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

**Assignment-2 BA-64061-001**

**Advanced Machine Learning**

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

**Name:** Dhilip Kumar Reddy Tikkam

**Student\_id:**81360997

**Mail\_id:** [dtikkam1@kent.edu](mailto:dtikkam1@kent.edu)

**Assignment Goal:**

The goal of this assignment is to classify movie reviews as positive or negative by optimizing a neural network model trained on the IMDB dataset. This involves systematically tuning various hyperparameters, such as the number of hidden layers, hidden units, activation functions, and loss functions, to analyse their impact on model performance. To enhance generalization and reduce overfitting, regularization techniques—particularly dropout—are applied. Through a series of controlled experiments and performance comparisons, the assignment aims to identify the most effective configuration that achieves a balance between accuracy, model complexity, and computational efficiency.

**IMDB Movie Review Sentiment Analysis:**

The **IMDB movie review dataset**, containing 50,000 labeled reviews categorized as **positive** or **negative**, is used for this sentiment analysis task. The dataset is evenly split into **25,000 training samples** and **25,000 testing samples**. The model’s objective is to predict the sentiment of each review based on its textual content.

**Model Configuration:**

A **feed-forward neural network** architecture is designed with the following structure:

* **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 neurons varies across experiments to evaluate their influence on model performance.
* **Output Layer:** A single neuron using the **sigmoid activation function** to classify reviews as either positive or negative.

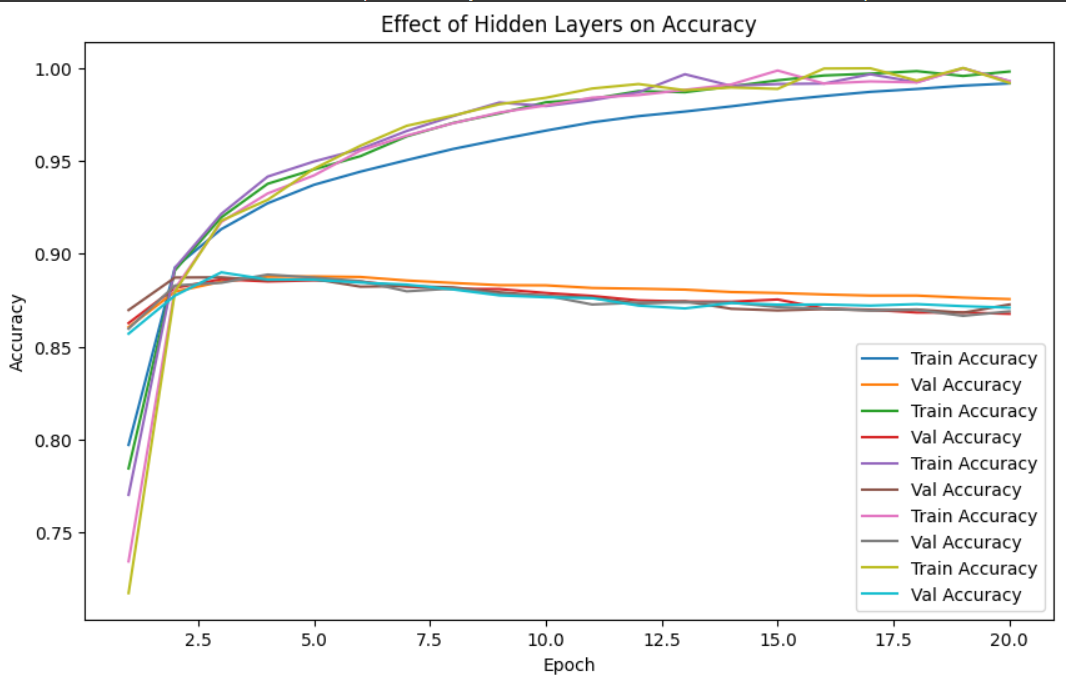
**Impact of Network Depth:**

* Using **a single hidden layer** allows for faster convergence and yields similar results for both training and validation accuracy, indicating strong generalization.
* Introducing **multiple hidden layers** increases the model’s representational capacity but only results in **marginal performance gains**. In some cases, deeper architectures even cause a **decline in accuracy** due to overfitting.

