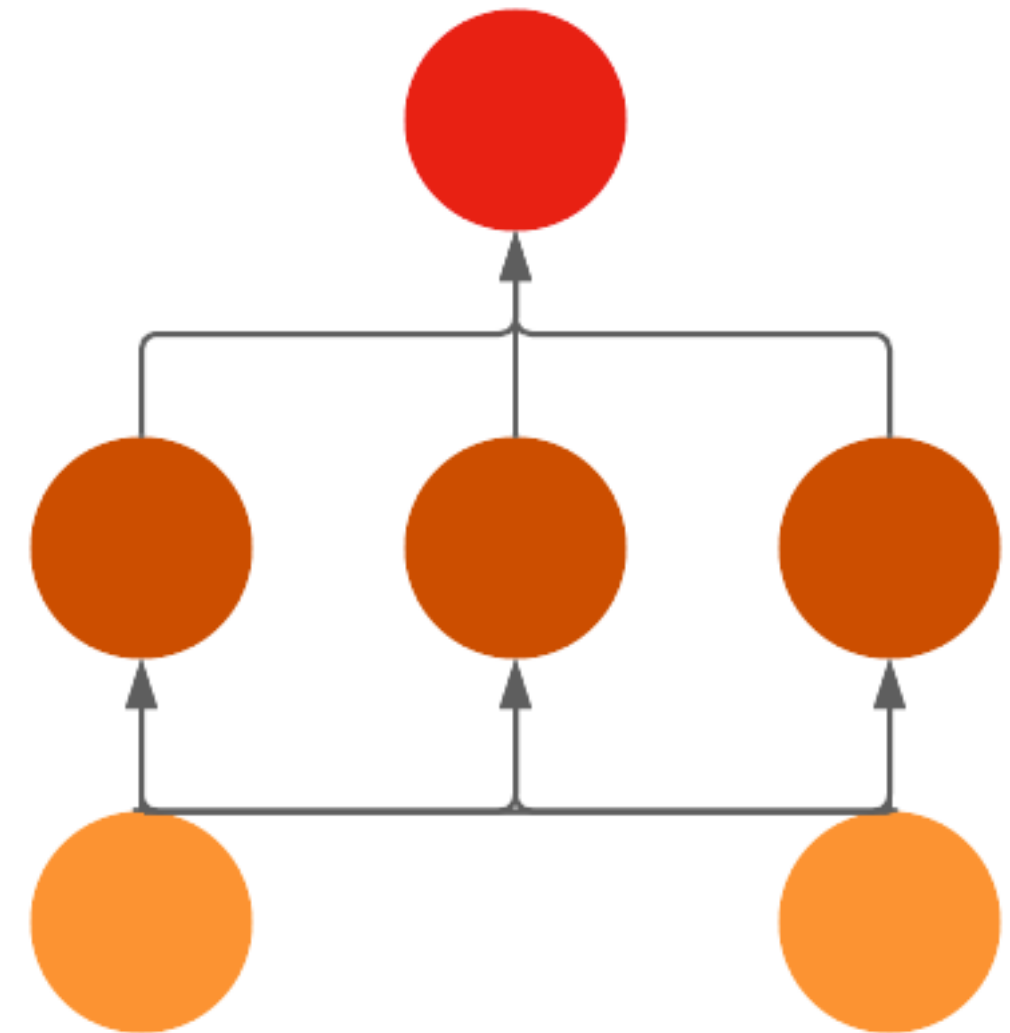
- Función lógica “O exclusivo”
- Discriminación de números pares
- Determinar si el dígito se corresponde con la entrada a la red



