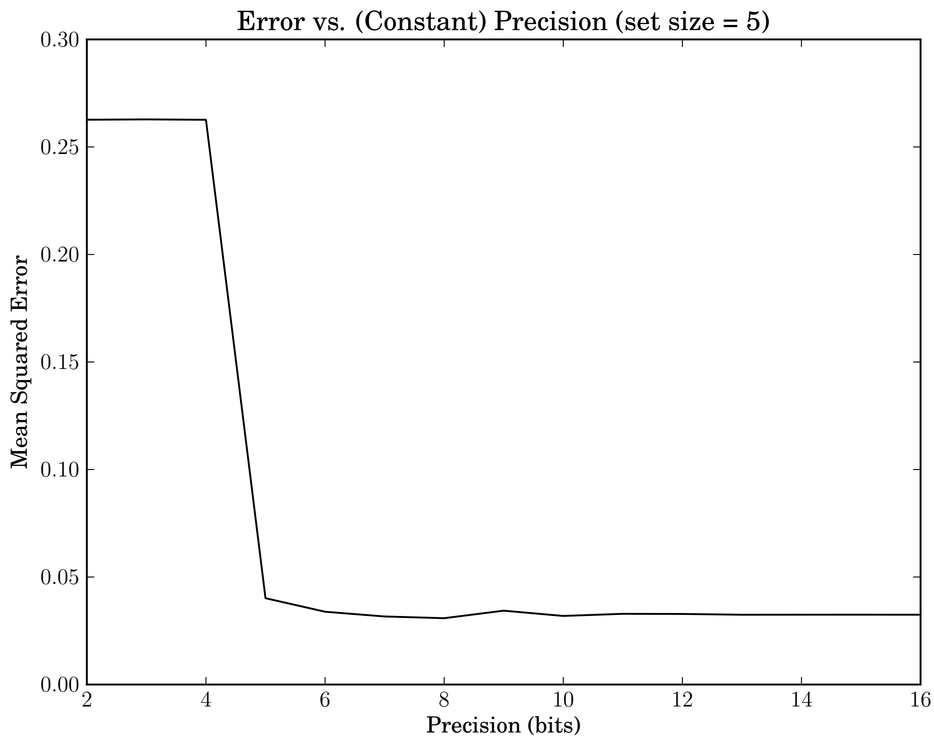
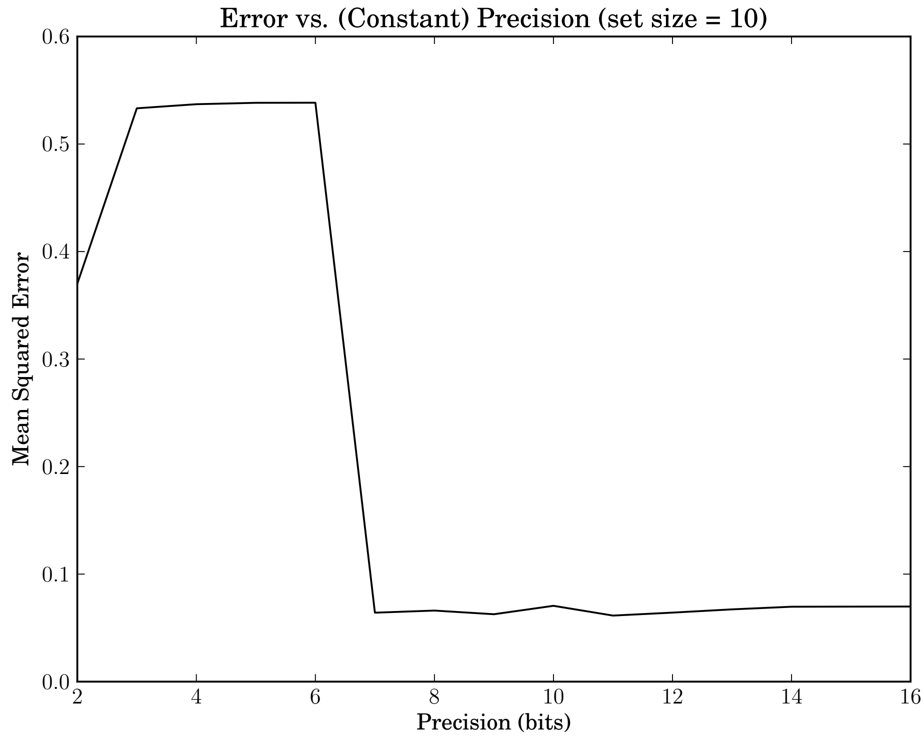
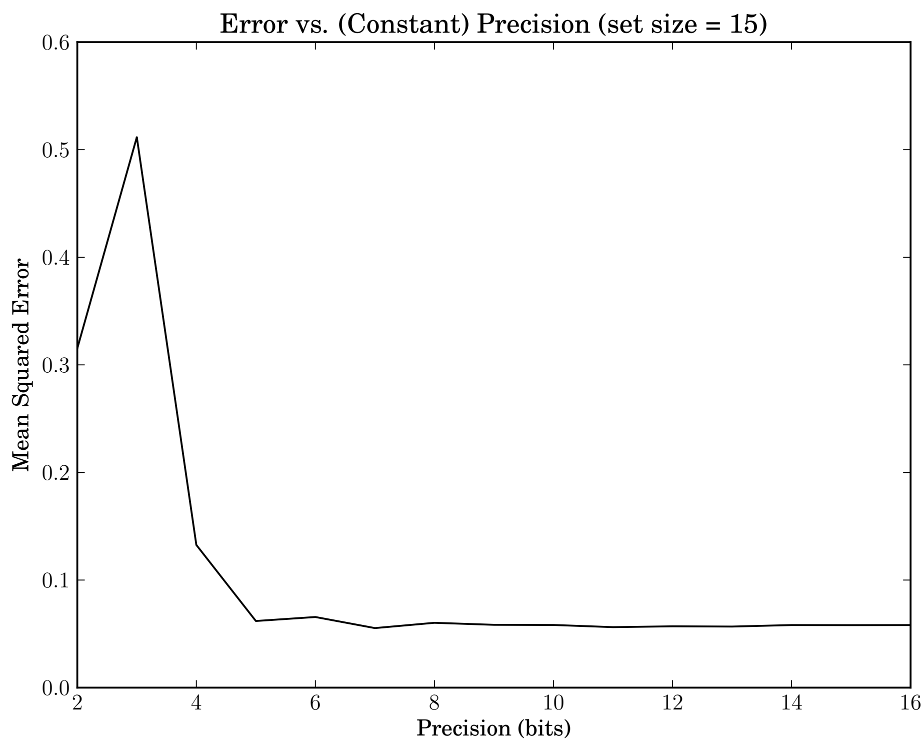
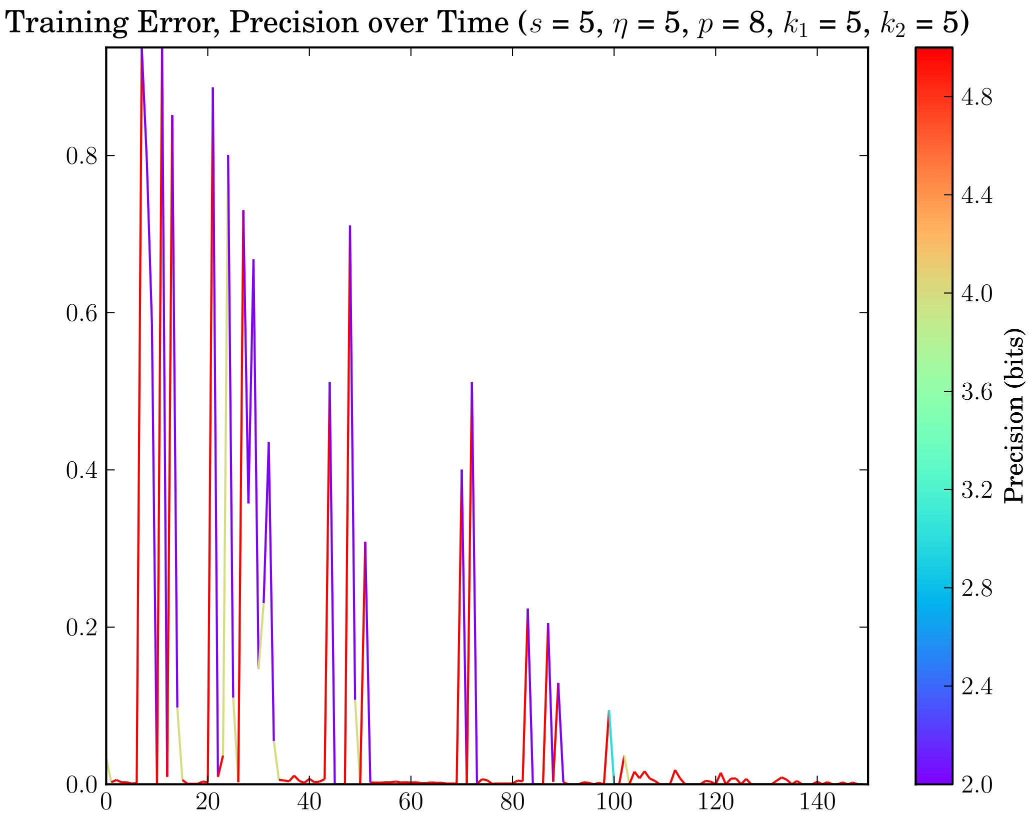
Mean squared testing error averaged over all testing data versus (constant) precision (in bits) of the network. Error is on a log scale. For a given set size s, network is trained on 2^{s-3} of