

SIA - TP N°3

Perceptrón simple y multicapa

Grupo 5

Integrantes:

- Catalán, Roberto José - 59174
- Dell'Isola, Lucas - 58025
- Galende, Lautaro - 60287

1

Perceptrón simple con función escalón

Problemas implementados

AND



XOR

Función AND

Entrada		Salida esperada
-1	1	-1
1	-1	-1
-1	-1	-1
1	1	1

Salidas obtenidas				
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
1	1	1	1	1



- 5 ejecuciones
- Parámetros utilizados
 - Épocas = 100
 - Iteraciones = 500
 - Tasa de aprendizaje = 0.1

Función XOR

Entrada		Salida esperada
-1	1	1
1	-1	1
-1	-1	-1
1	1	-1

Salidas obtenidas				
1	1	-1	-1	1
1	-1	1	1	1
-1	-1	-1	-1	1
1	1	-1	1	-1
×	×	×	×	×

- 5 ejecuciones
- Parámetros utilizados
 - Épocas = 100
 - Iteraciones = 500
 - Tasa de aprendizaje = 0.1

Perceptrón simple escalón

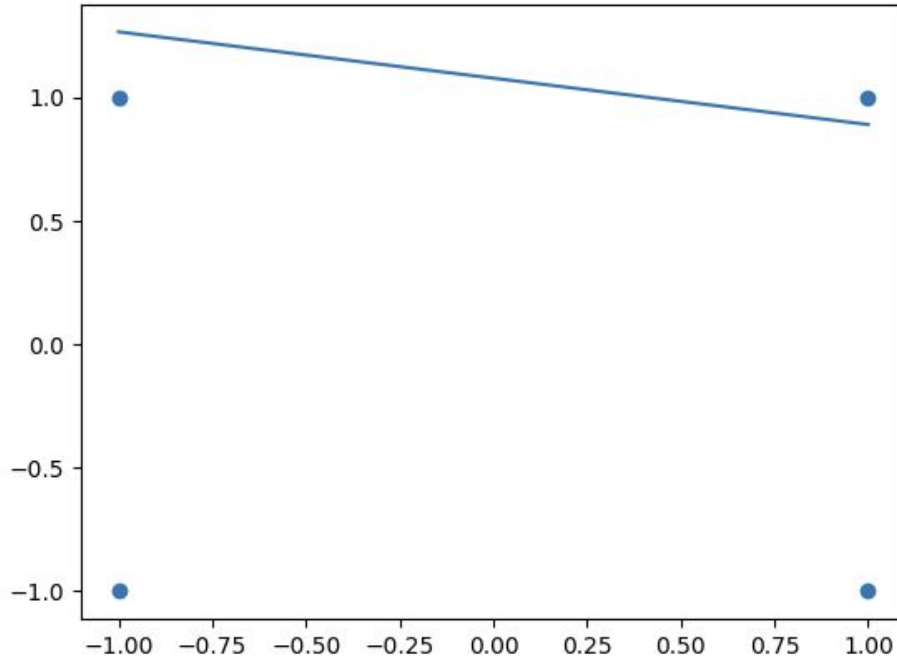
¿Qué problemas resuelve?

- Problemas de separabilidad lineal

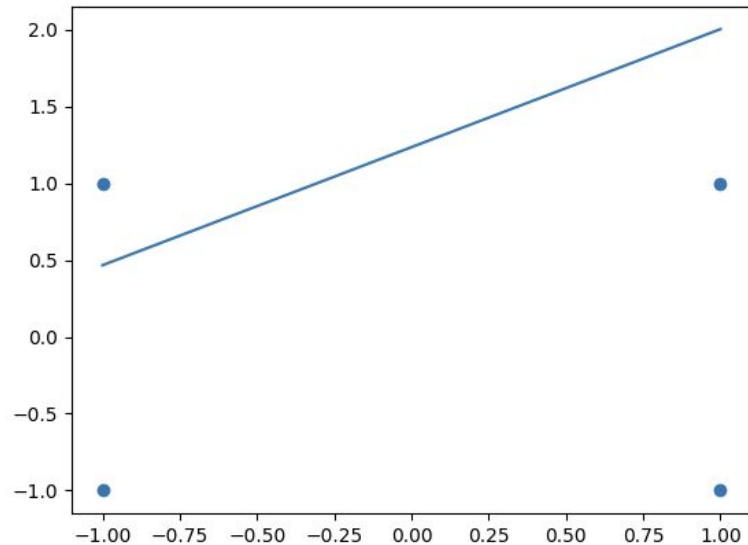
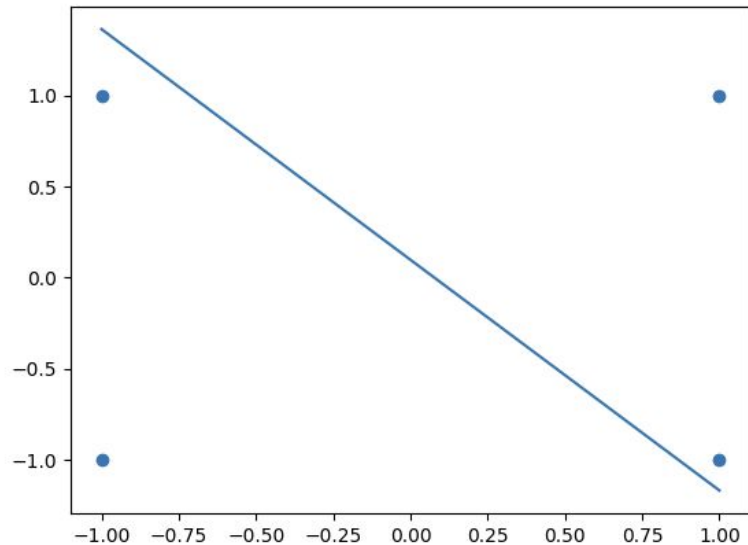
¿Cómo resolvió los problemas pedidos?

