FIT5201 Machine Learning – Assessment 2 Report

MD. Saadman Hossain

31043313

Part A: Document Clustering

Question 1 [EM for Document Clustering, 40 Marks]

1. Derive **Expectation** and **Maximization** steps of the hard-EM algorithm for Document Clustering.

   In Expectation Minimization (EM) algorithm for Document Clustering, we need to find the estimation of the clusters for each document and maximize the parameters in the process that best describe the cluster assignments for the document.

   Every document $\boldsymbol{d_n}$ is treated as a set of words in that document and their order does not matter. Also, it is assumed that every word that is present in a document $\boldsymbol{d_n}$ are part of dictionary $\boldsymbol{A}$.

   Model parameters for such a model are mixing parameter $\boldsymbol{\rho}$ for each cluster $\boldsymbol{k}$ and set of word proportions $(\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, .., \boldsymbol{\mu_k})$ representing word proportions for each cluster $\boldsymbol{K}$. We represent set of all the model parameters as $\boldsymbol{\theta} \coloneqq (\boldsymbol{\rho},\ \boldsymbol{\mu_1}, \boldsymbol{\mu_2}, .., \boldsymbol{\mu_k})$

   Expectation Step:

   a. For each document $\boldsymbol{d_n}$ and for each cluster $\boldsymbol{k}$, we need to calculate the responsibility factor. That is, the posterior probability $\boldsymbol{\gamma(Z_n, k)}$,of document $\boldsymbol{d_n}$ belonging to cluster $\boldsymbol{K}$ given parameters $\boldsymbol{\theta^{old}}$.

   $$\boldsymbol{\gamma(Z_n, k) = P\left(Z_{n,k} = 1 \middle| d_n, \theta^{old}\right) = \rho_k \prod_{w \in A} \mu_{k,w}^{c(w,k)}}$$

   b. In this step we do hard assignment for the document $\boldsymbol{d_n}$ to cluster $\boldsymbol{k}$ having maximum posterior probability among all clusters $\boldsymbol{K}$. We set $\boldsymbol{Z_{n,k} = 1}$ for to cluster having maximum posterior probability and set $\boldsymbol{Z_{n,k`} = 0}$ for other clusters.

   Maximization Step:

   Depending on the results from our expectation step we get the posterior probabilities $\boldsymbol{\gamma}$ for each document and for each cluster. Now in Maximization step we will try to maximize the likelihood by updating our parameters $\boldsymbol{\theta^{new} \leftarrow \theta^{old}}$

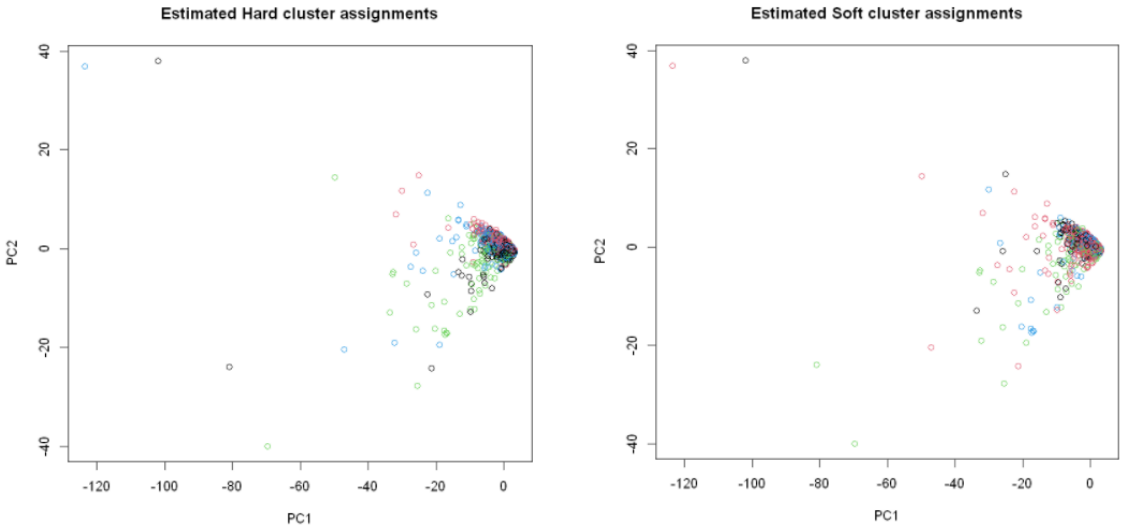
a. Mixing parameter $\boldsymbol{\rho_k}$ as,

