TP3 Perceptrón Simple y Multicapa

Sistemas de Inteligencia Artificial ITBA 2023 Q1

Grupo 4

Integrantes

Gaspar Budó Berra

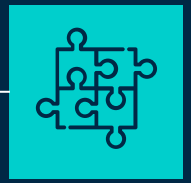
Marcos Dedeu

Marcus Galea Jacobsen

Santiago Hadad

Bruno Squillari

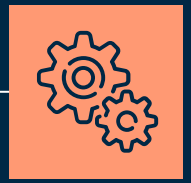
CONTENIDO



01

Ejercicio 1

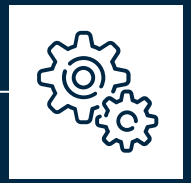
Perceptrón Simple con AND y XOR.



02

Ejercicio 2

Perceptrón Simple Lineal y no lineal.



03

Ejercicio 3

Perceptrón Multicapa para el problema del EJ1, detectar paridad y detectar dígitos.

Implementación

Implementacion:

- Perceptrón multicapa genérico con cálculos matriciales.

Cálculo de gradiente:

- Back propagation

Métodos de optimización:

- Gradiente descendiente
- Adam (b1 = 0.8 , b2= 0.8, e = 10e-8)

Función de costo a optimizar:

- Diferencia entre las predicciones y los resultados al cuadrado.

Funciones de activación:

- Step
- Tangente hiperbolica
- Sigmoidea
- Identidad

Condición de corte:

- Épocas de entrenamiento menor a 50000
- Error total menor a epsilon

Ejercicio 1

Consideraciones Ej1

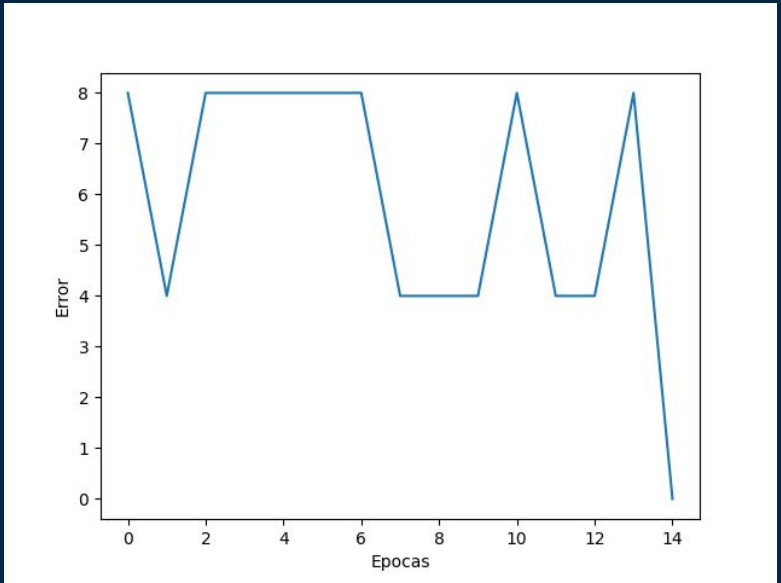
- ★ Aprendizaje supervisado de las funciones lógicas AND y XOR.
- ★ Perceptron simple
- ★ El conjunto de entrenamiento son todas las posibles salidas
- ★ No se pone a prueba la generalización del perceptrón
- ★ Minimización del error en el entrenamiento

Consideraciones Ej1

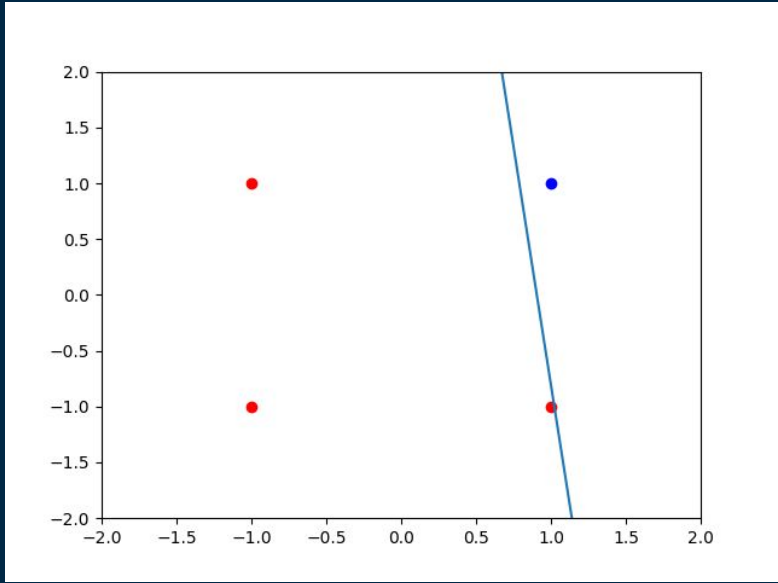
- ★ Adam como método de optimización.
- ★ Función de activación: Step(x)
- ★ Tasa de aprendizaje = 0.1
- ★ Epsilon = 10^{-11}
- ★ Los pesos se inicializan con valores aleatorios usando siempre la misma semilla.

Ej1: AND Resultados obtenidos

Error vs épocas de entrenamiento:

