|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Minimum Error | # of Epochs | Learning Rate | # of Hidden Layer Nodes | Error @ output node1 after first epoch | Error @ output node2 after first epoch | Error @ output node1 after last epoch | Error @ output node2 after last epoch | Metrics |
| 0.1 | 500 | 0.9 | 1 | -0.0304 | 0.0312 | -0.0252 | 0.02529 | Accuracy: 0.6470  Macro F1: 0.3928  Micro F1: 0.6470 |
| 0.1 | 500 | 0.9 | 1 | -0.0330 | 0.0327 | -0.02835 | 0.0283 | Accuracy: 0.7647  Macro F1: 0.4333  Micro F1: 0.7647 |
| 0.1 | 500 | 0.9 | 1 | -0.0330 | 0.0334 | -0.0285 | 0.0285 | Accuracy: 0.7777  Macro F1: 0.4375  Micro F1: 0.7777 |
| 0.1 | 500 | 0.9 | 1 | -0.0325 | 0.0330 | -0.0282 | 0.0282 | Accuracy: 0.7647  Macro F1: 0.4333  Micro F1: 0.7647 |
| 0.1 | 500 | 0.9 | 1 | -0.0458 | 0.0465 | -0.0421 | 0.0421 | Accuracy: 0.8888  Macro F1: 0.4705  Micro F1: 0.8888 |

Note: In the interest of time, these experiments were done on a reduced dataset, as training times for full size would take a total of an hour per experiment.

Experiment # 1 (5 different folds with 1 hidden layer node, 0.9 learning rate, and 500 epochs)

Experiment #2 (500 epochs, 0.9 learning rate, 10 nodes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Minimum Error | # of Epochs | Learning Rate | # of Hidden Layer Nodes | Error @ output node1 after first epoch | Error @ output node2 after first epoch | Error @ output node1 after last epoch | Error @ output node2 after last epoch | Metrics |
| 0.1 | 60 | 0.9 | 10 | -0.0016 | 0.0063 | -5.330e-08 | 4.525e-08 | Accuracy: 1.0  Macro F1: 1.0  Micro F1: 1.0 |
| 0.1 | 28 | 0.9 | 10 | -0.01357 | 0.0054 | -4.498e-07 | 4.046e-07 | Accuracy: 0.9411  Macro F1: 0.9100  Micro F1: 0.9411 |
| 0.1 | 33 | 0.9 | 10 | -0.0032 | 0.0017 | -7.627e-08 | 6.994e-08 | Accuracy: 0.8333  Macro F1: 0.7339  Micro F1: 0.8333 |
| 0.1 | 48 | 0.9 | 10 | -0.0062 | 0.0093 | -1.125e-08 | 1.738e-08 | Accuracy: 0.9411  Macro F1: 0.9100  Micro F1: 0.9411 |
| 0.1 | 26 | 0.9 | 10 | -0.0146 | 0.0175 | -2.161e-06 | 2.031e-06 | Accuracy: 0.8888  Macro F1: 0.71875  Micro F1: 0.8888 |

Experiment #3 (500 epoch max, 0.9 learning rate, 20 nodes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Minimum Error | # of Epochs | Learning Rate | # of Hidden Layer Nodes | Error @ output node1 after first epoch | Error @ output node2 after first epoch | Error @ output node1 after last epoch | Error @ output node2 after last epoch | Metrics |
| 0.1 | 41 | 0.9 | 20 | -0.0041 | 0.0019 | -6.241e-08 | 4.525e-08 | Accuracy: 1.0  Macro F1: 1.0  Micro F1: 1.0 |
| 0.1 | 26 | 0.9 | 20 | -0.0002 | 0.0002 | -6.090e-09 | 6.243e-09 | Accuracy: 0.9411  Macro F1: 0.9100  Micro F1: 0.9411 |
| 0.1 | 29 | 0.9 | 20 | -0.0004 | 0.0009 | -1.490e-10 | 1.607e-10 | Accuracy: 0.8333  Macro F1: 0.7339  Micro F1: 0.8333 |
| 0.1 | 48 | 0.9 | 20 | -0.0012 | 0.0014 | -5.076e-11 | 7.191e-11 | Accuracy: 0.9411  Macro F1: 0.9100  Micro F1: 0.9411 |
| 0.1 | 27 | 0.9 | 20 | -0.0152 | 0.0106 | -2.268e-06 | 2.287e-06 | Accuracy: 0.8333  Macro F1: 0.6516  Micro F1: 0.8333 |

