AML_Assignment 1

IMDB Sentiment Analysis Model Optimization Report:

IMDB dataset has 50,000 reviews. These reviews contain 10,000 words in total. These reviews have all already been given labels. The following are the steps I follow to produce better validation and test accuracy.

* I imported all the modules required to build and run a model. Then, I loaded the IMDB dataset.

*Since we can't put integers into neural networks, the data is then vectorized. So, I decided to divide the dataset into two sections of the dataset one for training and the other for testing—are separated out. 25000 reviews are included in each component.

* I built the first two models with 16 hidden units, ReLu activation function, rmsprop optimizer, and binary cross entropy loss function. The number of hidden layers in each model is Changed. It shows in the experiment 1 and experiment 2 as follows:

Experiment 1: Number of Hidden Layers

One Hidden Layer

- Validation Accuracy: 0.8691

- Test Accuracy: 0.8564

Experiment 2: Three Hidden Layers

- Validation Accuracy: 0.8637

- Test Accuracy: 0.8509

*We can say that one hidden layer performs better compared to three hidden layers since the data samples in the IMDB dataset are constrained. As a result of overfitting caused by multiple layers, test and validation accuracy is declining.

* Next model is built with one hidden layer, ReLU activation function, rmsprop optimizer, and binary cross entropy loss function. I decided to work with 32 units. Which is in experiment 3:

Experiment 3: Hidden Layers with 32 Units:

- Validation Accuracy: 0.8655

- Test Accuracy: 0.8515

*It shows that increasing the hidden units is also leading to the same problem of overfitting as that of the hidden layers. 16 hidden units is best combination to use as it is producing the best accuracy.

* Then I check the accuracies by changing the loss function to mse and compare it with binary cross entropy. These models are one hidden layer,16 hidden units, ReLU activation function, rmsprop optimizer which is in the experiment 4:

Experiment 4: MSE Loss Function

Mean Squared Error (MSE)

- Validation Accuracy: 0.8631

- Test Accuracy: 0.8527

*Then for a given dataset it is evident that the binary cross entropy loss function is better to use compared to MSE loss function.

*Again, in the next model which is in experiment 5 is built using one hidden layer,16 hidden units, rmsprop optimizer, Binary cross entropy loss function to compare the performance of tanh activation function.

Experiment 5: Tanh Activation Function

- Validation Accuracy: 0.8635

- Test Accuracy: 0.8432

By comparing all the above models, the Relu has given better accuracy than tanh. So, I decided to use relu as my activation function for the next models.

* In the next model that is experiment 6, I used the Drop out (0.5). In this model, I used one hidden layer, 16 hidden units, rmsprop optimizer, Binary cross entropy loss function and relu activation function.

Experiment 6: Dropout (Rate: 0.5)

- Validation Accuracy: 0.8823

- Test Accuracy: 0.8731

Accuracy is decreasing when dropouts are used. This could be the case given that the model is not overfitting, and the use of dropout is unfavourable in such a scenario.

*The next model is built using one hidden layer, 16 hidden units, rmsprop optimizer, Binary cross entropy loss function, and relu activation function. L2 regularizers are used and compared their performance.

Experiment 7: Regularization technique:

Model Architecture: Hidden layers with 16 units each and ReLU activation, with L2 regularization (regularization strength = 0.001) applied to the kernel.

- Validation Accuracy: 0.8786

- Test Accuracy: 0.8776

It is evident that regularize is not needed for this data as its usage is not helping to improve the accuracy.

Conclusion:

The effectiveness of a model in a neural network is influenced by several things. The size of the data sample is the first. To improve accuracy, we require additional data samples.

Now, in this case with 25000 data, one layer with 16 hidden units outperforms other hidden layers. Considering that the ReLU function gives the model a benefit.

I discovered that we shouldn't utilize it until it is necessary when I noticed that the accuracy was declining when I tried using various optimizers and regularizers.

The model might perform better in the case if we train with multiple layers, utilize regularizers, and dropouts and have a large amount of data.

Last but not least, the ideal collection of hyperparameters for the provided dataset is one hidden layer, 16 hidden units, Binary cross entropy loss function, relu activation function, and rmsprop optimizer.