Topics in Deep learning: Assignment 1

Shiyu Dong (shiyud)
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
shiyud@andrew.cmu.edu

1 Problem 1

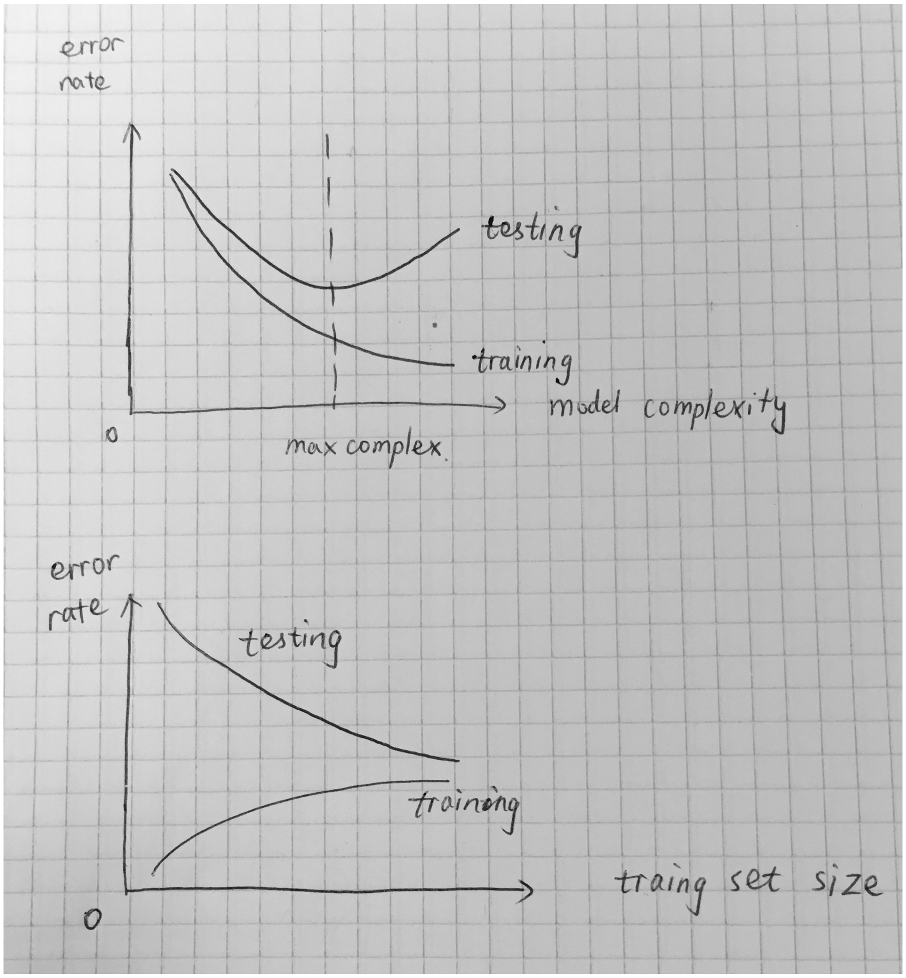

Figure 1: problem 1 figure

2 Problem 2

a) The minimum expected loss can be found by minimizing the following equation:

$$\int |y(x) - t|^q p(t|x)dt$$

for any possible value x.

To find the minimum value, we need to set the derivative to zero, which is:

$$\int q|y(x) - t|^{q-1} sign((y(x) - t))p(t|x)dt$$

$$= q\int_{-\infty}^{y(x)} (y(x) - t)^{q-1}p(t|x)dt - q\int_{y(x)}^{\infty} (t - y(x))^{q-1}p(t|x)dt = 0$$

By using $q = 1$, we can simplify the equation as:

$$\int_{-\infty}^{y(x)} p(t|x)dt = \int_{y(x)}^{\infty} p(t|x)dt$$

This means the function $y(x)$ we choose will make the probability mass for $t < y(x)$ is the same as for $t > y(x)$

b) For $q = 0$, we can notice that: $|y(x) - t|^q = 0$ for most of the time and except for when $y(x) = t$, the result turns to 1.

To minimize the expected loss, we want to set the maximum value of p(t|x) when $y(x) - t = 0$. Therefore it's given by the conditional mode.

3 Problem 3

If classes are all correctly labelled, we can get:

$$p(t|x, w) = \prod_{n=1}^{N} [y_n^{t_N}(1 - y_n)^{1-t_n}]$$

Take the negative log, we can get:

$$E = -\ln p(t|x, w) = -\sum_{n=1}^{N}[t_n \log y_n + (1 - t_n) \log(1 - t_n)]$$

which is known as the cross entropy error function.

