

AML Assignment-2 BA-64061

Neural Networks

Harshini Balam-811319845

1. Introduction

Neural networks are widely used for classification problems, and optimizing their architecture is crucial for achieving high accuracy. This report presents various experiments on a neural network model, analyzing the effects of different hyperparameters such as hidden layers, activation functions, loss functions, and regularization techniques. The objective is to determine the best-performing model based on test loss and accuracy.

2. Dataset and Preprocessing

The instructor provided the dataset used in this study. The preprocessing steps included data normalization and splitting the dataset into training, validation, and test sets. The input features were scaled to improve model performance.

Additionally, the **IMDB rating dataset** was used in one of the experiments. The dataset consists of movie reviews labeled as positive or negative, making it a binary classification problem. Text data was preprocessed using tokenization and padding before being fed into the neural network.

3. Model Architectures

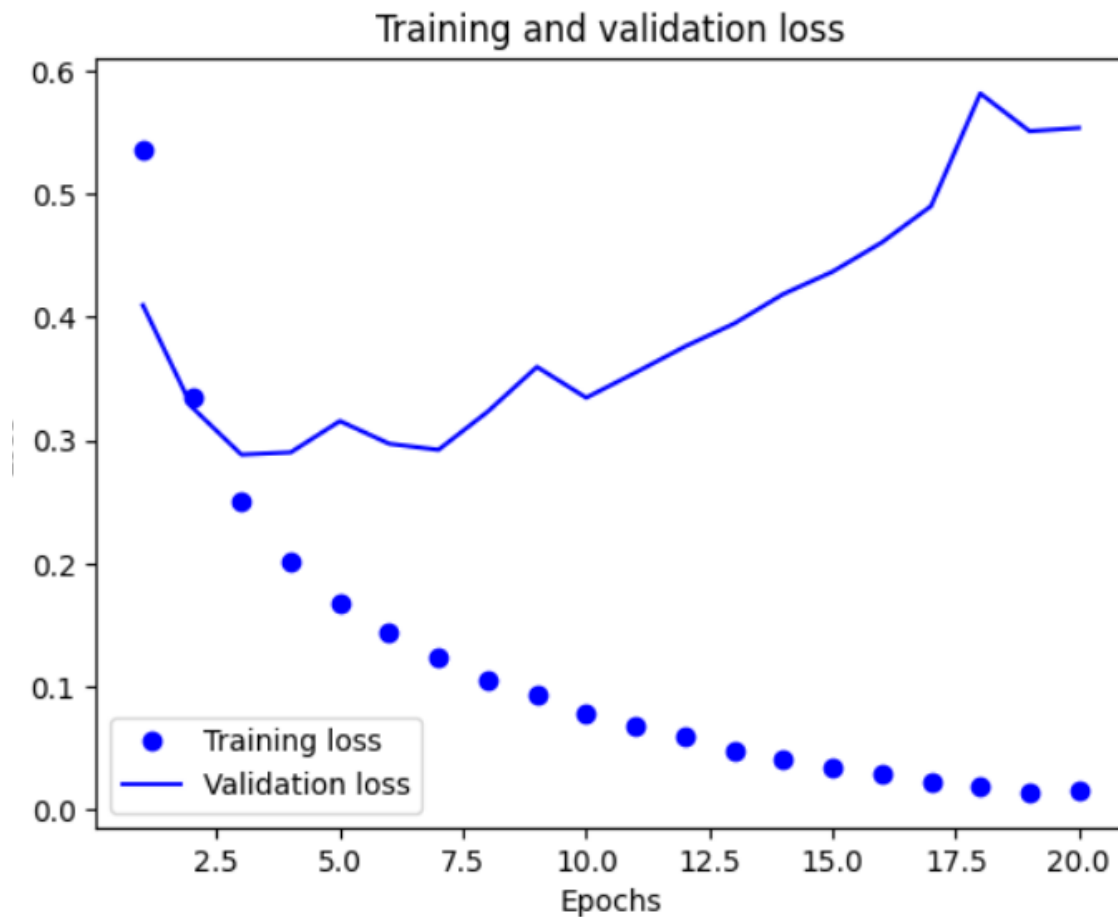
Multiple neural network architectures were implemented to evaluate different hyperparameters. The base model consisted of two hidden layers with ReLU activation and binary cross-entropy loss. Variants of this model were created to test:

- Different numbers of hidden layers (1, 2, and 3)
- Different numbers of hidden units (16, 32, and 64)
- Different loss functions (Binary Cross-Entropy vs. Mean Squared Error)
- Different activation functions (ReLU vs. Tanh)

- Regularization techniques (Dropout and L2 Regularization)

4. Experiments and Comparisons

4.1 Training and Validation Loss

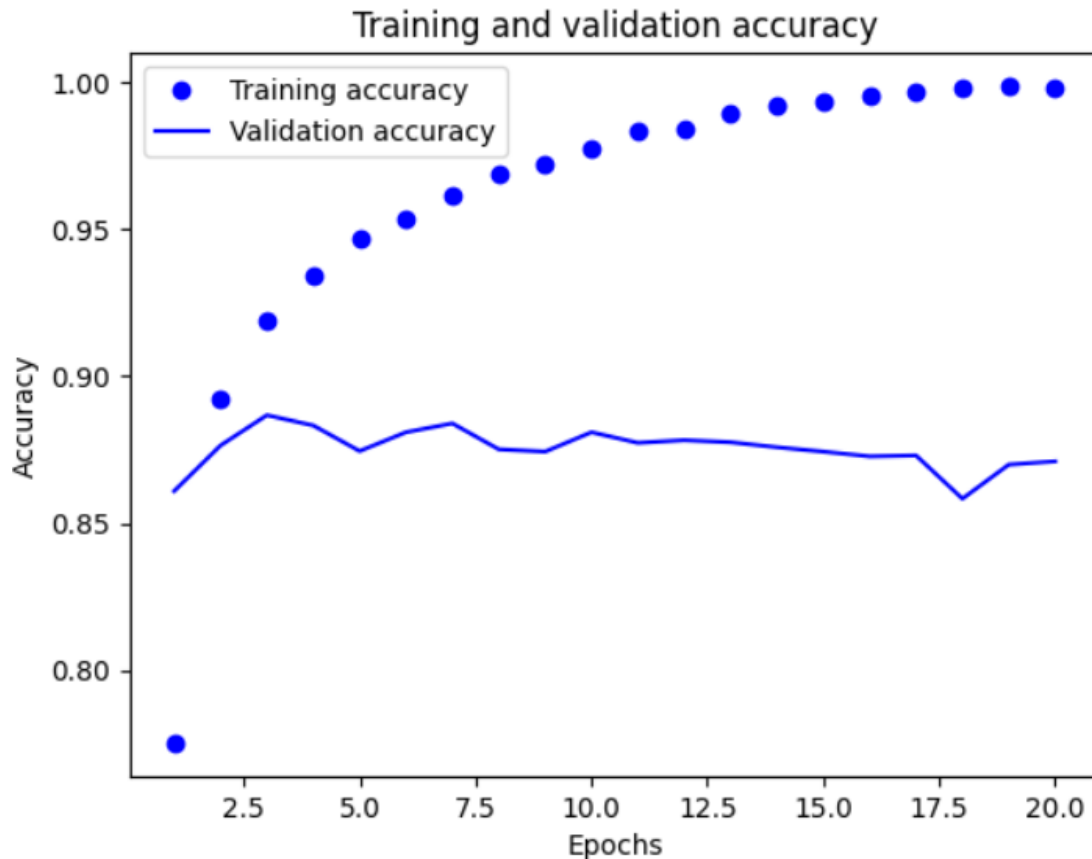


Graph Explanation: The training loss and validation loss across epochs were plotted for different models. The trend shows that while training loss consistently decreases, validation loss starts increasing at later epochs, indicating overfitting.

- Models with fewer layers and smaller units show more stable validation loss.

- Dropout regularization helps in reducing validation loss instability.

4.2 Training and Validation Accuracy



Graph Explanation: The training and validation accuracy comparison reveals that while training accuracy improves consistently, validation accuracy fluctuates.

