CS498 Homework 7

Xinchen Pan, Jiefei Shi, Fangxiaozhi Yu

April 14, 2017

Mean field inference for binary images

After downloading the data, we modified the function from https://martin-thoma.com/classify-mnist-with-pybrain/ to load the dataset. For the training dataset, there are 60000 instances together. Since each instance is 28×28 pixel grey scale image, we have 784 pixel values in total for each of them.

We binarized the first 500 images by mapping any value below 0.5 to -1 and any value above to 1. We then created a noisy version by randomly flipping 2%. For each of the pixel value, we generated a binomial random variable with 98% probability. If it is 1, we did not change value. Else, we changed the value to the opposite.

We denoised the images using

$$\log q_i(1) = \sum_{j \in N(i) \cap H} [\theta_{ij}(2\pi_j - 1)] + \sum_{j \in N(i) \cap X} [\theta_{ij}(X_j)] + K$$

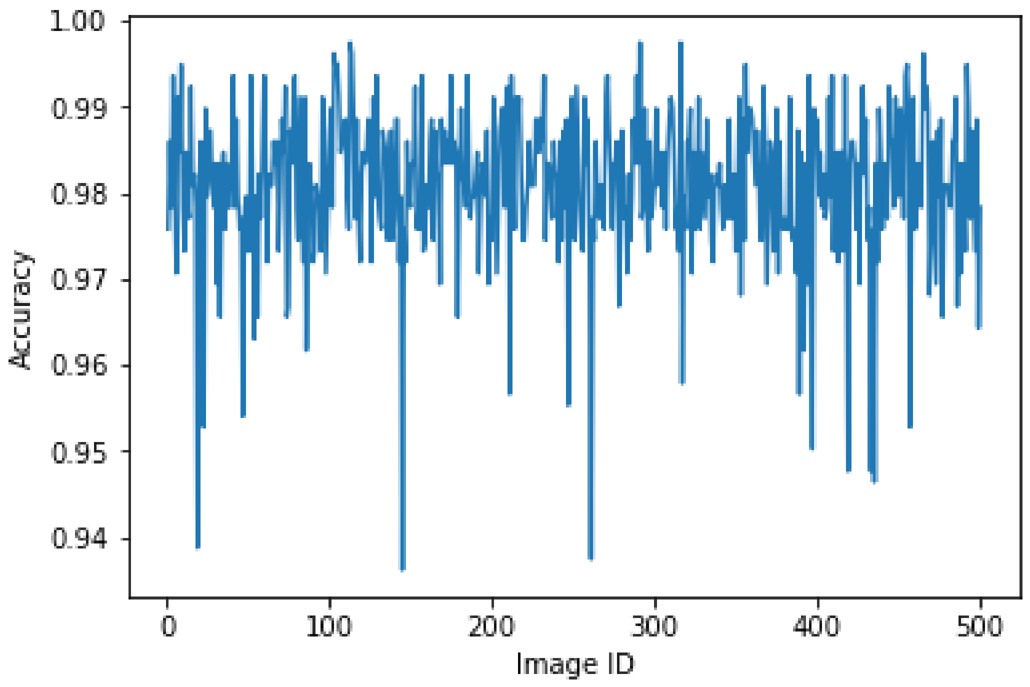
$$\log q_i(-1) = \sum_{j \in N(i) \cap H} [-\theta_{ij}(2\pi_j - 1)] + \sum_{j \in N(i) \cap X} [-\theta_{ij}(X_j)] + K$$

$$\pi_i = \frac{e^{\log q_i(1)}}{e^{\log q_i(1)} + e^{\log q_i(-1)}}$$

We checked convergence by calculating the sum of squared error of noise pixel and denoised pixel in each rotation. If it is smaller than a certain number, then we stopped the iterations.

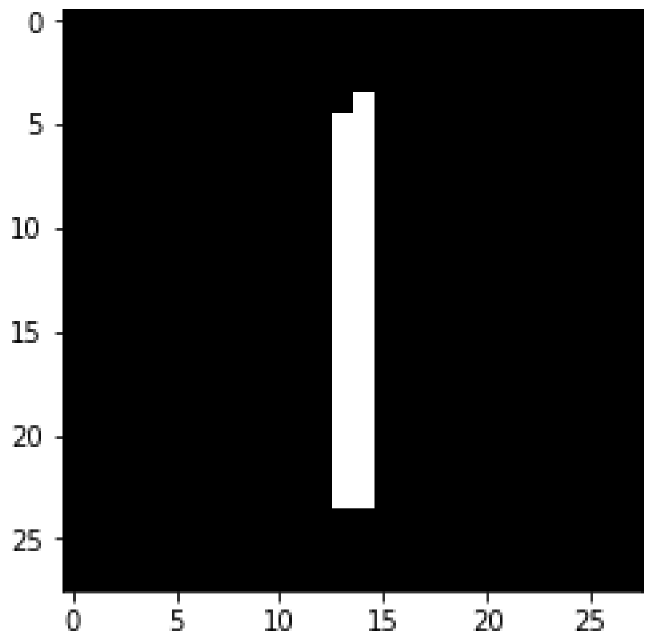
The fractions of all pixels are correct

The accuracies are in the csv file and we made a plot using these 500 values.