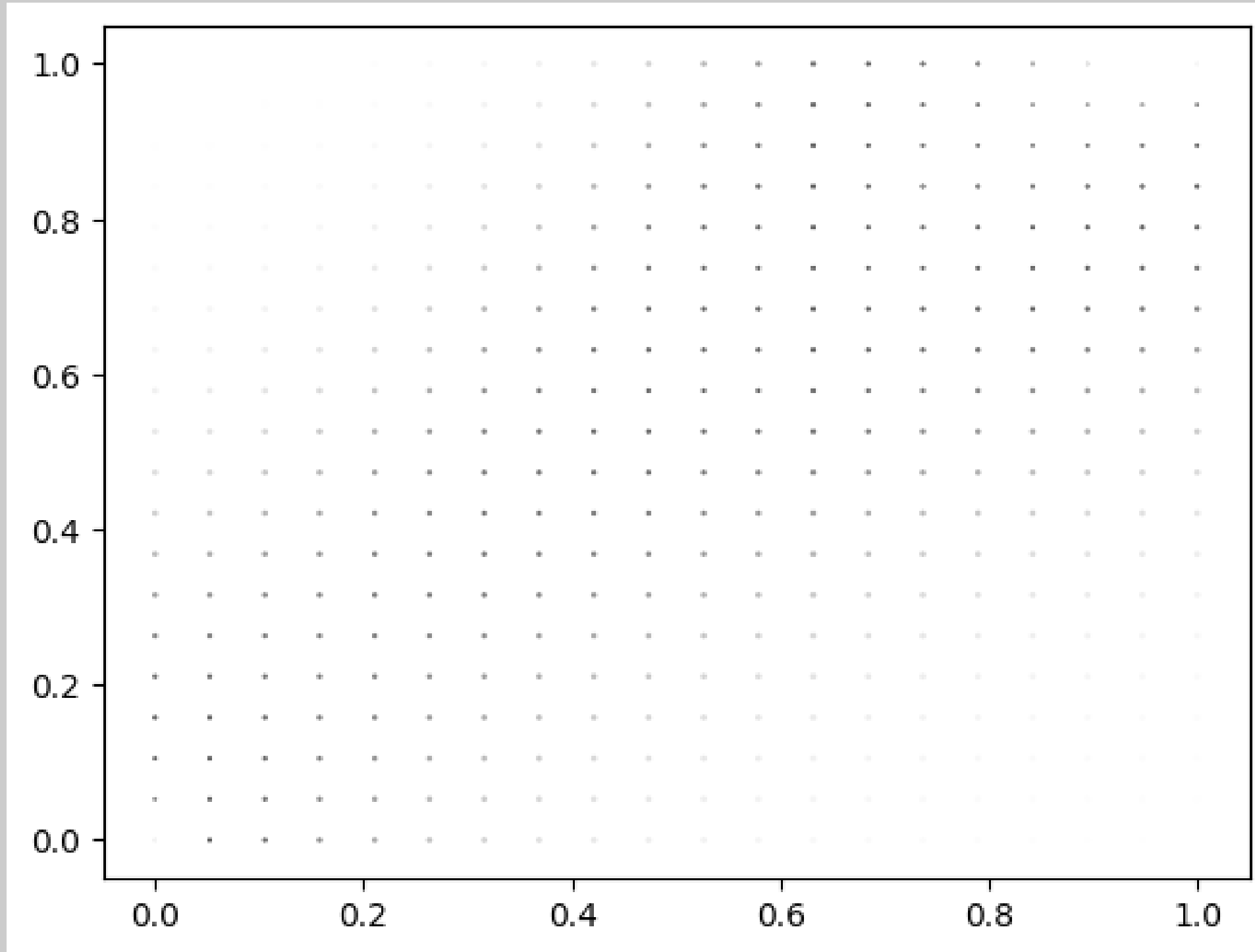
Perceptrón Multicapa



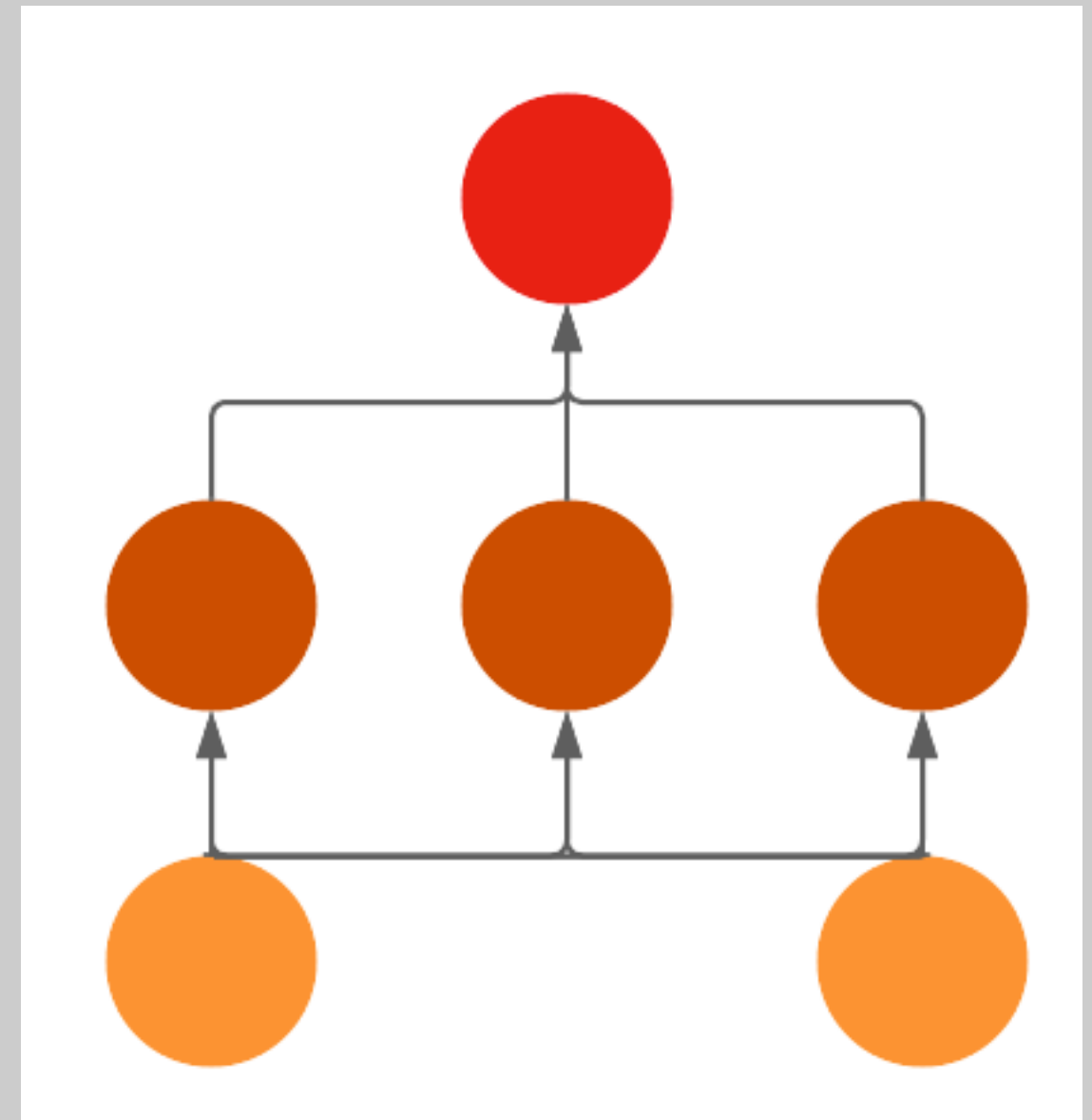
FUNCIÓN XOR
AHORA SE RESUELVE



