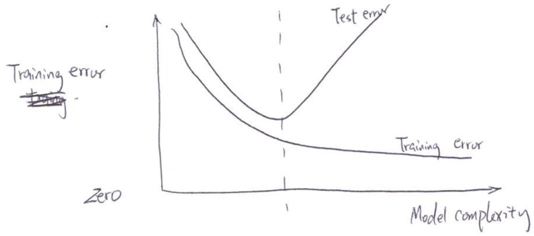
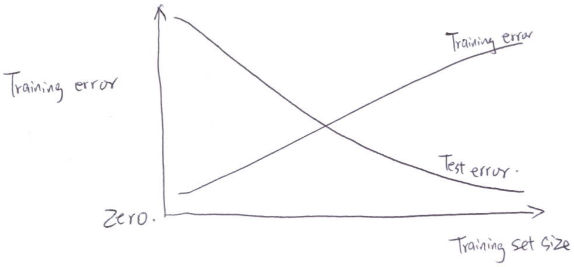
Assignment1

Name: Yang Zhang Andrew ID: yangz3
yang.zhang@cs.cmu.edu

1 Problem 1





2 Problem 2.a)

We want to prove

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

The minimum of the above loss funtion happens when the derivative equals 0. When q = 1 it becomes:

$$\frac{\partial \int |y(x)-t| p(x,t)dt}{\partial y(x)} = 0$$

Dividing the absolute into two range we get :

$$\int_{-\infty}^{y(x)} p(x,t)dt - \int_{y(x)}^{+\infty} p(x,t)dt = 0$$

10807 Class Project (2016).

3 Problem 3)

The error function corresponding to the negative log-likelihood:

$$E(w) = -\sum_i (t^i ln(y^i(1-\epsilon) + (1-y^i)\epsilon) + (1-t^i)ln((1-y^i)(1-\epsilon) + y^i\epsilon))$$

This error function makes the model robust to incorrectly labelled data, in contrast to the usual error function.

4 Problem 4.1)

The integration of the distribution has to be 1 in order to be a valid probability.

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

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

$$= \frac{q}{2(2\sigma^2)^{1/q}\Gamma(1/q)} \int_0^\infty exp(-\frac{x^q}{2\sigma^2})$$

According to Gamma function defination,

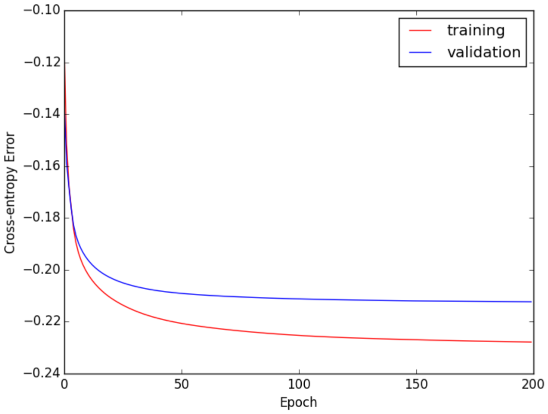
$$\int_0^\infty exp(-\frac{x^q}{2\sigma^2}) = \frac{2(2\sigma^2)^{1/q}\Gamma(1/q)}{q})$$

Thus

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

5 Problem 5.a Basic generalization)

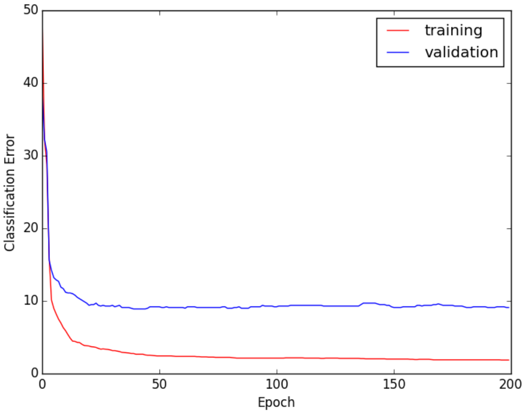
Below is the plot of the cross-entropy error over epoches, where I used a learning rate of 0.1, momentum of 0.5, 200 epoches with 100 hidden units:



We can see from the plot that the validation error reaches the bottom after around 20 epoches, however the training error keeps decreasing. In addition, after the early-stopping point, the validation error increases while the training error decreases which clearly indicates a overfitting.

