Dimensionality Reduction Using Neural Networks

Ayushman Singh Sisodiya(12188)

Guided By – Amitabh Mukherjee

Problem and Motivation

Dimensionality reduction facilitates the classification, visualization, communication, and storage of high-dimensional data. A simple and widely used method is principal components analysis (PCA), which finds the directions of greatest variance in the data set and represents each data point by its coordinates along each of these directions. I will do a nonlinear generalization of PCA that uses an adaptive, multilayer "encoder" network to transform the high-dimensional data into a low-dimensional code and a similar "decoder" network to recover the data from the code

Datasets

Mnist

2D-Robot-Arm

Challenges

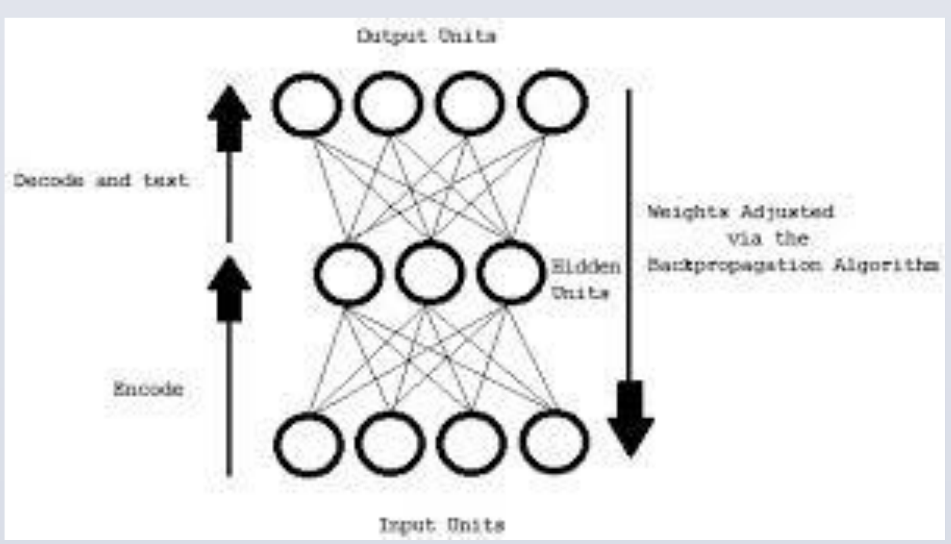
As I have used variant of gradient descent to train one of the main challenge is to overcome the popularly known vanishing gradient descent problem which is prominent in deep neural nets. Once the errors get back propagated to the first few layers, they are minuscule, and quite ineffectual. This causes the network to almost always learn to reconstruct the average of all the training data.

There is always the problem of under and over fitting in these type of learning algorithms

To overcome these challenges a method of pre-training is introduced by Hinton which initializes the weight of the network close to a good solution so that our network does not get stuck in a bad local minima