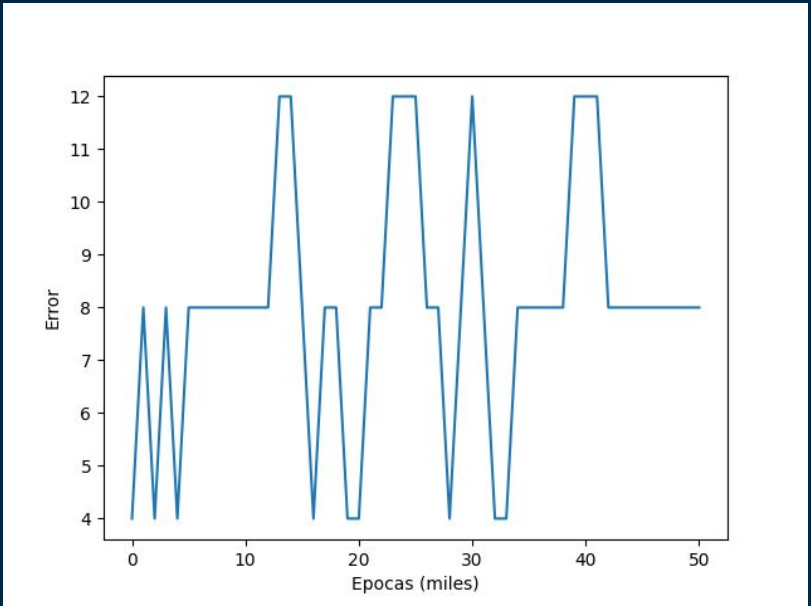


Hiperplano obtenido

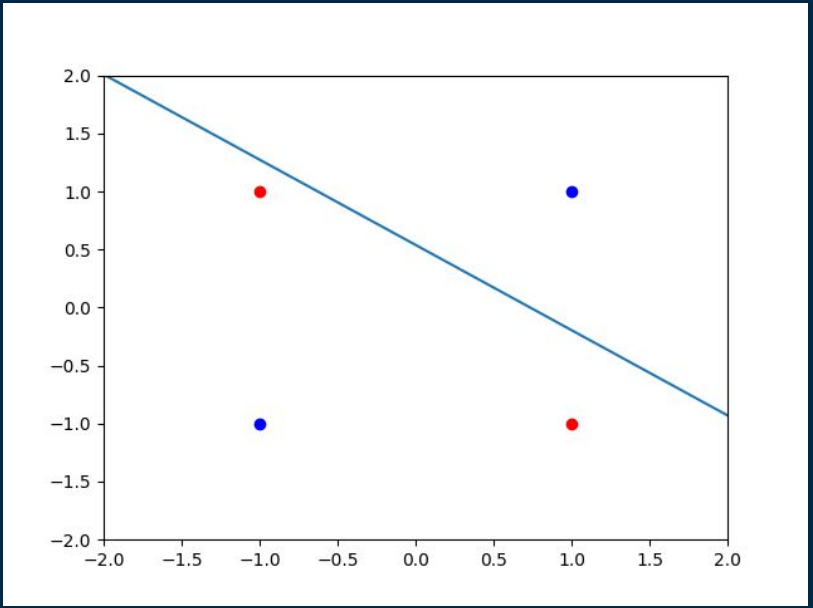


Ej1: XOR Resultados obtenidos

Error vs épocas de entrenamiento:



Hiperplano obtenido



Conclusiones Ej1

- El perceptrón simple resuelve problemas que son linealmente separables.
- El AND es linealmente separable por tanto se obtiene una solución exacta para el conjunto de entrenamiento.
- El OR no es linealmente separable corta por el límite impuesto al número de épocas de entrenamiento.

Ejercicio 2

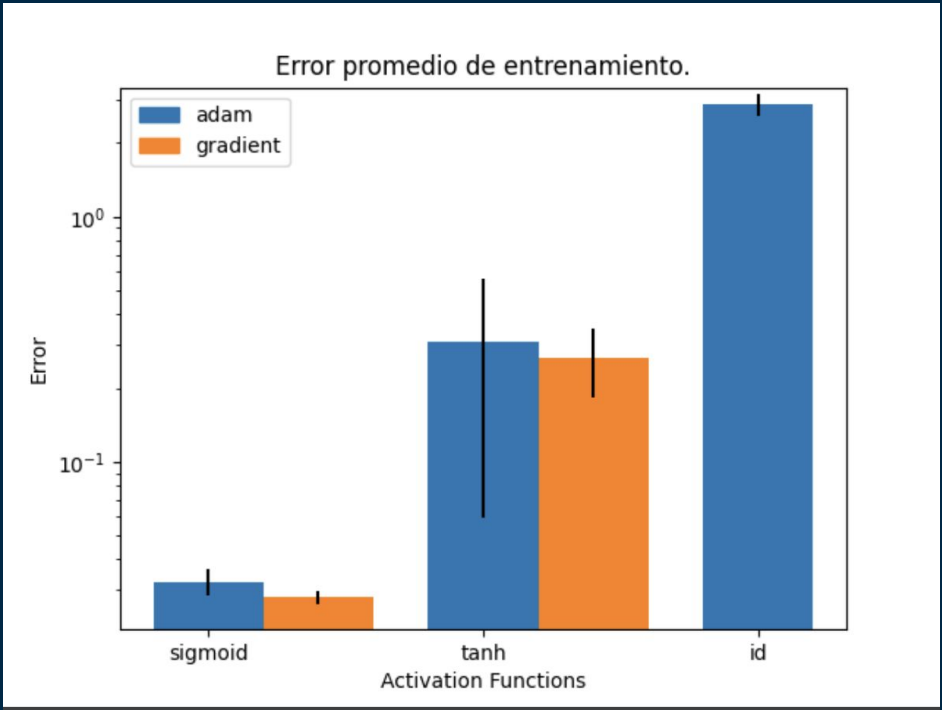
Posibles elecciones del conjunto de entrenamiento y generalización

- División aleatoria
- División estratificada
- División cruzada
- División temporal

Consideraciones Ej2

- ★ Se reescala el resultado esperado cuando corresponde por la función de activación usada
- ★ Optimizacion Adam
- ★ Conjunto de entrenamiento y generalización tomados aleatoriamente.
- ★ 80% entrenamiento y 20 % generalización.
- ★ El error de generalización se calcula como el error cuadrático medio.
- ★ Tasa de aprendizaje: 0.1
- ★ Epsilon: 10^{-11}

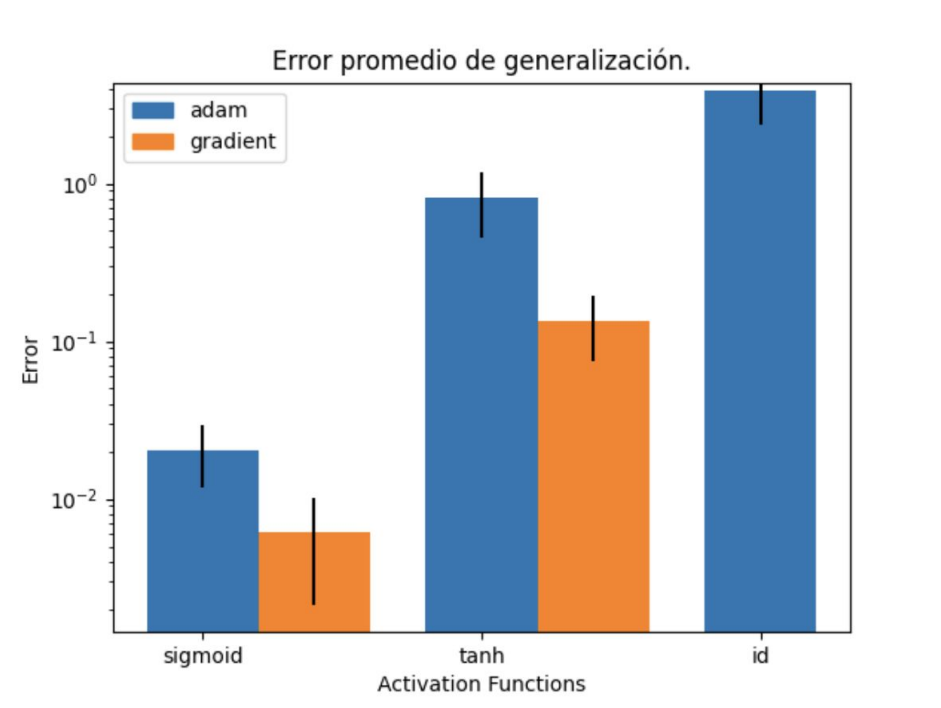
Ej 2 : Entrenamiento



El id con método de gradiente descendente lanza Overflow exception.

En todos los casos se alcanza la condición de corte de un máximo de 50000 iteraciones, debido a que no sería probable que lleguemos a un error menor al epsilon planteado.