- AND ✓
- XOR ✗

Hiperplano para el AND



Hiperplanos para el XOR





2

Perceptrón simple lineal y no Lineal



2.1

Perceptrón lineal

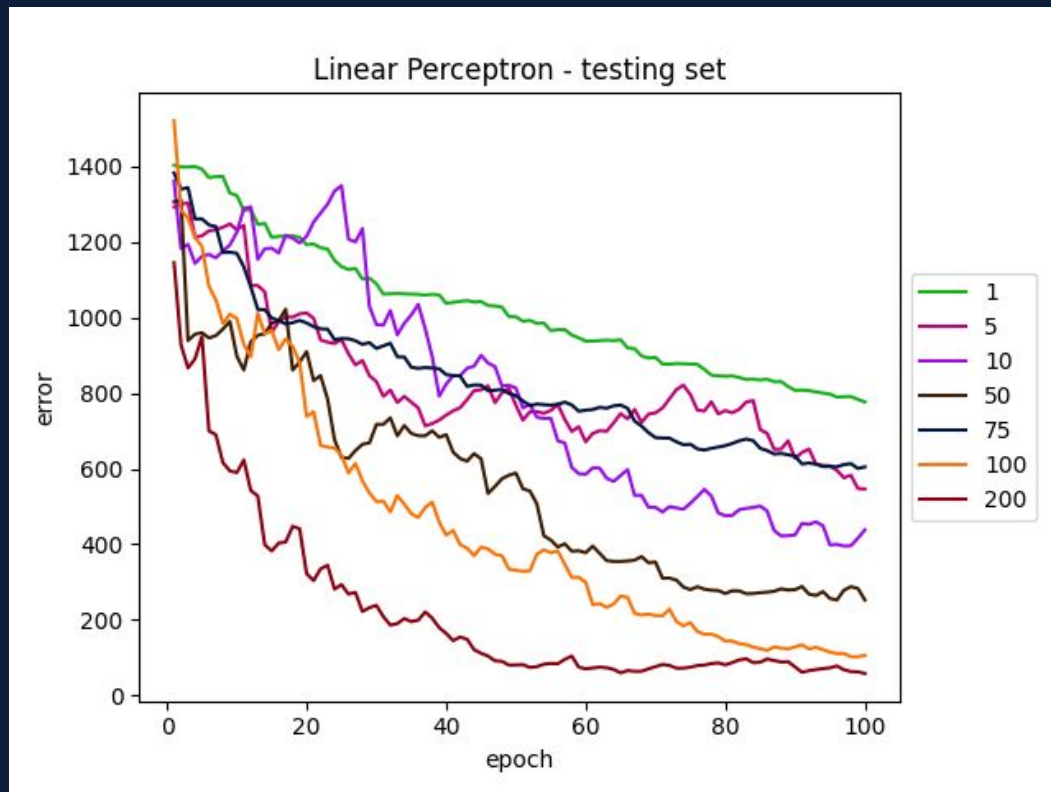
Perceptrón simple lineal

- Entradas
 - TP3-ej2-Conjunto-entrenamiento.txt
- Salidas esperadas
 - TP3-ej2-Salida-deseada.txt
- Parámetros utilizados
 - Épocas \rightarrow 100
 - Iteraciones \rightarrow 200
 - Tasa de aprendizaje \rightarrow 0.01
- Cálculo del error
 - Error cuadrático medio

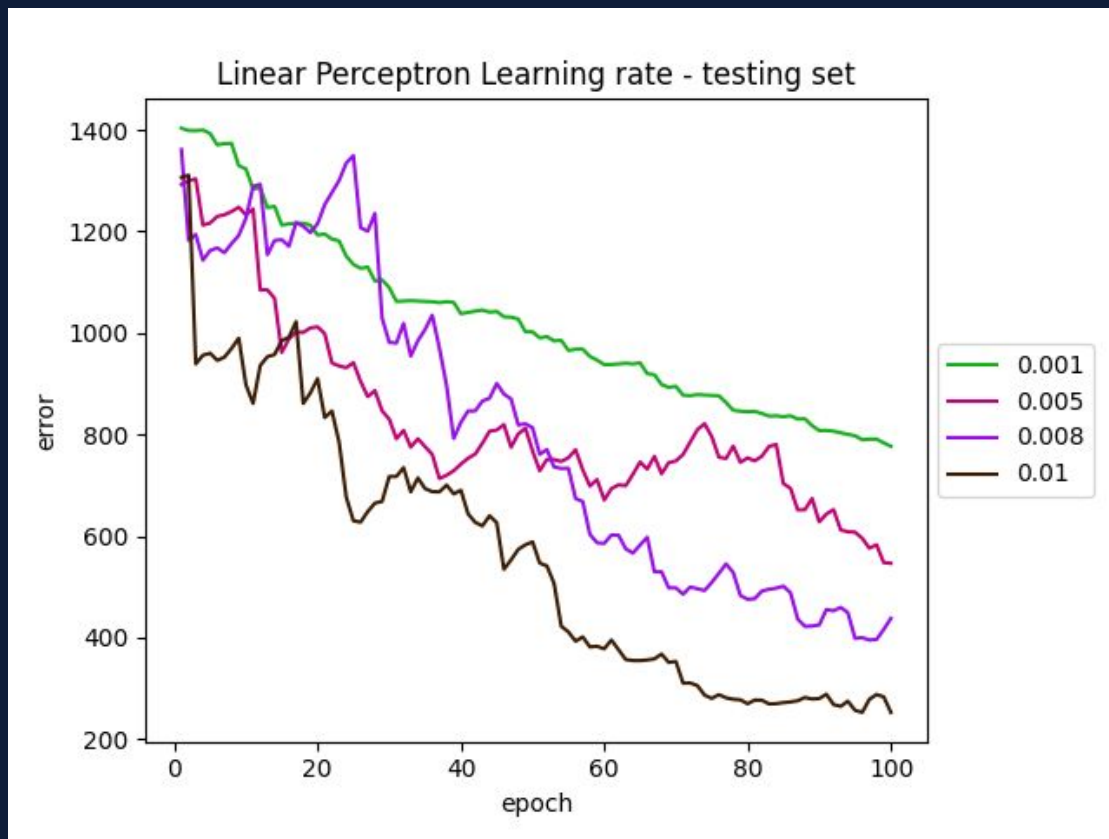
Análisis para encontrar los mejores parámetros