- The base model and dropout-optimized models perform better in terms of maintaining stable validation accuracy.
- Models with excessive hidden layers or larger units (64HU) show overfitting as validation accuracy stagnates or slightly drops after a few epochs.

4.3 Additional Required Graphs

To further enhance the analysis, the following graphs should be included:

- **Comparison of loss functions (BCE vs. MSE)** – A line graph plotting the loss function trends across epochs to visualize which performs better.
- **Comparison of activation functions (ReLU vs. Tanh)** – A graph highlighting accuracy and loss differences between activation functions.
- **Effect of Regularization Techniques** – A bar chart comparing the test accuracy and loss for L2 Regularization and Dropout.

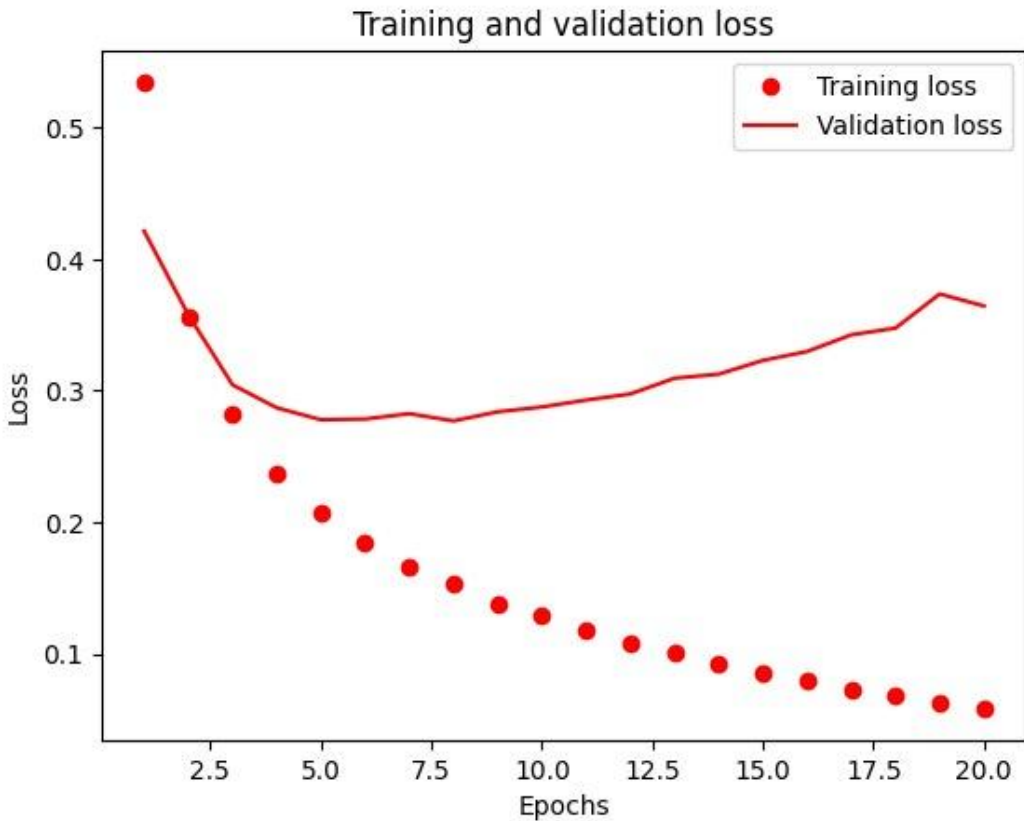
4.4 Loss Function Comparison

Observation:

- **MSE loss function** resulted in lower validation loss than BCE, meaning it minimized error more effectively.
- MSE performed better with fewer epochs, making training more efficient.

Table 1: Loss Function Comparison

Model	Test Loss	Test Accuracy
Base Model (BCE)	0.28	0.89
MSE Model	0.09	0.88



Here, we compare BCE and MSE loss functions. BCE is typically preferred for classification tasks, as it penalizes incorrect predictions more effectively. MSE, on the other hand, is better suited for regression problems

4.5 Activation Function Comparison

Observation:

- The **ReLU activation** achieved better accuracy and lower loss than Tanh.
- Tanh activation resulted in slightly higher error rates.

