CS 795 – Deep Learning – Home Work 2

Ashwin Ravishankar
aravisha@gmu.edu

Files	Description
Ravishankar_HW2_Self.ipynb	Implementation of Dropout in model developed by me
Ravishankar_HW2_Keras.ipynb	Implementation of Dropout in model developed in Keras
Ravishankar_HW2_Q2_Part1.ipynb	AutoEncoders & Reconstruction Plot
Ravishankar_HW2_Q2_Part2.ipynb	Encoder & K-NN
Ravishankar_HW2_Q2_Part3.ipynb	PCA & K-NN
Ravishankar_HW2_GetMNISTData.ipynb	Module to download MNIST data to local

Question 1.)
Runtime Metrics:

HW 1 - Network: 2 Input, 3 hidden (**sigmoid**), 3 hidden (**sigmoid**), 2 output(softmax).

Sl. No	Metric	Model - Home Work 1	
		Keras Framework	My Implementation
1	Test Set	1250	1250
2	# Correct Predicted	1082	1097
3	Accuracy	87.36%	87.76%
4	Precision	0.853968254	0.876190476
5	Recall	0.890728477	0.880382775
6	F1-Score	0.871961102	0.878281623

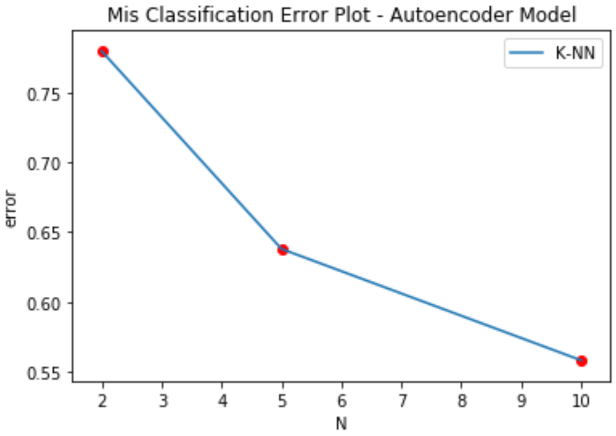
HW 2 - Network: 2 Input, 3 hidden (**relu**), 3 hidden (**relu**), 2 output(softmax).

Sl. No	Metric	Model - Home Work 2	
		Keras Framework	My Implementation
1	Test Set	1250	1250
2	# Correct Predicted	1082	1081
3	Accuracy	86.56%%	86.48%
4	Precision	0.819047619	0.822222222
5	Recall	0.905263158	0.900869565
6	F1-Score	0.86	0.859751037

Question 2.)

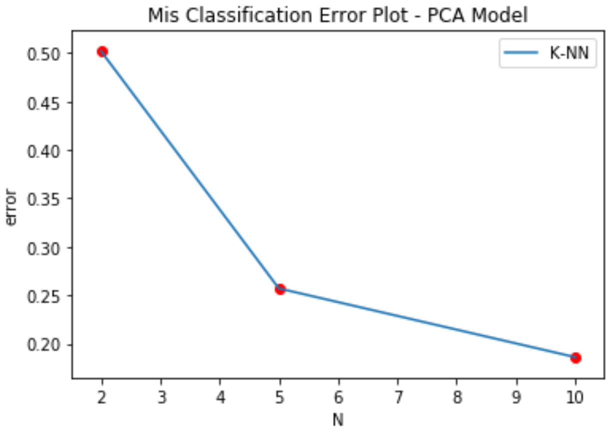
Comparison of plot of error w.r.t N for models:

- Encoder with K-NN classifier



-

- PCA with K-NN classifier

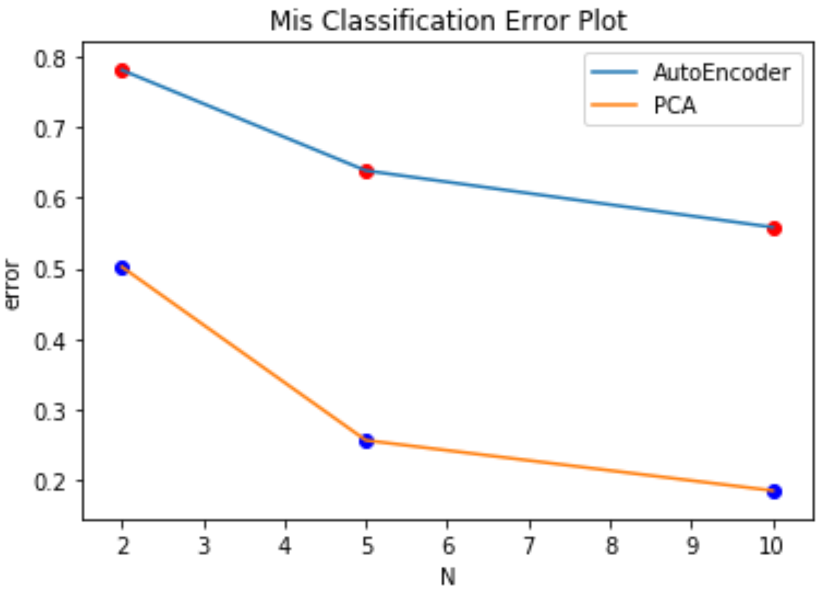


-

- As shown from the graph above, PCA performs significantly better.
- Below are the metrics.

Sl. No	**N**	**Accuracy**	
		AutoEncoder & K-NN	**PCA & K-NN**
1	2	22%	49.80%
2	5	36.20%	74.30%
3	10	44.20%	81.40%

- Putting the 2 plots together



-

Why am I asking you to compare to PCA? What is the connection here?

- Both – Encoder and PCA are essentially dimensionality reduction techniques.
- With pre-determined dimensionality of the latent space (N=2, 5, 10) we essentially convert the 28x28 image into low level features in the latent space.
- My guess is that 1000 images are not sufficient for the encoder to learn most accurate features, and hence performs very poorly compared to PCA.
- Also, the encoder model with linear activation function (I used relu – but experimented with sigmoid also) is a stochastic process, which, in a way looks identical to PCA.