Unsupervised LearningWith K-Means and Gaussian Mixture Models

Lan H. Le
Department of Mechanical and Aerospace Engineering
University at Buffalo
Buffalo, NY 14226
hoanglan@buffalo.edu

Abstract

This project's purpose is to use different types of clustering algorithm as well as different ways to preprocess data in classifying clothing images. The two types of clustering algorithms being used are K-Means clustering and Gaussian Mixture Model clustering. The algorithms are first applied on the original data and then on encoded data created by an Auto-Encoder.

1 Introduction

Classifying images has been one of the essential topics of machine learning. Compared to humans, it is considerably more difficult for machines to be able to distinguish objects in images.

In the last project, we classified Fashion-MNIST clothing images into ten classes using various types of Neural Network.

The task of this project is grouping Fashion-MNIST clothing images into ten clusters. Instead of label images into specific classes, we merely identify items that are similar or belong to the same group.

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, and represents the article of clothing. The rest of the columns contain the pixel-values of the associated image.

The labels are only used during testing and not used during training.

1. T-shirt/top
2. Trouser
3. Pullover
4. Dress
5. Coat
6. Sandal
7. Shirt

8. Sneaker
9. Bag
10. Ankle Boot

3 Preprocessing

The dataset is loaded using provided fashion mnist reader notebook. Since the input values are integers between 0 and 255, we need to scale the inputs by dividing them by 255. This becomes helpful when we train the auto-encoder.

4 Architecture

1. K-Means

The KMeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares. This algorithm requires the number of clusters to be specified. For this project, the number of clusters is 10 which is equal to the number of classes of clothing in the data.

We use Sklearns library to implement the K-Means algorithm.

2. Auto Encoder

”Auto-encoding” 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. Auto-encoder uses compression and decompression functions which are implemented with neural networks.

The auto-encoder used for this project is a convolutional auto-encoder with the following layers: encoder layers, “flatten” layer, and decoder layers. This auto-encoder is implemented using Keras. We train the auto-encoder using the existing fashion dataset.

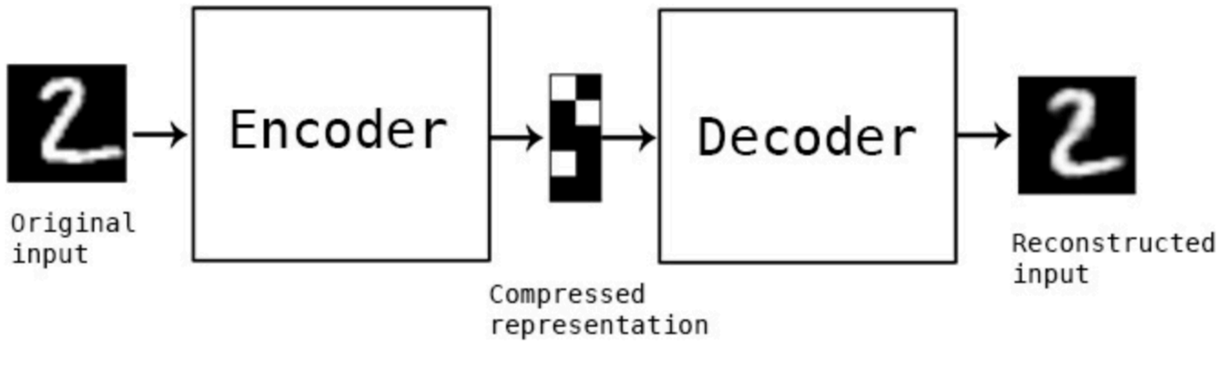


FIGURE 1: AUTO-ENCODER

3. Auto Encoder with K-Means Clustering

After training the auto-encoder, we extract the encoder portion of the auto-encoder. This encoder-only model has the same input as the auto-encoder but the output is the flattening layer. Using this encoder model, we are able to compress the original images into latent floating point values. We then apply K-Means clustering algorithm on these values. This is the same algorithm we use in part 1, only with different inputs.

4. Auto Encoder with GMM Clustering

For Gaussian Mixture Model clustering, we again utilize Sklearns to separate input data into clusters. The input data, similar to part 3, is the output of the encoder we train in part 2.

5 Results

K-means and GMM are unsupervised learning methods. When we fit our K-means and GMM models with training data, there is no direct connection between the data labels and our cluster predictions. Thus, we need to assign labels to our cluster predictions to match the data labels from the input. In this project, we try to do that by matching them such that the number of true-positive predictions is maximized.

After matching labels, we can proceed with finding out the accuracy of each method and constructing confusion matrix. We also use the Rand Index for reference.

1. K-Means

The Rand Index for training set is 0.3479. That for the test set is 0.3492. The best score possible is 1.0.

The accuracy for training set is 55.36%. For test set, the accuracy is 56%.