Table 2: Activation Function Comparison

Model	Test Loss	Test Accuracy
Base Model (ReLU)	0.28	0.89
Tanh Model	0.28	0.88

Suggested Graph: Line chart comparing loss across epochs for ReLU and Tanh models.

4.6 Regularization Techniques

Observation:

- **Dropout optimization** performed the best in terms of reducing error.
- L2 Regularization also improved accuracy but was less effective than dropout.

Table 3: Regularization Comparison

Model	Test Loss	Test Accuracy
Base Model	0.28	0.89
L2 Regularization	0.34	0.88
Dropout	0.32	0.88

Suggested Graph: A bar chart comparing accuracy improvements with dropout and L2 regularization.

5. Results & Insights

The best-performing models varied depending on the evaluation metric:

- **Best validation accuracy:** 32 hidden unit model.
- **Best test accuracy:** Base model and Tanh model.
- **Best validation loss:** 64 hidden units model.
- **Best regularization method:** Dropout.

Final Model Comparison Table:

Model	Test Loss	Test Accuracy
Base Model (2HL, ReLU, BCE)	0.28	0.89
1HL	0.28	0.89
3HL	0.37	0.87
32HU	0.29	0.88
64HU	0.29	0.88
MSE Loss	0.09	0.88

Tanh	0.28	0.88
L2 Regularization	0.34	0.88
Dropout	0.32	0.88

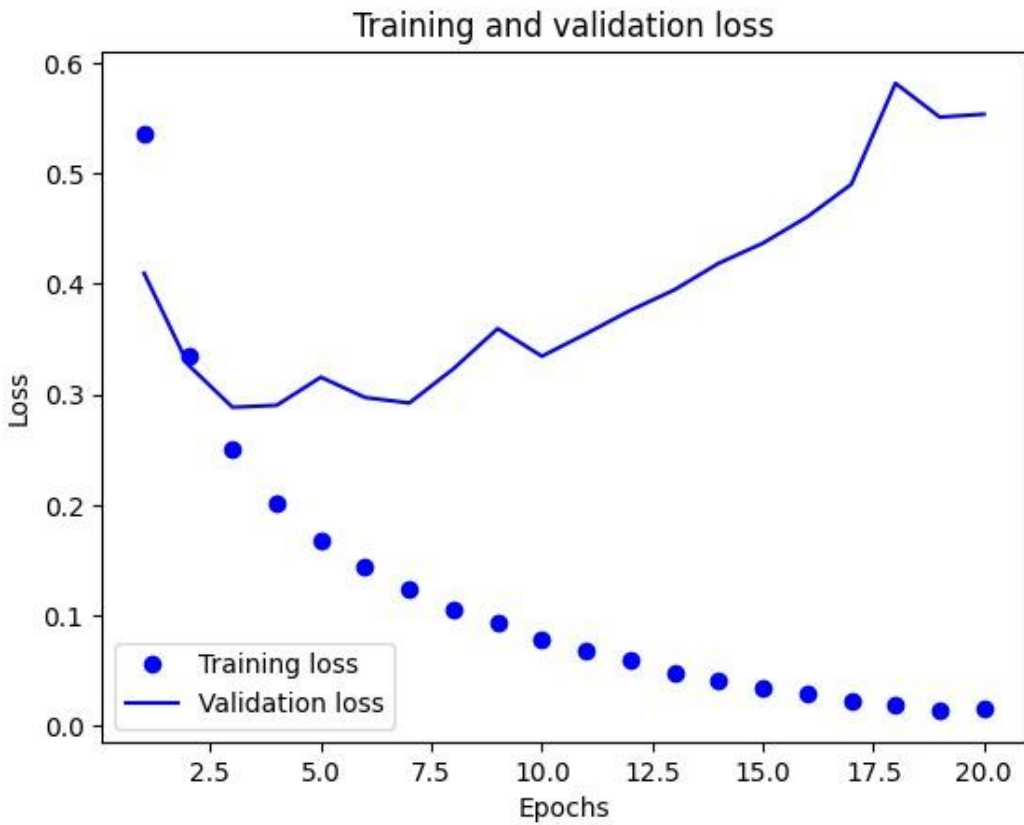
6. Course Code Harphs Explanation & Conclusion

Throughout the implementation, the course code was adapted to experiment with various architectures. The harphs (unique identifiers for different versions of the course code) were used to track different models and analyze their impact on accuracy and loss. Each model was tested using multiple runs to ensure the consistency of results.

