

Universidad de Guadalajara  
Centro Universitario de Ciencias Exactas e Ingenierías

Ingeniería en Computación  
Seminario de Solución de Problemas de Inteligencia Artificial II  
Profesor: Campos Peña Diego

Lepiz Hernández Manuel Alejandro  
Código: 214797998



Practica 1.

Ejercicio 2.

## Introducción

Se realizará en Python el entrenamiento y prueba de un perceptrón simple, usando un archivo .csv para la entrada de datos. Primeramente, se generarán distintos archivos para la prueba del perceptrón, usando una selección completamente aleatoria de los mismos, usando distintos porcentajes para cada archivo.

## Desarrollo

Se usó el mismo perceptrón que para la práctica anterior

```
[66] class Perceptron:
    def __init__(self, nEntradas, tasaAprendizaje, epochs):
        self.weights = np.zeros(nEntradas)
        self.bias = 0
        self.tasaAprendizaje = tasaAprendizaje
        self.epochs = epochs

    def predecir(self, entradas):
        activation = np.dot(self.weights, entradas) + self.bias
        return -1 if activation < 0 else 1

    def entrenamiento(self, dataEntrenamiento, capas):
        for i in range(self.epochs):
            for entradas, capa in zip(dataEntrenamiento, capas):
                prediccion = self.predecir(entradas)
                self.weights += self.tasaAprendizaje*(capa-prediccion)*entradas
                self.bias += self.tasaAprendizaje*(capa-prediccion)
```

Se generan las particiones de los archivos, siendo que se trabajaran mas adelante



