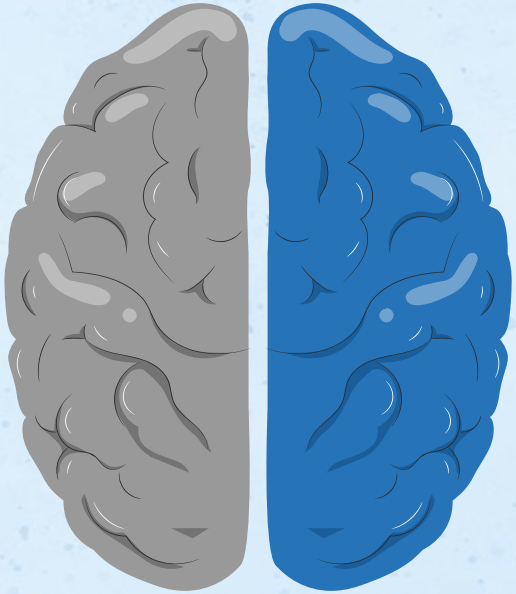
SIA - TP3

PERCEPTRÓN SIMPLE Y MULTICAPA

Grupo 6

- Francisco Sendot
- Lucia Digon
- Martín E. Zahnd
- Juan Ignacio Fernández Dinardo



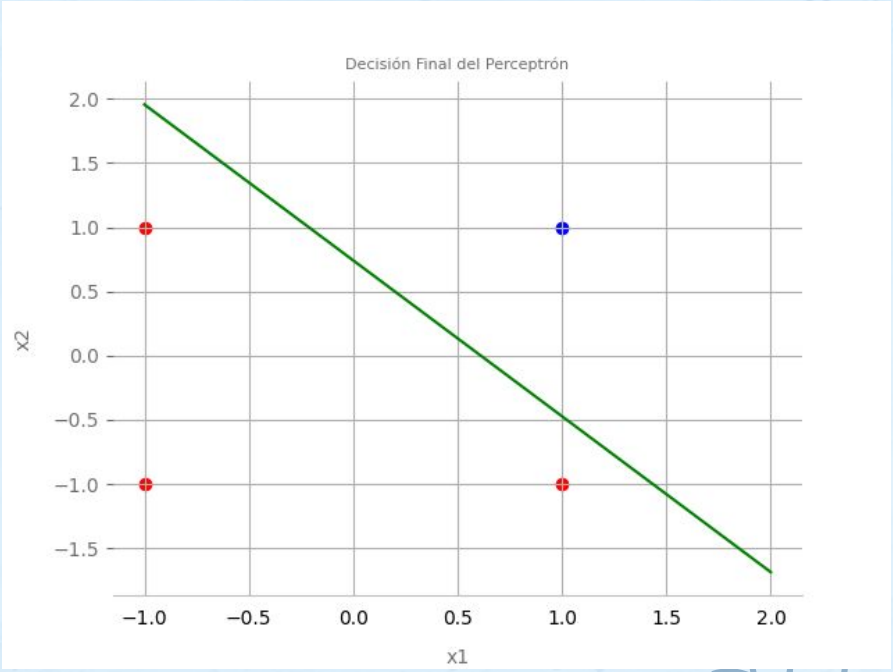
01

PERCEPTRÓN SIMPLE

PERCEPTRÓN SIMPLE

AND

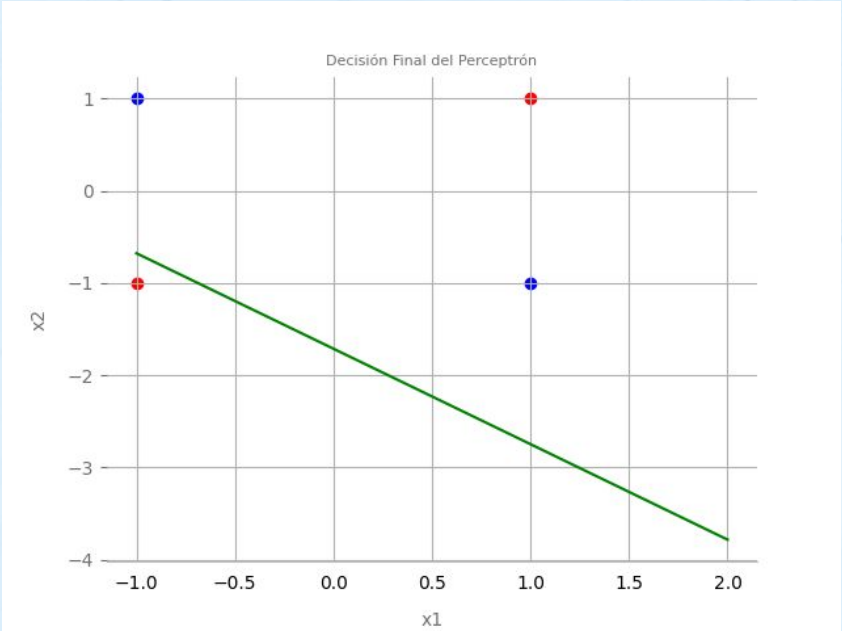
x1	x2	Salida esperada
-1	1	-1
1	-1	-1
-1	-1	-1
1	1	1



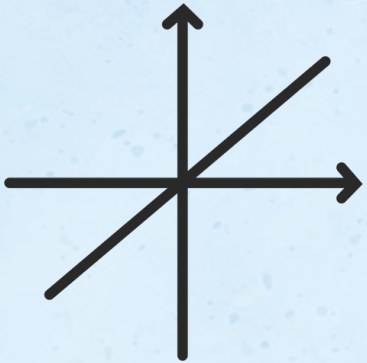
PERCEPTRÓN SIMPLE

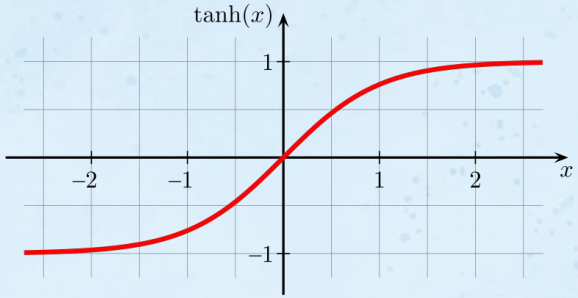
XOR

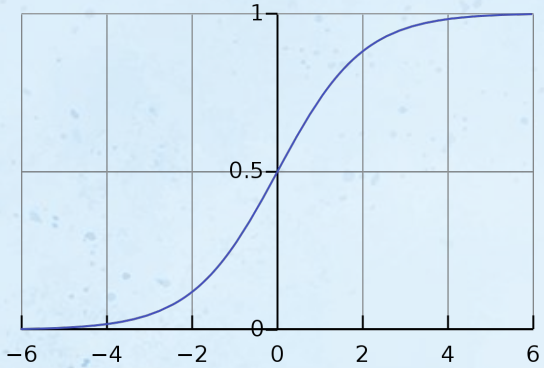
x1	x2	Salida esperada
-1	1	1
1	-1	1
-1	-1	-1
1	1	-1



