CLUSTER ANALYSIS ON FASHION-MNIST DATASET

Jyoti Sinha
Department of Computer Science
State University of New York at Buffalo
jsinha@buffalo.edu

Abstract

Clustering is a type of unsupervised learning method. In Clustering, the data points are divided into the number of groups such that the data points in the same group are similar to another data points. While the dis-similar data points comes in other groups. In this project, I used various methods to cluster the Fashion-MNIST dataset. In the first part, I used K-Means algorithm to cluster the dataset using Sklearns library. In the second part, I build the Auto-encoder based K-Means clustering model to cluster the condensed representation of the unlabeled fashion MNIST dataset using Keras and SKlearns library. In the third part, I build an Auto-Encoder based Gaussian Mixture Model to cluster the datasets. In my study, I found that the accuracy for my third part is highest and the accuracy for the first part is lowest.

Introduction:-

Clustering is the process of grouping similar entities together. It is an unsupervised machine learning technique. The goal of this type of unsupervised machine learning is to find similarities among the data points and group the similar data points together. Clustering provides us insight into underlying patterns of different groups. Clustering also helps us to reduce the dimensionality of the data when dealing with the large amount of the variables. K-Means algorithm is one of the simplest and most popular unsupervised machine learning algorithm. K-Means algorithm find out the k number of centroids and then locate the every data points to the nearest clusters and tried to keep the centroids as small as possible. In the K-Means algorithm , we randomly select the centroids starting from the first group and then perform the repetitive calculations to optimize the positions of the centroids. K-Means clustering is widely used for the data cluster analysis.

Dataset:-

Fashion-MNIST is a dataset of Zalando's article image which consists of 60,000 training examples and 10,000 testing examples. Each example is 28*28 grayscale image and has labels from 10 classes. Each image is 28 pixels in height and 28 pixels in width and a total of 784 pixels. Each pixel is associated with a single value which indicates the lightness and darkness of the pixel. The pixel value is an integer whose values lies between 0 and 255. There are a total of 785 columns in the training and testing datasets.

Labels:-

Each example of training and testing is associated with one of the following labels:-

- T-Shirt/top
- Trouser
- Pullover
- Dress

- Coat
- Sandal
- Shirt
- Sneaker
- Bag
- Ankle boot

In the dataset, each row is a separate image and column 1 is the class label.

Pre-processing:-

The image of the dataset are grayscale images with pixel values varying from 0 to 255 with the dimension of 28*28. So, it is important to pre-process the data before feeding it into the model. So, I converted each 28*28 images into a matrix of 28*28*1 and it is fed into the network. Then, I converted the the dataset into float32 format. Then, I rescale the pixel value in range 0-1 inclusive.

Basic Terminologies :-

Concept of K-means algorithm and Clustering:-

Clustering is the collection of the data points that are aggregated on the basis of certain similarities. It deals with a target number which is basically the number of centroid required in the dataset. Thus, K-Mean algorithm find out the total number of the centroid and provide every data to the nearest clusters and tries to keep the centroid small. The means in K-means refers to the averaging of the data points.

