Representation learning on Fashion-MNIST

Mukul Agarwal, Vishal A Raheja, Sonakshi Gupta and Vaibhav Tyagi
Project Mentor: Minqian Chen
Ilmenau University of Technology
P. O. Box 100565, D-98684 Ilmenau, Germany
Email: { Mukul.agarwal, Vishal-Amarbhai.raheja, Sonakshi.gupta, Vaibhav.tyagi }@tu-ilmenau.de

***Abstract* — Training an (variational) autoencoder and basic autorncoder on Fashion-MNIST dataset [6].Use the generated embeddings for clustering tasks and visualize the latent space and analyze the difference in the latent space of autoencoder and Variational Autoencoder.**

***Index Terms*—Fashion-MNIST, Autoencoder**

I. Introduction

Clustering of Fashion-MNIST dataset using unsupervised learning technique. Plenty of supervised learning techniques are available in the market, so our motivation is to rule out labels and use the information from the data itself to classify into respective classes. One of the relevant algorithm for unsupervised learning is the use of autoencoders,the basic architecture of autoencoders consists of three components Encoder, Latent space and Decoder. which uses the information from the latent space to classify the data. One of the variation of autoencoder is variational autoencoder which also considers the distribution of the data in the latent space for classification. The dataset used for the classification task consists of grayscale image of 28x28. We use Fully Connected Network(FCN) to reduce the input data to latent representation and inverse operation to reconstruct from latent representation.

II. Fashion-MNIST dataset

The Fashion-MNIST database is a comprehensive collection fashion items that is extensively used for training software models for image processing related to fashion industry. The database is also commonly used in the area of machine learning, that is, for training and testing.

Fashion-MNIST is a database, composed of a training dataset of 60,000 samples and a testing dataset of 10,000 samples for Zalando's article photographs as shown in Fig. 1 . Zalando is a multi-national apparel merchant company based in Germany, which was established in 2008. The dataset is made public by the company for download on zalandoresearch on GitHub URL [7]. Zalando Research is the department that developed the repository from inside the organization. It is a database composed of grayscale photographs of 10 categories of apparels objects. The list described in Table I gives the representation of all 0-9 integers to different classifiers.

It is a benchmark dataset used in image processing and deep learning. It is a shoe and clothing classification problems. Each image is a of 28x28 pixels, paired with a 10 class label. Fashion-MNIST is expected to serve as a straightforward substitute for measuring the performance of machine learning algorithms for the original Fashion-MNIST dataset.

Apparel	**Index**
T-shirt/top	= 0
Trouser	= 1
Pullover	= 2
Dress	= 3
Coat	= 4
Sandal	= 5
Shirt	= 6
Sneaker	= 7
Bag	= 8
Ankle boot	= 9

TABLE I: Fashion MNIST Categorization of Apparel

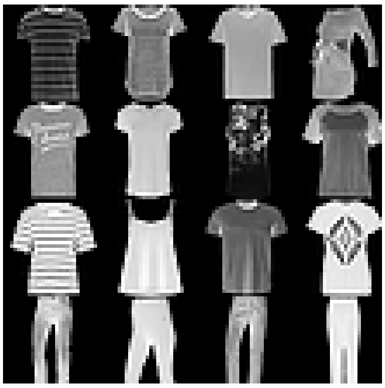


Fig. 1: Sample Image of Fashion-MNIST Dataset (Source: [6])

III. Autoencoder

Autoencoder is an unsupervised learning methodology for an artificial neural network. It learns how to compress and encrypt data effectively, and then learns how to recreate the data back to a representation that is as similar as possible to the original input from the reduced encoded representation. An autoencoder can learn the significant features of the data via this method. By nature, Autoencoder reduces the dimensions of data by studying how to ignore the irrelevant features in the data [2].