PERCEPTRÓN SIMPLE LINEAL y NO LINEAL







PROPORCIÓN DE ENTRENAMIENTO : 0.7

LINEAL

Epocas: 10000/10000

NO LINEAL: tanh

Epocas: 903/10000

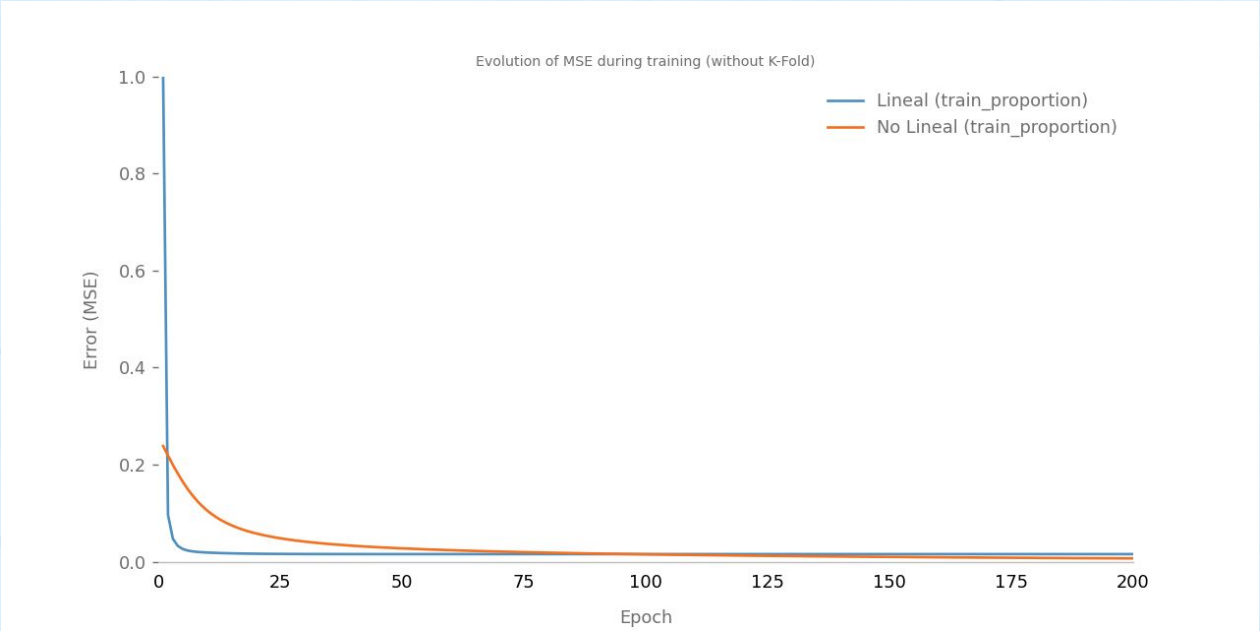
Error MSE: 9.03e-4

NO LINEAL: logistic

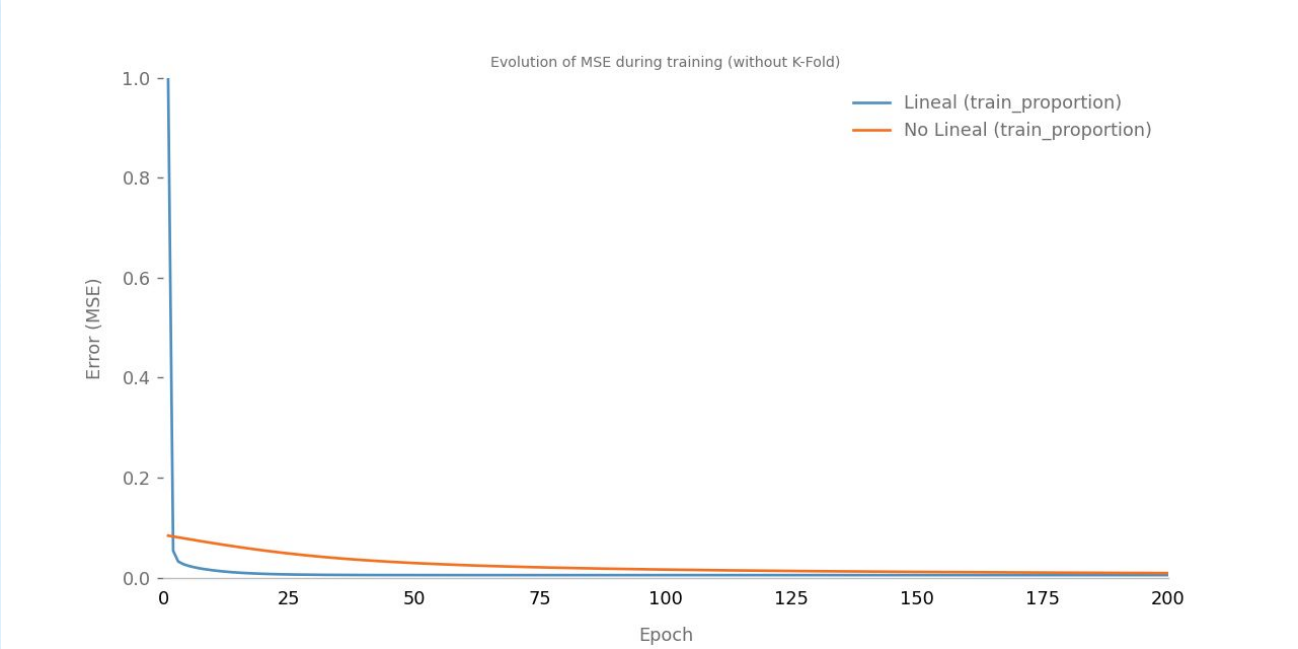
Epocas: 1911/10000

Error MSE: 1.91e-3

LINEAL VS NO LINEAL (tanh)