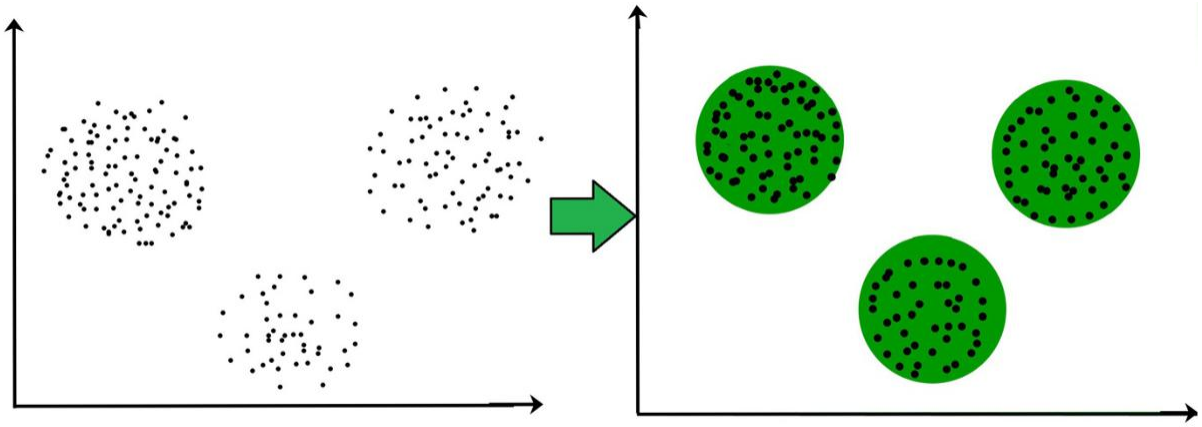


Image of Clustering

Working of K-Means algorithm:-

K-Means algorithm in the data mining begins by randomly selecting the first group of centroids which is the initial point for every cluster and then the position of the centroids are optimized by repetitive calculations.

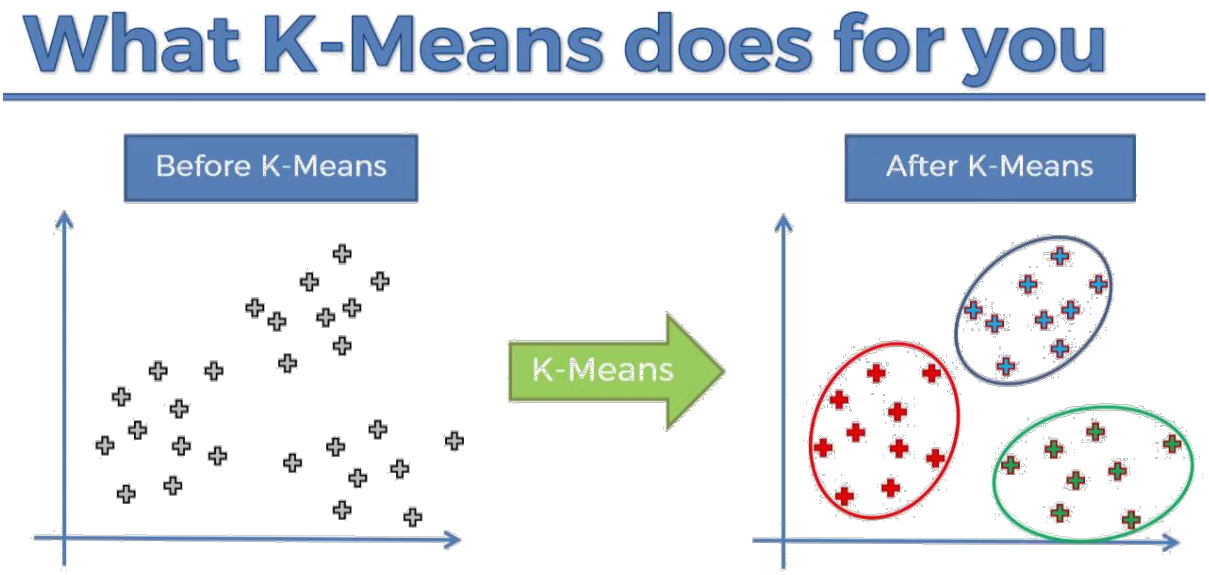


Image explaining the concept of K-means algorithm

Autoencoder:-

An autoencoder is just a type of neural network that is used to learn efficient data codings in the supervised way. The main goal of the autoencoder is to reduce the dimension by training the network in order to eliminate the noise.
Autoencoder is divided into two parts. The first is encoder part and the other is decoder part.
The role of the encoder is to compress the input data to the lower dimensional features. In my study, the encoder has compressed the 784 pixel of Mnist image into ten floating numbers as features. The decoder part, takes the compressed image as input and reconstruct it to as close to the original image.

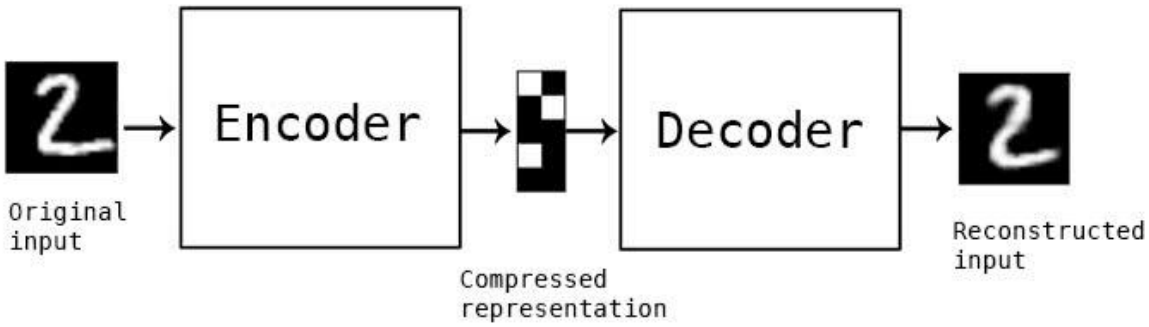


Image of encoder and decoder in the autoencoder.

