CSE 574 – Cluster analysis on fashion MNIST dataset using unsupervised learning.

Rohit Lalchand Vishwakarma
Department of Computer Science University at Buffalo
– State University of New York
Buffalo, NY 14214 *rohitlal@buffalo.edu*

Abstract

We are performing cluster analysis of Fashion MNIST dataset using an unsupervised learning. Here we will be recognizing a clustering image of the Fashion-MNSIT dataset into one of many clusters. The data modelling is done in the python using the Google Colab. Here we are creating three different training models which are first using simple K-means clustering algorithm from sklearns library, second with Auto-Encoder based K-means clustering using both keras and Sklearns library and third same as second but using Auto-Encoder based Gaussian Mixture model.

1 Introduction

1.1 Unsupervised Learning

In machine learning, unsupervised learning is a type of learning that helps find previously unknown patterns in data set without pre-existing labels. The only requirement to be called an unsupervised learning method is to learn a new feature space that captures the characteristics of the original space while maximizing some objective function.

Two of the main methods used in unsupervised learning are principal component and cluster analysis. Cluster analysis is used in unsupervised learning to group, or segment, datasets with shared attributes in order to extrapolate algorithmic relationships. Cluster analysis is a branch of machine learning that groups the data that has not been labelled, classified or categorized. Instead of responding to feedback, cluster analysis identifies commonalities in the data and reacts based on the presence or absence of such commonalities in each new piece of data. This approach helps detect anomalous data points that do not fit into either group.

1.2 K-Means Clustering

It is an unsupervised learning algorithm that allows you to identify similar groups or clusters of data points within your data i.e. grouping the data into K clusters where assignment to the clusters is based on some similarity or distance measure to a centroid.

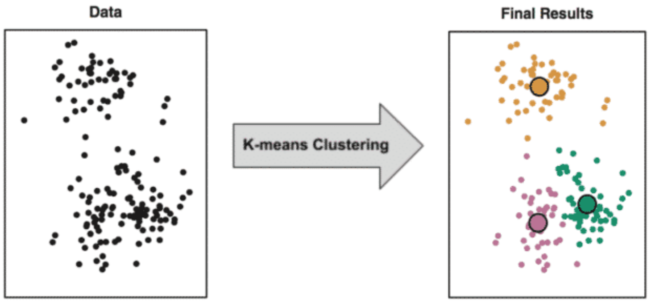


Fig 1.2 – K-Means Clustering

1.3 Auto-Encoder

An autoencoder is a type of artificial neural network used to learn efficient data coding in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal "noise". Along with the reduction side, a reconstructing side is learnt, where the autoencoder tries to generate from the reduced encoding a representation as close as possible to its original input.

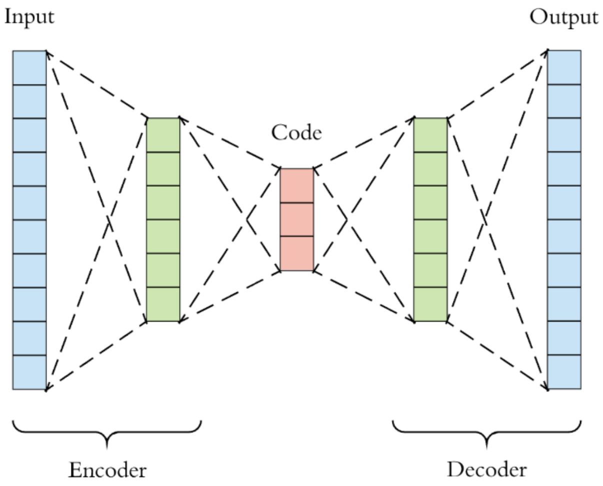


Fig 1.3 – Auto-Encoder

1.4 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. One can think of 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.