Ej 2 : Generalización



El id con método de gradiente descendente lanza Overflow exception.

En todos los casos se alcanza la condición de corte de un máximo de 50000 iteraciones, debido a que no sería probable que lleguemos a un error menor al epsilon planteado.

Conclusiones Ej2

- La función sigmoidea obtiene los mejores resultados.
- La función de activación identidad obtiene los peores resultados

What happens if we let f be the identity? Then, in a network with L layers (we'll leave out W_0 for simplicity, but keeping it wouldn't change the form of this argument),

$$A^L = W^{L\,T} A^{L-1} = W^{L\,T} W^{L-1\,T} \cdots W^{1\,T} X \ .$$

So, multiplying out the weight matrices, we find that

$$A^L = W^{\text{total}} X \ ,$$

which is a *linear* function of X! Having all those layers did not change the representational capacity of the network: the non-linearity of the activation function is crucial.

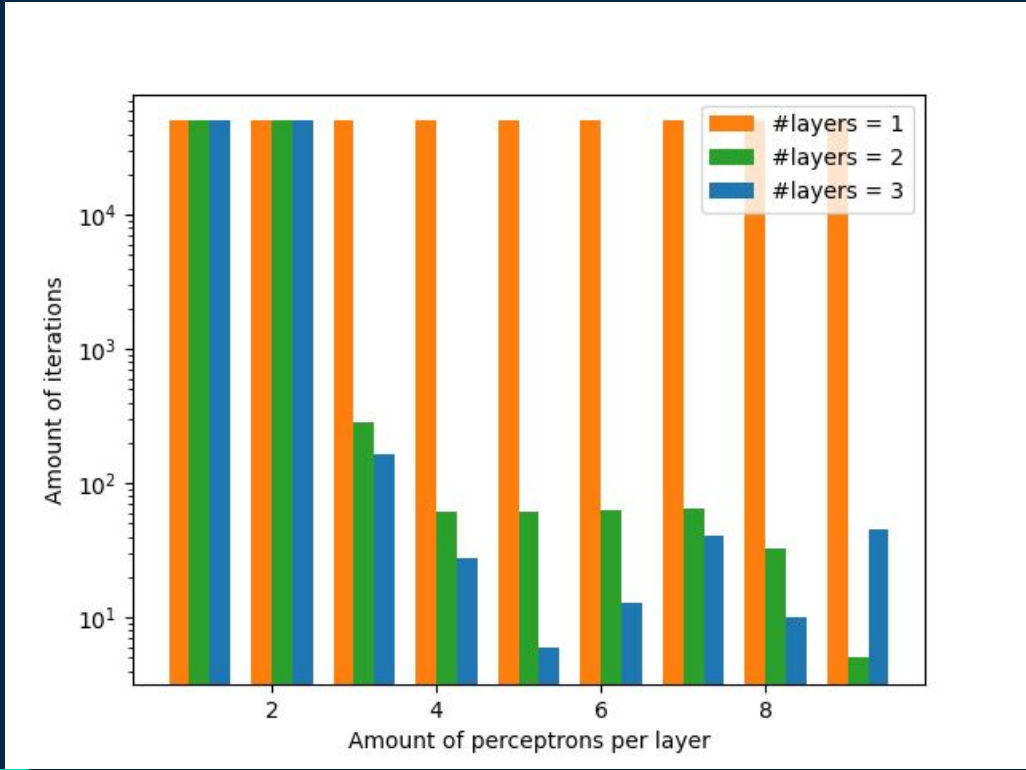
Ejercicio 3

Ejercicio 3a

Consideraciones

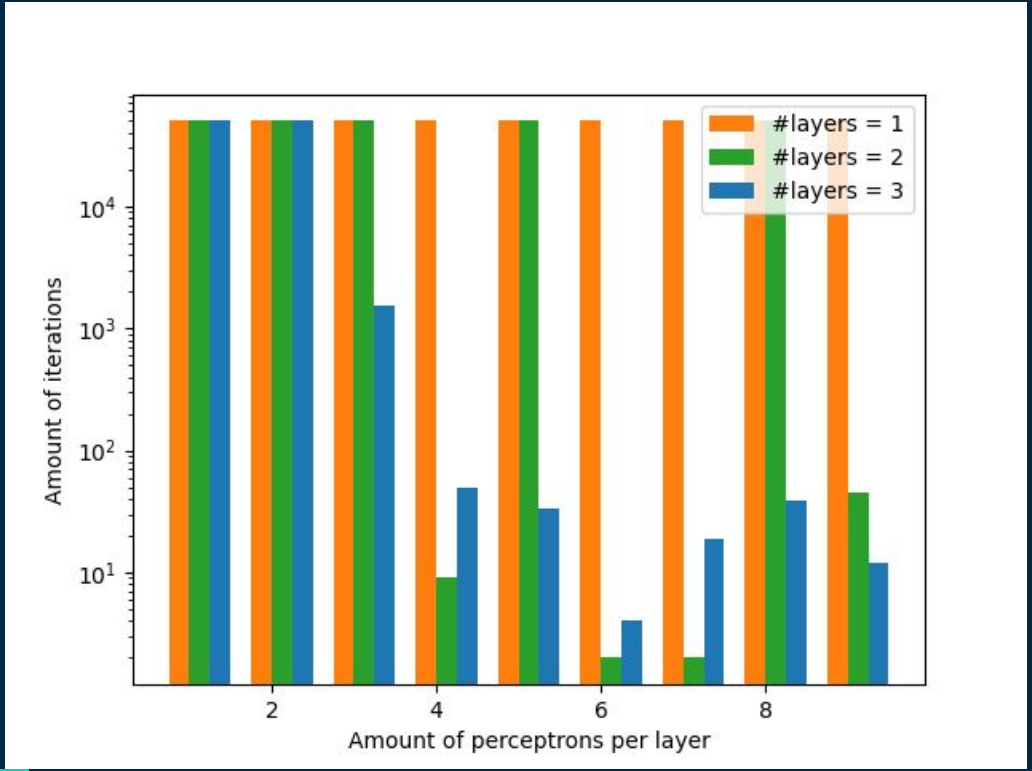
- ★ Sabemos que XOR lógico no es linealmente separable.
- ★ Misma condiciones iniciales que ejercicio 1.
- ★ El perceptrón multicapa nos brinda la posibilidad de obtener una solución exacta.
- ★ Se varía la arquitectura de la red neuronal.
- ★ Se usa 1, 2, y 3 capas de i neuronas cada una con 1<= i <=10.

Ej 3.a - Adam



- Menos iteraciones cuando hay más capas.

Ej 3.a – Gradiente



Condiciones idénticas al punto anterior.

Se sigue notando una mejora significativa al aumentar la cantidad de capas.

Conclusiones

- ★ La utilización de una sola capa no produjo resultados exactos para el XOR lógico.
- ★ A partir de 3 perceptrones por capa, llegamos a la condición de corte por cota de error y no por cantidad límite de iteraciones.
  - ○ Mayor cantidad de capas y perceptrones logran llegar a la solución en menos iteraciones.
- ★ El método Adam logra llegar a la cota de error en menos iteraciones que el gradiente descendente.

