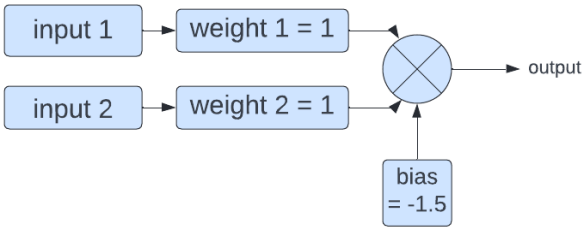
Computational Intelligence Coursework A

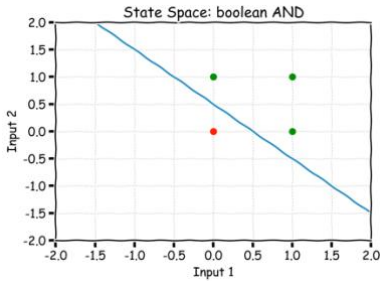
Exercise 1

For the logic gate representations below the activation function's hard threshold is $\begin{cases} 1 & x > 0 \\ 0 & all\ other\ x \end{cases}$

AND

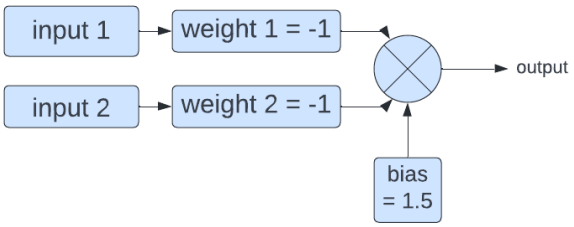
Input A	Input B	Output
0	0	0
0	1	0
1	0	0
1	1	1

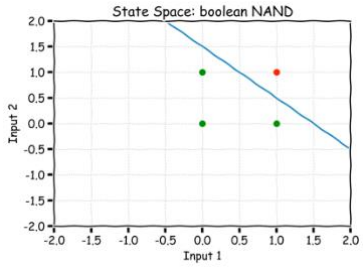




NAND

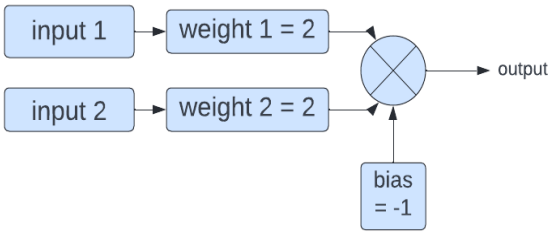
Input A	Input B	Output
0	0	0
0	1	0
1	0	0
1	1	1

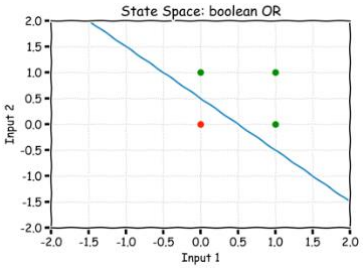




OR

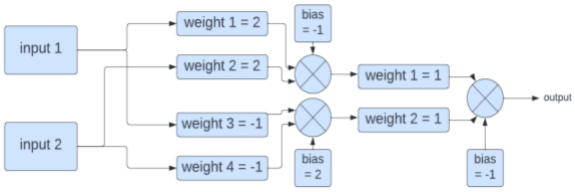
Input A	Input B	Output
0	0	0
0	1	0
1	0	0
1	1	1

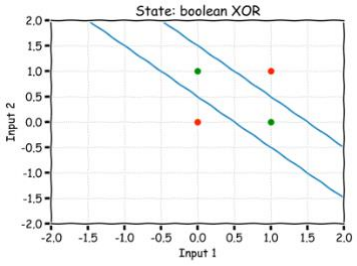




XOR

Input A	Input B	Output
0	0	0
0	1	0
1	0	0
1	1	1





Exercise 2

For this exercise the performance of the model has been assumed to be optimisation of the score/accuracy and training time/computation requirement with an equal weighting for each. The use case will best determine the measure of performance; however, this was not provided. Similarly, the script provided used a random selection from a normal distribution for the initial weights every time the class was called. This would introduce a small variance within results when running the same hyperparameters. The script results provided below used a fixed starting weights to ensure reproducibility of results. A single layer perceptron model was used.

MNIST

A coarse parametric sweep was conducted across hidden nodes starting from 100 to 1000 with a step size of 100 and learning rate from 0.1 to 1 with a step size of 0.1 producing Figure 1. From Figure 1 two general trends can be found.

