



# PERCEPTRON SIMPLE Y MULTICAPA

TP3  
SISTEMAS DE INTELIGENCIA ARTIFICIAL  
1ER CUATRIMESTRE 2021



Grupo 3:  
Gonzalo Hirsch - 59089  
Florencia Petrikovich - 58637  
Juan Martin Oliva - 58664





01

## RESUMEN

Resumen del trabajo realizado durante el TP



02

## ENFOQUE

Descripción de cómo se construyó el código y cómo enfocamos el trabajo

03

## RESULTADOS

Resultados obtenidos en diferentes pruebas

04

## CONCLUSIONES

Conclusiones a partir de los resultados

# RESUMEN

01

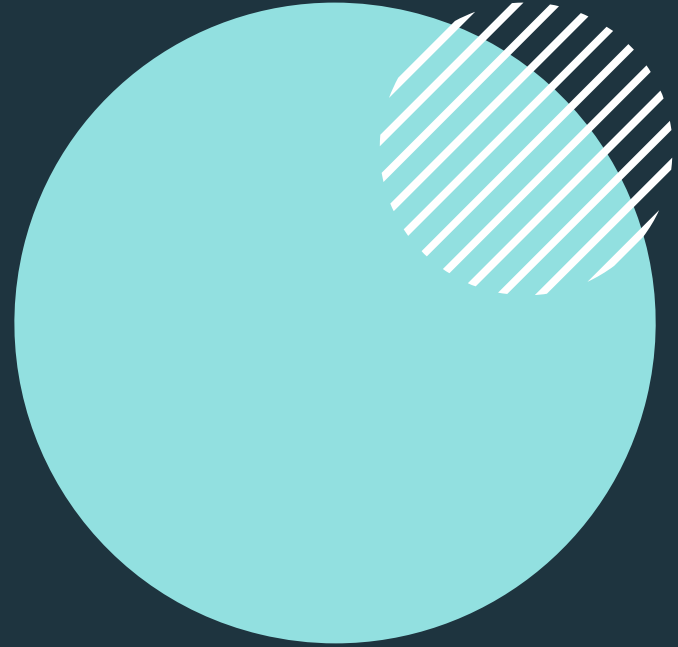
# RESUMEN DEL TRABAJO

## Implementación

Implementación de un motor de **perceptrones simples y multicapa** para **aprender** diferentes tipos de problemas.

## Problemas

Los **problemas** resueltos eran **linealmente y no linealmente separables**.

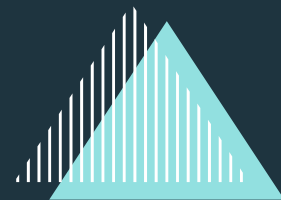




# ENFOQUE



02



# METODOLOGÍA



# DECISIONES DE DISEÑO

---

## ARQUITECTURA

Invertir más tiempo en diseñar arquitectura.

---

## INICIALIZACIÓN DE PESOS

Se construyeron **clases** capaces de ser **combinadas fácilmente** en una **red**.

---

## OPTIMIZACIONES APLICADAS

---

Se **abstrajo** el **problema** para poder **resolver todos los problemas** al mismo tiempo.

# DECISIONES DE DISEÑO

---

ARQUITECTURA

---

■  
■  
■  
INICIALIZACIÓN DE PESOS

---

OPTIMIZACIONES APLICADAS

---

Iniciar los pesos en valores pequeños ⇒  
Mejores resultados<sup>1</sup>

Fórmula usada (variación del link):

$$\text{random}(0, 1, N) \cdot \sqrt{\frac{1}{N}}$$

N = Cantidad de Pesos

random(0,1,N) = vector de N lugares random uniforme entre 0 y 1



# DECISIONES DE DISEÑO

---

ARQUITECTURA

---

■ ■ ■

---

INICIALIZACIÓN DE PESOS

---

■ ■ ■

---

OPTIMIZACIONES APLICADAS

---

Momentum ⇒ Tiene un gran impacto en la velocidad para encontrar una solución

# RESULTADOS

03



# EJERCICIO

# 1



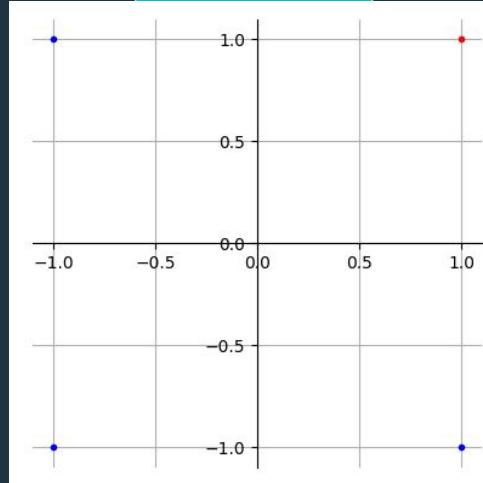
# PROBLEMA 1 - AND

Tipo → Linealmente Separable

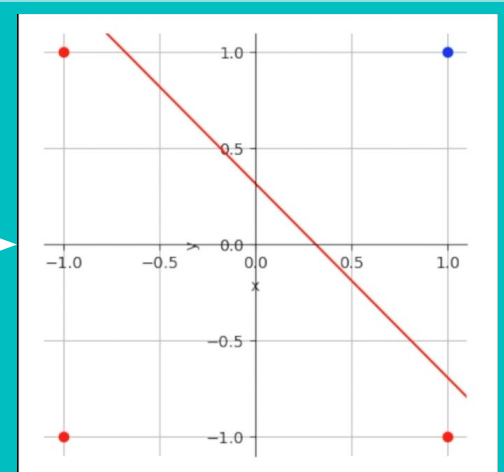
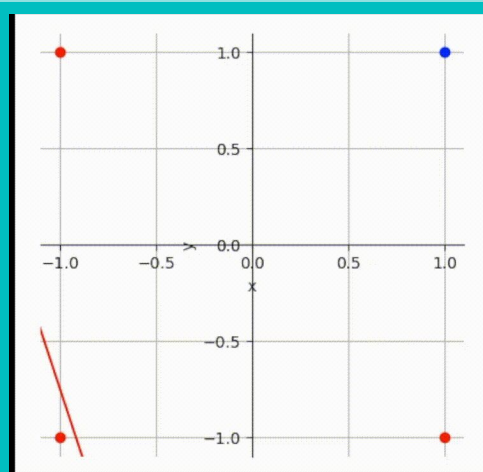
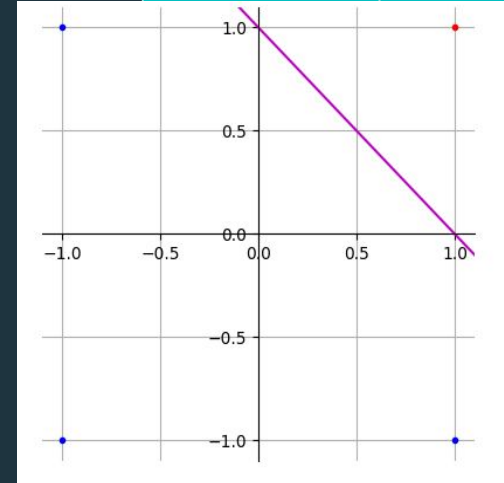
¿Solucionable? → Si por Perceptrón Simple

