

Neural Network Summary

Lakkoju Sai Venkata Dharani Poojitha

811263845

Overview: The dataset used in this study comprises 50,000 IMDb movie reviews, of which 50% are rated as "positive" or "good" and the other 50% as "negative." By using several approaches to the IMDb dataset, the study aims to improve the performance of the neural network model. Numerous changes will be made to the current neural network model, including adjustments to the number of hidden layers, activation function, loss function, units, and regularization strategies like dropout. The ensuing outcomes will be carefully scrutinized.

Goal and Method: Improving the neural network model iteratively is the main goal. This entails adding regularization strategies like dropout in addition to changing crucial parameters like the number of units, activation function, loss function, and hidden layers. The study employs a structured methodology to assess how these modifications impact the model's capacity for prediction.

Data Processing Techniques: This investigation requires robust data processing procedures. They include organizing, processing, modifying, computing, and analyzing data. These procedures are required in order to guarantee the effectiveness of later model training and evaluation, as well as to extract significant patterns and insights from the IMDb dataset the shorter ones with zeros to ensure that every review is the same length. As a result, each review was represented as a fixed-length vector with each element denoting a dictionary word's index.

We were allowed to select the maximum word count and duration for each review after the data import. Next, we constructed a simple neural network model with just 16 units for the hidden layer. Binary Cross Entropy, Mean Squared Error (MSE) as the loss function, ROI of the hyper-tuned hidden layer parameters, dropout, and the optimization techniques Adam, Regularization, and Tanh were the strategies we employed. Next, we looked at the previously recommended approaches in an attempt to improve the model's usefulness.

Next, we varied the number of hidden layers to build models with one, two, and three hidden layers. Using the test and training datasets, we compared, assessed, and trained the models. We discovered that adding three hidden layers increased test validity and accuracy when compared to utilizing just one hidden layer.

We used a variety of techniques, including the following, to guarantee test validity and accuracy:

Methods:

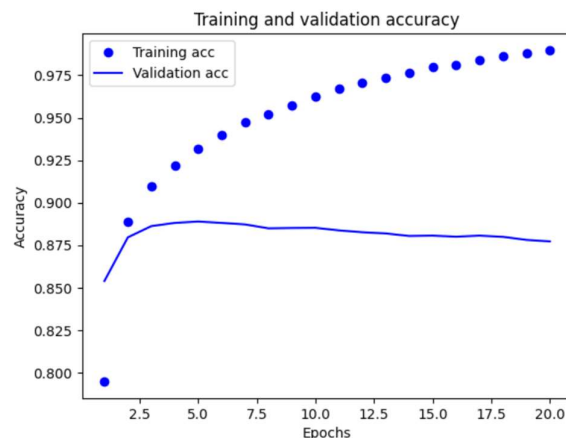
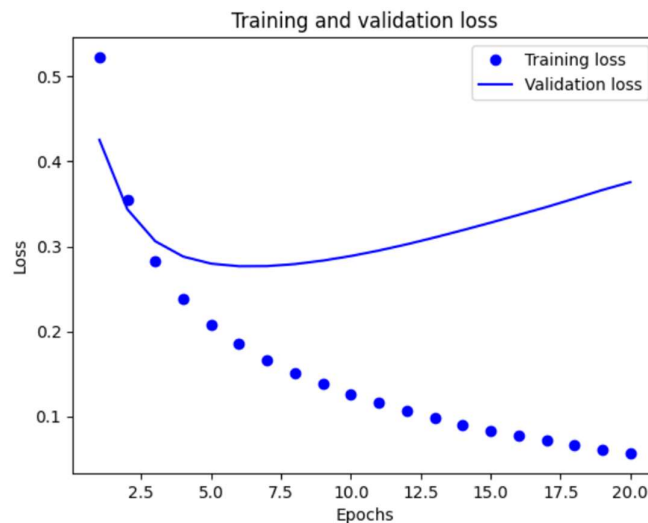
Following the import of the data, the length and maximum word count for each

review were determined. Next, we built a basic neural network model with a single 16-unit hidden layer. We used binary Cross entropy, MSE as the loss function, Adam, Regularization as the optimizer, and dropout and hypertuned as parameters for the hidden layer

The loss function is binary Cross entropy, with MSE, and regularization acting as the optimizer. Tanh and relu were the activation functions that we employed. In an effort to increase the model's usefulness, we then looked at the methods that had been previously mentioned. We then constructed models with one, two, and three hidden layers by varying the number of hidden layers. We evaluated, trained, and compared the models using the test and training datasets. We found that using three hidden layers was superior than using just one, as it increased test validity and accuracy.