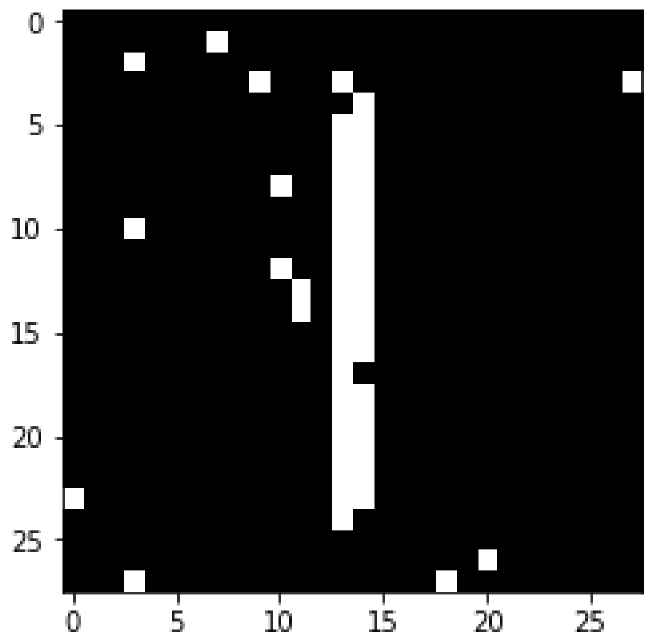


A figure showing the origin image, the noise image, and the reconstruction image

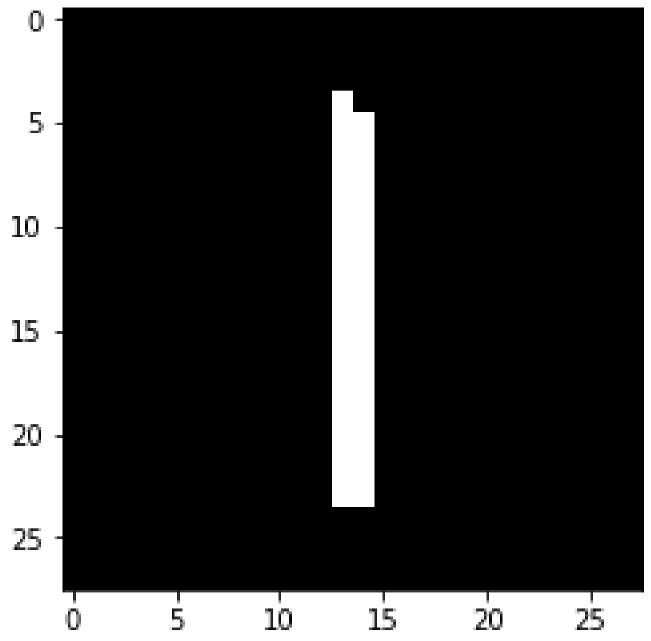
Original Image



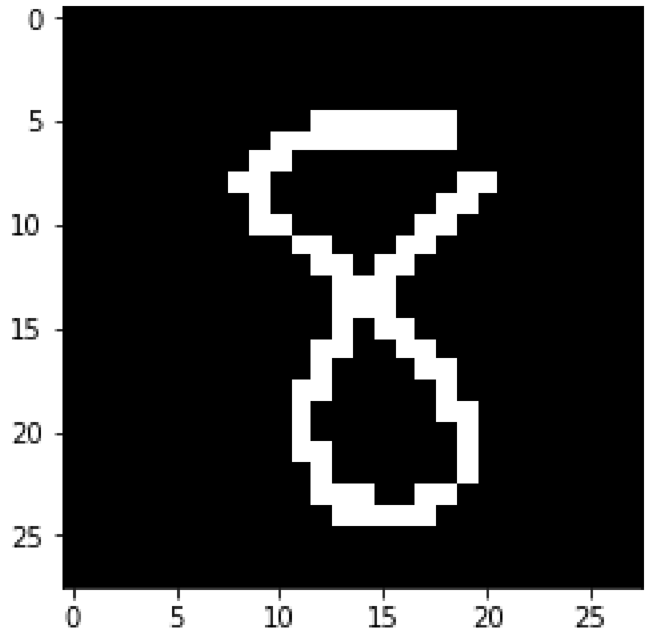