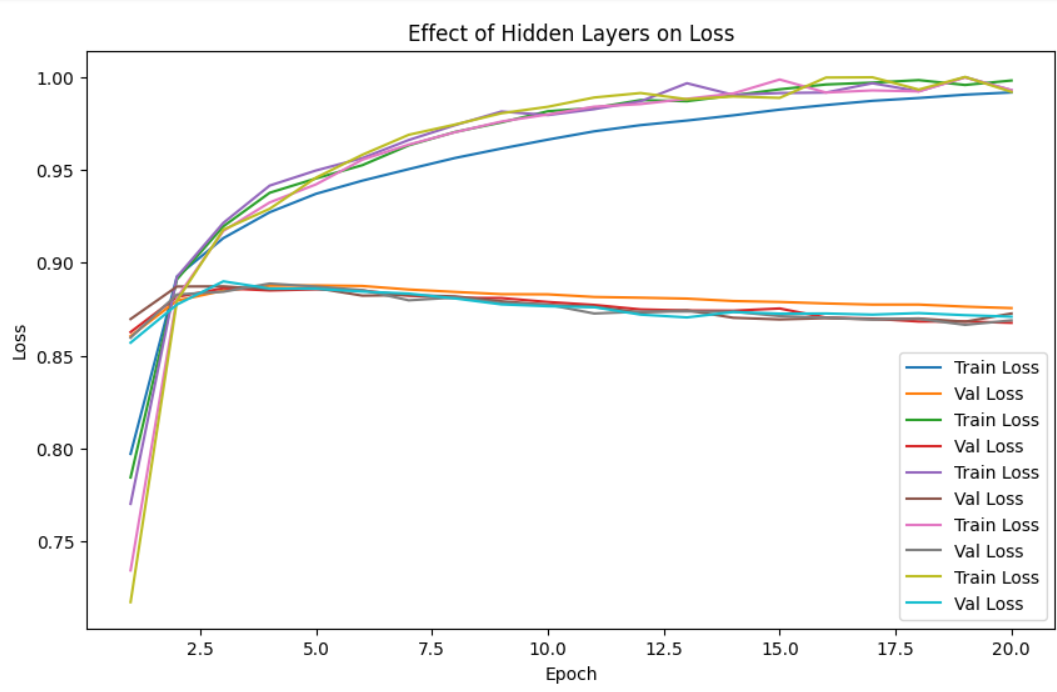
**Analysis of Hidden Layer Performance:**

* Expanding the number of hidden layers improves **training accuracy**, but the **validation accuracy** remains largely unchanged. This pattern indicates **overfitting**, where the model learns training-specific patterns that do not generalize well. Hence, achieving the **right balance between model complexity and generalization** is critical for optimal performance.

