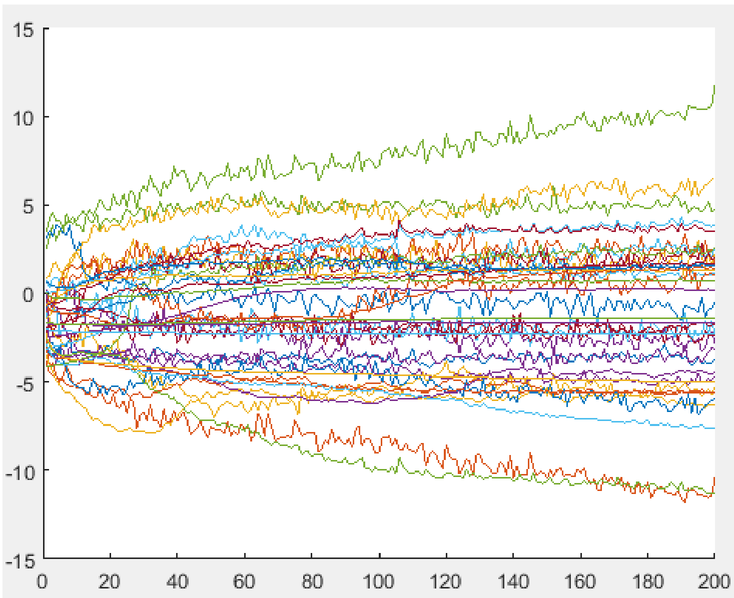
Alex Elkman

ECS 171 FQ 2016

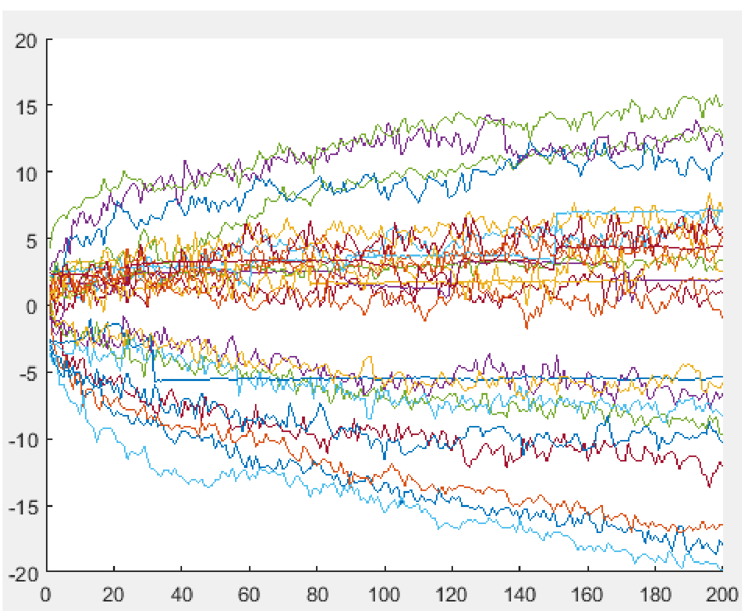
10/25/2016

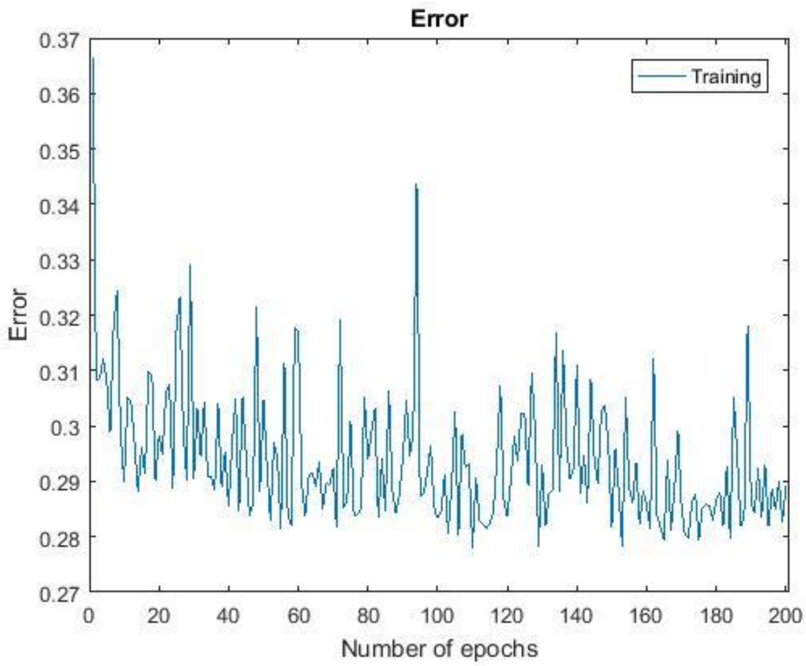
HW2

1) Weights from inputs

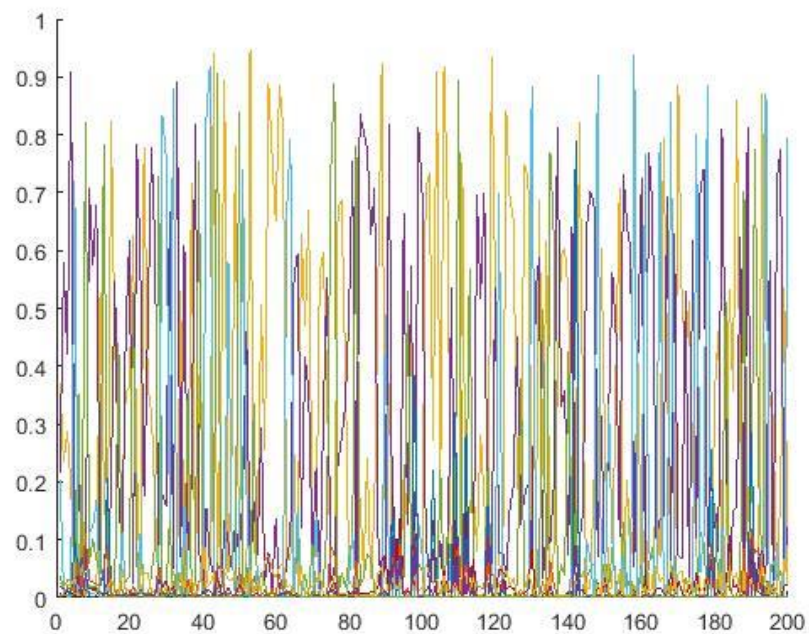


Weights from hidden layer




Error
Training
Error
0.37
0.36
0.35
0.34
0.33
0.32
0.31
0.3
0.29
0.28
0.27
0
20
40
60
80
100
120
140
160
180
200
Number of epochs

Output


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
120
140
160
180
200

2)

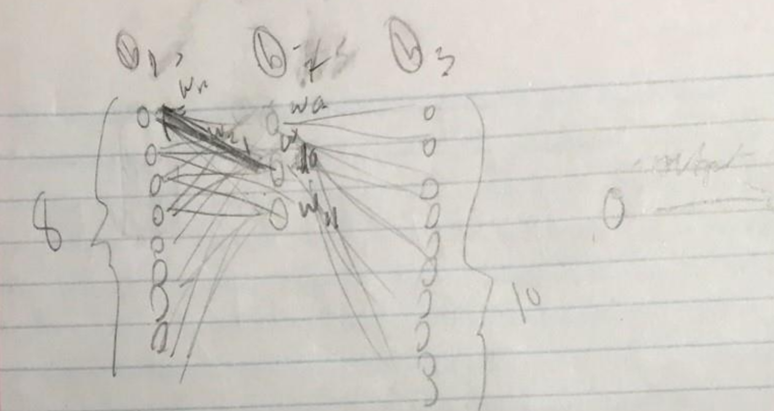
As compared to the error from Problem 1 (= 0.304070) we can see the results were better with a larger training sample. This makes sense since it would more accurately predict behavior with more results:

epoch 200/200. Took 0.070497 seconds. Mini-batch mean squared error on training set is 0.29283; Full-batch train err = 0.289321

Activation Function = g(0.0440) + g(0.1228) + g(0.0028) + g(0.7970) + g(0.0219) + g(.0005293) + g(0.0033) + g(0.223) + g(0.0040) + g(.00691)

Where g(z) is the sigmoid function of activation.

3)



$+b_1$

$net\,h_1 = w_{11} \cdot i_1 + w_{21} \cdot i_2 + w_{31} \cdot i_3 + w_{41} i_4 + w_{51} i_5 + w_{61} i_6 + w_{71} i_7$

$= -0.0828 \cdot 2.1138 + (-0.2572)(1.6917) + (-0.3267)(-1.7374) +$

$+b_1$

$net\,h_2 = w_{12} i_2 + w_{22} i_2 + w_{32} i_3 + w_{42} i_4 + w_{52} i_5 + w_{62} i_6 + w_{72} i_7 + w_{82} i_8$

$+b_1$

$net\,h_3 = w_{13} i_3 + w_{23} i_3 + w_{33} i_3 + w_{43} i_4 + w_{53} i_5 + w_{63} i_6 + w_{73} i_7 + w_{83} i_8$

$g(h_1) = out_{h_1}$

$g(h_2) = out_{h_2}$

$g(h_3) = out_{h_3}$