Ejercicio 3b

Ejercicio 3b

Ejemplo de entrada:

```
0 1 1 1 0
1 0 0 0 1
1 0 0 1 1
1 0 1 0 1
1 1 0 0 1
1 0 0 0 1
0 1 1 1 0
```

PAR

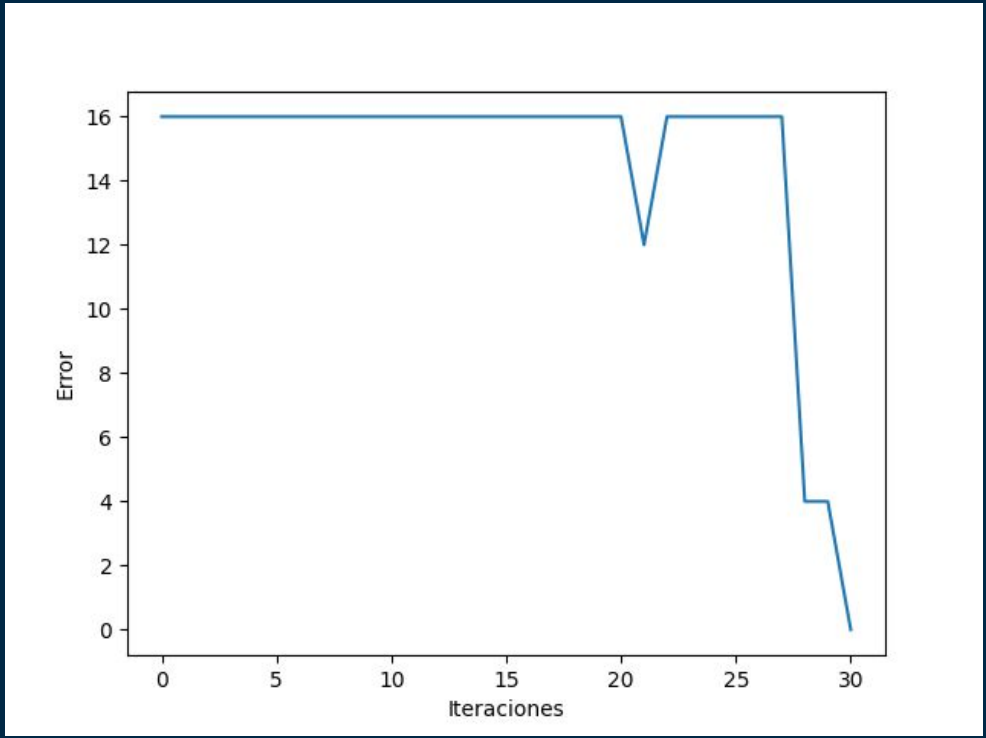
```
0 0 1 0 0
0 1 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 1 1 1 0
```

IMPAR

Consideraciones

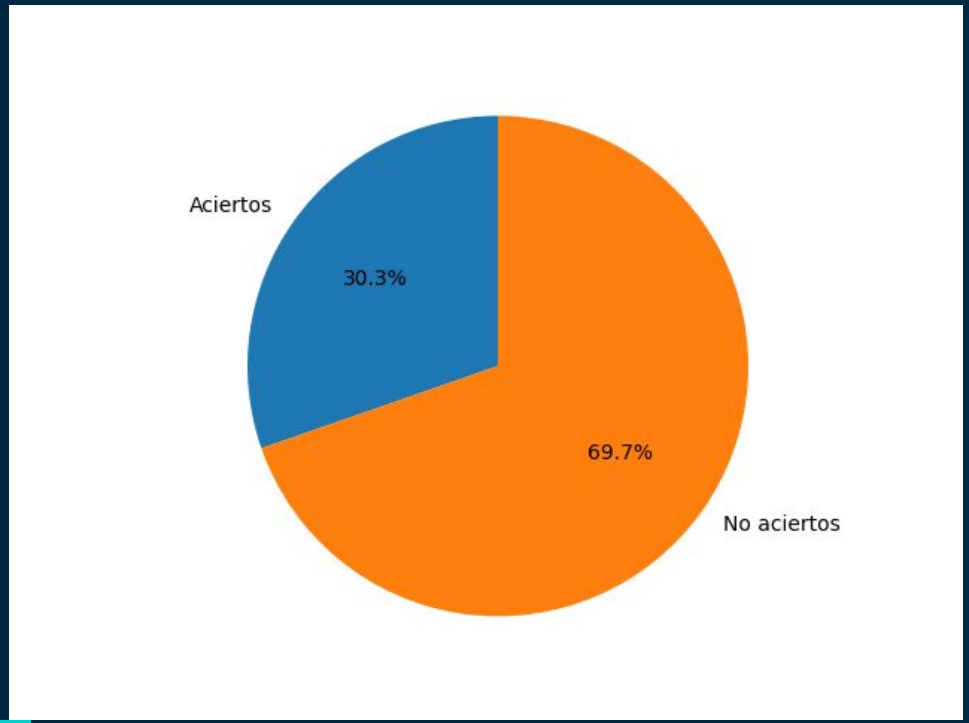
- ★ Se utiliza el método de activación Sigmoid.
- ★ Gradiente descendente.
- ★ Tasa de aprendizaje = 0.1
- ★ Epsilon de error = 0.1
- ★ Entrada: 35 unos o ceros
- ★ Perceptrones por capa: [36, 36, 1]

Error vs épocas de entrenamiento



- Para un entrenamiento, puede verse que el error de entrenamiento es nulo, obteniendo una solución exacta.

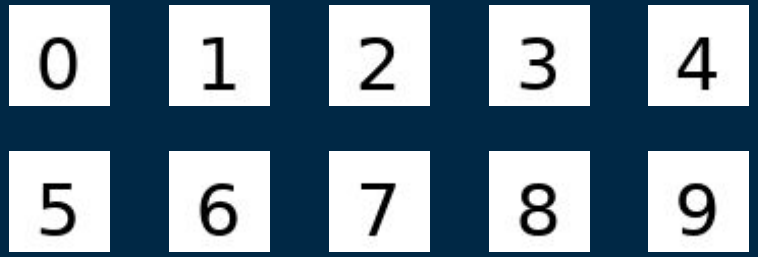
Probabilidad de acierto



- Se realizan 1000 entrenamiento con 9 números aleatorios
- Se cuenta la cantidad de veces que acierta en el elemento restante

Ejercicio 3c

Conjunto de entrenamiento



Conjunto de prueba

100 Imágenes por cada nivel de ruido:

Nivel 60: 