If there are incorrectly labelled data, then the likelihood function will change to:

$$p(t|x, w) = \prod_{n=1}^{N} [(1 - \varepsilon)y_n + \varepsilon(1 - y_n)]^{t_N} [(1 - \varepsilon)(1 - y_n) + \varepsilon y_n]^{1-t_n}$$

Take the negative log, we can get:

$$E = -\sum_{n=1}^{N}[t_n \log((1 - \varepsilon)y_n + \varepsilon(1 - y_n)) + (1 - t_n) \log((1 - \varepsilon)(1 - y_n) + \varepsilon y_n))]$$

4 Problem 4

1. To prove this distribution is normalized, integrate the distrubution over x.

$$\int p(x|\sigma,q)dx = \frac{q}{2(2\sigma^2)^{1/q}\Gamma(1/q)}\int \exp(-\frac{|x|^q}{2\sigma^2})dx$$

Define I, where

$$I = \int_{-\infty}^{\infty} \exp(-\frac{|x|^q}{2\sigma^2})dx = 2\int_{0}^{\infty} \exp(-\frac{x^q}{2\sigma^2})dx$$

Then Define u, where

$$u = \frac{x^q}{2\sigma^2}$$

so that,

$$I = 2\int_{0}^{\infty} \exp(-u)d(2u\sigma^2)^{1/q}$$

$$I = 2\int_{0}^{\infty} \frac{2\sigma^2}{q}(2\sigma^2u)^{1-q/q}\exp(-u)du = \frac{2(2\sigma^2)^{1/q}\Gamma(1/q)}{q}$$

so that,

$$\int p(x|\sigma,q)dx = 1$$

So the distribution is normalized.
When $q = 2$, $\Gamma(1/2) = \sqrt{\pi}$, so that,

$$p(x|q) = \frac{1}{\sqrt{2\sigma^2\pi}}\exp(-\frac{|x|^q}{2\sigma^2})$$

which we can conclude is a Gaussian distribution.
2. The likelihood function is:

$$p(t|w,\sigma^2) = \frac{q}{2(2\sigma^2)^{1/q}\Gamma(1/q)}\int \exp(-\frac{|t-y(x,w)|^q}{2\sigma^2})$$

Derive the log likelihood:

$$\log p(t|w,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}|y(x_n,w)-t_n|^q - \frac{N}{q}\log(2\sigma^2) + C$$

So that,

$$max(\log p(t|w,\sigma^2)) = min(\frac{1}{2\sigma^2}\sum_{n=1}^{N}|y(x_n,w)-t_n|^q + \frac{N}{q}\log(2\sigma^2))$$

5 Problem 5

a) For both training set and validation set, they all start from high cross entropy. And the cross entropy drops as we keep on training.

For training set, the cross entropy drops faster and it can decrease to zero after several epochs.

For validation set, the cross entropy drops slower and it will end up at around 0.0 and will eventually slightly increase as we keep training.

In the figure blow, the red dot line stands for training set and the blue dot line stands for validation set.