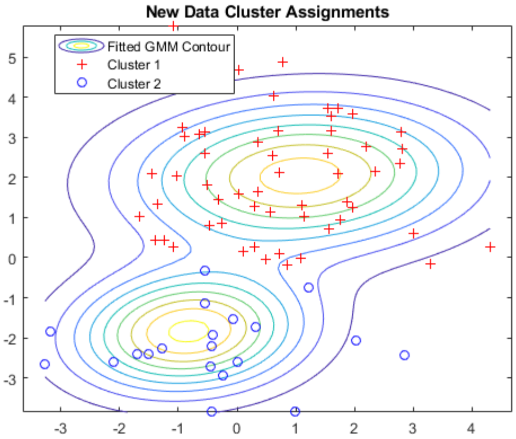


Fig 1.4 – Gaussian Mixture Model

2 Dataset

The Fashion-MNIST is a dataset of Zalando's article images, consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255. The training and test data sets have 785 columns. The first column consists of the class labels (see below), and represents the article of clothing. The rest of the columns contain the pixel-values of the associated image.

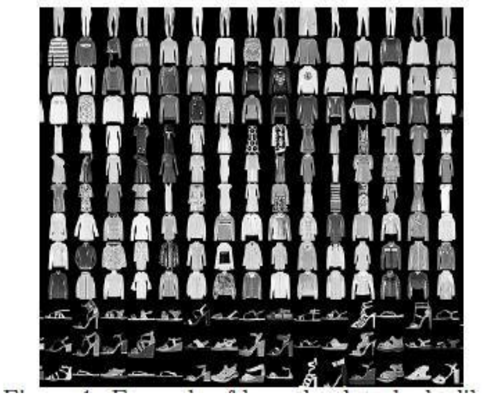
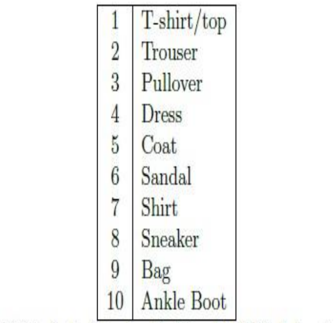


Fig 2.1 – Fashion Data 28X28 for 13 rows



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

Table 1: Labels for Fashion-MNIST dataset

3 Preprocessing

3.1 Importing Dataset

The Data that we imported is of matrix form 28X28 having 60000 entries of training. In order to to use in the model where we are performing simple K-Means algorithm, we need to first convert the matrix of the data into two dimensions from three dimensions. Hence, we multiplied that data 28X28 and generated 784 rows. So, our final data for training is 60000X784.

But for performing Auto-Encoder based K-means clustering and Auto-Encoder based Gaussian Mixture Model clustering will be using the original data format. This is because we are suing Convolutional Neural Network for Auto-Encoder which works fine with 3-dimensional data.

3.2 Splitting Train Set into Train and Validation Set

For Autoencoder with K-Means clustering and Gaussian Mixture Model we are creating Training set and Test set along with Validation set. As we need to validate the training data we will be using the train data of 60000X28X28 (Feature variable) to be split into 45000X28X28 (Feature variable) for Training data and for validation 15000X28X28. Our Test data will be of original dimension i.e. 10000X28X28.

4 Architecture

4.1 K-Means Model

From sklearn.cluster we are importing K-Means.to fit our training data where we are using 10 clusters. After that we are performing prediction on the testing data. We are using adjusted_mutual_info_score to get the accuracy by comparing the actual values and our predicted values.

4.2 Convolutional Neural Network

It is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. We are using this model in our Auto-Encoder process.

