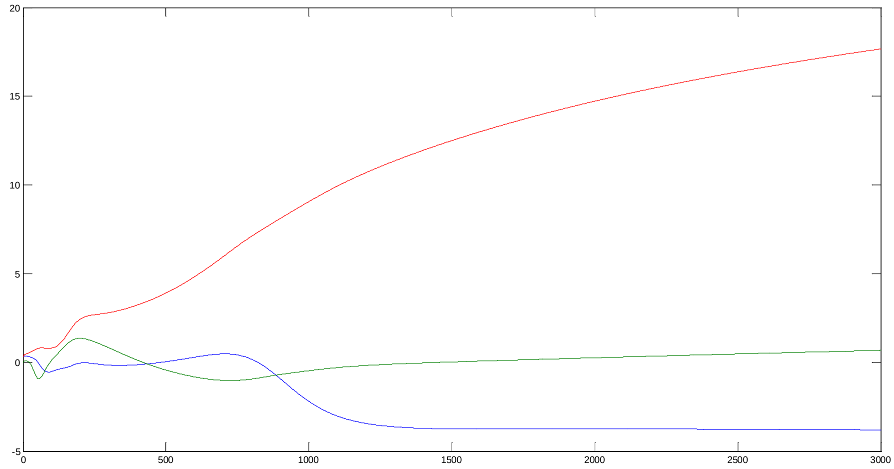
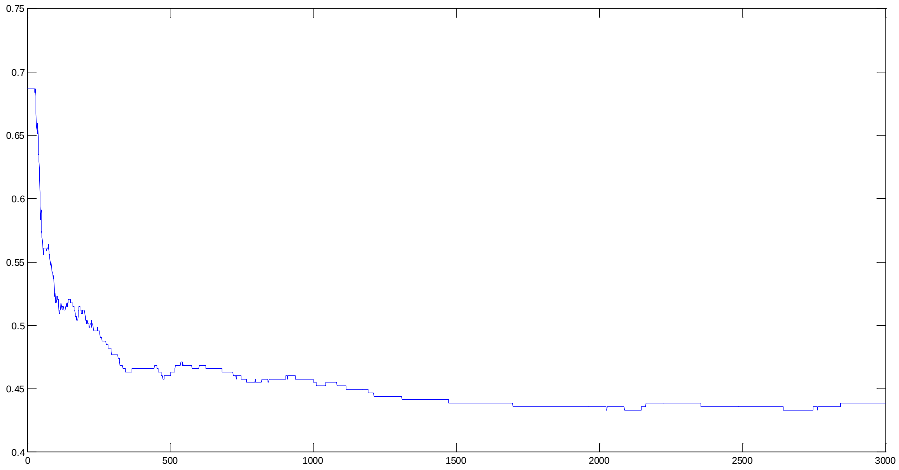
Jiahuan He
ID: 91249074

Reading the data into Matlab:
[sequenceName, mcg, gvh, alm, mit, erl, pox, vac, nuc, class] = textread('yeast.txt', '%s%n%n%n%n%n%n%n%n%s');

Q1.
In building the neural network, we use the sigmoid function as our activation function (See the network training function in M file 'sigmoid.m'). And the neural network is with 3 layers, 8 input, 3 hidden nodes and 10 output. Thus we have 9 x 3 + 4 x 10 = 67 weights to estimate.
The following plot shows the changes of weights from input 1-'mcg' to three hidden nodes at every iteration.

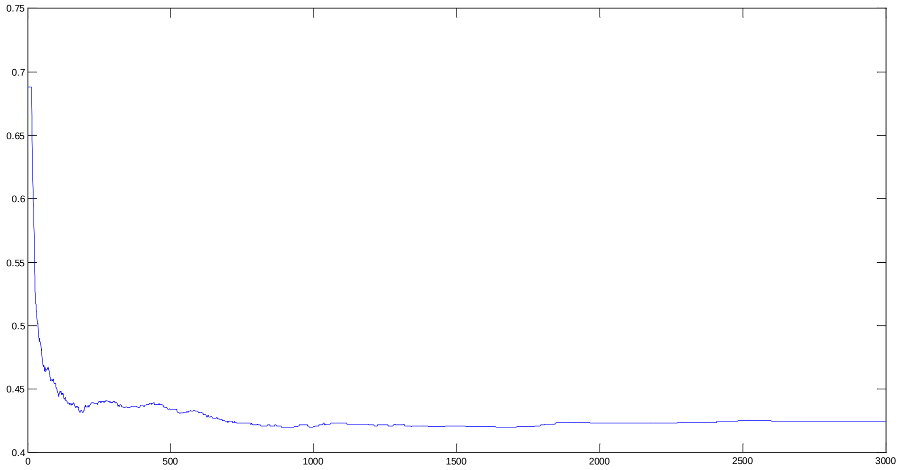


The following picture shows the misclassification rate of the testing set with 3000 loops. After about 1500 iterations, the rate converge to approximately 0.44. In other words, 56% of the testing set samples are correctly classified.