In the third task, I performed clustering using Gaussian Mixture Model for learning of the clusters. Gaussian Mixture model is a model which also cluster the unlabelled data in the same way as K-means. But, there are certain advantages of Gaussian Mixture model over the K-means. K-means doesn't account for the variance. Another disadvantage is that k-means algorithm just place a circle

at the centre of each cluster. K-means also perform hard classification where as Gaussian Mixture perform soft classification. This will work when the data is circular but it would not work when the data is not circular. K-means can do the classification of the data points and tells that which data points belongs to which cluster but can't tell about the probability that a given point will belongs to each of the possible clusters.

Auto-Encoder with K-Means Clustering:-

The main goal of K-means algorithm is to select the centroid that reduces the inertia or within-cluster sum of squares criterion:-

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

Inertia is basically recognized as how the internally the coherent clusters are. Interia is not a normalized metric: we just know the lower values are better and zero is optimal. But, in very high dimensional space, Euclidean distances tends to become inflated. This problem can be reduced by using dimensionality reduction algorithm like Auto -encoder or Principal component analysis

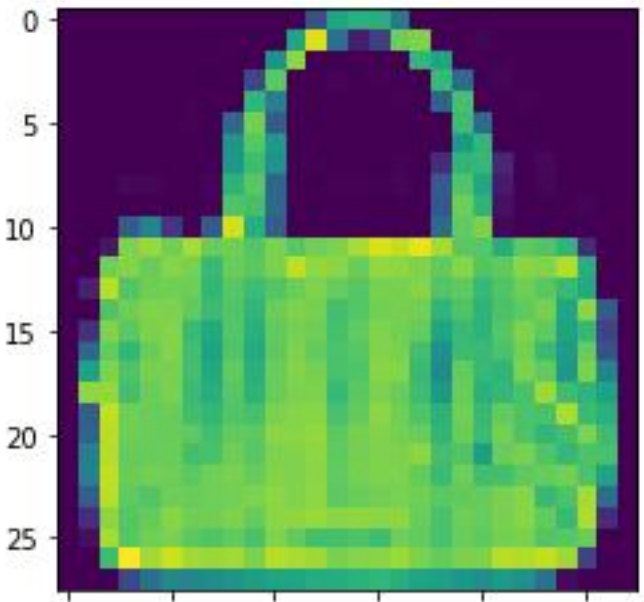
Auto-Encoder with GMM Clustering:-

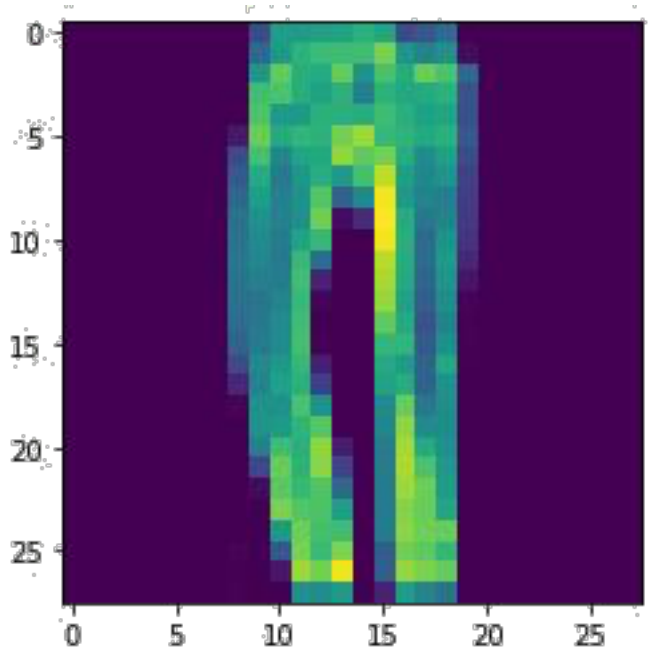
The Gaussian mixture is a probabilistic model that assumes that all the data points are generated from mixture of finite numbers of Gaussian distribution. The Gaussian Mixture object implement the expectation-maximization algorithm for fitting mixture of Gaussian model. It can draw confidence ellipsoids for multivariate model and compute the Bayesian information criterion the access the number of clusters in the data.
In my project, I have used SELU for the Gaussian mixture model and 'Selu' is a activation function for neural networks.