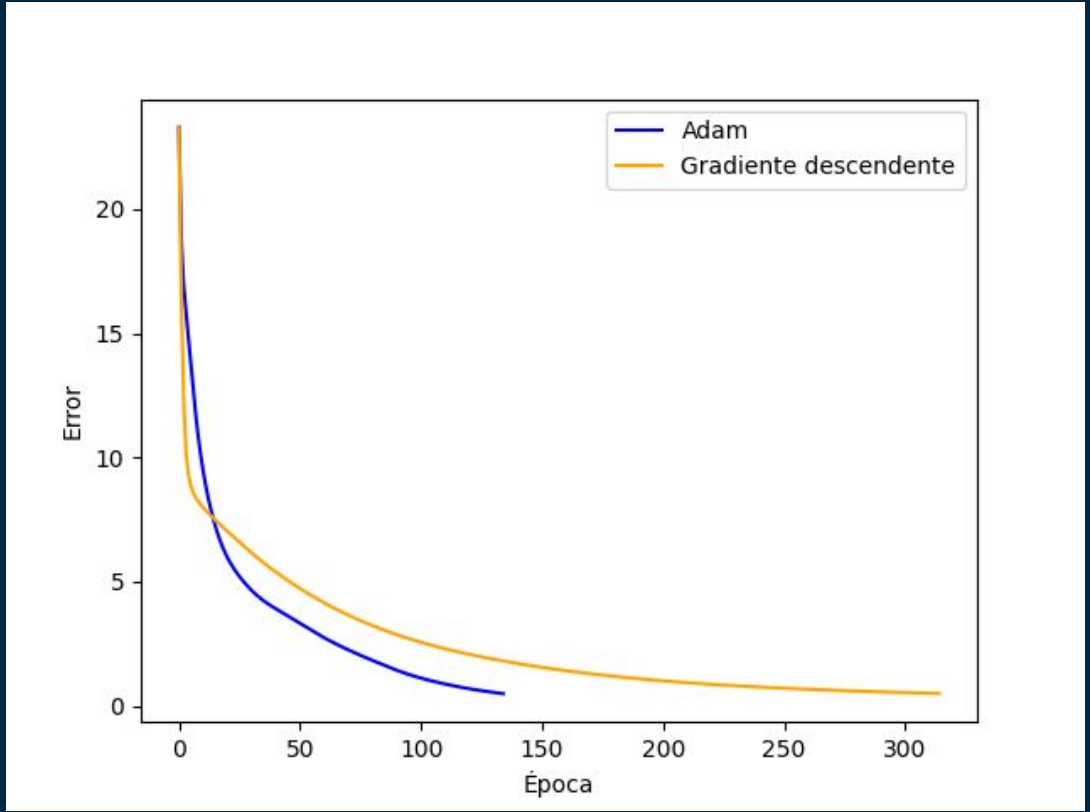
Nivel 160: 

Nivel 255: 

Consideraciones

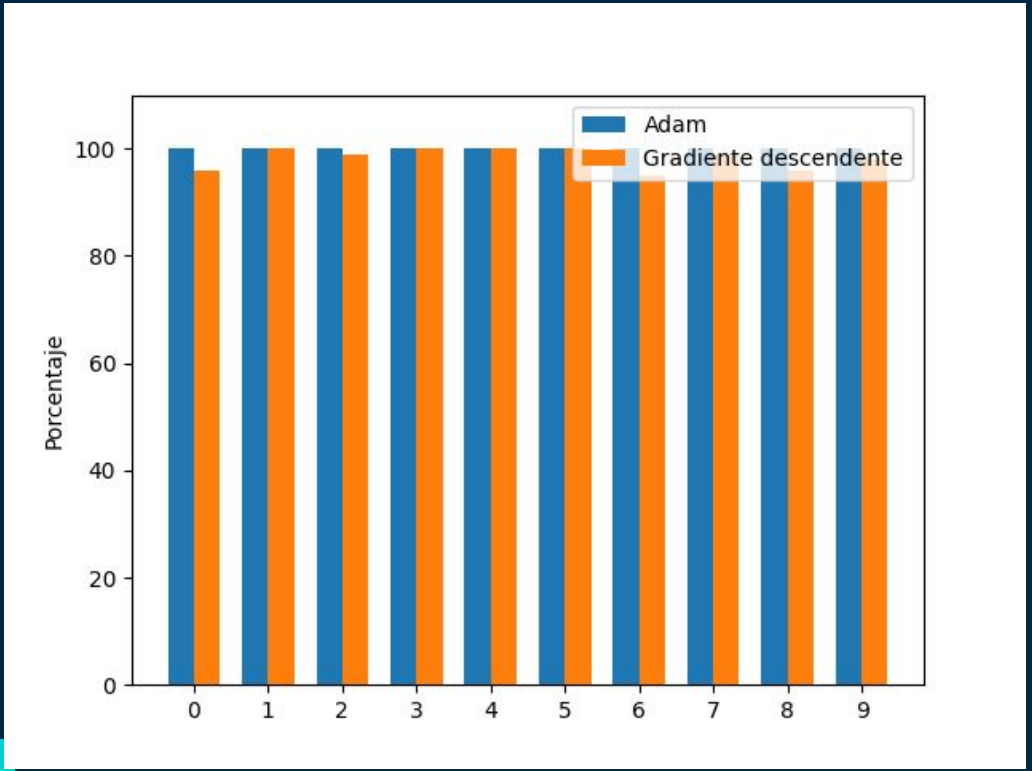
- ★ Se utiliza el método de activación sigmoid
- ★ ADAM => Tasa de aprendizaje = 0.001
- ★ Gradiente Descendiente => Tasa de aprendizaje = 0.1
- ★ Epsilon de error = 0.5
- ★ Entrada: 2500 pixeles
- ★ Perceptrones por capas: 50, 20, 10

Error vs Época según el método



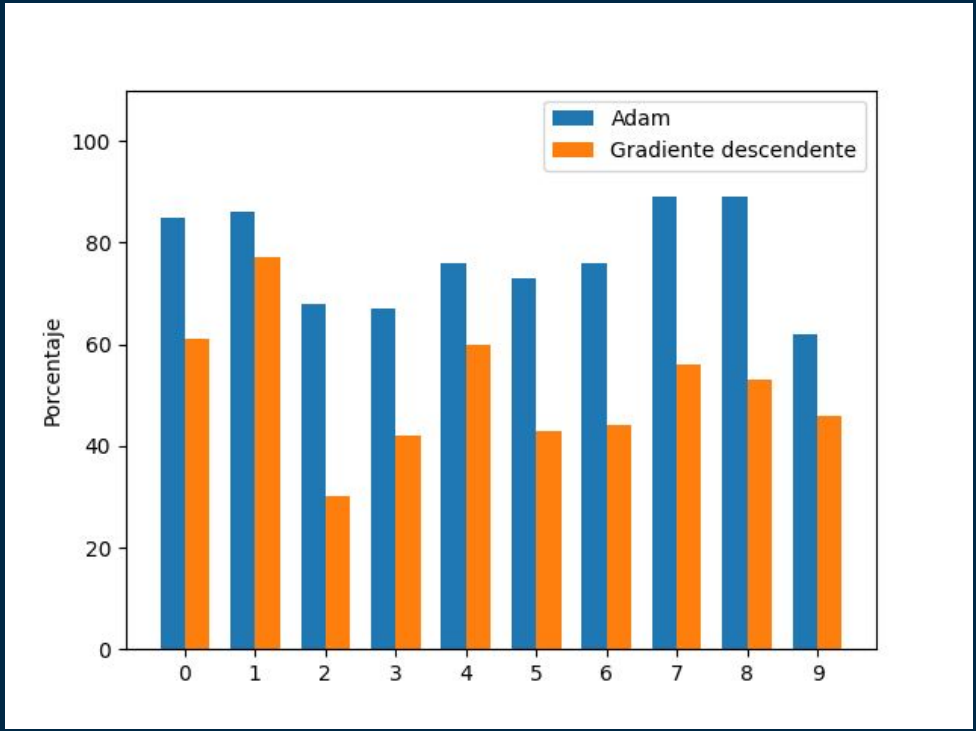
- Nivel de ruido: 160

Porcentaje de aciertos para cada Nro.



