(a) Provide the learning rates you used (they can be different for each number of hidden units).

We used 1.0, 0.1, 0.01, and 0.001 as the learning rates for both 15 and 30 hidden units. We chose these rates in order to find the order of magnitude of the most effective learning rate.

For 30 hidden units, we found the optimal learning rate was on the order of 0.1.

(b) In the previous problem, we determined the number of epochs to use based on

analyzing the graph of error against the number of epochs. Devise an automatic way (something you can code up) that determines when to stop training and describe it. Implement your idea. (Hint: use the validation set. Note: your stopping condition should also take a maximum epochs to train for so that we don’t train forever).

We monitor the validation set performance of the neural network. Our stopping condition is when the validation set performance decreases twice in a row. Empirically we found that a one-time decrease in validation was not enough to guarantee a good stopping point and a three-in-a-row decrease in validation performance rarely occurred in general.

(c) For both 15 and 30 hidden units, use your algorithm to determine when to stop. Graph training set and validation set error against the number of epochs you trained for.

GRAPHS

(d) How many epochs did you use for 15 hidden units? For 30 hidden units?

For 30 hidden units with learning rate 1.0, we stopped after 6 epochs (2 consecutive decreases after). With learning rate 0.01, we stopped after 83 epochs. With learning rate 0.001, we stopped after 83 epochs. With the optimal learning rate 0.1, we stopped after 79 epochs.

(e) Which network structure (15 or 30 hidden units) would you choose based on these experiments? Justify your answer.

Using optimal learning rate 0.1 and stopping after 79 epochs, we get performance of 0.961.

(f) What is the test set performance of the network you chose? How does this compare to the committee of perceptrons?