Advantage of Selu over Relu:-
1)Selu can't die but Relu can die.
2) There is no problem with vanishing gradients in Selu .This is the reason that Selu can enable deep neural networks.
3) Selu learn faster and better as compared to the other activation functions.

Diagrams:-

Creation of image using the datasets:-

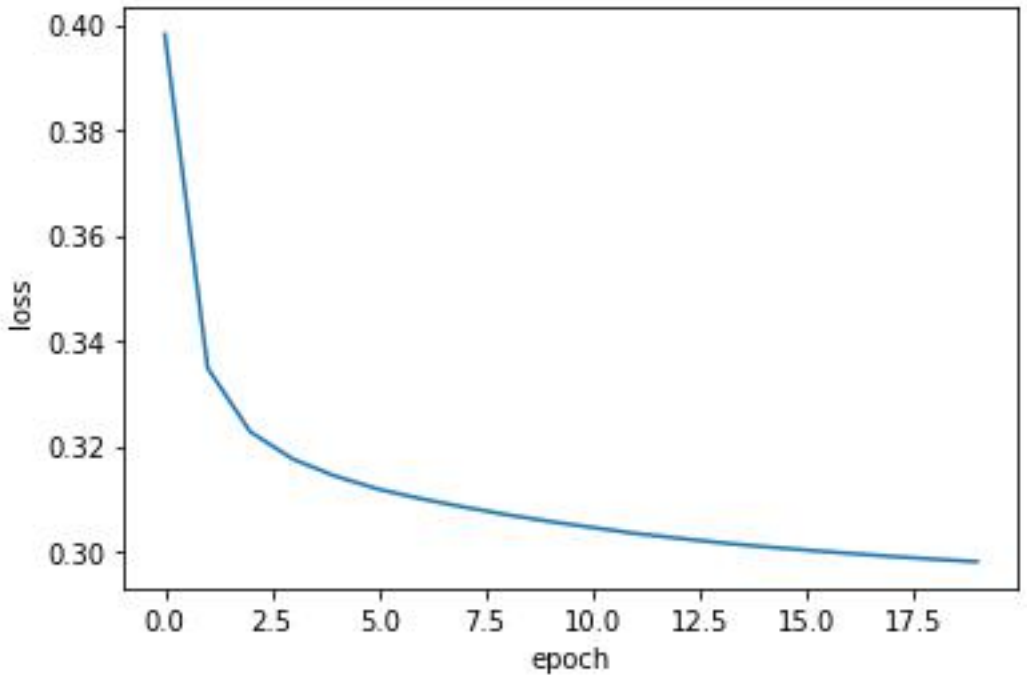




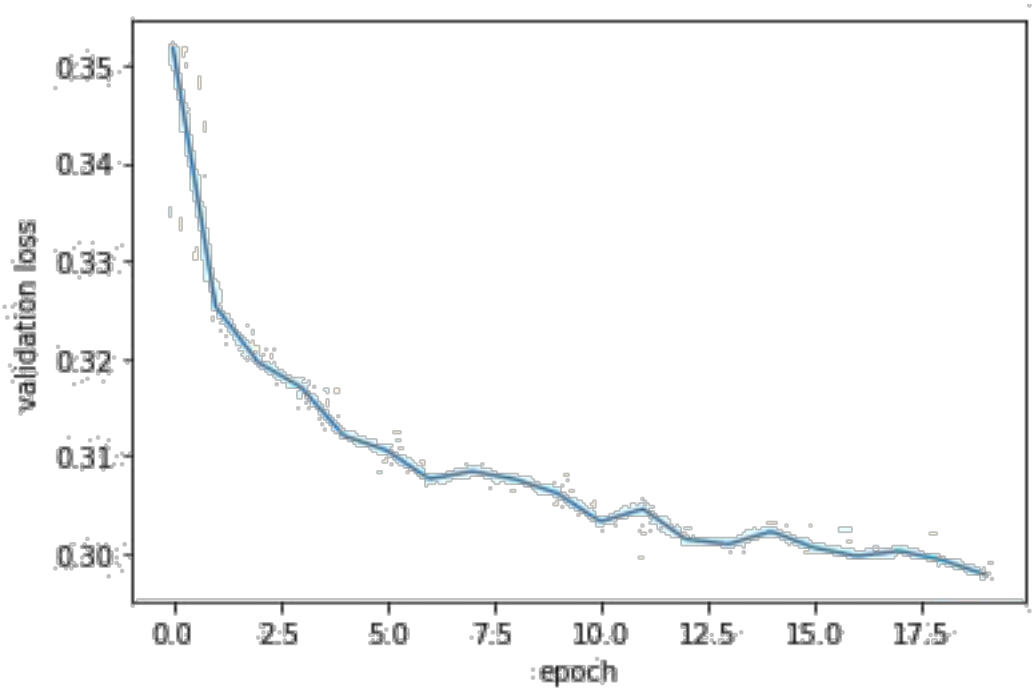
Result:-

In the first task, I get `0.5368` % accuracy.

