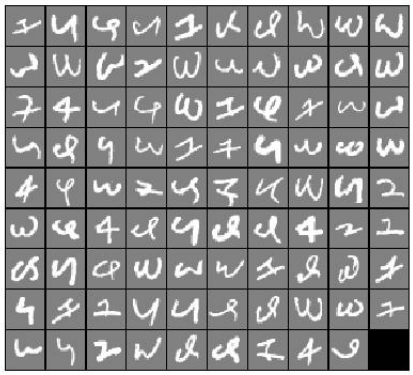
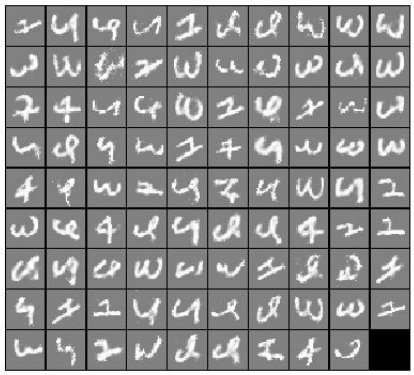
Restricted Boltzmann Machine

Implemented Restricted Boltzmann Machine, and experimented with subset of the MNIST. Only considered 2,3,4 as train set (28x28) from MNIST for the purpose of the time, and there was total of 8937 train data and 1548 test data. I used 400 hidden unit, 0.05 learning rate, and 0.8 final momentum. Here is some sample of data that it generated after running 1000 epoch.

Generating data

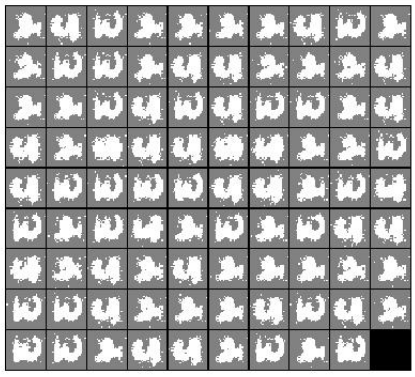




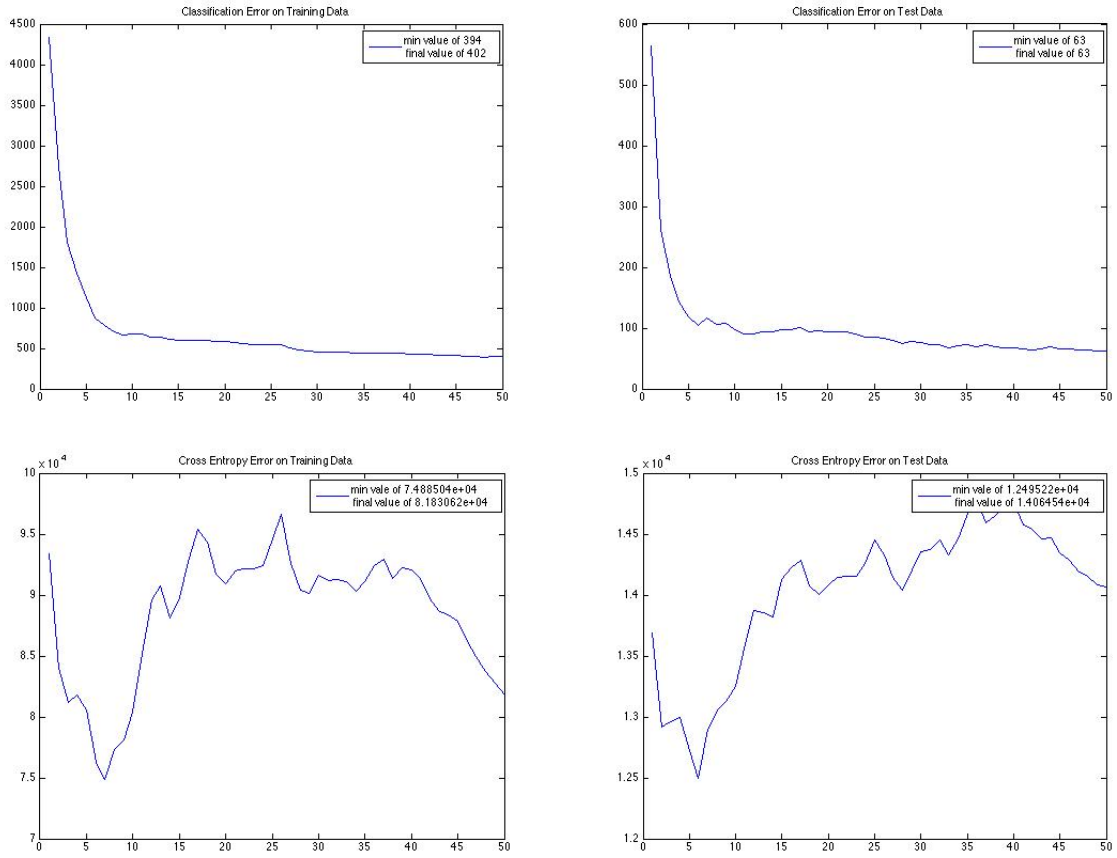
First image is the actual subdata, and second iamge is the subdata that rbm tried to generate.