Perceptrón Multicapa



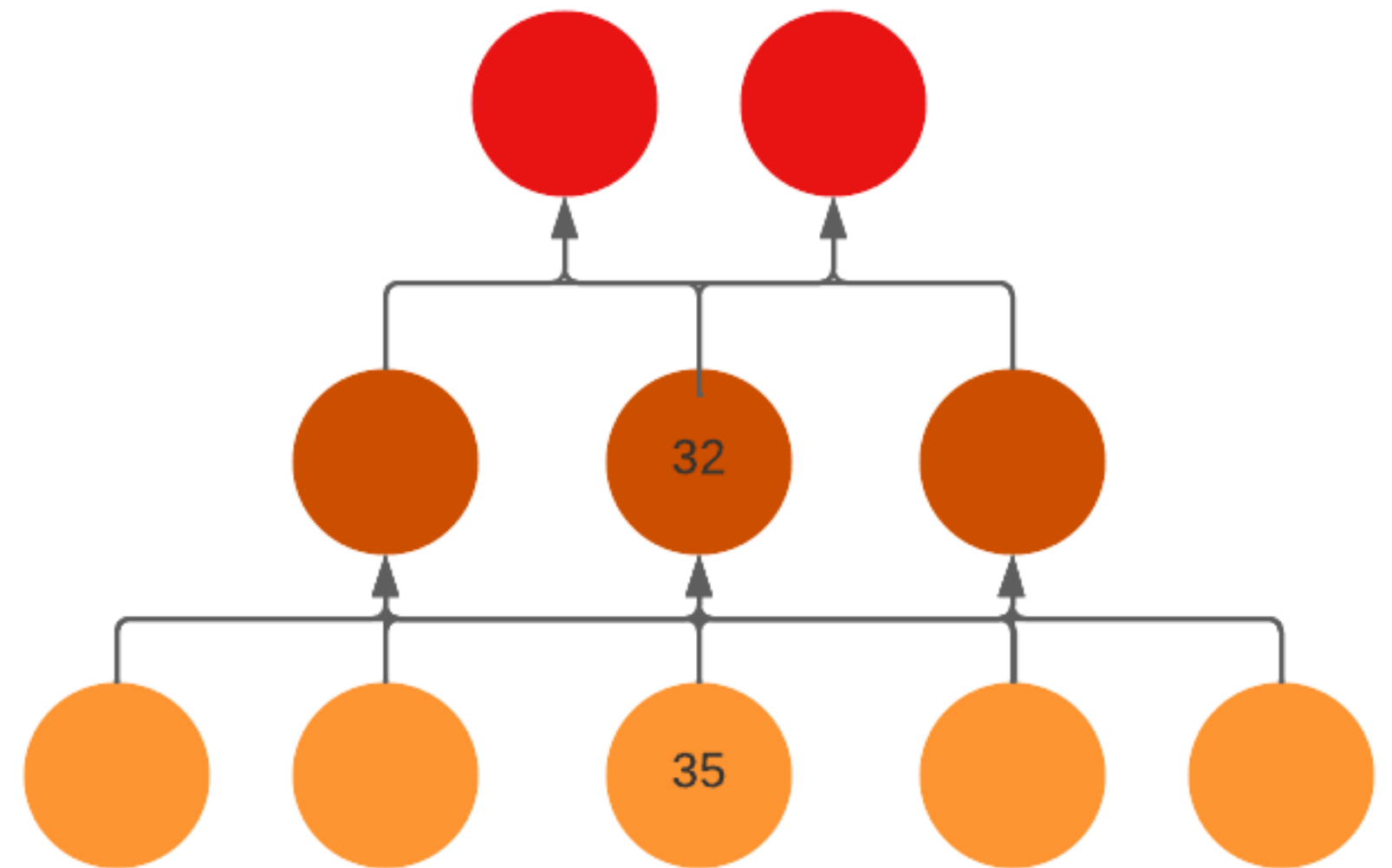
FUNCIÓN XOR
AHORA SE RESUELVE



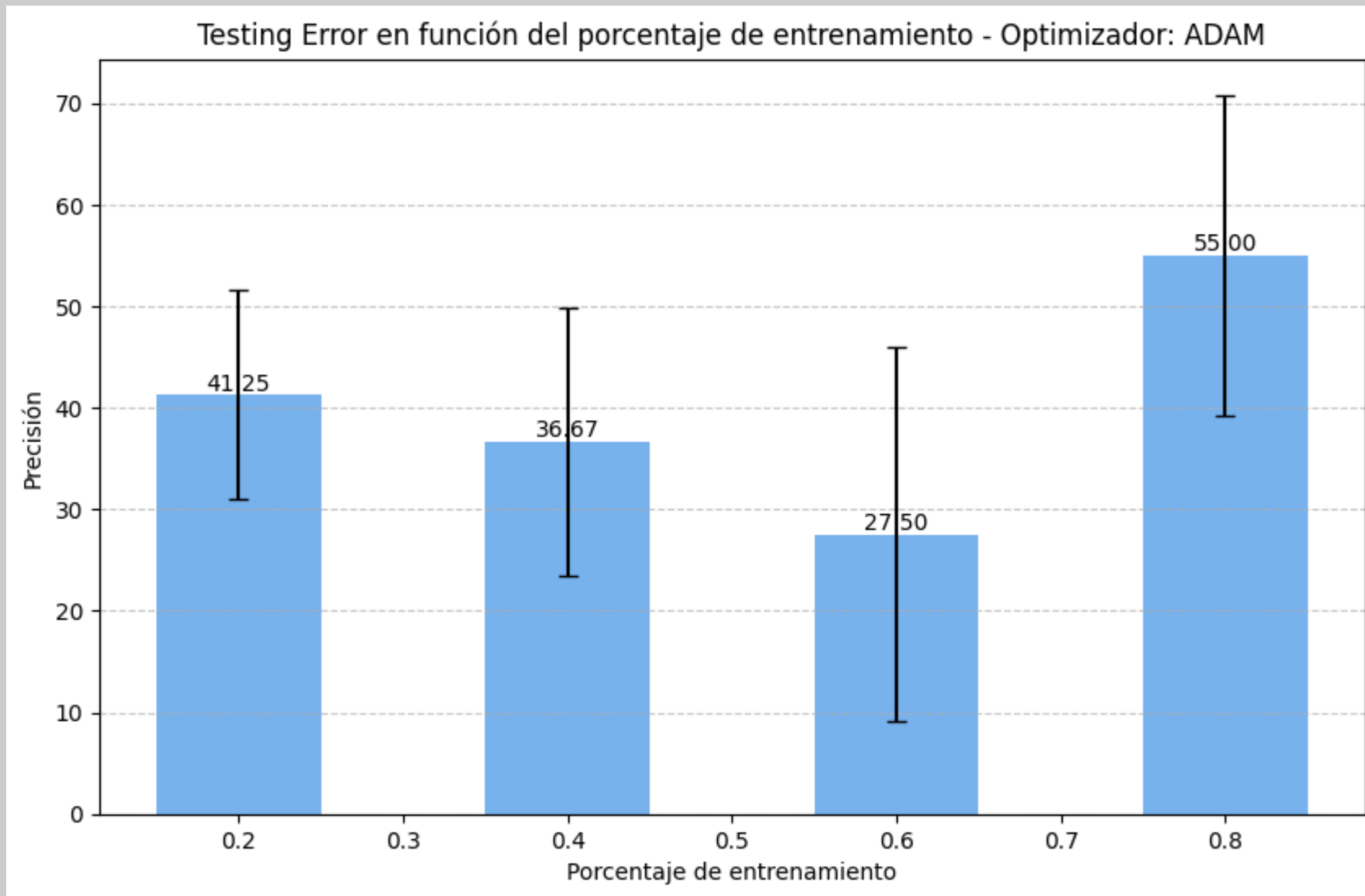
Perceptrón Multicapa

DISCRIMINACIÓN DE NÚMEROS PAR

Learning Rate: 0.01
Training Percentage: 0.5
Epochs: 5000
Bias: 1
Beta: 1
Epsilon: 0.5
Activation: Sigmoid