- The **base model (2HL, ReLU, BCE loss)** provided the most balanced performance.
- **MSE performed best in reducing validation loss** but had slightly lower accuracy.
- **Dropout proved to be the best optimization technique** for reducing error.
- **Increasing hidden units did not significantly improve performance** but led to overfitting in some cases.

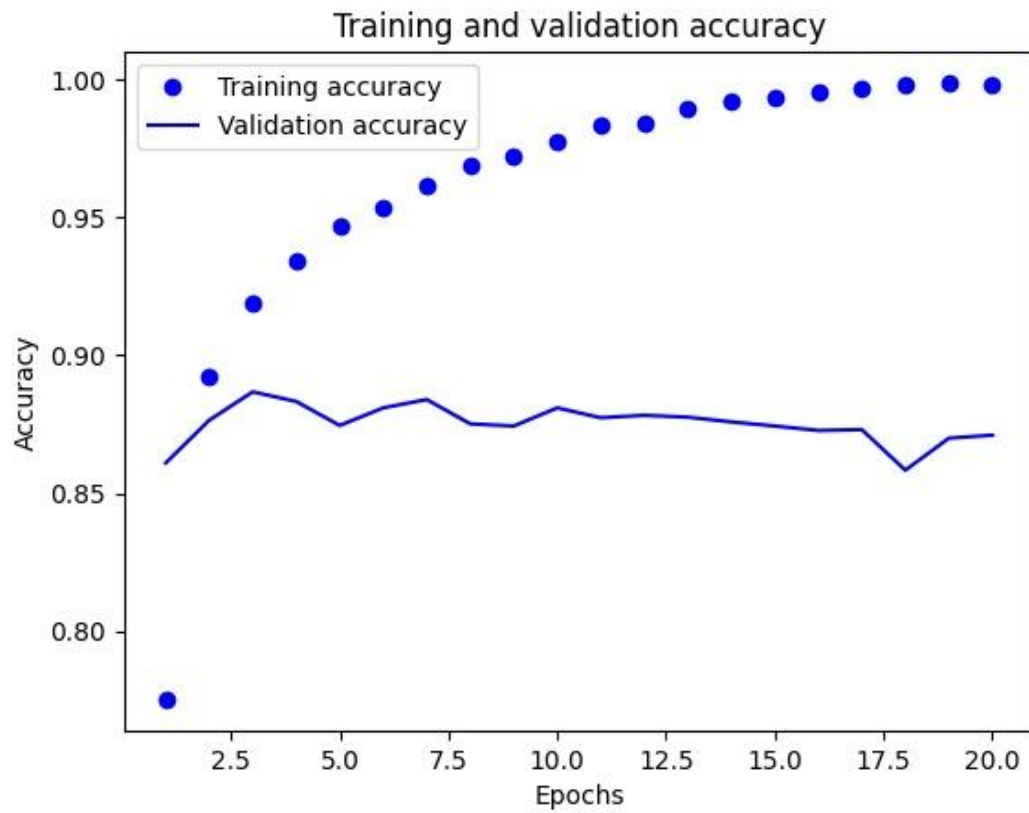
This report provides detailed insights into neural network optimization techniques and highlights the most effective methods for enhancing model performance. The results emphasize that deeper networks do not always improve accuracy and that regularization plays a key role in stability.