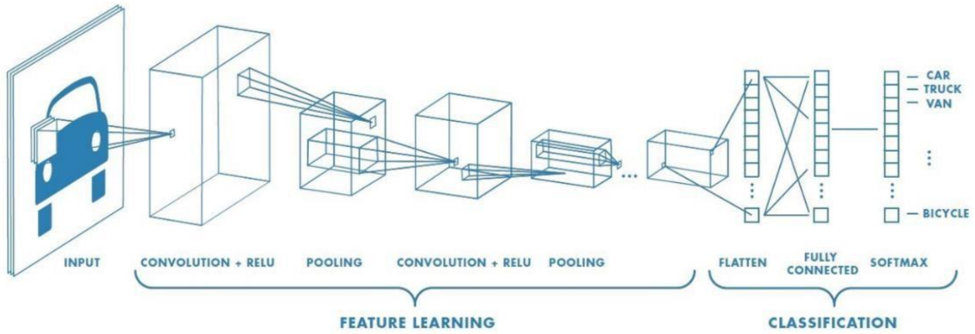


Fig 4.2-Convolutional Neural Network Model

4.3 Auto-Encoder with K-Means clustering

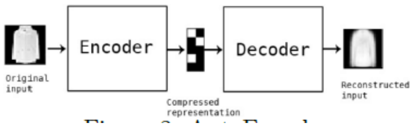


Fig 4.3 Auto-Encoder

Auto-Encoder is used for data encoding and decoding. It uses Convolutional Neural Network which has twelve layers, six for encoding and six for decoding. This network has number of nodes, activation function such as "relu" and MaxPooling2D to reduce the dimensionality of the data. Finally, we are using "adam" optimizer to optimize and Mean Square Error for getting the loss at each epochs. So, here we are using only the encoder part to get the reduced data dimension of training set i.e. encoded data which we are using in KMeans function to fit the model which is used to predict on Testing set.

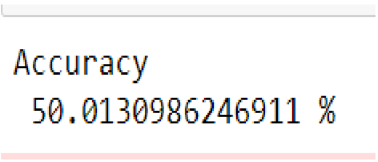
4.4 Auto-Encoder with Gaussian Mixture Model

We are using the same encoded data from the encoder part of the Auto-Encoder and applying it to Gaussian Mixture Model function to fit the model and finally using it to predict on Testing set. We imported the GaussianMixture from sklearn.mixture.