the 2^s possible inputs (selected at random); tested on the remaining inputs. Learning rate $\eta = 4.0$, 400 iterations over training data. The first bit of each input represents the distance (bit in the string) that should be attended to (all 0 in these experiments). Hidden layer contains 20 neurons.

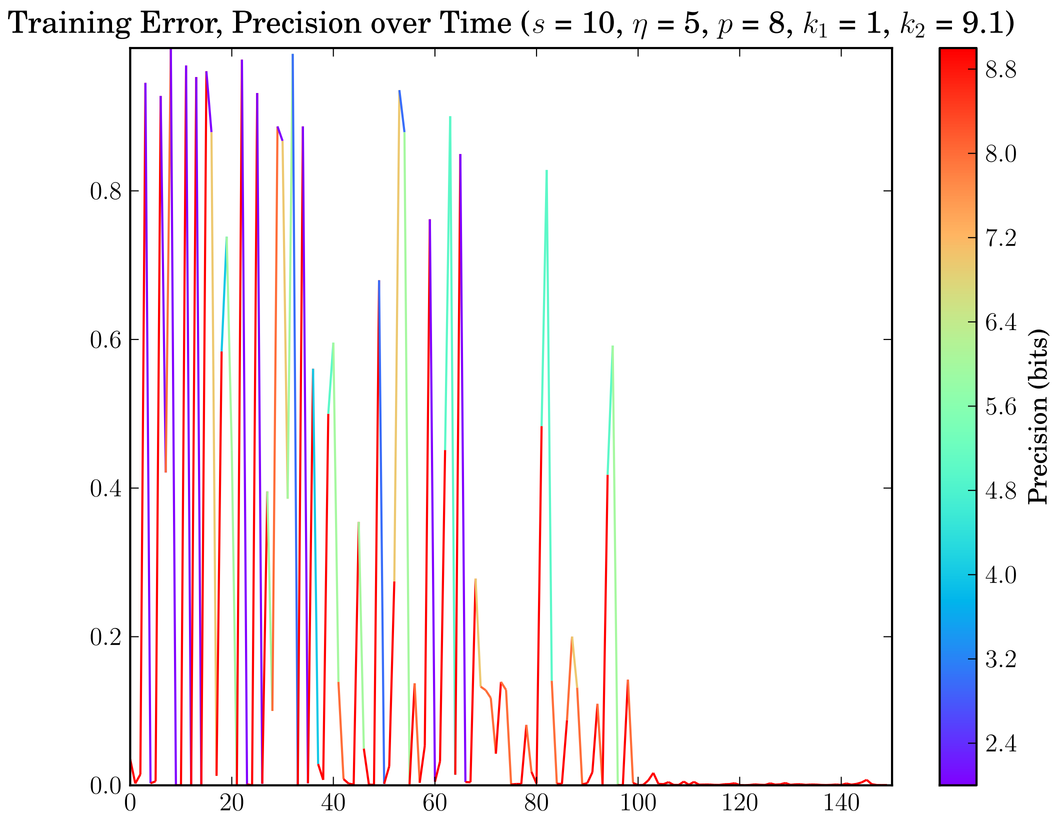