Se decidió realizarlo con 100 épocas
para todos los parámetros

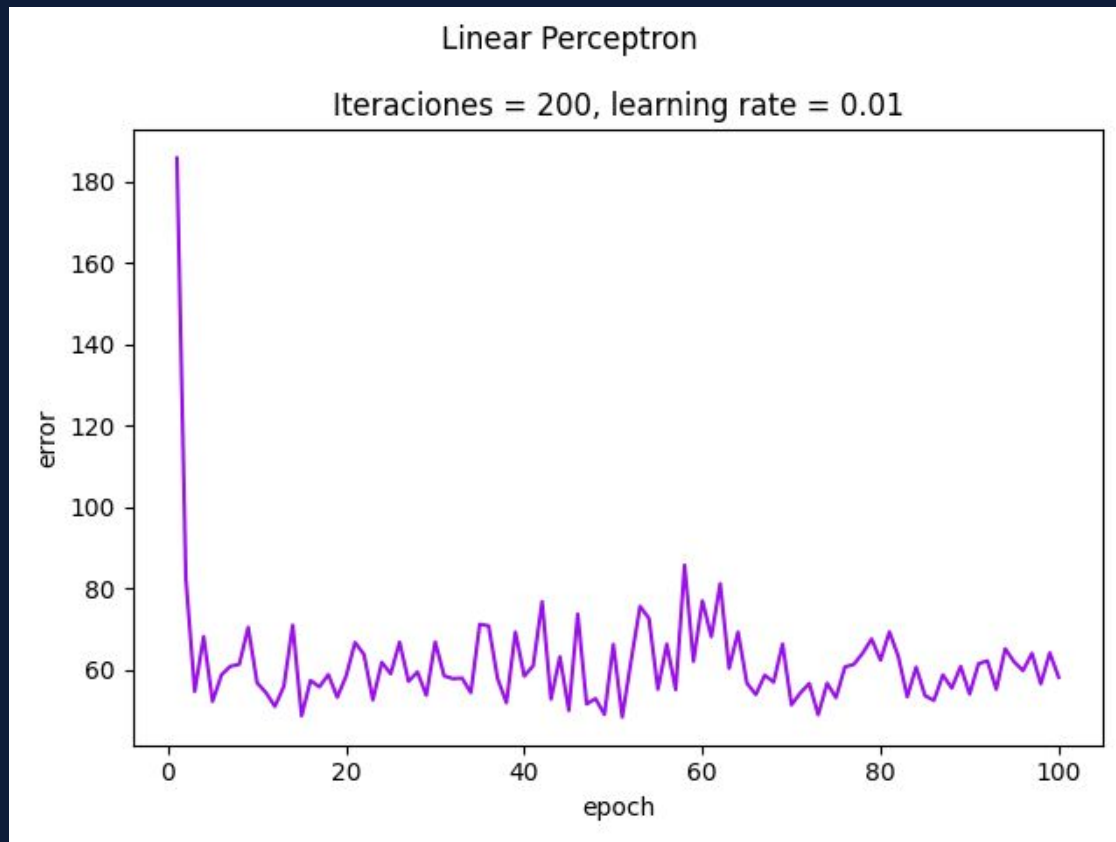
Parámetros - Cantidad de iteraciones



Parámetros - Tasa de aprendizaje



Error vs Épocas





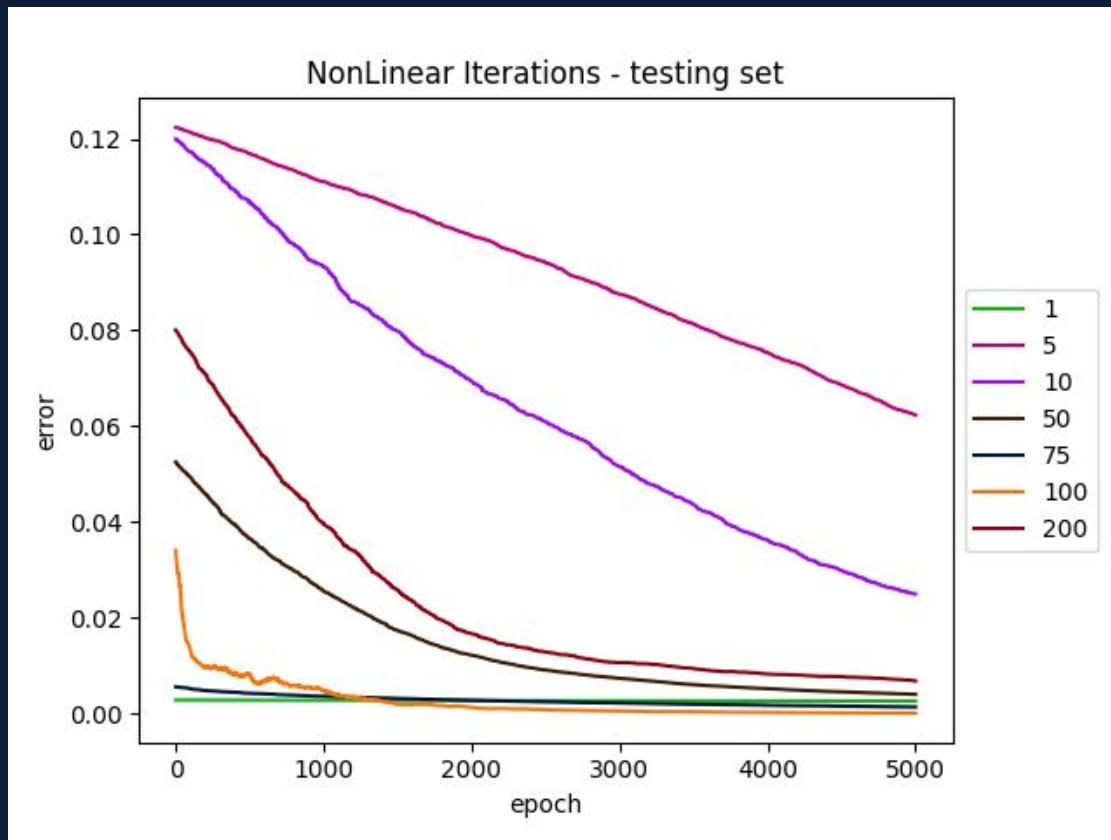
2.2

Perceptrón no lineal

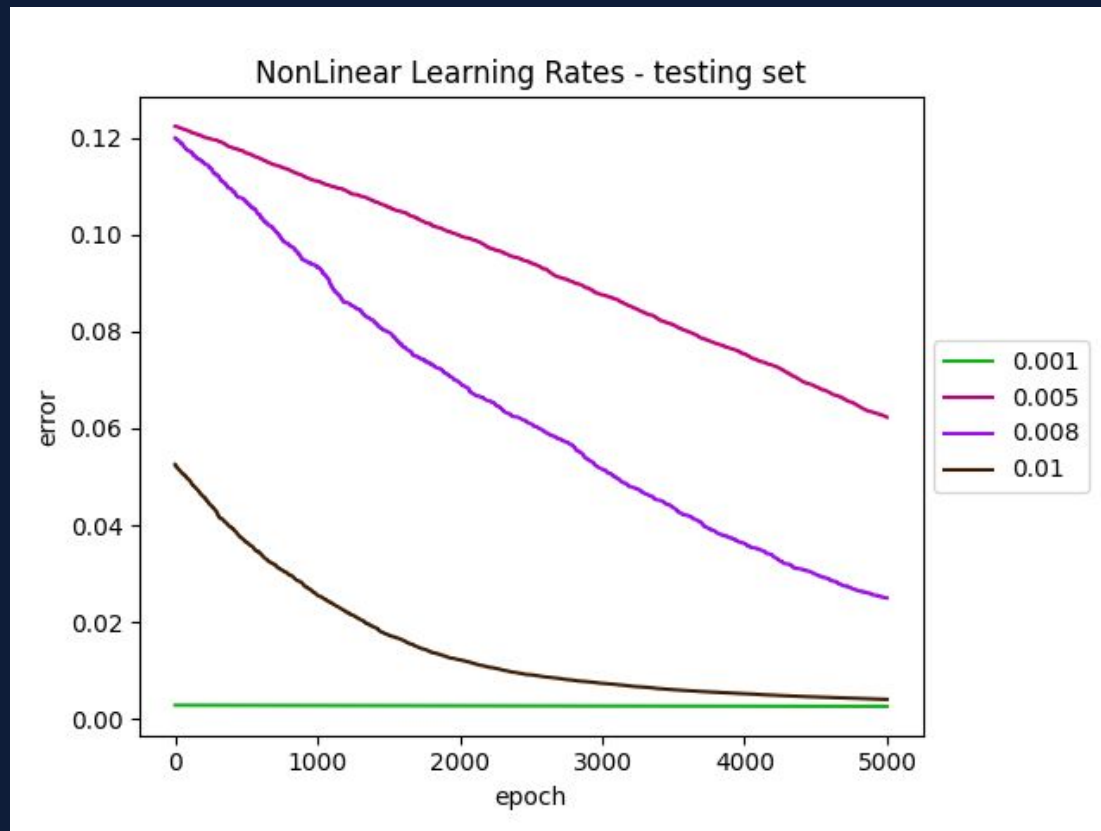
Perceptrón simple no lineal

- Entradas
 - TP3-ej2-Conjunto-entrenamiento.txt
- Salidas esperadas
 - TP3-ej2-Salida-deseada.txt
- Parámetros utilizados
 - Épocas \rightarrow 100
 - Iteraciones \rightarrow 100
 - Tasa de aprendizaje \rightarrow 0.01
 - $\beta = 1$
- Salida se encuentra normalizada
- Cálculo del error
 - Error cuadrático medio

