Project-3: CLUSTER ANALYSIS USING UNSUPERVISED LEARNING

Shubhangi Mane
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
UB Id: 50295705
smane2@buffalo.edu

ABSTRACT

This project aims at performing unsupervised learning using KMeans Clustering and Gaussian mixture model. We use Fashion MNIST dataset, which is a dataset used for apparel recognition. In this project we perform three tasks.

1. We use Kmeans Algorithm to cluster original dataset using sklearns library
2. We build an Auto-Encoder based Kmeans Clustering Model to cluster the unlabelled dataset using Keras and Sklearns library.
3. We build an Auto-Encoder based Gaussian Mixture Model to cluster the unlabeled dataset using Keras and Sklearns library

Initially we record the clustering accuracy for the KMeans model trained using Sklearns, then an auto encoder model is built using Keras library. The auto encoder model here, compresses each image into latent floating point values. In our tasks we have 10 clusters which represents each label in the dataset. By training the auto-encoder, we have the encoder learned to compress each image into latent floating point values. Then cluster centroids are identified and compressed latent codes are clustered using Kmeans and Gaussian mixture Model. We have recorded the confusion matrix and accuracy is identified for each of the tasks.

INTRODUCTION:

Clustering is the process of dividing the entire data into clusters based on the patterns in the data. Clustering is an unsupervised learning, as we don't know the target to predict, we look at data and group the data with similar observations.

KMEANS:

K-means clustering is unsupervised machine learning algorithms that forms clusters of data based on the similarity between data instances. For this particular algorithm to work, the number of clusters has to be defined beforehand. The K in the K-means refers to the number of clusters. After the centroids are identified it then allocates every data point to the nearest cluster.

AUTO ENCODER:

Autoencoding is a data compression algorithm where the compression and decompression functions are data-specific, loss, and learned automatically from examples. To build an auto-encoder, you need three things: an encoding function, a decoding function, and a distance function between the amount of information loss between the compressed representation of your data and the decompressed representation.

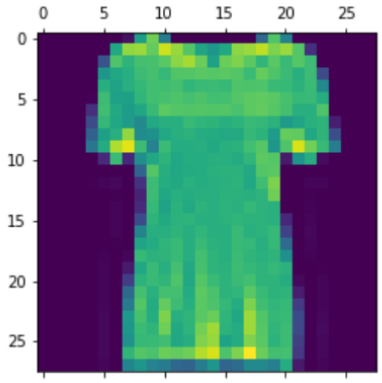
GAUSSIAN MIXTURE MODEL:

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. The Gaussian Mixture object implements the expectation-maximization (EM) algorithm for fitting mixture-of-Gaussian model.

DATASET:

Fashion MNIST dataset contains 70000 instances with 784 attributes. Each image is stored in computers as matrix of pixels, where each pixel is one or more number. The values are in the range

0 to 255, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. Each example is a 28X28 grayscale image, associated with each is a label from 10 classes. Fashion MNIST reader notebook is used to load the dataset. We have used neural networks to train the model. This analysis aims to observe which features are most helpful in predicting label of images and to see general trends that may aid us in model selection and hyper parameter selection.



The labels associated with the images are as follows:

Label	Image Type
1	T-Shirt/top
2	Trouser
3	Pullover
4	Dress
5	Coat
6	Sandal
7	Shirt
8	Sneaker
9	Bag
10	Ankle Boot

SPLITING THE DATASET:

We have divided the data set into two parts namely Training set, Test set. Training input set contains 60000 instances with 784 attributes and Training output set contains 60000 instances with 1 attribute. Test input set each contain 10000 instances with 784 attributes and Test output set contains 10000 instances with 1 attribute.

The model is initially fit on a Training dataset. The model is trained on the training dataset using a supervised learning method. In practice, the training dataset often consist of pairs of an input vector and the corresponding output vector which is commonly denoted as the target, either predicts the label.

PREPROCESSING:

Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn Most of the times, your dataset will contain features highly varying in magnitudes, units and range. We need to bring all features to the same level of magnitudes. This can be achieved by scaling. We have scaled these values to a range of 0 to 1 before feeding them to the neural network model. To do so, we have divided the values by 255.Normalization is applied for both the inputs and output datasets.

