Inteligencia artificial avanzada para la ciencia de datos I

Momento de Retroalimentación: Módulo 2 Uso de framework o biblioteca de aprendizaje máquina para la implementación de una solución. (Portafolio Implementación)

Sobre el código

Para esta entrega se realizó el código de un perceptron utilizando el método de descenso de la gradiente estocástico que puede recibir una cantidad n de características de entradas y una de salida en formato csv. Para el csv, la última columna del csv es la característica de salida.

Se realizó utilizando la librería pandas para la lectura y el tratamiento de los datos y usando la librería numpy para algunas operaciones como el producto punto.

Al usuario se le da a elegir los siguientes hiperparámetros:

- Número de épocas: el usuario puede elegir un número de épocas que quiere que sucedan en el algoritmo. Este hiperparámetro es opcional y si no se indica, se atravesarán épocas hasta que los pesos lleguen a la convergencia (lo cual no significa un buen resutado)..
- Learning rate: El usuario puede elegir el nivel de learning rate que él desee. Este hiperparámetro no es opcional.
- Porcentaje de datos para entrenamiento: Este hiperparámetro indicará qué porcentaje de los datos se desean para usarse para entrenar al modelo y, por consiguiente, qué porcentaje de datos se usarán para testing del dataset. Este hiperparámetro no es opcional.

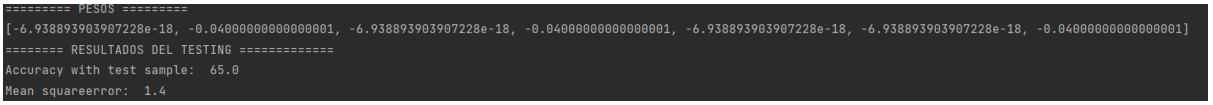
Es importante mencionar que aunque los hiperparámetros sean iguales en distintas corridas, es probable que no se obtengan los mismos resultados en cada corrida del algoritmo ya que el perceptron está diseñado para tomar muestras aleatorias de los datos cada vez que se corra el algoritmo.

Pruebas

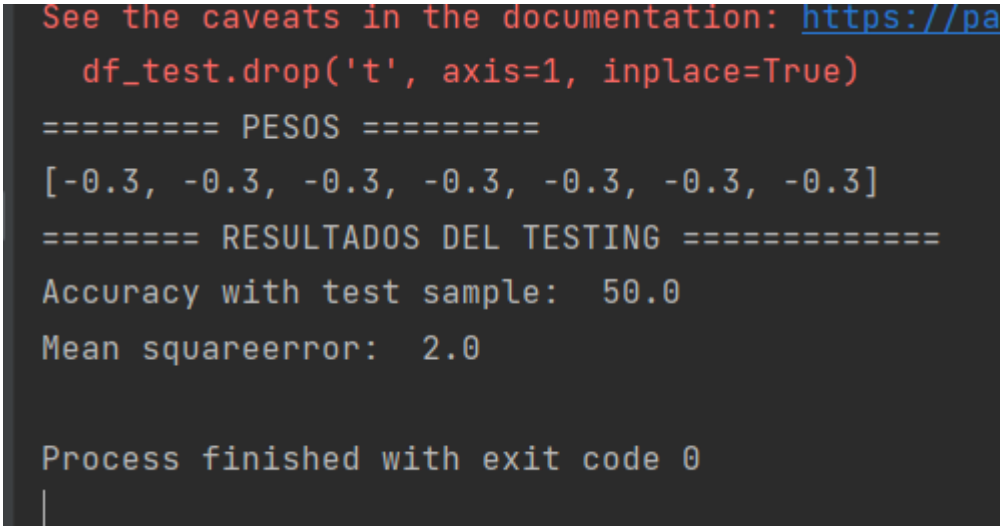
En el repositorio de github se tienen 3 datasets distintos para pruebas:

- gatoperro: se tienen varios datos sobre perros y gatos donde el modelo tratará de predecir si con las características dadas se está hablando de un perro o de un gato.
- test: donde se tienen características de personas y el algoritmo intentará determinar si una persona me cae bien o no.
- xor: donde se tiene una tabla de verdad y el algoritmo intentará predecir el restante de la tabla de verdad habiendo visto solo una parte de ella.

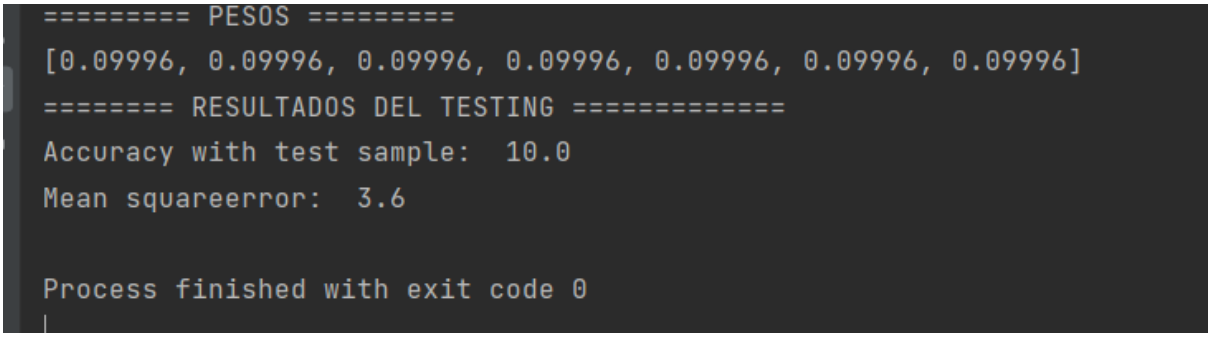
Prueba con el archivo *test* y un learning rate de 0.1, 70% de datos de entrenamiento y 3 épocas.



En el ejemplo anterior se realizaron las 3 épocas sin llegar a la convergencia.
Ahora vamosa ver otra corrida con los mismos hiperparámetros.



Como podemos ver, los pesos llegaron a la convergencia antes de las 3 épocas y los resultados son bastante diferentes. Esto es debido a que toman muestras aleatorias del dataset. Incluso con hiperparámetros diferentes puede llegarse a la convergencia rápido como en el ejemplo siguiente (alpha: 0.1, entrenamiento: 85% y 2 épocas)



Además, también es importante mencionar que el dataset es muy pequeño como para realizar más épocas.

Análisis de accuracy y error

Este análisis se realiza después de que el algoritmo finaliza, se calcula el accuracy de los pesos con el dataset de pruebas y se calcula el mean square error también. Todo esto nos dará un panorama sobre cómo está nuestro modelo y si debemos ajustar nuestros hiperparámetros.