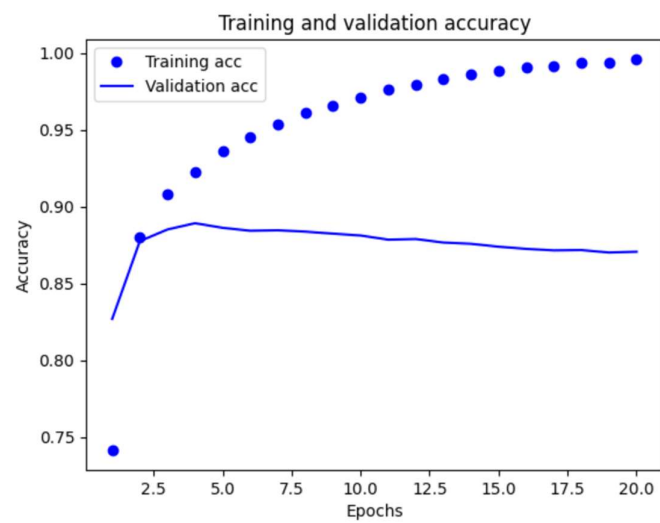
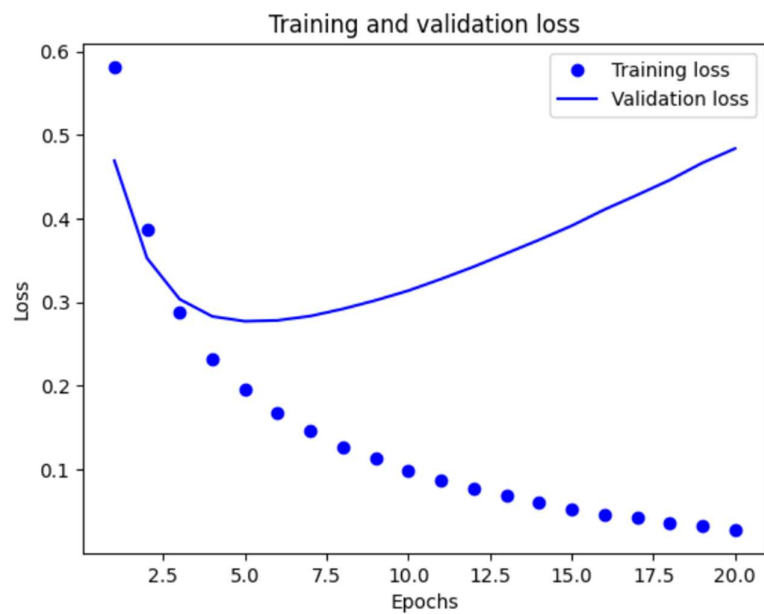
The many methods we employed for test accuracy and validation are listed below:

Neural network with - 1-hidden layer,16-units , loss= binary crossentropy,activation=relu



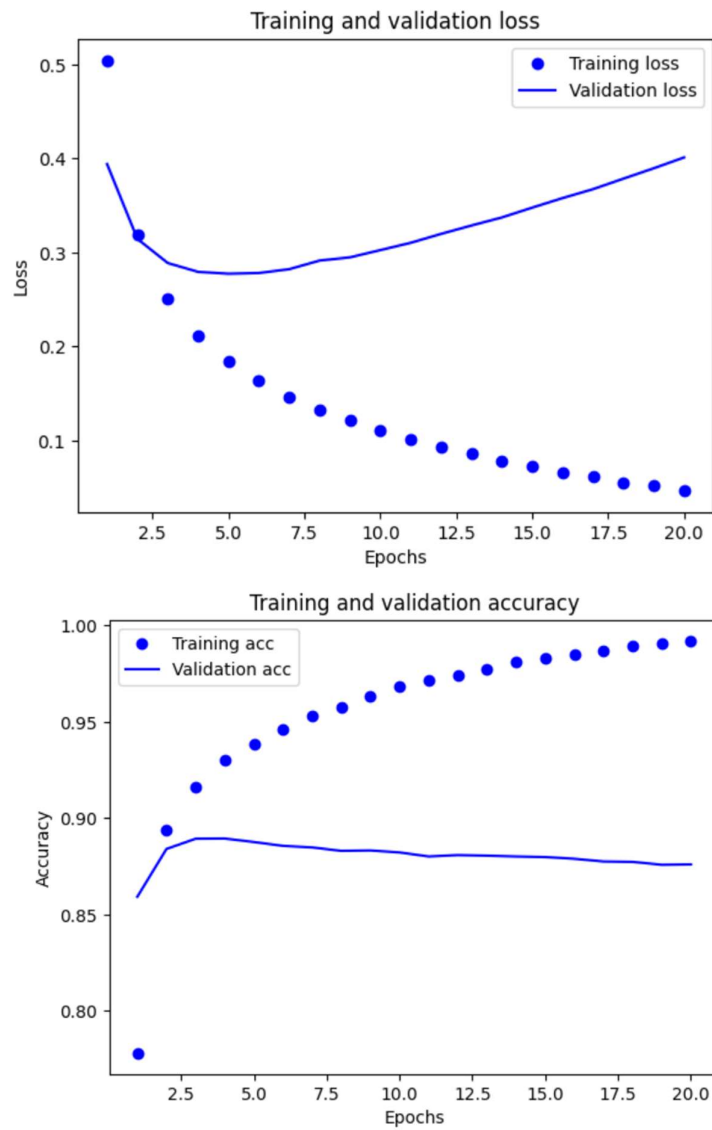
Accuracy:88.75%

Neural network with - 3-hidden layer,16-units , loss= binary crossentropy,activation=relu