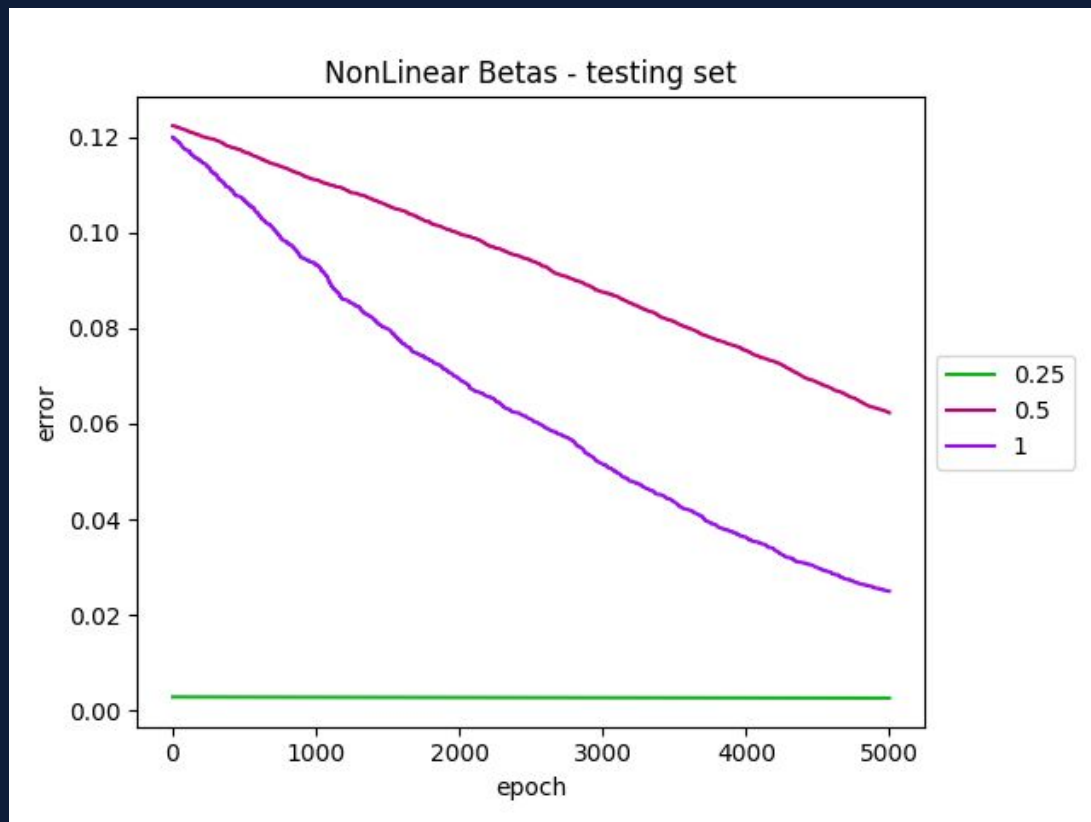
Parámetros - Cantidad de iteraciones



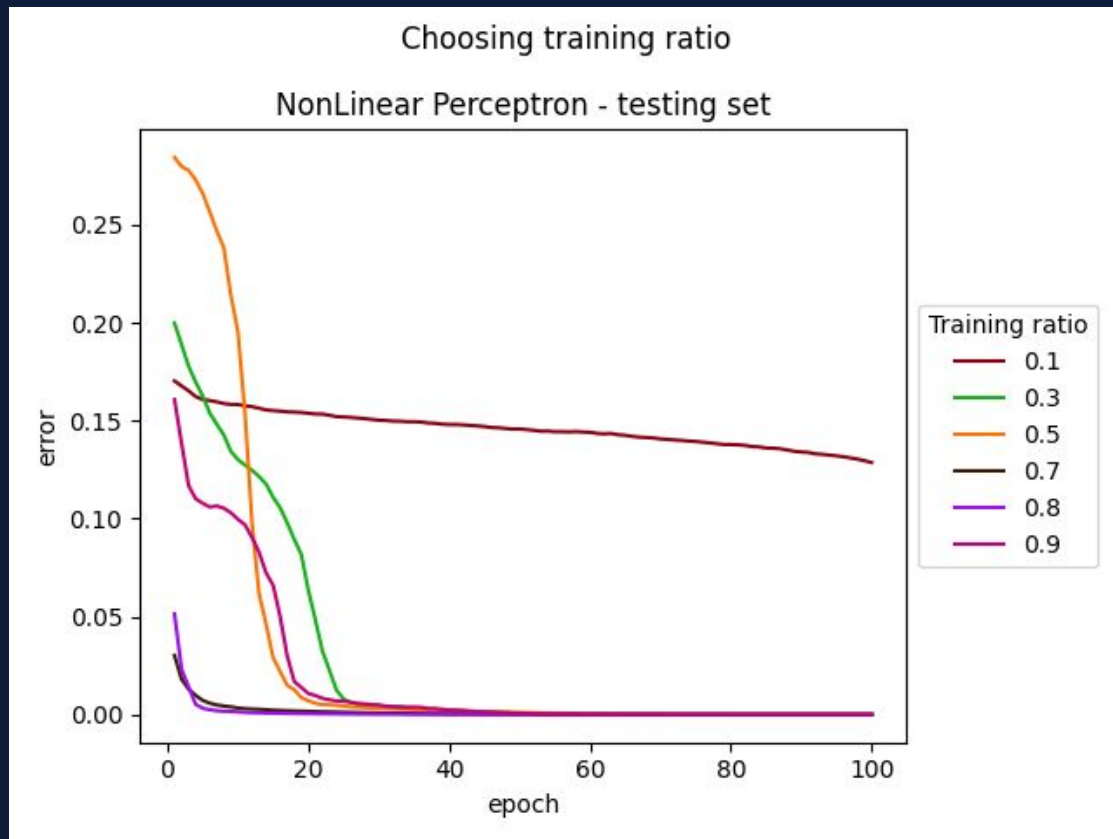
Parámetros - Tasa de aprendizaje



Parámetros - Beta

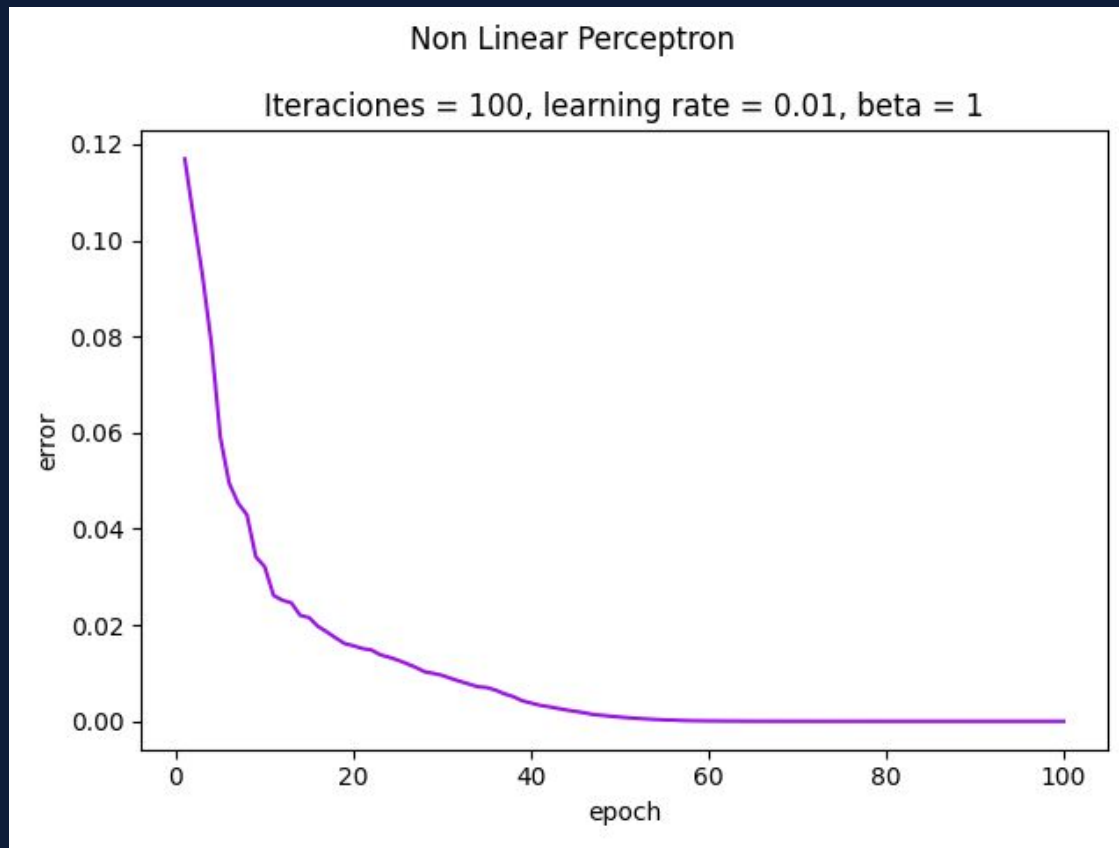


Parámetros - Training Ratio



⇒ 80% - 20%

Error vs Épocas



Capacidad de aprendizaje

Perceptrón simple lineal

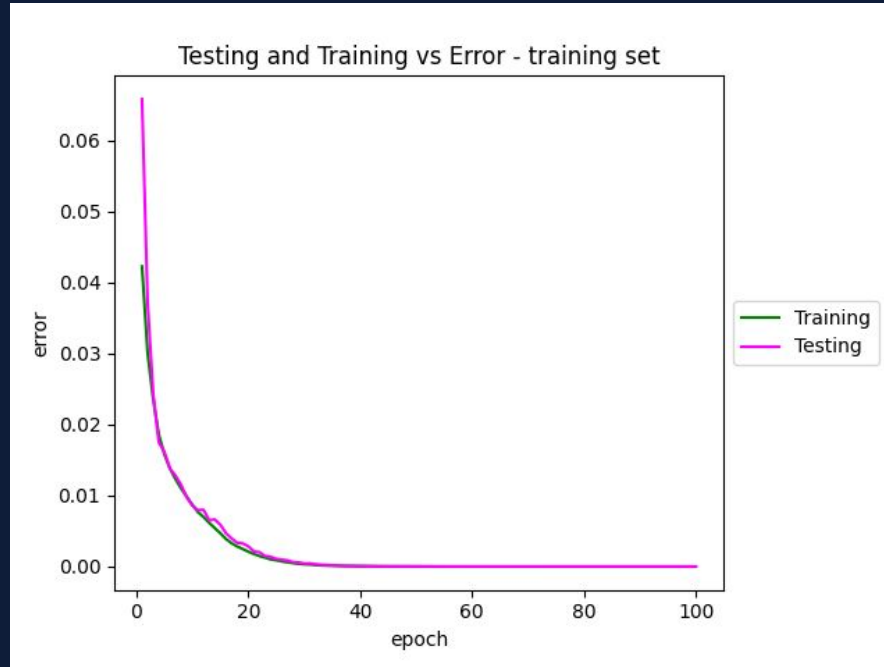
- Mala
 - No logra aprender la función dada con suficiente precisión

Perceptrón simple no lineal

- Buena
 - Logra aprender la función dada con un error aceptable

Perceptrón simple no lineal

Capacidad de generalización



Perceptrón simple lineal y no lineal

¿Cómo escoger el mejor conjunto de entrenamiento?

¿Cómo evaluar la máxima capacidad de generalización para este conjunto de datos?

