NEURAL NETWORKS SUMMARY
GEETHIKA VULLI
811290653

Introduction:

The IMDb review dataset includes 50,000 movie reviews, with 25,000 labeled "positive" or "good" and 25,000 labeled "negative". These labels reflect the reviewer's subjective assessment of the film and were determined by analyzing the general tone of the reviews. Many natural language processing activities, including sentiment analysis, have made extensive use of this dataset in academic as well as commercial settings. Using the IMDb dataset, this research aims to investigate several strategies for enhancing a neural network model's performance. We will make several changes to an existing neural network model, including changing the number of hidden layers, units, loss function, activation function, and regularization techniques like dropout, and then analyze the results.

Data Processing

We carried out a number of preprocessing operations on the IMDb review dataset in order to transform the unprocessed text data into a format that could be used to train a neural network. primarily since incorporating every word in the dataset would create an extremely high-dimensional input space, we only took into account the top 10,000 most frequent words in the dataset. Next, we used a dictionary to map the terms in the top 10,000-word list to their associated indices in order to turn the text reviews into integer representations.

We had to change the integer representations to tensors because neural networks cannot accept integers as input. In order to accomplish this, we truncated lengthier reviews and padded shorter ones with zeros to make sure all reviews were the same length. As a consequence, each review was represented as a fixed-length vector with each element denoting the dictionary word's index.

Lastly, we transformed the integer representations into binary values by one-hot encoding. As a result, the data were represented as a binary matrix, where each row represented a review and each column a specific dictionary word.

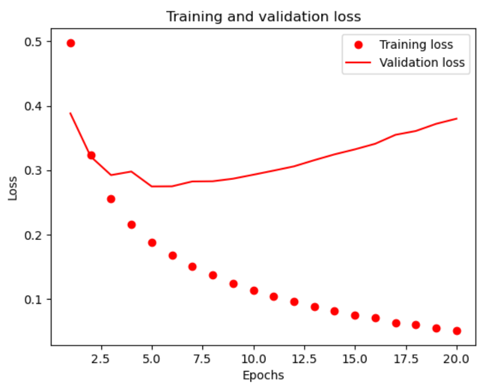
We divided the dataset into training and testing sets to assess the effectiveness of our neural network model. We ensured that the sentiment distribution was almost the same in both sets by selecting 80% of the data at random for training and the remaining 20% for testing. We trained the model on a subset of data and tested its performance on new data.

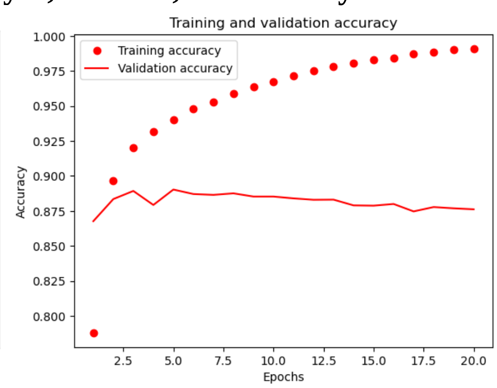
Approaches:

We proceeded by importing the data and defining the upper limit on the number of words that each review might appear at as well as the maximum amount of time that each review could last. Following that, we constructed a simple neural network model consisting of a single 16-unit hidden layer. **We used relu ,tanh as the activation functions and binary Cross entropy,MSE as the loss function and Adam, Regularization as the optimizer and dropout and hyper tuned as parameters for the hidden layer.** Next, we looked into above approaches to enhance the model's functionality. subsequently we created models with one, two, and three hidden layers by experimenting with the number of hidden layers. We trained, evaluated, and compared the models using the training and test datasets. We found that having three hidden layers led to higher validation and test accuracy as compared to using only one hidden layer.