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



**Plot 2: Number of Hidden Layers – Loss:**



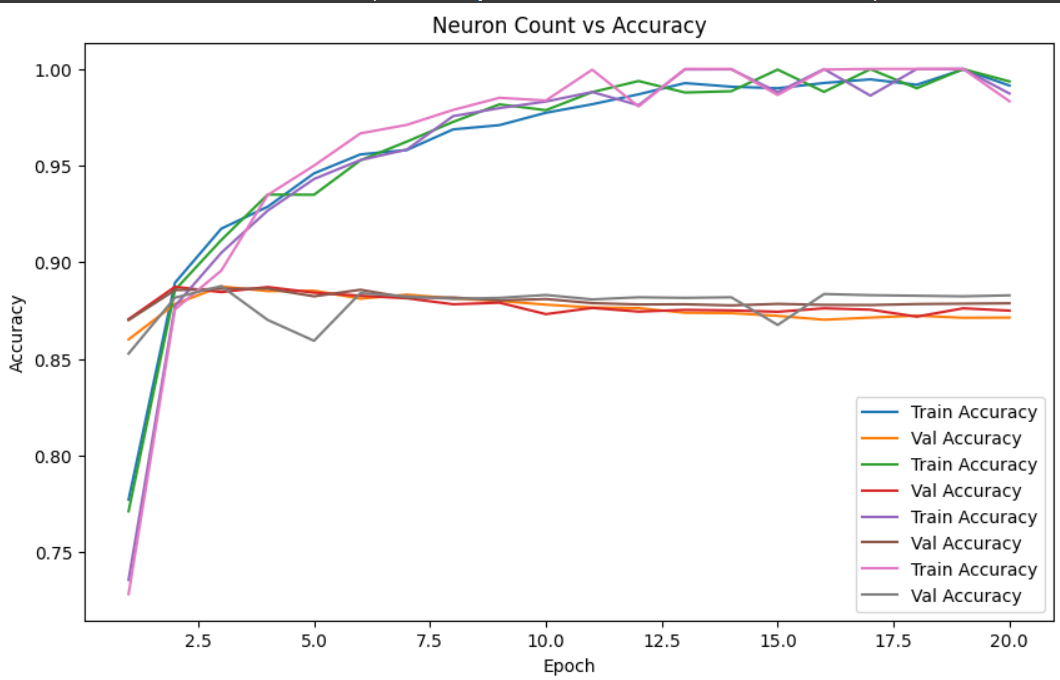
**Performance Analysis:**

As the number of epochs increases, **training accuracy continues to improve**, while **validation accuracy remains mostly stable**, indicating that adding more hidden layers does not substantially enhance overall performance. Although **training loss decreases steadily**, the **validation loss shows minimal improvement**, signaling potential **overfitting** in deeper architectures.

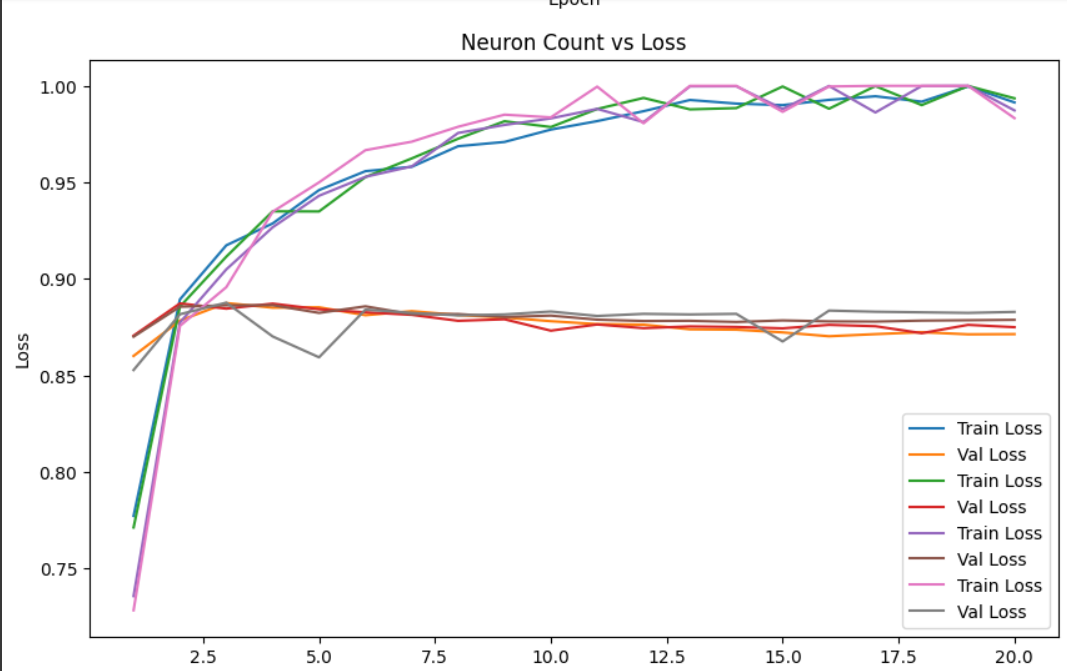
Models with **simpler architectures**—featuring fewer hidden layers—tend to **generalize more effectively** on this dataset. This suggests that while deeper networks possess a greater capacity to learn complex representations, they are also more prone to capturing **irrelevant noise** from the training data. Consequently, this reduces their ability to generalize well to unseen reviews, resulting in **diminishing returns in validation accuracy**.