$$\boldsymbol{\rho_k} \coloneqq \sum_{n=1}^{N} \boldsymbol{\gamma}(\boldsymbol{Z_n}, \boldsymbol{k})$$

b. Word proportion parameters for each cluster $\boldsymbol{k}$ as,

$$\boldsymbol{\mu_{k,w}} \coloneqq \frac{\sum_{n=1}^{N} \boldsymbol{\gamma}(\boldsymbol{Z_n}, \boldsymbol{k})}{\sum_{w \in A} \sum_{n=1}^{N} \boldsymbol{\gamma}(\boldsymbol{Z_n}, \boldsymbol{k})}$$

4. Perform a PCA on the clustering that you get based on the hard-EM and soft-EM algorithms. Report how and why the hard and soft-EM are different.

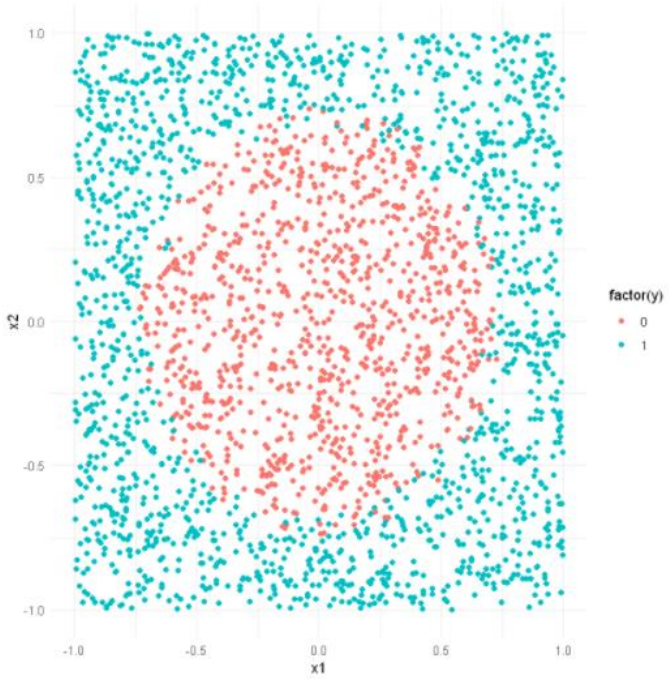


Hard expectation maximization can assign fix cluster to each document, setting $\boldsymbol{Z_{n,k}} = \mathbf{1}$ for the cluster. While soft expectation maximization provides posterior probability of assignments of a particular datapoint to a cluster. Therefore, Soft EM best describes the data when there's not clear decision for which cluster the data point should belong to. In such cases, Soft EM provides the partial assignment of a datapoint. Based on this knowledge and graphs given above, we can conclude that boundaries of cluster created with Hard EM are well defined, whereas clusters created with Soft EM algorithm does not.