Noise Image



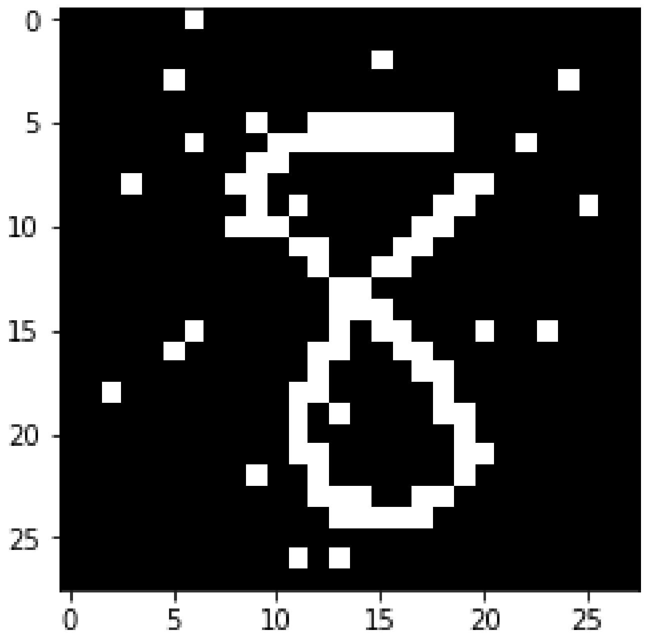
Reconstruction Image (most accurate)



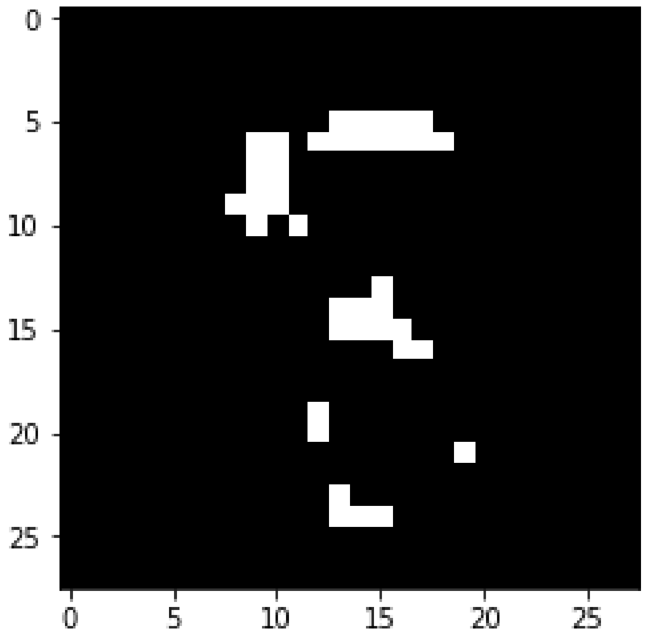
Original Image



Noise Image



Reconstruction Image (Least accurate)