Solución Alcanzada

Distribución



Solución



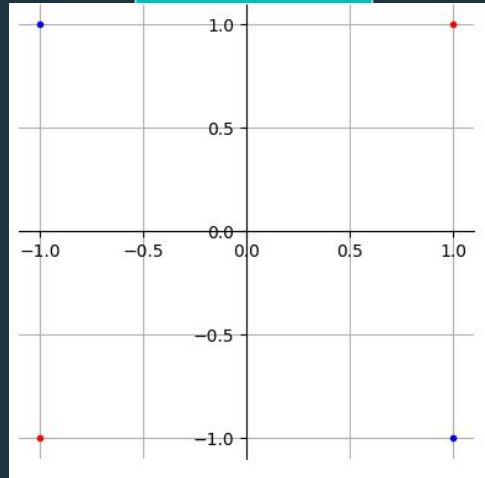
# PROBLEMA 2 - XOR

Tipo → No Linealmente Separable

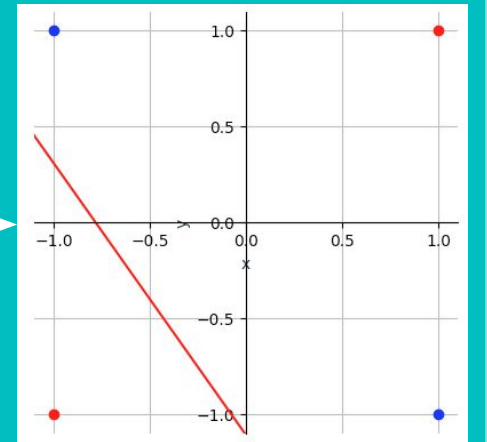
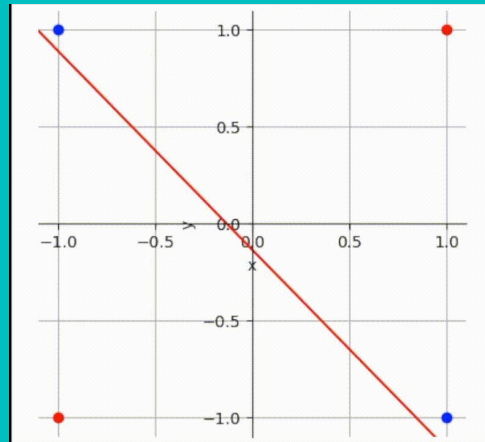
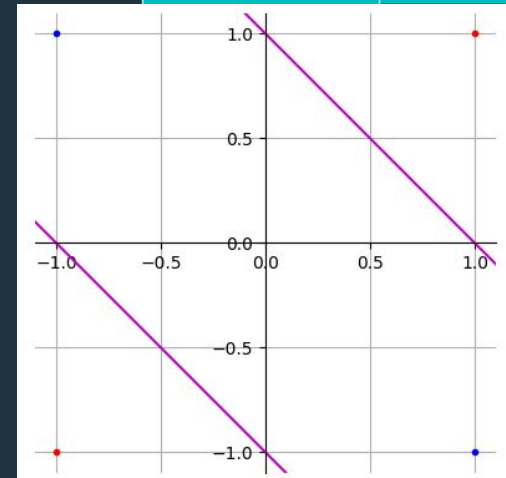
¿Solucionable? → No por Perceptrón Simple

Solución Alcanzada

Distribución



Solución





# EJERCICIO

# 2



# MOMENTUM

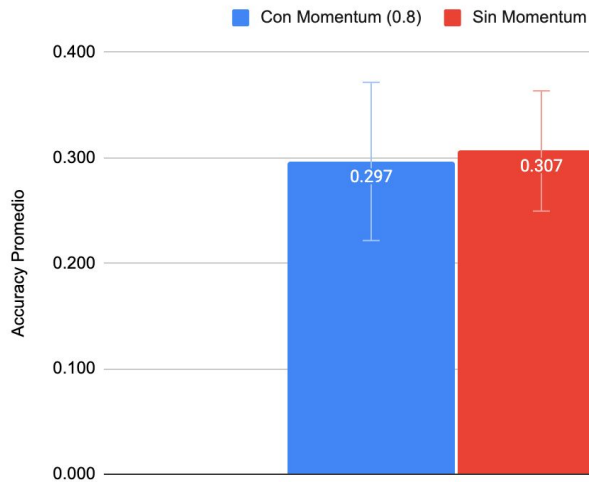
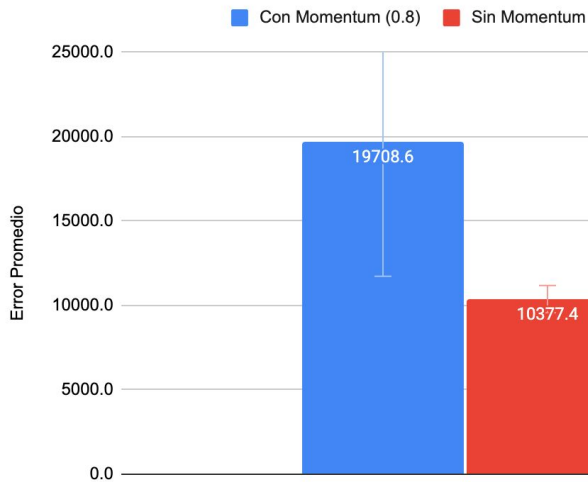
Pruebas para determinar si era mejor activar **momentum** o no

¡El error es casi el **doble**!

En todos los casos usa **todo** el dataset

Usan la **Activación Lineal**

**Delta** de Accuracy es 5



NO  
Momentum

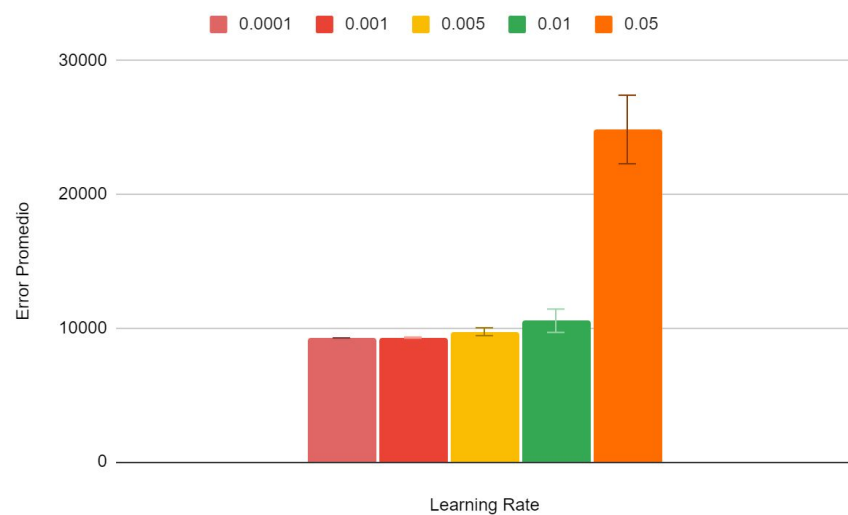
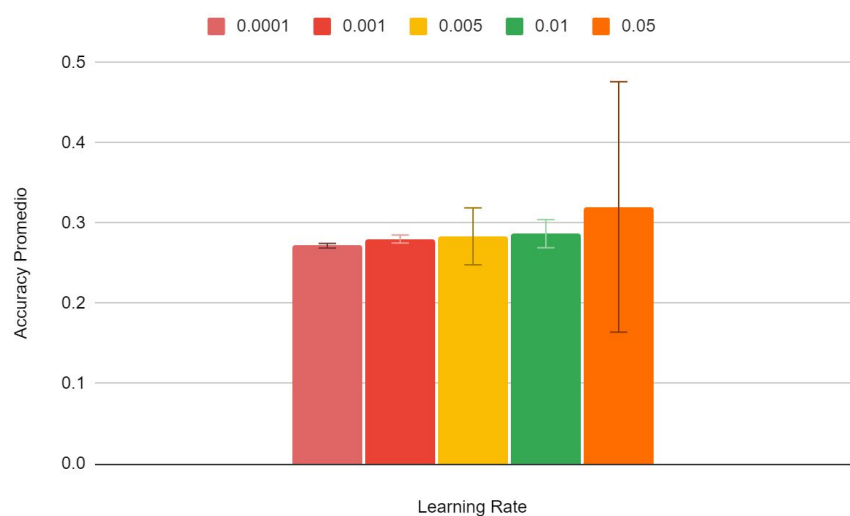
# LEARNING RATE

TODOS LOS DATOS

Análisis de Mejor  
Learning Rate

En todos los casos  
usa **todo** el  
**dataset**

Mejores métricas:  
0.001, 0.01 y 0.1





# APRENDIZAJE

27.3%

Accuracy Activación Lineal (Delta = 5) (Desvío: 0.0076)