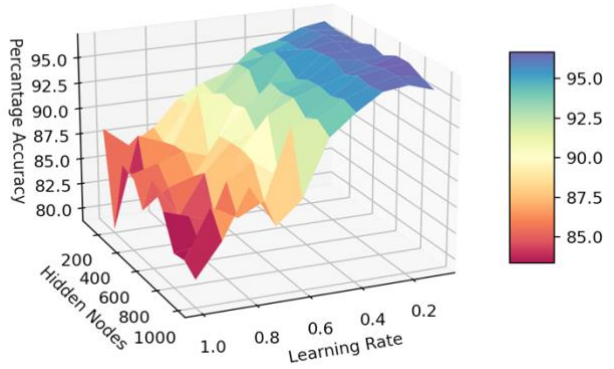


FIGURE 1 ACCURACY VS HIDDEN NODES VS LEARNING RATE

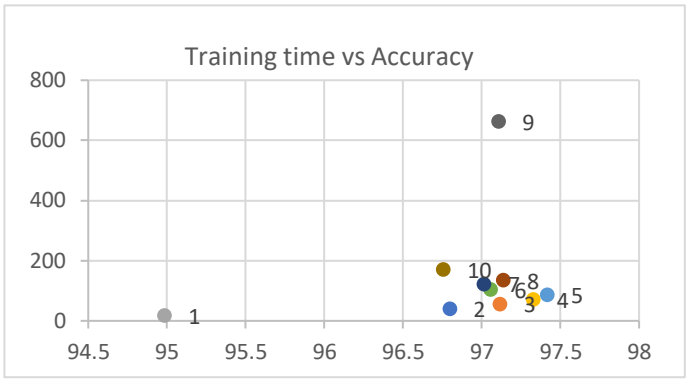


FIGURE 2 TRAINING TIME VS ACCURACY FOR DIFFERENT EPOCHS

Accuracy increases to its local maximum at 100-400 and 600-800. As learning rate decreases accuracy increases linearly. From the graph it is shown that at the lower learning rates the number of hidden nodes has little effect on the accuracy however greatly increases computation time. Therefore, 100 to 400 hidden nodes at 0.05 learning rate and 1 iteration were swept at a step of 10 of was performed producing 150 as the best accuracy with a low computation time.

A secondary sweep of number of iterations 1 to 10 with a step of 1 at 150 hidden nodes and learning rate of 0.05 was conducted. Results in the lower right quadrant were considered to be desirable results with 5 iterations having the

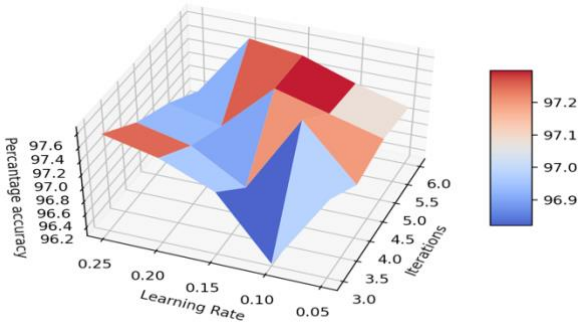


FIGURE 3 ACCURACY VS HIDDEN NODES VS LEARNING RATE

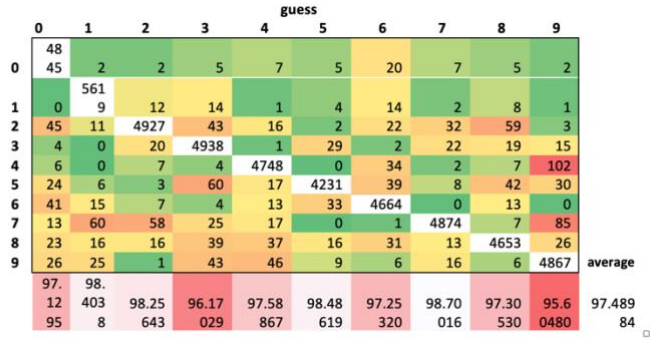


FIGURE 4: CONFUSION MATRIX

greatest accuracy with a low computation time. A fine sweep was conducted across learning rate from 0.05 to 0.25 with a step of 0.05 and iterations from 3 to 6 in steps of 1, producing Figure 3. The training time was not considered as previous iterations have shown that within the learning rate and iteration ranges selected training times are very similar. This resulted in the best score of 97.49% for 150 hidden nodes, 0.05 learning rate, at 5 iterations. From Figure 4 nines are the most confused and seven are the most accurate.