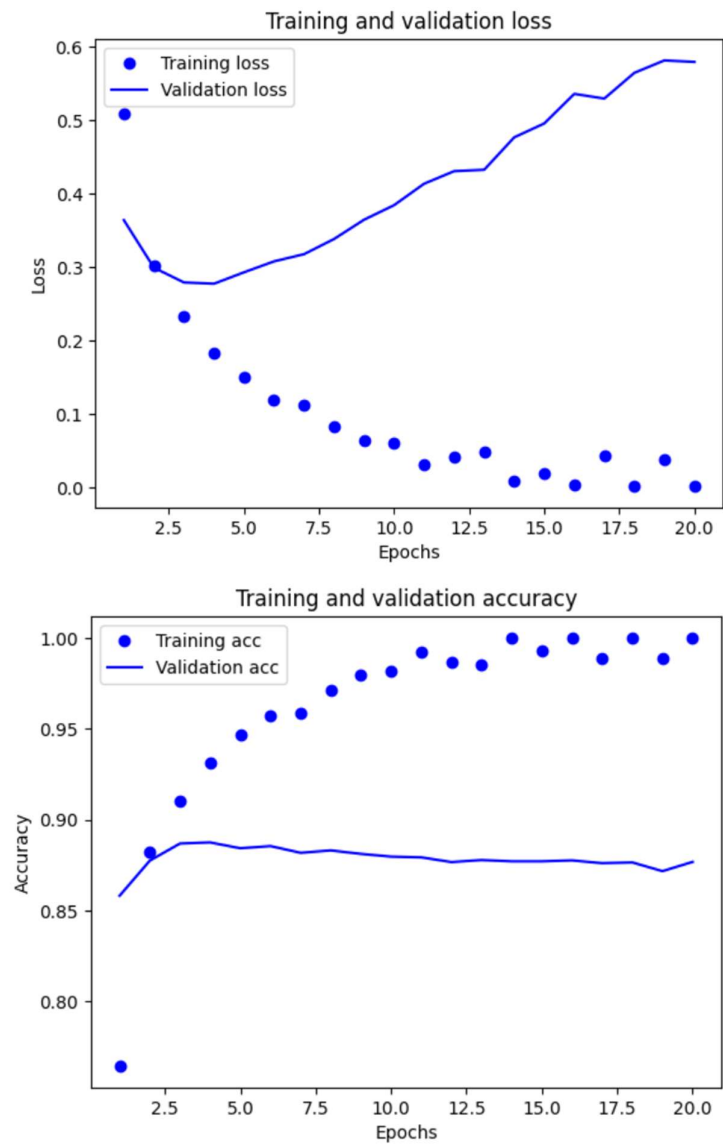
Accuracy:88.77%

Neural network with - 3-hidden layer,32-units , loss= binary crossentropy,activation=relu