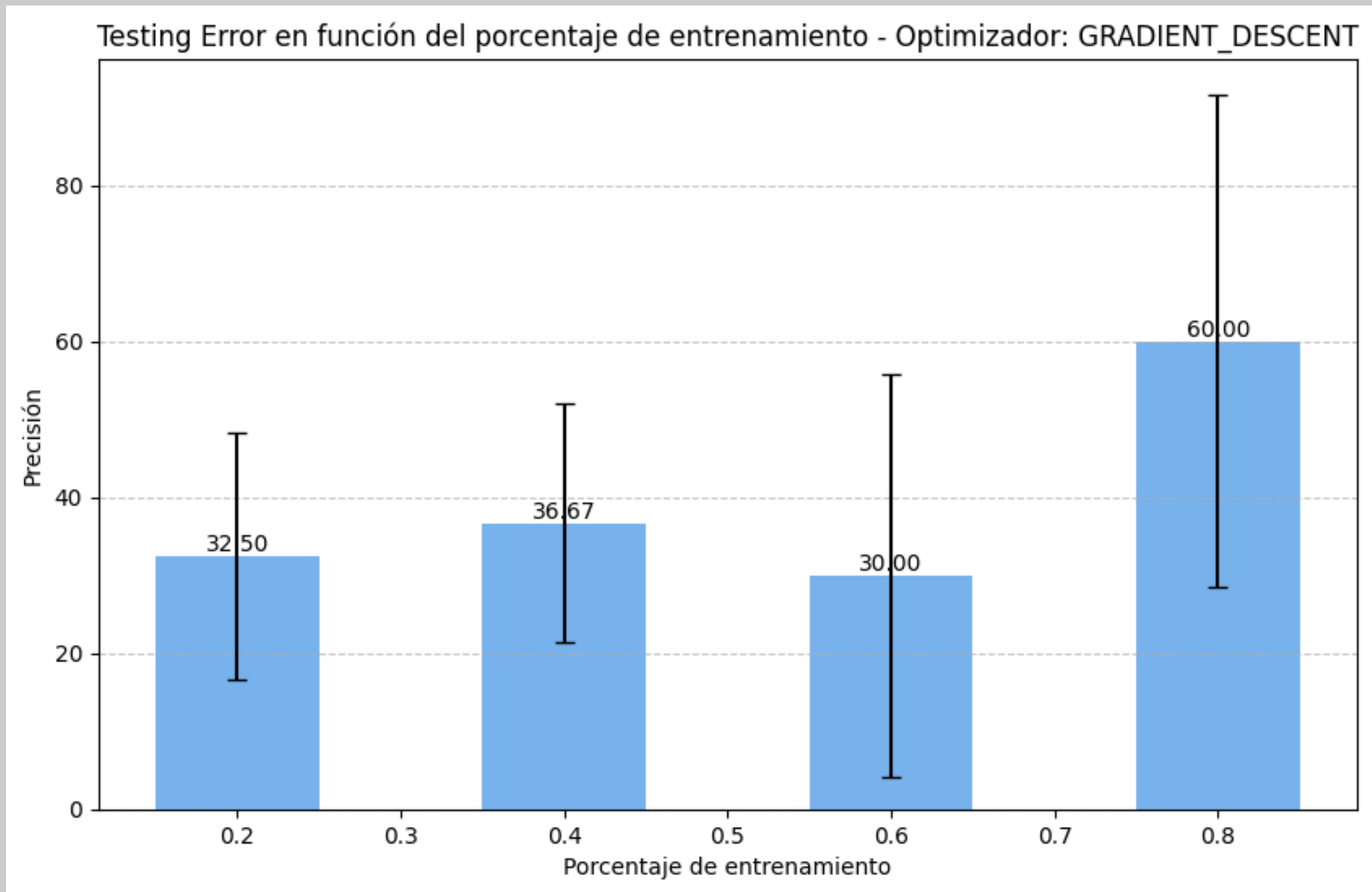
Perceptrón Multicapa



DISCRIMINACIÓN DE NÚMEROS PAR

Al dividir el conjunto de datos en entrenamiento y testeo, estamos privando a la red de que aprenda la paridad de todos los números. Por lo que le cuesta ser eficiente a la