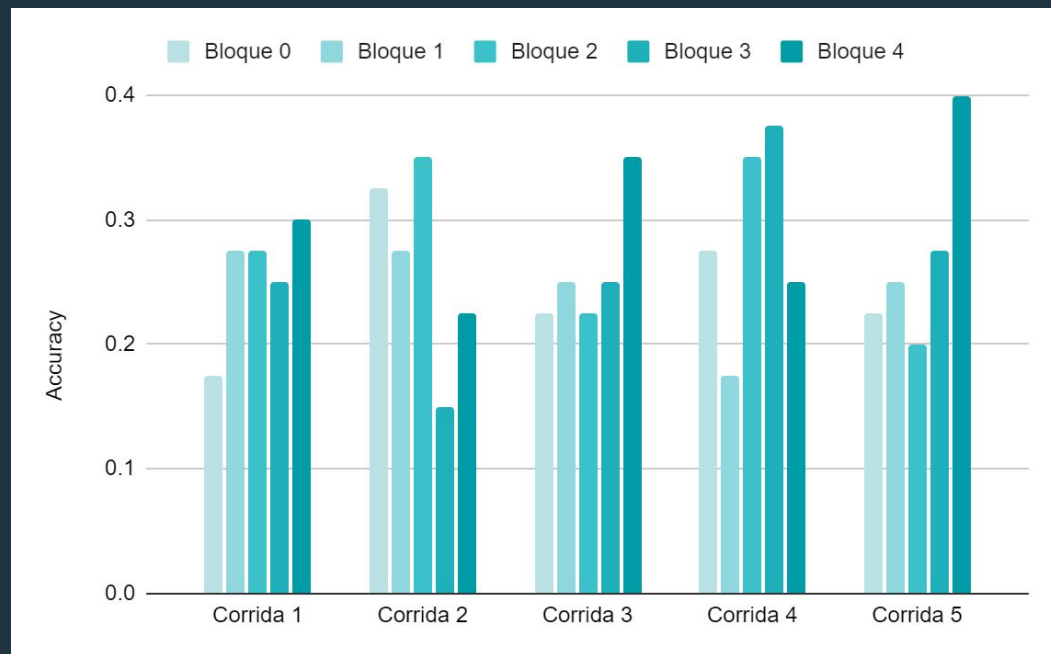


3.1%

Accuracy Activación No Lineal (Delta = 0.01)  
(Desvío: 0.0042)