2. Auto Encoder

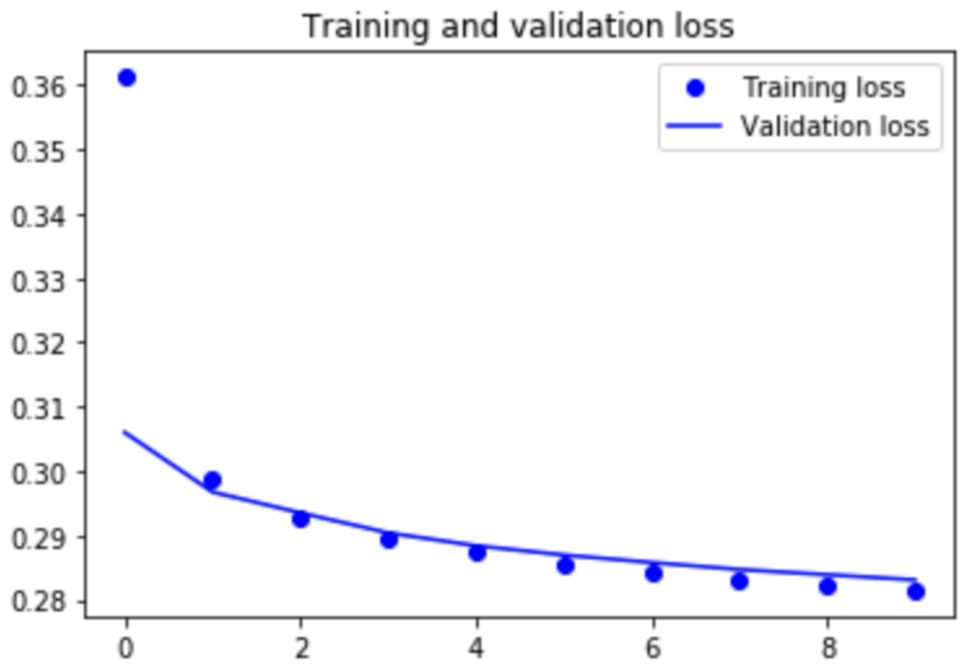
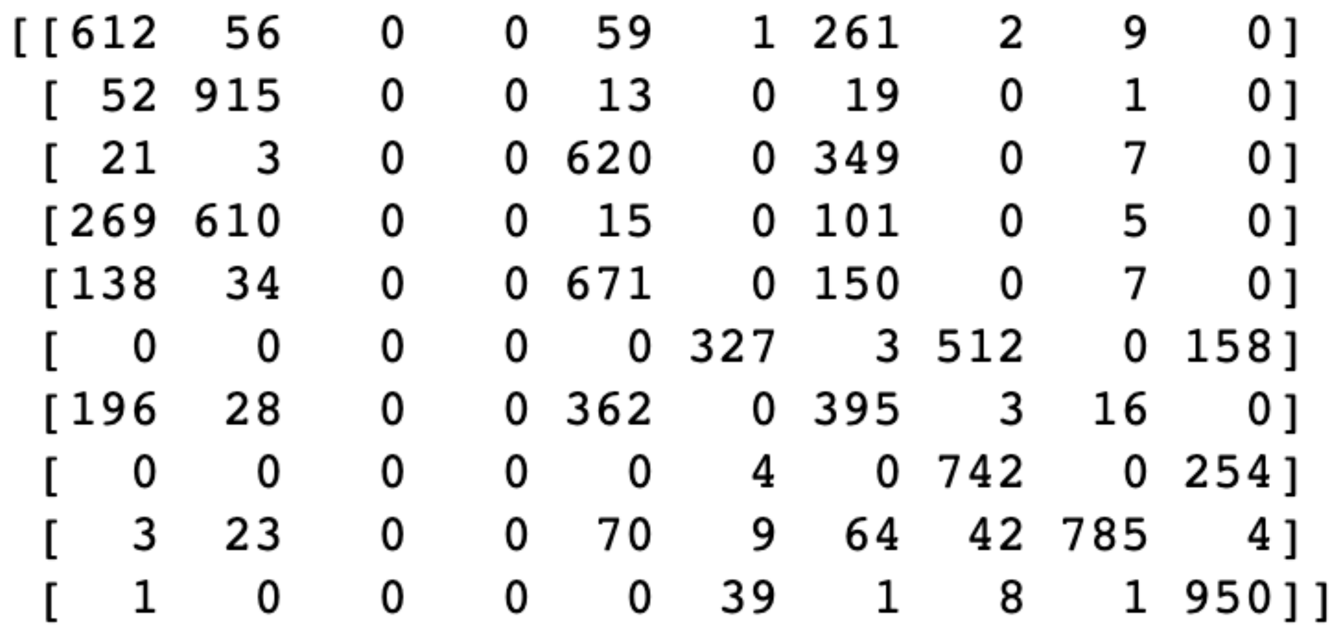


Figure 2: Graph of training loss and validation loss vs number of epochs

3. Auto Encoder with K-Means Clustering

The Rand Index for training set is 0.3817. That for the test set is 0.375. The best score possible is 1.0.

The accuracy for training set is 55.47%. For test set, the accuracy is 54.98%.