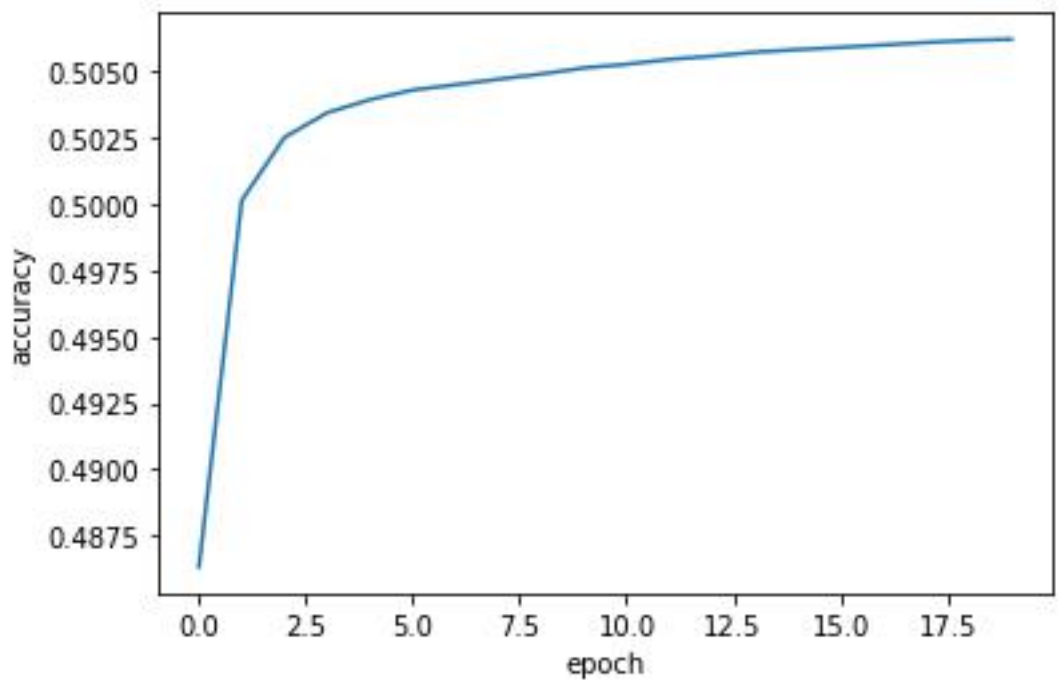
Graphs of Auto-Encoder using K-Means:-



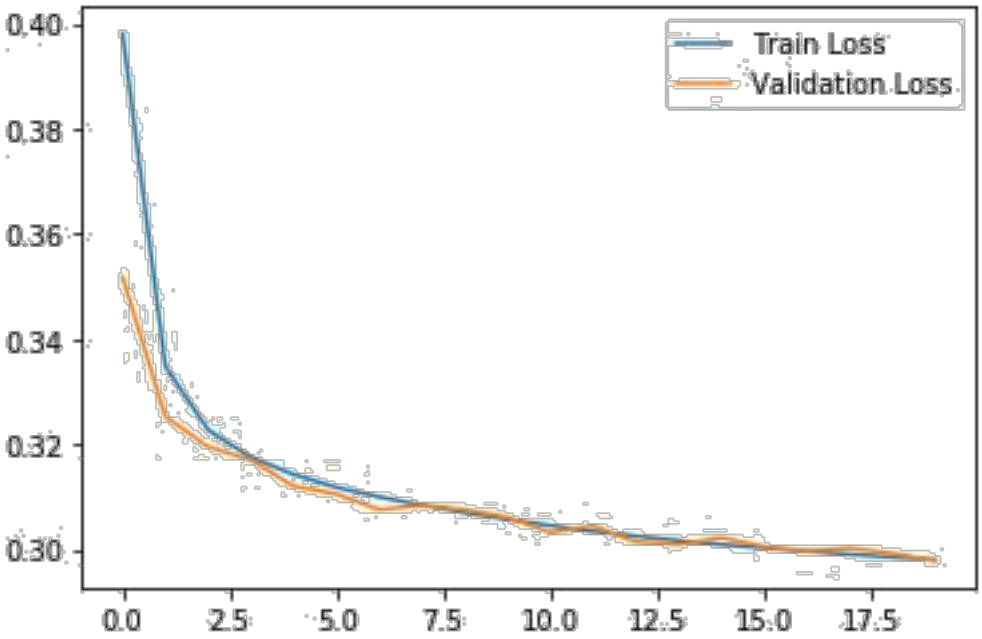
Plot of epoch vs loss



Plot of epoch vs validation loss



