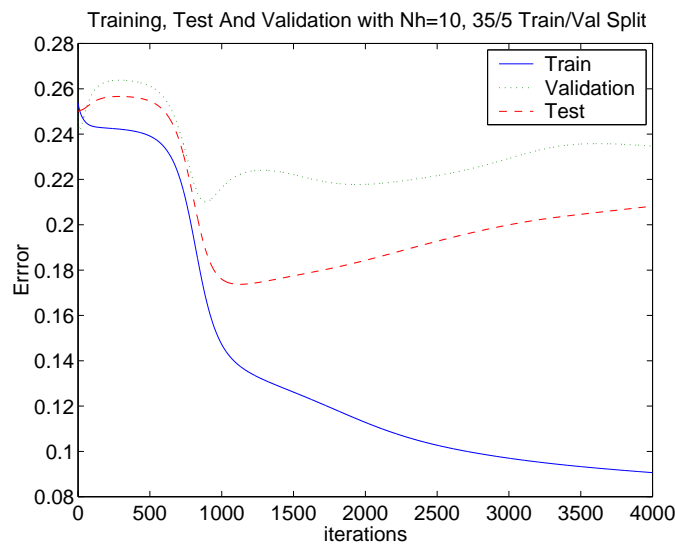


## ASSIGNMENT 5, Solutions

### 1 (300pts) Early Stopping with a Validation Set

(a)



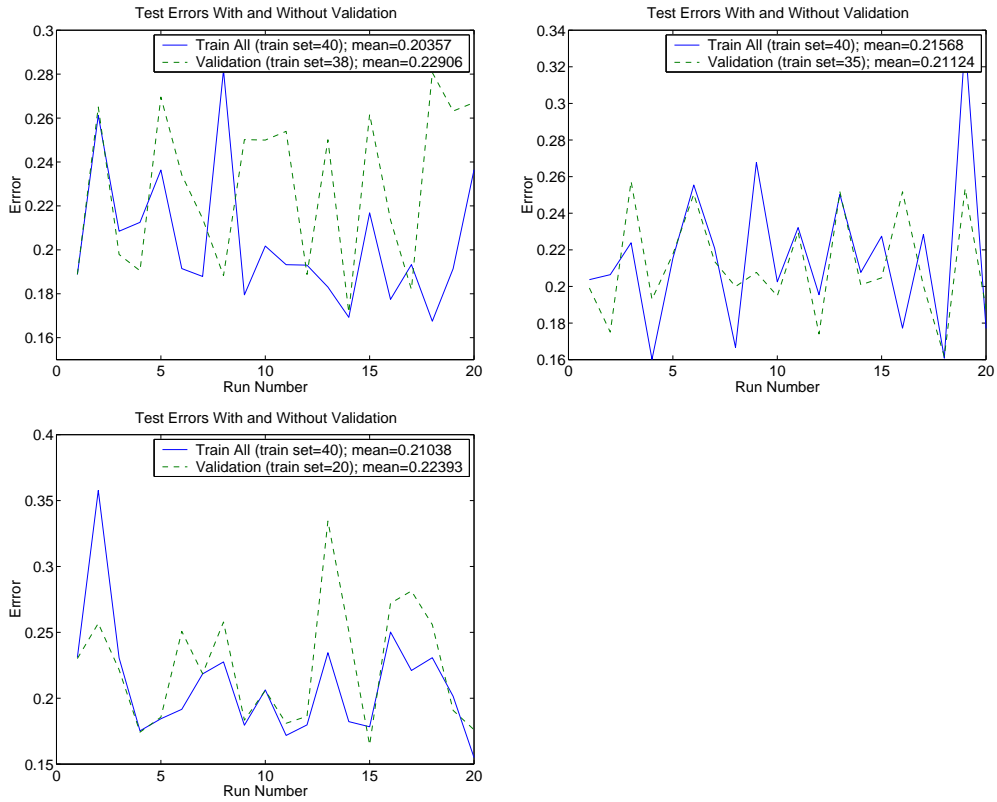
(b) We show the test errors in part (c)

- i. The performance can vary depending on the data set and the initial weights. To make sure that the difference between performance is due only to the use of validation, these other variables (such as data set and initial weights) must be kept constant.
- ii. Each test error measures the performance for the given learning scenario when learning from some given data set and starting from some given initial weight, training for 4000 iterations with  $\eta = 0.2$ .