Validación cruzada



3

Perceptrón multicapa

Problemas implementados

XOR



Par o impar



**Identificar
dígito**

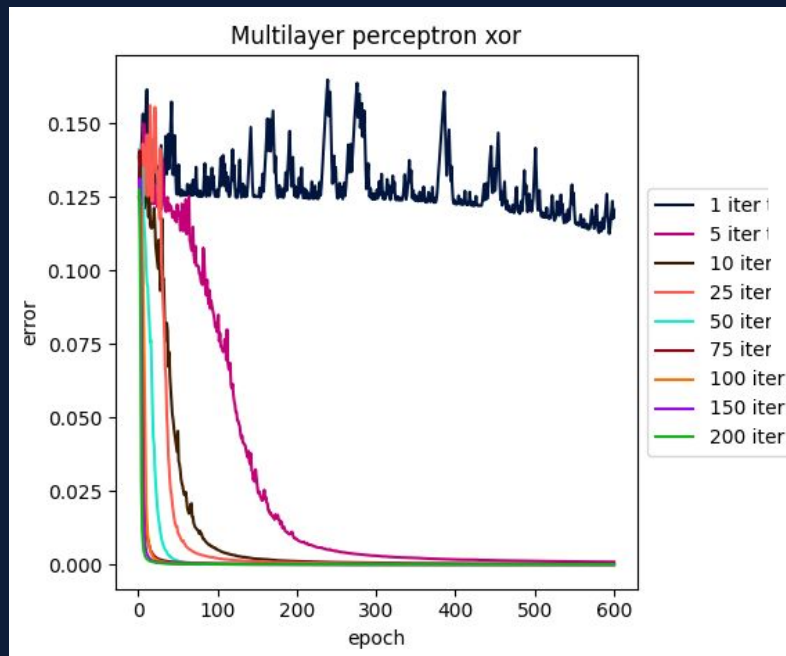


3.1

Función XOR

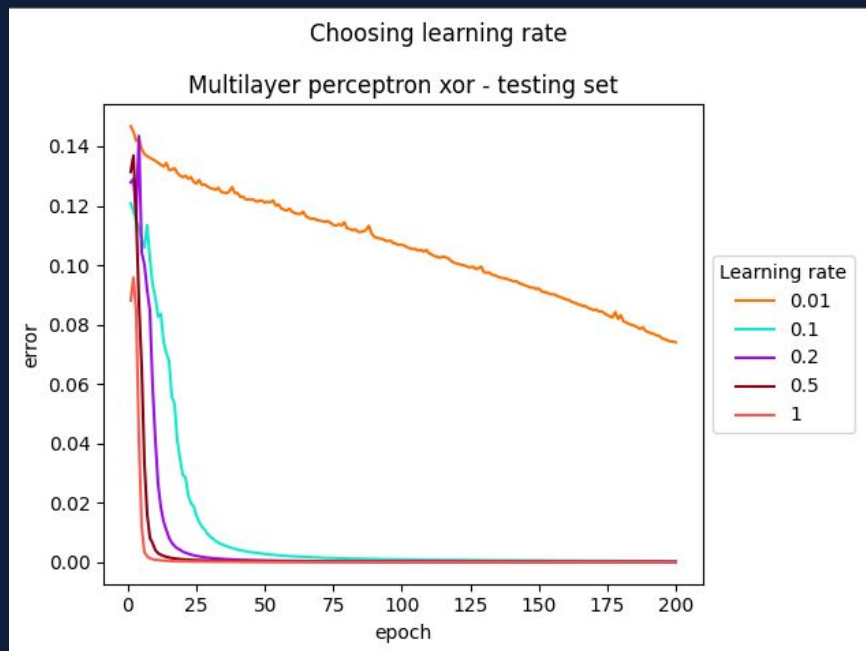
Función XOR

Elección de parámetros: cantidad de iteraciones



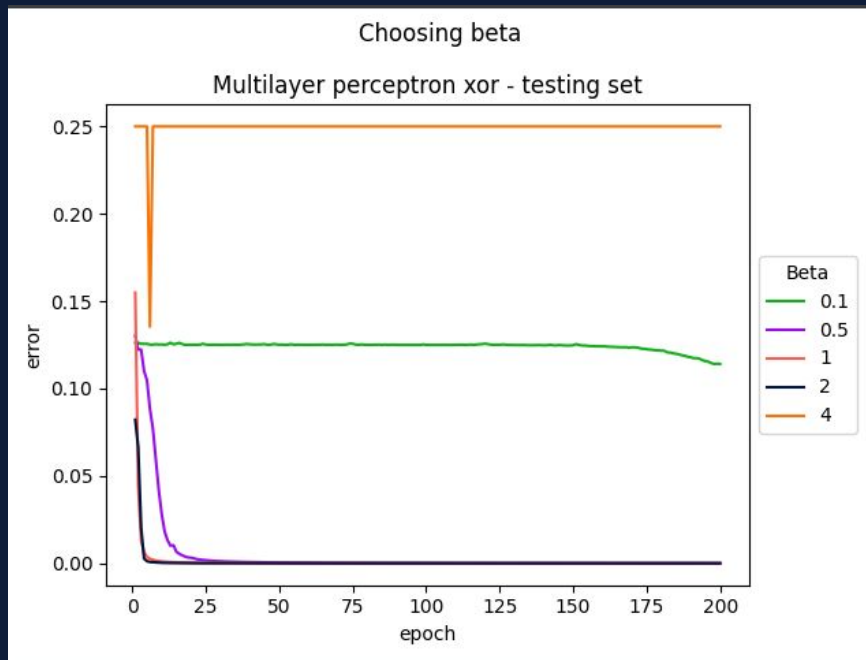
Función XOR

Elección de parámetros: tasa de aprendizaje



Función XOR

Elección de parámetros: beta



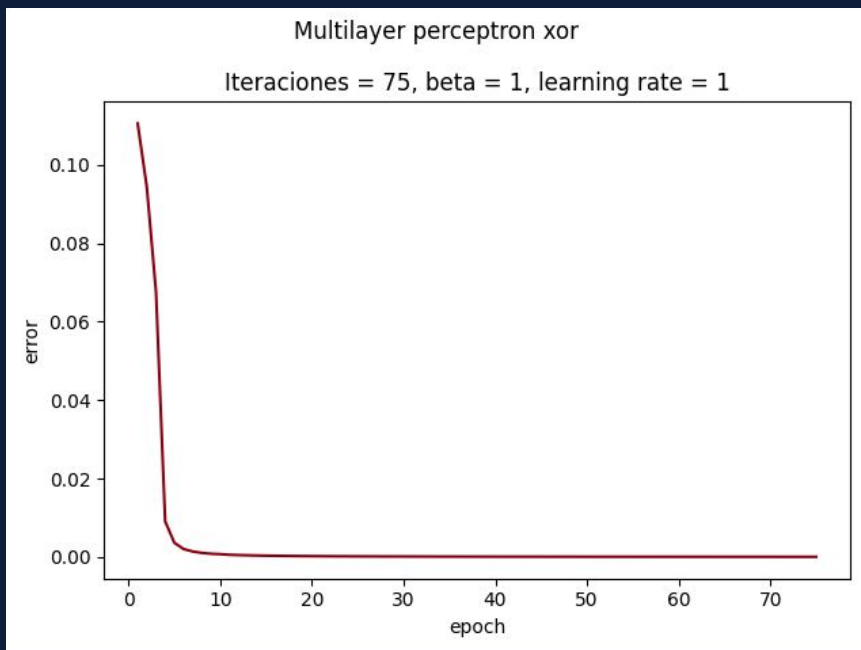
Función XOR

Entrada		Salida esperada
-1	1	1
1	-1	1
-1	-1	-1
1	1	-1

- Parámetros
 - Épocas $\rightarrow 75$
 - Iteraciones $\rightarrow 75$
 - Tasa de aprendizaje $\rightarrow 1$