Plot of epoch vs accuracy



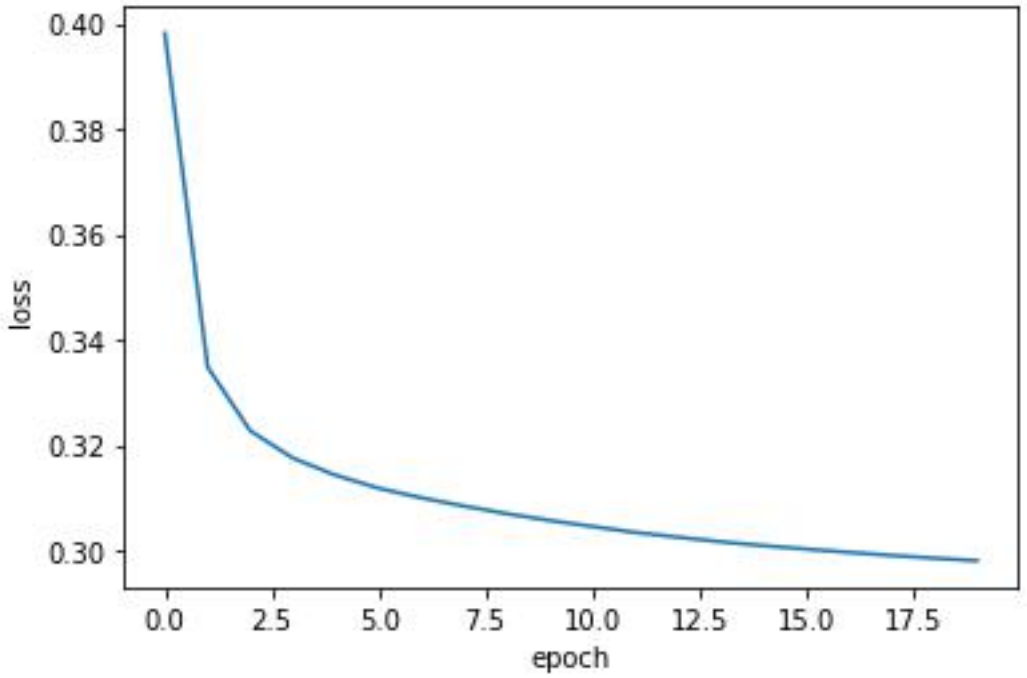
Plot of Train loss and Validation loss

In the second task, I get `54.315%` accuracy.