Fashion MNIST

The same process detailed above was conducted. The coarse sweep produced the same trend of maximums but was instead localised to 100-300 and 700-800. The next sweep was between 100-300 nodes at a step of 25 for 1 iteration at a learning rate of 0.05 and found that 150 hidden nodes have the highest accuracy with a low computation time. The secondary sweep then swept through 1 to 10 iterations with a step of 1 of the highest accuracy configurations giving 5 iterations as the greatest accuracy with a low computation time. This configuration was then compared using a fine sweep of 3 to 6 iterations in steps of 1, learning rate in steps of 0.05 from 0.05 to 0.25. This gave the best model as 150 hidden nodes, learning rate of 0.15, and 6 iterations with an

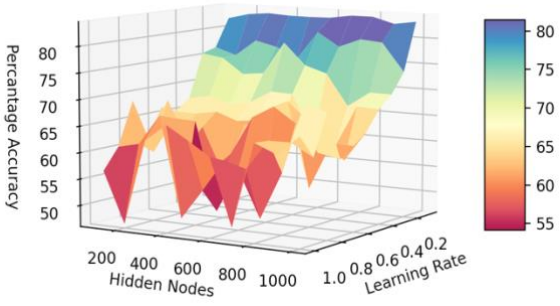
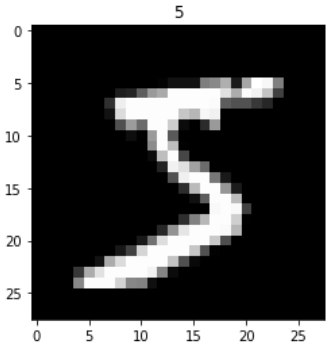


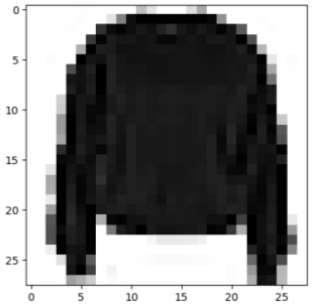
FIGURE 5: ACCURACY VS HIDDEN NODES VS LEARNING RATE

accuracy of 86.68%. A confusion Matrix was plotted and found category 3 was the most confused.

Fashion MNIST vs MNIST

The greatest score for MNIST was 97.49% and 86.68% for the Fashion MNIST. When exploring the data, it was found that there is an uneven distribution of labels in both the training and test sets of the MNIST compared to even distribution in the Fashion MNIST. This difference may account for a small difference in the scores as by favouring the ratio of the most confused labels the data set can remain the same size and potentially improve on specific labels. Most of the difference in accuracy can be seen in the greater variance in each label for fashion data set compared to MNIST. It stands to reason that for the fashion data set there will need to be more intermediate features to note to classify effectively. Similarly, MNIST has a stricter problem space aiding in classification.





Future Improvements

Due to time constraints starting weights were fixed to ensure reproducible results for comparison however this may lead to only capturing local maximums not the global maximum. To improve from this, multiple simulations should be run then the error calculated with more simulations reducing the error.

To improve the performance the quality and variance of the data can be improved. The MNIST and Fashion MNIST data set provided are pre-centred and size normalised. It could be improved by using changing the contrast shifting to improve edge detection, random rotations (20 degrees or less or 9 may become 6), and shifting pixels on the inputs to the network. This would like help to distinguish similar features such as ones and sevens having a similar shape. This changing pixel contrast would have the greatest effect on the fashion dataset as by exploring the data it was found that there is a greater pixel variance making fewer quality features. The data set can also be expanded by adding more data this would be dictated by the confusion matrix to preferentially add more of the confused labels or adjust the ratio to reflect confusion.

A different hyperparameter tuning technique could be used to improve the model score or allow for greater automation in the process. Techniques to explore the hyperparameter space are random search and grid search, these are quicker and less computation heavy but do not search the entire space. This makes it more likely to miss the global maximum. Alternatively, from the trained model feature selection with retraining can reduce model size and computation time while increasing accuracy.

To improve the MLP an additional layer could be added. This would increase the depth of the model giving a “deep” model. Deep models can be computationally efficient than training a single layer network with a large number of neurons. It is important to note that traditionally training MLPs with more than a few layers leads to problems caused by vanishing gradients and most applications do not require a second or third layer. Similarly, using an adaptive learning rate will aid gradient descent to minimum/maximum. Exploring the bias/variance trade off graphically to determine overfitting or underfitting is important to ensure the optimal choices/path for improvement is selected. For the quite defined classes/labels in both data sets overfitting is preferred to underfitting with seeing the greatest effect on the MNIST.

Another potential improvement from the 1-layer perceptron model is to use principal component analysis. This technique would allow for reasonable compression and classification by reforming the pixel space in terms of the most relevant ‘principal’ components with the goal of reducing training and interpretation time. For the MNIST this technique will have a limited effect on the training and interpretation time as PCA doesn’t respect the spatial relationship between pixels that other models such as CNNs can. This is especially true for datasets which are not pre-centred as PCA lacks invariance to translation. PCA however, will outperform most basic machine learning models/solutions when looking to improve training and interpretation time. For this use case it would allow for either quicker training or using a larger dataset for the same training time. Like PCA using a convolutional neural network (CNN) would reduce the high dimensionality of images without losing its information. This makes it suited for image classification, including higher dimensional images (especially colour images). The CNN would potentially add about 1% of accuracy to MNIST but would have a much greater effect on the fashion data as this would likely have more intermediate features to develop/capture.