ECE-6360-COMPUTER VISION

To: Dr. Sunanda Mitra

From: Sharanya Ramakrishna, Tsung-Sheng Huang, Supriya Ramaswamy

Date: 4th May 2018

Title: Assignment #5

ABSTRACT:

In the given assignment we are expected to learn the working of an autoencoder and explore the dimensionality reduction process of the autoencoders. We were then asked to compare the results obtained by comparison with one of the standard methods used in dimensionality reduction i.e principal component analysis. The results obtained by us is presented below.

Autoencoders are basically stacks of RBM's(Restricted Boltzmann machines) which contains an encoder to transform high dimensional data to low dimensional data and a decoder to recover that data from the 'code'.

WORKING OF AUTOENCODERS:

1. An ensemble of binary vectors can be modeled using a two-layer network called a "restricted Boltzmann machine" in which stochastic, binary pixels are connected to stochastic, binary feature detectors using symmetrically weighted connections. [1]

2. The pixels correspond to "visible" units of the RBM because they are the observed data; while the feature detectors correspond to "hidden "units. A joint configuration (v,h) of the visible and hidden units has energy given by[1]

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i \in \text{pixels}} b_i v_i - \sum_{j \in \text{features}} b_j h_j - \sum_{i,j} v_i h_j w_{ij}$$

v_i & hj are binary states of pixel 'i' and feature 'j' , bi and bj are biases, wij is weight between them.

v	h	$-E$	e^{-E}	$p(\mathbf{v}, \mathbf{h})$
1 1	1 1	2	7.39	.186
1 1	1 0	2	7.39	.186
1 1	0 1	1	2.72	.069
1 1	0 0	0	1	.025
1 0	1 1	1	2.72	.069
1 0	1 0	2	7.39	.186
1 0	0 1	0	1	.025
1 0	0 0	0	1	.025
0 1	1 1	0	1	.025
0 1	1 0	0	1	.025
0 1	0 1	1	2.72	.069
0 1	0 0	0	1	.025
0 0	1 1	-1	0.37	.009
0 0	1 0	0	1	.025
0 0	0 1	0	1	.025
0 0	0 0	0	1	.025
			39.70	

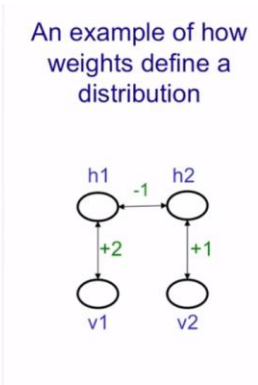


Figure 1: Example for calculating energy for a joint distribution

3. The probability is given using the sigmoid function,

$$\sigma (b_j + \Sigma_i v_i w_{ij})$$

where σ(x) is the logistic function 1/[1+exp(-x)][1]

4. Once the binary states have been chosen for the hidden units, the reconstruction in binary format is obtained.

5. The change in weight (weight update) is given using the formula given below

$$\Delta w_{ij} = \varepsilon\left(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{recon}}\right)$$

ε : learning rate

<vi,hj> $_{data}$: fraction of times that pixel 'i' and 'j' are ON together when pixel 'i' and feature detector 'j' are ON at the same time, when the feature detectors are driven by data. [1]

<vi,hj> $_{recon}$: fraction of times that pixel 'i' and 'j' are ON together when pixel 'i' and feature detector 'j' are ON at the same time, when the feature detectors are driven by the reconstruction. [1]

6. After learning one layer of feature detectors we can treat their activities- when they aren't being driven by data- as data for learning of second layer of features. Hence the first layer of features then become the visible layer for the second RBM. [1]

ALGORITHM :

1. In minstdeep.m design the autoencoder i.e set the number of units in hidden layer(784 in our case), number of units in penetration layers (inner hidden layers, in our case numpen1 = 500,numpen2 = 250,) final dimensionality reduced layer (30 in our case).

2. Number of training samples = 60,000

3. Before taking batchwise data , we need to randomize the data.

4. Number of Epochs used : 102, (computer got frozen after sometime using 200 epochs)

5. The results obtained is compared with PCA.

6. The error obtained is as follows

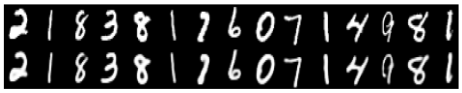
   Autoencoder train error: 3.9017

   Autoencoder test error: 4.146

   PCA error: 8.3289

OBSERVATIONS :

A> TRAINING RESULTS : REDUCED TO 30 UNITS (ERROR : 4.146%)



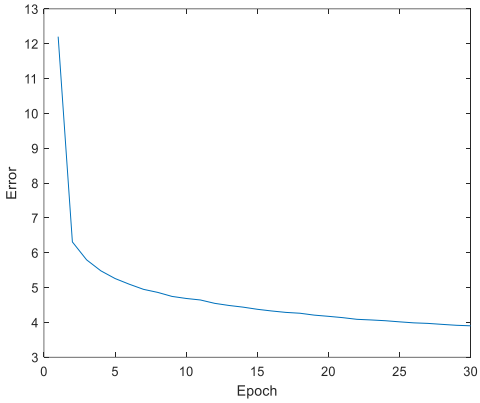


Figure 2 : Results and error for 30 unit reduced layer

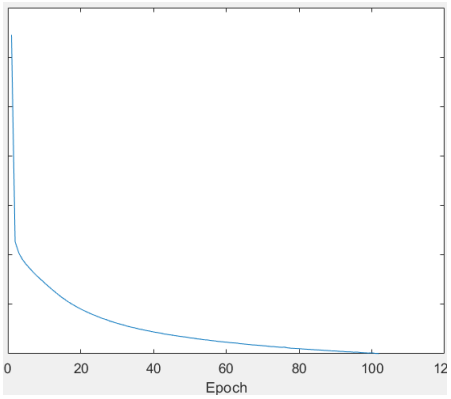
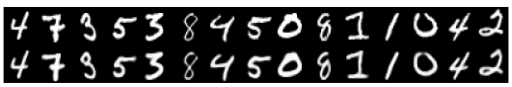