 - Capas ocultas $\rightarrow 1$
 - Neuronas $\rightarrow 5$
 - $\beta = 1$

Función XOR

Error vs épocas



Función XOR

Resultados

- El perceptrón resuelve el problema



3.2

Par o impar

Par o impar

Entradas

- TP3-ej3-mapa-de-pixeles-digitos-decimales.txt
 - Imágenes de 5 x 7 pixeles
 - Representan los números del 0 al 9

Salidas esperadas

- (1, 0)
 - Si el número es par
- (0, 1)
 - Si el número es impar

Par o impar

Conjunto de entrenamiento

- Subconjunto del conjunto de entrada

Conjunto de testeo

- Complemento del subconjunto seleccionado

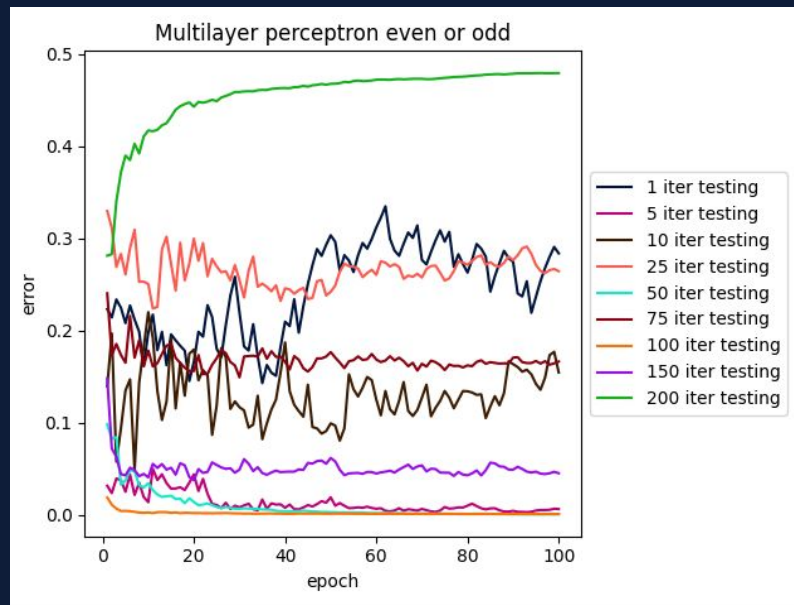
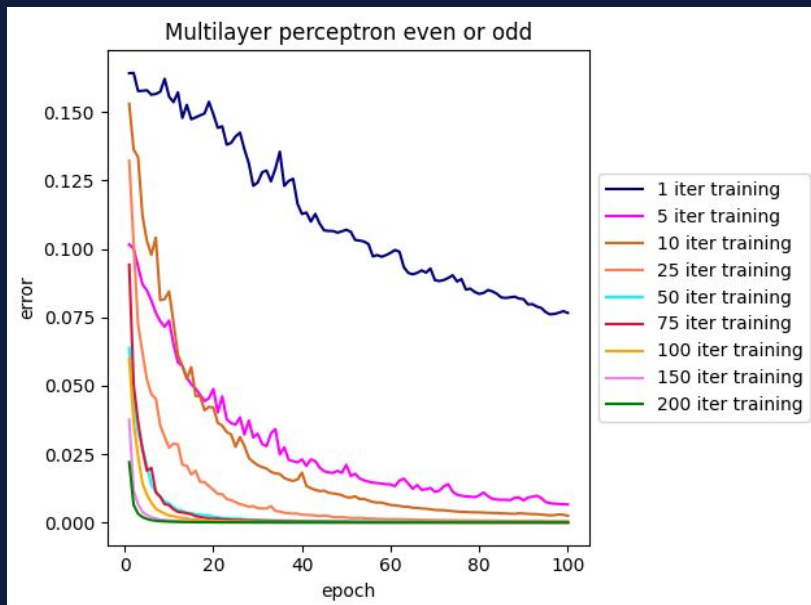
Par o impar