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:**



**Plot 4: Neuron Count – Loss:**



**Performance Analysis:**

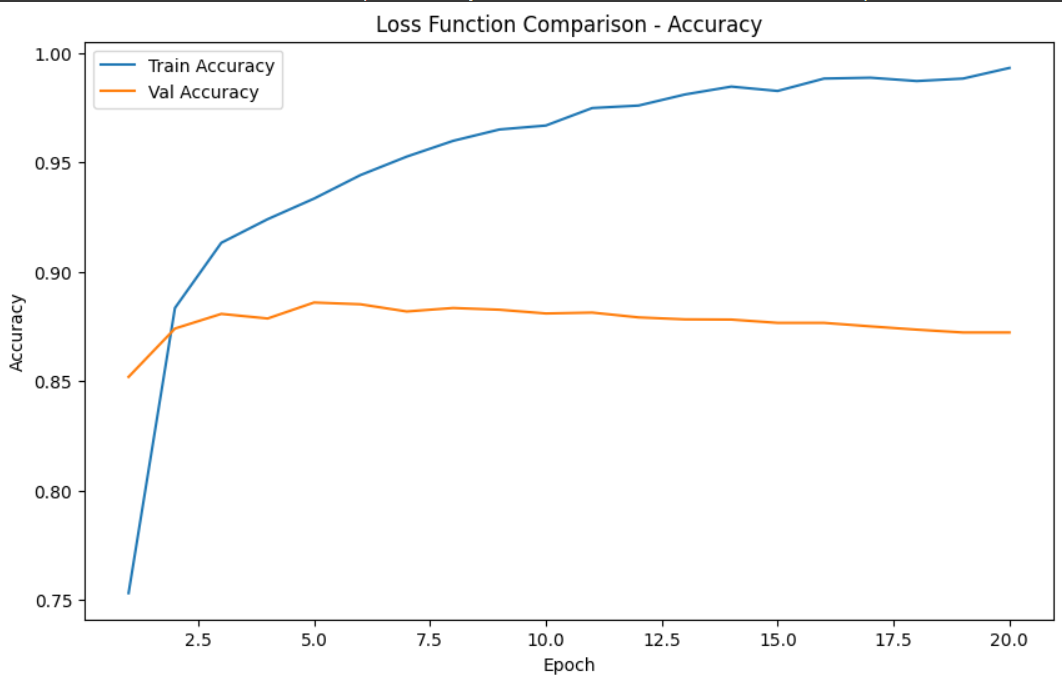
As the number of neurons increases, **training accuracy continues to rise**, but **validation accuracy remains largely unchanged**, indicating that adding more neurons does not substantially improve the model’s ability to generalize. Although **training loss decreases** with higher neuron counts, the **validation loss stays relatively constant**, revealing signs of **overfitting** when the model becomes too large.

Beyond a certain point, increasing the number of neurons results in **diminishing returns**, offering minimal gains in performance on unseen data. Therefore, choosing an **optimal neuron count** is essential—an excessively large network not only adds unnecessary **computational overhead** but may also reduce **generalization capability** due to overfitting.