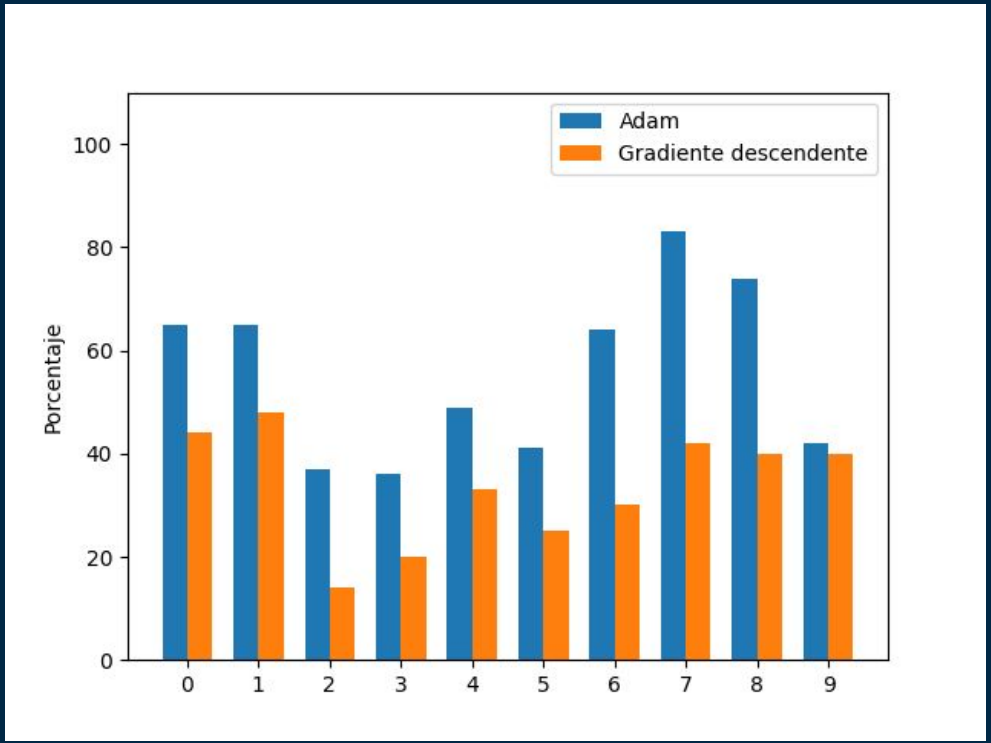
- Nivel de ruido = 70

Porcentaje de aciertos para cada Nro.



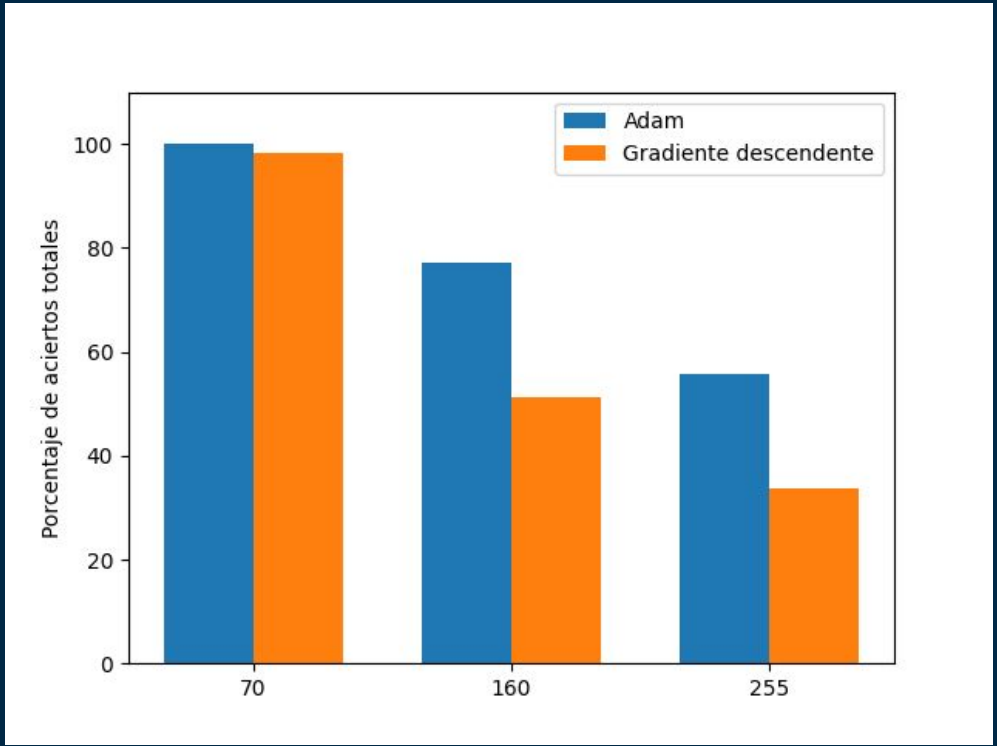
- Nivel de ruido = 160

Porcentaje de aciertos para cada Nro.