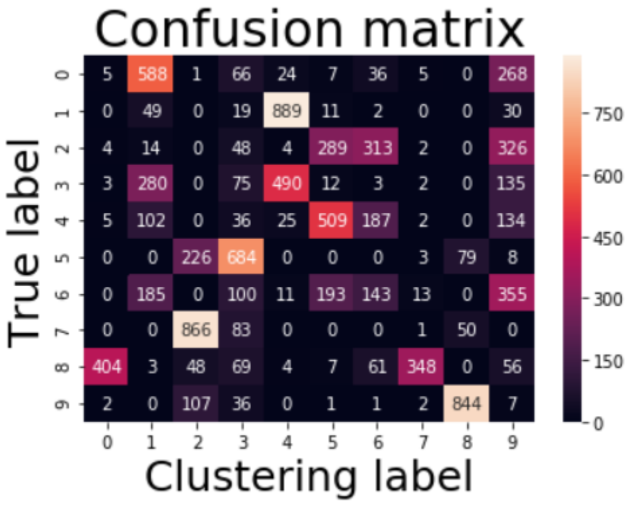
5 Results

We got the following results for our models

5.1 K-Means Clustering

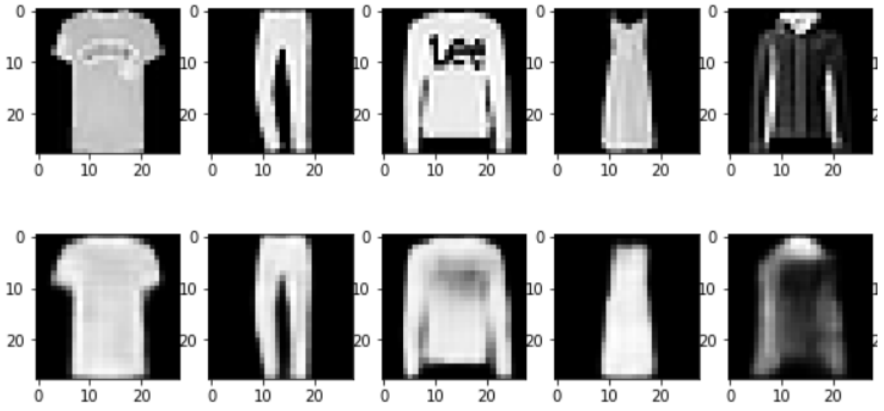


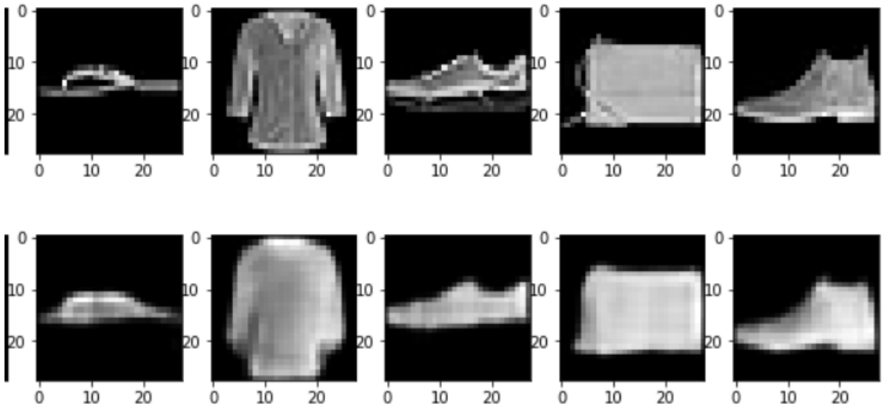
```
Accuracy
 50.0130986246911 %
```

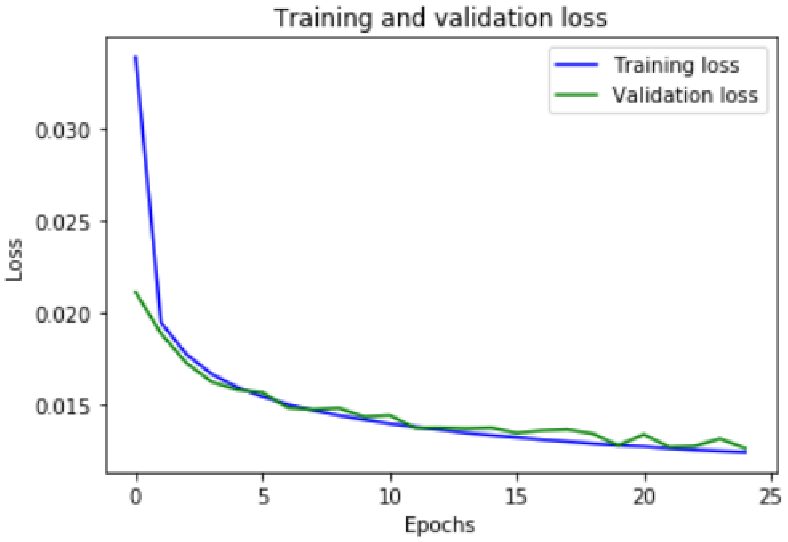


5.2 Training Auto-Encoder Network

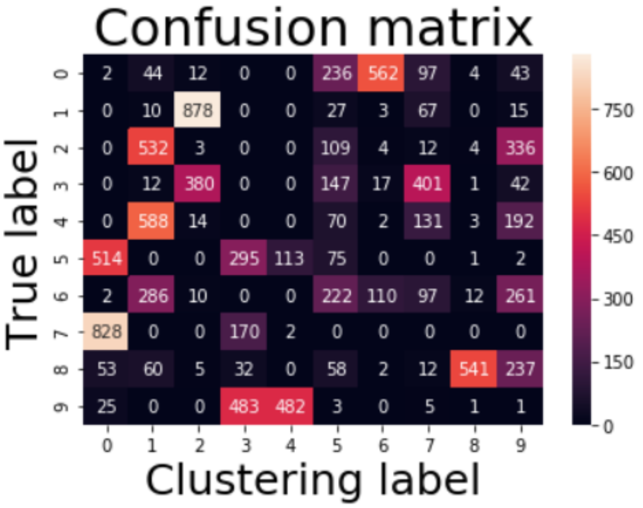
The 1st row reflects the actual images and the 2nd row reflects the image quality after applying auto-encoder.