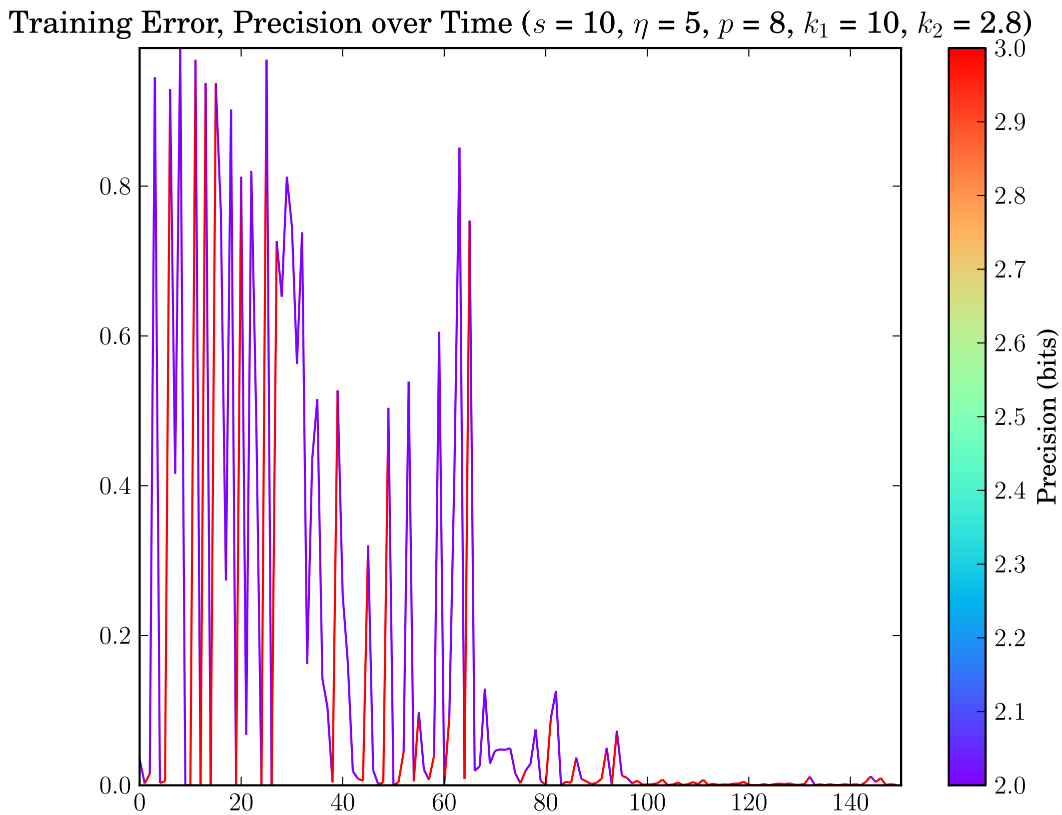

Error vs. (Constant) Precision (set size = 10)
Mean Squared Error
Precision (bits)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2
4
6
8
10
12
14
16


Error vs. (Constant) Precision (set size = 15)
Mean Squared Error
Precision (bits)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2
4
6
8
10
12
14
16