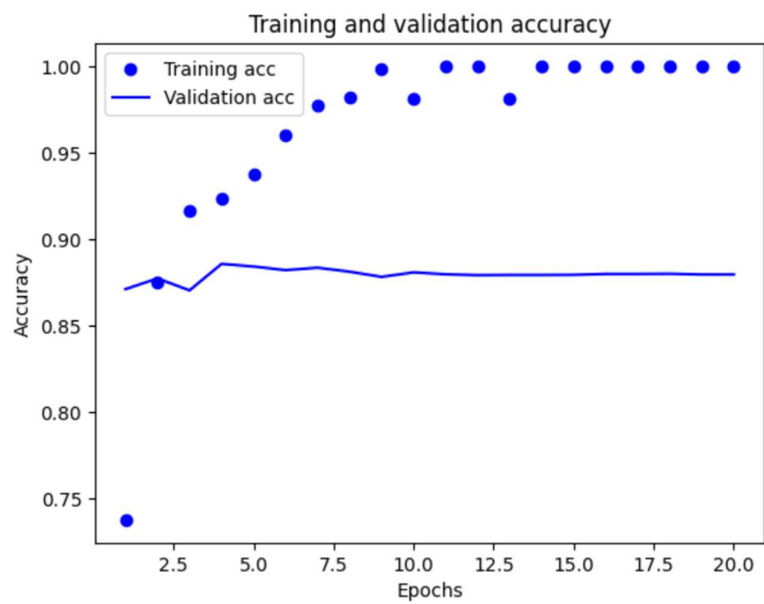
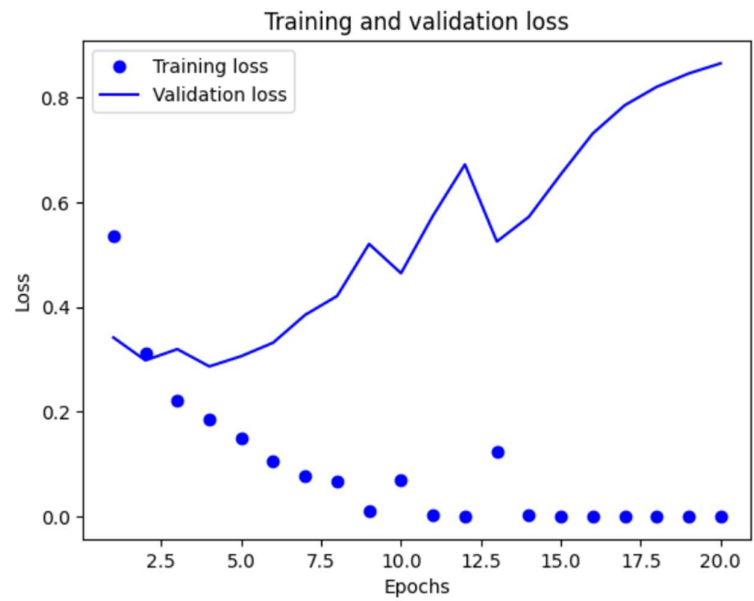
Accuracy: 86.70%

Neural network with - 2-hidden layer,64-units , loss= binarcrossentropy,activation=relu



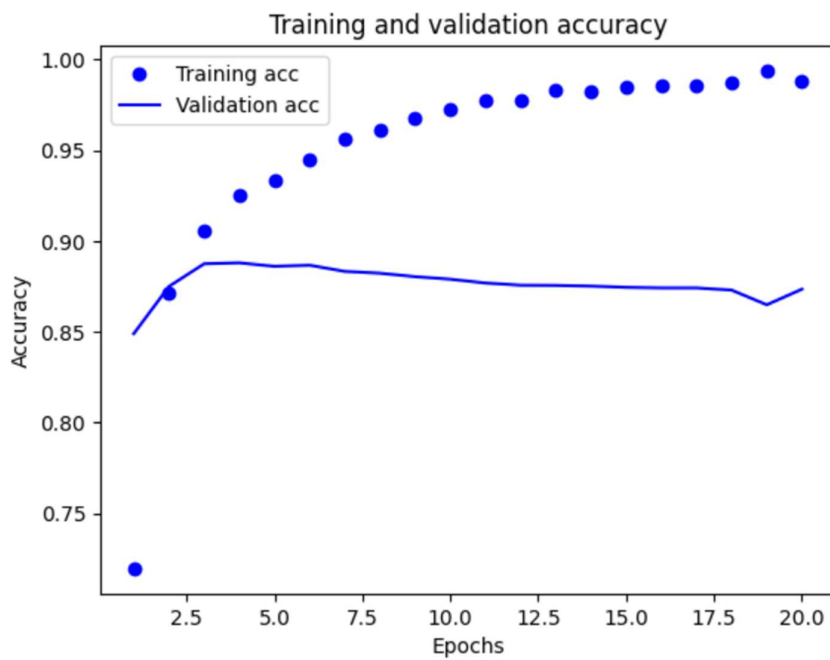
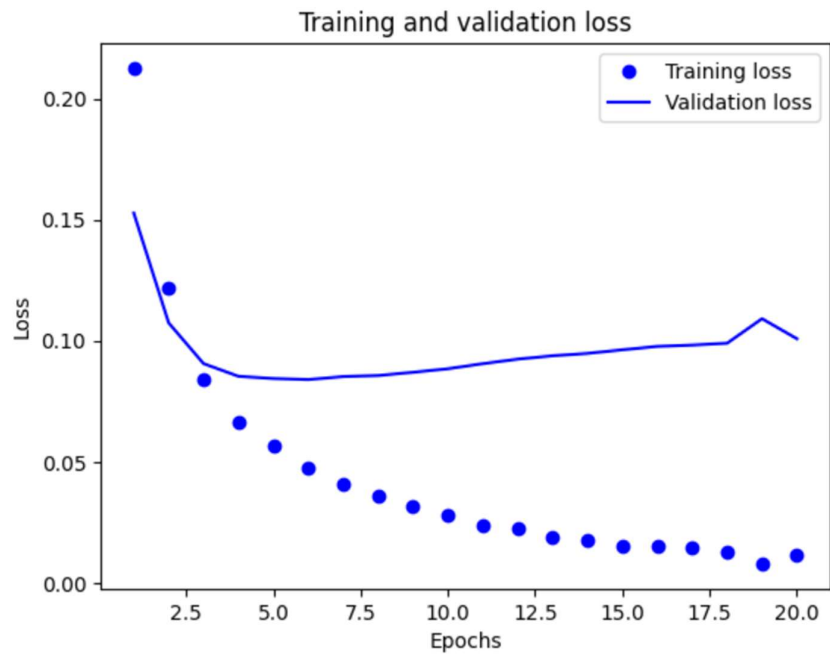
Accuracy: 84.89%