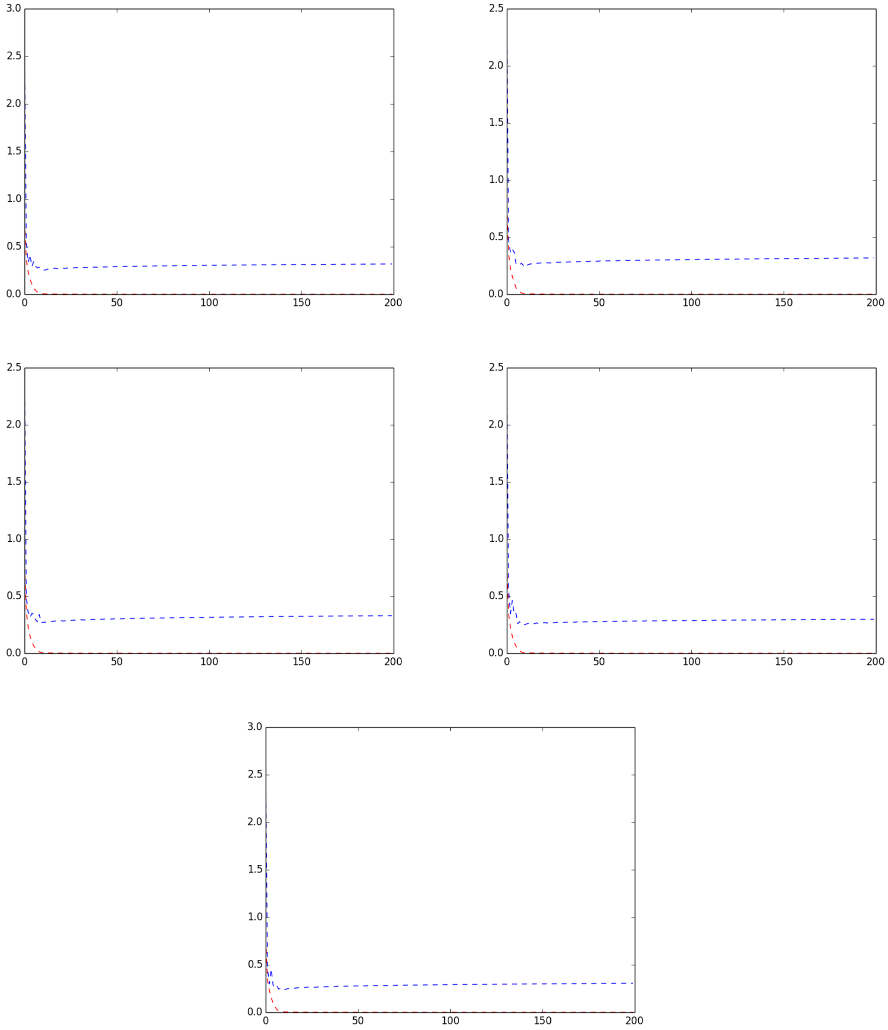

Figure 2: Cross Entropy error

b) For classification error, the validation set finally have a error rate of about 0.08. The plot looks very similar to the cross-entropy error.

In the figure blow, the red dot line stands for training set and the blue dot line stands for validation set.

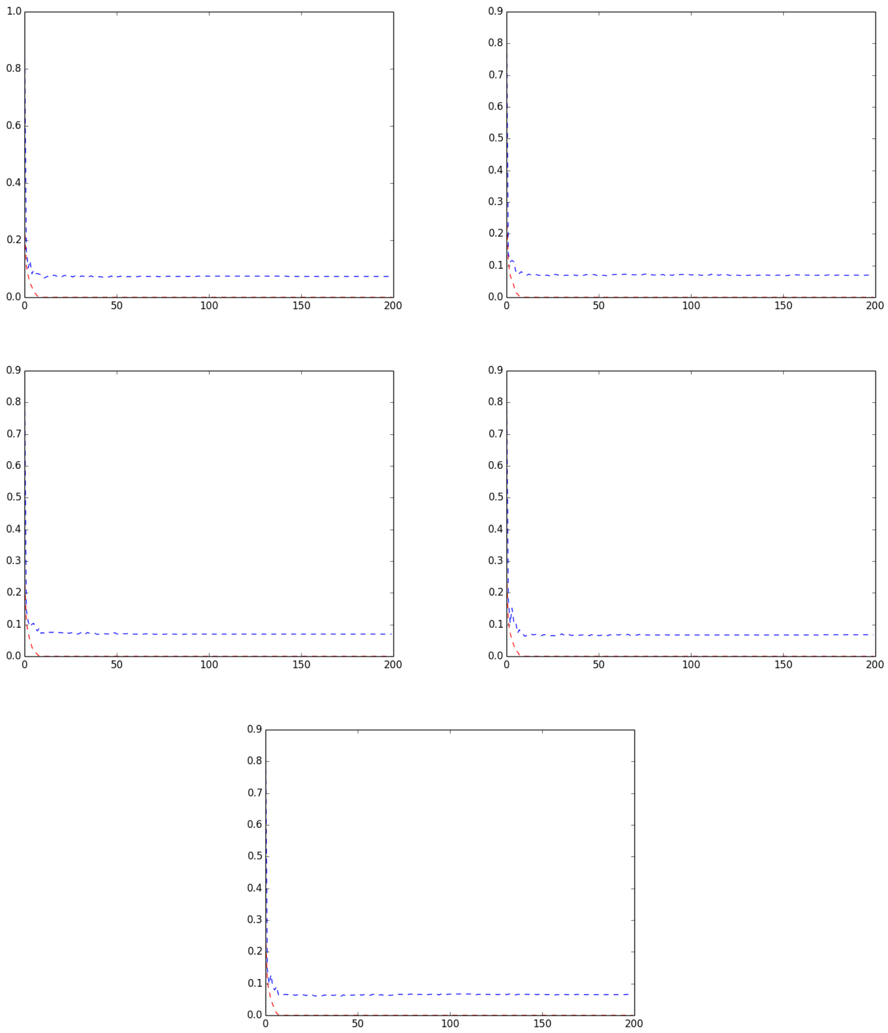

Figure 3: Classification error

c) One of the best results is shown as the figure blow.(learning rate: 0.1, hidden layer: 100)

We can observe from the figure that some parameters has been visualized as a handwritten number shape.

