4SL3 Lab #3

David Wang 400073796

**Setting of the Classifiers**

The number of epochs used in this code was 10 with a 0.005 learning rate.

Initial weights used: matrices filled with value 0.5

Previously experimented weights with higher errors:

1.matrices filled with value of 20.

2.Matrices gradually ascending by 0.1 for each value inside the row starting at 0.

**Validation Misclassifications Errors:**

|  |  |  |  |
| --- | --- | --- | --- |
| D2 | | | |
|  | hidden layer 2 | | |
| hidden layer 1 | 2 | 3 | 4 |
| 2 | 0.131195 | 0.131195 | 0.131195 |
| 3 | 0.131195 | 0.131195 | 0.131195 |
| 4 | 0.131195 | 0.131195 | 0.131195 |

|  |  |  |  |
| --- | --- | --- | --- |
| D3 | | | |
|  | hidden layer 2 | | |
| hidden layer 1 | 2 | 3 | 4 |
| 2 | 0.020408 | 0.020408 | 0.020408 |
| 3 | 0.020408 | 0.020408 | 0.020408 |
| 4 | 0.020408 | 0.020408 | 0.020408 |

|  |  |  |  |
| --- | --- | --- | --- |
| D4 | | | |
|  | hidden layer 2 | | |
| hidden layer 1 | 2 | 3 | 4 |
| 2 | 0.029155 | 0.029155 | 0.029155 |
| 3 | 0.020408 | 0.020408 | 0.020408 |
| 4 | 0.020408 | 0.020408 | 0.020408 |

Best classifier is D3 with 2 nodes in hidden layer 1, and 2 nodes in hidden layer 2. With a validation error of around 2%. Its training misclassification error is **0.0145.** D2 error might be larger because there are not enough features to make accurate correlations, while feature 4 might be redundant in D4 analysis.

**Best weights for D2: n1=2, n2=2**

weight1: [[0.66896921 0.99807091 0.45271596]

[0.66896921 0.99807091 0.45271596]]

weight2 [[-0.00130997 0.82907622 0.82907622]

[-0.00130997 0.82907622 0.82907622]]

weight3 [[ 1.94549962 -0.93773678 -0.93773678]]

**Best weights for D3: n1=2, n2=2**

weight1: [[0.65953573 0.91598134 0.94011931 0.86151416]

[0.65953573 0.91598134 0.94011931 0.86151416]]

weight2 [[-0.0168144 1.08907468 1.08907468]

[-0.0168144 1.08907468 1.08907468]]

weight3 [[ 2.35861802 -1.36923188 -1.36923188]]

**Best weights for D4: n1=3, n2=3**

weight1: [[ 0.59592324 0.81818723 0.85284771 0.7812607 -0.01439123]

[ 0.59592324 0.81818723 0.85284771 0.7812607 -0.01439123]

[ 0.59592324 0.81818723 0.85284771 0.7812607 -0.01439123]]

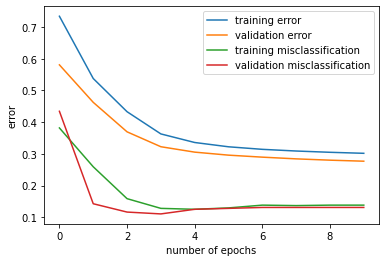
weight2 [[-0.01629093 0.8923055 0.8923055 0.8923055 ]

[-0.01629093 0.8923055 0.8923055 0.8923055 ]]

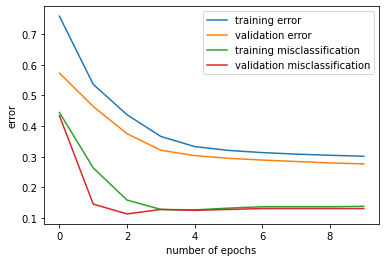
weight3 [[ 2.42446514 -1.28278556 -1.28278556]]

