

# **Centro universitario de ciencias exactas e ingenierías**

*Inteligencia artificial 2*

## **Practica 1 – perceptrón**

*Alumnos:*

Ceja Zapata Carlos Alberto - 215629258

Jiménez García Mariana - 219294471

Profesor: Julio Esteban Valdez López

Sección D02

*Fecha: 14 de febrero del 2023*



## Introducción

La inteligencia artificial esta cada vez mas presente en los procesos de desarrollo y procesamiento de información. Existen muchas inteligencias artificiales que sorprenden con la capacidad con la que aprenden instrucciones simples y como, una vez entrenadas, pueden realizar mejor su proceso con los datos y seguir aprendiendo. Para esta actividad, trabajaremos con el Perceptrón, pero primero debemos entender que es lo que puede realizar:

El perceptrón es una neurona artificial o una unidad de red neuronal que realiza ciertos cálculos para detectar capacidades de datos de entrada.

Para esta actividad, implementaremos un Perceptron en el que trabajaremos con coordenadas en un plano cartesiano, mediante los pesos y las bias, calcularemos la pendiente de la grafica para después, agregar puntos manualmente en la gráfica, y que estos sean pintados del color correspondiente dependiendo de donde se ubiquen.

Esta practica se elaboró en Python, usando la librería matplotlib para la creación de la gráfica.

## Capturas de pantalla

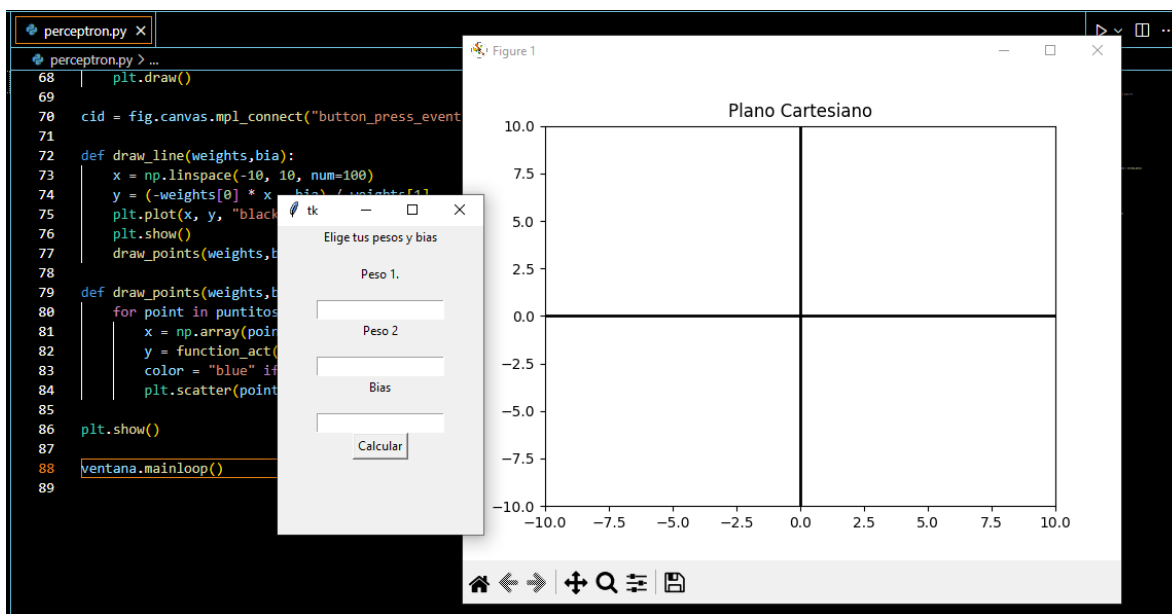


Imagen 1. Código perceptron.py en Visual Studio.

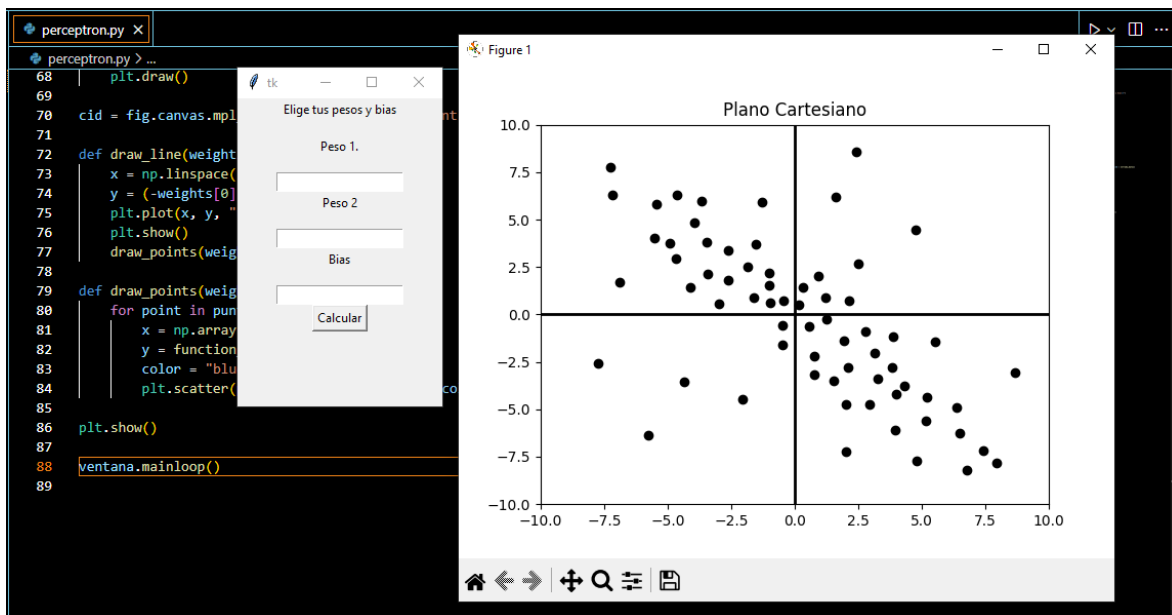


Imagen 2. Puntos ingresados la grafica antes de ser clasificados.

