ADDIS ABABA UNIVERISITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

CENTER OF INFORMATION TECHNOLOGY AND
Engineering

RBFN

PREPARED BY:
Ashenafi Chufamo

SUBMITTED TO:
D.r Beakal Gizachew

Jan 2024

Introduction

This report documents the implementation and evaluation of two algorithms: K-means clustering and Least Mean Squares (LMS) for function approximation and classification problems. The implementation includes both custom implementations and the use of corresponding sci-kit-learn algorithms. The evaluation includes comparing the results obtained from the custom implementations with the results obtained from the scikit-learn algorithms.

Function Approximation problem

For the function approximation problem, the following steps were performed:

- A set of input data points X was generated using the np.linspace function, ranging from 0 to 10 with 100 data points.
- The corresponding target values y were generated by applying the np.sin function to the input data points.
- K-means clustering was applied to the input data using the custom k_means function and the scikit-learn KMeans algorithm. The number of clusters (K) was set to 3.
- LMS was applied to the input data and target values using the custom lms function and the scikit-learn LinearRegression algorithm.
- The results were compared by printing the centroids obtained from the custom K-means algorithm (custom_centroids), the centroids obtained from the scikit-learn K-means algorithm (sklearn_kmeans.cluster_centers_), the weights obtained from the custom LMS algorithm (custom_weights), and the weights obtained from the scikit-learn Linear Regression algorithm (sklearn_regression.coef_).

Results

Custom K-means centroids

[[1.56565657]
[4.84848485]
[8.28282828]]
These are the centroids obtained from the custom K-means algorithm. Each row represents a centroid, and the number of columns depends on the dimensionality of the data.

Scikit-learn K-means centroids

[[5.1010101]
[1.66666667]
[8.43434343]]
These are the centroids obtained from the scikit-learn K-means algorithm.

Custom LMS weights

[-4.38600399e+150 -2.91699831e+151]
These are the weights obtained from the custom LMS (Least Mean Squares) algorithm. The custom_weights variable contains the learned weights for the linear regression problem.

Scikit-learn Linear Regression weights

[[-0.01730202]]
These are the weights obtained from the scikit-learn Linear Regression algorithm.

Classification Problem

For the classification problem, the following steps were performed:

- A synthetic dataset (X, y) was generated using the make_classification function from scikit-learn. The dataset consists of 100 samples with 2 informative features and 2 classes.
- K-means clustering was applied to the input data using the custom k_means function and the scikit-learn KMeans algorithm. The number of clusters (K) was set to 2.
- LMS was applied to the input data and target values using the custom LMS function and the scikit-learn LogisticRegression algorithm.
- Predictions were made using the learned models, and the results were compared by printing the centroids obtained from the custom K-means algorithm (custom_centroids), the centroids obtained from the scikit-learn K-means algorithm (sklearn_kmeans.cluster_centers_), the weights obtained from the custom LMS algorithm (custom_weights), the weights obtained from the scikit-learn Logistic Regression algorithm (sklearn_classification.coef_), and the accuracies of the models.

Results

Custom K-means centroids

[[-1.29452448 -0.78043004]
[0.88963137 0.48091756]]
Explanation: These are the centroids obtained from the custom K-means algorithm.

Scikit-learn K-means centroids

[[0.82894503 0.50091753]
[-1.31270276 -0.87349737]]
Explanation: These are the centroids obtained from the scikit-learn K-means algorithm.

Custom LMS weights

[-2907386.36602777 52173113.93367405 42752586.39743453]
Explanation: These are the weights obtained from the custom LMS (Least Mean Squares) algorithm.

Scikit-learn Logistic Regression weights

[[3.23218612 -0.84604963]]
Explanation: These are the weights obtained from the scikit-learn Logistic Regression algorithm.

Custom K-means accuracy

0.84
Explanation: This is the accuracy of the custom K-means algorithm achieved by comparing the predicted labels (custom_labels) with the true labels (y).

Scikit-learn K-means accuracy

0.8
Explanation: This is the accuracy of the scikit-learn K-means algorithm achieved by comparing the predicted labels (sklearn_labels) with the true labels (y).

Custom LMS accuracy

0.84
Explanation: This is the accuracy of the custom K-means algorithm achieved by comparing the predicted labels (custom_labels) with the true labels (y).

Scikit-learn LMS accuracy

0.99
Explanation: This is the accuracy of the scikit-learn K-means algorithm achieved by comparing the predicted labels (sklearn_labels) with the true labels (y).

Conclusion

In this report, I have implemented and evaluated K-means clustering and Least Mean Squares (LMS) algorithms for function approximation and classification problems.

For the function approximation problem, I applied K-means clustering to the input data and compared the centroids obtained from the custom implementation with the centroids obtained from the scikit-learn algorithm. I also applied LMS to the input data and target values and compared the weights obtained from the custom implementation with the weights obtained from the scikit-learn algorithm. The results showed that the centroids and weights obtained from the custom implementations were different from those obtained from the scikit-learn algorithms.

For the classification problem, I applied K-means clustering to the input data and compared the centroids obtained from the custom implementation with the centroids obtained from the scikit-learn algorithm. I also applied LMS to the input data and target values and compared the weights obtained from the custom implementation with the weights obtained from the scikit-learn algorithm. Additionally, I calculated the accuracy of the models. The results showed that the centroids obtained from the custom K-means algorithm were different from those obtained from the scikit-learn K-means algorithm. The weights obtained from the custom LMS algorithm were also different from those obtained from the scikit-learn Logistic Regression algorithm. Furthermore, the accuracy of the custom K-means algorithm was higher than the accuracy of the scikit-learn K-means algorithm. This is because my custom k-Means function had a better initialization method that made it easier for optimal centroid. But for the LMS the scikit-learn algorithm had an impresive 99 percent accuracy which mine had only 84%. This is by the Hyperparameters it used, these parameters include the number of itterations used and the convergence threshold.