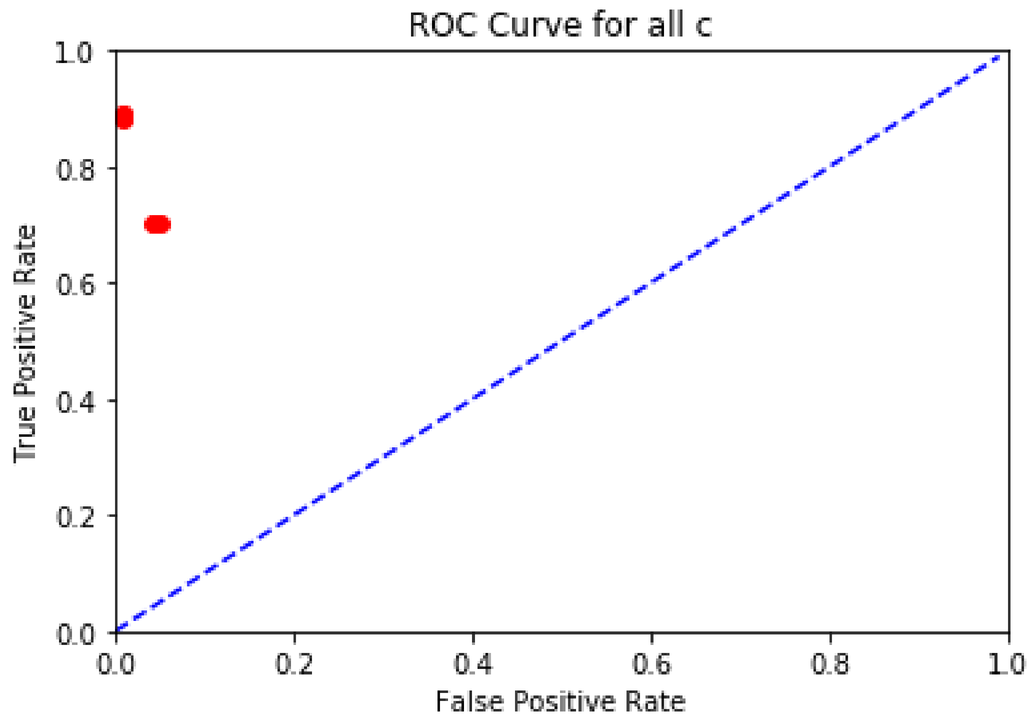


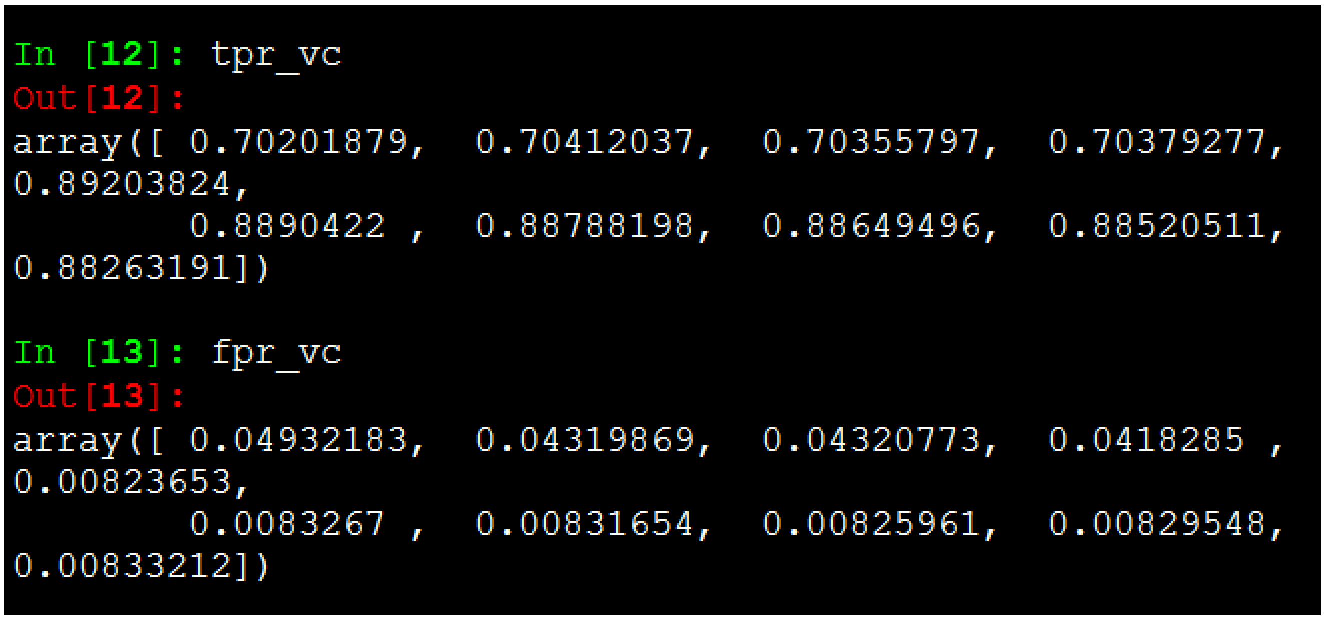
Roc curve

We tried 0 different c values, which are $-1, -0.6, -0.4, -0.25, 0.2, 0.3, 0.5, 0.8, 0.9, 1$.

The average true positive rate and false negative rate over 500 images are very close.

Below is the ROC curve plot which contains 10 points. Because the values are very close, it looks like therea are only two points.





```
In [12]: tpr_vc
Out[12]:
array([ 0.70201879,  0.70412037,  0.70355797,  0.70379277,
0.89203824,
        0.8890422 ,  0.88788198,  0.88649496,  0.88520511,
0.88263191])

In [13]: fpr_vc
Out[13]:
array([ 0.04932183,  0.04319869,  0.04320773,  0.0418285 ,
0.00823653,
        0.0083267 ,  0.00831654,  0.00825961,  0.00829548,
0.00833212])
```