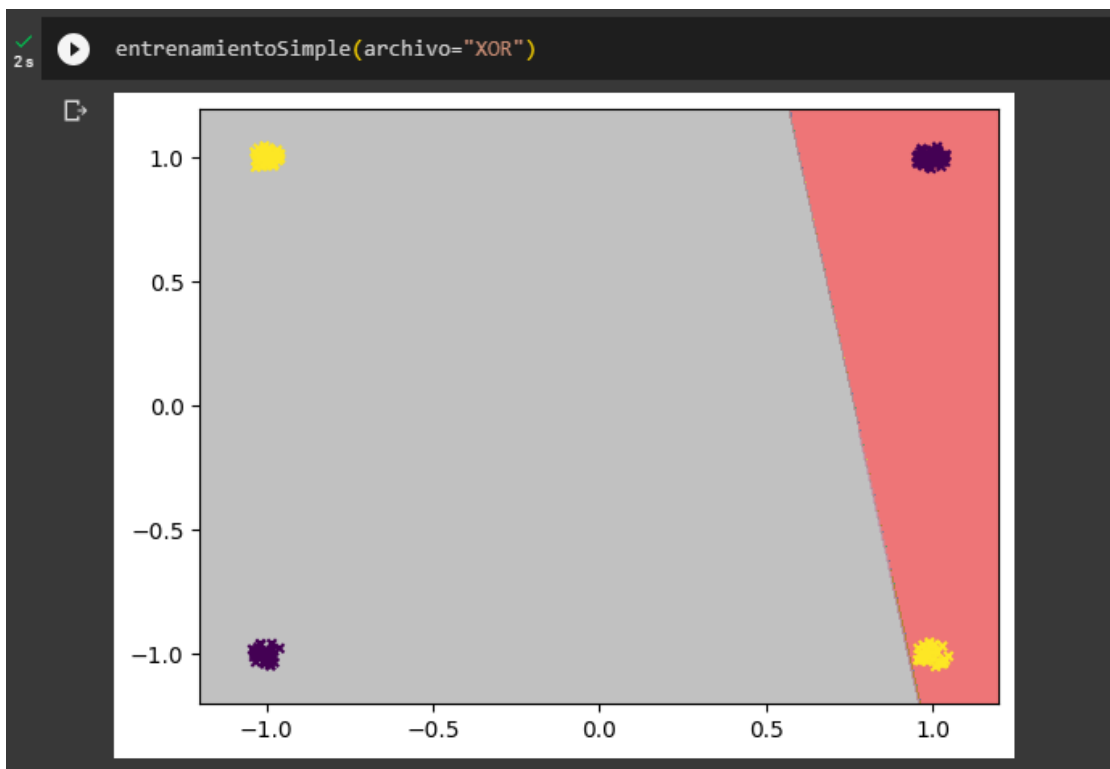
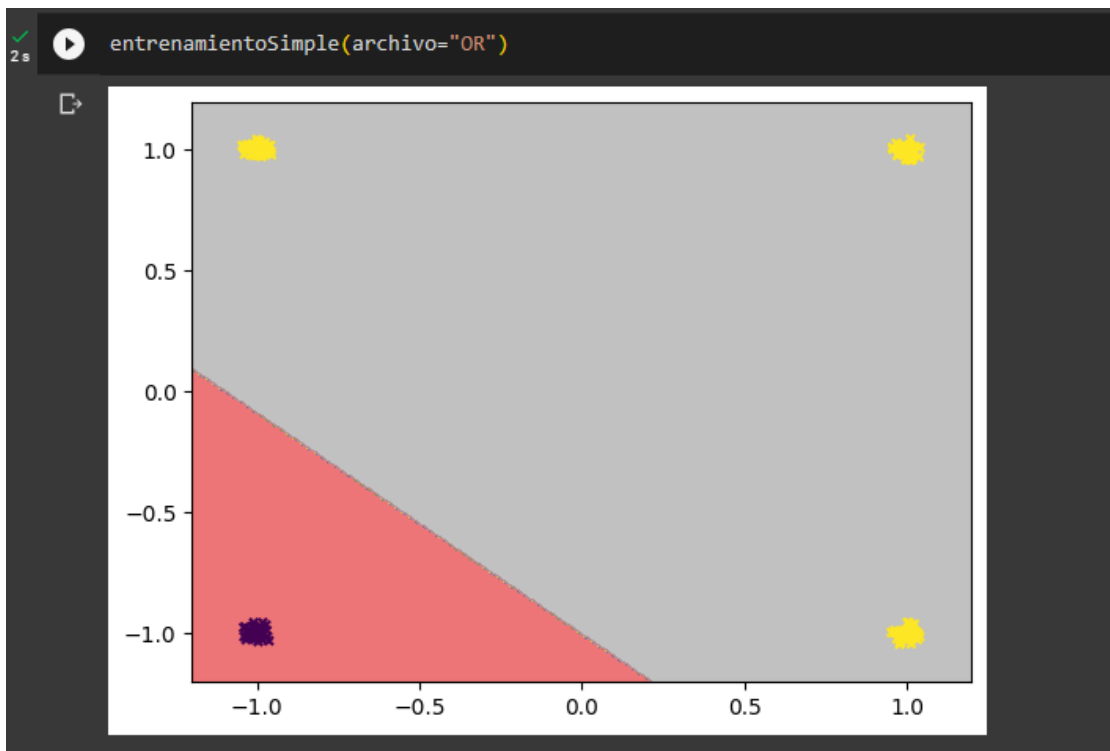
```
def generarParticion(self, directorio, particiones, nombre):
    entrenamiento = []
    capa = []

    for particion in particiones:
        archivo = open(f"{directorio}{particion}", mode="r")
        lineas = archivo.readlines()
        for linea in lineas:
            cadenas = linea[:len(linea)-1].split(',')
            x1 = float(cadenas[0])
            x2 = float(cadenas[1])
            x3 = float(cadenas[2])
            x4 = int(cadenas[3])
            entrenamiento.append([x1, x2, x3])
            capa.append(x4)

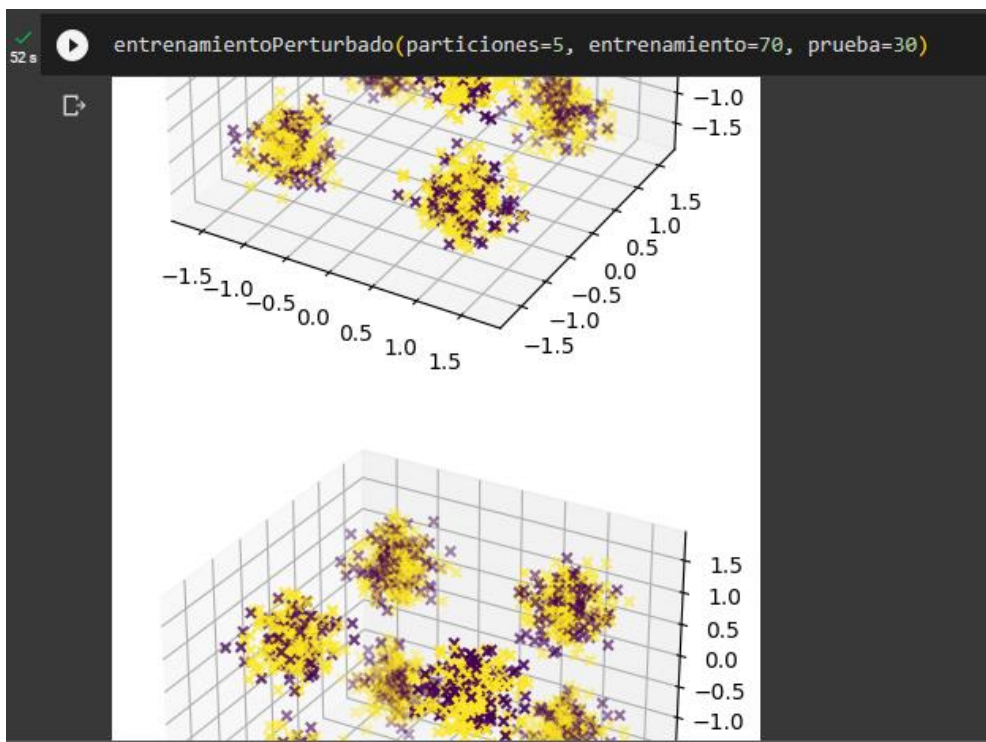
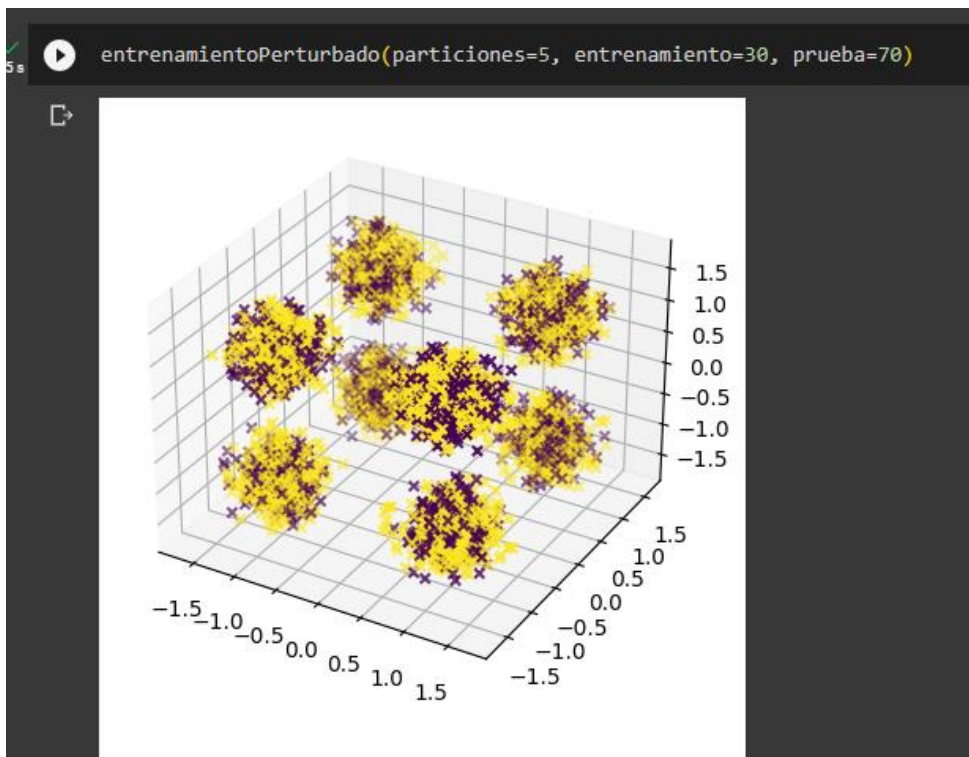
    random.shuffle(entrenamiento)
    random.shuffle(capa)

    dataSet = open(fr"DataSets/Particiones/Particion{nombre}.csv", mode="w")
    for i in range(len(entrenamiento)):
        dataSet.write(f"{entrenamiento[i][0]},{entrenamiento[i][1]},{entrenamiento[i][2]},{capa[i]}\n")
    dataSet.close()
    archivo.close()
```

Mandamos a llamar las funciones del perceptrón para cada uno de los entrenamientos, y cada uno de los archivos.



Luego hacemos esto para cada uno de los archivos de Esferas, siendo que generamos 5 particiones de cada uno.



## Conclusiones

Se realizaron correctamente las particiones, se comprobaron los archivos para asegurar que no fueran iguales.

El perceptrón funcionó correctamente con los set de datos proporcionados, aunque estos no fueron separados de manera correcta debido a las limitaciones mismas del perceptrón simple.