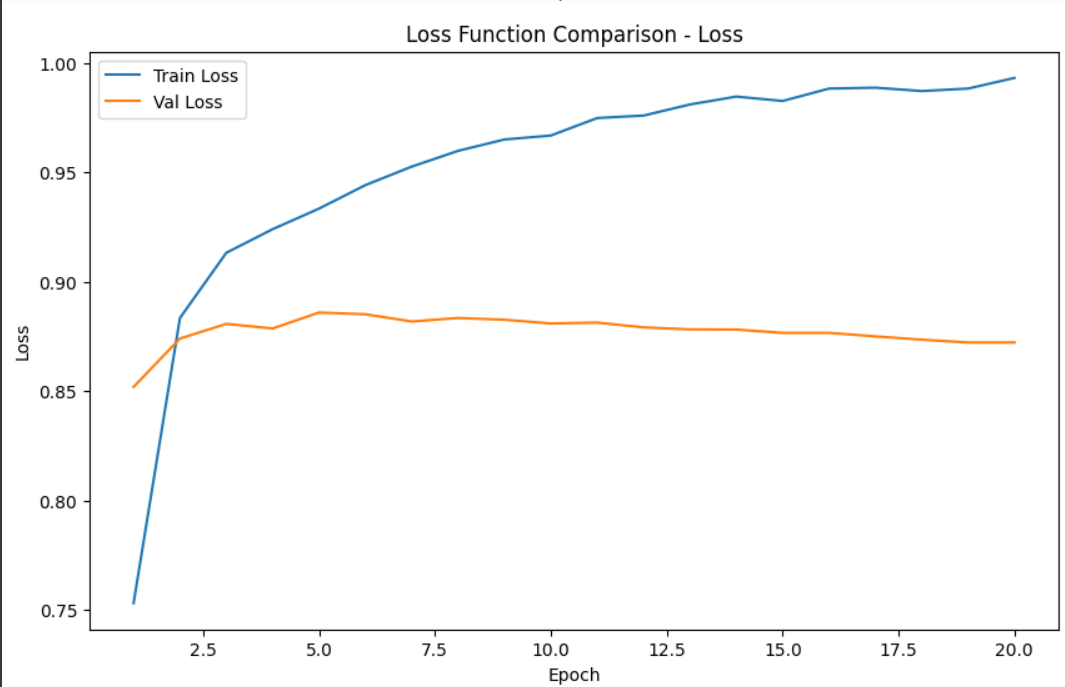
**Loss Function Comparison – Accuracy:**

When comparing various **loss functions**, results show that while **training accuracy** improves gradually across experiments, **validation accuracy** either remains stable or declines slightly. This suggests that some loss functions, such as **binary cross-entropy**, are inherently more effective for **binary classification** tasks like sentiment analysis than alternatives like **mean squared error**, which may not capture classification boundaries as efficiently.