ARCHITECTURE:

The K-means clustering algorithm uses iterative refinement to produce a final result. The algorithm inputs are the number of clusters K and the data set. The data set is a collection of features for each

data point. The algorithms starts with initial estimates for the K centroids, which can either be randomly generated or randomly selected from the data set. The algorithm then iterates between two steps:

1. Data assignment step:

Each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance. More formally, if ci is the collection of centroids in set C, then each data point x is assigned to a cluster based on

$$\underset{c_i \in C}{\operatorname{argmin}}\ dist(c_i,\ x)^2$$

where dist(·) is the standard (L2) Euclidean distance. Let the set of data point assignments for each ith cluster centroid be Si.

2. Centroid update step:

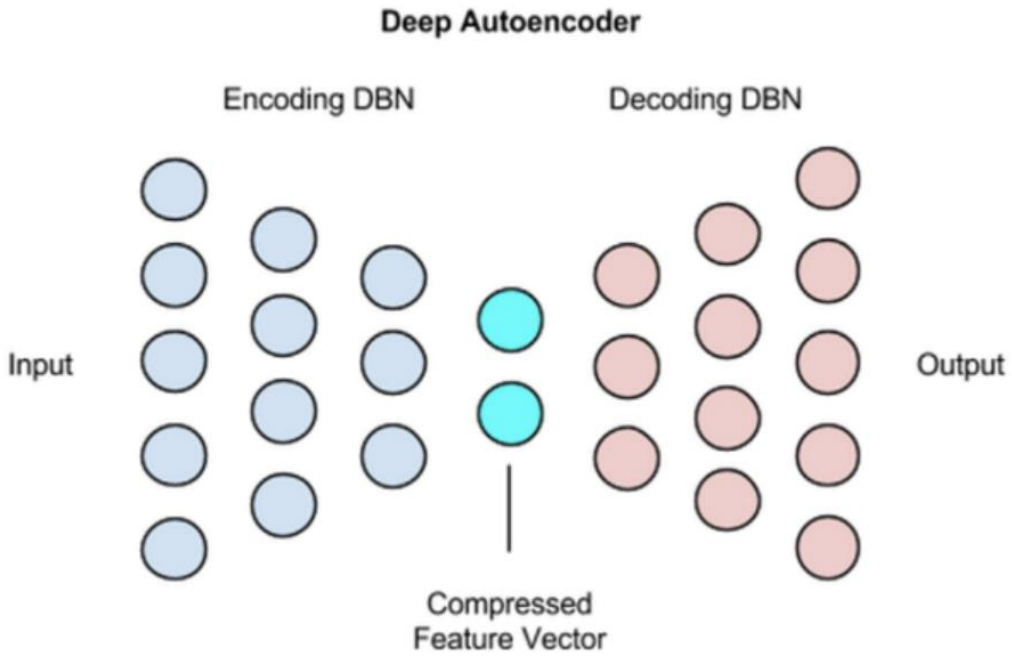
In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.

$$c_i = \frac{1}{|S_i|}\sum_{x_i \in S_i} x_i$$

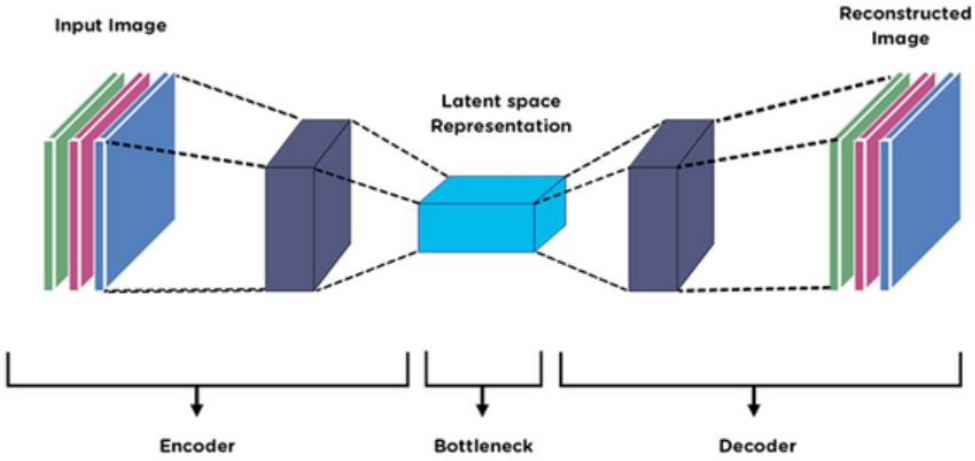
The algorithm iterates between steps one and two until a stopping criteria is met i.e., no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached.
The algorithm described above finds the clusters and data set labels for a particular pre-chosen K which is 10 for our dataset. To find the number of clusters in the data, we need to run the K-means clustering algorithm for a range of K values and compare the results. In general, there is no method for determining exact value of K, but an accurate estimate can be obtained from observations.

