**Neural Network Model Performance Enhancement for IMDB Dataset**

**Assignment-2**

**BA-64061-001 Advanced Machine Learning**

**Chaojiang (CJ) Wu, Ph.D.**

Sai Bharath Goud Pudari

[spudari@kent.edu](mailto:spudari@kent.edu)

Student ID: 811354970

**Assignment Goal**

This assignment's objective is to categorize movie reviews as either positive or negative by optimizing a neural network model for sentiment classification using the IMDB dataset. This is accomplished by methodically adjusting a number of hyperparameters, including the number of hidden layers, hidden units, activation functions, and loss functions, in order to examine how they affect the performance of the model. To ensure the model performs well on unknown data, regularization techniques—in particular, dropout—are used to promote generalization and lessen overfitting. The assignment seeks to determine the optimal configuration that strikes a balance between accuracy, complexity, and processing efficiency by carrying out several tests and comparing the outcomes.

**Dataset: IMDB Sentiment Classification**

The IMDB dataset, consisting of movie reviews labeled as positive or negative, is used for this task. The dataset includes 50,000 reviews, divided into 25,000 training examples and 25,000 test examples. The model’s task is to classify the sentiment of the reviews.

**Experimental Setup**

The neural network model is based on a simple feed-forward architecture with the following characteristics:

- Input layer: A 10,000-dimensional vector representing the binary multi-hot encoding of the top 10,000 most frequent words in the reviews.

- Hidden layers: The number of layers and units varies depending on the experiment.

- Output layer: A single unit with a sigmoid activation function to classify reviews as positive or negative.

**Effect of Hidden Layers:**

* With one hidden layer, the model converges faster, and the validation accuracy remains close to the training accuracy, indicating good generalization.
* Adding more hidden layers increased the complexity but resulted in only marginal improvements (or even slight declines) in accuracy, potentially due to overfitting.

**Hidden Layer Accuracy:**

Increasing the number of hidden layers improves training accuracy but does not significantly enhance validation accuracy, indicating potential overfitting and the need for an optimal balance in model complexity.

**Plot 1: Number of Hidden Layers – Accuracy**

A graph of a number of layers

AI-generated content may be incorrect.

**Plot 2: Number of Hidden Layers – Loss**A graph of a loss

AI-generated content may be incorrect.

Analysis:

The training accuracy improves with more epochs, but validation accuracy remains relatively constant, indicating that additional hidden layers do not significantly enhance performance. The training loss decreases as expected, but validation loss does not show substantial improvement, suggesting overfitting in deeper networks. A simpler architecture with fewer hidden layers appears to generalize better for this dataset. This suggests that while deeper networks have a higher capacity to learn complex patterns, they may also capture noise from the training data, leading to reduced generalization and diminishing returns in validation performance.

**Neuron Count – Accuracy**

Increasing the number of neurons per layer improves training accuracy, but validation accuracy remains relatively stable, suggesting that beyond a certain point, adding more neurons leads to diminishing returns and potential overfitting.

**Plot 3: Neuron Count - Accuracy**

A graph of a number of lines

AI-generated content may be incorrect.

**Plot 4: Neuron Count - Loss**

A graph of a number of lines

AI-generated content may be incorrect.

**Analysis:**

The training accuracy improves consistently with an increasing number of neurons, but the validation accuracy remains relatively flat, indicating that more neurons do not significantly enhance generalization. Similarly, while the training loss decreases, the validation loss remains almost constant, suggesting potential overfitting when using too many neurons. This implies that increasing neuron count beyond a certain threshold leads to diminishing returns and does not improve model performance on unseen data. Therefore, selecting an optimal number of neurons is crucial, as an excessively high neuron count may increase model complexity without meaningful performance gains, leading to higher computational costs and potential overfitting.

**Loss Function Comparison – Accuracy:**

The accuracy comparison of different loss functions shows that while training accuracy improves steadily, validation accuracy remains stable or declines slightly, suggesting that certain loss functions, like binary cross-entropy, are better suited for classification tasks than others, such as mean squared error.

**Plot 5: Loss Function Comparison – Accuracy**

A graph of a line

AI-generated content may be incorrect.

**Plot 6: Loss Function Comparison – Loss**

A graph of loss and loss

AI-generated content may be incorrect.

**Analysis:**

The training accuracy improves steadily, but validation accuracy remains stagnant and slightly declines, suggesting overfitting when using certain loss functions. Similarly, the training loss decreases while the validation loss remains nearly constant, indicating that the model memorizes training data rather than generalizing well. This suggests that an alternative loss function better suited for classification tasks, such as binary cross-entropy, may improve model generalization.

**Activation Function Comparison:**

* The model with ReLU activation shows a sharp increase in accuracy during the initial epochs, leading to a higher final accuracy compared to the Tanh-activated model, which converged more slowly and with less stability.

**Plot 7: Activation Function Comparison – Accuracy**

A graph of a line graph

AI-generated content may be incorrect.

**Plot 8: Activation Function Comparison – Loss**

A graph of a line graph

AI-generated content may be incorrect.

**Analysis:**

The training accuracy improves significantly over epochs, but the validation accuracy stagnates and even slightly declines, indicating overfitting with certain activation functions. Similarly, the training loss continues to decrease, while the validation loss remains relatively constant, suggesting that the model is learning the training data well but not generalizing effectively. This suggests that choosing an appropriate activation function, such as ReLU, can help prevent overfitting and improve overall model performance.

**Dropout Regularization – Accuracy**

Applying dropout regularization helps prevent overfitting by reducing training accuracy slightly while maintaining stable validation accuracy, ensuring better generalization to unseen data.

**Plot 9: Dropout Regularization – Accuracy**

A graph of different colored lines

AI-generated content may be incorrect.

**Plot 10: Dropout Regularization – Loss**

A graph of a dropout rate

AI-generated content may be incorrect.

**Analysis:**

The training accuracy continues to increase across epochs, but validation accuracy remains stable, suggesting that dropout helps prevent overfitting. Similarly, while training loss decreases consistently, validation loss remains nearly constant, indicating that an optimal dropout rate improves generalization. A moderate dropout rate balances model complexity and generalization, preventing the model from memorizing training data while maintaining stable validation performance.

**Best variables for the Model**

* **Hidden Layers: 1 Layer**
* **Hidden Units: 256 Units**
* **Loss Function: Binary Crossentropy**
* **Activation Function: ReLU**
* **Regularization: Dropout Rate of 0.7**

**Results:**

Final Evaluation on Test Data-

* Test Loss: 0.5992,
* Test Accuracy: 0.8630

**Conclusion**

The best fit model for the neural network is characterized by a single hidden layer containing 256 units, which optimally balances complexity and performance. It employs the Binary Crossentropy loss function, which yielded the highest validation and test accuracies. The ReLU activation function further enhanced the model's performance, contributing to an impressive validation accuracy. Additionally, a dropout rate of 0.7 was utilized for regularization, effectively minimizing overfitting while maintaining strong accuracy metrics. Overall, this configuration resulted in a test loss of 0.4936 and a test accuracy of 0.8619, marking it as the most effective model for the dataset.