**AML ASSIGNMENT 1 REPORT**

In our analysis, we assessed the performance of a neural network model that underwent training for 20 epochs, employing a batch size of 512. Our emphasis was on examining the fluctuations in the model’s validation loss across different configurations.

**Validation Loss Functions:**

Initially, we conducted a comparison between two loss functions: Binary Cross Entropy (BCE) and Mean Squared Error (MSE) with a regularization hyperparameter set to 0.005 specifically for MSE. Furthermore, we investigated different configurations of the number of nodes (16, 32, and 64), while keeping a consistent three-layer architecture and utilizing the "tanh" activation function, with the inclusion of "Relu" in some instances for comparative analysis.

• Nodes=16 with 3 Layers, BCE vs MSE tanh; (Regularization Applied)

In the case of Mean Squared Error (MSE), the validation loss commenced at 0.17 and progressively diminished to 0.12. Conversely, when employing Binary Cross-Entropy (BCE) as the loss function, the validation loss initiated at 0.4 and experienced fluctuations, ultimately rising to 0.58.

• Nodes=32 with 3 Layers, BCE vs MSE, ReLU; (Regularization Applied)

In this setup, the mean squared error (MSE) exhibited an initial validation loss of 0.15, steadily declining to approximately 0.13. Conversely, when using binary cross-entropy (BCE) as the loss function, the initial validation loss was higher at 0.5, showing fluctuations before eventually stabilizing around 0.4.

• Using regularization approaches, we experimented with configurations that included 32 nodes, 3 layers, Tanh activation function, and either BCE or MSE.

• The MSE initial validation loss started at 0.24 and steadily decreased to 0.1 over time.

• Conversely, in the case of Binary Cross-Entropy (BCE), the validation loss initially began at 0.4, progressively declined throughout the epoch, but ultimately rose to 0.6.

• We experimented with a different setup, employing 64 nodes, 3 layers, and a ReLU activation function, incorporating regularization techniques into the model.

Top of Form

• In this instance, the Mean Squared Error (MSE) for validation initially started at 0.2, experienced fluctuations during the middle of the training process, but ultimately decreased to 0.13.

• In the same manner, when using Binary Cross-Entropy (BCE) with this setup, the initial validation loss stood at 0.5, fluctuated throughout training, and eventually rose to 0.7.

• Finally, we investigated the application of regularization techniques using a configuration of 64 nodes, 3 layers, and a Tanh activation function.

• In this configuration, the Mean Squared Error (MSE) for validation initially started at 0.2. It exhibited some fluctuations during the training process before eventually decreasing to approximately 0.1.

• On the flip side, when using Binary Cross Entropy (BCE) with these configurations, the validation loss initially began at 0.5. While there were some fluctuations observed during the training process, it eventually stabilized and consistently hovered around that value.

**Summary of Validation and Training Accuracy:**

We examined the validation and training accuracy across different configurations as part of our analysis.

• In a particular instance, we configured a neural network comprising 16 nodes distributed across three layers. The activation function employed was tanh, and we incorporated L2 regularization and dropout techniques.

The validation accuracy experienced variations during the training process, ultimately stabilizing around an accuracy of roughly 87%.

As the training accuracy consistently elevated, ultimately reaching approximately 94%, we conducted various trials with network configurations, such as employing 32 nodes and incorporating 3 layers with the tanh activation function. To mitigate overfitting, we implemented preventive measures, including the utilization of regularization techniques like L2 regularization and dropout.

Throughout the validation stage, variations in precision were noted, eventually stabilizing at 87%. In the case of training accuracy, a progressive ascent was witnessed, ultimately reaching a level of 94%.

We explored an alternative setup involving 64 nodes and 3 layers, utilizing the tanh activation function. Additionally, we implemented regularization methods to address overfitting concerns in our testing.

Comparable to the setup, the validation accuracy experienced variations before stabilizing at 87%. The training accuracy, on the other hand, increased steadily and reached a peak of 95%. It is important to highlight the discrepancy between training and validation accuracies, indicating potential data-related challenges. Utilizing sampling techniques may be beneficial in mitigating this issue.

**Summary for Test Accuracy:**

In assessing test accuracy, we juxtaposed two models: the first featured 64 hidden nodes, three layers, and employed the tanh activation function during a four-epoch training period. Meanwhile, the second model utilized the ReLU activation function.

The neural network employing the hyperbolic tangent (tanh) activation function demonstrated an 88% accuracy on the test set, accompanied by a loss rate of 16.32.

Alternatively, employing ReLU activation resulted in a test accuracy of 86%, accompanied by a loss rate of 21.25%.

**Optimizer:**

We performed trials utilizing two optimization algorithms: Adam and RMSprop.

• **Adam:** This optimization technique customizes the learning rates for each parameter by considering their respective gradients. It proves to be both effective and dependable. The model successfully attained an accuracy of 88% through the utilization of the Adam optimizer.

• **RMSprop:** This learning rate optimizer employs a dynamic adjustment mechanism by computing an average of gradients over time. It proves beneficial for various tasks. The test sets accuracy, when utilizing RMSprop, reached 86%.

When examining neutral networks setups, we observed that a configuration featuring 64 hidden nodes, three layers, L2 regularization, and dropout consistently demonstrated superior performance in validation and test accuracy. This model, which maintained a relatively straightforward structure, outperformed its counterparts, with the Adam optimizer proving more effective than RMSprop in this scenario.

Top of Form

Top of Form

Top of Form

Top of Form