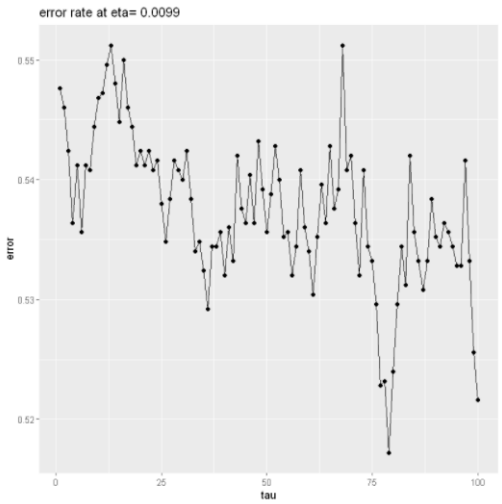
Part B. Neural Network vs. Perceptron

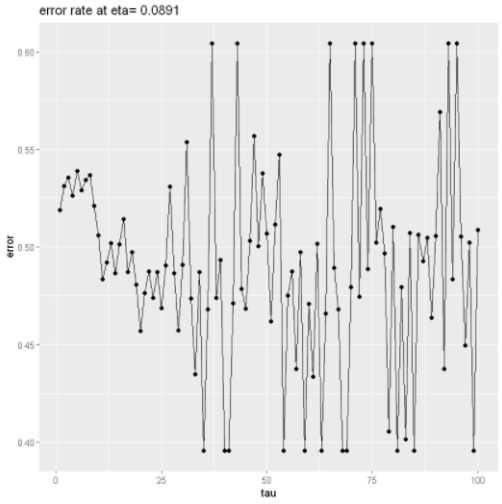
Question 2 [Neural Network's Decision Boundary, 30 Marks]

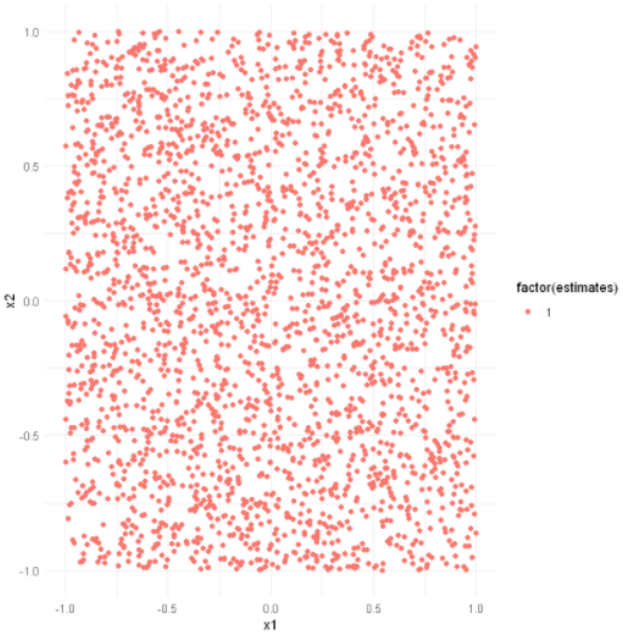
1.Load **Task2B_train.csv** and **Task2B_test.csv** sets, plot the training data with classes are marked with different colours, and attach the plot to your PDF report.