6 Problem 5.b Classification error)

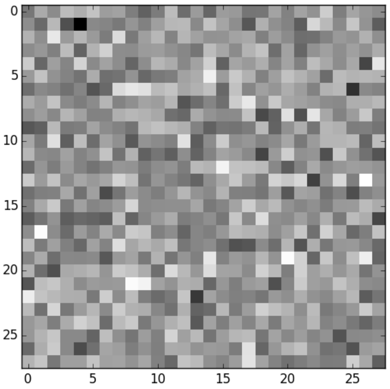
Below is the plot of the classification error over epoches, where I used a learning rate of 0.1, momentum of 0.5, 200 epoches with 100 hidden units:



We can see from the plot of classification error and cross-entropy error that there both are early-stopping points and the behaviours of the two error function are similar.

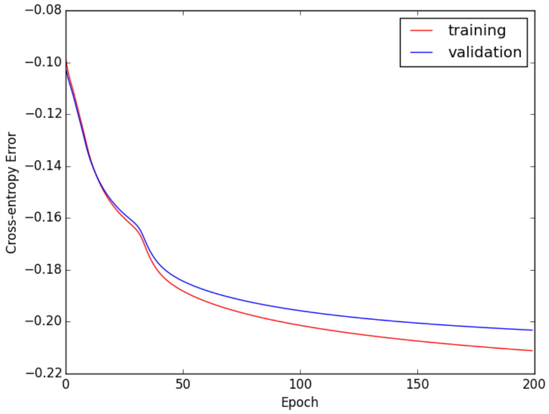
7 Problem 5.c Visualizing Parameters)

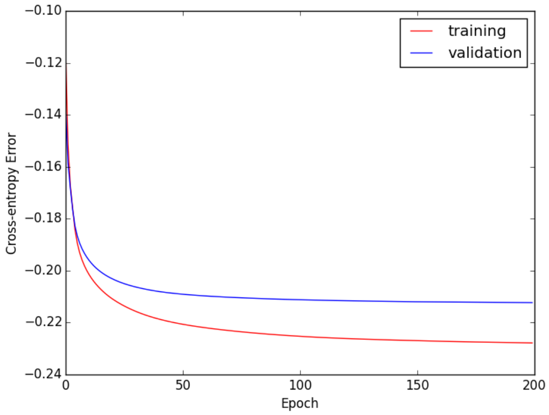
Below is the plot of the learned W as 100 28x28 images overlapped. In this picture brighter pixels indicate bigger weights