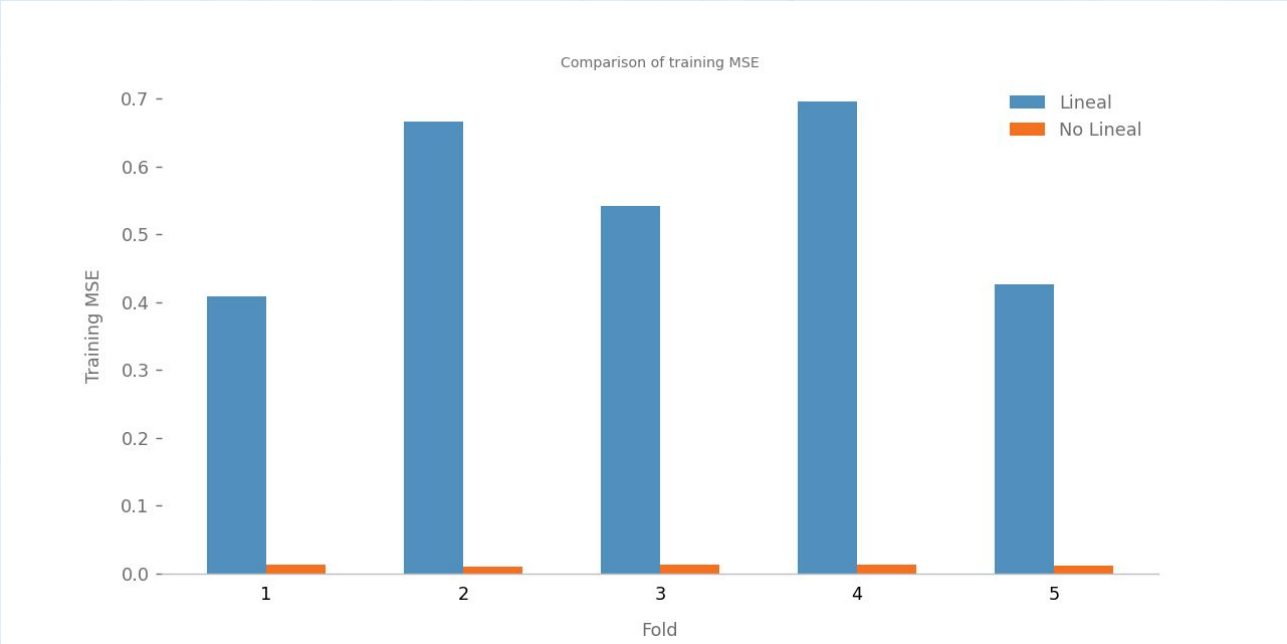


LINEAL VS NO LINEAL (logistic)



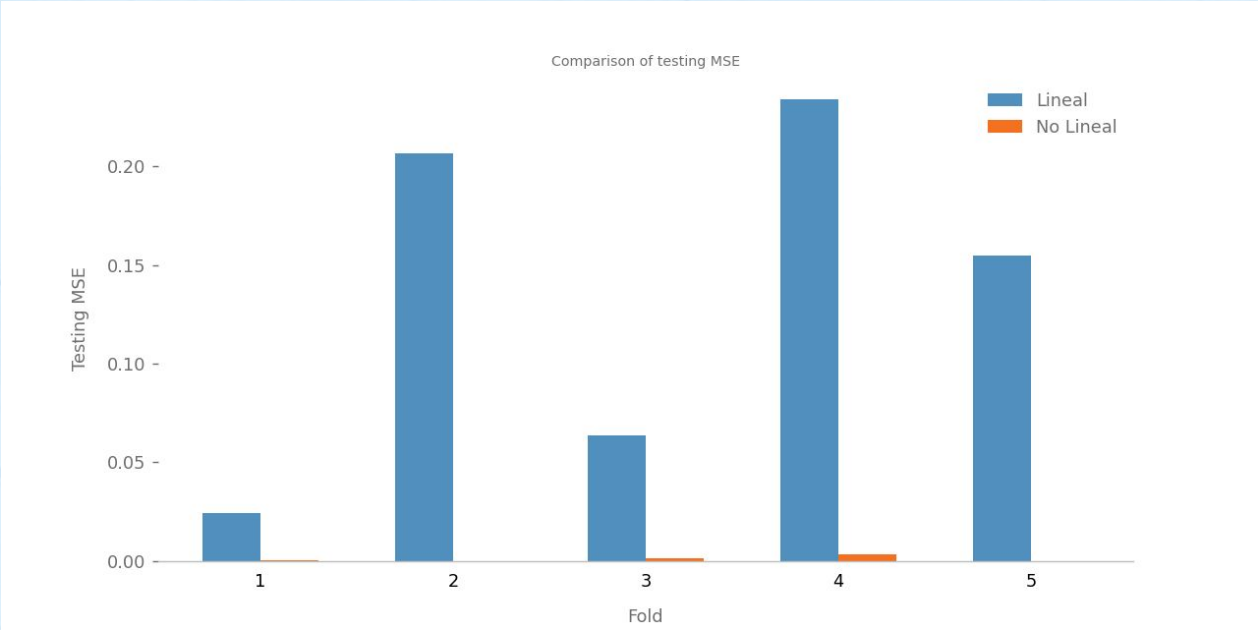
VALIDACIÓN CRUZADA: 5-FOLDS

LINEAL VS NO LINEAL (tanh)



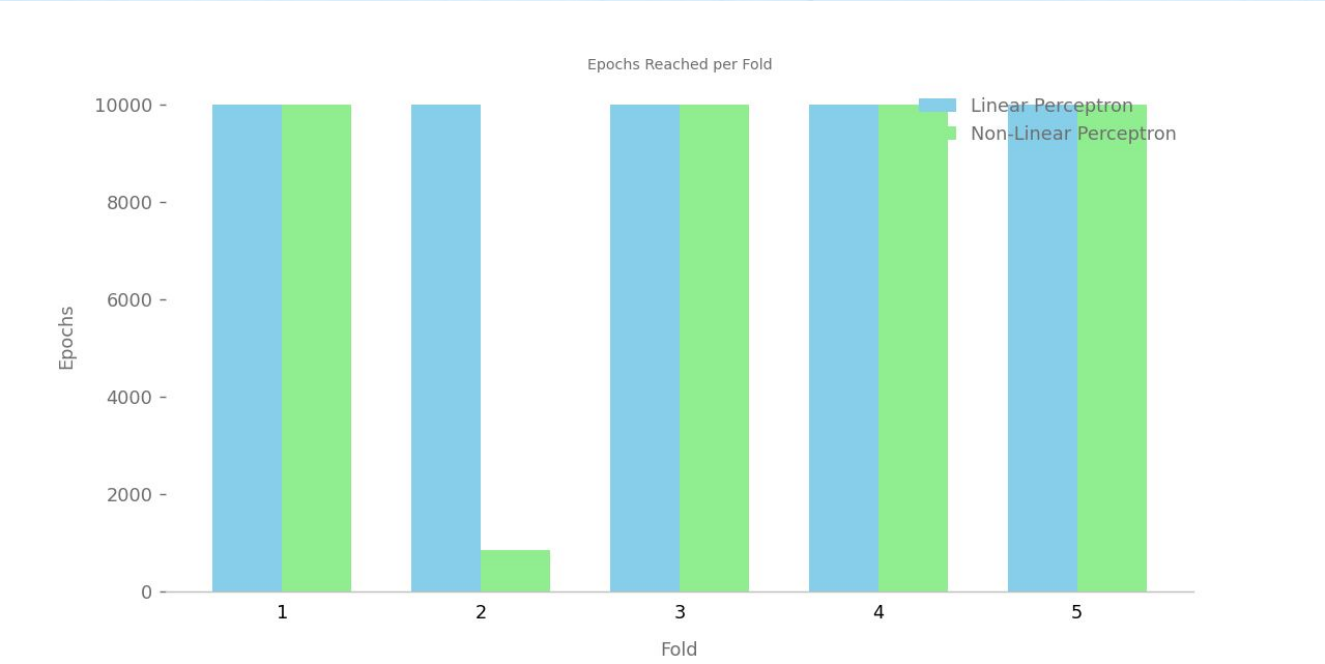
VALIDACIÓN CRUZADA: 5-FOLDS

LINEAL VS NO LINEAL (tanh)