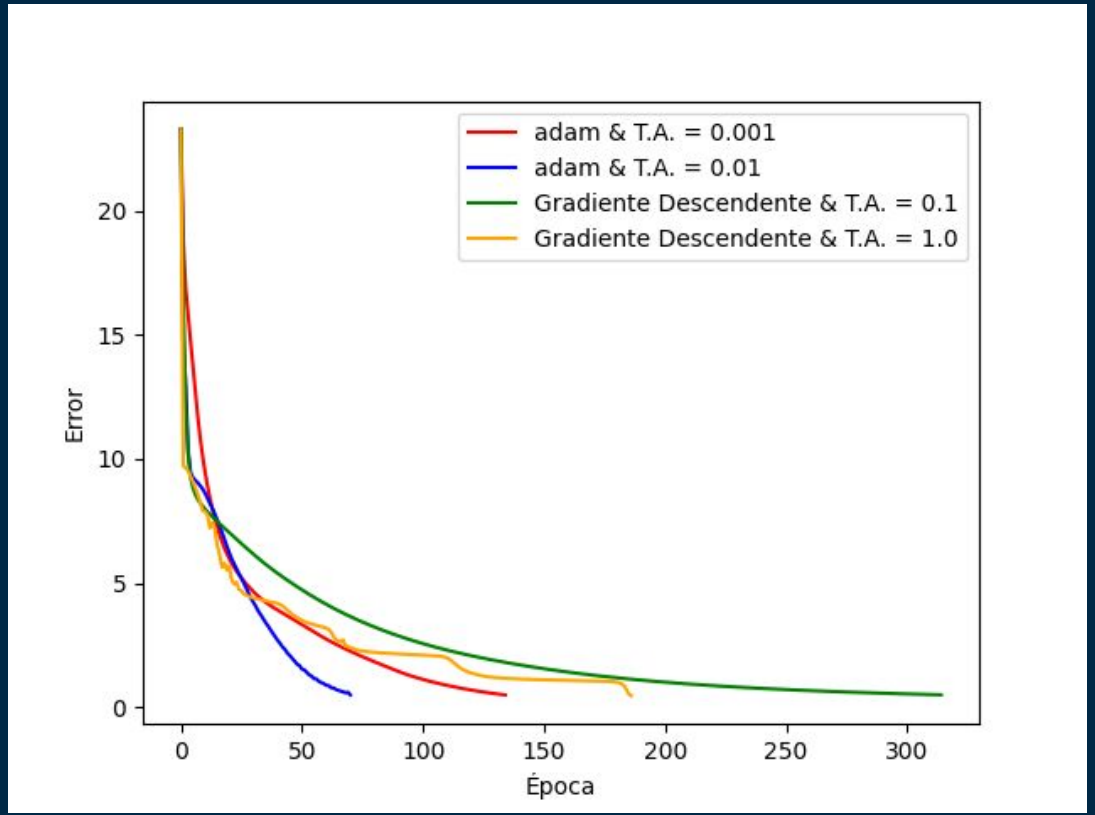


- Nivel de ruido = 255

Porcentaje de aciertos según ruido

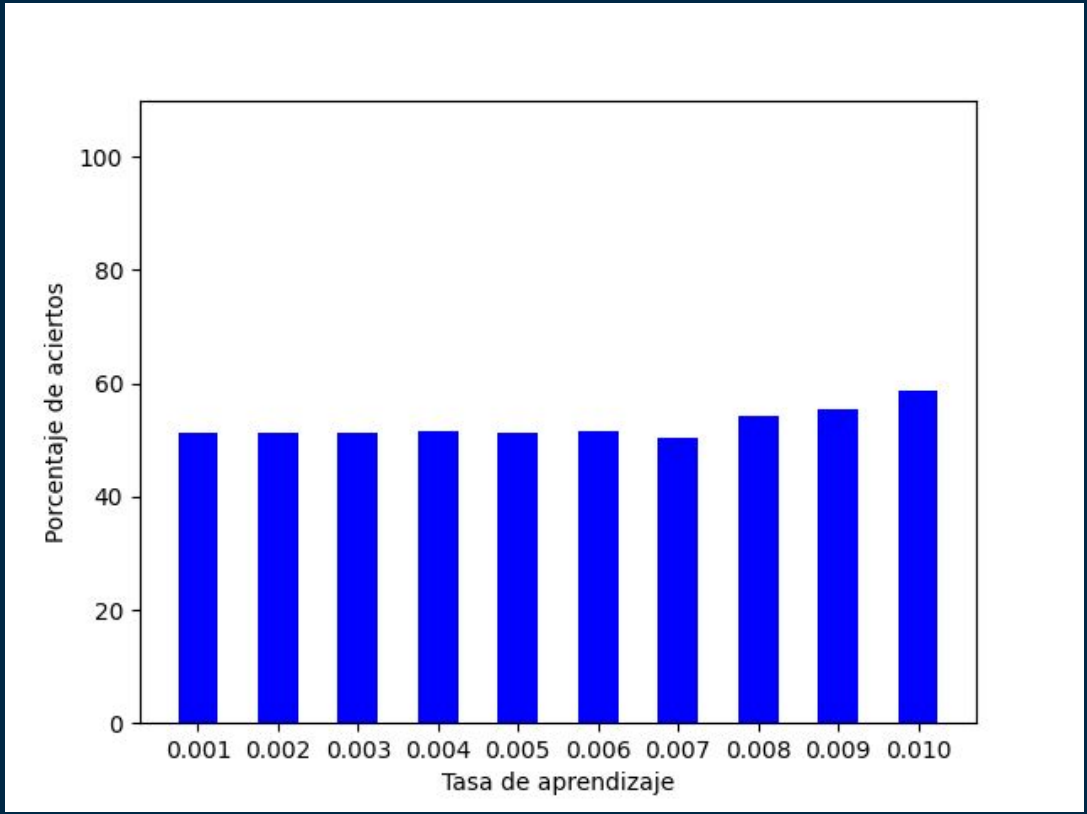


Error vs Época según el método y T. A.



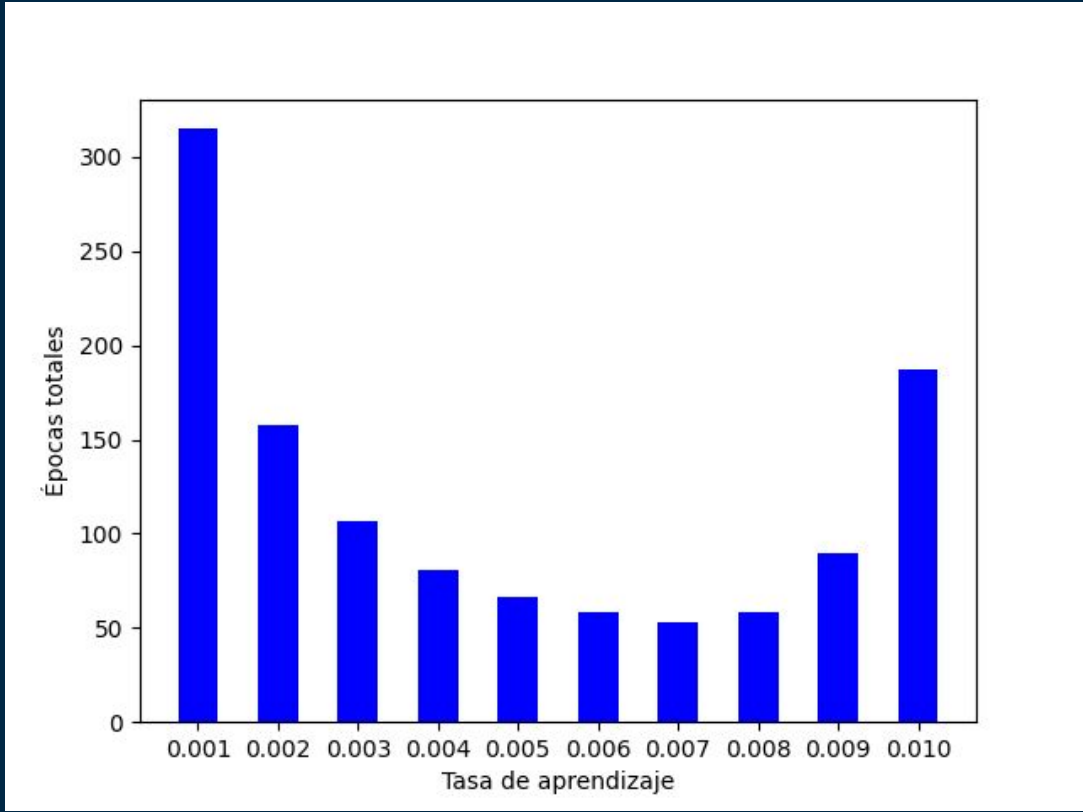
- Nivel de ruido: 160

Porcentaje de aciertos según T. A.