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



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



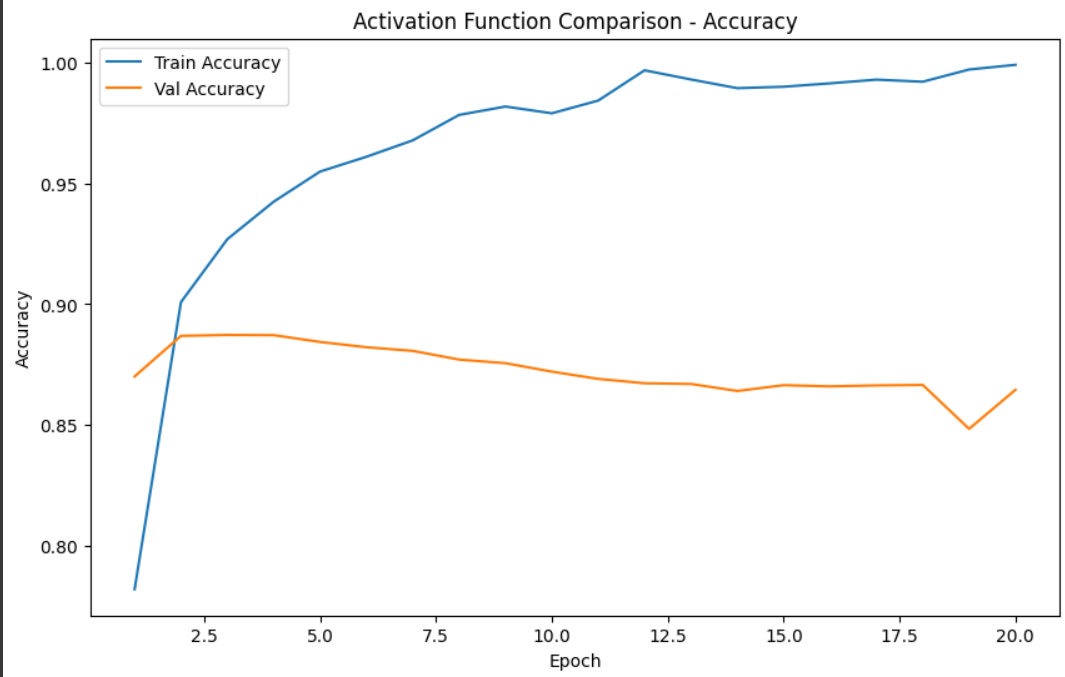
**Loss Function Analysis:**

The **training accuracy** increases steadily over epochs, but the **validation accuracy** remains stagnant or slightly decreases, indicating **overfitting** when certain loss functions are used. Similarly, while **training loss** continues to decline, the **validation loss** shows little to no improvement, implying that the model is **memorizing the training data** instead of learning generalized patterns. These results highlight the importance of choosing a loss function that aligns with the task. Specifically, using **binary cross-entropy**, which is well-suited for binary classification problems, can help enhance **model generalization** and reduce overfitting.