AUTO-ENCODER:

An auto-encoder is an artificial neural network that trains a model to give a reconstructed input as the output of the network. It is composed of two symmetrical Deep Belief Networks that has four to five shallow layers representing the encoding half of the net and second set of four or five layers that make up the decoding half. Auto-encoders can be stacked and trained in a progressive way, we train an auto-encoder and then we take the middle layer generated by the AE and use it as input for another AE and so on. This is the first step towards deep learning, the stacked auto-encoders will learn how to represent data, the first level will have a basic representation, and the second level will combine that representation to create a higher-level representation and so on. Think about images, a first level auto-encoder will learn to detect borders as features, the second level will combine those borders to learn traces and patterns etc. Below is how an auto-encoder looks like:



The network uses **dimensionality reduction** to restructure the inputs, so that the value are more scaled up and better for training a model



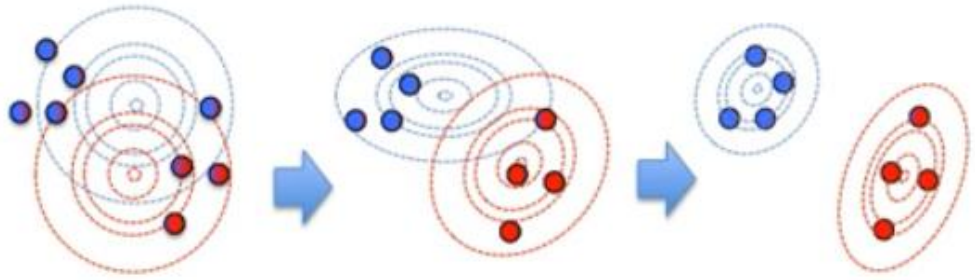
GUASSIAN MIXTURE MODEL:

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. The Gaussian Mixture object implements the expectation-maximization (EM) algorithm for fitting mixture-of-Gaussian model. A GaussianMixture.fit method is provided that learns a Gaussian Mixture Model from train data. Given test data, it can assign to each sample the Gaussian it mostly probably belong to using the GaussianMixture.predict method. Mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians.

For GMMs, we will find the clusters using a technique called “Expectation Maximization”. This is an iterative technique that feels a lot like the iterative approach used in k-means clustering.

In the “Expectation” step, we will calculate the probability that each data point belongs to each cluster (using our current estimated mean vectors and covariance matrices). This seems analogous to the cluster assignment step in k-means.

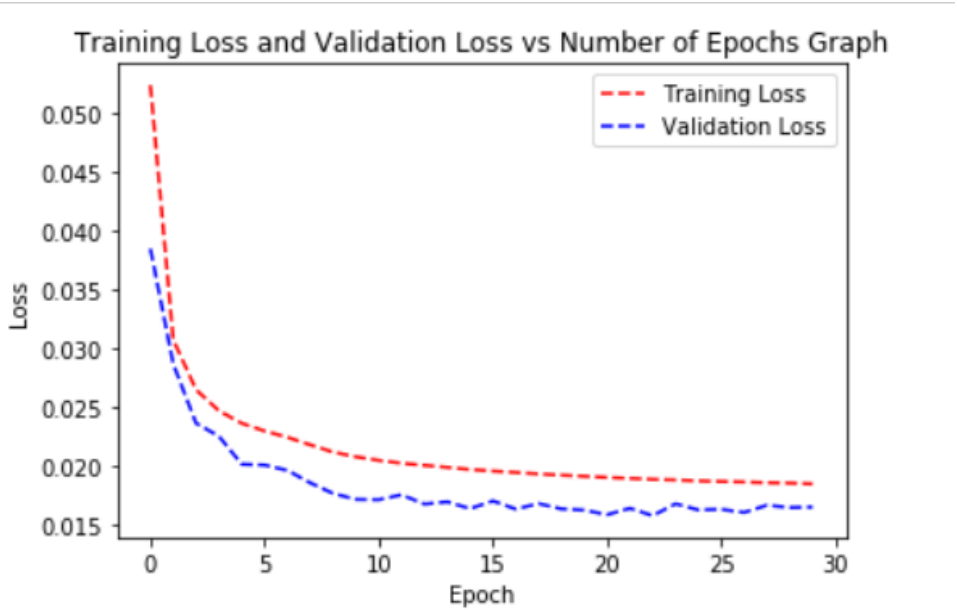
In the “Maximization” step, we’ll re-calculate the cluster means and covariance’s based on the probabilities calculated in the expectation step. This seems analogous to the cluster movement step in k-means.