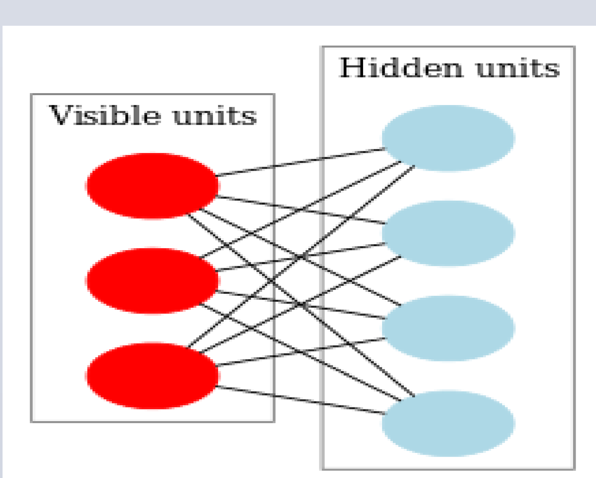
Methodology

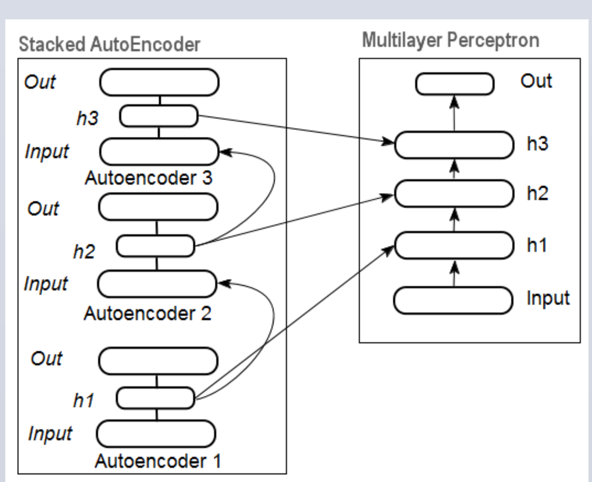
How?



1 - Pre-training

This basically involves layer wise training the network like a RBM or a shallow stacked auto encoder





$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{i,j} h_j \qquad P(v,h) = \frac{1}{Z} e^{-E(v,h)}$$

2 - Initializing weights and tied weights constraints

3 - Fine Tuning

Now the network is trained by standard feed forward and back-propogation

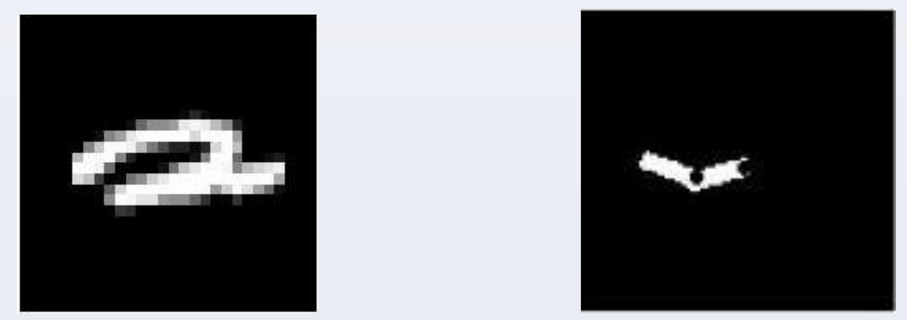
Results

I have used two types of networks architectures

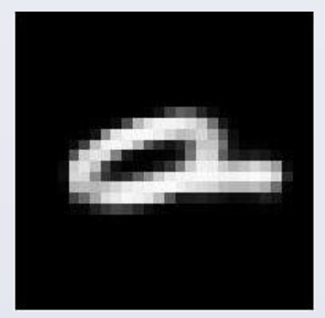
1 - 784-1000-500-30-500-1000-784

2 - 784-500-250-30-250-500-784

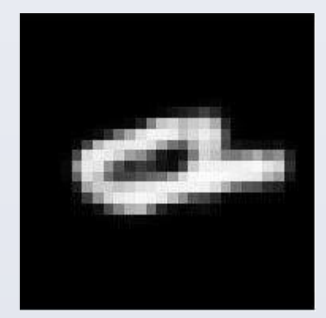
The reconstruction of a particular image over various architecture are -