Note: w_9, w_{10}, w_{11} are the weights for hidden layer

$net_{o_1} = out_{h_1} \cdot w_9 + out_{h_2} \cdot w_{10} + out_{h_3} w_{11} + b_3$

$net_{o_2} = out_{h_1} w_{12} + out_{h_2} w_{13} + out_{h_3} w_{14} + b_4$

$net_{o_3} = \ldots$

$\vdots$

$net_{o_{10}} = \ldots + b_5$

Final output [0.113 0.752 0.0041 0.427 0.0123 0.513 0.0123 0.123]

4)

In the below matrix, the first row represents the [8 3i 10] test where i = 1:4. Second row is [8 3i 3i 10], and third row is [8 3i 3i 3i 10]. We can see that this error is reflective of the final error of the neural networks with 200 epoch propagations. Since higher error is bad, we can see that the optimal config would be using the least layers possible, since the average error is lowest in the first row. Using more layers may be better for more complex problems, but for a data set like this, it just overcomplicates things.

0.3041	0.2690	0.2533	0.2597
0.2936	0.2679	0.3008	0.2337
0.3133	0.2918	0.2608	0.2444

5)

```
>> nn.a{3}

ans =

   0.0440    0.1228    0.0028    0.7970    0.0219    0.0006    0.0033    0.0223    0.0040    0.0007
```

From the testing results we can see that the best fit is the 4th column which is ME3. \

6)

Uncertainty can be directly correlated to the error. The higher the expected error of a given result, the less we can accurately predict that it will be classified correctly.

We could go so far as to say that Probability of Correct Classification, PCC = 1-[ACTIVE]nn.a{3}

Where the active member of nn.a{3} is the highest probability output. Subtracting 1 from this will give us the probability of success in prediction.