Neural network with - 3-hidden layer,128-units
,loss=binarcrossentropy,activation=relu



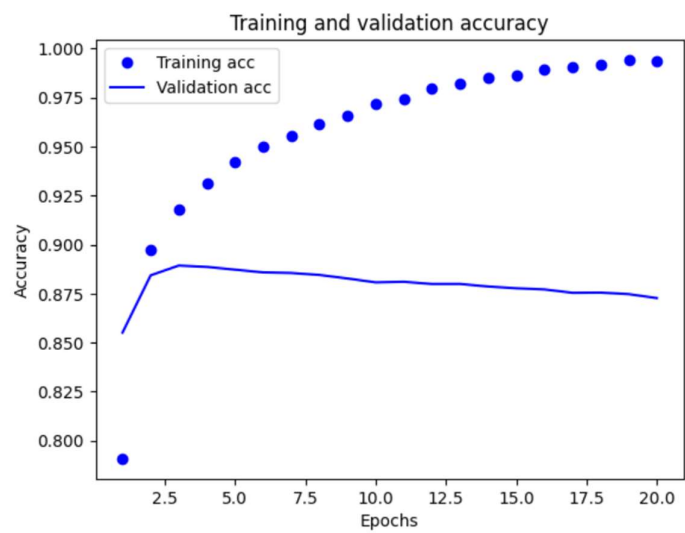
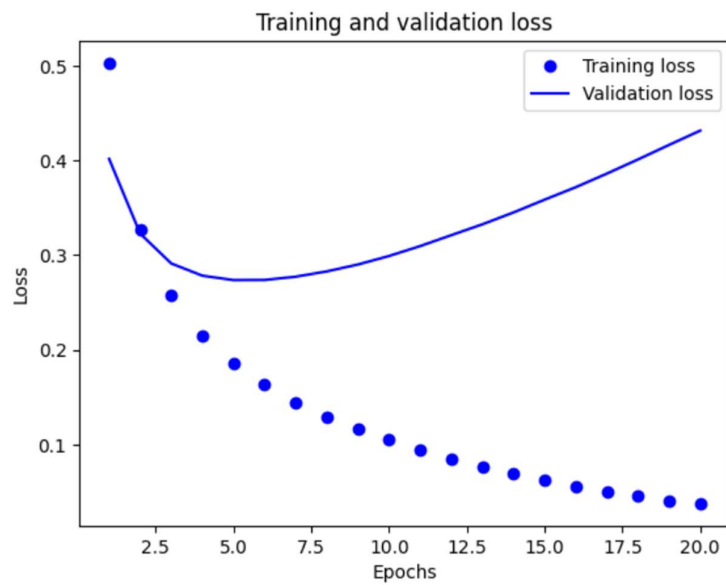
Accuracy: 82.44%

Neural network with - 3-hidden layer,16-units ,loss=MSE ,
activation=relu

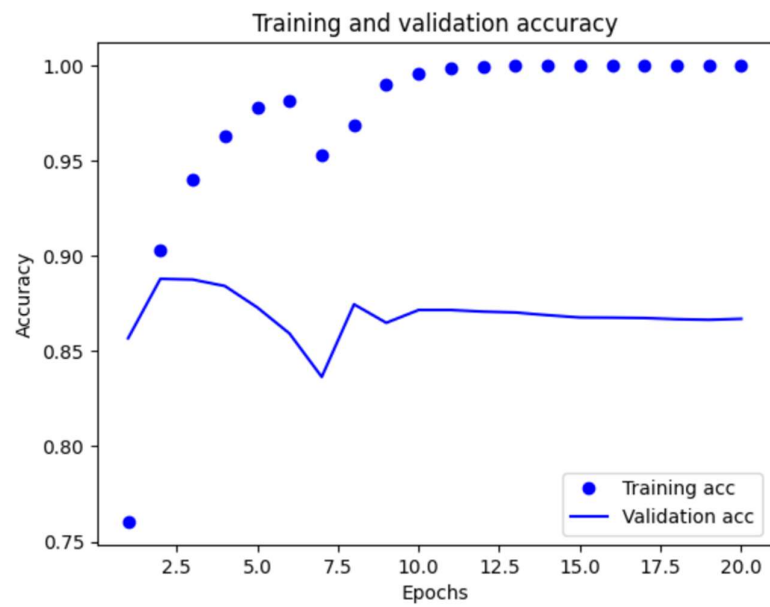
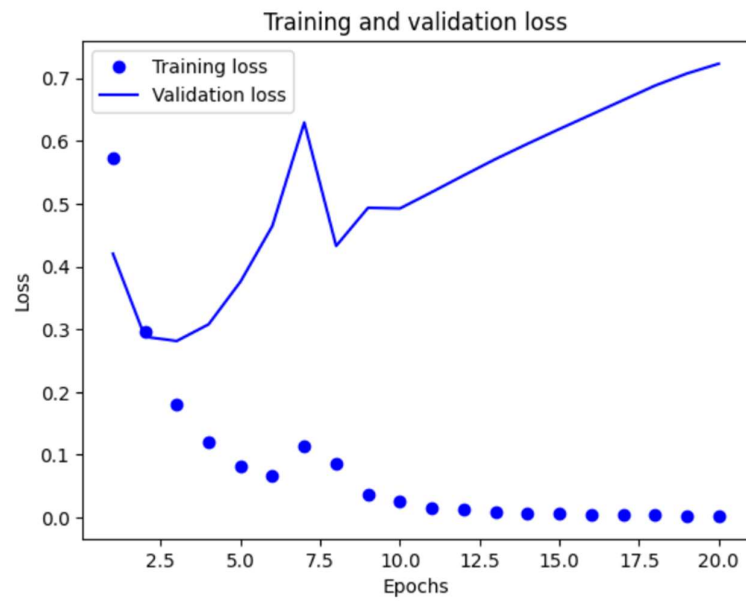


