Un dibujo de un personaje animado

Descripción generada automáticamente con confianza bajaCentro universitario de Ciencias Exactas e Ingenierías

Departamento de Ciencias Computacionales

Seminario de Solución de problemas de Inteligencia Artificial II

Valdez López Julio Esteban

Novoa Ortega Diego

215485477

Ingeniería en Computación

D01

Practica 1

20/Enero/2022

**Introducción**

El psicólogo Frank Rosenbalt desarrollo un modelo simple de una neurona. A este modelo le llamó perceptrón en 1958. Una de las características más importantes es que pueden reconocer patrones. El perceptrón está formado por un conjunto de sensores de entrada que reciben esos patrones de entrada que deben reconocer.

El perceptrón para desarrollar es un perceptrón simple, este tiene un modelo unidireccional que consta de una red con una capa de salida n neuronas y una salida, la cual utiliza señales binarias.

**Pantallazos**

Gráfico

Descripción generada automáticamente

Las entradas del programa decidí hacerlas por medio de consola, cuando se ingresan los datos la gráfica se muestra. Para esas entradas la gráfica que se muestra es esta, que en este caso es la compuerta AND.

Gráfico, Gráfico de líneas, Gráfico de dispersión

Descripción generada automáticamente

En este caso se muestra en comportamiento de la compuerta OR con las entradas mostradas en la consola.

**Código Fuente**

import matplotlib.pyplot as plt

import numpy as np

#Definimos las variables para utilizar en el programa

Theta = 0

W = []

X = []

m = 0

b = 0

#Abrimos el archivo para obtener los valores de las entradas

with open('input.txt') as f:

    entrys = f.readlines()

#Separamos las entradas del documento en un vector

for entry in entrys:

    vector = entry.rstrip("\n").split(' ')

    vector[0] = int(vector[0])

    vector[1] = int(vector[1])

    X.append(vector)

#Imprimimos en consola para que el usuario ingrese los valores

w1 = float(input("W1: "))

w2 = float(input("W2: "))

theta = float(input("Theta: "))

#asignamos los valores ingresados por el usuario

Theta = theta

W = [w1, w2]

m = - W[0] / W[1]

b = Theta / W[1]

#Calculamos el producto punto con los valores

vF = np.dot(X ,W) - Theta >= 0

#Imprimimos el plano cartesiano para mostrar la gráfica

ejeX = np.arange(-2, 3, 1)

ejeY = np.arange(-2, 3, 1)

zeros = [0, 0, 0, 0, 0]

plt.plot(ejeX, zeros, 'k')

plt.plot(zeros, ejeY, 'k')

#Definimos los puntos por los que pasa la gráfica de acuerdo con las entradas del usuario

for i in range(len(X)):

    if vF[i]:

        plt.plot(X[i][0], X[i][1], 'og')

    else:

        plt.plot(X[i][0], X[i][1], 'or')

#Imprimimos el resultado de la salida

plt.axline((0, b), slope = m)

#mostramos la gráfica

plt.title('Perceptron')

plt.show()

**Conclusión**

Esta práctica me pareció una buena introducción para el curso, la teoría fue bien explicada por medio de vídeos y presentaciones, el código no fue del todo complicado por los conocimientos previos adquiridos en las primeras clases de Inteligencia Artificial del semestre pasado.

Fue interesante volver al funcionamiento de las compuertas que personalmente es algo que me gusta, también es sencillo de representar por la librería utilizadas y lo bien documentadas que están.

En prácticas posteriores tendré que aprender a utilizar interfaces gráficas en Python para implementarlas y poder mejorar mis programas.