Pattern Recognition and Neural Networks Assignment - 3 Report

Kaustuv Ray, Samrat Yadav, Pratyush Gauri
IISc Bangalore

;;

In this Assignment, we will try to implement valuable methodoligies like implementation of Deep Convolutional Generative Adversarial Network (DCGAN), Virtual Auto Encoder (VAE), Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE). Also we have compared K-NN and K-means Clustering algorithm on the same dataset and in the end an Ensemble Classifier (Adaboost) is used to classify our dataset. The dataset used here is PneumoniuMNIST Dataset.

I. DCGAN

A. Hyperparameters Tuning

Generator: We have used keras.sequential() from Tensorflow library to initialise the network and fine tuned it by the addition of four additional convolutional layers with stride value of 1 and 2 to create the generator model.

Discriminator: We have used keras.sequential() from Tensorflow library to initialise the network and fine tuned it by the addition of two additional convolutional layers and a flattening layer to create the generator model.

B. Observations

It is observed that there is no significance is obtained in the generated image till 20 epochs. After 30 epochs, the image can be distinguished but more than 30 iterations doesn't seem to provide any improvement to it. This can be observed in the images shown(Fig 1,2,3,4,5).

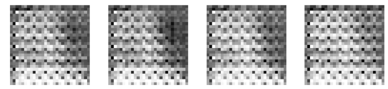

Fig. 1: 10 epochs

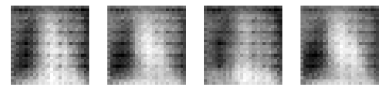

Fig. 2: 20 epochs

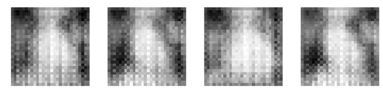

Fig. 3: 30 epochs

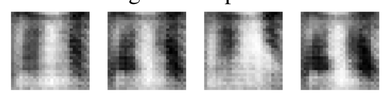

Fig. 4: 40 epochs

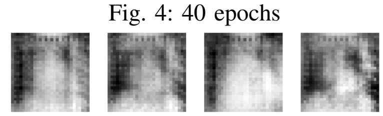

Fig. 5: 50 epochs

II. VAE

To generate data that strongly represents observations in a collection of data, we can use a variational autoencoder. Autoencoder is a type of model that is trained to replicate its input by transforming the input to a lower dimensional space(the encoding step) and reconstructing the input from the lower-dimensional representation (the decoding step).

A. Experiments

Encoder we have used Tensorflow library to input layer and three additional layer is added along with flattening layer to structure encoder. **Decoder** we have used Tensorflow library to input layer and three additional layer is added along with flattening layer to structure encoder.

B. Observation

It is observed that generated images (Fig.6,7) are not very prominent but the general outline of the chest is visible. Also a plot of latent space clustering is shown for different labels(Fig.8).

III. T-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data.

A. Experiments

We used t-SNE from sklearn on the PneumoniaTMNIST dataset. Then we generated 2d and 3d visualizations of the data(Fig,9,11).

B. Observations

The 2d and 3d representations shown in the image clearly shows that there are only a few overlapping of datapoints and the labels are well differentiated otherwise.

IV. PCA VS T-SNE

We have shown the comparison between PCA and t-SNE through 2d plots(Fig.9,10) and 3d plots(Fig.11,12).

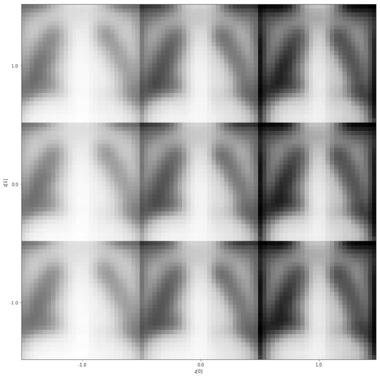

Fig. 6: VAE Generated images 3x3

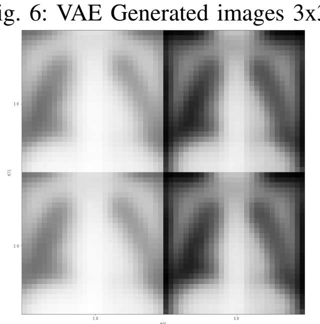

Fig. 7: VAE Generated images 2x2

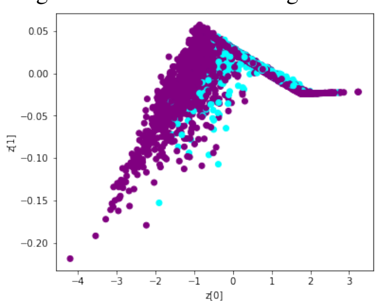

Fig. 8: VAE latent space clusters different classes

A. *Experiments*

We used PCA from sklearn and used it on PneumoniaTMNIST dataset. Then we generated 2d and 3d visualizations of the data(Fig,10,12).

B. *Observation*

We do observe that t-SNE provides better results as compared to the PCA. It provides well scattered datapoints clearly representing each of the datapoints but PCA has a bit clumsy approach.