Accuracy: 86.68%

Neural network with - 1-hidden layer,16-units ,loss=MSE ,
activation=tanh

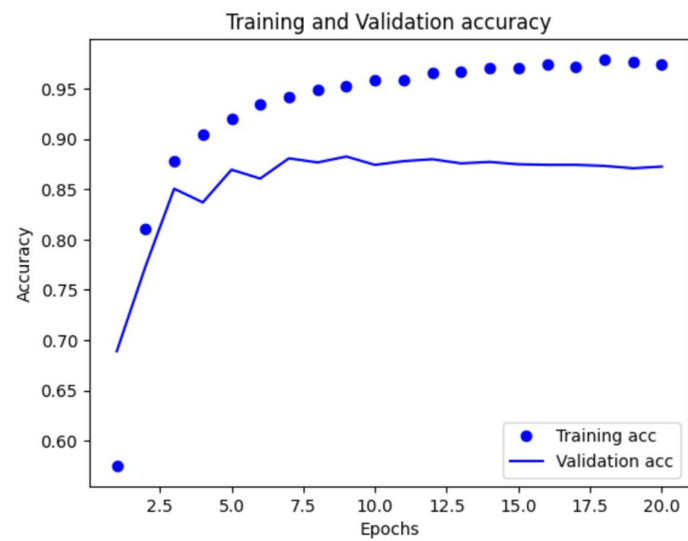
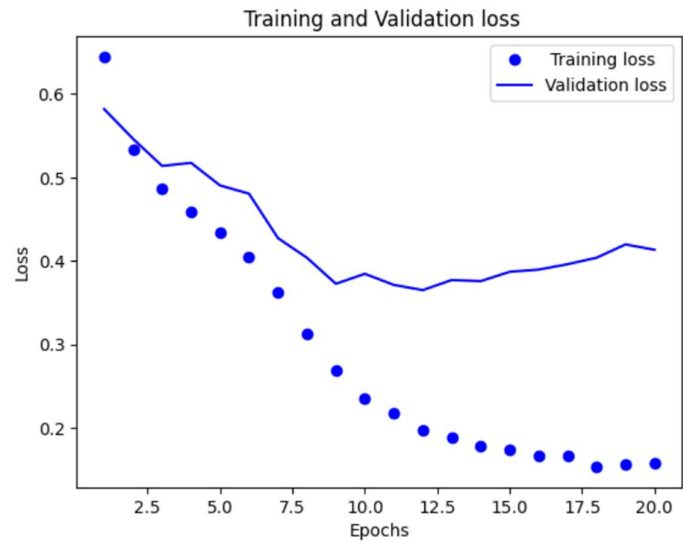