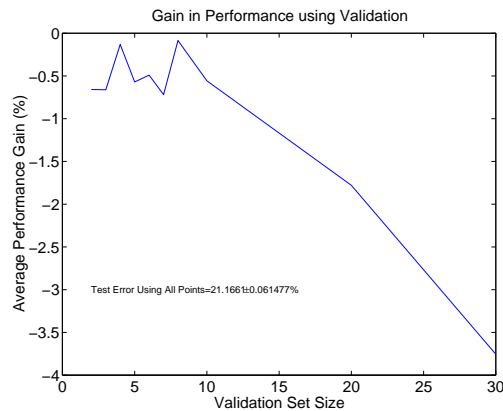
(c)

- i. Each average represents an estimate of the performance of the given learning method averaged over possible (data set, initial weight) pairs, where each method uses the same (data set, initial weight) pair.
- ii. Since performance can vary from data set to data set, and one method can be better with one data set (initial weight) but worse with another data set (initial weight), we report how each method fares on the average, to get a better estimate of the expected performance of the system. However, in real life, one has one specific data set, so one might rather have the performance comparison based on that specific data set. But, we do not know which data set we will see, so we can only compare methods based on average performance.

In the following figures we have plotted 20 cases with 2,5 and 20 validation set sizes (although only the plot for validation set size 5 was asked for).



- (d) For a very small validation set, picking based on the validation set can be misleading. For very large validation sets, there is not enough data to train on (one does not get to good enough hypotheses). Thus we expect performance to be worst at very small and very large validation set sizes, and the best one we expect to occur somewhere in between. This behavior is illustrated in the following figure. The optimal validation set size appears to be between 4-10. Though individual results may vary, it appears that validation is not a good thing for this learning problem.



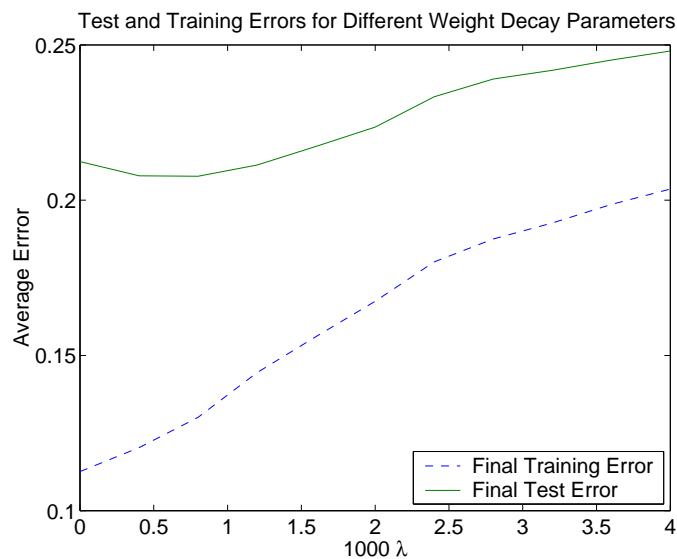
- (e) Use cross validation. This will determine the best validation set with a data set of size  $N - 1$ , and then one uses that optimal size with the data set of size  $N$ .

## 2 (300pts) Weight Decay

(a)

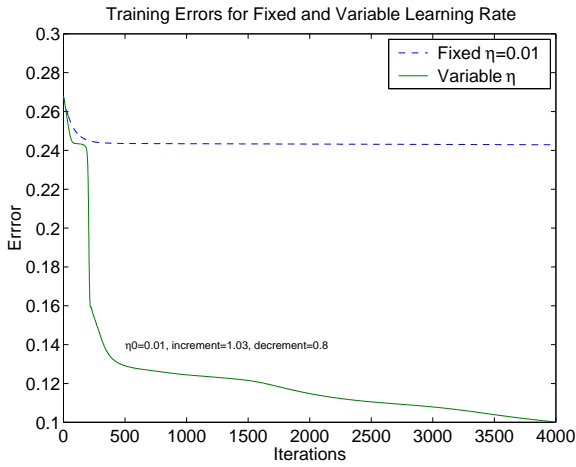
- i. Once again for the same reason as in the previous question. To make sure that any observed performance difference is due only to the different weight decay parameter, everything else must be kept constant.
- ii. Each error represents the test performance using the specific data set, and initial weights for the given weight decay parameter.