hora de predecir la paridad de los números no aprendidos.

Perceptrón Multicapa



DISCRIMINACIÓN DE NÚMEROS PAR

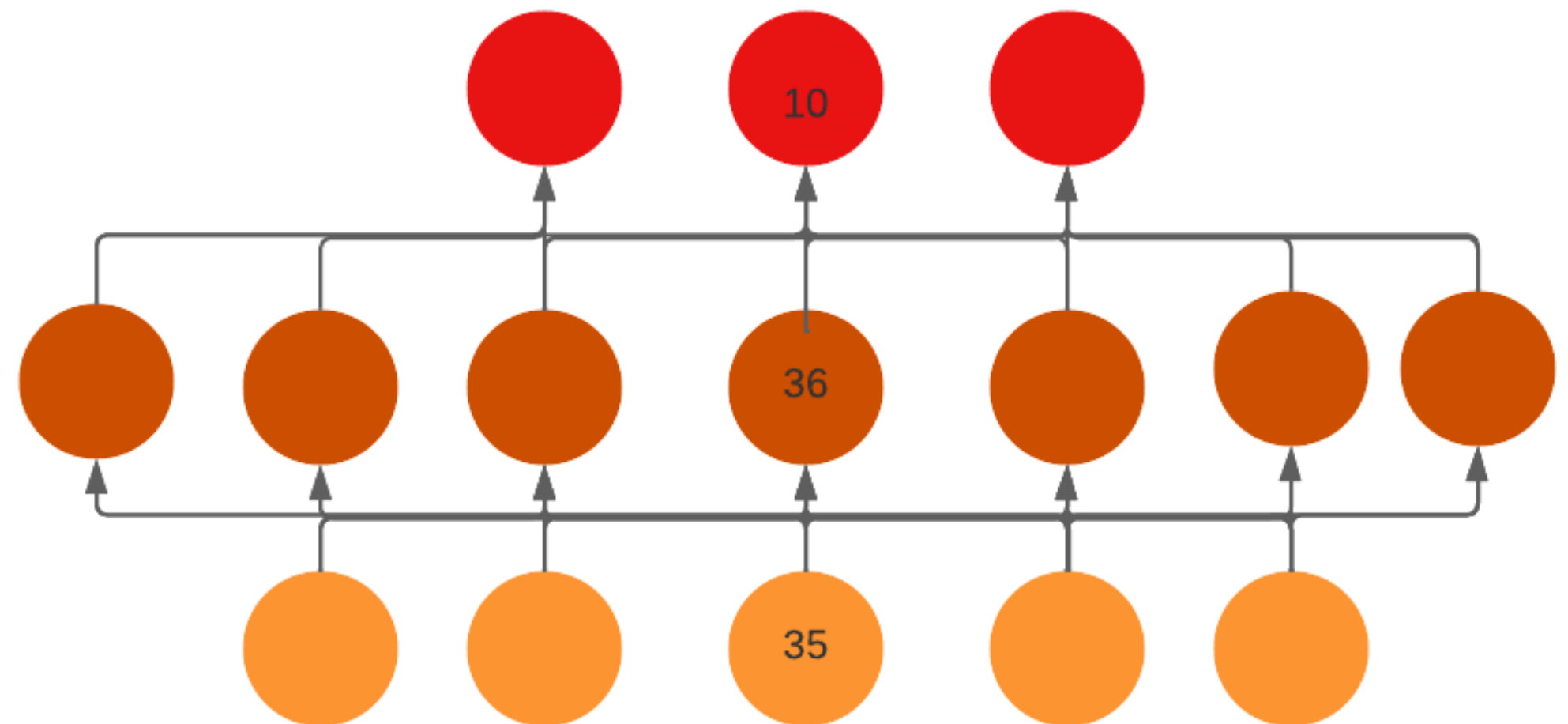
Cuando se testea un número que nunca entreno es prácticamente aleatorio que logre acertar la paridad del mismo.