# VALIDACIÓN CRUZADA - ACTIVACIÓN LINEAL

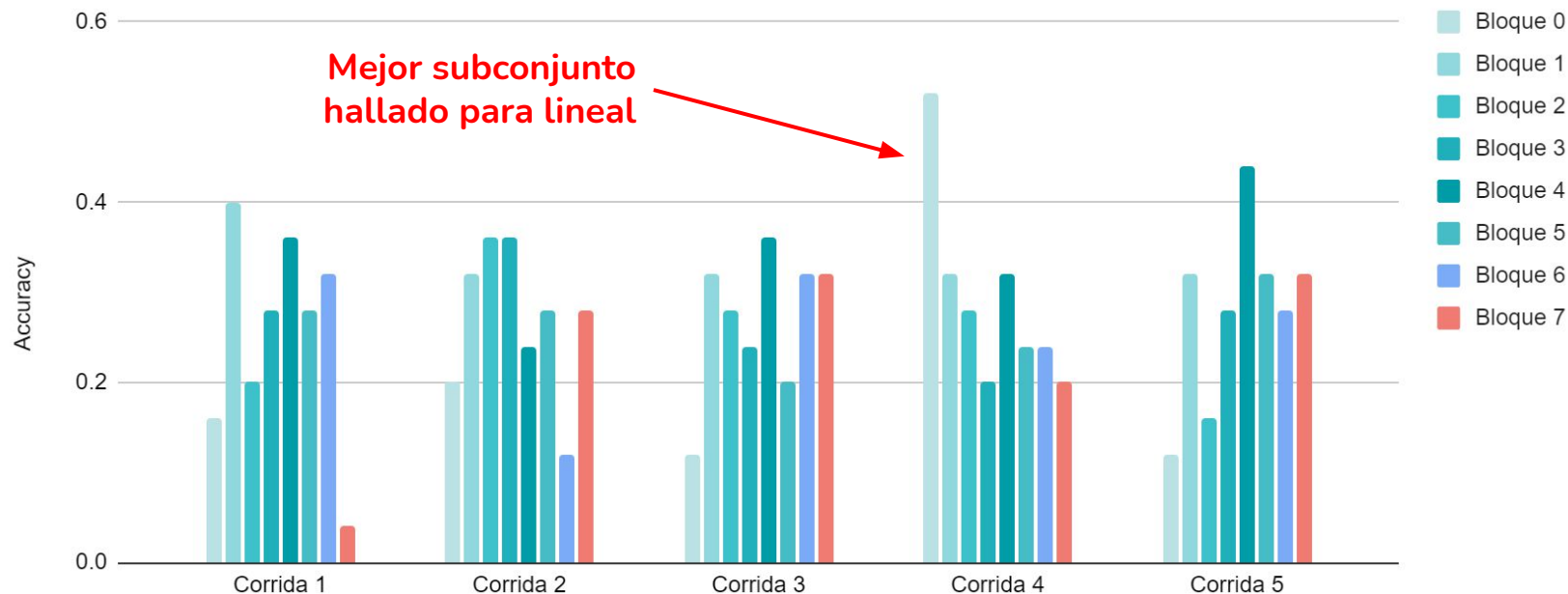
K = 5, 40 DATOS POR BLOQUE



**Randomización del orden  
de los datos antes de cada  
corrida**

# VALIDACIÓN CRUZADA - ACTIVACIÓN LINEAL

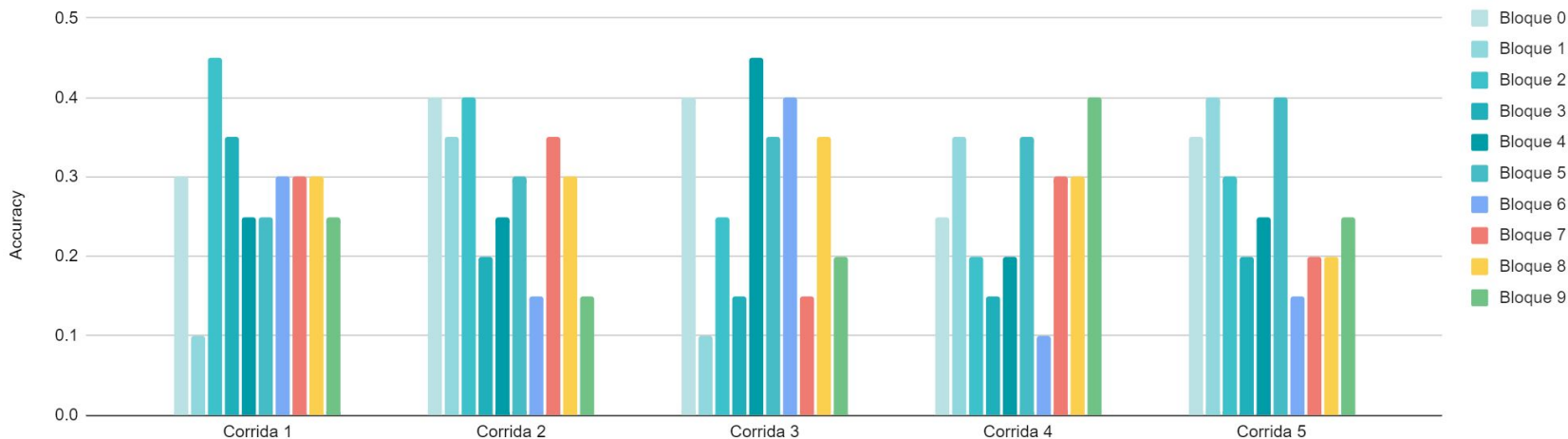
K = 8, 25 DATOS POR BLOQUE



# VALIDACIÓN CRUZADA - ACTIVACIÓN LINEAL

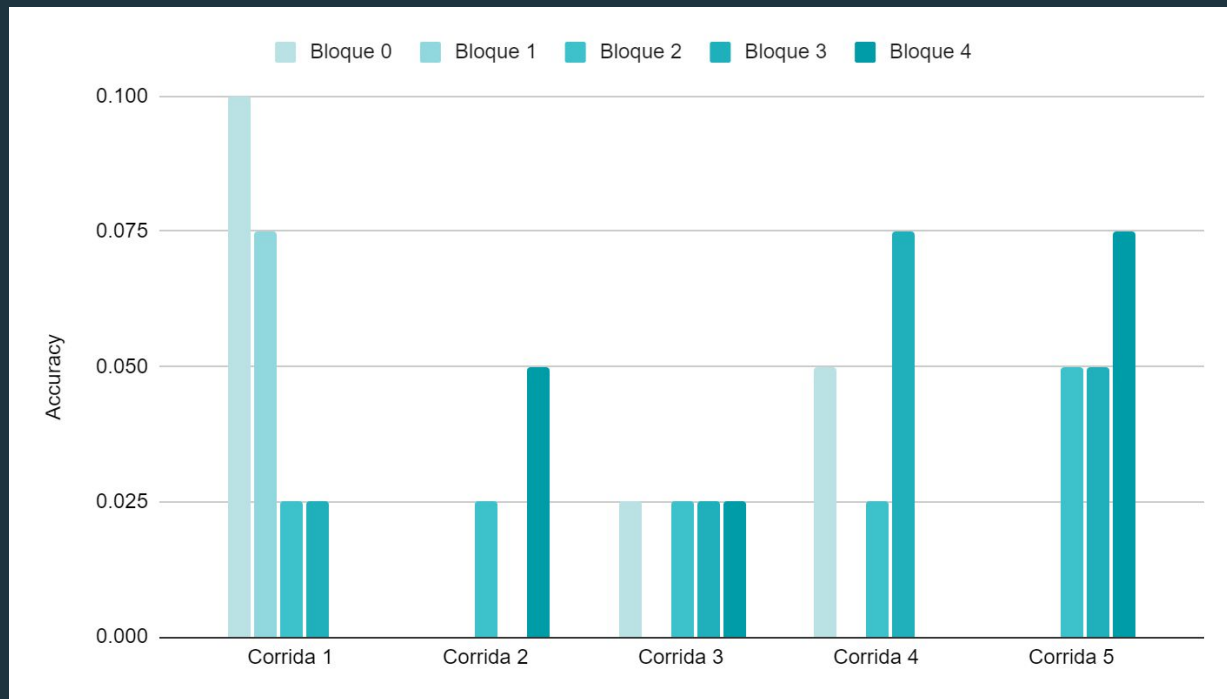
K = 10, 20 DATOS POR BLOQUE

Para cada corrida, siempre hay ciertos bloques que al excluirlos del entrenamiento y usarlos en las pruebas disminuye la capacidad de generalización.



# VALIDACIÓN CRUZADA - ACTIVACIÓN NO LINEAL

K = 5, 40 DATOS POR BLOQUE

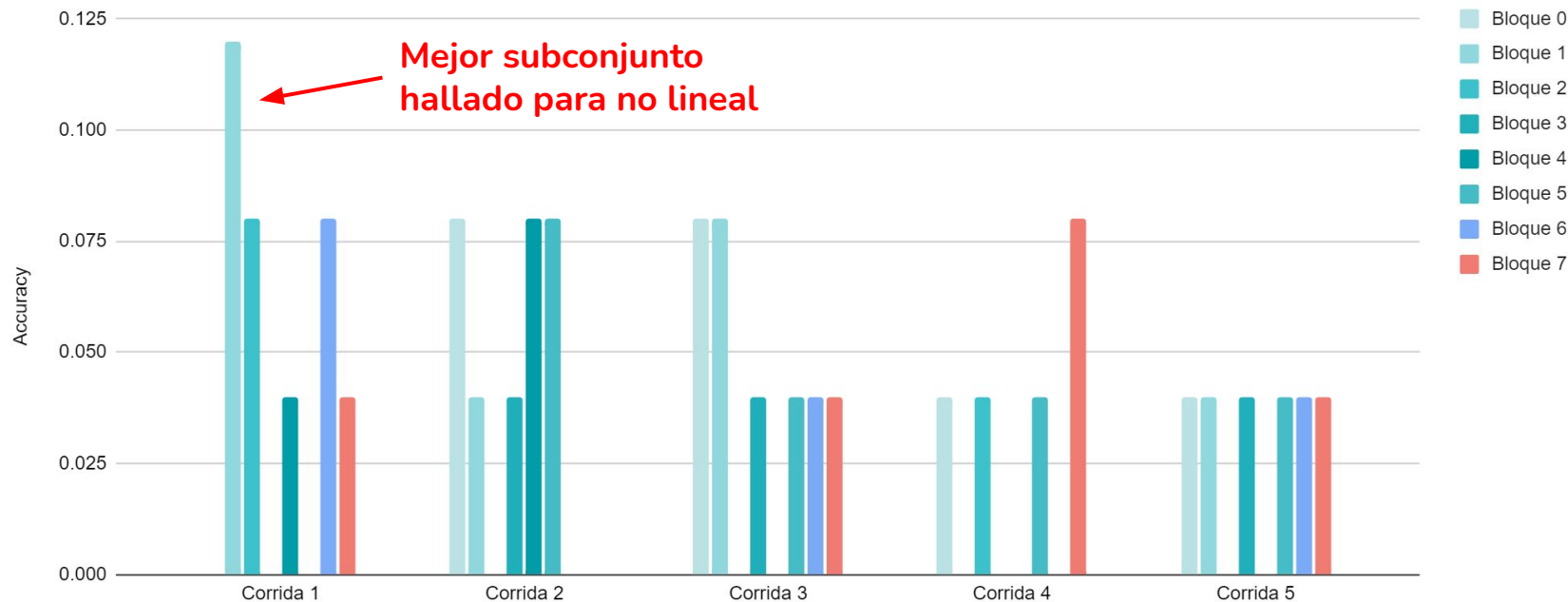


En mismas corridas, la elección de bloques afecta mucho el accuracy de la generalización.

En varios casos, el accuracy del testeo supera al del entrenamiento.