Below are the different approaches we used for validation and test accuracy:

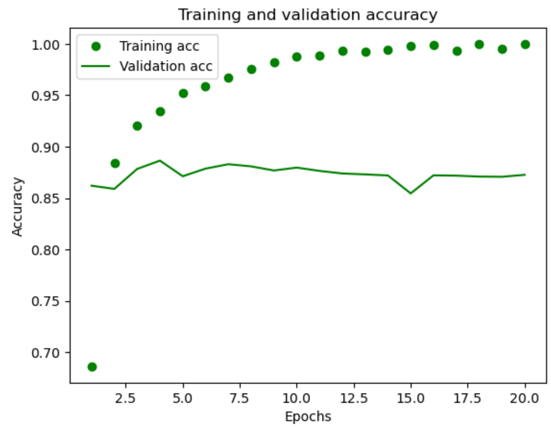
Neural network with – 1-hidden layer,16-units , loss= binary crossentropy,activation=relu





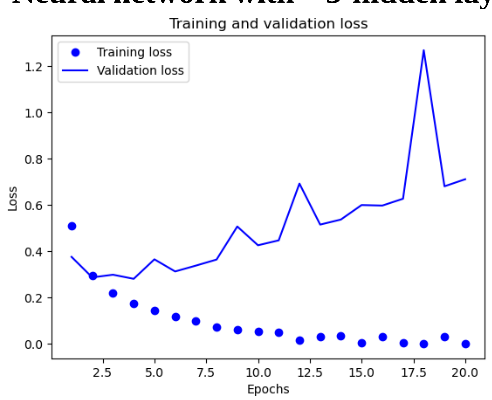
- **Accuracy = 88.6%**

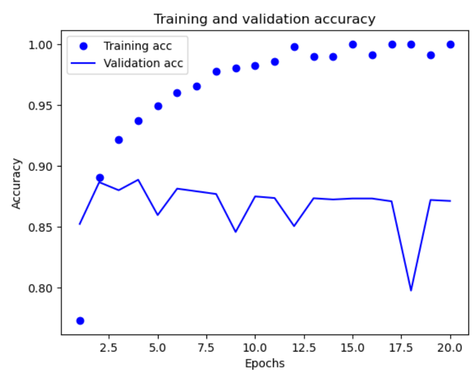
Neural network with – 3-hidden layer,16-units , loss= binary crossentropy,activation=relu



- **Accuracy=88.4%**

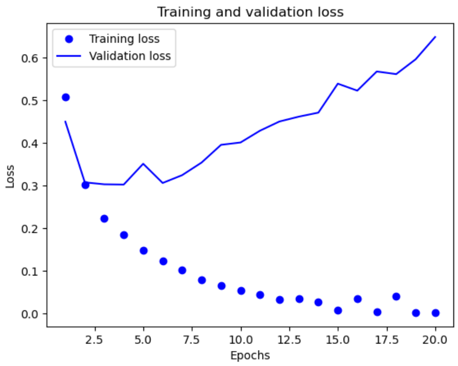
Neural network with – 3-hidden layer,32-units , loss= binary crossentropy,activation=relu

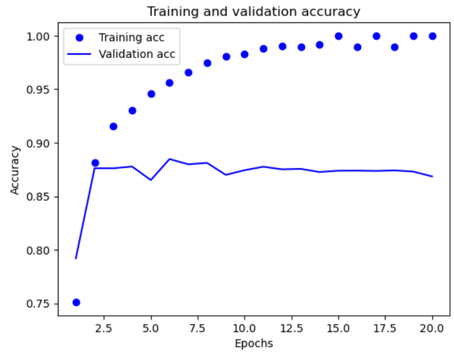




- **Accuracy=86.3%**

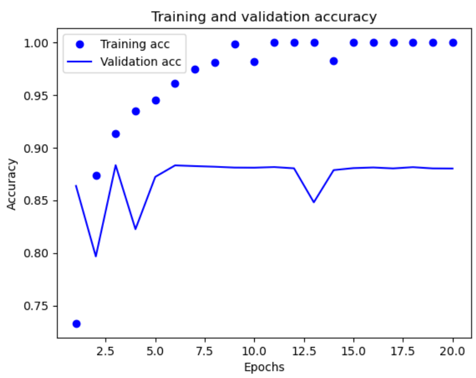
Neural network with – 2-hidden layer,64-units , loss= binarcrossentropy,activation=relu