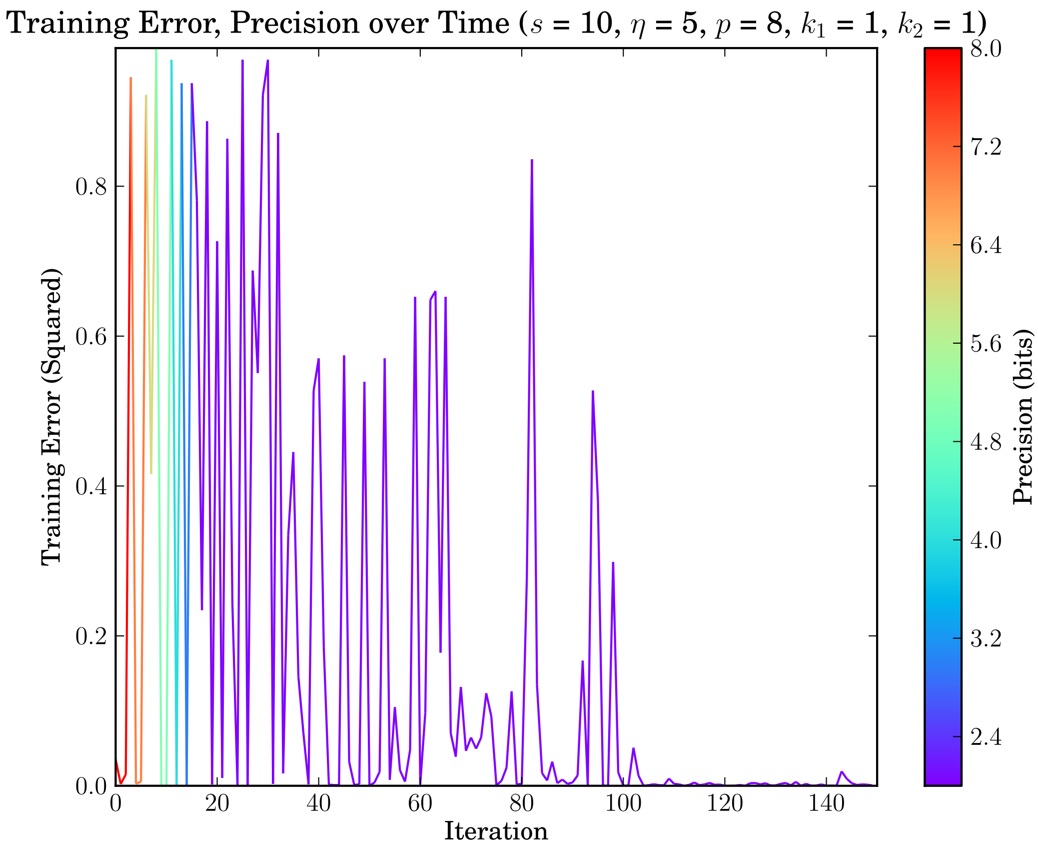
Squared training error over time (iterations, where each iteration trains on all input samples), where the color of the plot corresponds to precision; hotter colors are higher precision. Initial precision $p = 8$, learning rate $\eta = 5.0$, 150 iterations performed over training data. Precision is changed as follows: $p^+ = k_1 dep + k_2$, k_1, k_2 are constants, d is the direction in which we want precision to change (negative if accuracy, taken as the average over the last three iterations, is above the desired amount, positive otherwise), e is the training error from the last iteration, and p is the current network precision. Plots for a few combinations of k_1, k_2 shown.