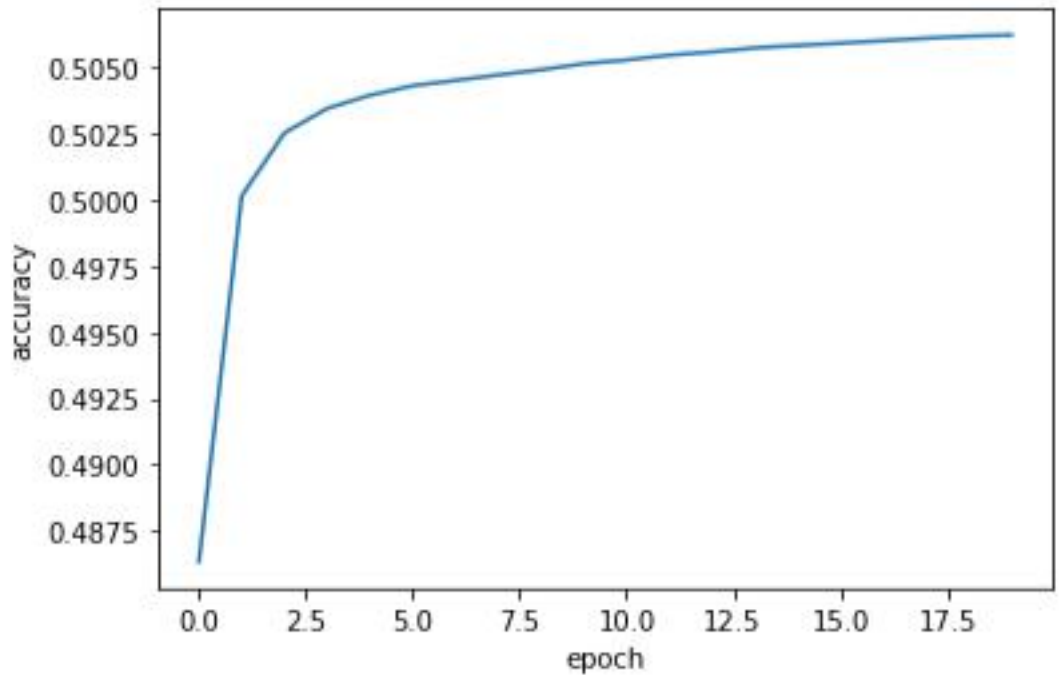
```
[[ 22   1 261 615   0   5  37  52   2   5]
 [891   0  32  53   0   0   1  13   1   9]
 [  3   0 319  18   0   3 250  36   2 369]
 [487   0 141 292   0   4   5  64   0   7]
 [ 22   0 129 126   0   5 128  31   1 558]
 [  0 189  10   0  52   0   0 695  54   0]
 [ 11   0 344 195   1   2 137  80   4 226]
 [  0 753   0   0   4   0   0  78 165   0]
 [  3  25  66   3   2 426 123  50 294   8]
 [  0  19   6   0 487   2   0  40 445   1]]
```

Confusion matrix

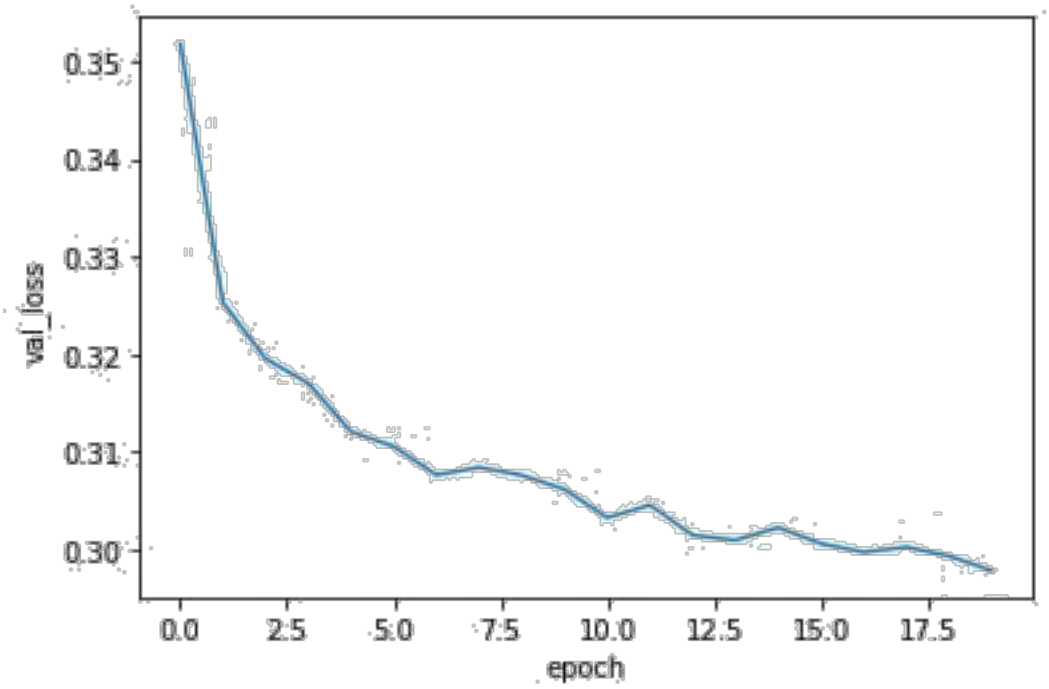
Graphs of Auto-Encoder using Gaussian Mixture Model:-