For example, (column 1, row 4) looks like number 3, etc.

We can interpret the parameter as a image filter and apply the neural network is to apply the image filter as sliding window.

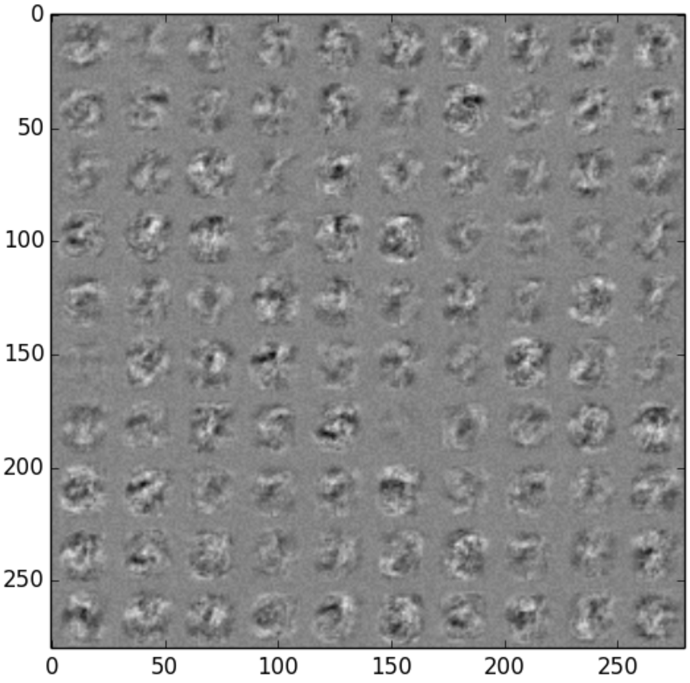

Figure 4: Visualizing Parameters

d) We can observe that: a lower rate (like 0.01) will make the training converge slower. A faster rate (like 0.1, 0.2) will speed up the converge. But if the learning rate is too fast (like 0.5), the learning will get a positive feedback and it will not converge.

According to the plot, a learning rate like 0.1 or 0.2 will speed up the training and is able to converge. So it's best to choose a value that is close to 0.1 and 0.2.

Figure 5-8 shows the performance for different learning rate, where the red dot line stands for training set and the blue dot line stands for validation set. The left images are cross-entropy error and the right images are classification error.

Figure 9-11 shows the performance for different momentum. Momentum helps getting out of local minimum and speed up the converge. But when the momentum is too large (like 0.9), it will not converge.

e) In figure 9-12, the red dot line stands for training set and the blue dot line stands for validation set. The left images are cross-entropy error and the right images are classification error.

The hidden units doesn't effect too much on the converge. But the hidden units of 20 and 500 show a little bit of overfitting.

f) Dropout speeds up the convergence. We can also achieve a smaller error rate with dropout. According to the figure 16-17, dropout works slightly better for a larger network.

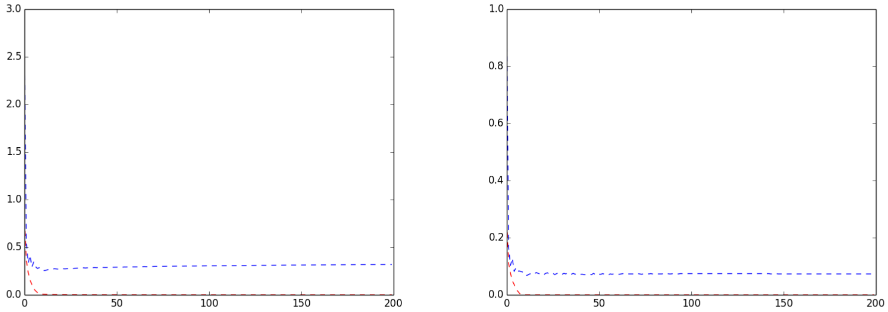

Figure 5: Learning rate 0.1

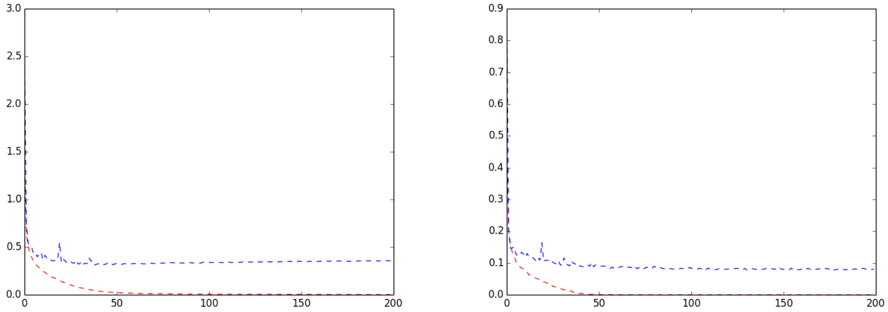

Figure 6: Learning rate 0.01

g) Parameter:

- learning rates: 0.1
- momentum: 0.2
- number of hidden units: 100
- number of epochs: 100
- L2 regularization: No
- Dropout values: No dropout

Terminal command to run:

```
python BasicNN.py digitstrain.txt digitsvalid.txt digitstest.txt
--rate 0.1 --momentum 0.2  --layer 784 100 10 --epoch 100
```

Performance:

- training: cross-entropy error: 0.000271327328539, classification error: 0.0
- validation: cross-entropy error 0.304652124923 classification error: 0.07
- test error cross-entropy error 0.378257010894 classification error: 0.077

Figure 18 shows the cross-entropy error and classification error, where the green dot line stands for testing set, the red dot line stands for training set and the blue dot line stands for validation set.

Figure 19 shows the visualization of W.

h) Parameter:

- learning rates: 0.1

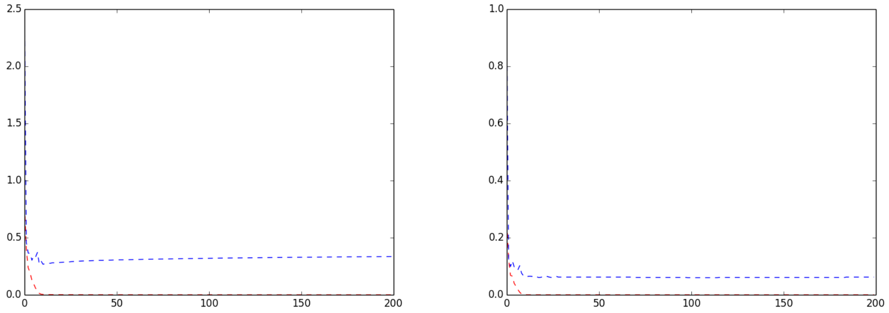

Figure 7: Learning rate 0.2

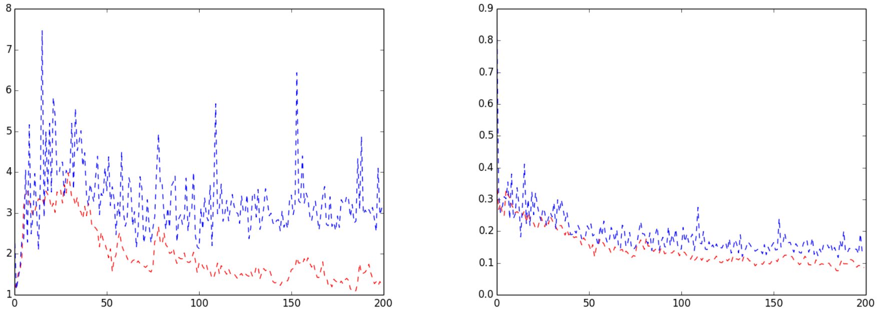

Figure 8: Learning rate 0.5

- momentum: 0.2
- number of hidden units: 200, 100
- number of epochs: 100
- L2 regularization: No
- Dropout values: No dropout

Terminal command to run:

```
python TwoLayerNN.py digitstrain.txt digitsvalid.txt digitstest.txt
--rate 0.1 --momentum 0.2  --layer 784 200 100 10 --epoch 100
```

Performance:

- training: cross-entropy error: 0.000109742784164, classification error: 0.0
- validation: cross-entropy error 0.392318928081 classification error: 0.064
- test error cross-entropy error 0.452778623363 classification error: 0.0736666666667

Figure 20 shows the cross-entropy error and classification error, where the green dot line stands for testing set, the red dot line stands for training set and the blue dot line stands for validation set.

Figure 21 shows the visualization of W.

Compared to single layer network, it has slightly higher cross-entropy error for both validation and testing set, but slightly lower classification error. It also converge a little bit slower.

The visualizing of first layer parameters is roughly the same as single layer network. But it's more clear. The visualization of single layer is more blurred compared to two layer network.