# VALIDACIÓN CRUZADA - ACTIVACIÓN NO LINEAL

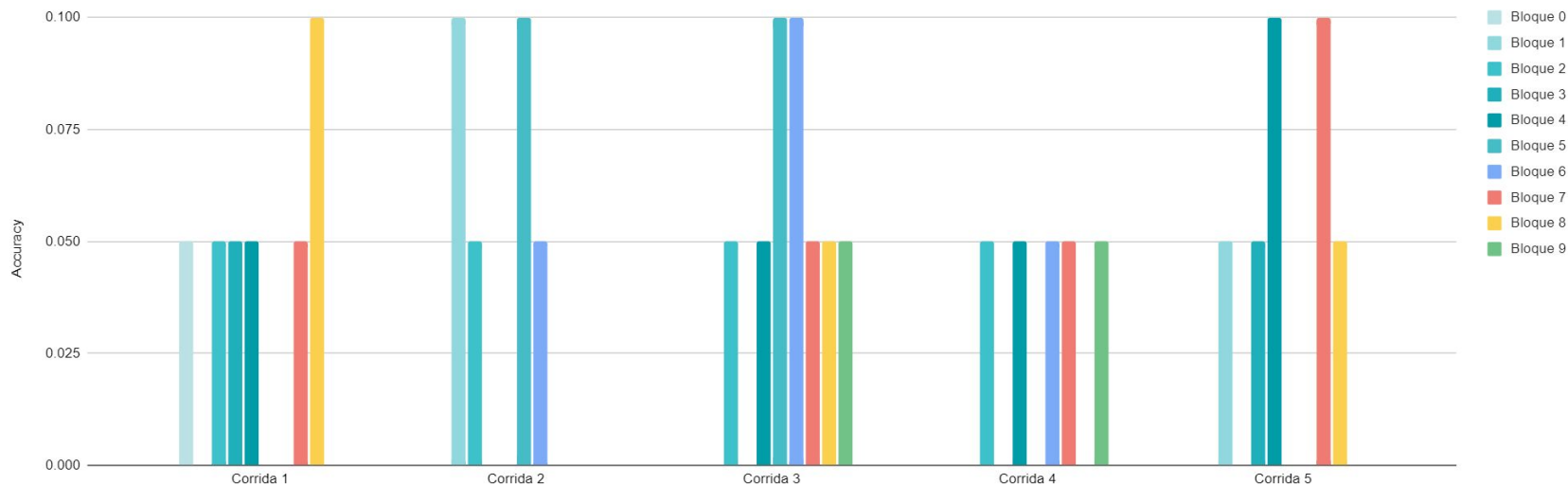
K = 8, 25 DATOS POR BLOQUE



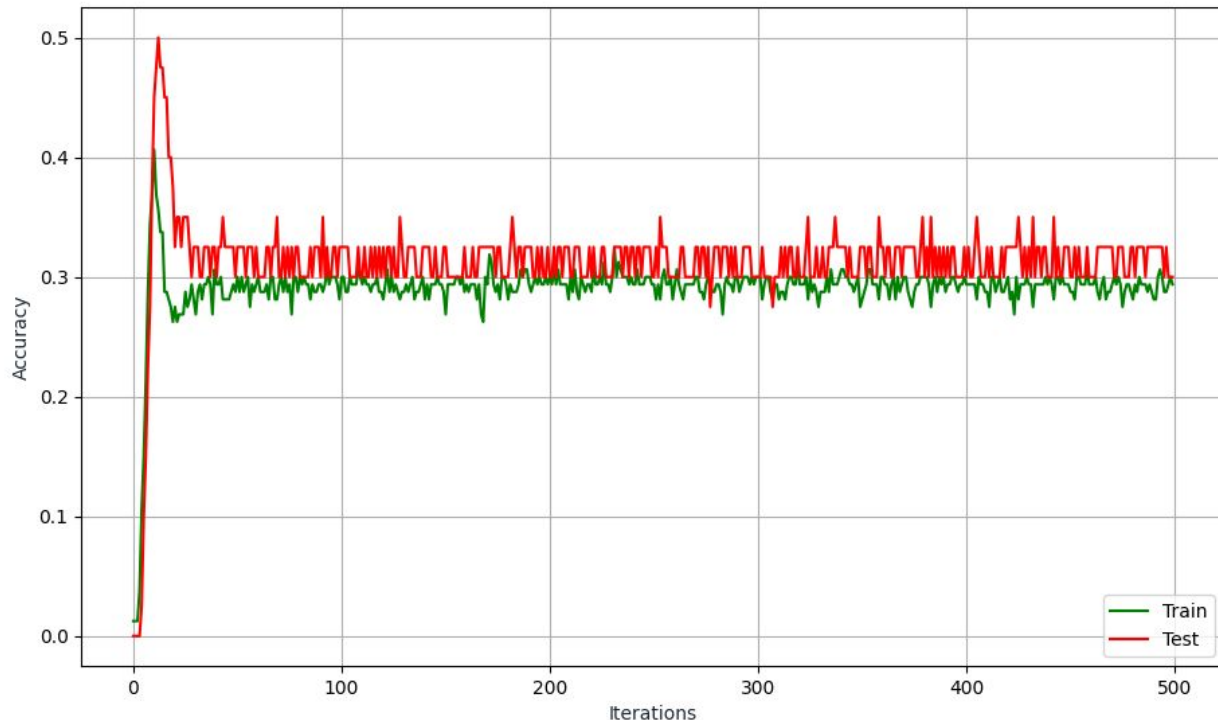
# VALIDACIÓN CRUZADA - ACTIVACIÓN NO LINEAL

K = 10, 20 DATOS POR BLOQUE

Randomización del orden de los datos antes de cada corrida



# APRENDIZAJE Y GENERALIZACIÓN

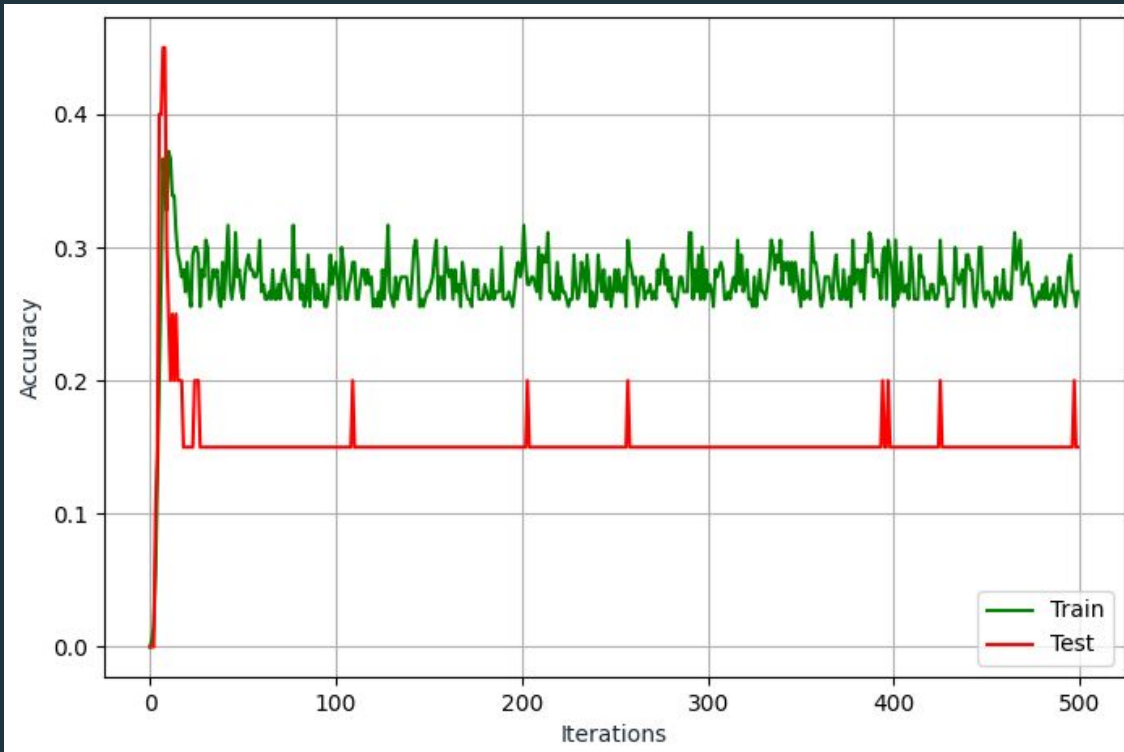


Train:  $\frac{1}{3}$  del dataset  
Test:  $\frac{1}{3}$  del dataset  
Learning Rate: 0.001

Puede **generalizar bien** el Perceptrón, tiene accuracy similar al entrenamiento



# APRENDIZAJE Y GENERALIZACIÓN



Train: 9/10 del dataset  
Test: 1/10 del dataset  
Learning Rate: 0.001

Llega a **overfitting** el perceptrón, tiene mucha mejor accuracy en entrenamiento

# CONCLUSIONES

## APRENDIZAJE

Depende mucho del dataset,  
en este caso parece tener  
mucho ruido



Activación Lineal



Activación No Lineal

## GENERALIZACIÓN

El Perceptrón puede  
generalizar acorde a su  
entrenamiento (depende de  
subconjunto)

## ENTRENAMIENTO

Validación Cruzada para ver  
mejores subconjuntos +  
pruebas de distintos  
tamaños



# EJERCICIO

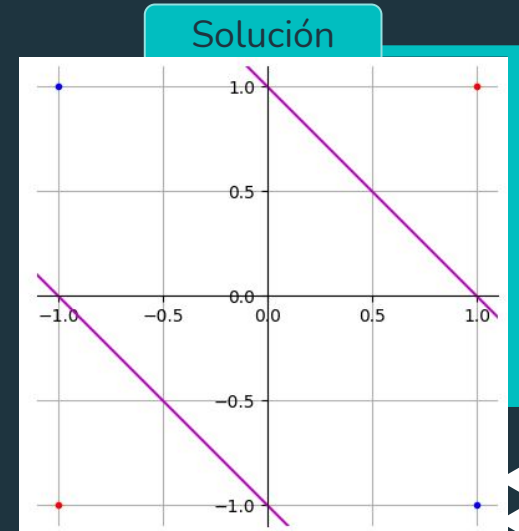
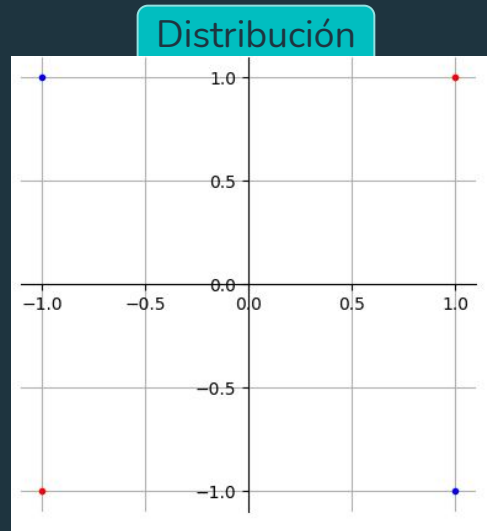
# 3



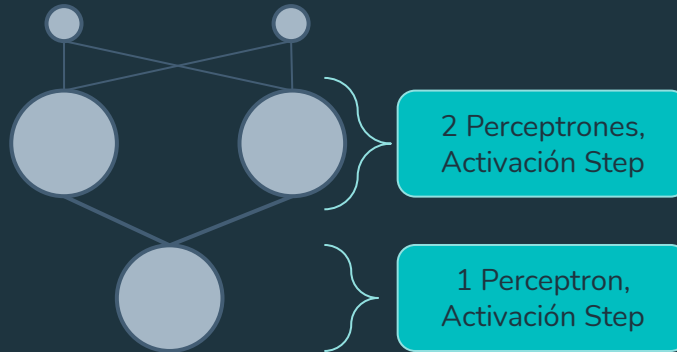
# PROBLEMA 1 - XOR

Tipo → No Linealmente Separable

¿Solucionable? → Si por Perceptrón Multicapa



Configuración Usada



Momentum  
( $\alpha=0.8$ )

Learning Rate  
0.01

# PROBLEMA 2 - PARIDAD DE NÚMEROS



## LEARNING RATE MOMENTUM

Efectos de la tasa de aprendizaje y momentum sobre las iteraciones.