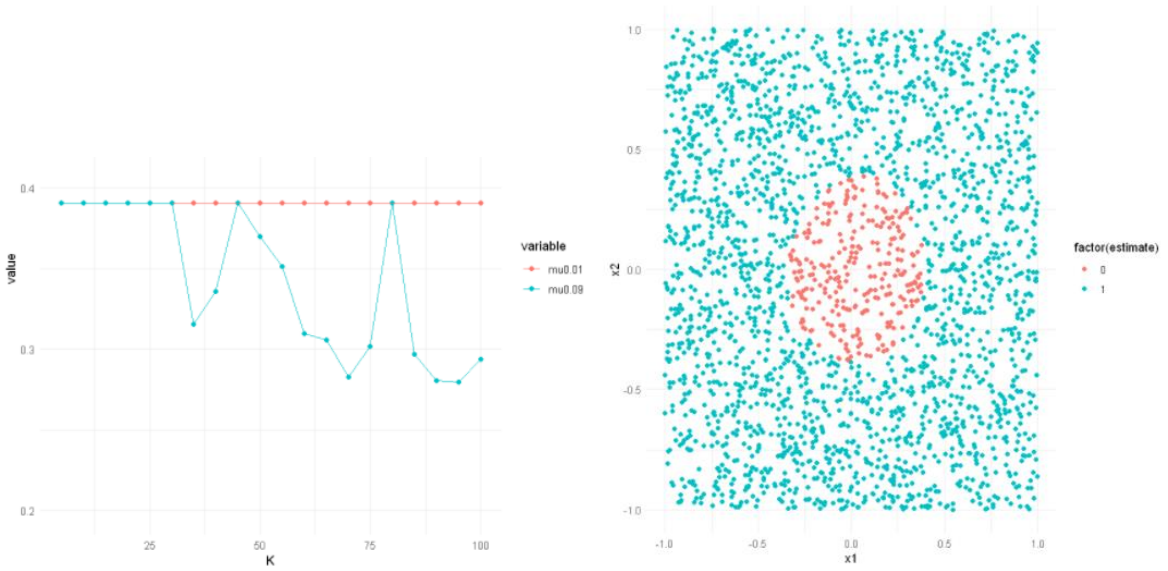
2.Train two perceptron models on the loaded training data by setting the learning rates η to .01 and .09 respectively, using a code from Activity **3.1**. Calculate the test errors of two models and find the best η and its corresponding model, then plot the test data while the points are coloured with their estimated class labels using the best model that you have selected; attach the plot to your PDF report.







3.For each combination of K (i.e., number of units in the hidden layer) in {5, 10, 15, …, 100} and μ (learning rate) in {0.01, 0.09}, run the 3-layer Neural Network given to you in Activity **5.1** and record testing error for each of them (40 models will be developed, based on all possible combinations). Plot the error for μ 0.01 and 0.09 vs K (one line for μ 0.01 and another line for μ 0.09 in a plot) and attach it to your PDF report. Based on this plot, find the best combination of K and μ and the corresponding model, then plot the test data while the points are coloured with their estimated class labels using the best model that you have selected; attach the plot to your PDF report.



4.In your PDF report, explain the reason(s) responsible for such difference between perceptron and a 3-layer NN by comparing the plots you generated in Steps II and III.

Answer-

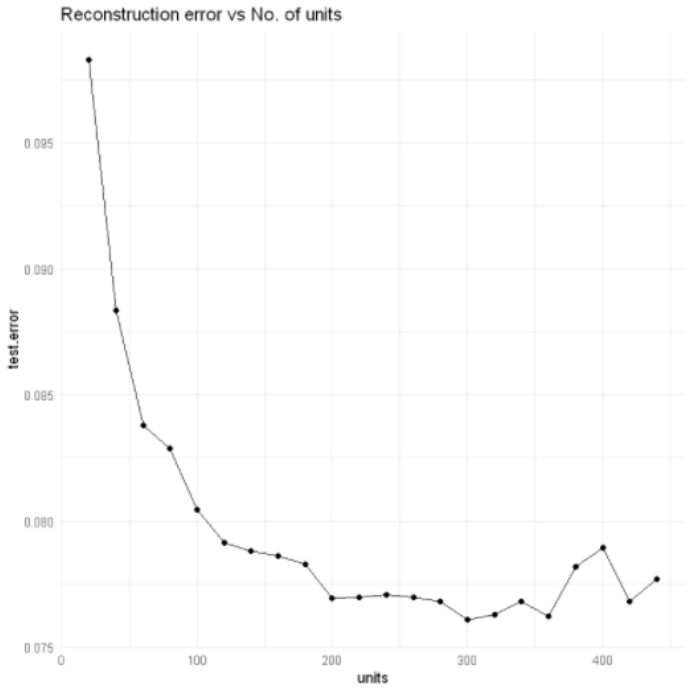
Perceptron model can use linear equations as linear boundary to separate and classify the data points. Thus, it only works well for data which is linearly separable. As we can see in the plot for Q1, the data is not linearly separable therefore, perceptron model is not able to define a boundary between data points. The resulting model is a more biased model.