Accuracy:85.39%



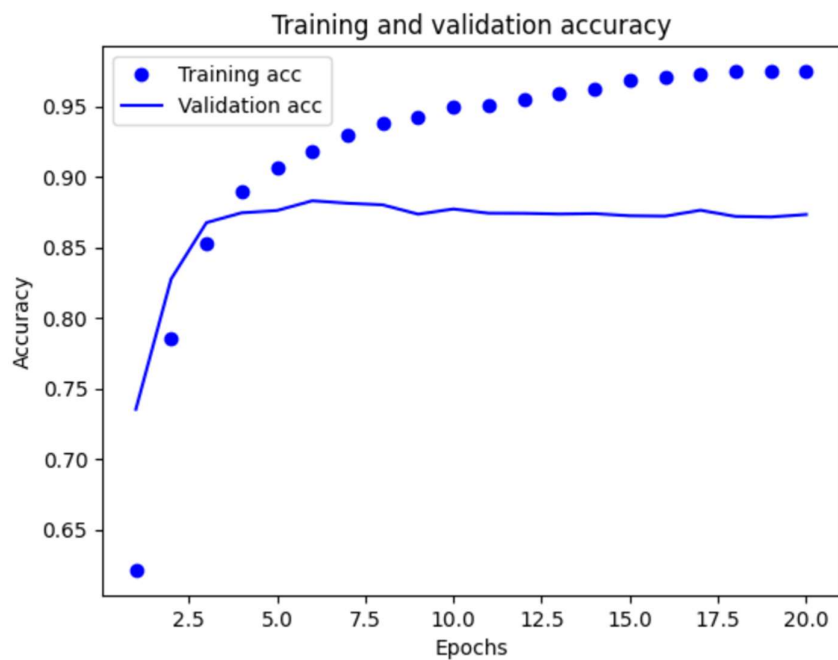
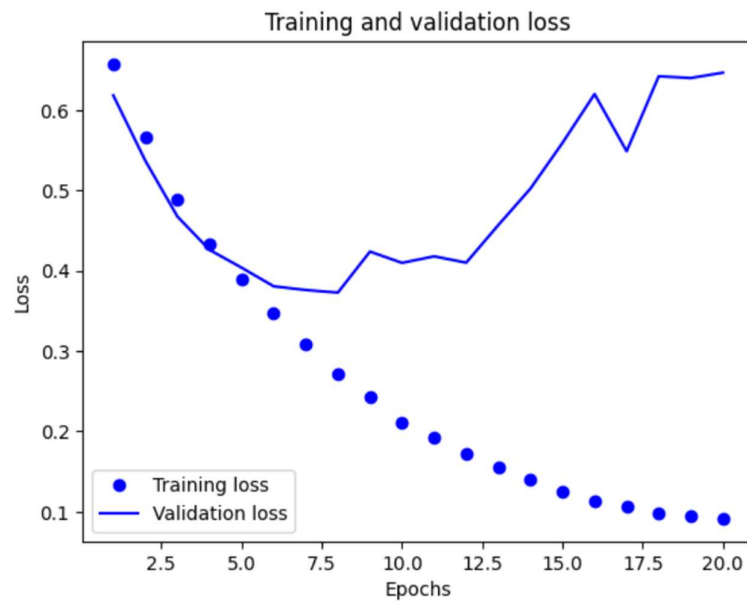
Accuracy: 84.58%

Neural network with - 2-hidden layer,16-units ,loss=binary Cross entropy , activation=relu, optimizer=rmsprop(regularization)



