AML ASSIGNMENT 1 REPORT

In the process of our analysis, we looked at a network model's performance after 20 training epochs. 512 was the batch size used. We aimed to assess the variation in the validation loss of the model under different configurations.

Validation Loss Functions: First, we contrasted the Mean Squared Error (MSE) with a regularisation hyperparameter of 0.005 and the Binary Cross Entropy (BCE) loss function. Furthermore, we investigated differences in node counts (16, 32, and 64) while preserving a three-layer architecture using "tanh" as the activation function (or "Relu" for comparison in certain situations). In simple terms validation loss, Mean Squared Error (MSE) consistently outperformed Binary Cross Entropy (BCE) with tanh activation, exhibiting more stable and decreasing trends.For MSE, regularisation techniques were used, which helped to reduce overfitting.

• Nodes: 16 with 3 Layers; Regularisation Applied; BCE vs. MSE Tanh
We saw that the validation loss for MSE started at 0.17 and progressively dropped to 0.12.
However, the validation loss fluctuated and began at 0.4 when BCE was used as the loss function. ultimately rose to 0.58.
• Nodes: 32 with 3 Layers; Regularisation Applied; BCE vs. MSE, ReLU
The validation loss for MSE in this configuration began at 0.15. declined gradually to a point of about 0.13.
When using BCE as the loss function, we discovered that the validation loss started out at 0.5 but fluctuated before coming down to about 0.4.

• Using regularisation techniques, we experimented with configurations that included 32 nodes, 3 layers, BCE or MSE, and a tanh activation function.
• The MSE validation loss started out at 0.24 and decreased gradually to 0.1. Conversely, for BCE, the validation loss increased to 0.6 after initially decreasing to 0.4 until the epoch.
• Using 64 nodes with three layers and a ReLU activation function while utilising regularisation techniques was another configuration we experimented with.
• In this instance, the validation loss for MSE started at 0.2, varied during training, and finally dropped to 0.13.

• In a similar vein, the validation loss for BCE under this configuration began at 0.5. varied throughout the training as well. increased to 0.7 in the end.
• As a final experiment, we investigated applying regularisation techniques to 64 nodes with 3 layers and a tanh activation function.

• In this configuration, the validation loss for MSE started at 0.2. fluctuated a bit during training before levelling off at a value of about 0.1.
• In contrast, the validation loss for BCE under these conditions began at 0.5. showed some variations while in training, but eventually stayed consistent at that level.

Summary of Validation and Training Accuracy:

We also examined the accuracy of training and validation across configurations. In one instance, we used the Tanh activation function, L2 regularisation, and dropout techniques to set up a network with 16 nodes spread across three layers. The validation accuracy fluctuated throughout the training process. ultimately reached a level of about 87% accuracy, while the training accuracy grew steadily to reach a level of roughly 94%. We experimented with different network configurations, utilising the Tanh activation function, with 32 nodes and 3 layers. We used regularisation techniques like dropout and L2 regularisation to prevent overfitting.

We noticed variations in accuracy during the validation stage. In the end, it came to 87%. The training accuracy grew steadily until it reached 94%.

We also tested a different configuration that used three layers and 64 nodes with the Tanh activation function. Once more, we used regularisation strategies to reduce overfitting.

The validation accuracy showed fluctuations similar to the configuration. Attained 87% in the end. Training accuracy progressively increased. reached a 95% success rate. It is noteworthy, nevertheless, that a discrepancy existed between the validation and training accuracies, indicating potential problems with the data. Sampling techniques may be used to help with this issue.

Summary for Test Accuracy:

We compared two models—one with three layers, 64 hidden nodes, and a tanh activation function—in order to assess test accuracy. trained over four epochs, whereas the ReLU activation function was used in the other model.

With a loss rate of 16.32, the tanh activation model yielded an 88% test accuracy.

ReLU activation, on the other hand, produced an 86% test accuracy and a 21.25 loss rate.

Optimizer:

Adam and RMSprop were the two optimizers we used in our experiments.

• Adam:

Based on gradients, this optimizer modifies the learning rate for each parameter separately. It is dependable and effective. With the Adam optimizer, the model's accuracy was 88%.

• RMSprop:

This optimizer computes a moving average of gradients to adjust the learning rate. For tasks, it's helpful. Using RMSprop, the test set's accuracy was 86%. When comparing neural network configurations, we discovered that the model with three layers, 64 hidden nodes, L2 regularisation, and dropout tended to perform better in terms of test accuracy and validation while maintaining a relatively simple model. The Adam optimizer performed better in this instance than RMSprop.