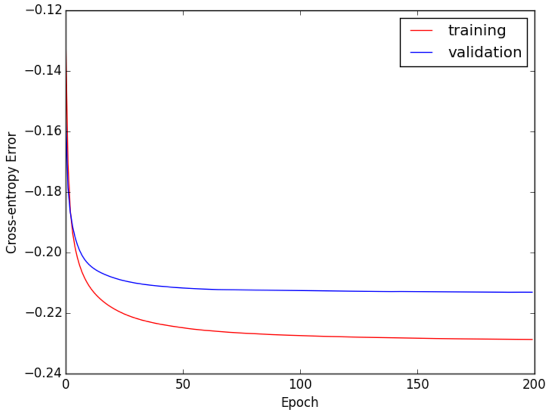


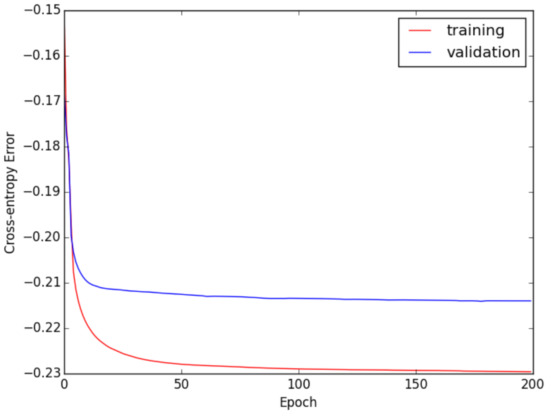
8 Problem 5.d Learning rate)

Below are the plots of cross-entropy error with learning rates from 0.01, 0.1, 0.2 to 0.5 with fixed momentum of 0.5:


−0.08
−0.10
−0.12
−0.14
−0.16
−0.18
−0.20
−0.22
Cross-entropy Error
0
50
100
150
200
Epoch
training
validation

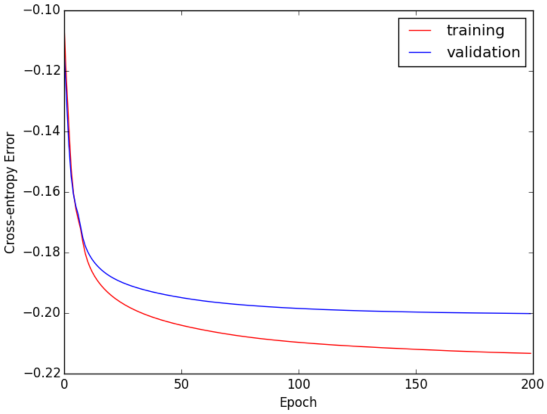

−0.10
−0.12
−0.14
−0.16
−0.18
−0.20
−0.22
−0.24
Cross-entropy Error
0
50
100
150
200
Epoch
training
validation

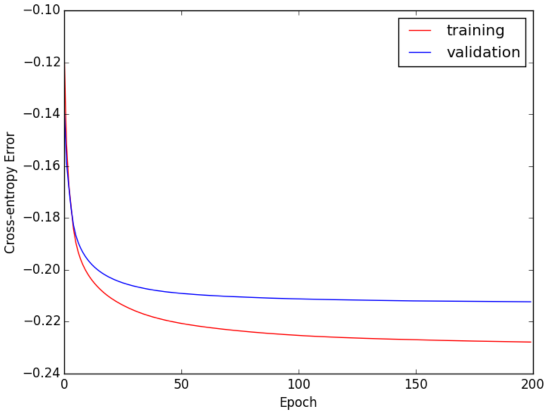

−0.12
−0.14
−0.16
−0.18
−0.20
−0.22
−0.24
Cross-entropy Error
0
50
100
150
200
Epoch
training
validation

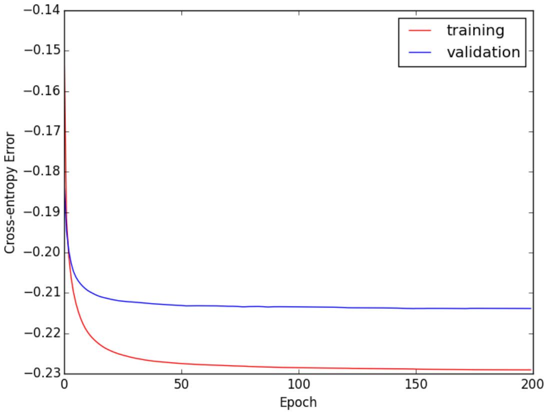


As we can see from the plots with different learning rates, bigger learning rates tend to converge faster.

