

# NEURAL NETWORKS

**Name:**

**Student ID:**

**GitHub:**

## **Introduction:**

In this assignment, I have tried several hyperparameters of a neural network that have been trained on an IMDB sentiment classification task. It has been about observing with the number of layers, units in each layer, activation functions, and loss functions upon the performance of a model. This experiment is regarding training various models and comparing their accuracy over validation and test sets.

## **Model Information:**

I used the **Keras Sequential API** to build and experiment with different neural network configurations. Each experiment focused on one or more of the following hyperparameters:

Number of Hidden Layers: I varied the number of hidden layers between 1, 2, and 3.

Units per Layer: I tested models with 32 and 64 units per layer.

Activation Functions: I compared the performance of models using “Tanh” activation function.

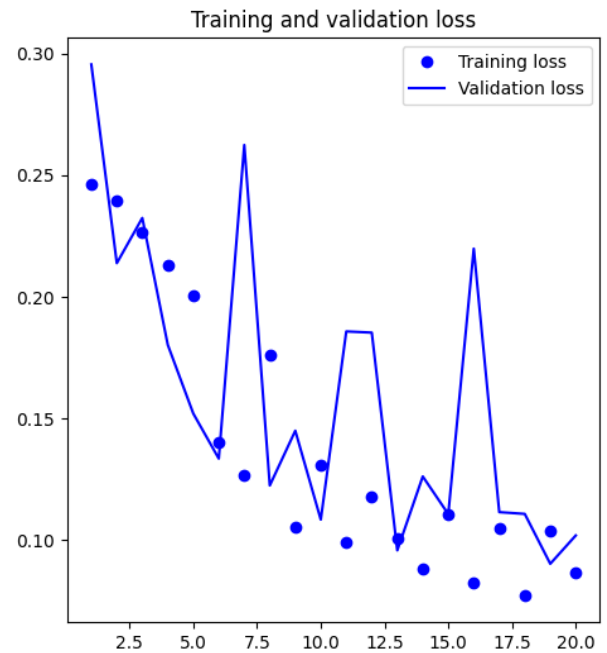
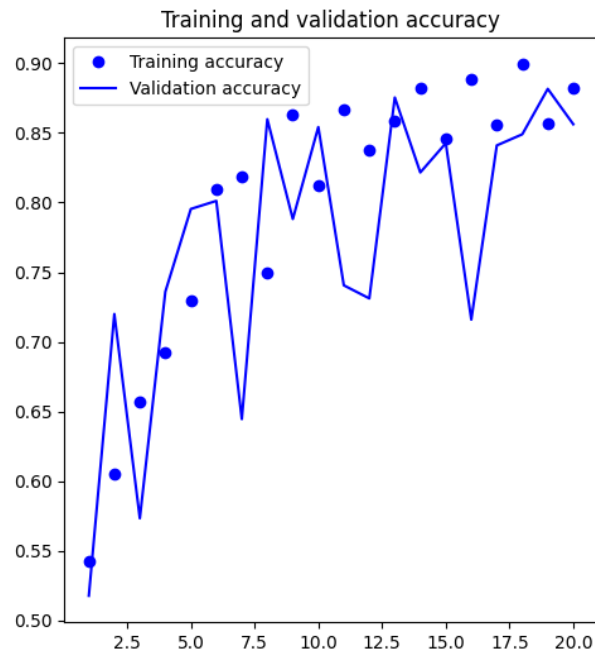
Loss Functions: I tested with the “mse” instead of “binary\_crossentropy”.