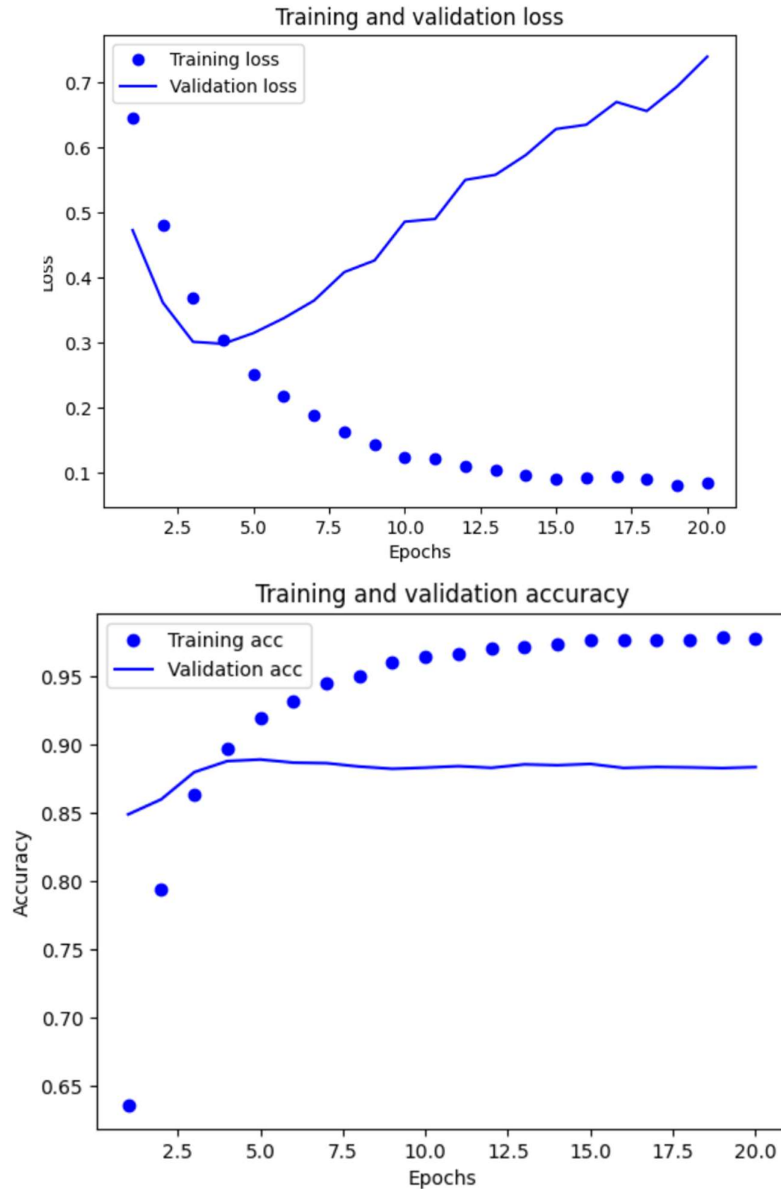
Accuracy: 86.90%

Neural network with - 3-hidden layer,16-units ,loss=binary Cross entropy , activation=relu,
optimizer=rmsprop(regularization),dropout=0.5



Accuracy: 86.70%

Neural network with - 3-hidden layer,32-units, loss=binary Cross entropy, activation=relu, optimizer=rmsprop(regularization), droupout=0.5, Hyper tuned parameters (kernel_regularizer=regularizers. l2(0.0001))



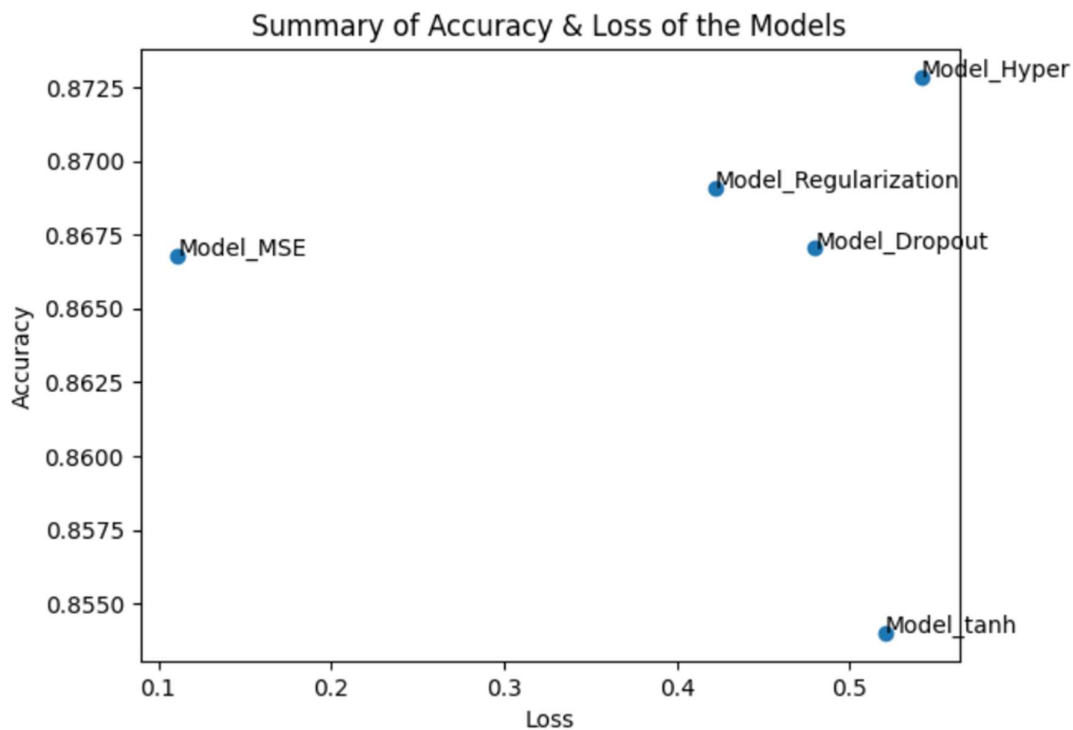
Accuracy: 87.28%

Conclusion:

In order to avoid overfitting, I tried dropout regularization at last. Using dropout layers, I created a new model containing training and test datasets. In comparison to the baseline model, I discovered that the validation accuracy was improved by employing dropout regularization.

It follows that various modifications to neural network models should result in a range of accuracy and loss functions. The Model Hyper yielded the best accuracy and loss, suggesting that the IMDB dataset would benefit from three thick layers with a dropout rate of 0.5.

The loss value of the MSE loss function is lower than that of binary cross-entropy. The precision of the tanh activation function is decreased by the vanishing gradient issue. It was demonstrated



Model Hyper is more accurate than Model MSE, which has the lowest loss value. In comparison to other models, the accuracy of the Model Regularization is weak.

As a result, we may conclude that the Model Hyper is the most successful of all the models that were looked at.