Figure 3 : Results and error for 2 unit reduced layer

B> TESTING RESULTS : REDUCED TO 30 UNITS (ERROR : 3.9017%)



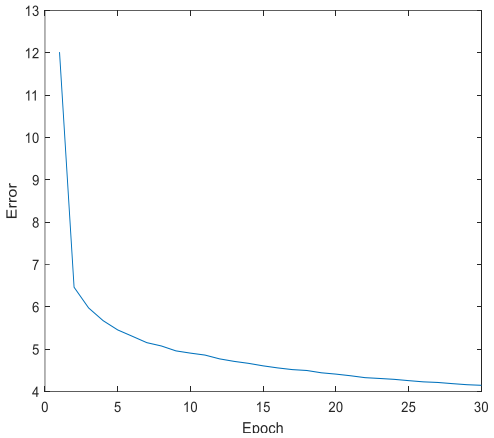


Figure 4 : Results and error for 30 unit reduced layer

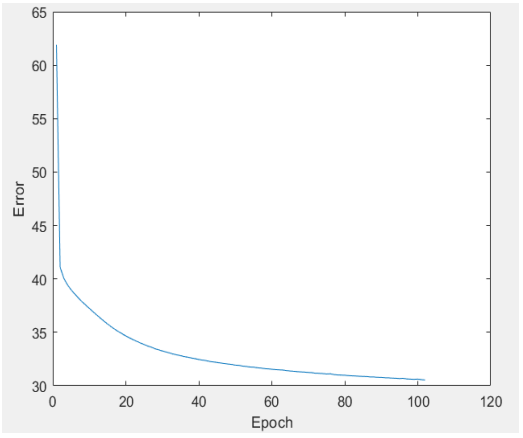


Figure 5: Results and error for 2 unit reduced layer

C> PCA ERROR : (ERROR : 8.3298 %)

Figure 6 : PCA classification result

D> 2D CODES FOR VISUALIZING SEPARATED DATA :

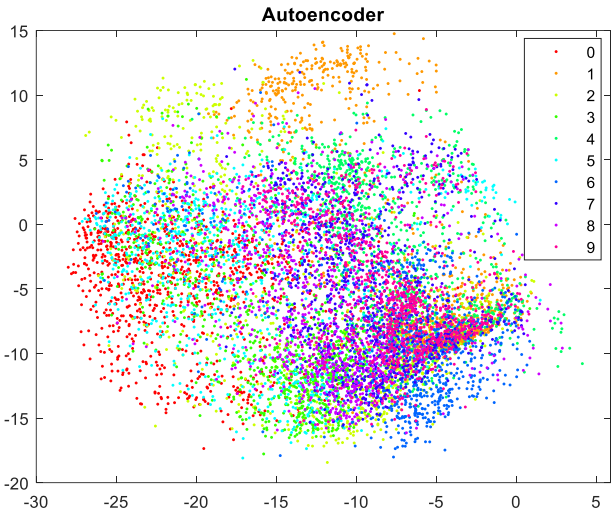


Figure 7: 2D code found by 784-1000-500-250-2 autoencoder

As can be seen from the figure, the data is not very well separated. This resaon is we used only 120 epochs instead of 200 epochs (computer was getting hanged).

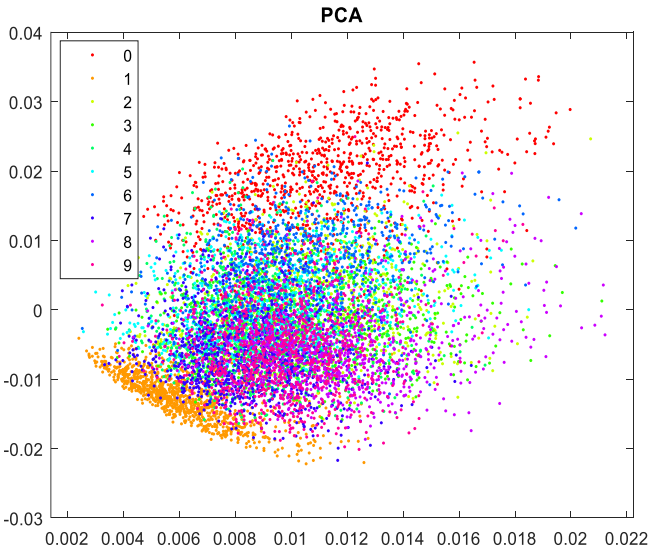


Figure 8: 2D code for each class of digits obtained by taking 1st 2 principle components of 60,000 images

CONCLUSION :

1. The error rates for training and testing of data using autoencoders was much lesser than PCA error.
    - Training data error: 3.9017 %
    - Testing data error: 4.146 %
    - PCA error: 8.3298 %
2. Hence autoencoder gives better classification of data compared to PCA and successful dimensionality reduction while classifying the data.

REFERENCES :

1. Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks."
2. The MNIST dataset found at : http://yann.lecun.com/exdb/mnist/index.html
3. https://en.wikipedia.org/wiki/Restricted_Boltzmann_machine