(b)

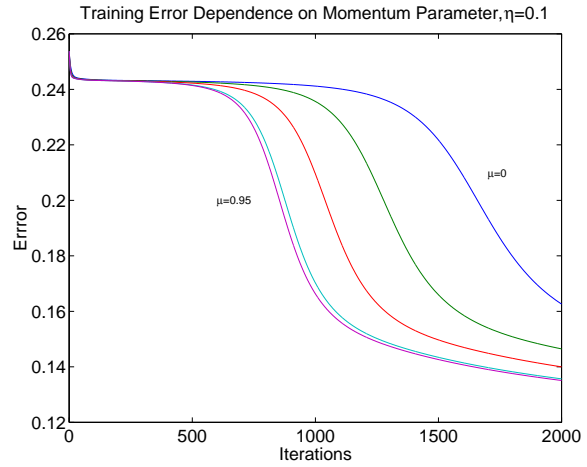


- i. For the same reason as in question (1).
  - ii. An estimate of the average performance to expect if one trains for 4000 iterations with  $\eta = 0.1$  for the specific weight decay parameter, where the average is over data sets and initial weight vectors.
  - iii. The larger  $\lambda$  the smaller the weights will be. i.e., larger  $\lambda$  (softly) “restricts” one to a smaller learning model (smaller weights). Thus, with smaller learning models, the training error is expected to increase. The generalization error is expected to decrease, so the test error (the sum of training and generalization errors) is expected to display some kind of dip where there is some optimal  $\lambda$  (learning model size). Thus as  $\lambda$  increases, we expect to see the training error rise and the test error display the usual down then up behavior as we are accustomed to seeing in the approximation-generalization tradeoff.
- (c) Once again, cross validation. This will determine the optimal  $\lambda$  to use with a set of size  $N - 1$  which we then use for the full data set.

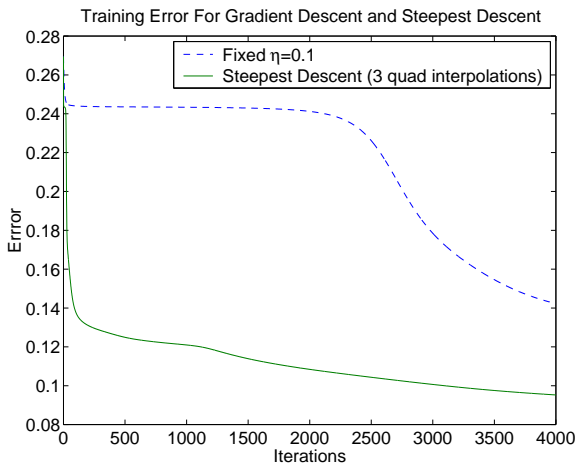
### 3 (400pts) Variable Learning Rate, Momentum, Steepest Descent



(a)

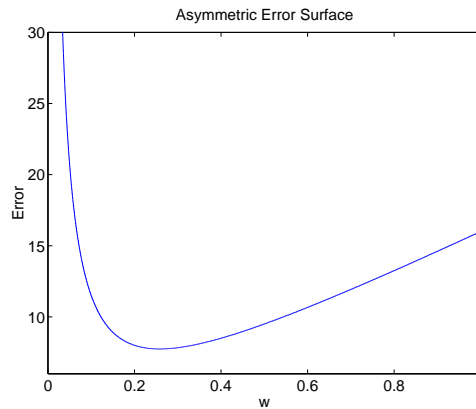


(b)



(c)

(d) For the same reason as in the previous 2 questions. The performance of the methods could drastically depend on the the initial weight. Consider the following error surface.



If the initial weight were on the left, one might approach the minimum much more quickly than if it were on the right. Thus it could be possible for a worse technique to beat a better one depending on where the initial weight is. We thus want to keep the initial weight the same.

## 4 (150pts Extra Credit) Bagging Using Bootstrap

(a) Perform the following experiment.

1. Select a data set of size 15. Select an initial weight  $\mathbf{w}(0)$ .
2. Create 10 bootstrapped sets from this data set.
3. For each of these 11 data sets, start from the *same* initial weight and train using the training algorithm of choice.
4. After training is done, you have 11 functions.  $g$ , trained on the full data set, and  $g_i$  trained on the 10 bootstrapped sets. Let  $g_{bag} = \frac{1}{10} \sum g_i$ . Now compute the test error for  $g$  and  $g_{bag}$ . These test errors represent the performance of bagging vs. no bagging for the particular data set and initial weights chosen.
5. Repeat this expt (go back to (1)) some number of times, say 100.
6. Obtain the average performance gain due to bagging. This represents that average performance gain to expect from bagging for this learning problem, averaged over initial weights and data sets.

It is important that each  $g_i$  is trained from the same initial weights, as the choice of initial weight could drastically affect the performance for a fixed data set! If one allowed the bagged functions to also have different weights, then one is combining the bagging technique (which generates new functions by generating different data sets through bootstrap) with the more general committees technique. In this case actually, one will expect even better performance.

(b) We show the performance gain due to bagging for 100 runs. The mean performance gain is also shown. We see that bagging gains almost 5.5% in test error on average! Thus, bagging would be a good idea for this learning problem.