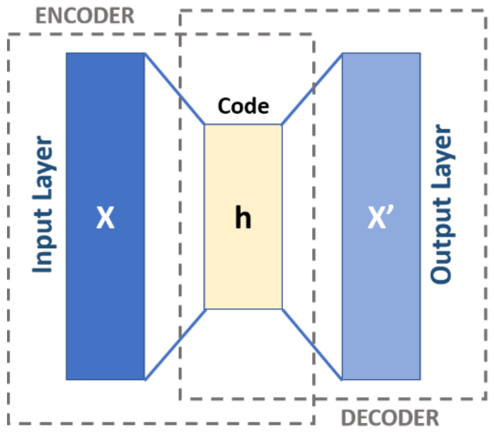


Fig. 2: Block diagram of Autoencoder (Source: [4])

A. Components of an Autoencoder

Autoencoders' block diagram and main components are shown in Fig 2. It's main components are as follow -

- Encoder: An encoder is a feed-forward, completely connected neural network that compresses the input into a latent representation and encodes the image into a reduced dimension as a compressed representation.
- Latent Space: This is the latent space created by the encoder network and used by the decoder network to reconstruct the original data.
- Decoder: Decoder is also an encoder-like feed-forward network that has an encoder-like framework. It is the duty of this network to recreate the input from the latent space back to the initial metrics.

B. Architecture of an Autoencoder

The encoder and decoder are both comprehensively connected neural feed-forward networks. The code, with the dimensional space of our preference, is a single layer of an Artificial Neural Network. The proportion of code layer nodes (code size) is the mathematical function that we set before the autoencoder is trained.

Next, to generate the code, the input passes via the encoder, which is a thoroughly connected Artificial Neural Network. The decoder, that has a close resemblance to the Artificial Neural Network, then only generates the output using the algorithm. The purpose is to generate an output that is equivalent to the input. Remember that the architectural design of the decoder is the encoder's true reflection or in other words its mirror image. This is not a prerequisite, although it is generally the case. The only criterion is that the dimensionality must be the same for the input and output. It is possible to program for something in the centre. Autoencoders are trained using backpropagation in the same manner as Artificial Neural Networks.

There are also some factors that would need to be adjusted first before autoencoder is trained:

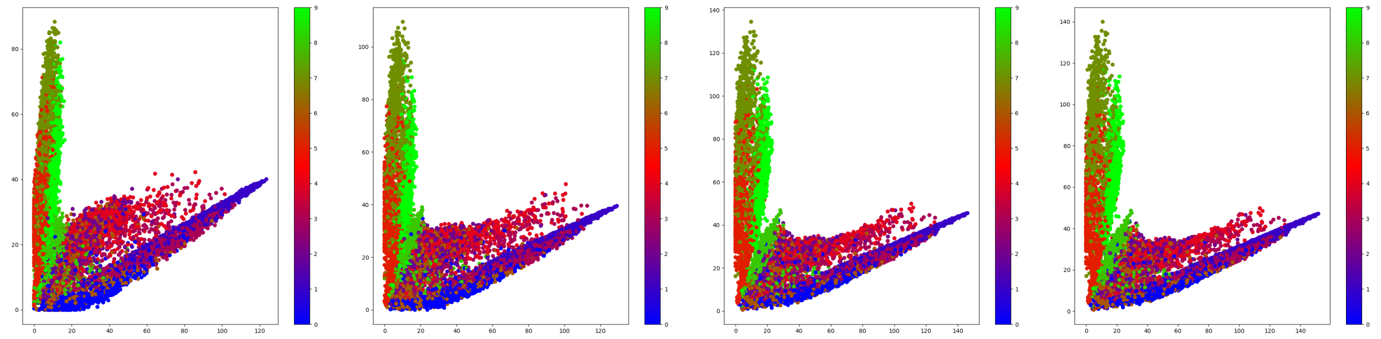
- Size of the code: It is the number of middle layer nodes. More compression results from smaller code sizes.
- Number of layers: Without considering input and output, the encoder can be just as extensive as much as we want.
- Number of nodes for each layer: the layers are stacked one by one. Autoencoders that are typically stacked look like the structure of a sandwich. The number of nodes for each layer is inversely proportional to the decoder again for every consequent layer of the encoder. The decoder is also symmetrical to the encoder, in terms of layer configuration and architecture. These criteria are not fixed and we can change them according to our requirements.
- Loss function: It's a way of determining how well the basic algorithm models the data provided. If estimates differ far more from actual outcomes, a very significant amount will stump up the loss function. With the assistance of any optimization function, the loss function progressively learns to decrease the prediction error.