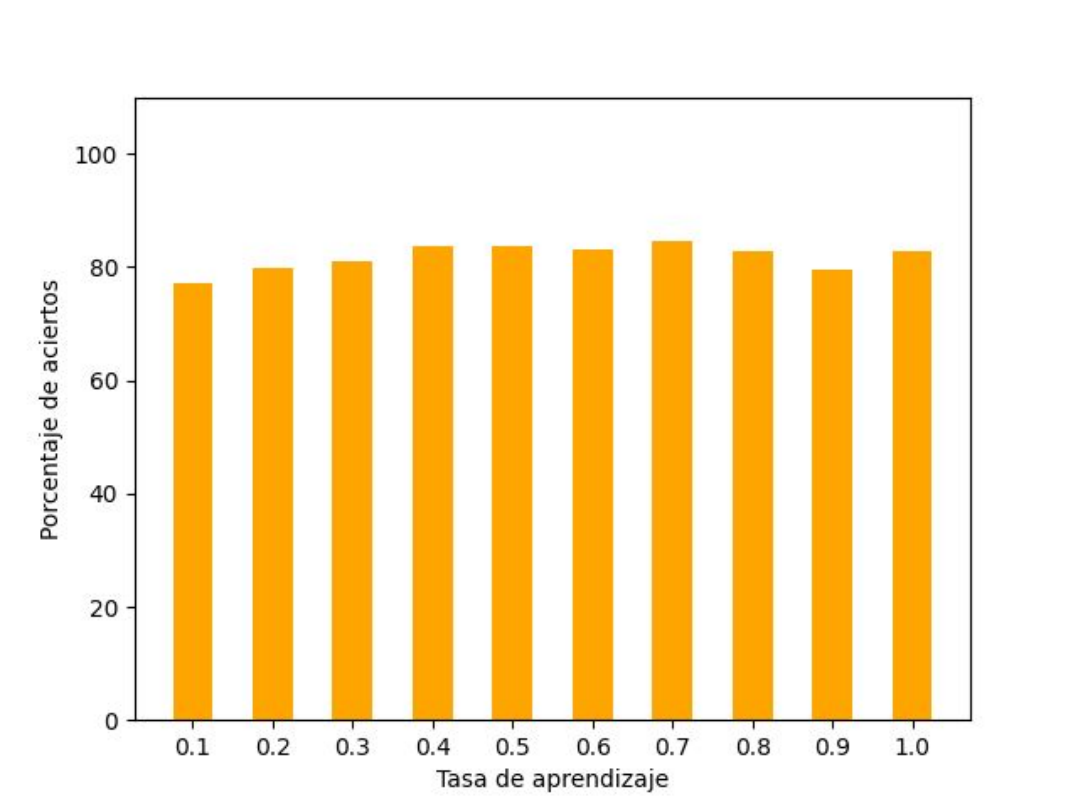
- Nivel de ruido: 160
- ADAM

Épocas totales según T. A.



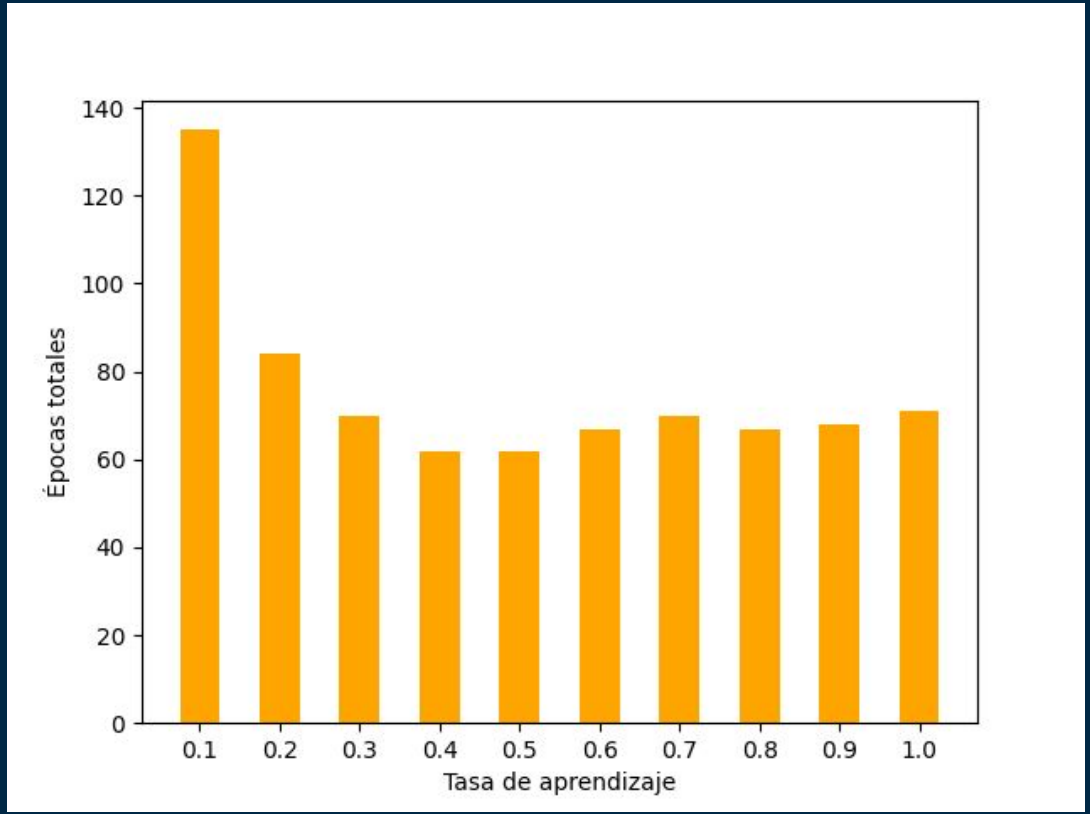
- Nivel de ruido: 160
- ADAM

Porcentaje de aciertos según T. A.



- Nivel de ruido: 160
- Gradiente descendente

Épocas totales según T. A.



- Nivel de ruido: 160
- Gradiente descendente

Conclusiones:

- A mayor ruido, peores resultados de efectividad para la generalización.
- En este caso, ADAM tuvo mejor desempeño que Gradiente Descendiente

- La tasa de aprendizaje no afecta en el porcentaje de aciertos pero si impacta en la cantidad de épocas.