First evaluation:

After playing around with the number of nodes, we can see a significant increase in the performance of our model as we increase the number of nodes in our hidden layer. However, the difference was most notable from having 1 to having 10 than from having 10 to having 20. The 1-node model was not better than simply classifying everything as the majority class label (around 76%). Whereas with more nodes, we can see an increased performance in all of our metrics. In fact, another observation is that #2 and #3 actually converged by surpassing the minimum error threshold, whereas the first model simply went through the 500 epochs.

I will be keeping 20 nodes for the rest of our experiments, as a constant.

Experiment #4 (500 epoch max, 0.1 learning rate, 20 nodes):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Minimum Error | # of Epochs | Learning Rate | # of Hidden Layer Nodes | Error @ output node1 after first epoch | Error @ output node2 after first epoch | Error @ output node1 after last epoch | Error @ output node2 after last epoch | Metrics |
| 0.1 | 354 | 0.1 | 20 | -0.0295 | 0.0387 | -4.397e-09 | 3.142e-09 | Accuracy: 0.94113  Macro F1: 0.9377  Micro F1: 0.9411 |
| 0.1 | 243 | 0.1 | 20 | -0.0402 | 0.0215 | -9.057e-09 | 6.736e-09 | Accuracy: 0.9411  Macro F1: 0.9100  Micro F1: 0.9411 |
| 0.1 | 244 | 0.1 | 20 | -0.0460 | 0.0286 | -9.970e-10 | 7.021e-10 | Accuracy: 0.8333  Macro F1: 0.7339  Micro F1: 0.8333 |
| 0.1 | 409 | 0.1 | 20 | -0.024 | 0.0501 | -9.881e-11 | 8.803e-11 | Accuracy: 0.9411  Macro F1: 0.9100  Micro F1: 0.9411 |
| 0.1 | 225 | 0.1 | 20 | -0.0460 | 0.0401 | -2.203e-07 | 2.785e-07 | Accuracy: 0.8333  Macro F1: 0.6516  Micro F1: 0.8333 |

Experiment #5 (500 epoch, 0.5 learning rate, 20 nodes):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Minimum Error | # of Epochs | Learning Rate | # of Hidden Layer Nodes | Error @ output node1 after first epoch | Error @ output node2 after first epoch | Error @ output node1 after last epoch | Error @ output node2 after last epoch | Metrics |
| 0.1 | 138 | 0.5 | 20 | -0.0357 | 0.0252 | -1.588e-07 | 1.105e-07 | Accuracy: 1.0  Macro F1: 1.0  Micro F1: 1.0 |
| 0.1 | 93 | 0.5 | 20 | -0.0643 | 0.0452 | -1.534e-06 | 6.053e-07 | Accuracy: 0.9411  Macro F1: 0.9100  Micro F1: 0.9411 |
| 0.1 | 99 | 0.5 | 20 | -0.0393 | 0.0446 | -3.385e-08 | 3.476e-08 | Accuracy: 0.8333  Macro F1: 0.7339  Micro F1: 0.8333 |
| 0.1 | 185 | 0.5 | 20 | -0.0224 | 0.0291 | -2.362e-09 | 2.938e-09 | Accuracy: 0.9411  Macro F1: 0.9100  Micro F1: 0.9411 |
| 0.1 | 91 | 0.5 | 20 | -0.0646 | 0.0573 | -9.191e-06 | 6.358e-06 | Accuracy: 0.8888  Macro F1: 0.8  Micro F1: 0.8888 |

After playing around with the learning rate, it seems a higher learning rate causes our network to converge in less epochs. However, the performance of the models aren’t affected so much, as the metrics are very comparable across both experiments (#4 and #5)

Overall, when comparing this to the first project, we can see a difference in all recorded metrics, with the ones from this model being hire. However, the Macro F1 score seems to have really considerably gone up, telling us we have actually become better at predicting each class label. The fact that our accuracy is well above .77 in all experiments except the first one, also shows we are performing better than the original class label distribution. Whereas in the first project, this wasn’t the case.