Below are the plots of cross-entropy error with momentum of 0.0, 0.5, 0.9 with a fixed learning rate of 0.1:





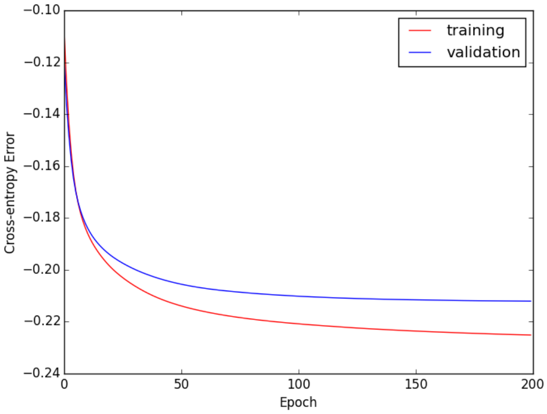


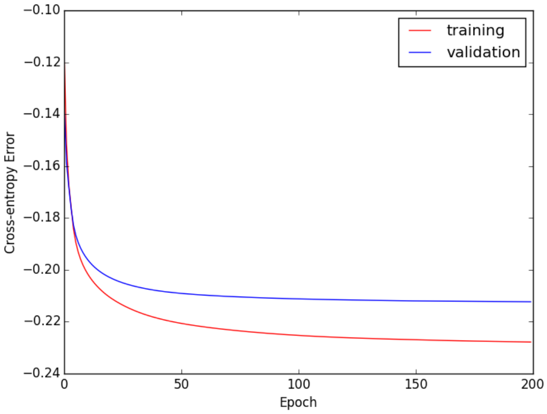
As we can see from the plots with different momentums, bigger momentum tend to converge faster. It also has comparable cross-entropy error as small momentums.

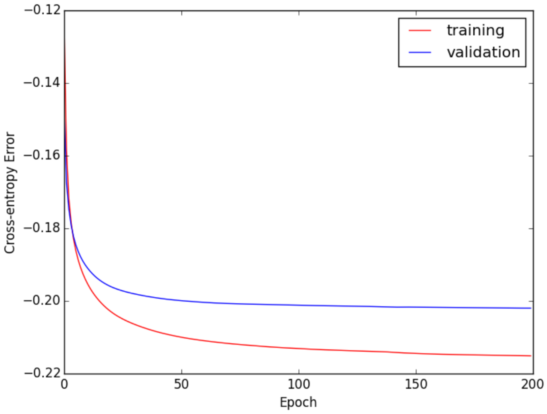
Combining the result from learning rate and momentum, I would choose the best value to be around 0.5 learning rate and around 0.9 momentum. To finalize the parameters, I will perform little tweaks on the learning rate and momentum, but around these values.

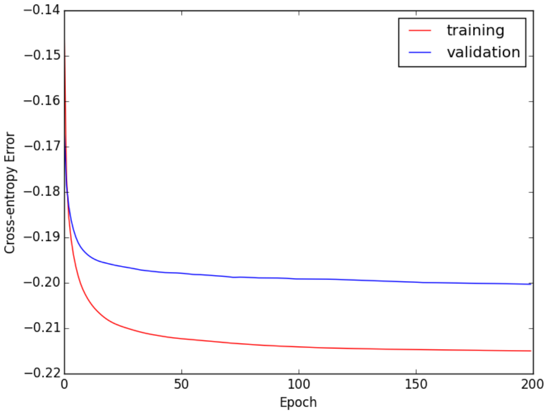
9 Problem 5.e) Number of hidden units

Below are the plots of cross-entropy error with different numbers of hidden units (i.e. 20, 100, 200, and 500):









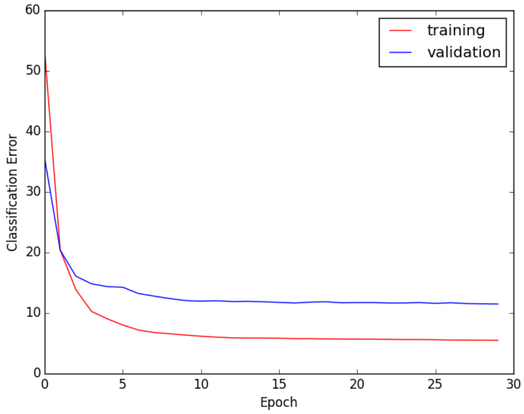
According to what I found, networks with less hidden units took significantly shorter time to train each epoch. However, networks with more hidden units converge faster in terms of number of epoches. For generalization, more hidden units tend to generate slightly worse than less hidden units. 500-hidden-unit network has 0.081 more cross-entropy error than 20-hidden-unit network.