## Results:

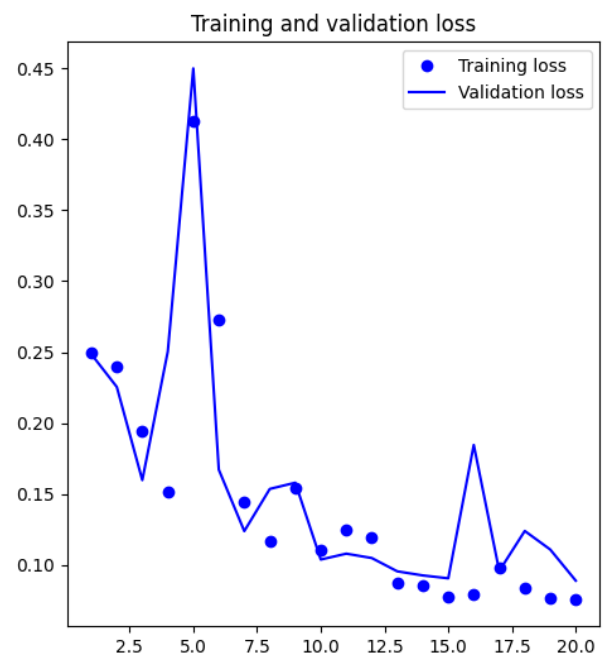
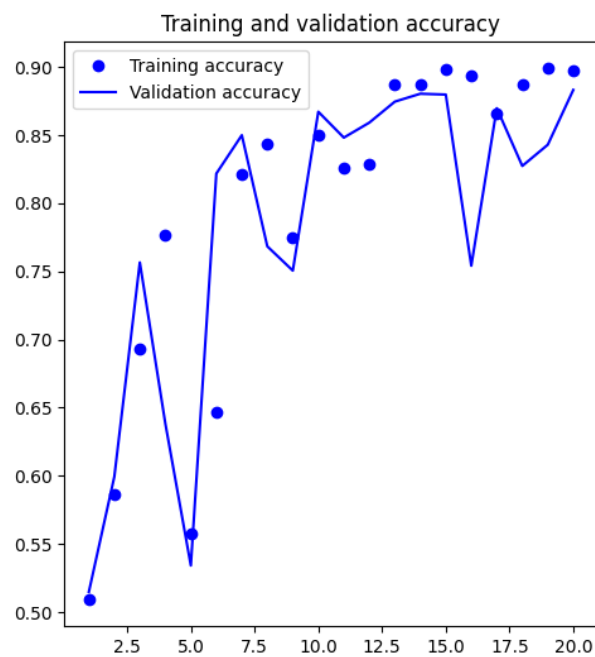
Model Configuration	Hidden Layers	Units	Activation Function	Loss Function	Test Accuracy
One Hidden layer	1	64	tanh	mse	85.68
One Hidden layer	1	32	tanh	mse	88.095
Three Hidden Layers	3	64	tanh	mse	85.71
Three Hidden Layers	3	32	tanh	mse	87.80

## Graphs:

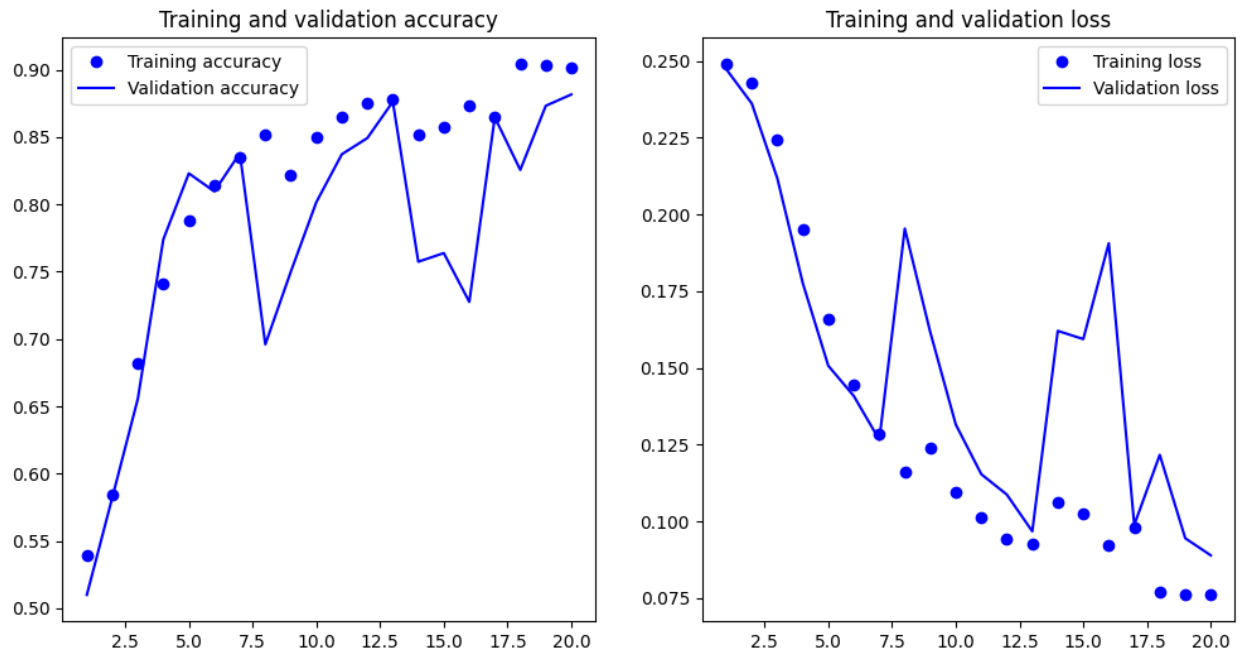
Three Hidden layers and 64 units tanh Activation function and mse loss function