IV. Variational autoencoder

Variational Autoencoder(VAE) is similar to the base architecture of autoencoder but with the slight modification in the representation of the latent space. The latent space of VAE is made such that it represents a probability distribution of specific properties. Here the used probability distribution is Gaussian function with zero mean and unit variance. The KL divergence loss keeps the track of the latent distribution in addition with reconstruction loss. Because of this the latent space is more continuous and complete in the sense that, similar images having the similar latent space are mapped close to each other. The encoder generates latent distributions for the various functions of the input images during training [3]. Block diagram of variational autoencoder is shown in figure 5.

Since the model knows the features or images rather than discrete values as Gaussian distributions, it is able to be used to produce new images. In order to create a vector that is fed into the decoding network, the Gaussian distribution is sampled to make an image dependent on the vector of samples. The model basically learns typical features of the training images and assigns them some chance that they will occur. It is then possible to use the probability distribution to reverse engineer an image, creating new images matching the original, training images.

The encoded information is assessed when learning the neural network and the identification framework produces two vectors, evaluating the mean and standard deviation of the images. Depending on these factors, a distribution system is developed. For the independent latent states, this is done. Then the decoder takes random samples from the appropriate distribution and uses them to recreate the original network inputs.



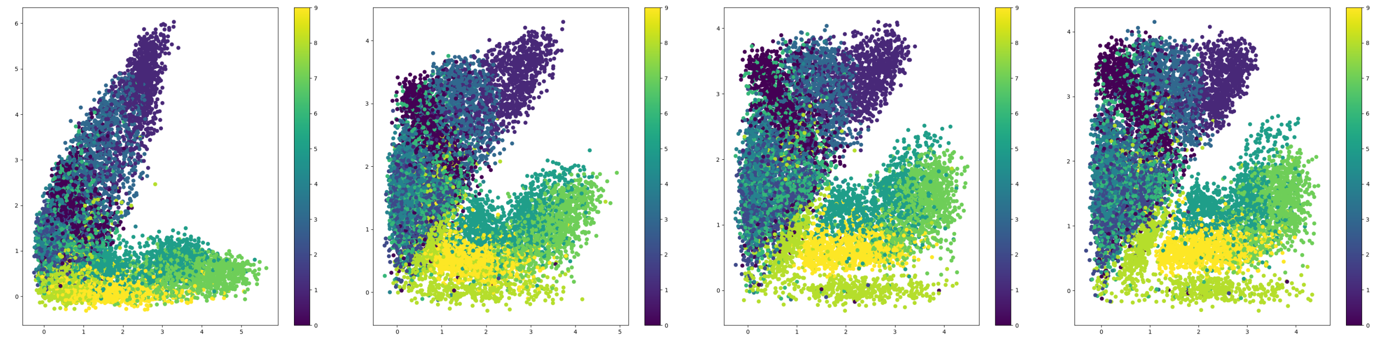
(a) AE Trained epoch 1

(b) AE Trained epoch 25

(c) AE Trained epoch 50

(d) AE Trained epoch 75

Fig. 3: Clustering Representation of Basic Autoencoder



(a) VAE Trained epoch 1

(b) VAE Trained epoch 25

(c) VAE Trained epoch 50

(d) VAE Trained epoch 75

Fig. 4: Clustering Representation of Variational Autoencoder

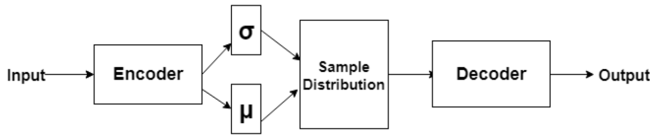


Fig. 5: Block diagram of variational autoencoder (Source: [1]

V. Clustering

Clustering is primarily an assortment of data on the basis of their resemblance and disparity. It is basically a type of method of unsupervised learning. An unsupervised method of learning is a method in which references are drawn from datasets consisting of input data without labelled answers. It is generally used to find meaningful structure, explanatory underlying processes, generative characteristics, and groupings inherent in a set of examples as a method. The function of clustering is to divide the proportion of data or data points into a number of clusters in such a manner that the data points of the same clusters are more similar to the data points of the same cluster and different from the data points of other clusters [5].