Parámetros

- Épocas \rightarrow 100
- Iteraciones \rightarrow 50
- Tasa de aprendizaje \rightarrow 0.2
- Capas ocultas \rightarrow 1
- Neuronas \rightarrow 30
- $\beta = 1$

Par o impar

Resultados



Par o impar

Resultados y conclusiones

- NO resuelve el problema
 - Era de esperar, pues los dibujos de los dígitos no tienen relación con su paridad



3.3

Identificar dígito

Identificar dígito

Entradas

- TP3-ej3-mapa-de-pixeles-digitos-decimales.txt
 - Imágenes de 5 x 7 pixeles
 - Representan los números del 0 al 9

Salidas esperadas

- Arreglo de 10 elementos con el elemento i en 1 y todos los demás en 0
 - Siendo i el número representado por la imagen de entrada

Ejemplo:

- Entrada: imagen que representa el 2
- Salida esperada: [0, 0, 1, 0, 0, 0, 0, 0, 0]

Identificar dígito

Conjunto de entrenamiento

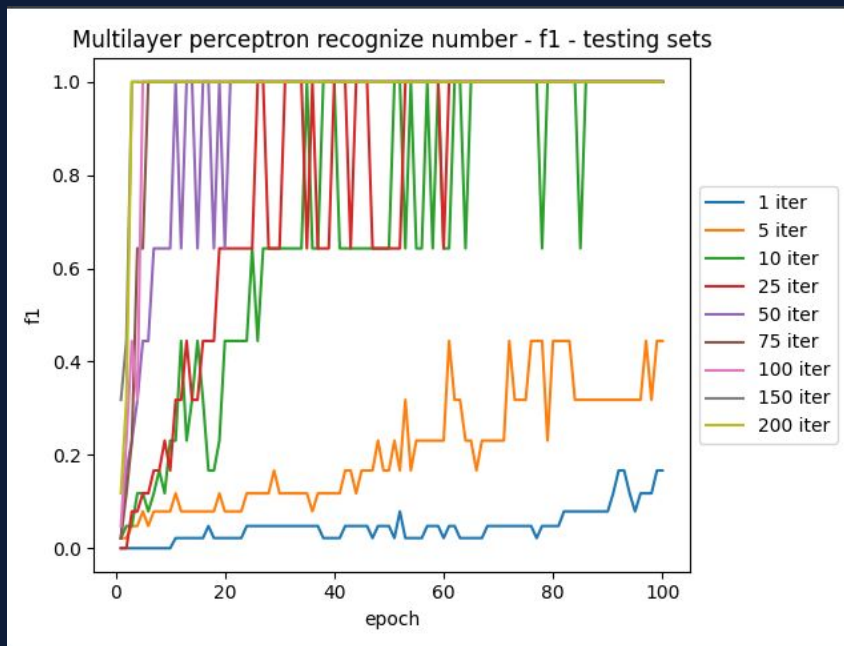
- Conjunto de entrada (imágenes con los dígitos del 0 al 9)

Conjunto de testeo

- Conjunto de entrada con ruido
 - Cada bit de la imagen tiene probabilidad 0.02 de intercambiar su valor