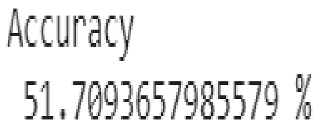




5.3 Auto-Encoder with K-Means clustering layer:

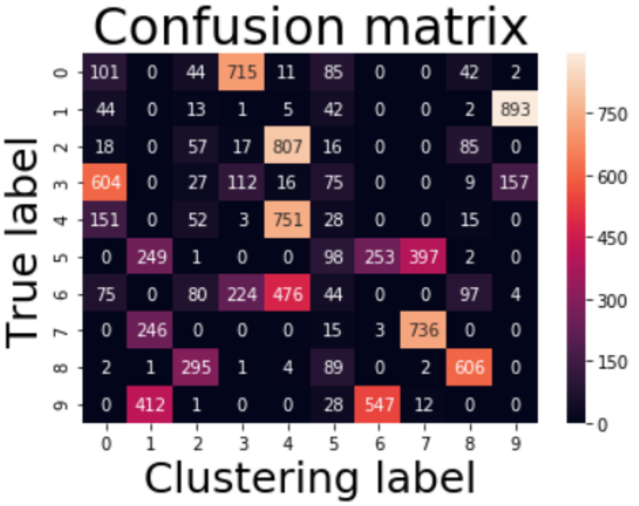


	0	1	2	3	4	5	6	7	8	9
0	2	44	12	0	0	236	562	97	4	43
1	0	10	878	0	0	27	3	67	0	15
2	0	532	3	0	0	109	4	12	4	336
3	0	12	380	0	0	147	17	401	1	42
4	0	588	14	0	0	70	2	131	3	192
5	514	0	0	295	113	75	0	0	1	2
6	2	286	10	0	0	222	110	97	12	261
7	828	0	0	170	2	0	0	0	0	0
8	53	60	5	32	0	58	2	12	541	237
9	25	0	0	483	482	3	0	5	1	1

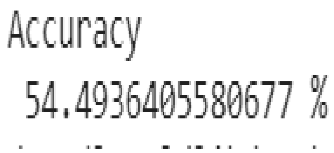


```
Accuracy
 51.7093657985579 %
```

5.4 Auto-Encoder with Gaussian Mixture Model layer:



	0	1	2	3	4	5	6	7	8	9
0	101	0	44	715	11	85	0	0	42	2
1	44	0	13	1	5	42	0	0	2	893
2	18	0	57	17	807	16	0	0	85	0
3	604	0	27	112	16	75	0	0	9	157
4	151	0	52	3	751	28	0	0	15	0
5	0	249	1	0	0	98	253	397	2	0
6	75	0	80	224	476	44	0	0	97	4
7	0	246	0	0	0	15	3	736	0	0
8	2	1	295	1	4	89	0	2	606	0
9	0	412	1	0	0	28	547	12	0	0



```
Accuracy
 54.4936405580677 %
```

6 Conclusion

We can conclude that Auto-Encoder model with Gaussian Mixture Model we are getting the maximum accuracy of 54.49% as compared to that of Auto-Encoder model with K-Means Model where accuracy is 51.71% and with simple K-Means algorithm accuracy is 50.01%.

We can tune hyperparameters where we change number of nodes in convolutional Neural Network, increase the number of epochs and change the learning rate to get the better accuracy.

7 References

1] iml_project3.pdf (Document for Project 3)

2] https://www.kaggle.com/s00100624/digit-image-clustering-via-autoencoder-kmeans

3] https://www.datacamp.com/community/tutorials/autoencoder-keras-tutorial

4] https://en.wikipedia.org/wiki/Unsupervised_learning

5] https://towardsdatascience.com/k-means-clustering-8e1e64c1561c

6] https://www.mathworks.com/help/stats/clustering-using-gaussian-mixture-models.html