As it establishes the intrinsic classification among the unlabeled data available, clustering is very significant. For a successful clustering, there seem to be no specifications. What parameters they should use to meet their preferences varied depending on the individual. For example, we may be interested in finding homogeneous cluster for data redundancy, or looking for ”natural clusters” and defining their unexplained properties, identifying relevant and sufficient clusters or finding odd data objects in the dataset. This algorithm would draw some conclusions that reflect the similarities of points and make multiple and similarly true clusters of each assumption.

VI. Experimentation

We normalise the Fashion-MNIST dataset to maximum of 255 for pre-processing, and data augmentation will be done to make the dataset more diverse [6].

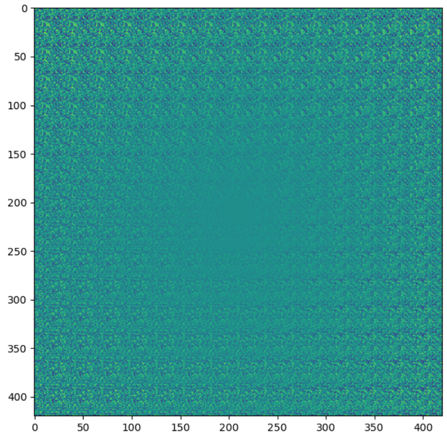


Fig. 6: Visualization of the latent manifold for Autoencoder

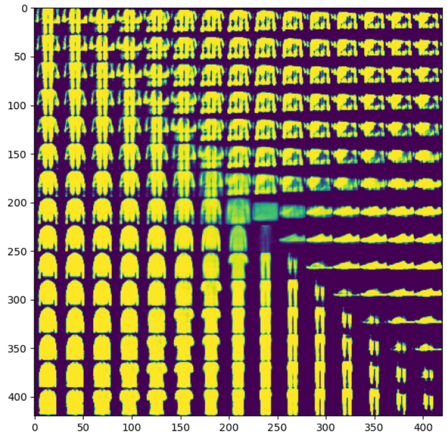


Fig. 7: Visualization of the latent manifold for Variational Autoencoder

We will Implement Basic Autoencoder and Variational Autoencoder technique and make a competitive analysis on the basis of the representation of the latent space and how well we can reconstruct the new images from the sampled latent space. Cluster classification representation of the latent space with respect to epochs. Keras python Framework is used for the implementation.

VII. Results

The figure 3 and 4 represents the clustering of latent space of Basic autonecoder and Variational Autoencoder respectively for all the 10 classes. As we can observe from the images how the latent space is distributed both of the cases.

VIII. Conclusion

The latent space in VAE is more continuous and complete as compared to the Autoencoder. Hence if we sample a point in the VAE latent space and pass it through the decoder we will get an image that has some meaning attached to it. As compared to the Autoencoder's latent space.

References

[1] Kang Atul and Kang Atul, "Variational autoencoders", Sep 2019.
[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, http://www.deeplearningbook.org.
[3] Diederik P Kingma and Max Welling, "An introduction to variational autoencoders", *arXiv preprint arXiv:1906.02691*, 2019.
[4] Michela Massi, "File:autoencoder schema.png", 2019.
[5] Anuja Nagpal, "Clustering - unsupervised learning", Nov 2017.
[6] Zalando Research, "Fashion mnist", Dec 2017.
[7] Zalandoresearch, "github.com/zalandoresearch/fashion-mnist".