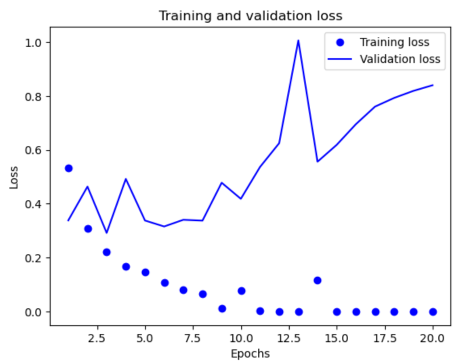




- **Accuracy=86.1%,**

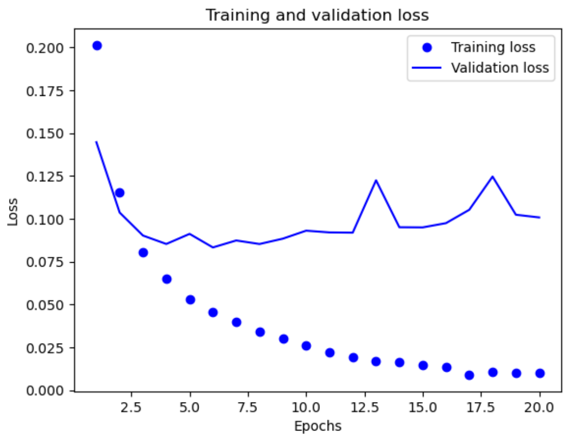
Neural network with – 3-hidden layer,128-units ,loss=binarcrossentropy,activation=relu

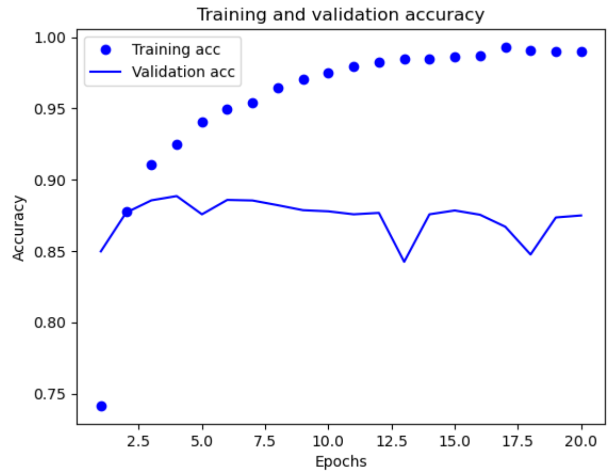




- **Accuracy=82.9%**

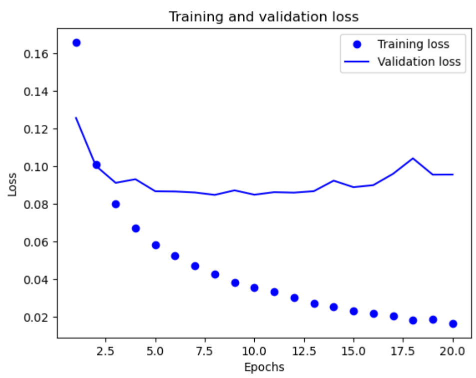
Neural network with – 3-hidden layer,16-units ,loss=MSE , activation=relu

