The below graphs show the results of a model with sigmoid activation in all neurons, trained for 500 epochs. A final test accuracy of 93.2% was achieved, while 94.3% was achieved in training. The following graphs show the output of task 1.

Chart, line chart

Description automatically generatedChart

Description automatically generated with medium confidence

**Test/Train Split**: The original split was 50/50. Increasing this to have 70% test data resulted in much lower accuracy. Train accuracy dropped to 77.3%, and test accuracy dropped to 85.7%. This makes sense as the model has much less data to train on, and so each epoch makes less of a difference on the weights. Flipping these percentages and having 30% test data let to a slight improvement in accuracy. Each batch has slightly more data to work with and as such, gave a test accuracy of 94%, and a train accuracy of 93.6%.

**Number of Neurons in Hidden Layer**: The original amount was 5 neurons. Unsurprisingly, this yielded a lower accuracy. The test accuracy only reached 87.3%. This will be because the model did not have enough weights to properly adjust for different datapoints. Surprisingly, increasing the number of nodes did not improve the test accuracy at all. All values from 5 up to 12 resulted in a slightly worse accuracy, ranging from 89% to 94%.

**Number of Epochs**: The original number of epochs was 500. Dropping the epochs by a factor of 10 to 50 yielded a much lower accuracy as the model simply was not given enough time to train. Test accuracy only reached 64%. On the other hand, increasing the epochs by a factor of 10 had much less of an impact. 5000 epochs resulted in a final test accuracy of 94.7%, an increase of only 0.4%. There is a fear of overfitting when running a relatively simple model for this many epochs. As seen in the graphs below, the training accuracy has reached 100% and the error plots are beginning to diverge from each other.

Graphical user interface

Description automatically generated with low confidenceA picture containing shape

Description automatically generated

**Learning Rate**: The overfitting is highlighted when increasing the learning rate to 0.1 from 0.01.

A picture containing graphical user interface

Description automatically generatedA picture containing shape

Description automatically generated

While the accuracy is not changing much, the errors seen in the test set is increasing consistently. Cutting the original learning rate in half provides a slower change in error. It takes much longer for the error curves to begin diverging. However, as the weights and biases of the neurons are now being adjusted by a smaller value a final test accuracy of 93.3% was achieved.

A picture containing graphical user interface

Description automatically generatedA picture containing shape

Description automatically generated

**Batch Size**: Altering the batch size has a similar impact to altering the number of epochs. Setting the batch size to 1 greatly increased the run time of the model’s training. It took nearly 3 minutes to train, up considerably from the 4.6 seconds of training for batches of 128. Selecting a more reasonable small batch size, of 64, had no real change on the training. Similarly, increasing the batch size to 400 had little impact to the accuracy, it did however decrease the training time by a further 2 seconds on average.

**Activation Functions**: Initially all the neurons had sigmoid activation in them. Changing the hidden layer to have Relu activation slightly improved the test accuracy to 94.7%. The model could not train correctly with the output layer as relu, it does not function with the crossentropy due to how it squashes negative numbers.

Final Model: From my findings after investigating the above parameters, I chose to continue with the following:

* Test/Train split of 30/70
* 5 neurons in hidden layer
* 3000 epochs to ensure overfitting is avoided
* Learning rate of 0.005
* Batch size of 128
* Relu activation in hidden layer, sigmoid in output layer