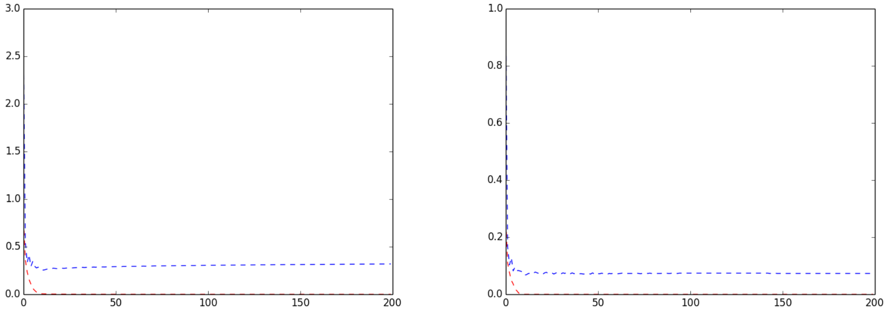

Figure 9: Momentum 0

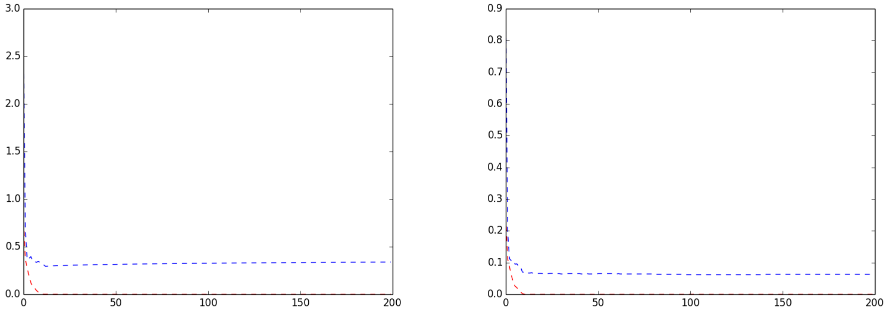

Figure 10: Momentum 0.5

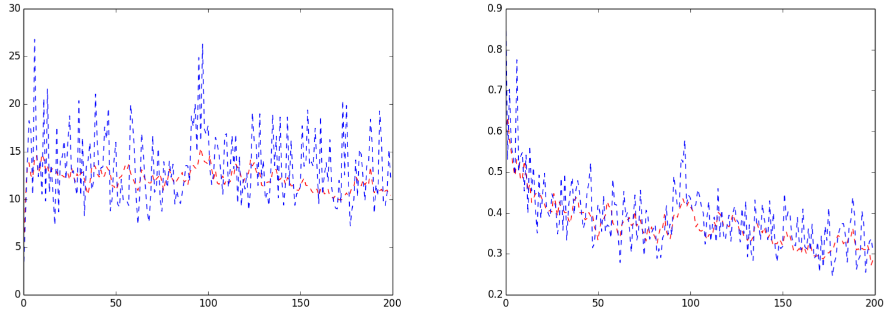

Figure 11: Momentum 0.9

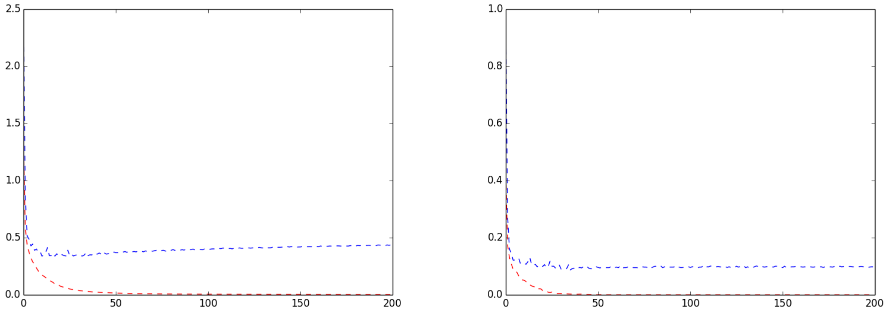

Figure 12: Hidden Units 20

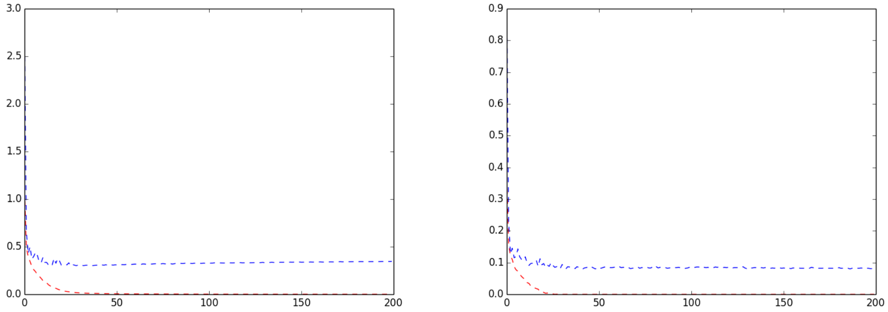

Figure 13: Hidden Units 100