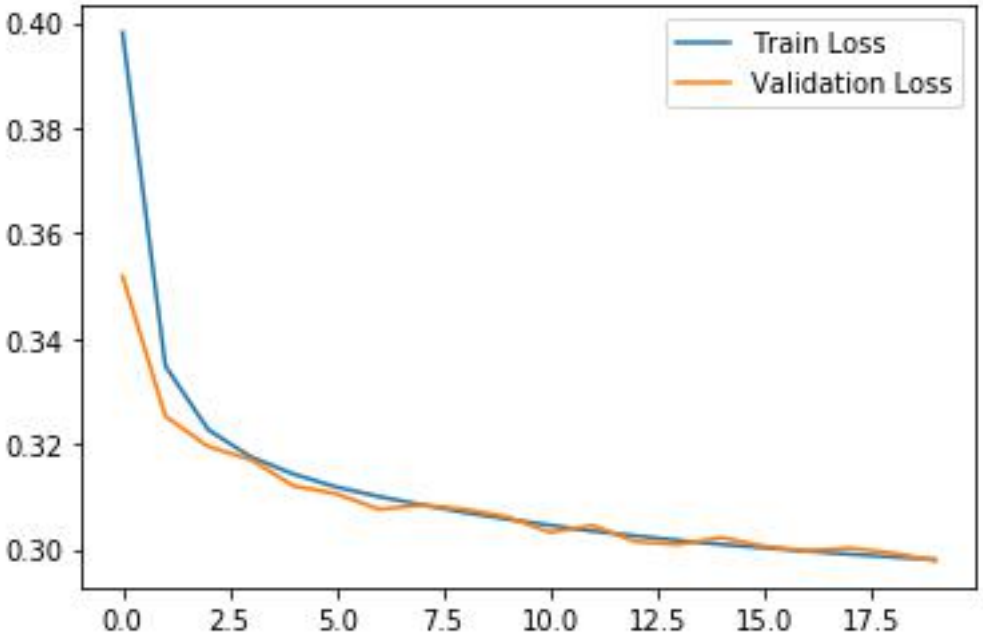
Plot of epoch vs loss



Plot of epoch vs accuracy



Plot of epoch vs validation loss



Plot of Training loss and Validation

In the third task, I get 56.015 % accuracy.

```
[[  42   2   38   5   1 241   5 651   15   0]
 [914   1    1   0   0  21  10  46    7   0]
 [  4   2  241   3   0 193 530  17   10   0]
 [583   0    5   4   0 105   9 283   11   0]
 [ 25   1  116   5   0 103 621 121    8   0]
 [  0  59    0   0 470  10   0   0  409  52]
 [ 15   4  130   2   0 255 364 212   17   1]
 [  0 164    0   0 818   0   0   0   14   4]
 [  3 308  123 430  17  70   8   3   36   2]
 [  0 458    0   2  20   6   1   0  28 485]]
```

Confusion matrix

Conclusion:-

The auto-encoder based Gaussian mixture model has highest accuracy than the auto-encoder based K-Means algorithm. The K-Means algorithm alone provides the less accuracy. Thus, auto-encoder based Gaussian mixture model is best in the terms of accuracy than the auto-encoder based Gaussi an mixture model algorithm and K-Means alone.

Reference:-

https://towardsdatascience.com/unsupervised-learning-of-gaussian-mixture-models-on-a-selu-auto-encoder-not-another-mnist-11fceccc227e

https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e

https://stats.stackexchange.com/questions/89809/is-it-important-to-scale-data-before-clustering

https://stackoverflow.com/questions/20027645/does-kmeans-normalize-features-automatically-in-sklearn

Slide of the professor

Pdf of project 3 description