Logotipo

Descripción generada automáticamente

***Burgoin Trasviña Julio Ruben***

Sistemas Inteligentes

*Profesor: Gerardo Estrada Gutierrez*

**Práctica 3**

*05/03/2024*

**Introducción**

En el campo del aprendizaje automático y la inteligencia artificial, el perceptrón es un algoritmo fundamental que se utiliza para la clasificación de datos. Su funcionamiento se basa en simular el comportamiento de una neurona artificial, donde se procesan múltiples entradas y se genera una única salida. En este contexto, una de las funciones de activación más comúnmente utilizadas es la función sigmoidal, que mapea cualquier valor de entrada a un rango entre 0 y 1, lo que la hace especialmente útil para problemas de clasificación binaria.

En esta práctica, se implementará un perceptrón en Python para clasificar un conjunto de datos de entrada en dos categorías distintas. Se utilizará una matriz de entrada que representará los datos de entrada y otra matriz que contendrá los valores de salida esperados para entrenar el perceptrón. Además, se definirá la función sigmoidal necesaria para calcular las salidas del perceptrón y se inicializarán los pesos del modelo con valores aleatorios.

El objetivo final de esta práctica es familiarizarse con la implementación básica de un perceptrón en Python y comprender el proceso de entrenamiento de un modelo de aprendizaje automático para la clasificación de datos.

**desarrollo (Código)**

import numpy as np

entradas = np.array([[0,0,1],

                    [1,1,1],

                    [1,0,1],

                    [0,1,1]])

salida\_esperada = np.array([[0],

                            [1],

                            [1],

                            [0]])

def sigmoid(x):

    return 1 / (1 + np.exp(-x))

def sigmoid\_gradiente(x):

    return x \* (1-x)

np.random.seed(42)

pesos=np.random.rand(3,1)

tasa\_aprendizaje=0.01

num\_iteraciones = 5000

for iteracion in range(num\_iteraciones):

    entrada\_ponderada=np.dot(entradas,pesos)

    salida\_obtenida=sigmoid(entrada\_ponderada)

    error=salida\_esperada-salida\_obtenida

    ajustes=error\*sigmoid\_gradiente(salida\_obtenida)

    pesos+=np.dot(entradas.T,ajustes) \* tasa\_aprendizaje

print("Salida obtenida despues del entrenamiento:")

print(salida\_obtenida)

print("\nPesos despues del entrenamiento")

print(pesos)

**“Resultado”**

Texto

Descripción generada automáticamente

**Conclusiones**

Conclusión:

En esta práctica, se ha llevado a cabo la implementación de un perceptrón en Python para la clasificación de datos utilizando la función sigmoidal como función de activación. Se ha creado una matriz de entrada con los datos de entrenamiento y otra matriz con los valores de salida esperados, permitiendo así entrenar el modelo y ajustar los pesos de manera adecuada. Además, se ha definido la función sigmoidal necesaria para calcular las salidas del perceptrón y se han inicializado los pesos con valores aleatorios.

El perceptrón es un concepto fundamental en el campo del aprendizaje automático y sirve como base para algoritmos más complejos. Esta práctica proporciona una introducción práctica a la implementación y entrenamiento de un perceptrón en Python, lo que permite comprender mejor cómo funcionan estos algoritmos y cómo se pueden aplicar en la clasificación de datos.

**Opinión**

Implementar un perceptrón desde cero en Python ha sido una experiencia enriquecedora. Aunque el algoritmo es relativamente simple, es fascinante observar cómo se pueden utilizar matrices y funciones matemáticas para realizar operaciones de clasificación. Esta práctica ofrece una base sólida para comprender conceptos más avanzados en aprendizaje automático y proporciona una apreciación más profunda de los algoritmos subyacentes en muchas aplicaciones de inteligencia artificial. Además, el proceso de experimentar con diferentes configuraciones y ajustes de parámetros permite una mayor comprensión de cómo se puede mejorar el rendimiento del modelo.

**bibliografía**

*Descubre la librerías Pandas y NumPy en Python. (s.f.). Imagina Formación: Cursos Online y Presencial, bonificados.* [*https://imaginaformacion.com/tutoriales/pandas-numpy-en-python#:~:text=NumPy%20proporciona%20una%20estructura%20de,operaciones%20de%20manipulación%20y%20análisis*](https://imaginaformacion.com/tutoriales/pandas-numpy-en-python#:~:text=NumPy%20proporciona%20una%20estructura%20de,operaciones%20de%20manipulación%20y%20análisis)*.*