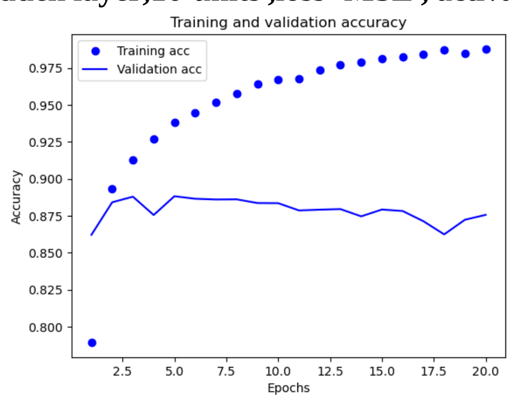




- **Accuracy = 86.5%**

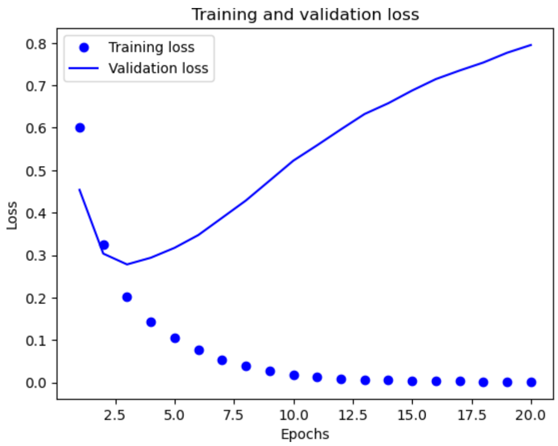
Neural network with – 1-hidden layer,16-units ,loss=MSE , activation=tanh

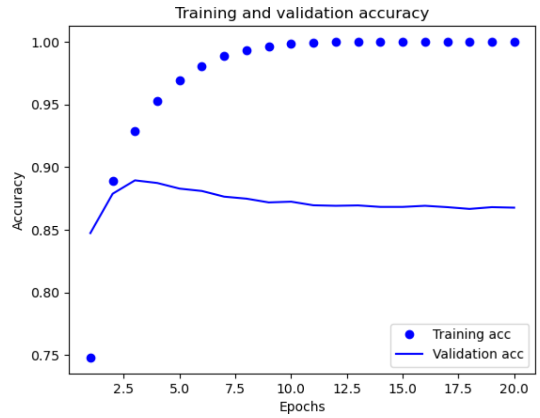




- **Accuracy=86.9%**

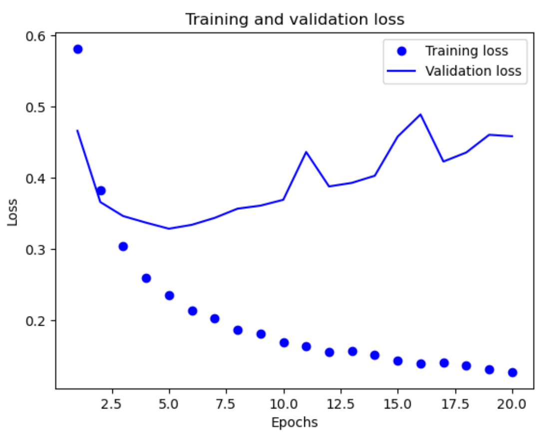
Neural network with – 3-hidden layer,16-units ,loss=binary crossentropy , activation=relu,optimizer=adam