## CONFIGURACIÓN

Efectos de la configuración de la red sobre una subdivisión del conjunto de datos.



## VALIDACIÓN CRUZADA

Efectos del dataset de entrenamiento y testeo sobre la cap de generalizacion.

# MOMENTUM

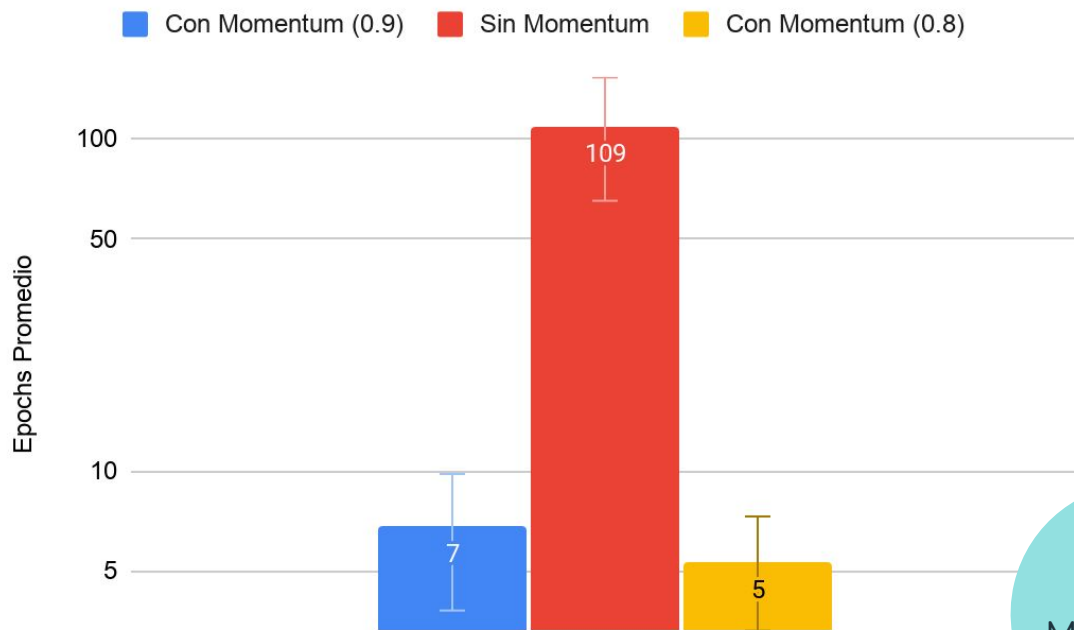
Pruebas para determinar si era mejor activar **momentum** o no

En todos los casos usa **todo el dataset**

Problema 2 -  
Clasificación de  
Números

Usa **Activación Step**

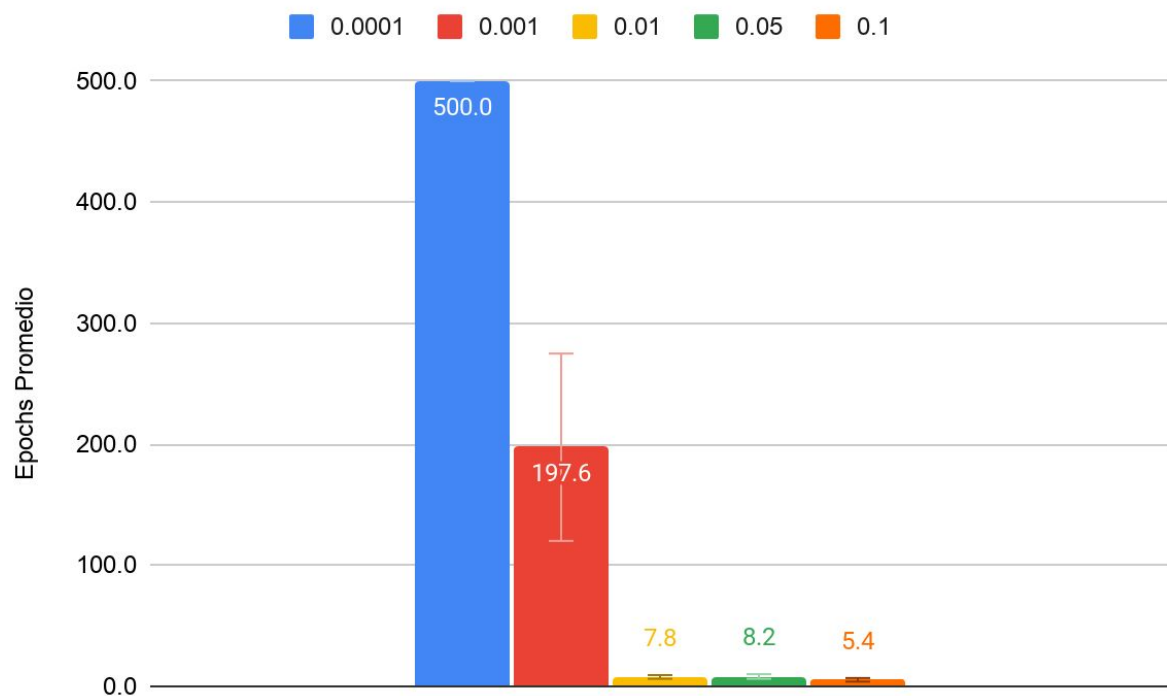
¡Tarda hasta **15.5**  
**veces** menos!



SI  
Momentum  
0.8

# LEARNING RATE

TODOS LOS DATOS



Análisis de Mejor  
Learning Rate

En todos los casos  
usa **todo** el  
**dataset**

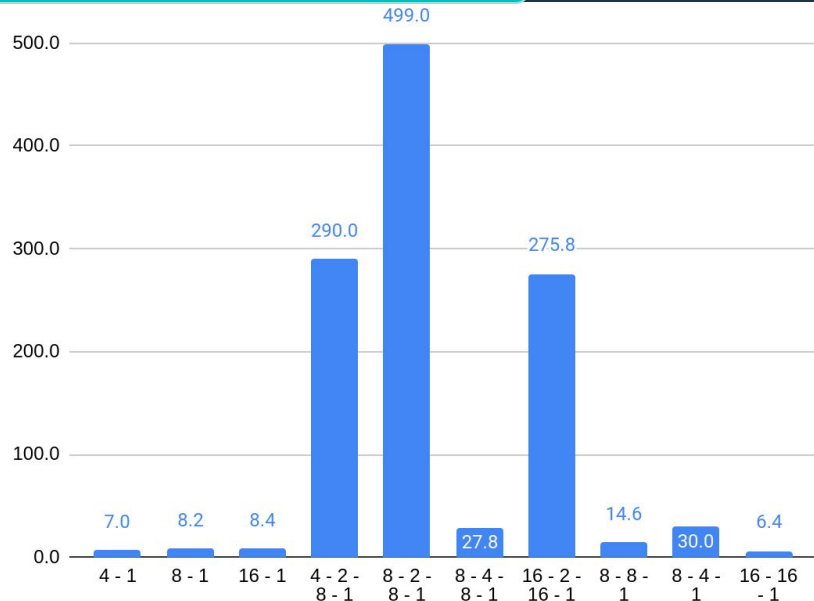
Mejores métricas:  
**0.01, 0.05 y 0.1**