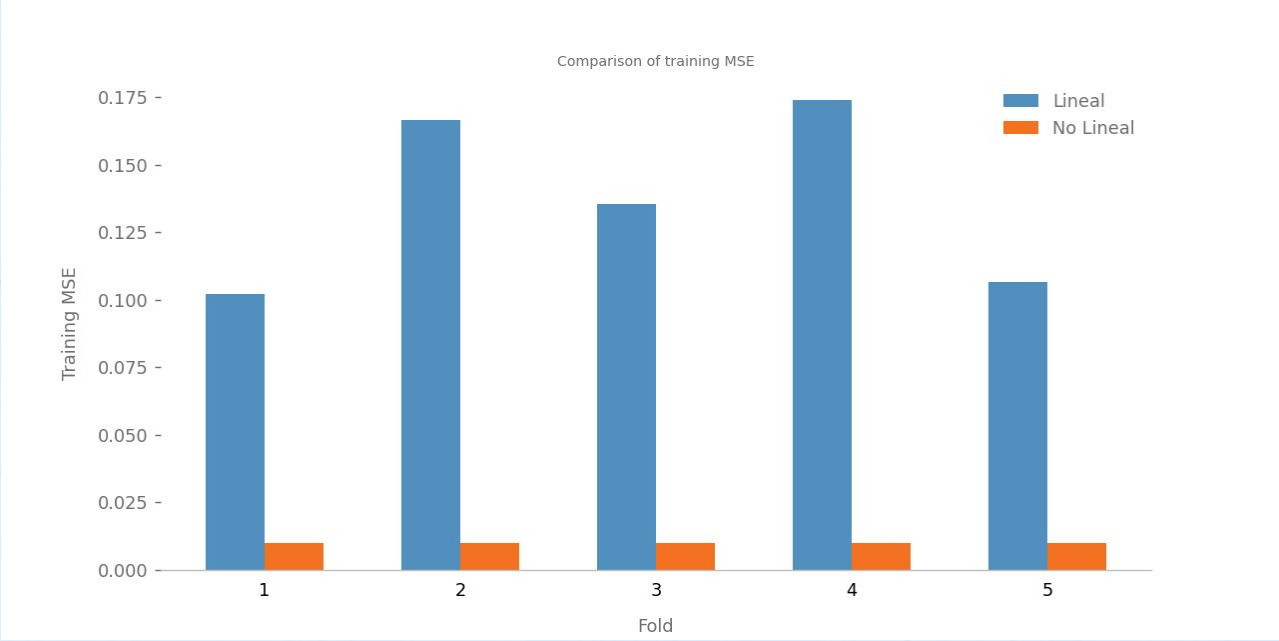
VALIDACIÓN CRUZADA: 5-FOLDS

LINEAL VS NO LINEAL (tanh)



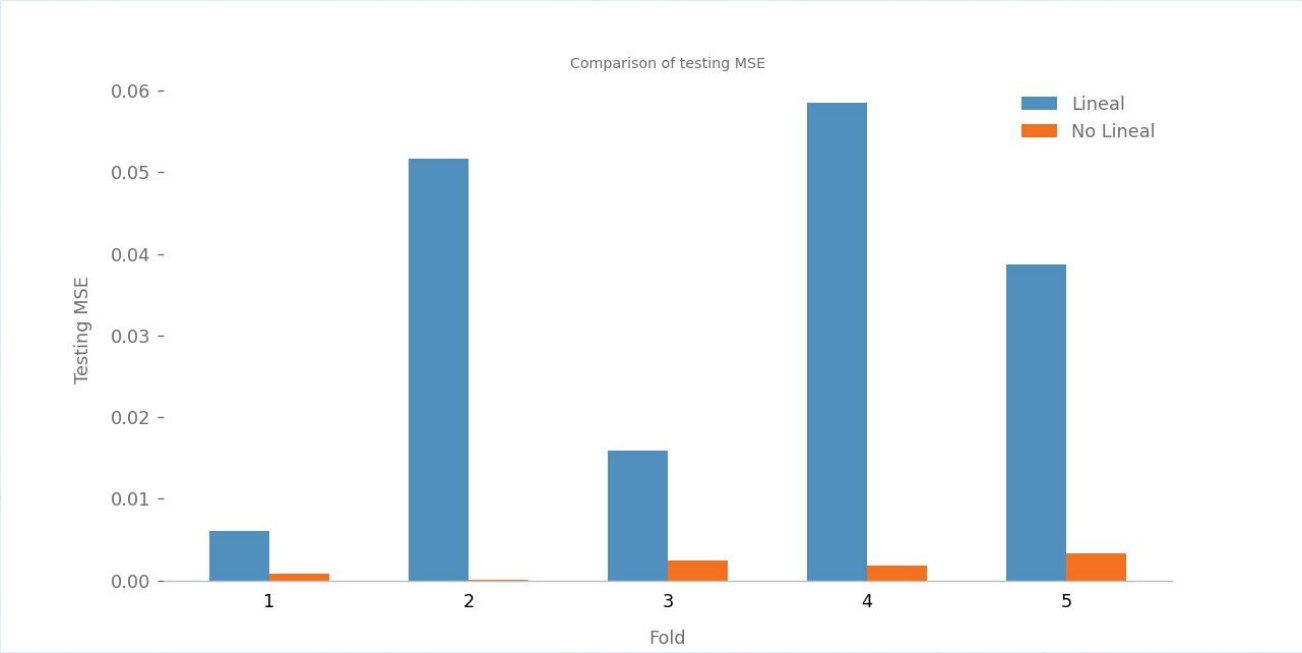
VALIDACIÓN CRUZADA: 5-FOLDS

LINEAL VS NO LINEAL (logistic)