Classification

After running rbm to train as shown above, and used the weight from rbm as initial weights to learn the neural network.



This image is the image that it generated using the weights after it learned to classify. This network got 0.04 error for train set, and 0.038 error for the test set.



Gaussian Restricted Boltzmann Machine with learning variance

Implemented Gaussian Restricted Boltzmann Machine with same number of hidden units as above, and tested on the same dataset are used as above. The settings for the hyperparameter were 0.001 learning rate, and 0.9 final momentum, and we pretrained on grbm for 850 epoch. Hidden activation units were computed using binary function, and visible units were computed using gaussian function.

$$p(h = 1|v) = \frac{1}{1 + e^{\sum_j w_j v_j + b_j}}$$

$$p(v = x_i|h) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x - b_i - \sigma_i \sum_j h_j w_{ij})}{2\sigma_i^2}}$$

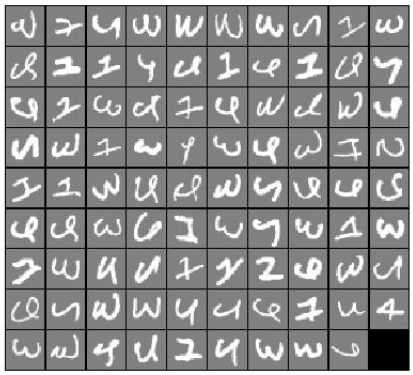
The energy term was:

$$E(X,Y) = \sum_i \frac{x_i - b_i}{2\sigma^2} - \sum_j (b_j h_j) - \sum_{i,j} h_j w_{ij} v_i$$

The gradient of variance was:

$$\frac{\partial log P^*(x)}{\partial \sigma_i} = \sigma_i x_i + 2\sigma_i x_i b_i + \sum_j^N p(h_j|v) W_{ij}$$