# ANÁLISIS DE LA CONFIGURACIÓN

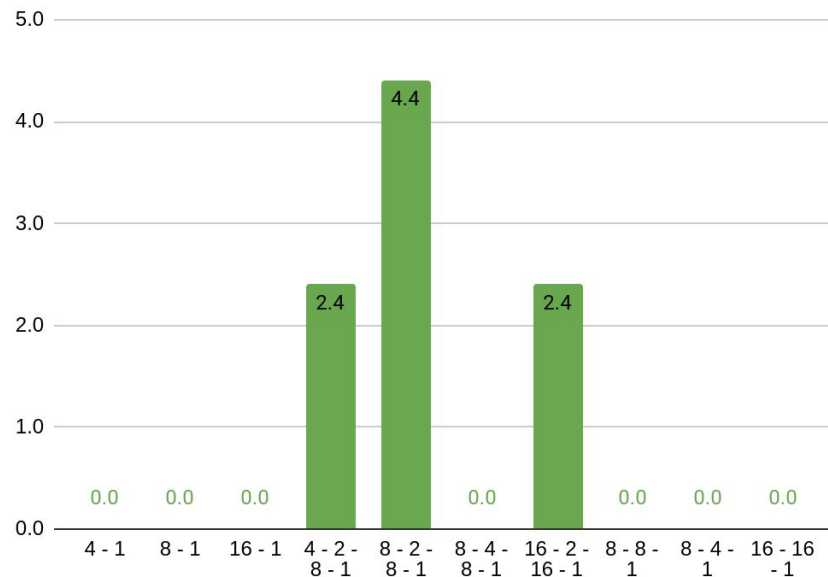
## DATOS DE ENTRENAMIENTO

Mitad del dataset  
para **entrenar** y la  
otra para **pruebas**

### Épocas por configuración



### Error por configuración



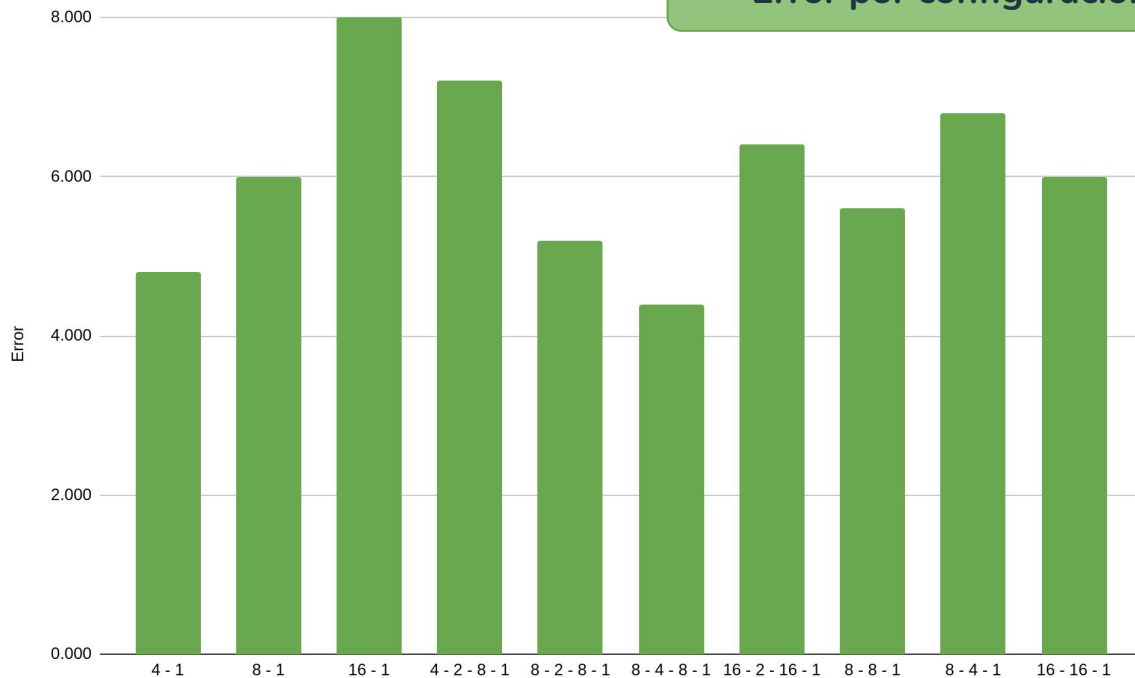


# ANÁLISIS DE LA CONFIGURACIÓN

## DATOS DE PRUEBAS

Mitad del dataset  
para **entrenar** y la  
otra para **pruebas**

Error por configuración



**MENOR ERROR**

Configuraciones:

8 - 4 - 8 - 1

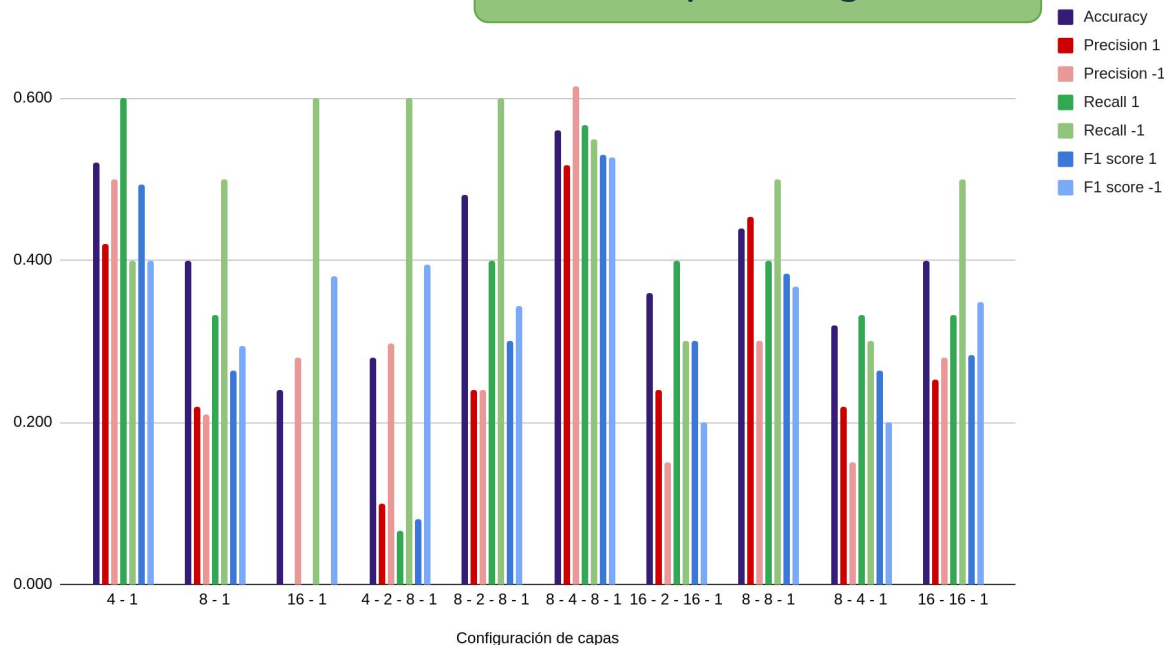
4 - 1

# ANÁLISIS DE LA CONFIGURACIÓN

## DATOS DE PRUEBAS

Mitad del dataset  
para **entrenar** y la  
otra para **pruebas**

Métricas por configuración



MEJORES  
MÉTRICAS

Configuraciones:

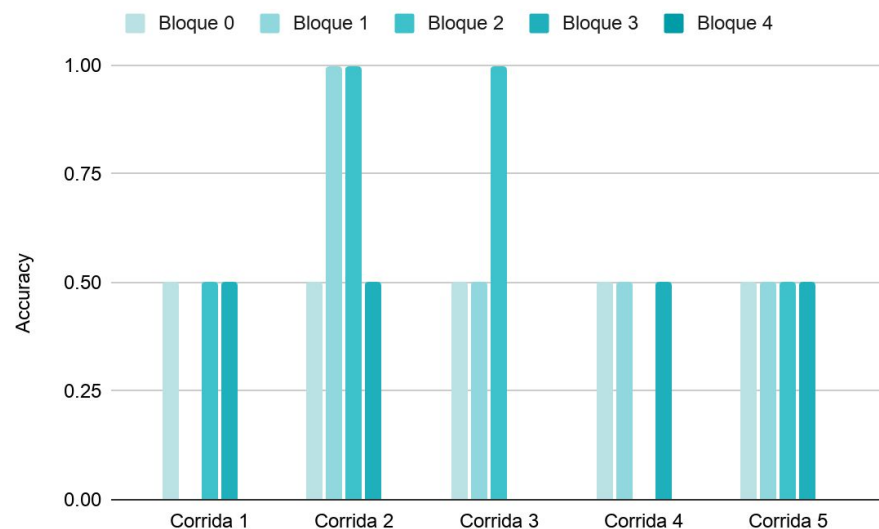
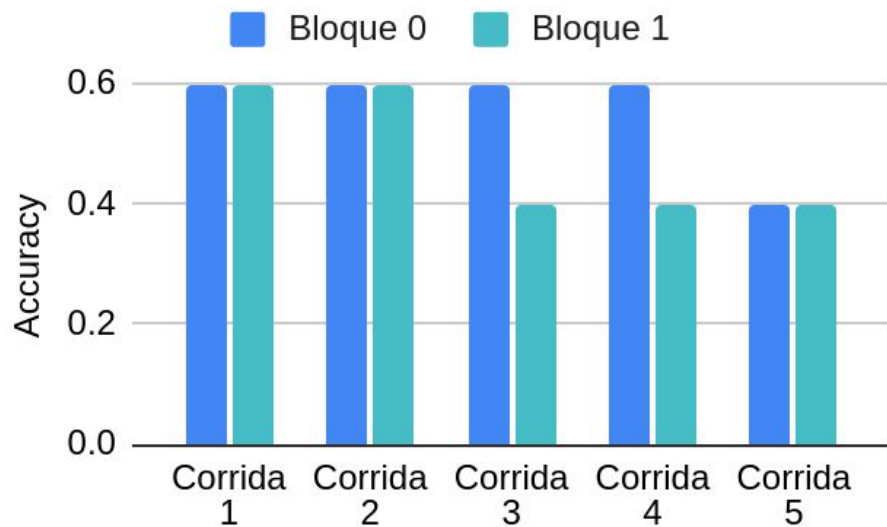
8 - 4 - 8 - 1