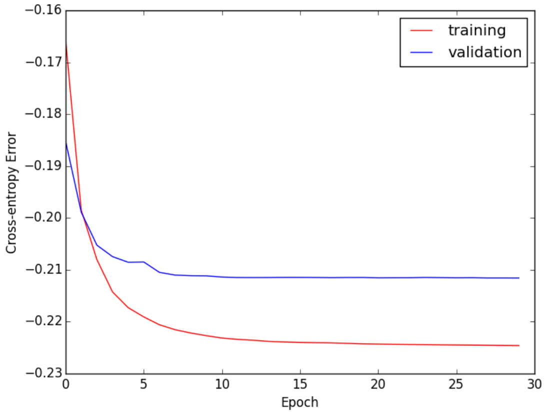
10 Problem 5.f) Dropout

Didn't implement this functionality in the current network. Will catch up and implement in future.

11 Problem 5.g) Best performing single-layer network

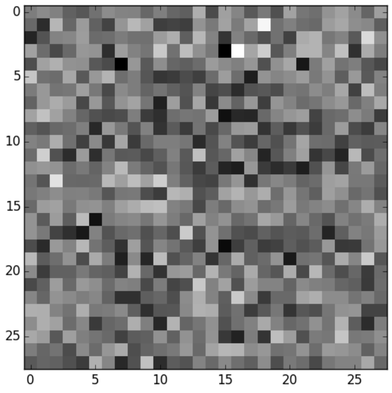
In the cross-validation, I explored the learning rates, momentum, the number of hidden units and the number of epoches. I found the learning rate of 0.6, momentum of 0.9, with 100 hidden units and 30 epoches tend to provide the best result. My rubrics are both the converging time (how fast it nees to train) and the final accuracy (how accurate is the trained model). Blow is a plot of using these parameters to train the single-layer network:





The result indicates error rate of 8.46% 11.25%. and 10.30% for training, validation and test sets. The three datasets have -0.224, -0.210 and -0.0209 cross-entropy errors respectly.

The visualization of W is shown as follows:

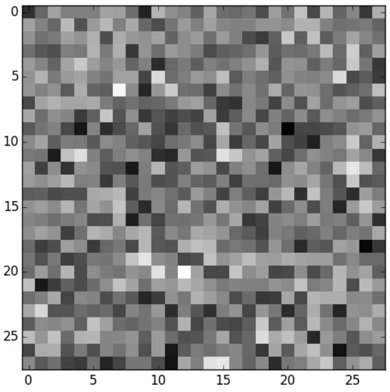


12 Problem 5.h) Extension to multiple layers

In this two-layer neural network, I set the learning rate to be 0.07, momentum 0.3, with 100 hidden units and 28 epoches. In general, 2-layer neural network took longer to pass through each epoch, however, it converges faster with less number of epoches. According to the cross-validation result, the validation error was 26.17% with a cross-entropy error of -0.171.

The classification errors for the training , validation and test data sets are 1.64%, 9.67%, and 8.12%. While the cross-entropy errors are -0.255, -0.223, and -0.0229.

Visualizaion of the 1-layer W:



The first layer filters in a 2-layer network tend to be less sparse than the filters in a 1-layer network.

Accourding to the result I got, there seems to be no significant difference between the 1-layer and the 2-layer network in terms of generalization capabilities.