Implementation and Analysis of Unsupervised Learning

Makarand Shyam Mandolkar
Department of Mechanical and Aerospace Engineering
University at Buffalo
Buffalo, NY 14214
makarand@buffalo.edu

Abstract

In this project we have made use of the Fashion-MNIST clothing classification problem dataset for our analysis, which is a comparatively a new and standard dataset being used in machine learning problems. In this project we have studied the implementation and analysis of the methodology of K-Means Clustering, Auto Encoder and Decoder, and Gaussian Mixture Model (GMM). Upon implementation, we were able to achieve an accuracy of 51.29% in K-Means, in K-Means with autoencoder the accuracy achieved is 56.54% and 59.29% for Gaussian Mixture Model using auto-encoder models

I

Introduction

The method or the type of machine learning algorithm which is used to draw results from a dataset, which consist of input data without labelled responses is particularly known as Unsupervised learning. The most commonly used method of unsupervised learning is cluster analysis, which is used for exploratory data analysis to find grouping in data or hidden patterns. As a result of which, it has made us choose one of the clustering type algorithm, in our case, K-Means Clustering. In this project we have made use of K-means clustering and K-means clustering using autoencoder and decoder model which is designed through TensorFlow and Keras. TensorFlow and Keras are one of the best machine learning frameworks and high-level APIs to train model and build them. We have also used Gaussian Mixture, that is a function comprised of several gaussians, each of which is identified by k ∈ {1, 2,3,..., K}, where K is the number of clusters of our dataset. To implement this the coding is done in python from scratch and used certain epoch values 100 epoch for training and validating the data. We have also made use of deep convolutional network layers, that are 3 in numbers. The kernel size used is 64, 32, 32. The learning rate used is 0.005. We are normalizing the data for the results. Here it is to be noted that Fashion-MNIST dataset is one of the two part of the original MNIST. It's found that many times from this dataset, linear classifiers also achieve high accuracy. The images of this dataset are in low quality at 28*28 pixel and dataset contains 60000 greyscale images which are classified into 10 categories.

Dataset Definition

For training and testing of our classifiers, we will use the Fashion-MNIST 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.

The class labels we are using are listed below.

- T-shirt/top
- Trouser
- Pullover
- Dress
- Coat
- Sandal
- Shirt
- Sneaker
- Bag
- Ankle boot
-

Pre-Processing

We are scaling (normalizing) the data, for which we perform the following function train-norm = train-norm / 255.0.

Architecture

Below is the description of K-Means, Gaussian Mixture models and Autoencoder.

K-Means

The K-means algorithm works by clustering data by trying to separate samples in n groups of equal variances, minimizing a criterion known as the inertia or within-cluster sum-of-squares. It scales well to large number of samples and has been used across a larger number of application areas in many different fields. The k-means algorithm divides a set of samples into disjoint clusters, each described by the mean of the samples in the cluster. The means which are called cluster centroids; note that they are not, in general, points from set of samples, although they live in the same space.

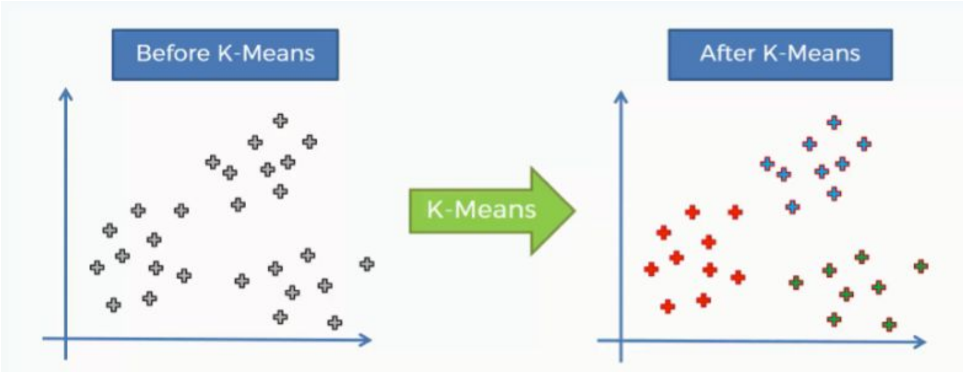


Figure 1: K-Means Clustering

Auto-Encoder with K-Means Clustering

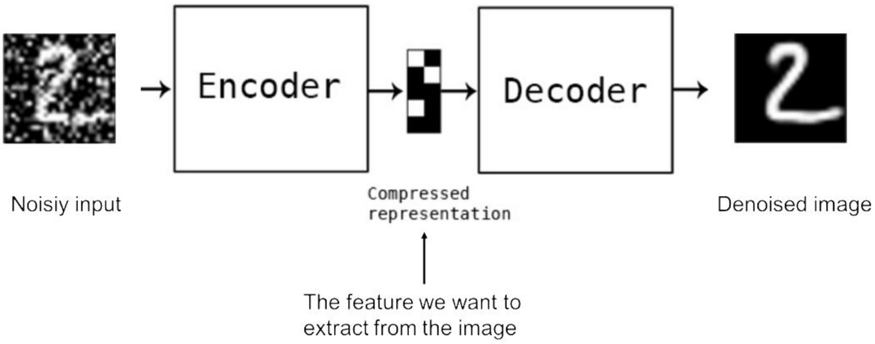
The K-means algorithm aims to choose centroids that minimize the inertia, or within-cluster sum-of- squares criterion:

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

Inertia can be recognized as a measure of how internally coherent clusters are. Inertia is not a normalized metric: we just know that lower values are better and zero is optimal. But in very high-dimensional spaces, Euclidean distances tend to become infated (this is an instance of the so-called curse of dimensionality). Running a dimensionality reduction algorithm such as Principal component analysis (PCA) or Auto-encoder prior to k-means clustering can alleviate this problem and speed up the computations.

Auto Encoder

Autoencoder uses compression and decompression functions which are implemented with neural networks as shown. Autoencoding is a data compression algorithm where the compression and decompression functions are data-specific, lossy, and learned automatically from examples rather than engineered by a human. For the building of autoencoder, three things are essential: 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 (i.e. a "loss" function). The encoder and decoder will be chosen to be parametric functions (typically neural networks), and to be differentiable with respect to the distance function, the parameters of the encoding/decoding functions can be optimized to minimize the reconstruction loss, using Adagrad optimizer.


Encoder
Decoder
Noisiy input
Compressed representation
Denoised image
The feature we want to extract from the image

Autoencoder

Auto-Encoder with GMM Clustering

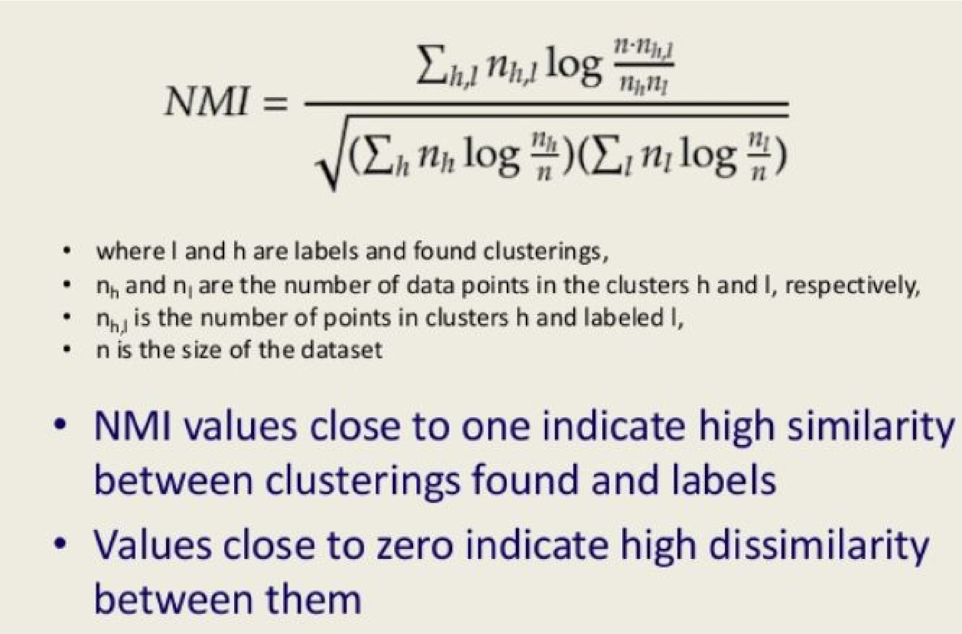
A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. The Gaussian Mixture object implements the expectation-maximization (EM) algorithm for fitting mixture-of-Gaussian models. It can also draw confidence ellipsoids for multivariate models, and compute the Bayesian Information Criterion to assess the number of clusters in the data. A Gaussian Mixture.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 belongs to using the Gaussian Mixture.predict method.

Results

The task which is performed is using unsupervised learning algorithms like K-means, Auto-Encoder based K-Means and Auto-Encoder based GMM clustering to perform cluster analysis on Fashion-MNIST dataset and report accuracy.

Evaluation metrics

Normalized Mutual Information (NMI) score is used to evaluate the accuracy of each model. NMC (Y, C) is given as:



$$NMI = \frac{\sum_{h,l} n_{h,l} \log \frac{n \cdot n_{h,l}}{n_h n_l}}{\sqrt{(\sum_h n_h \log \frac{n_h}{n})(\sum_l n_l \log \frac{n_l}{n})}}$$

- where l and h are labels and found clusterings,
- n_h and n_l are the number of data points in the clusters h and l, respectively,
- $n_{h,l}$ is the number of points in clusters h and labeled l,
- n is the size of the dataset

- NMI values close to one indicate high similarity between clusterings found and labels
- Values close to zero indicate high dissimilarity between them