4 - 1

# VALIDACIÓN CRUZADA

$K = 2 / K = 5$

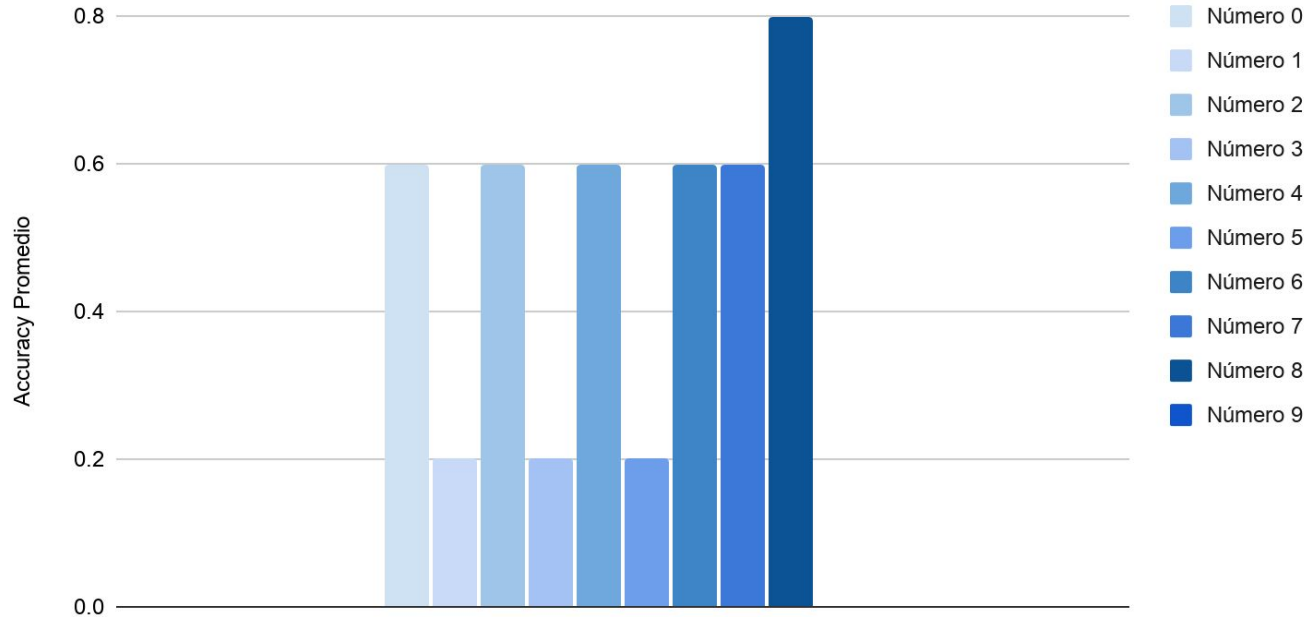
Randomización del orden de los datos antes de cada corrida



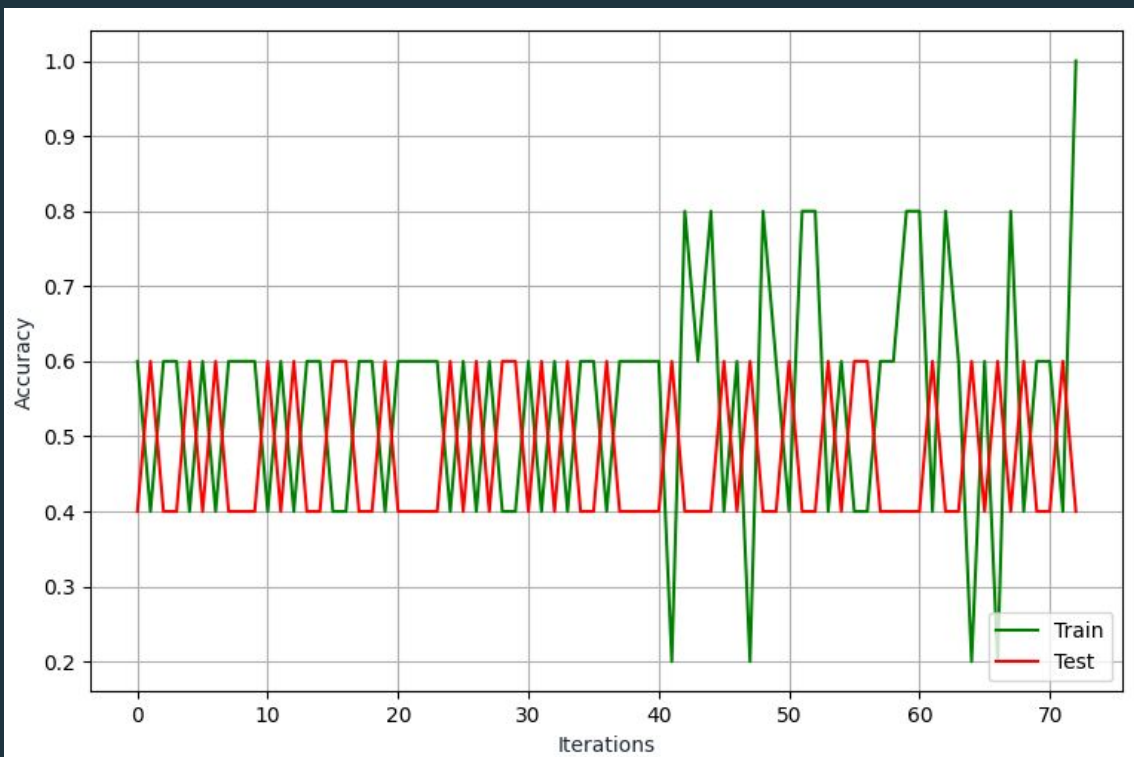
# VALIDACIÓN CRUZADA

$K = 10$

Sin randomización de orden, cada barra es un **promedio de aciertos** al usar dicho dígito para pruebas.



# APRENDIZAJE Y GENERALIZACIÓN



Train:  $\frac{1}{2}$  del dataset  
Test:  $\frac{1}{2}$  del dataset  
Learning Rate: 0.01

Hay **overfitting** a medida que aprende más, la generalización baja



# APRENDIZAJE Y PERCEPTRÓN SIMPLE

Corriendo el dataset con un **Perceptrón Simple**, lo aprende al

**100%**



Es **linealmente separable**, y en generalización se comporta como la red

# CONCLUSIONES DE EJERCICIO 3

## OVERFITTING

Con el dataset de dígitos se llegó a overfitting

## GENERALIZACIÓN

Con Validación Cruzada generaliza bien el  $\approx 50\%$  independientemente de los bloques usados, y con 2 clases es muy azaroso

## L. SEPARABLE

El dataset del problema 2 puede ser aprendido 100% con un Perceptrón Simple

# CONCLUSIONES

04



# CONCLUSIONES ALCANZADAS

## OPTIMIZACIONES

Momentum no conviene usarlo siempre

## PROBLEMAS

Importante ver los tipos de problemas

## VALIDACIÓN CRUZADA

En el EJ2 mejora la accuracy en la generalización, pero en el EJ3 no mejora

## MÉTRICAS

Las métricas permiten evaluar diferentes configuraciones

# ¡GRACIAS!

¿Preguntas?

ghirsch@itba.edu.ar  
fpetrikovich@itba.edu.ar  
juoliva@itba.edu.ar