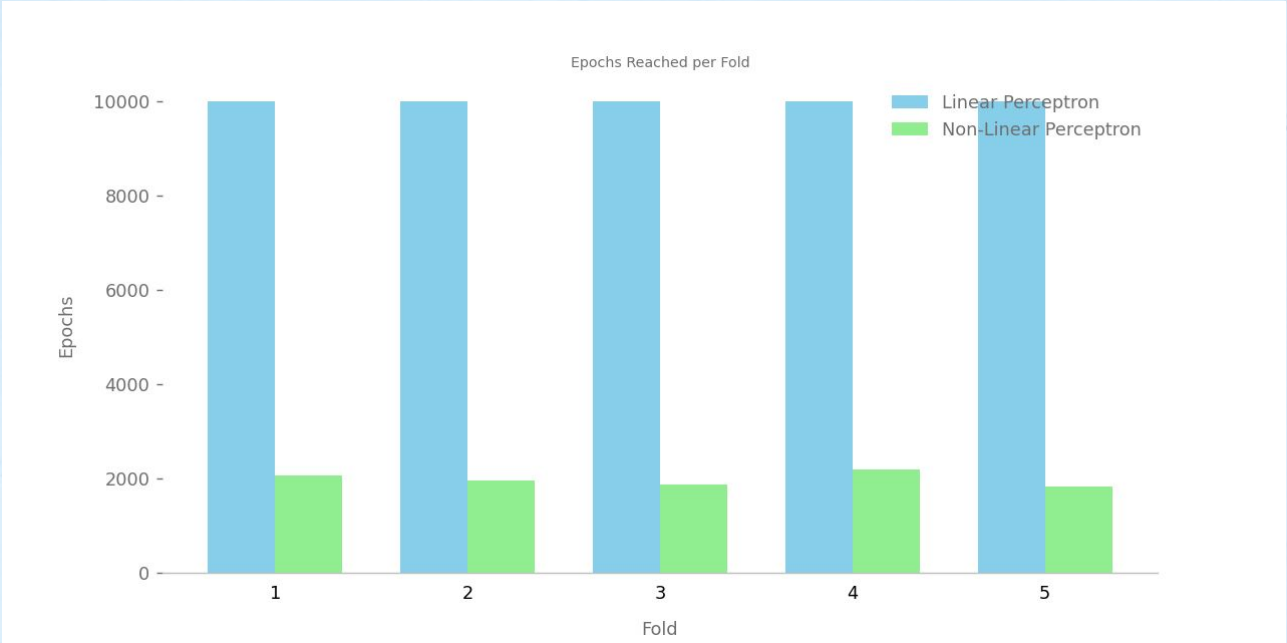
VALIDACIÓN CRUZADA: 5-FOLDS

LINEAL VS NO LINEAL (logistic)



VALIDACIÓN CRUZADA: 5-FOLDS

LINEAL VS NO LINEAL (logistic)



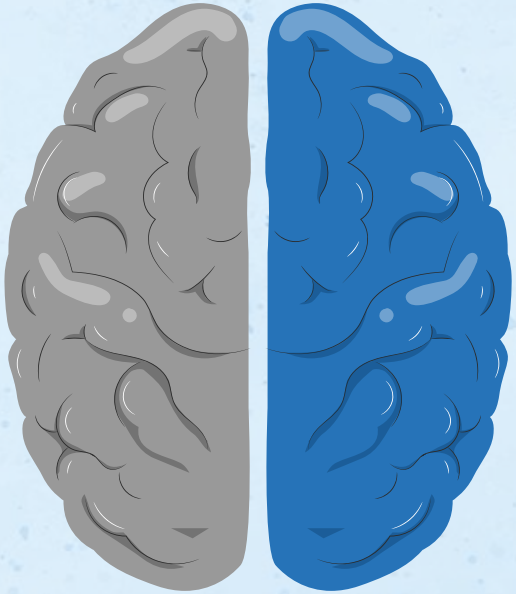
CONCLUSIONES

- El perceptrón simple logra resolver la operación AND ya que es linealmente separable, sin embargo, no puede resolver el XOR, ya que éste no es linealmente separable

- Mediante la validación cruzada se puede observar que hay conjuntos que resultan ser más representativos que otros, puesto que entrenar a la red con estos, permiten una mejor generalización sobre aquellos del set de test.

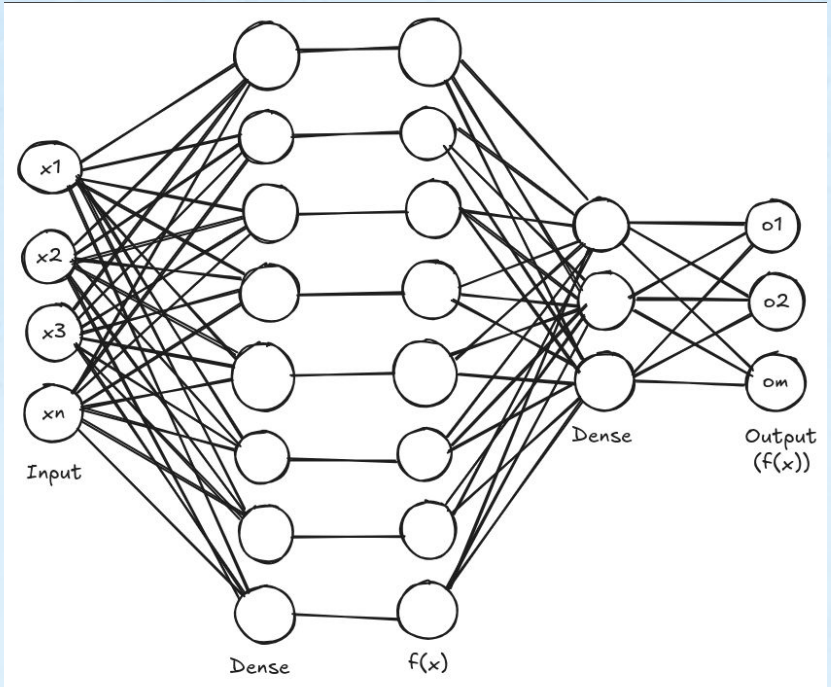
- Al particionar directamente sobre el set se corre el riesgo de que los datos pueden estar mal distribuidos y, de esta forma, el modelo puede sobre ajustarse y no generalizar bien.

- Tanto la función de activación logística o la tanh pueden ayudarnos a que con una el error converja más rápidamente. Esto dependerá de la naturaleza y relación entre los datos, pero, sin embargo, no hay una respuesta universal de cual es mejor.



