AML – ASSIGNMENT2

NEURAL NETWORKS

SUMMARY REPORT

There are fifty thousand movie reviews on IMDb, divided equally into two categories (25,000 reviews for each category: "positive" and "negative"). Utilizing this dataset, this study attempts to investigate several approaches for improving an existing neural network model's performance. We evaluate these alterations' effects on model performance by methodically adjusting various parameters and strategies, including as the activation function, loss function, number of units, hidden layers, and the use of regularization approaches like dropout.

Information Processing:

To prepare the raw text data from the IMDb dataset for neural network training, there were a few important steps that needed to be taken. We focused on the top 10,000 most often occurring terms to minimize the dimensionality of the input space due to the dataset's enormous vocabulary size. After that, a dictionary was used to map each chosen word to a distinct numeric identification, converting the text data into a numeric format. To make it compatible with neural networks, this integer representation was then transformed into tensors. For lengthier reviews, we used trimming, and for shorter reviews, zero-padding, to guarantee consistent vector size throughout all reviews. By following this process, fixed-length vectors were produced, each of whose elements represents an index word in the chosen dictionary.

Techniques:

We imposed limitations on the maximum word count and review duration to produce a consistent dataset for training. The first neural network model we built included a single hidden layer and sixteen units. We trained the model using the Adam optimizer and used dropout regularization to handle any possible overfitting. Mean squared error (MSE) and binary cross-entropy were the two loss functions we used in our experiments using the two main activation functions, tanh and ReLU.

We gradually raised the number of hidden layers to further improve the model's capacity and complexity, resulting in models with one, two, or three hidden layers. Both training and test datasets were used to thoroughly assess these models' performance. Compared to models with a single hidden layer, our results showed that increasing hidden layers considerably improved test accuracy and validation performance.

Accuracy Percentage and Hidden Layers:

• Accuracy of 1 hidden layer, 16 units: 87.26%
•16 units with 3 hidden layers; accuracy = 88.33%
• 32 units, 3 hidden layers: accuracy = 86.48%
• 64 units, 2 hidden layers; accuracy = 87.37%
• 128 units with 3 hidden layers; accuracy = 87.14%
• 16 units, 3 hidden layers: 88.62% accuracy
• 16 units, 1 hidden layer; accuracy = 87.09%
• 16 units with 3 hidden layers; accuracy = 86.74%
• 16 units, 2 hidden layers: accuracy = 88.19%
• 32 units, 3 hidden layers: accuracy = 86.48%

Conclusion:

We built a new model with dropout layers as part of our study on dropout regularization as a means of reducing overfitting. Remarkably, as compared to the baseline model, this regularized model showed lower validation accuracy. As the data show, different neural network architecture and training strategies result in different degrees of accuracy and loss. The Model Hyper outperformed the model with three dense layers and a dropout rate of 0.5 among the examined models, achieving the highest accuracy and lowest loss.

The accuracy of models using the tanh activation function was negatively impacted, perhaps because of the vanishing gradient problem, even though the MSE loss function yielded the lowest loss values in the context of binary cross-entropy. In the end, Model Hyper outperformed the model using Mean Squared Error (MSE) by a small margin, even though the Adam optimizer worked well. Overall accuracy was decreased for the regularized model, indicating that Model Hyper was the highest-performing model among the models examined.

Subsequent Tasks:

Future research endeavors may delve further into more complex structures like recurrent neural networks (RNNs) or convolutional neural networks (CNNs) to further improve model performance. These architectures may offer more profound understanding of the sequential nature of text data. Furthermore, the accuracy and resilience of the model could be improved by adding methods like learning rate scheduling, data augmentation, and more complex embedding layers.