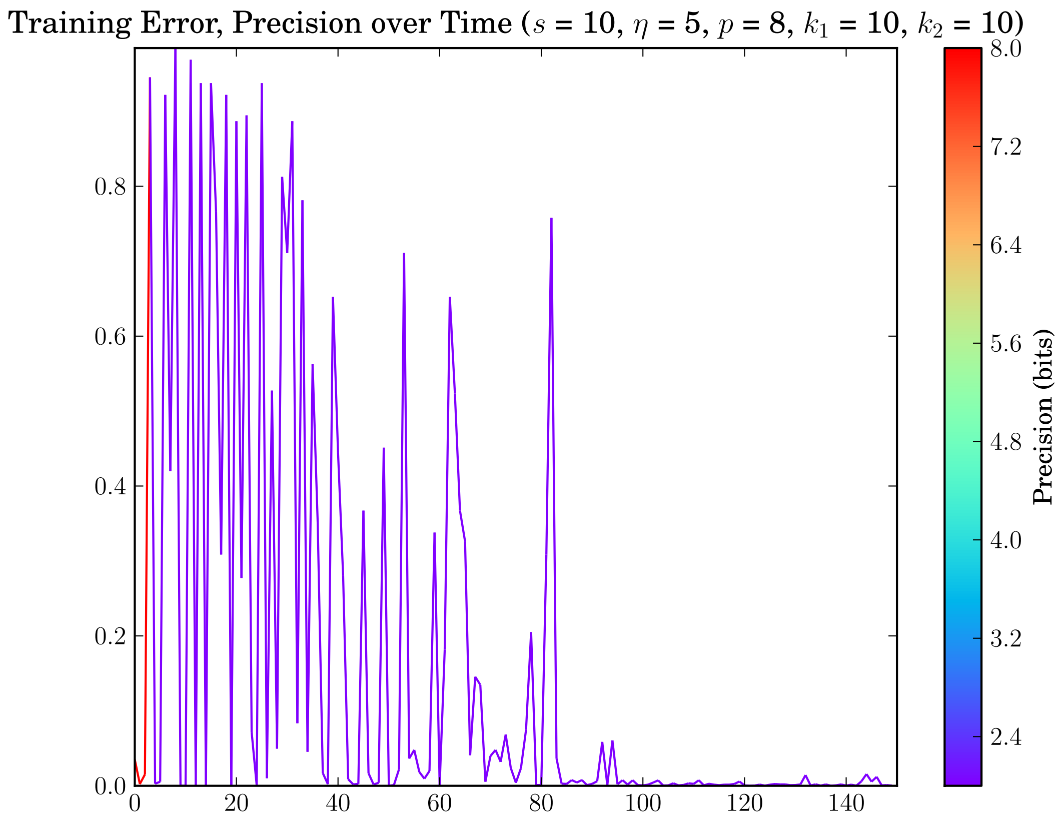



Training Error, Precision over Time (s = 10, η = 5, p = 8, k_1 = 1, k_2 = 9.1)
0.0
0.2
0.4
0.6
0.8
0
20
40
60
80
100
120
140
8.8
8.0
7.2
6.4
5.6
4.8
4.0
3.2
2.4
Precision (bits)



Squared training error over time (iterations, where each iteration trains on all input samples), where the color of the plot corresponds to precision; hotter colors are higher precision. Initial precision $p = 8$, learning rate $\eta = 5.0$, 150 iterations performed over training data. Precision is changed as follows: $p^{+} = p + k_1 de$, k_1 a constant, d is the direction in which we want precision to change (negative if accuracy, taken as the average over the last three iterations, is above the desired amount, positive otherwise), e is the training error from the last iteration, and p is the current network precision. Plots for a few values of k_1 shown.


Training Error, Precision over Time (s = 10, η = 5, p = 8, k_1 = 1, k_2 = 1)
Training Error (Squared)
Iteration
Precision (bits)
0.0
0.2
0.4
0.6
0.8
0
20
40
60
80
100
120
140
8.0
7.2
6.4
5.6
4.8
4.0
3.2
2.4


Training Error, Precision over Time (s = 10, η = 5, p = 8, k_1 = 10, k_2 = 10)
0.0
0.2
0.4
0.6
0.8
0
20
40
60
80
100
120
140
8.0
7.2
6.4
5.6
4.8
4.0
3.2
2.4
Precision (bits)