02

PERCEPTRÓN MULTICAPA



ARQUITECTURA

La arquitectura consiste en 4 capas:

- La primera capa es Dense y expande la cantidad de Nodos
- La segunda es una función de activación
- La tercera es Dense para reducir los nodos al tamaño del output
- La última es otra función de activación que da el output

ENTRENAMIENTO

BATCH

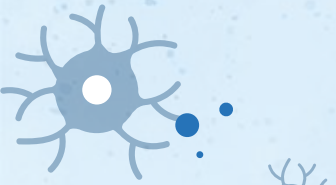
La actualización de los pesos de la red se hace luego de calcular el Δw **para todos los elementos** del conjunto de datos

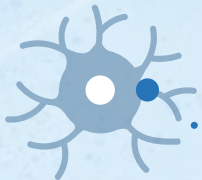
MINI BATCH

La actualización de los pesos de la red se hace luego de calcular el Δw **para un subconjunto de elementos** del conjunto de datos

ONLINE

La actualización de los pesos de la red se hace luego de calcular el Δw **para un elemento** del conjunto de datos

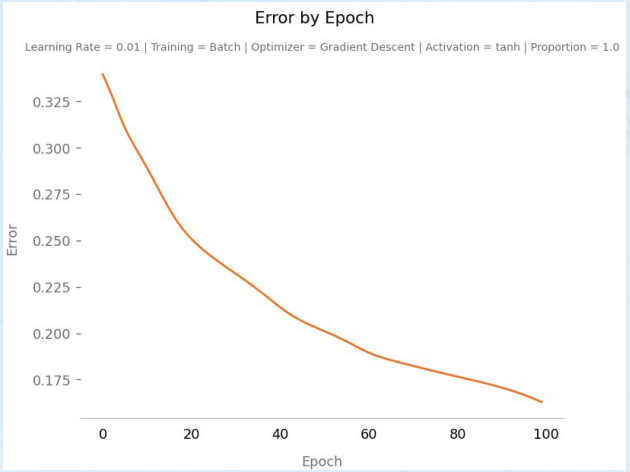