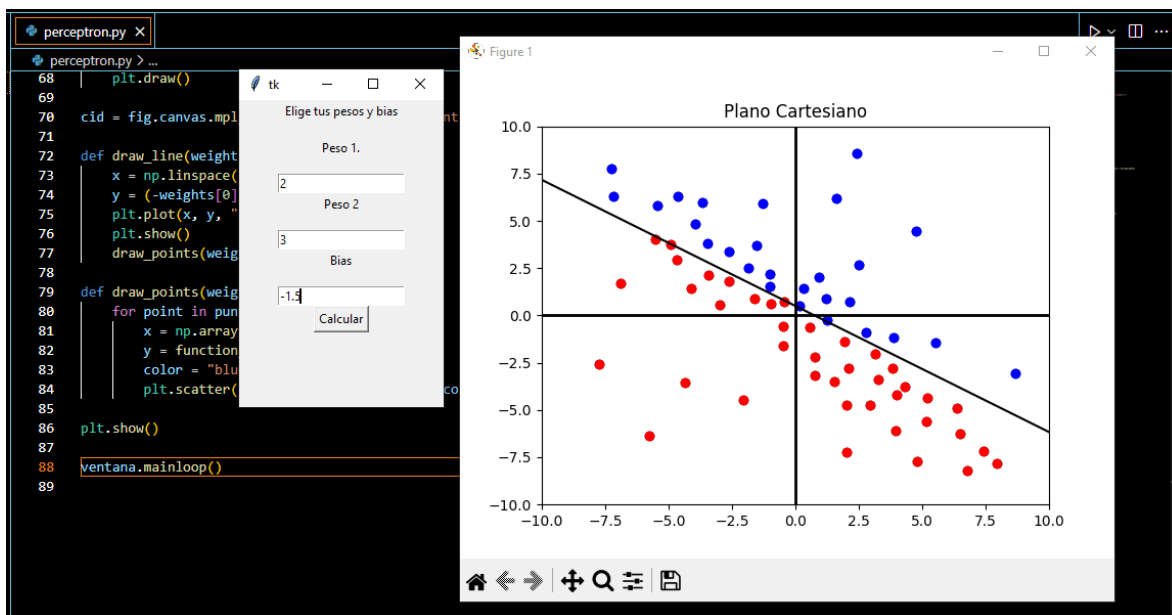


Imagen 3. Puntos clasificados mediante pesos y bias.

## Código Fuente

```
import tkinter
from math import e
import matplotlib.pyplot as plt
import numpy as np

puntitos = []

#Interfaz
ventana = tkinter.Tk()
ventana.geometry("200x300")

etiqueta = tkinter.Label(ventana, text = "Elige tus pesos y bias\n")
etiqueta.pack()

#entradas
tkinter.Label(ventana, text="Peso 1.\n").pack()
peso1 =tkinter.Entry(ventana)
peso1.pack()

tkinter.Label(ventana, text="Peso 2\n").pack()
peso2 =tkinter.Entry(ventana)
peso2.pack()

tkinter.Label(ventana, text="Bias\n").pack()
bias =tkinter.Entry(ventana)
bias.pack()

# Función de activación
def function_act(x, w, b):
    y = np.dot(x, w) + b
    if y >= 0:
        return 1
    else:
        return 0

def entrada_datos():
    pe1 = peso1.get()
    pe2 = peso2.get()
    b = bias.get()
    pes1 = float(pe1)
    pes2 = float(pe2)
    bia = float(b)
```

```

weights = np.array([pes1, pes2])

draw_line(weights,bia)

entrada1 =tkinter.Button(ventana, text = "Calcular",command =
entrada_datos)
entrada1.pack()

#Grafica
fig, ax = plt.subplots()
ax.set_xlim(-10, 10)
ax.set_ylim(-10, 10)
ax.set_title("Plano Cartesiano")

#Linea horizontal
ax.axhline(y=0, color='black', lw=2)

#Linea vertical
ax.axvline(x=0, color='black', lw=2)

def onclick(event):
    x, y = event.xdata, event.ydata
    puntitos.append([x, y])
    plt.scatter(x, y, color="black")
    plt.draw()

cid = fig.canvas.mpl_connect("button_press_event", onclick)

def draw_line(weights,bia):
    x = np.linspace(-10, 10, num=100)
    y = (-weights[0] * x - bia) / weights[1]
    plt.plot(x, y, "black")
    plt.show()
    draw_points(weights,bia)

def draw_points(weights,bia):
    for point in puntitos:
        x = np.array(point)
        y = function_act(x, weights, bia)
        color = "blue" if y >= 0.5 else "red"
        plt.scatter(point[0], point[1], color=color)

plt.show()

ventana.mainloop()

```

## Conclusiones

---

De esta practica aprendimos que el perceptrón y otros algoritmos de inteligencia artificial son realmente eficientes para el procesamiento de información y su clasificación.

Creemos que es muy importante aprender realmente como es que funcionan estas tecnologías, ya que cada vez mas facilitan las tareas del usuario. Esta es la ventaja que tienen sobre las demás tecnologías, pueden aprender instrucciones y replicarlas las veces que sean necesarias y, además, adaptarse a las circunstancias en las que se encuentren. Aunque debemos admitir que es primordial el buen entrenamiento de estas tecnologías, un mal entrenamiento puede producir datos erróneos que tendremos que corregir.