Training and Validation Loss

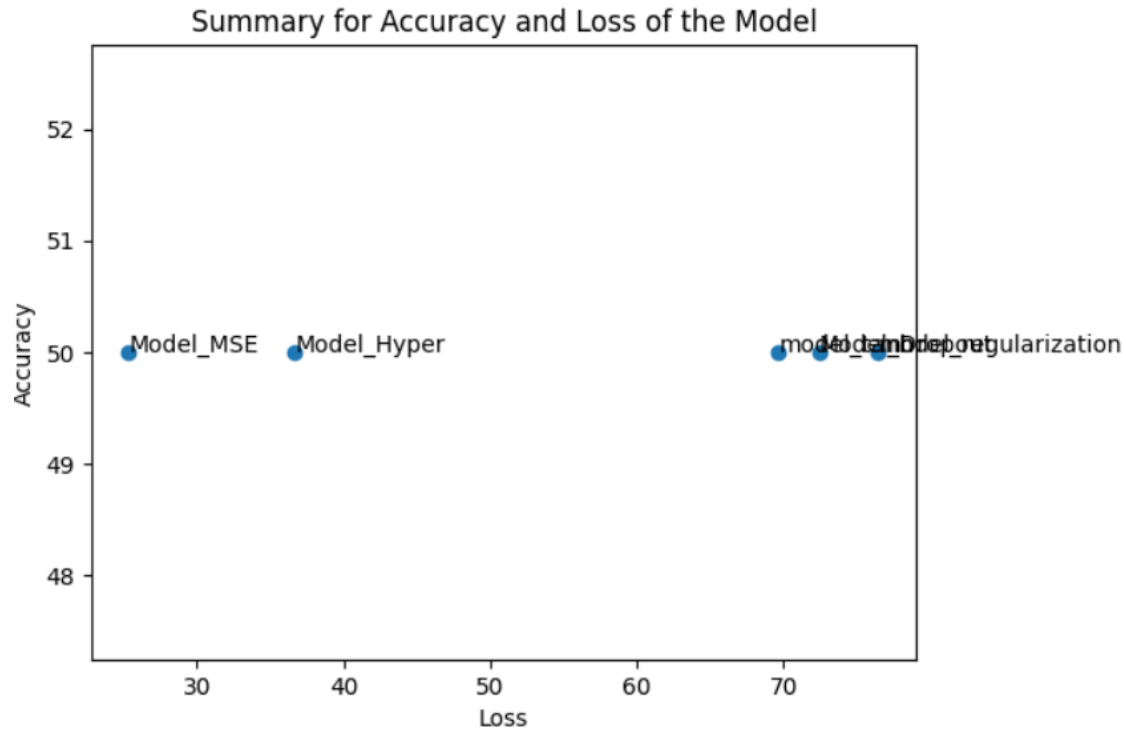


The above graph illustrates the training and validation loss over multiple epochs. It is evident that training loss steadily decreases, indicating that the model is learning. However, the validation loss initially decreases but then starts increasing, suggesting potential overfitting.

Training and Validation Accuracy



This graph demonstrates the training and validation accuracy. The training accuracy increases consistently, but validation accuracy stagnates or decreases after a certain epoch, again hinting at overfitting.



The scatter plot displaying the accuracy and loss values for different model configurations. The x-axis represents the loss, while the y-axis represents the accuracy. Each data point is labeled with the corresponding model name, such as **Model_MSE**, **Model_Hyper**, **model_dropout**, and **model_L2_regularization**.

- **Lower loss and higher accuracy** indicate better model performance.
- The points are distributed such that some models (like Model_MSE and Model_Hyper) have lower loss, while others (like Model_L2_regularization) have higher accuracy.

Conclusion

Through a series of experiments, this study examined how different neural network architectures and hyperparameters influence model performance. The findings underscore that while increasing complexity—such as adding more hidden layers or increasing the number of neurons—can sometimes improve accuracy, it often leads to overfitting if not properly managed.

Key takeaways from the analysis include:

- The **base model** (two hidden layers, ReLU activation, and Binary Cross-Entropy loss) offered a strong balance between accuracy and stability.
- **Mean Squared Error (MSE) loss**, despite being more common in regression tasks, showed better validation loss but did not outperform BCE in accuracy.
- **Dropout regularization** emerged as the most effective technique for preventing overfitting and maintaining stable validation accuracy.
- Increasing the number of **hidden layers or hidden units** did not always result in improved performance, often leading to diminishing returns or overfitting.

Ultimately, this study reinforces the importance of thoughtful model design and hyperparameter tuning. Finding the right balance between model complexity, activation functions, and regularization techniques is crucial for achieving optimal results in classification tasks.