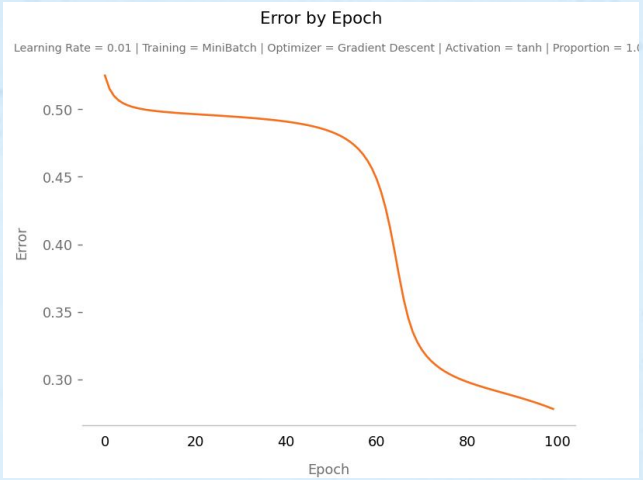
COMPARACIÓN ENTRENAMIENTO

Paridad

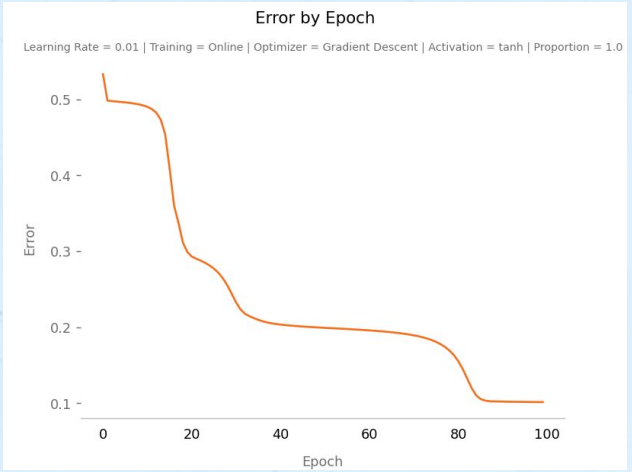
BATCH



MINI BATCH



ONLINE



OPTIMIZADORES

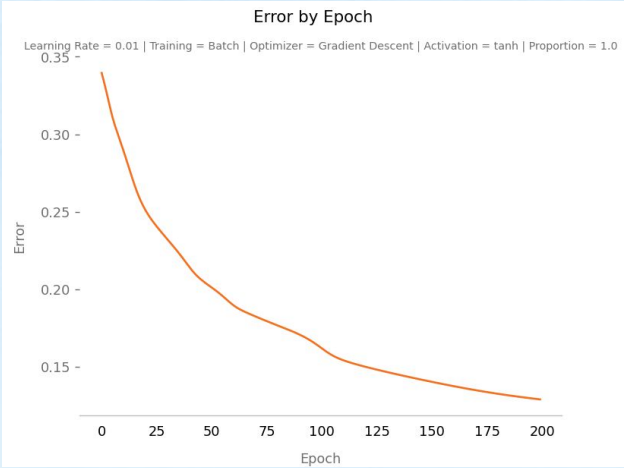
GRADIENT DESCENT

MOMENTUM

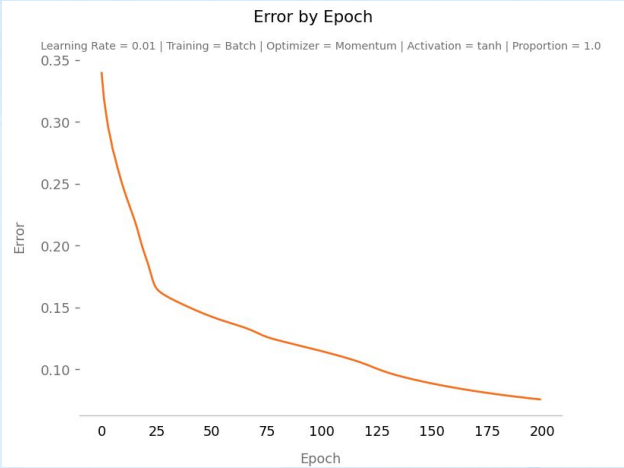
ADAM

COMPARACIÓN OPTIMIZADORES

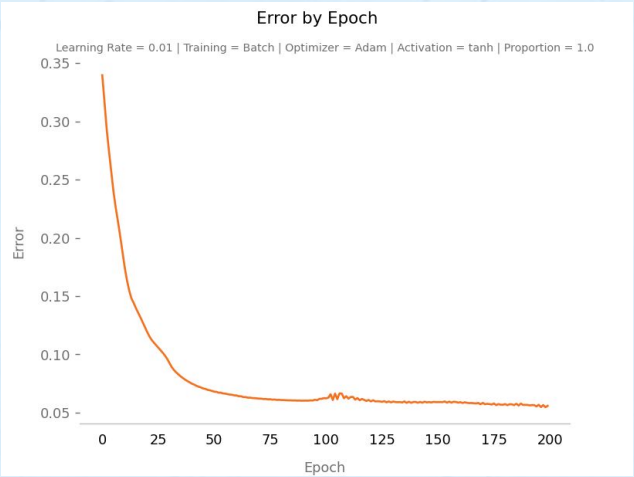
GRADIENT DESCENT



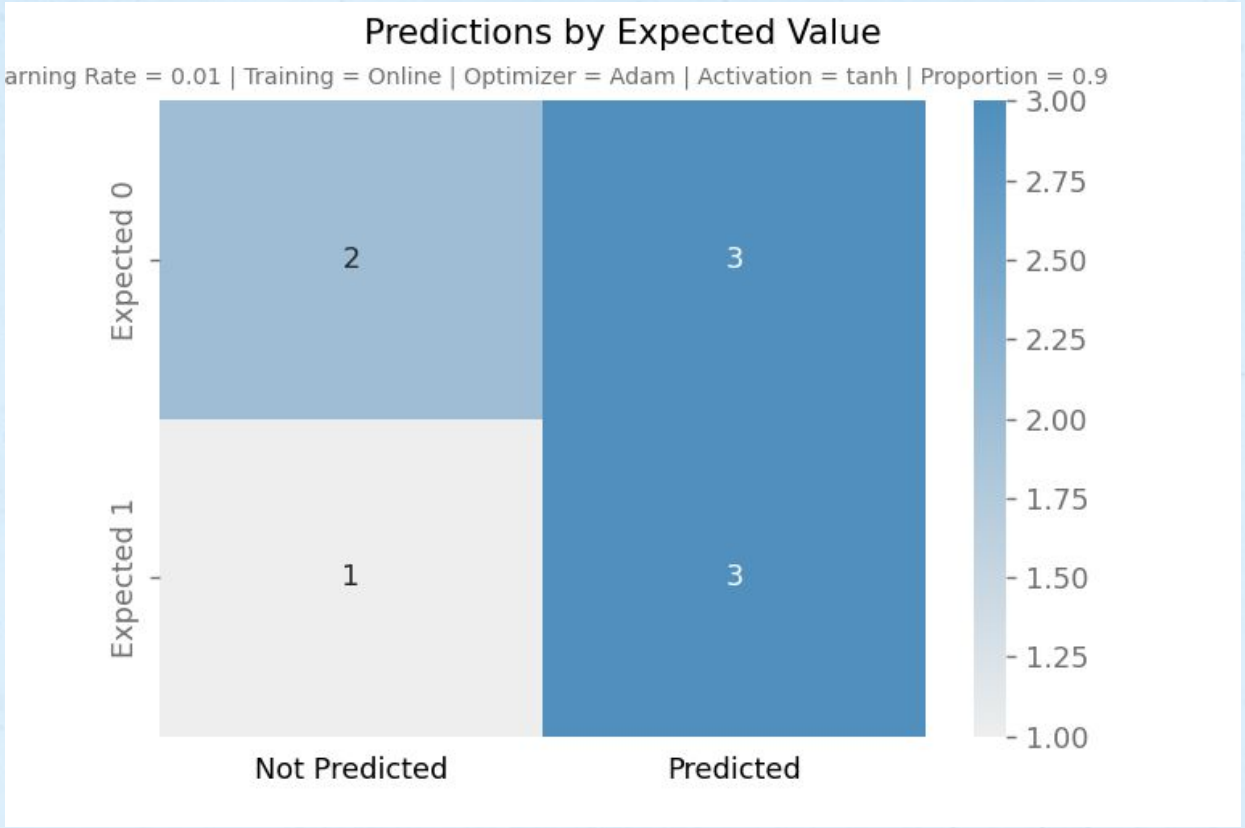
MOMENTUM



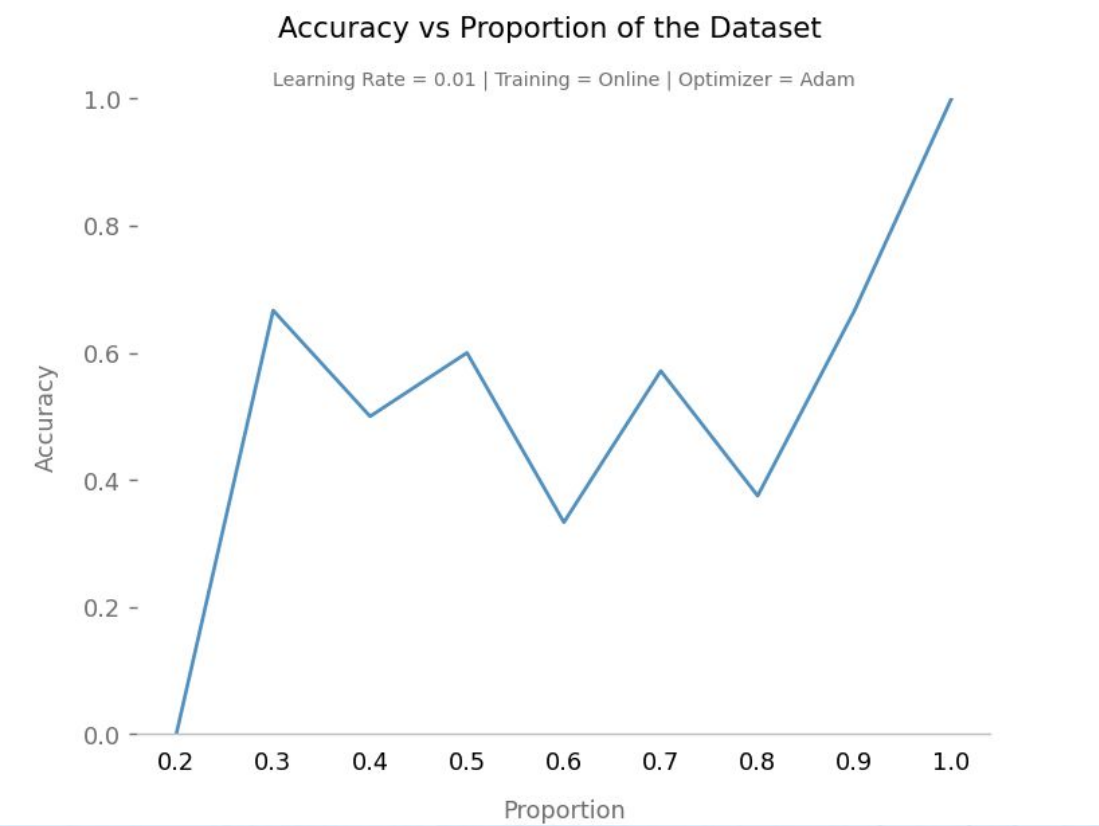
ADAM



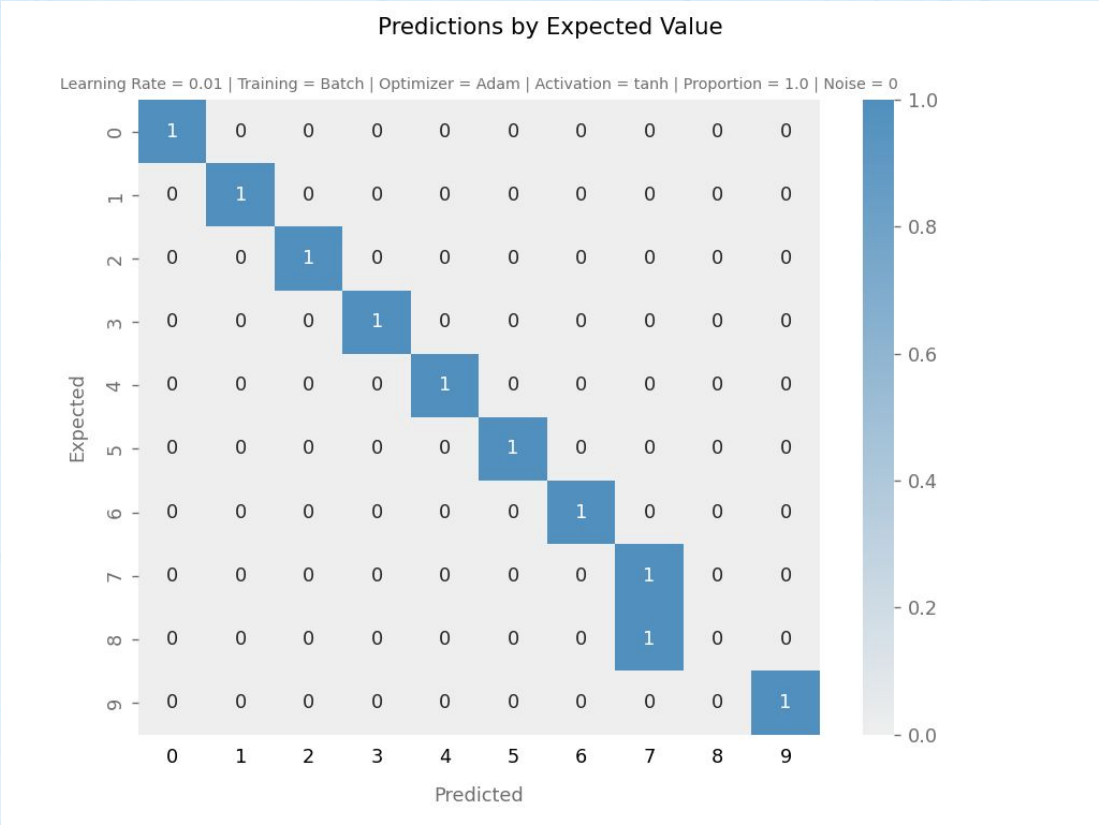
DISCRIMINACIÓN DE PARIDAD



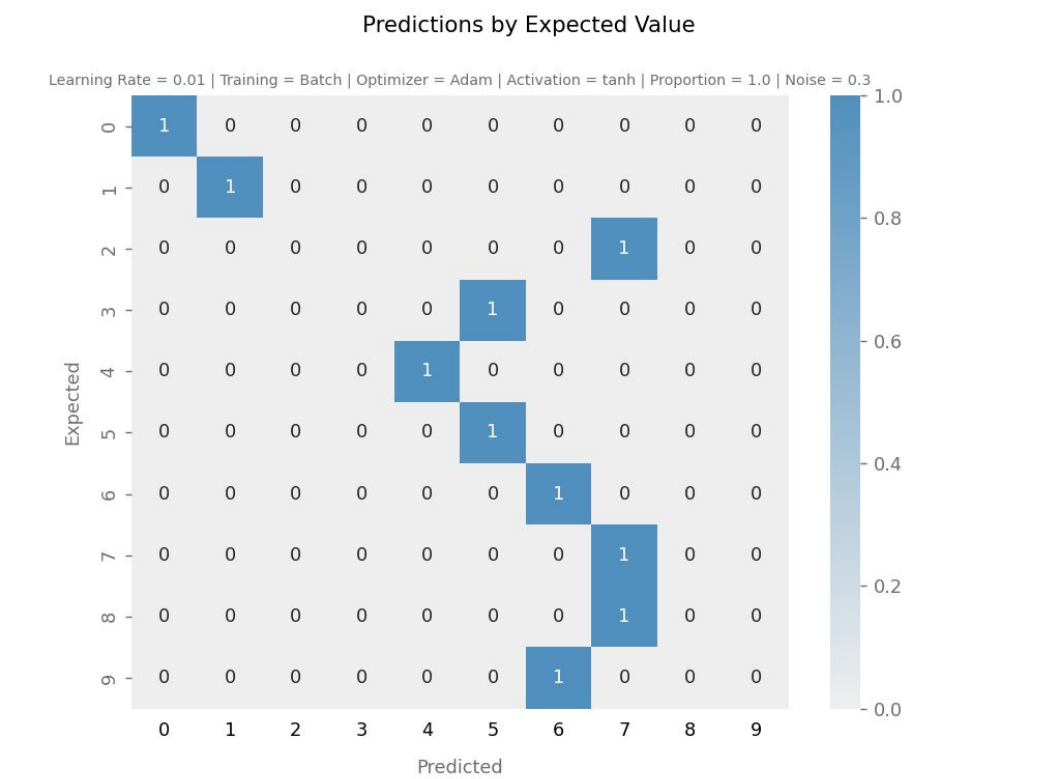
DISCRIMINACIÓN DE PARIDAD



DISCRIMINACIÓN DE DÍGITO



DISCRIMINACIÓN DE DÍGITO: CON RUIDO



DISCRIMINACIÓN DE DÍGITO: MNIST

- El Dataset ahora tiene un input y output de 60000
- Son **MUCHOS DATOS**
- Se tienen que cambiar decisiones de arquitectura para que pueda correr en tiempo y forma
  - *epoch*=200
  - Pasamos de dos capas de Dense a una sola, con sólo una función de activación
- Se usa Mini Batch con un *batch size* de 10000
- Como optimizador se eligió *Adam*
- Como función de activación se eligió *Tanh*
- Tiene un dataset para test (no es necesario agregar ruido)

CONCLUSIONES

- La discriminación por paridad está adivinando, no puede commprender o predecir con precisión qué es un número par o impar.
- En la discriminación por dígito sin ruido tiene bastante precisión, pero cuanto más alto es el ruido que se le agrega, más imprecisa se vuelve la red.
- Según la manera de entrenamiento, varía la forma en la que va disminuyendo el error. Lo mismo sucede en el caso de los optimizadores
- Cuanto más pequeño es el *learning rate*, menor es el error, pero necesita más *epochs*

BIBLIOGRAFÍA

- Apuntes de la catedra

GRACIAS