Perceptrón Multicapa

**DISTINCIÓN DE NÚMEROS
(0-9)**

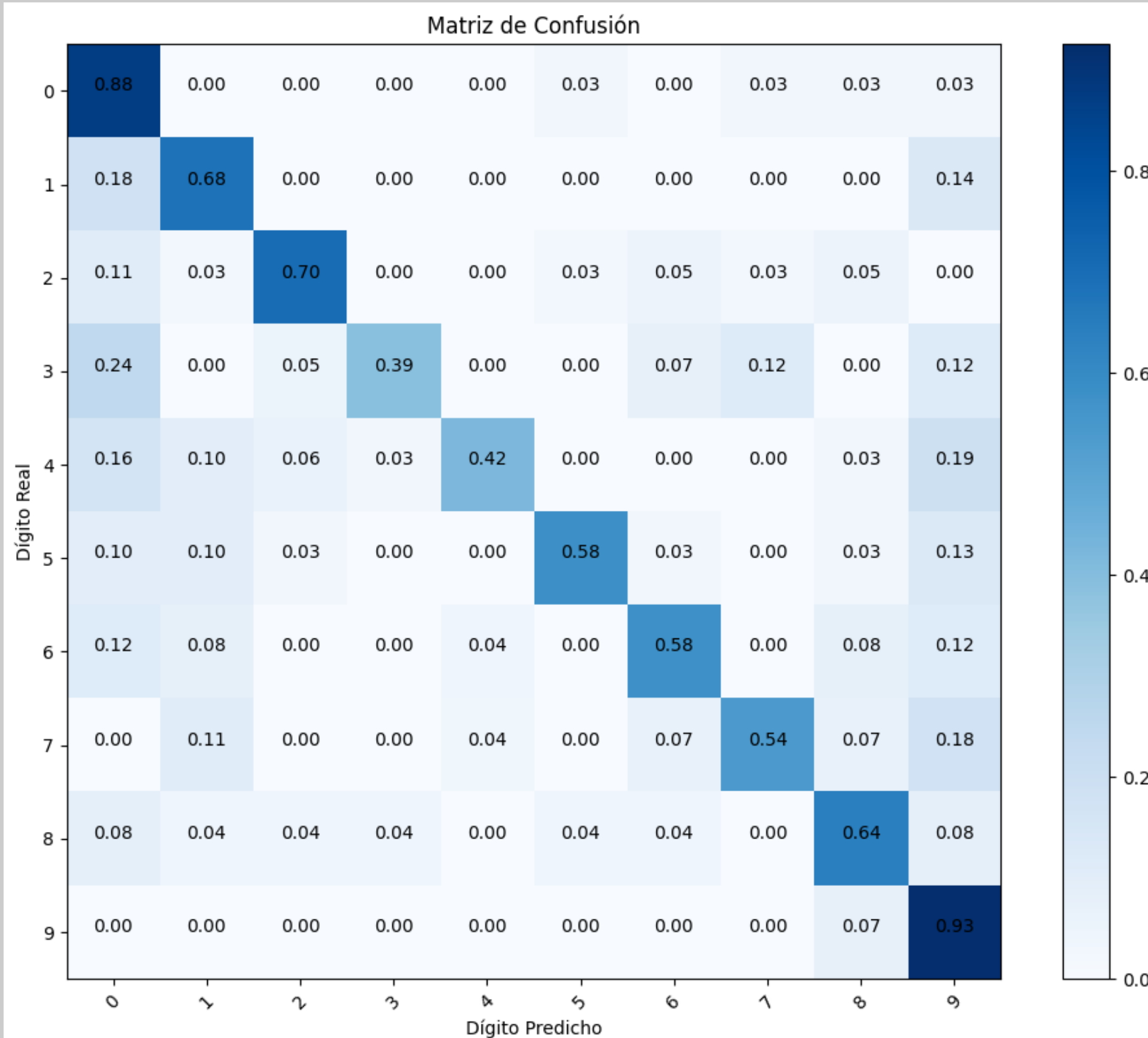
Learning Rate: 0.001
Training Percentage: 1*
Epochs: 1000
Bias: 1
Beta: 1
Epsilon: 0.5
Optimizer: "ADAM"