RESULTS AND OBSERVATIONS:

1. **KMEANS Clustering Algorithm:**
   We have clustered the given dataset into 10 clusters. The clustering accuracy using KMEANS is: 51.16%

2. **Training using Auto-Encoder:**
   For auto-encoder network, we split the data into training and validation and the plots for the respective Loss vs Number of epoochs are:

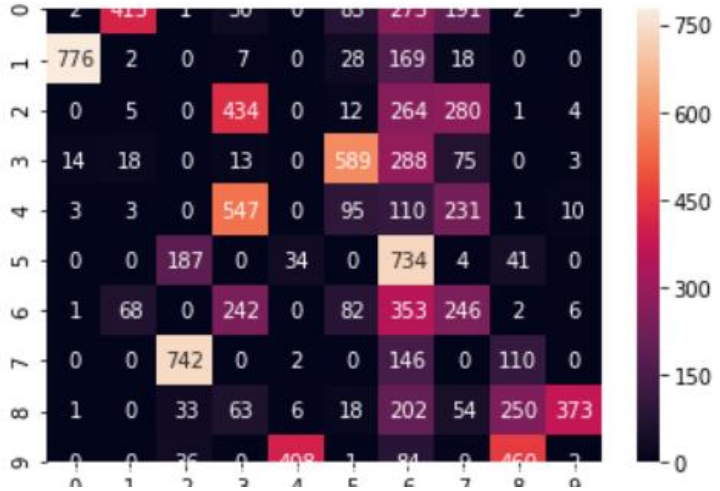


The minimum loss for Training is: 0.0185
The minimum loss for Validation is: 0.0166

3. **Auto-Encoder with K-Means Clustering:**

The accuracy reported was 56.24%

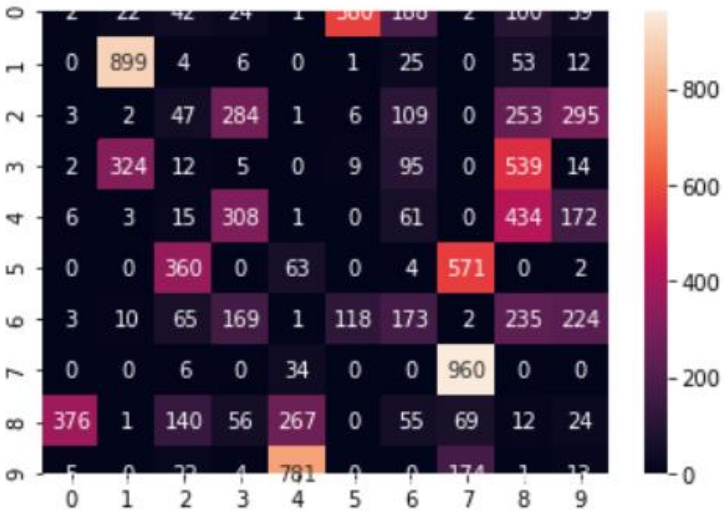
The confusion matrix for the predicted clusters is:



4. **Auto-Encoder with Gaussian Mixture Model:**

The accuracy reported for Gaussian Mixture Model : 54%

The confusion matrix for the predicted clusters is:



REFERENCES:

1. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
2. https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-meansclustering/https://www.tensorflow.org/tutorials/keras/classification
3. https://stackabuse.com/k-means-clustering-with-scikit-learn/https://skymind.ai/wiki/neural-network

4. https://www.datacamp.com/community/tutorials/k-means-clustering-python
5. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
6. http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/
7. https://brilliant.org/wiki/gaussian-mixture-model/
8. https://mccormickml.com/2014/08/04/gaussian-mixture-models-tutorial-and-matlab-code/
9. https://brilliant.org/wiki/gaussian-mixture-model/