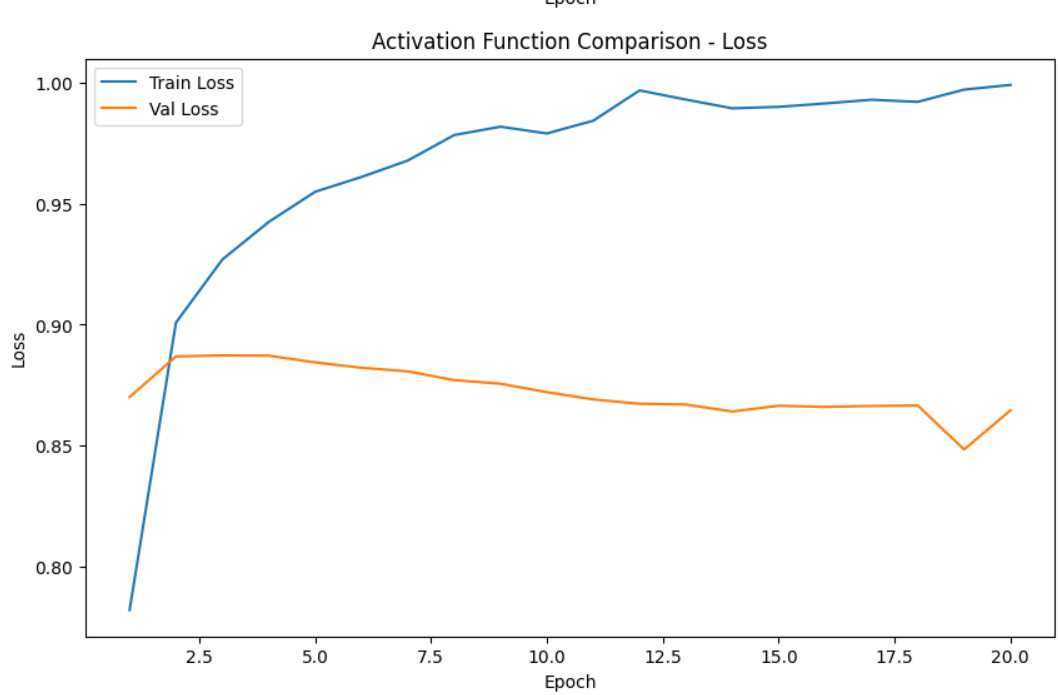
**Activation Function Comparison:**

* The model employing the **ReLU activation function** exhibits a **rapid increase in accuracy** during the initial training epochs and achieves a **higher final accuracy** overall.
* In contrast, the model using the **Tanh activation function** converges **more slowly** and demonstrates **less stable performance**, suggesting that ReLU provides better learning dynamics and efficiency for this sentiment classification task.

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



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



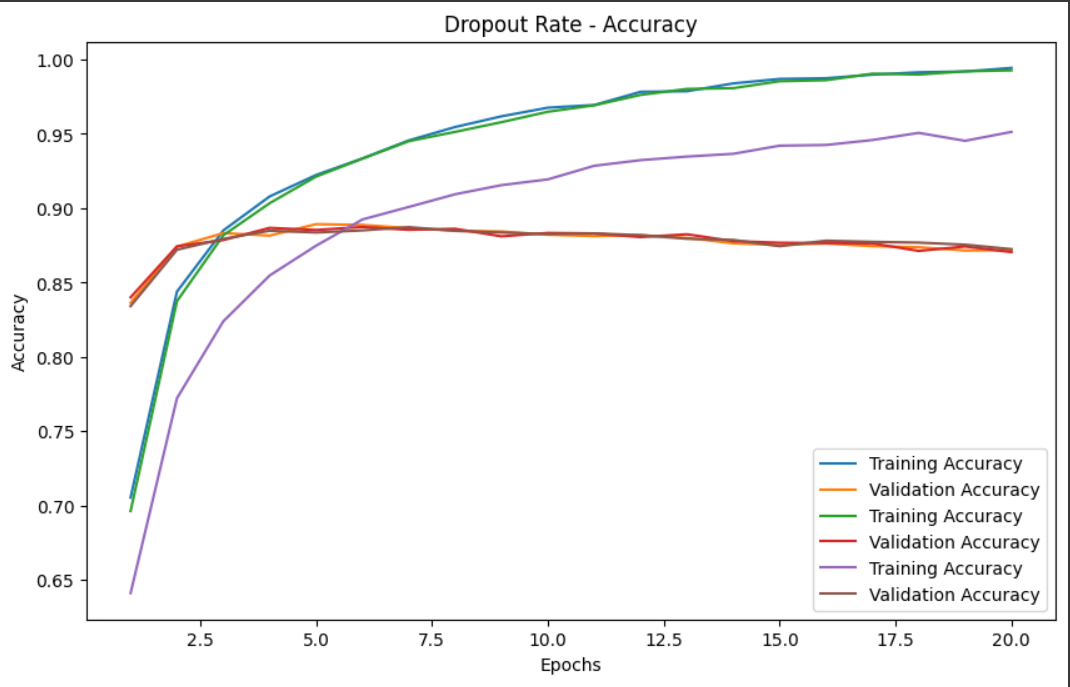
**Activation Function Analysis:**

The **training accuracy** increases notably across epochs, while the **validation accuracy** remains stagnant or shows a slight decline, indicating **overfitting** when certain activation functions are used. Similarly, **training loss** continues to decrease as the model learns the training data, but the **validation loss** stays relatively unchanged, suggesting poor generalization. These findings emphasize the importance of selecting an appropriate activation function. Specifically, the **ReLU activation function** proves more effective, as it helps mitigate overfitting and enhances the model’s overall performance and convergence speed.