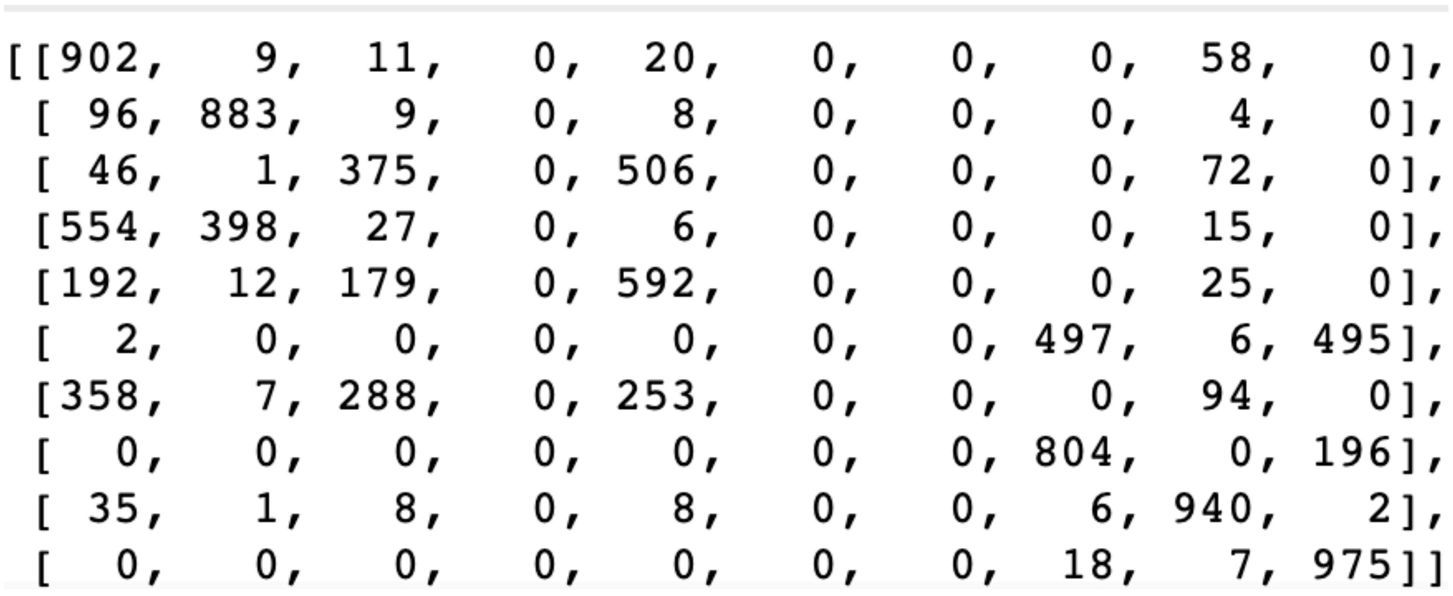
```
[[612  56   0   0  59   1 261   2   9   0]
 [ 52 915   0   0  13   0  19   0   1   0]
 [ 21   3   0   0 620   0 349   0   7   0]
 [269 610   0   0  15   0 101   0   5   0]
 [138  34   0   0 671   0 150   0   7   0]
 [  0   0   0   0   0 327   3 512   0 158]
 [196  28   0   0 362   0 395   3  16   0]
 [  0   0   0   0   0   4   0 742   0 254]
 [  3  23   0   0  70   9  64  42 785   4]
 [  1   0   0   0   0  39   1   8   1 950]]
```

Figure 3: Confusion matrix for K-Means models

4. Auto Encoder with GMM Clustering

The Rand Index for training set is 0.3751. That for the test set is 0.37. The best score possible is 1.0.

The accuracy for training set is 54.74%. For test set, the accuracy is 54.71%.



```
[[902,   9,  11,   0,  20,   0,   0,   0,  58,   0],
 [ 96, 883,   9,   0,   8,   0,   0,   0,   4,   0],
 [ 46,   1, 375,   0, 506,   0,   0,   0,  72,   0],
 [554, 398,  27,   0,   6,   0,   0,   0,  15,   0],
 [192,  12, 179,   0, 592,   0,   0,   0,  25,   0],
 [  2,   0,   0,   0,   0,   0,   0, 497,   6, 495],
 [358,   7, 288,   0, 253,   0,   0,   0,  94,   0],
 [  0,   0,   0,   0,   0,   0,   0, 804,   0, 196],
 [ 35,   1,   8,   0,   8,   0,   0,   6, 940,   2],
 [  0,   0,   0,   0,   0,   0,   0,  18,   7, 975]]
```

Figure 4: Confusion matrix for GMM models

6 Conclusion

With K-Means and GMM clustering algorithms, we are able to group our input data into 10 clusters which corresponds to the number of classes of the original data. However, when comparing these clusters with actual data we can see that the accuracy is not sufficiently high. Thus, it seems that for the task of classification neural networks perform better than clustering algorithms, especially for data with known labels. In the future, we would like to research more about how to improve the outcome of clustering algorithms to a level that is comparable to other classifying methods.

Acknowledgments

The descriptions of the dataset and algorithms are derived from the original project description.

References