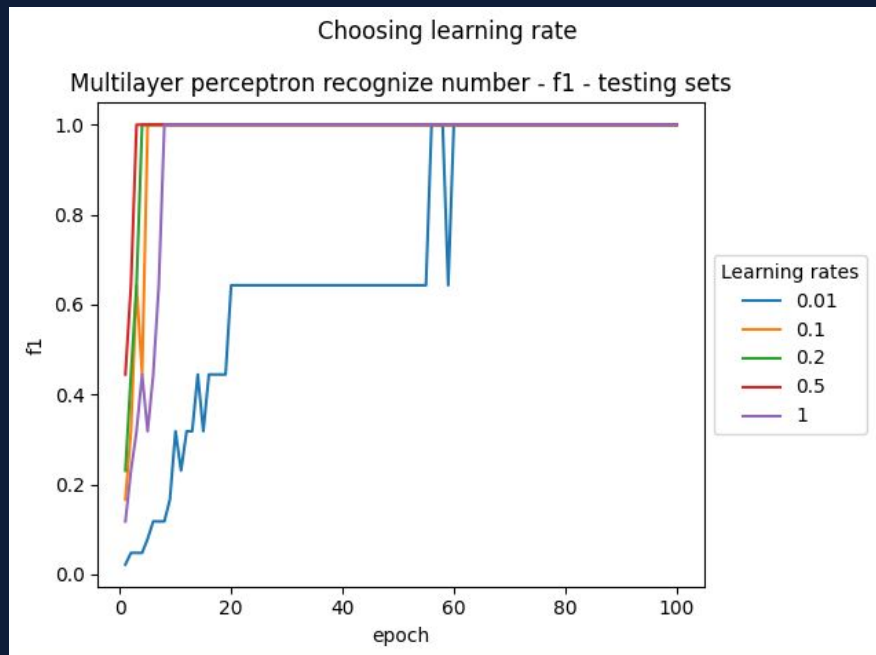
Identificar dígito

Elección de parámetros: cantidad de iteraciones



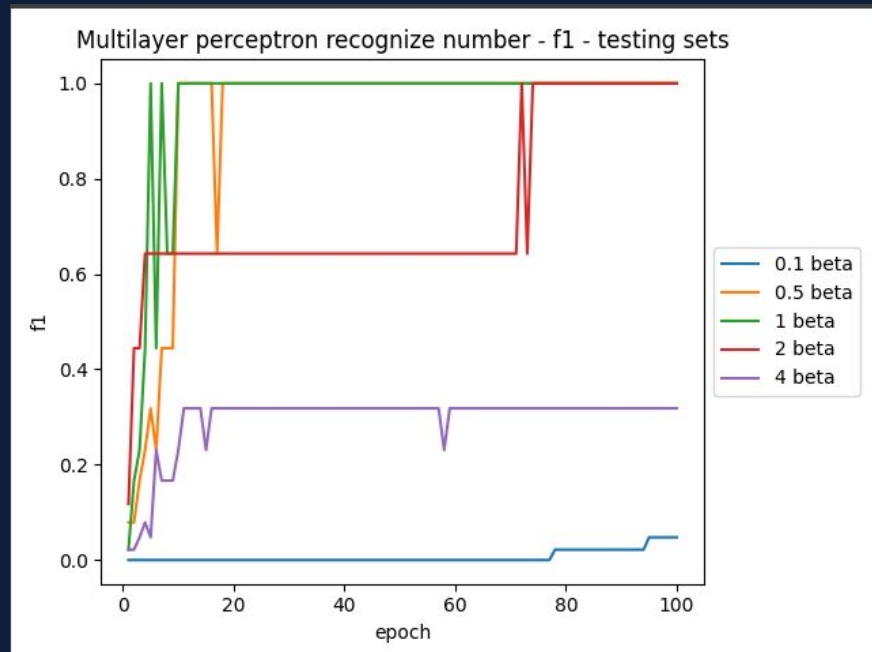
Identificar dígito

Elección de parámetros: tasa de aprendizaje



Identificar dígito

Elección de parámetros: beta



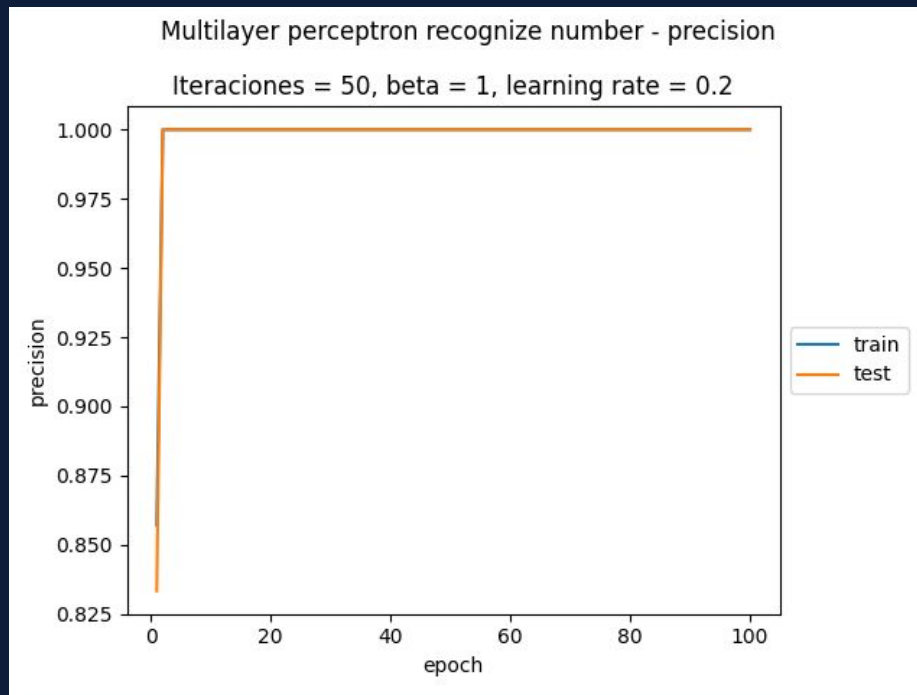
Identificar dígito

Parámetros

- Épocas \rightarrow 100
- Iteraciones \rightarrow 50
- Tasa de aprendizaje \rightarrow 0.2
- Capas ocultas \rightarrow 1
- Neuronas \rightarrow 30
- $\beta = 1$

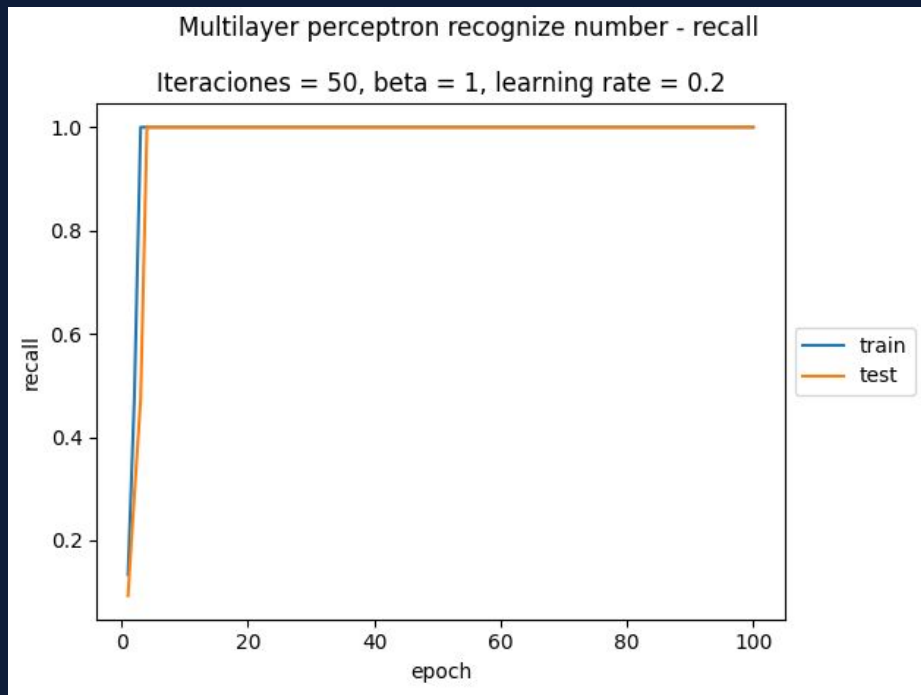
Identificar dígito

Métricas: precision



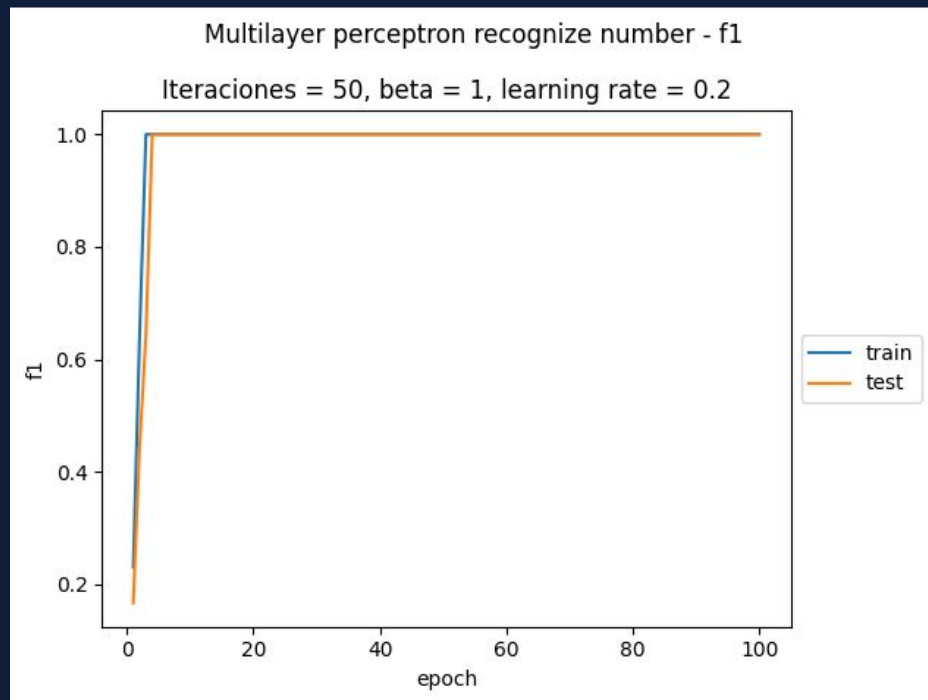
Identificar dígito

Métricas: recall



Identificar dígito

Métricas: f1-score



Identificar dígito

Resultados y conclusiones

- Logra resolver el problema de manera exitosa en pocas épocas
 - La configuración elegida es adecuada

GRACIAS