Q2.

The following picture is the misclassification rate for all 1484 samples with 3000 iterations.



The following 9 by 3 matrix is the weights from input to hidden layer:

-4.22	-14.43	14.71
7.25	-6.09	-7.13
2.01	-4.27	-13.32
5.24	36.49	3.29
-0.24	8.83	-17.04
0.09	-2.10	5.23
-0.36	-0.04	-4.25
4.05	1.30	-3.10
-20.51	7.67	2.06

The following 4 by 10 matrix is the weights from hidden layer to output:

-6.5	-4.2	-14.0	1.9	-5.1	1.0	-11.0	-4.9	-4.2	-3.9
2.4	-2.6	1.3	-5.5	4.6	-0.3	10.5	1.1	0.4	0.5
2.2	3.2	14.6	-7.2	-4.6	-5.9	0.6	-1.4	0.3	-1.0
3.2	1.7	-3.9	2.9	0.7	-6.6	-3.8	1.7	-0.7	-1.9

Q3.

For the first sample, the feature vector show as follow.

$$S_1 = \begin{pmatrix} 1 & 0.58 & 0.61 & 0.47 & 0.13 & 0.50 & 0 & 0.48 & 0.22 \end{pmatrix}$$

The initial weights matrix W_{ith}, which is between input and hidden layer is a matrix. To be simple, we set all its elements to be zero, thus

$$S_1 \times W_{ith} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

We put every element into the sigmoid function. Similarly, the initial matrix W_{hto}, which is between hidden layer and output is also a matrix with all 0 elements,

$$(1 \quad 0.5 \quad 0.5 \quad 0.5) \times W_{hto} = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)$$

And the output is $(0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5)$

The class of the first sample is MIT, thus its Target vector is

$$(0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)$$

Set the learning rate to be 0.1, and we have,

$\delta^{output} = (-0.125 \quad -0.125 \quad 0.125 \quad -0.125 \quad -0.125 \quad -0.125 \quad -0.125 \quad -0.125 \quad -0.125 \quad -0.125)$

Since W_{hto} is still a zero matrix, thus

$\delta^{hidden} = (0 \quad 0 \quad 0 \quad 0)$

Then, we update our matrix, which gives us exactly the same thing as the computer does (for the computer output, see file 'hw2.m').

$$W_{hto} = W_{hto} + 0.1 \times \begin{pmatrix} 1 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} (-0.125 \quad -0.125 \quad 0.125 \quad -0.125 \quad -0.125 \quad -0.125 \quad -0.125 \quad -0.125 \quad -0.125 \quad -0.125)$$

-0.0125	-0.0125	0.0125	-0.0125	-0.0125	-0.0125	-0.0125	-0.0125	-0.0125	-0.0125
-0.0063	-0.0063	0.0063	-0.0063	-0.0063	-0.0063	-0.0063	-0.0063	-0.0063	-0.0063
-0.0063	-0.0063	0.0063	-0.0063	-0.0063	-0.0063	-0.0063	-0.0063	-0.0063	-0.0063
-0.0063	-0.0063	0.0063	-0.0063	-0.0063	-0.0063	-0.0063	-0.0063	-0.0063	-0.0063

$W_{ith} = W_{ith} + 0.1 \times S_1^T (0 \quad 0 \quad 0)$

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Q4.

MSE	*1 layer*	*2 layers*	*3 layers*
3 nodes	***0.001384***	***0.001421***	***0.001368***
5 nodes	***0.001344***	***0.001341***	***0.001335***
8 nodes	***0.001340***	***0.001310***	***0.001309***
10 nodes	***0.001321***	***0.001313***	***0.001323***

The above chart shows the performance of neural networks with different number of nodes and layers in terms of MSE. However it is not guaranteed, adding node and layer to neural networks will probably improve the performance of the model.

Q5.

Based on the output of the neural network, this sample will be classified as 'CYT'.

Q6.

We can select 75% (or 60%, 80% etc.) of the sample with identical distribution of the whole sample. Then use the subset to train the model and predict the new input. We do this iteratively for n times, and the mode of the n results as the final prediction 'Class', assume 'Class' appears m times in all n iterations, then we define the variance of new input as

$$n \times \frac{m}{n}(1-\frac{m}{n})$$