**D2 Graphs**

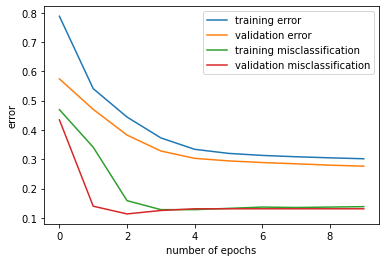
Hidden layer 1: 2 nodes, hidden layer 2: 2 nodes



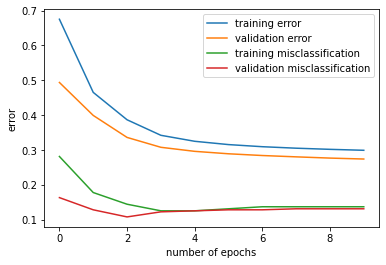
Hidden layer 1: 2 nodes, hidden layer 2: 3 nodes



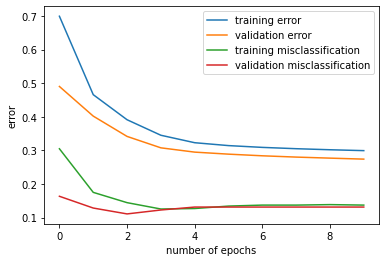
Hidden layer 1: 2 nodes, hidden layer 2: 4 nodes



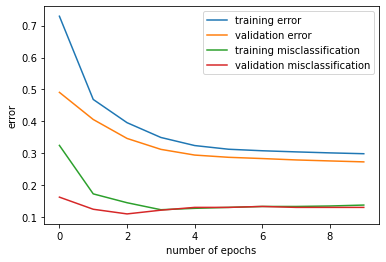
Hidden layer 1: 3 nodes, hidden layer 2: 2 nodes



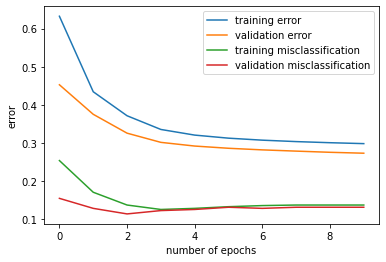
Hidden layer 1: 3 nodes, hidden layer 2: 3 nodes



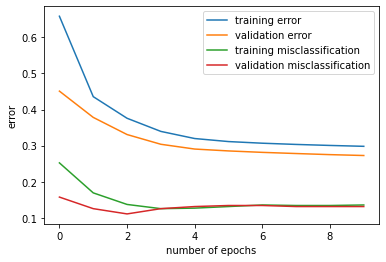
Hidden layer 1: 3 nodes, hidden layer 2: 4 nodes



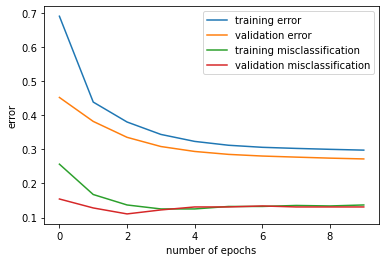
Hidden layer 1: 4 nodes, hidden layer 2: 2 nodes



Hidden layer 1: 4 nodes, hidden layer 2: 3 nodes



Hidden layer 1: 4 nodes, hidden layer 2: 4 nodes



As more nodes are used the training and validation error goes down decently at the first epoch. Interestingly enough the validation misclassification rate seems to always be the lowest at epoch 2 while the training misclassification gradually lowers until epoch 3 and starts to even out. N1 seems to also have a direct correlation with the validation error as it starts at around 0.6 at n1=2, 0.5 when n1=3 and 0.45 when n1=4. This means that the first hidden layer has much more effect on the graph and system than the second hidden layer. The training error seems to also be the highest when n1=2 around the high 0.7’s however after n1 passes the 3 hidden layer threshold it seems like it typically averages out around 0.7. It is curious to see the training and validation misclassification rates always seems to converge on top of each other, however the training error always seems to be slightly larger than the validation error at all times. This is probably because the validation error only registers at the end of every epoch. So it uses the best weights for all the iterations.