Machine Learning-Algorithms for Optimization of Big Data Joint Project

Final Project Report, Generative Model using PPCA
Group - 03, Group Name - Pantomaths

Shreyas Patel[1401025], Malav Vora[1401062], Akhil Vavadia[1401095], Amee Bhuva[1401009], Manini Patel[164404]
School of Engineering and Applied Science, Ahmedabad University

***Abstract*—In this paper, we will look at how we can generate observed data values by training Generative Model with help of Principal Components of the training data. We have generated new images using Generative Model with help of Principal Components of given data set. The new generated images will look like the original one but it really doesn't exist in given data set. To help Generative Model in training, we have generated principal component using Probabilistic Principal Component Analysis with EM algorithm.**

***Index Terms*—Expectation Maximization, Probabilistic Principal Component Analysis, Generative Models, Generator, Discriminator, Convolution Neural Networks, Factor Analysis**

I. Introduction

Generative adversarial networks are a type of artificial intelligence algorithms used in unsupervised machine learning, implemented by a system of two neural networks, Generator and Discriminator. First, Generator is used to generate new images from given random input data and Discriminator is used to differentiate whether the generated image is natural or not. If the discriminator gives value closer to or equal to 1, then the generated image is said to be real else it is fake. Here, our proposed generative model follows semi-supervised form of learning.

II. Traditional Approach

In proposed generative model, basically generator tries to generate image from random input noise. Then this generated image will be passed to discriminator, which checks whether generated image is real or fake from it's past experience. Over a training, it will learn about generating images of that space. So basically during training, every time generator starts with random weights and it will converge to training image space. But if some how generator knows that important information about that space from starting then it will help generator to learn image space very fast, then generator only works with image weights. In our model, we have used principal components to provide generator that important information about image space directly.

III. Our Approach

Here we have used Probabilistic Principal Component Analysis with Expectation Maximization Algorithm to find principal components and data mean. Reason behind using this algorithm is explained in below sub-section.

A. Expectation Maximization with Probabilistic Principal Component Analysis

As we know, Principal Component Analysis is a technique through which we can find principal components of given data set which help us in dimension reduction and in other data analysis. But observed data vectors may have corrupted data entries and missing entries as well, so modeling with classical PCA may lead to bad analysis, so probabilistic modeling of observed data vectors is required which leads to PCA models called Probabilistic PCA with EM algorithm. From our previous analysis of this algorithm, we can say that it is comparatively faster then other algorithms like Robust PCA. Here we have generated principal components and data mean, and then provided into Generative Model.

So from given large data set, we have extracted Principal Components of given data set which contains important information of given set of images. This also helps us in reducing dimensions of data set which takes care of computation time. Then, we have used this Principal Components in our Generator Neural Network to help this model in producing new images of that space by using concept of factor analysis. Then, this generated image and original image will be passed through Discriminator which tells whether generated image is real or fake. According to that generator will be trained over iteration and at last generator learns about producing images which looks like real image. To extract wide range of weights related to features, generator and discriminator use convolution neural networks. Block diagram of this approach is as in fig.1.

B. Factor Analysis

A latent variable model seeks to relate a m*n-dimensional observation vector t to a corresponding q dimensional vector of latent variables x, where the linear relationship is linear as follow:

$$t/x = Wx + \mu + \epsilon \tag{1}$$

In our case, we have ignored error part(ϵ). W has size of [m*n]*[q], latent variable(x) has size of [q]*[1] and data mean has size of [m*n]*[1]. This manipulation will return us

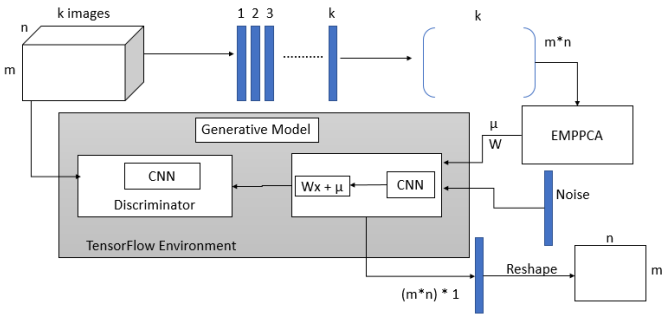

Fig. 1. Generative Model with Principal Components

[m*n]*[1] column vector which will be re-sized to [m]*[n] image. This generated image will be given to discriminator to check whether this image is real or fake. We have implemented this model on python with TensorFlow environment.