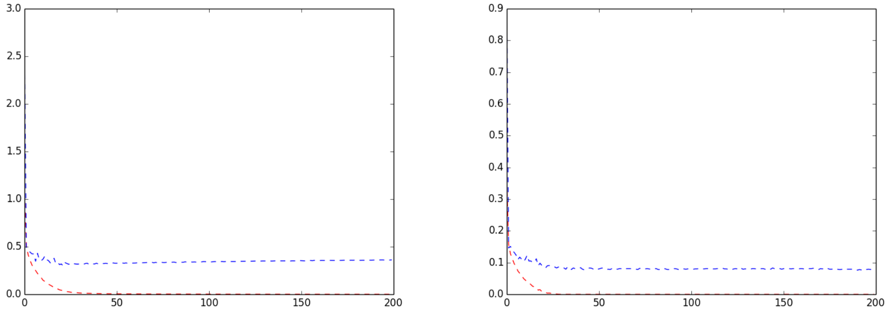

Figure 14: Hidden Units 200

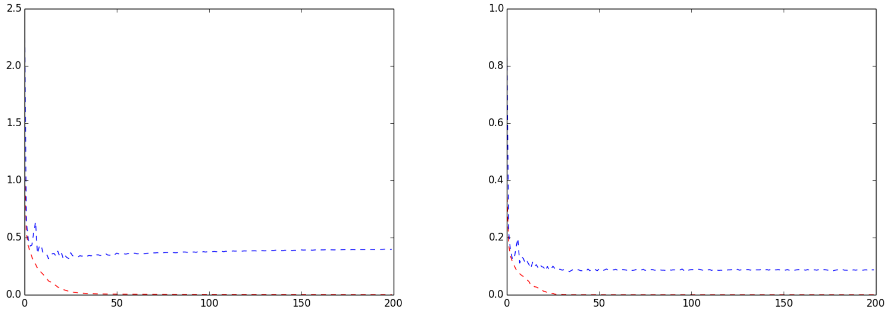

Figure 15: Hidden Units 500

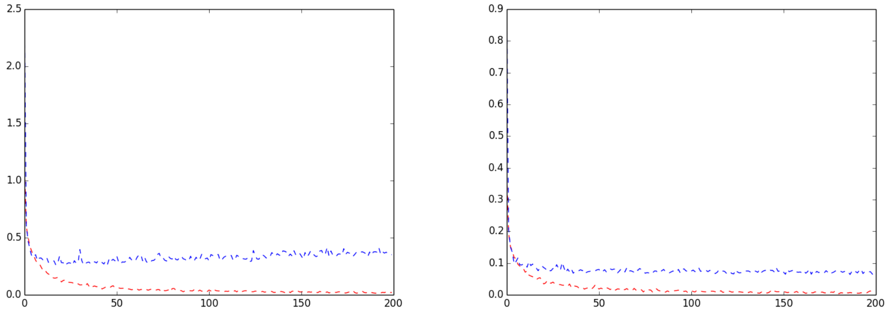

Figure 16: Dropout 0.5 with hidden unit 100

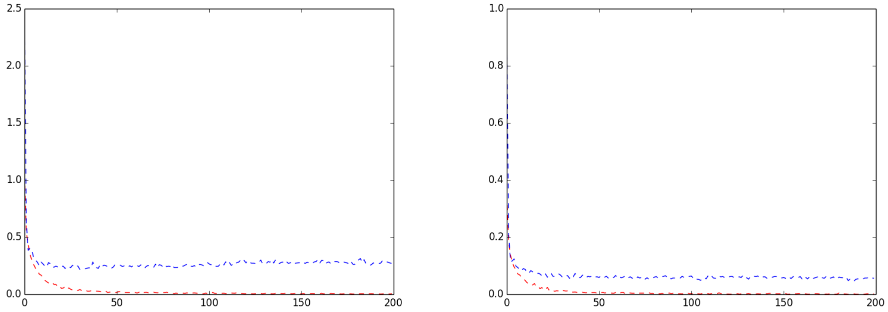

Figure 17: Dropout 0.5 with hidden unit 200