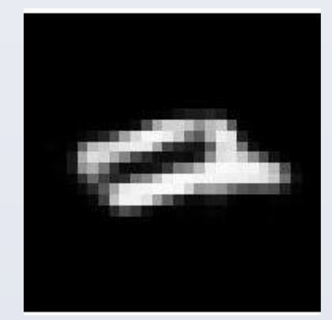
Original Images



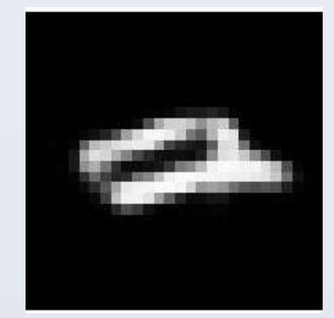
RBM-2



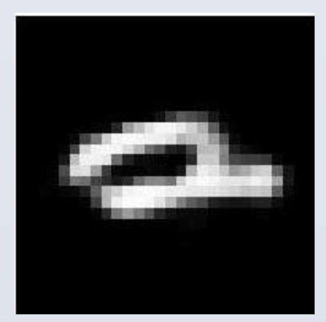
RBM-1



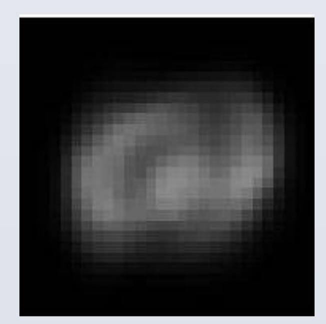
Stacked-2-noise-0.5



Stacked-2-noise-0

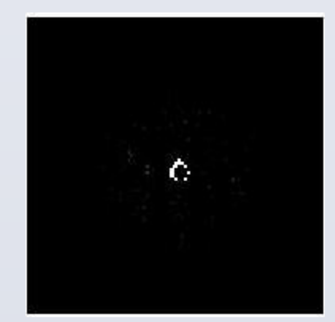


Stacked-1-noise-0.5



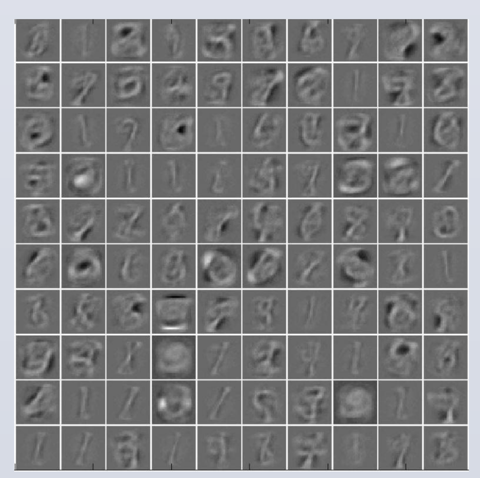
Stacked-1-noise-0

RBM_2D-robot-arm

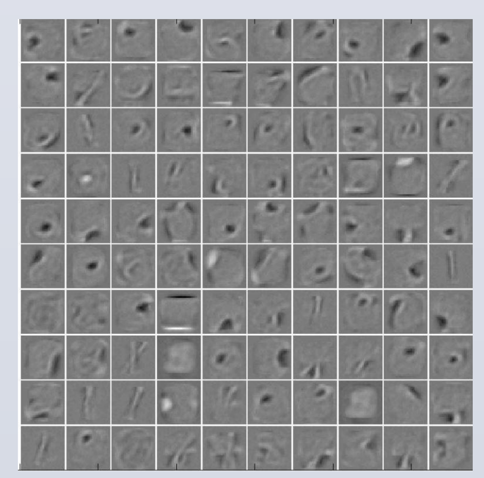


Stacked_2D-robot-arm

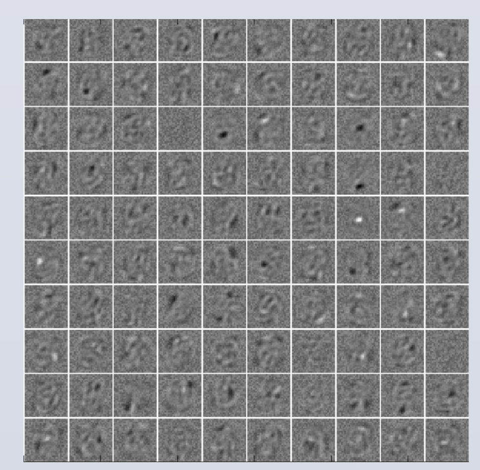
The visualization of the weights are



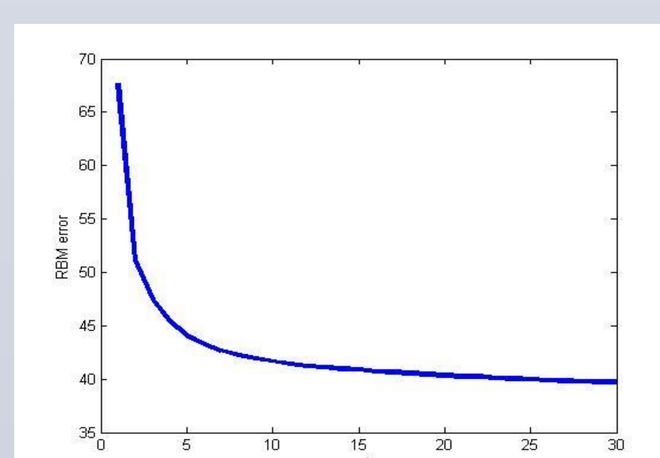
RBM_784_100-epoch_1



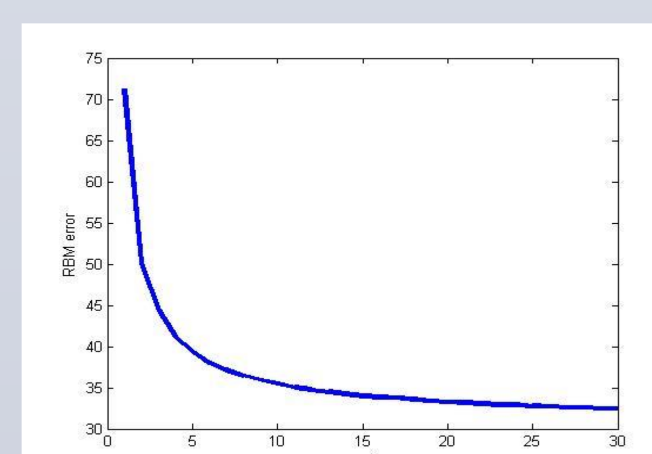
RBM_784_100-epoch_10



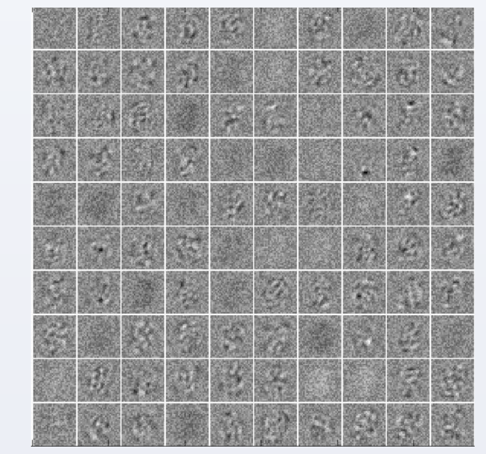