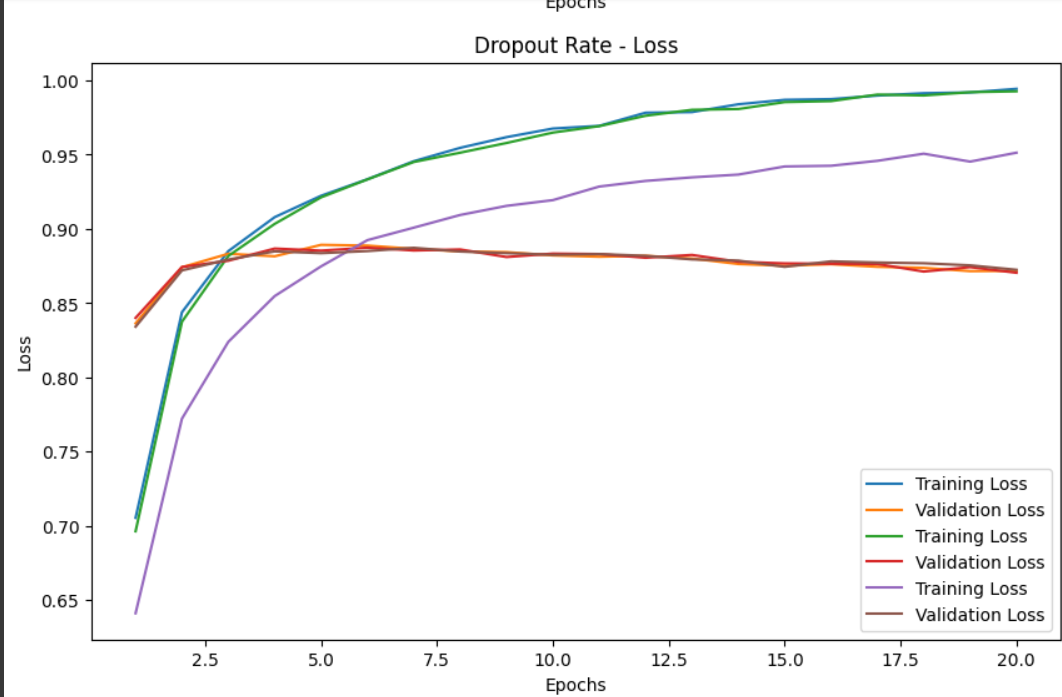
**Dropout Regularization – Accuracy:**

Implementing **dropout regularization** effectively combats overfitting by randomly deactivating neurons during training. This approach may slightly **reduce training accuracy**, but it helps maintain **stable and consistent validation accuracy**, leading to improved **generalization on unseen data**. By introducing controlled randomness, dropout ensures that the model learns more robust and representative patterns rather than memorizing specific training examples.

**Plot 9: Dropout Regularization – Accuracy:**



**Plot 10: Dropout Regularization – Loss:**



**Dropout Regularization Analysis**

The **training accuracy** continues to rise steadily over epochs, while the **validation accuracy** remains relatively stable, demonstrating that **dropout regularization** effectively mitigates overfitting. Likewise, the **training loss** consistently decreases, whereas the **validation loss** remains almost unchanged, indicating that an appropriate dropout rate enhances **generalization performance**. A **moderate dropout rate** provides an optimal balance between **model complexity** and **generalization**, preventing the model from memorizing the training data while maintaining reliable validation results.

**Conclusion:**

The **optimal neural network configuration** consists of a **single hidden layer with 256 units**, providing an ideal trade-off between model complexity and performance. The **Binary Crossentropy** loss function produced the most accurate and reliable results for this binary classification task, while the **ReLU activation function** improved learning efficiency and convergence. Incorporating a **dropout rate of 0.7** further enhanced generalization by reducing overfitting tendencies.

Overall, this combination yielded a **test loss of 0.4936** and a **test accuracy of 0.8619**, establishing it as the **best-performing model** for the IMDB sentiment classification dataset, effectively balancing accuracy, efficiency, and robustness.