Derrick Nguyen

Qqn

CSDS391: Introduction to Artificial Intelligence

**Programming Assignment 2 Write Up**

**Problem 1:**

Class K\_means(): perform custom k-means clustering algorithm on the iris dataset. The data is in CSV form and in the data folder.

Necessary libraries are imported: pandas, numpy, matplotlib, seaborn, sklearn, random

Graphical user interface, text

Description automatically generated

The class methods include:

* initialize\_centroids: which randomly initializes centroids
* initialize\_centroids\_from\_data: which initializes centroids from the data
* assign\_clusters: which assigns clusters based on the current centroids
* valid\_centroids: check if a centroid has no data points
* find\_updated\_centroids: which calculates updated centroids based on the assigned clusters
* find\_distortion: which calculates the distortion of the clustering result converged, which checks if the algorithm has converged
* update\_centroids: which updates the centroids
* converged: check if the centroid changes location
* iterate: which iterates until convergence is achieved.

Plot part a)

Chart, scatter chart

Description automatically generated

Distortion plot part b)

Chart, line chart

Description automatically generated

Learning Process plot part c)

* number of clusters: 3
* Chart, scatter chart

  Description automatically generated
* Number of clusters: 2

Chart, scatter chart

Description automatically generated

Decision boundary plot part d)

Chart, scatter chart

Description automatically generated

In order to plot the decision boundary, we have to pick 2 features of the data points and then find the max value and min value of each feature to create a grid that cover the range of those values. Then we put data points on the plot and calculate the Euclidian distance to the centroid to assign data points into clusters and change the value of the grid based on this assignment. and set the color. Below is the code snapshot

Text

Description automatically generated

**Problem 2**

Class ANN(): implements a neural network to iris data

Input: vector data, initial parameters, pattern classes.

The class methods include:

* activation\_function(self, input): method that implements the sigmoid activation function.
* forward\_propagation(self): method that computes the output of the neural network using forward propagation.
* back\_propagation(self): method that performs backpropagation to compute the gradient of the loss function with respect to the weights.
* calculate\_mse(self): method that computes the mean squared error (MSE) of the neural network. update\_weight(self): method that updates the weights of the neural network using the calculated gradient.
* converged(self): method that checks whether the weights have converged to their optimal values.
* train(self): method that trains the neural network by iterating over the dataset and updating the weights. The training stops after a fixed number of iterations. The method returns the weight change over time.

1. Find\_min\_squared:

Each data node is calculated using forward propagation using non-linear sigmoid activation function and then mean squared error of the total data sample will be calculated using calculate\_mse method.

Text

Description automatically generated

Text

Description automatically generated

**Final result: 0.25**

1. Decision boundary plot

For weight = 0

Chart

Description automatically generated

For weight in range [-1,1]

Chart

Description automatically generated

C and d)

Text, letter

Description automatically generated

Text, letter

Description automatically generated