* Se testea con los mismos dígitos
+ ruido

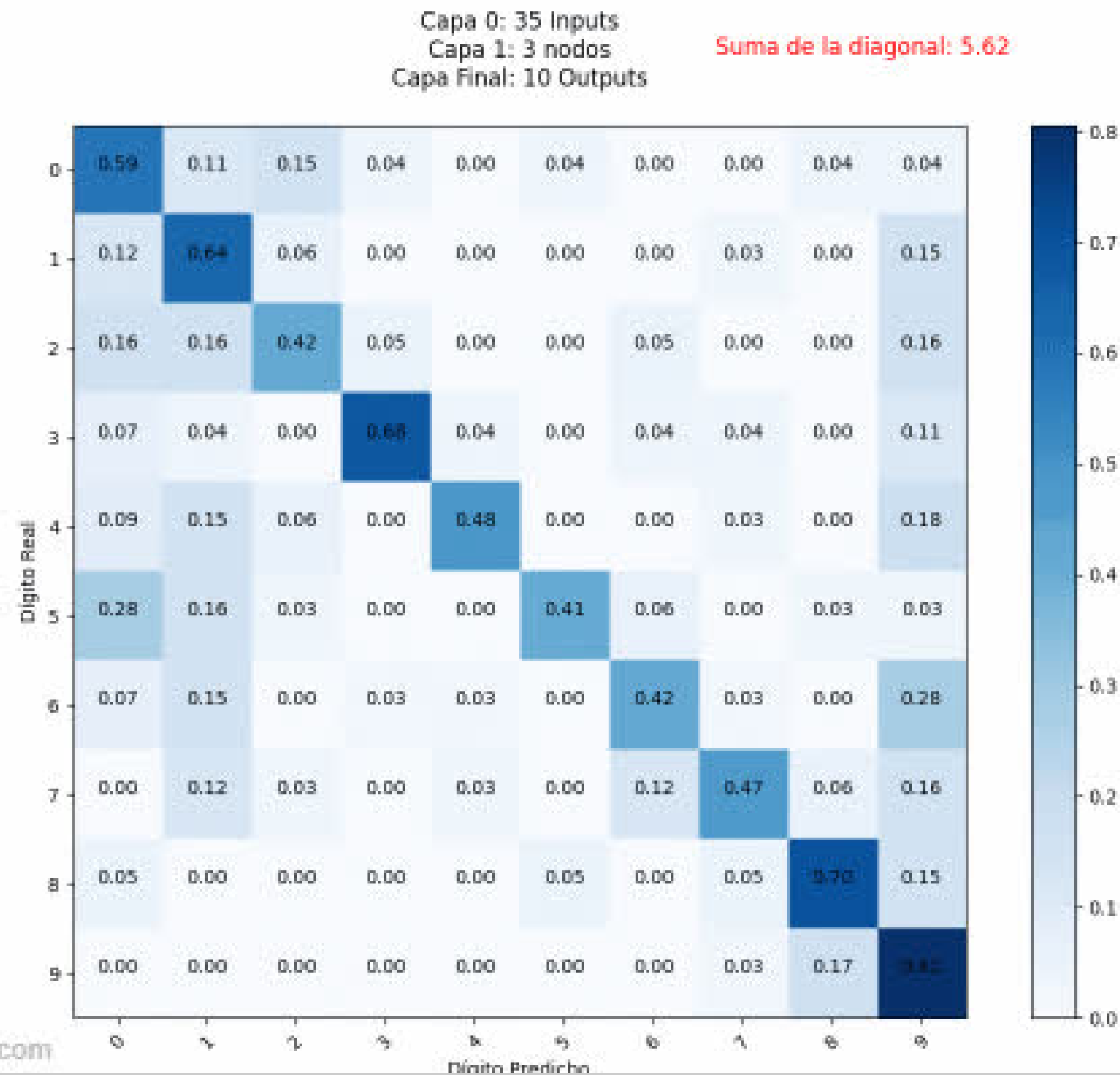


X

Con un poco de ruido, la red fue capaz de utilizar todos los datos como entrenamiento y testearse con datos nuevos.



Perceptrón Multicapa



DISTINCIÓN DE NÚMEROS (0-9)

Notamos que agregar más nodos y/o capas no implica una mayor precisión a la hora de clasificar.

Conclusiones

- Un conjunto acotado de datos no permite que la red entrene bien.
- En caso de tener un conjunto de datos acotado, podemos transformarlo un poco (ruido) para hacer más grande el conjunto disponible y así permitir un mejor entrenamiento del perceptrón.
- La combinación de capas ideal de la red neuronal debe ser experimentada. No hay una receta (por ejemplo, más neuronas = mejor resultado)