In the other case, neural networks contain a network of interconnected neurons where each neuron represents a non-linear equation. Thus, neural network can fit very well even on complex problems and can learn data where data points are separated by non-linear boundary. The increasing number of hidden layers and/or neurons define its ability to learn the complex problems.

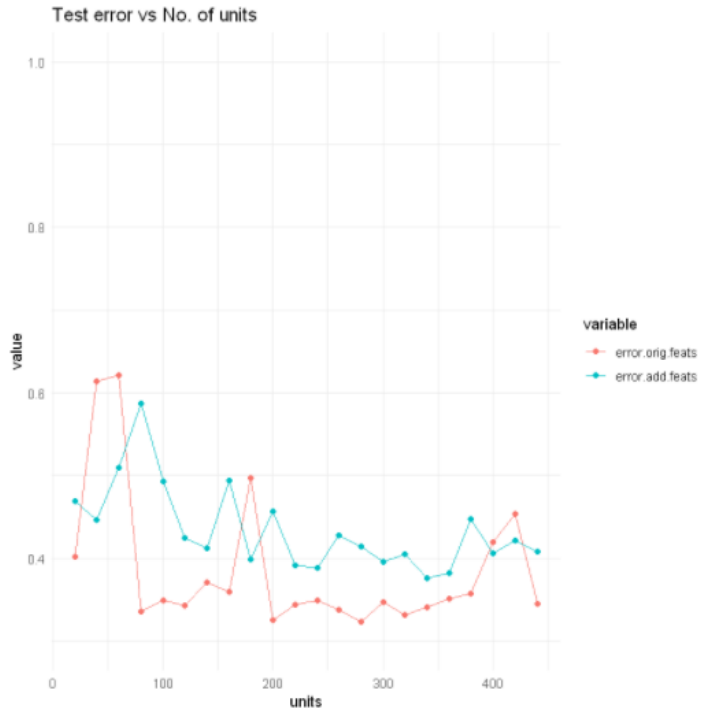
Part C. Self-Taught Learning

Question 3 [Self Taught Neural Network Learning, 30 Marks]

3.For each model in Step II, calculate and record the reconstruction error which is simply the average (over all data points while the model is fixed) of Euclidian distances between the input and output of the autoencoder (you can simply use "h2o.anomaly()" function). Plot these values where the x-axis is the number of units in the middle layer and the y-axis is the reconstruction error. Then, save and attach the plot to your PDF report. Explain your findings based on the plot in your PDF report.



6.Plot the error rates for the 3-layer neural networks from Step IV and the augmented self-taught networks from Step V, while the x-axis is the number of hidden neurons and y-axis is the classification error. Save and attach the plot to your PDF report. In your pdf, explain how the performance of the 3-layer neural networks and the augmented self-taught networks is different and why they are different or why they are not different, based on the plot.



explaining how the performance of the 3-layer neural networks and the augmented self-taught networks is different and why they are different or why they are not different.

Autoencoders when it comes to unlabelled data, learns the features of the data well and can reconstruct the original data well. It is assumed that, given more data to complex model then it can learn well and provide better prediction results. Thus, we can use adding new additional features learned by the autoencoders with to our original labelled data to train our classifier.

In the Step 4, we trained our 3-layer NN with original labelled data. As the complexity of model increased, it was able to learn more features and give much better predictions. Although increasing model complexity on simple data may lead to overfitting.

In Step 5, we added features from the autoencoder, to train our model expecting an increase in performance. As the training data features increased, our neural 3-layer NN model was not large or deep enough to learn additional complex features and remained underfitted.

Comparing plots from Step 4 and 5, 3-layer NN learned all the possible features with given complexity, whereas augmented self-taught network given additional complex features was not able to learn well as expected than the simple 3-layer NN and ended up giving quite similar performance.