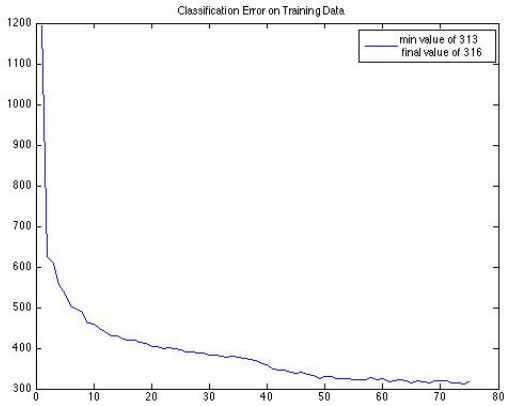
Generating data

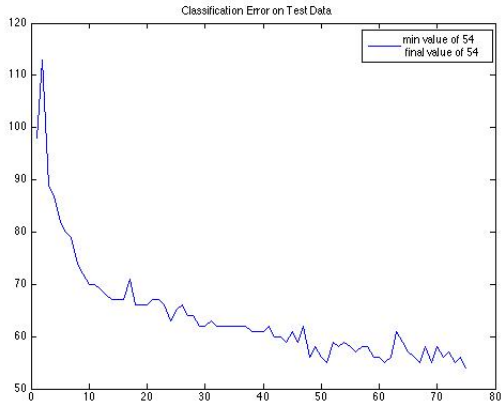


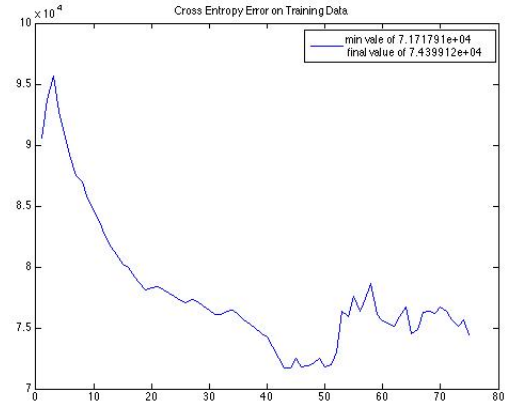
Here are subset of the train data. The figure on the left is the sub-images of the training data. The figure on the right is the generated data using GRBM.

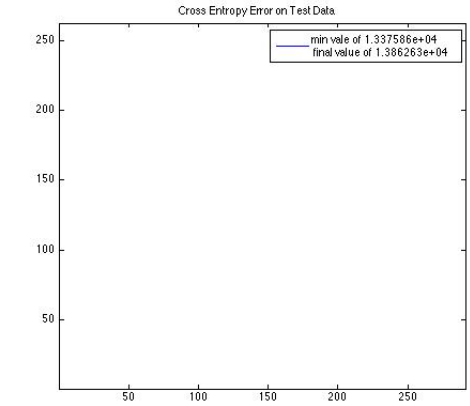
Classification

This image is the image that it generated using the weights after it learned to classify. This network got 0.035 error for train set, and 0.0348 error for the test set.









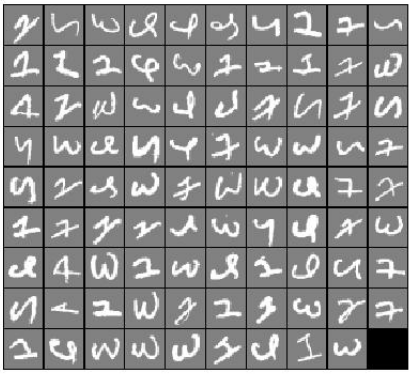
Restricted Boltzmann Machine with Rectified Linear Unit

Implemented RBM with rectified linear unit for the visible data. The setting on the hyper parameter as same as above ones. I used noisy rectified linear unit until the two third of the training, and used regular rectified linear unit last one third of the training.

$$NRLUofP(v|h) = log(1 + e^{y+N(0,1)})$$
$$y = hW + b$$

My hypothesis was that this network should to better than other two, but the square error was around twice higher than regular RBM. (So not sure if my implementation might be wrong).

Generating data



Here are subset of the train data. The figure on the left is the sub-images of the training data. The figure on the right is the generated data using RBM with RLU.

Reference

Y.Tang, A. Mohamed. Multiresolution Deep Belief Networks.

R.R. Salakhudinov, G.E.Hinton. Using Deep Belief Nets to Learn Covariance Kernel for Gaussian Process.

V. Nair, G.E.Hinton. Rectified Linear Units improves Restricted Boltzmann Machine. Proc. 27th International Conference on Machine Learning.