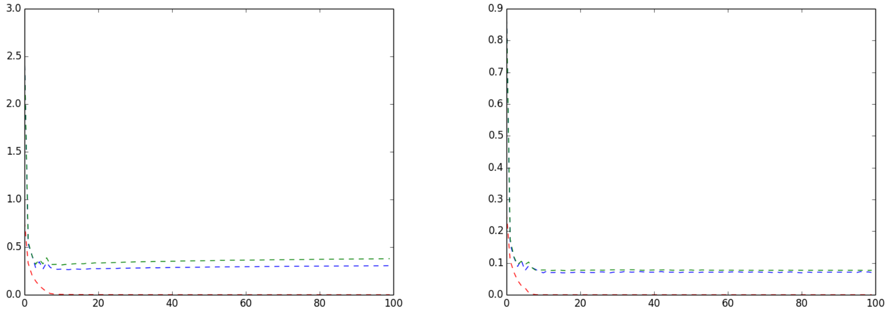

Figure 18: Best performing single layer network

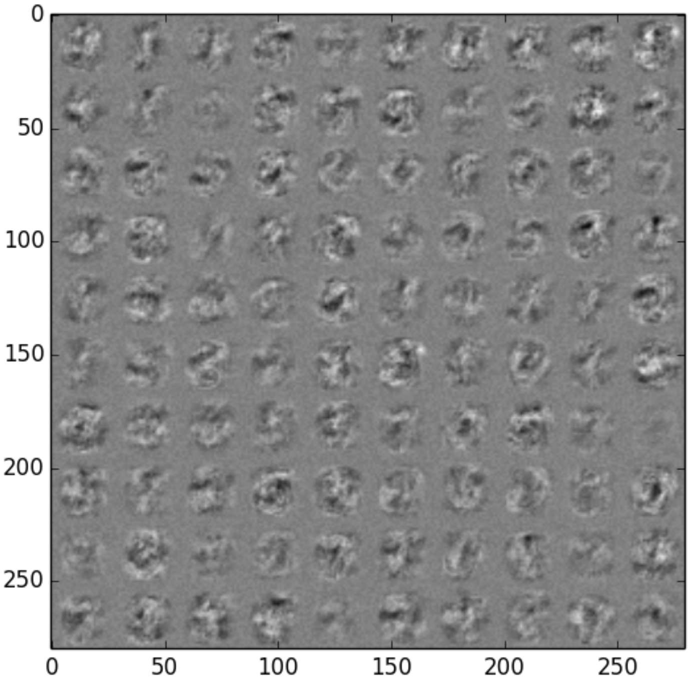

Figure 19: Best performing single layer network

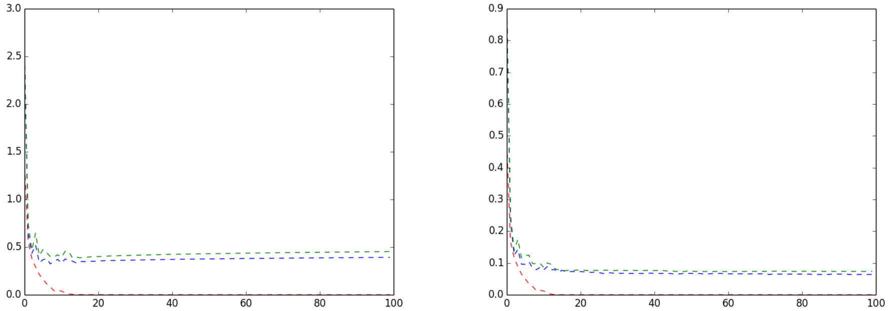

Figure 20: Best performing two layer network

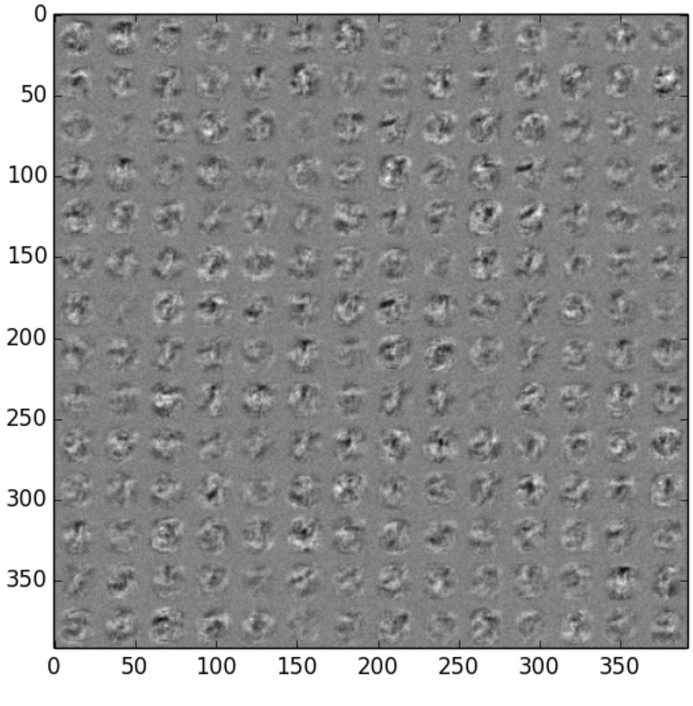

Figure 21: Best performing two layer network