Stacked 784_100 noise 0.5 epoch 1



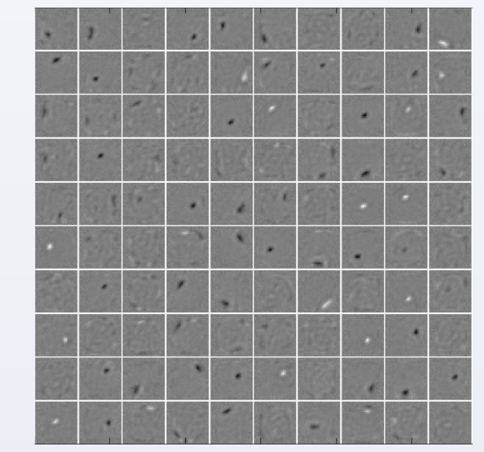
RBM 768_100



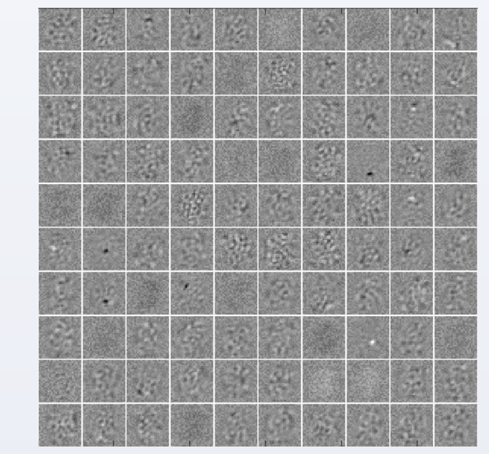
RBM 768_1000



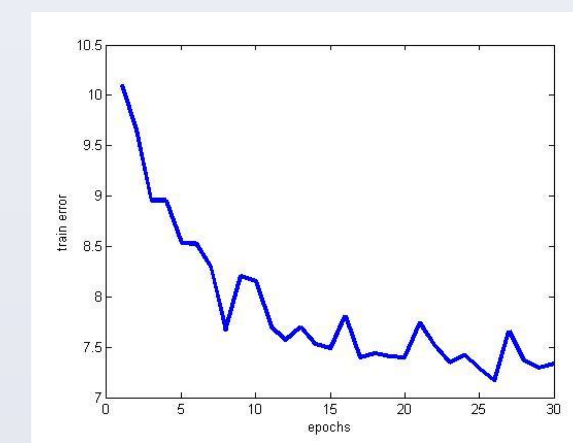
Stacked 784_100 noise 0 epoch 1



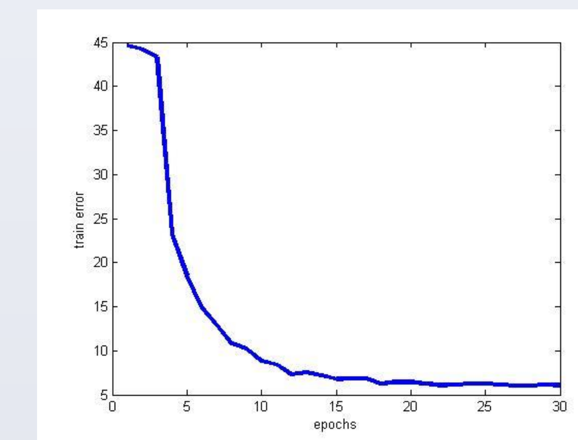
Stacked 784_100 noise 0.5 epoch 1



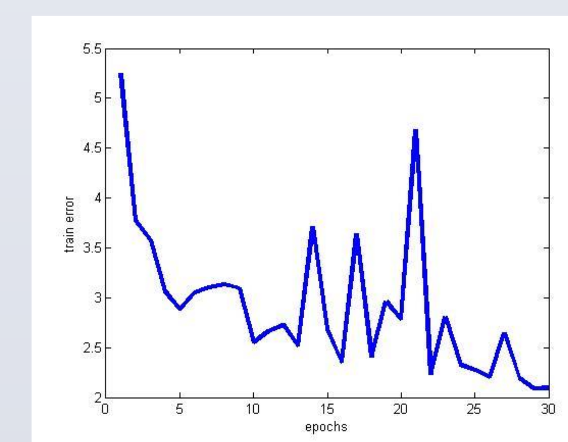
Stacked 784_100 noise 0 epoch 10



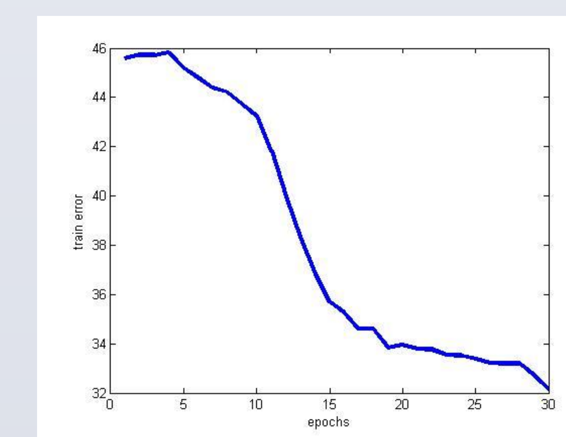
Stacked 784_100 noise 0.5



stacked 784_1000 noise 0.5



Stacked 784_100 noise 0



Stacked 784_1000 noise 0

Reconstruction mean squared over test data for MNIST for various architecture and pre-training methods are -

RBM-784-500-250-30 for 10 epochs - 5.0478

RBM-784-1000-500-30 for 10 epochs - 5.5902

Stacked-784-500-250-30 with 0.5 noise for 10 epochs - 3.3391

Stacked-784-500-250-30 with 0 noise for 10 epochs - 5.0929

Stacked-784-1000-500-30 with 0.5 noise for 10 epochs - 3.7217

Stacked-784-1000-500-30 with 0 noise for 10 epochs - 26.0131

References

[1] Georey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks.Science, 313(5786):504{507, 2006.

[2] Chun Chet Tan and C Eswaran. Reconstruction and recognition of face and digit images using autoencoders.Neural Computing and Applications, 19(7):1069{1079,2010.

[3] Ankit Bhutani, Alternate Layer Sparsity and Intermediate Fine-tuning for Deep Autoencoders

[4] R. B. Palm, Prediction as a candidate for learning deep hierarchical modela of data,2012