IV. Results

To test this model, we have used basically two data sets. 1) MNIST image data set 2) CIFAR-10 image data set. Firstly, we have applied MNIST data set which contains 55000 images of 0-9 digits of size 28*28. From this data set, we have taken different 6179 images of digit 1. After generating column vector matrix of 784*6179, we have extracted q=256 principal components using EM with PPCA algorithm. So, W of size 784*256 and data mean of 784*1 have been passed into generator neural network. After training for 1000 iteration with batch size of 16, proposed GM has generated image as in fig.2 and traditional GM has generated image as in fig.3. As we can see proposed GM generates slightly better image then traditional one over 1000 iteration.

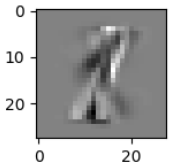

Fig. 2. Proposed Generative Model Output - MNIST Data - Digit 1

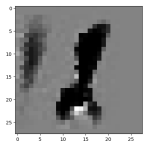

Fig. 3. Traditional Generative Model Output - MNIST Data - Digit 1

We have also compared this two model on basis of discriminator loss which is showed in fig.5 where as we can see from graph that learning rate of proposed GM is much higher then traditional GM. Next we have selected CIFAR-10 data set which contains very wide range images of size 32*32. We have selected 5000 images of car with very wide range of features. Same Model is applied to this set of images. Again we have 256 principal components for 1024*5000 column vector matrix. After training with 10000 iterations and batch size of 16, generative model has generated image as in fig.5.

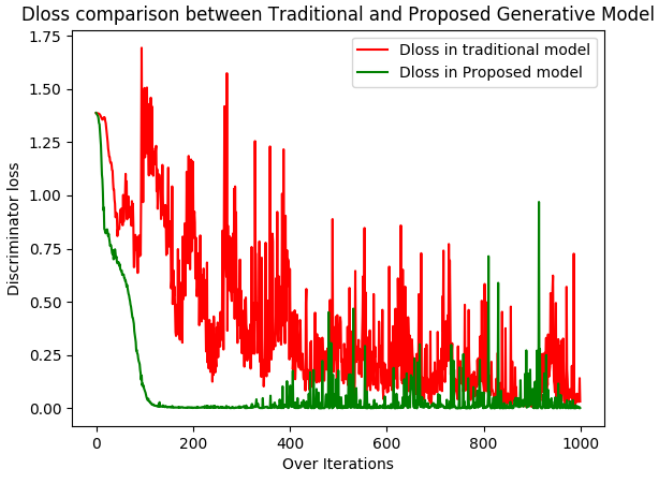

Fig. 4. Comparison between proposed GM and Traditional GM based on Dloss on MNIST Data set

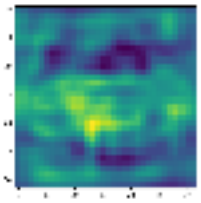

Fig. 5. Generated using Proposed GM - CIFAR-10 - Car

V. Conclusion

We have implemented our model and tested on MNIST data set and CIFAR-10 data set and we were successfully able to generate new images using proposed Generative Model. From above analysis, we can also say that proposed model gives better results in terms of time complexity also.

VI. Acknowledgement

We would like to express our deep gratitude to Prof. Mehul Raval and Prof. Ratnik Gandhi for giving us this opportunity to work and learn about this emerging area of Machine Learning and Big Data, and for their patient guidance and enthusiastic encouragement throughout this project. We would also like to thank our mentors Mr. Rahul Patel and Mr. Shashwat Sanghavi, who were very generous in sharing their time and knowledge with us.

References

[1] tensorflow.org/get started/mnist/beginners
[2] pythonprogramming.net/machine-learning-tutorial-python-introduction/
[3] Michael E.Tipping and Christopher M.Bishop Probabilistic PCA, Microsoft Research
[4] wikipedia.org/wiki/Generative adversarial networks
[5] github.com/adeshpande3/Generative-Adversarial-Networks
[6] github.com/bamos/dcgan-completion.tensorflow