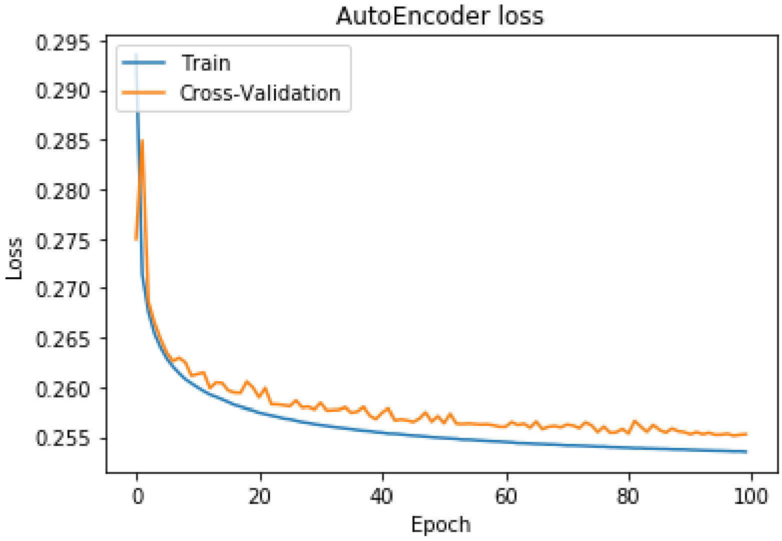
Normalized Mutual Information score formula

K-Means Clustering

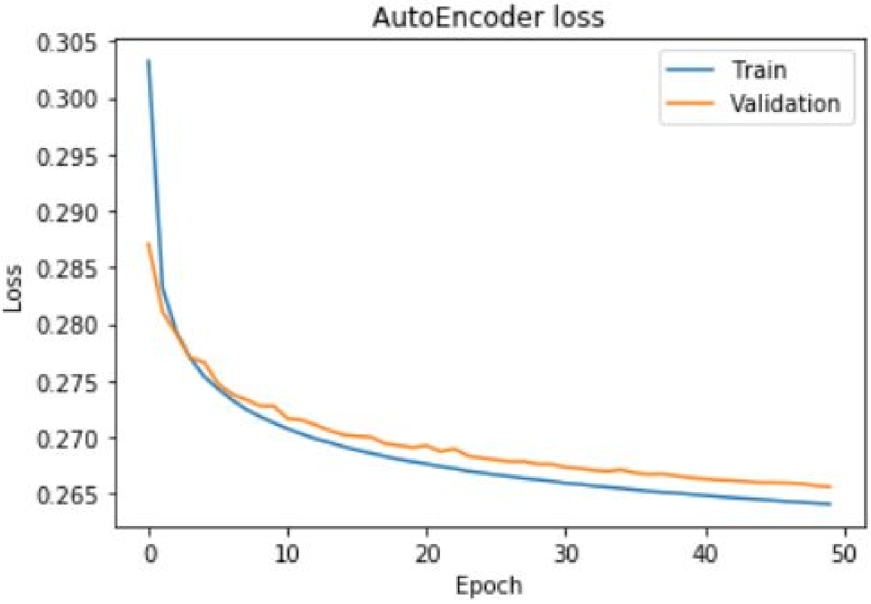
With number of clusters equal to 10 , using K-means Clustering, the highest accuracy I achieved without using the Auto-Encoder was 51.29%.

Auto-Encoder T

The Deep Convolutional architecture yielded the best loss vs the number of epochs, on trying several architecture for training the encoder. For loss function, 'binary cross entropy' is used and for optimizer Stochastic Gradient Descent (SGD) is used, but when using 'Adagrad' the loss was decreasing faster when compared with SGD. Figure below shows the graph of training loss and validation loss vs number of epochs while training for autoencoder.



Training and Validation loss vs number of epochs for SGD



Training and Validation loss vs number of epochs for Adagrad

Auto-Encoder based K-Means clustering. After reducing the dimensions from 784 to 512 using encoder function of the autoencoder, the best accuracy achieved was 56.54%.

Auto-Encoder based GMM clustering. After reducing the dimensions from 784 to 512 using encoder function of the autoencoder, the best accuracy achieved was 59.29%.

Conclusion

Thus, we were able to see how Machine Learning algorithms can be put into fashion industry. On trying out auto-encoders with various values and found out good results of training accuracy for algorithms of unsupervised learning using K-means is 51.29 percent. Now we implemented auto encoder with k-means we got 56.54%. And when we change the model from K-Means to Gaussian Mixture Model, we got accuracy of 59.29%. This kind of result can be obtained even if you increase your learning rate, hence by increasing your iterations you can increase your accuracy. For increasing the accuracy kernel size/weights also play a vital role, hence we should consider kernel size as a parameter for accuracy. Hence from this project we can come to the conclusion that unsupervised learning is not one of the most powerful tools for image classification.

References

[1] Srihari, Sargur & Chauhan, Mihir. iml_Project3.pdf

[2]Bilal,https://course.ccs.neu.edu/cs6140sp15/7_locality_cluster/Assignment-6/NMI.pdf