V. Reconstructions from PCA

The image reconstruction is done of PneumoniaMNIST dataset and we do observe that there is significant change in first four components and afterwards the image is quite similar. This is expected as the initial components contain large amount of information as compared to later components. This is shown in Fig.13,14,15.

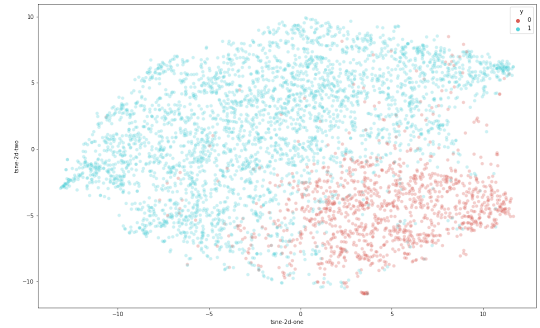

Fig. 9: 2d plot of t-SNE

Fig. 10: 2d plot of PCA

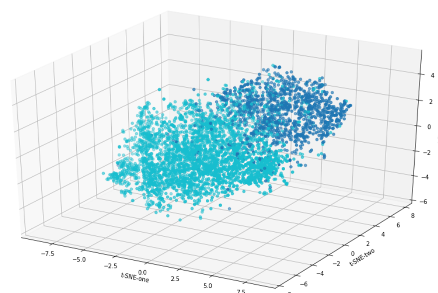

Fig. 11: 3d plot of t-SNE

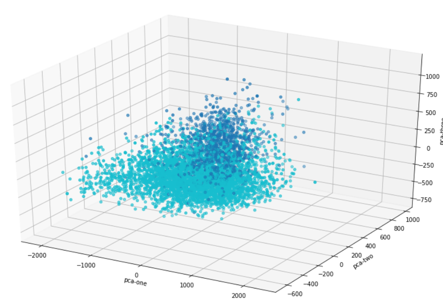

Fig. 12: 3d plot of PCA

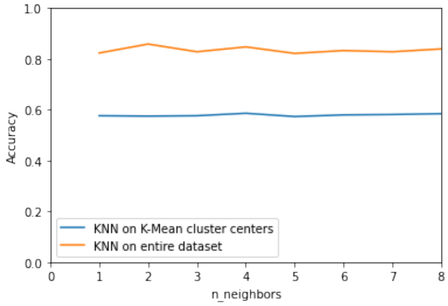

Fig. 16: KNN on entire dataset vs KNN on K mean centroid.

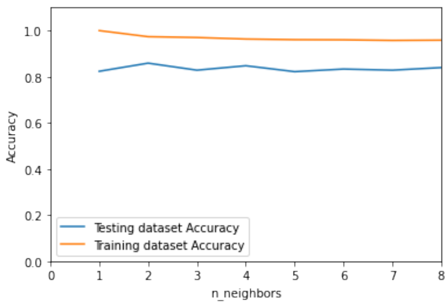

Fig. 17: KNN on entire dataset training vs testing

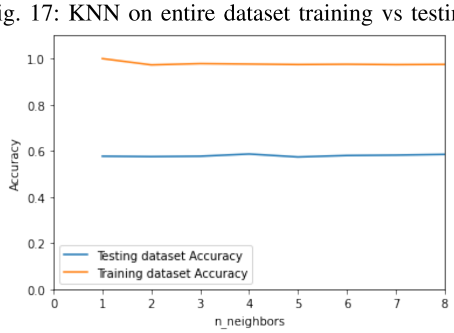

Fig. 18: KNN on Kmeans training vs testing

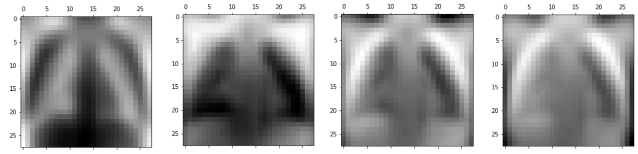

Fig. 13: PCA Reconstruction from first 1,2,3,4 components

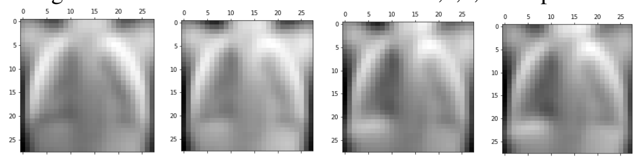

Fig. 14: PCA Reconstruction from first 5,6,7,8 components

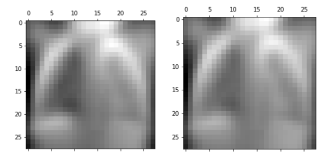

Fig. 15: PCA Reconstruction from first 9,10 components

VI. K-MEANS AND K-NN

K-means clustering is an unsupervised learning method nut KNN is a supervised learning algorithm. K-NN is a classification or regression machine learning algorithm while K-means is a clustering machine learning algorithm. We have implemented both on the PneumoniaMNIST dataset and we have also tried to implement KNN on the clusters centroid as neighbours

A. *Observations*

We do observe that the KNN algorithm provides us a 87 percent accuracy but the k-means could only achieve 57 percent. We have shown the different comparisons in the figure(Fig. 16,17,18). The accuracy iterations as number of neighbours are increased is tabulated in the table shown.