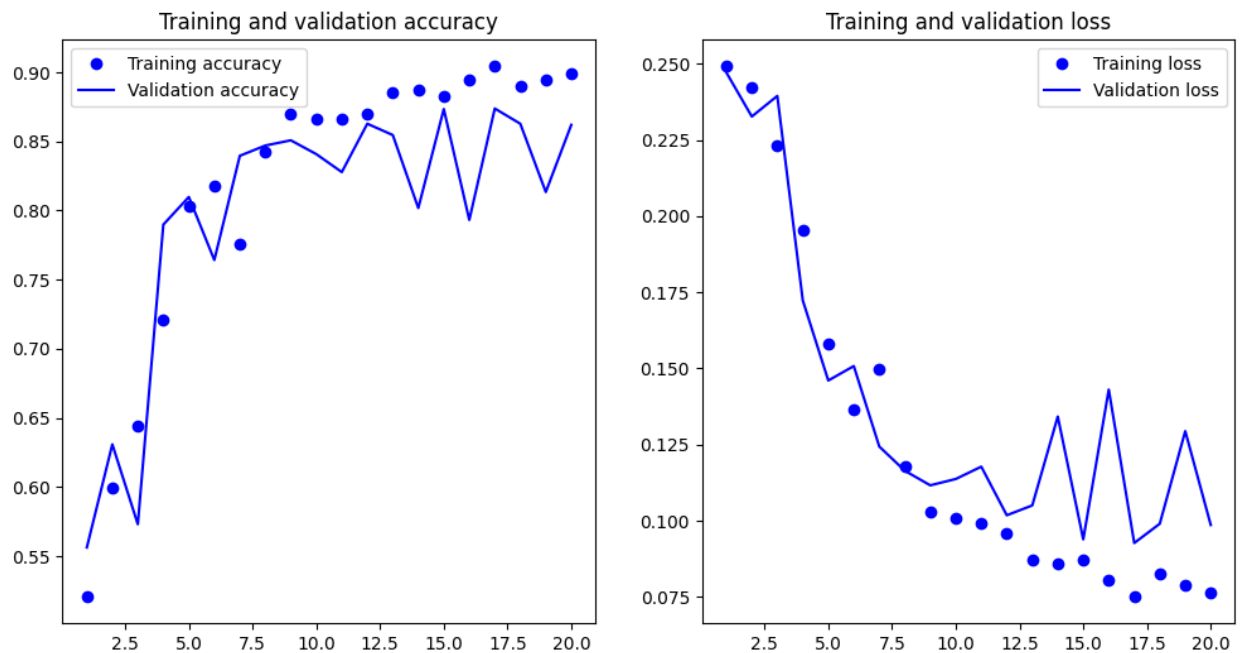
Three Hidden Layers and 32 units tanh Activation function and mse loss function



## One Layer 32 units tanh Activation function and mse loss function



## One Layer and 64 units tanh Activation function and mse loss function



## Summary :

In this assignment, various neural network configurations were experimented with to classify sentiments from the IMDB dataset. The models were built using the Keras Sequential API, with a focus on tuning key hyperparameters, including the number of hidden layers, units per layer, activation functions, and loss functions. The aim was to analyze how changes in these hyperparameters impacted the model's accuracy on validation and test sets.

### Hyperparameter Configurations:

Number of Hidden Layers: Varied between 1 and 3.

Units per Layer: Tested models with 32 and 64 units per layer.

Activation Function: The Tanh function was employed.

Loss Function: Mean Squared Error (MSE) was used instead of the typical `binary_crossentropy` for this classification task.

### Key Results:

The most accurate model featured one hidden layer with 32 units, Tanh activation, and MSE as the loss function, achieving a test accuracy of 88.1%.

Increasing the model complexity (e.g., using three hidden layers) did not significantly enhance accuracy, with the best result for such a configuration reaching 87.8%.

The findings suggest that simpler models can perform comparably, or even better, than more complex architectures for this specific task. The experiment illustrates the importance of fine-tuning hyperparameters to achieve optimal performance.