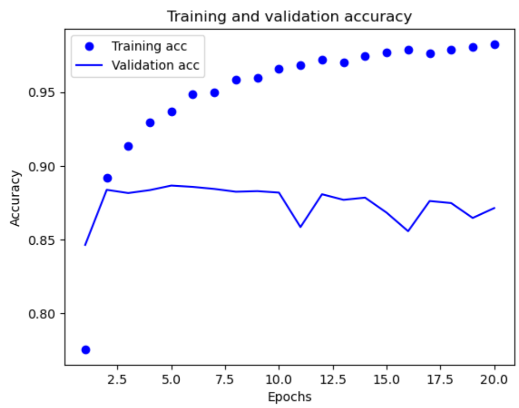




- **Accuracy = 85.9%**

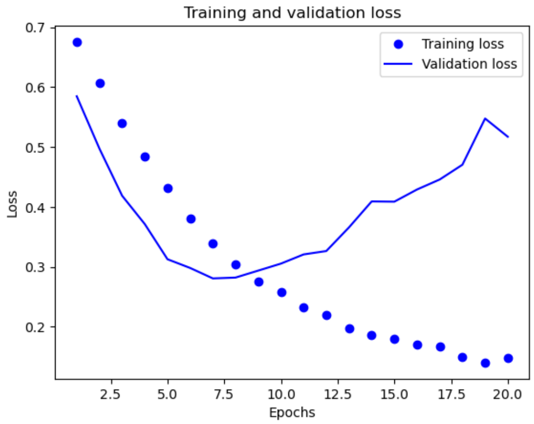
Neural network with – 2-hidden layer,16-units ,loss=binary Cross entropy , activation=relu, optimizer=rmsprop(regularization)

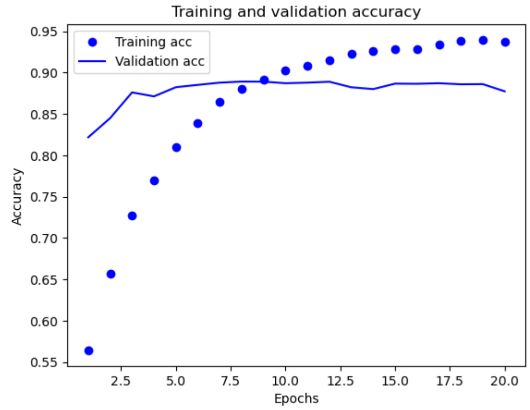




- **Accuracy = 86.1%**

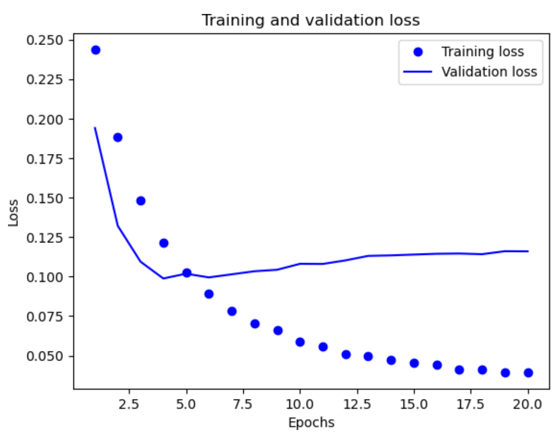
Neural network with – 3-hidden layer,16-units ,loss=binary Cross entropy , activation=relu, optimizer=rmsprop(regularization),dropout=0.5

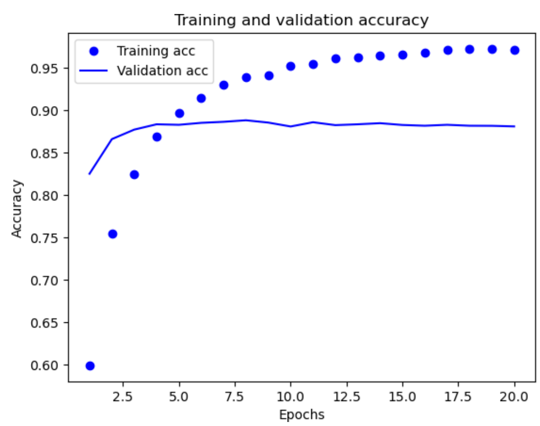




- **Accuracy=86.6%**

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))





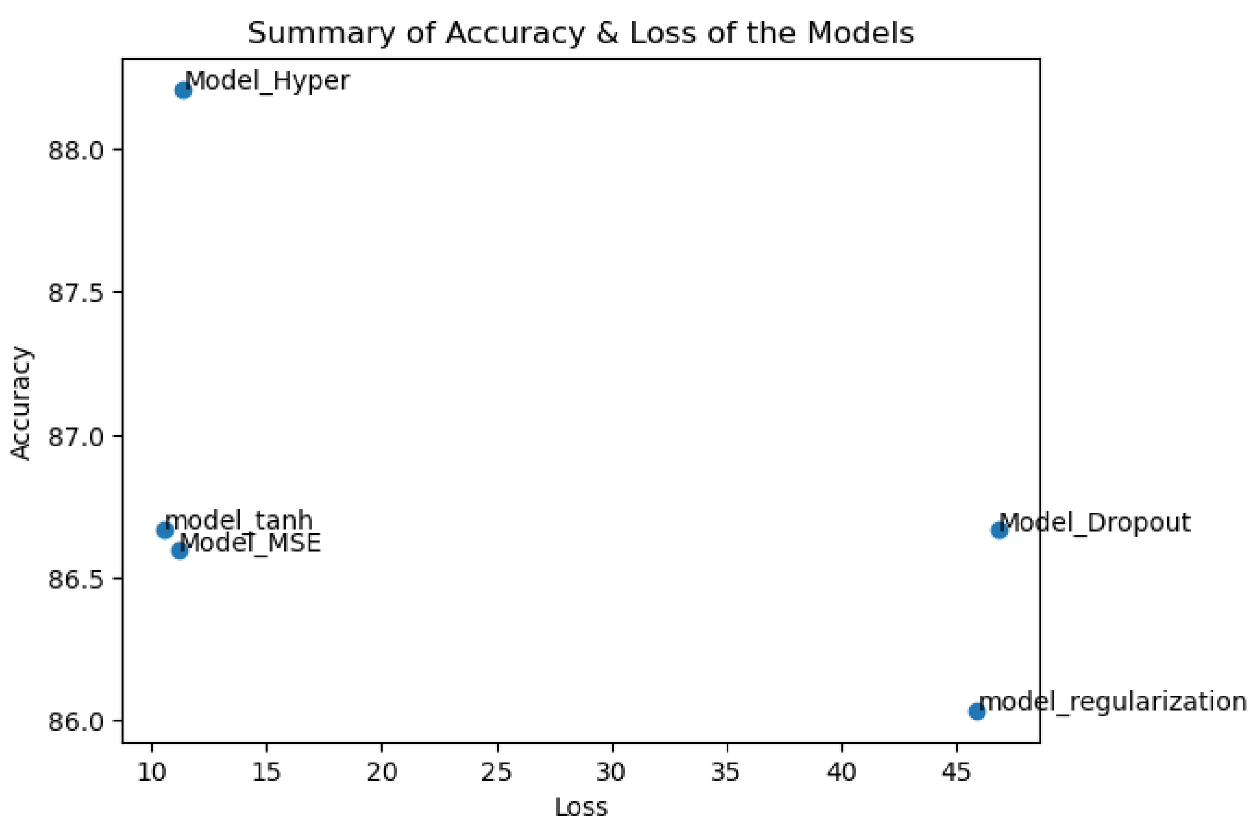
- **Accuracy=88.2%**

Conclusion:

In the end, we experimented with dropout regularization to prevent overfitting. Using training and test datasets, we developed a new model using dropout layers. When compared to the baseline model, we found that using dropout regularization increased the validation accuracy. Different neural network model alterations are deduced to have varying degrees of accuracy and loss.

The Model Hyper achieved the highest accuracy and loss, demonstrating that optimal performance for the IMDB dataset may be achieved by utilizing three thick layers with a dropout rate of 0.5. The MSE loss function had the lowest loss value when compared to binary cross-entropy. The vanishing gradient problem results in a lower accuracy for the tanh activation function. It was demonstrated that the Adam optimizer function was effective for computing the model.

Regularization reduced overfitting and produced smaller losses, with the accuracy of the L-2 model being slightly higher. The dropout technique eventually resulted in a decrease in the loss function but no change in accuracy. The graph indicates that the Model Hyper, with a comparatively small loss, has the best accuracy.



Model Hyper is more accurate than Model MSE, which has the lowest loss value. In comparison to other models, the Model Tanh exhibits low accuracy, whilst the Model Regularization demonstrates a significant loss and low accuracy.

We may therefore conclude that, out of all the models taken into consideration, the Model Hyper performs the best.