**NEURAL NETWORKS**

**Introduction**: Out of the 50,000 movie reviews in the IMDb review dataset, 25,000 are classified as "positive" or "good," and the remaining 25,000 are classified as "negative." This study uses the IMDb dataset to investigate many methods for improving a neural network model's performance. Many modifications will be made to an existing neural network model, such as adjustments to the number of hidden layers, units, activation function, loss function, and regularization techniques like dropout. After then, the results will be examined.

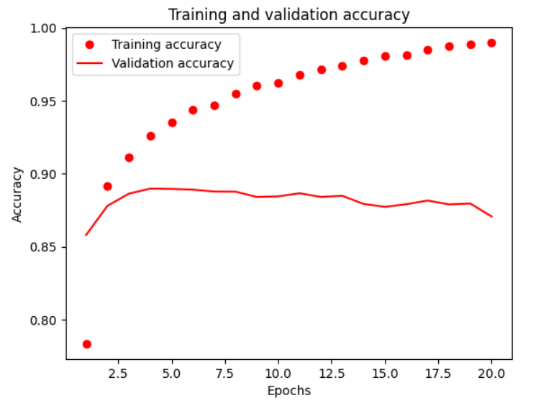
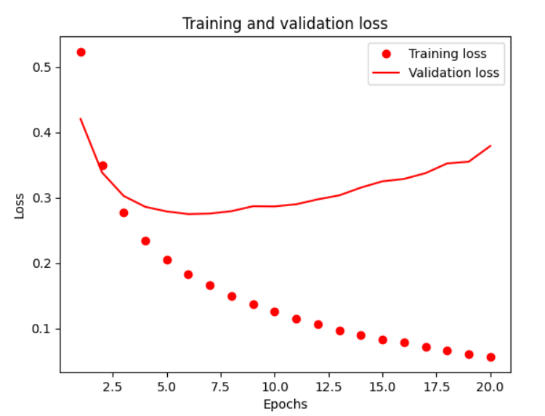
**Data Processing:** We carried out a few preparation steps to transform the unprocessed text data from the IMDb review dataset into a format appropriate for neural network training. Due to the extremely high-dimensional input space that would arise from include every word in the dataset, we only took into account the top 10,000 most often occurring phrases. We then used a dictionary to translate the terms in the top 10,000-word list to the appropriate indices in order to turn the text reviews into numeric representations. We needed to transform the integer representations to tensors in order to use neural networks. To make sure that every review had the same length, we padded shorter reviews with zeros and abridged longer reviews. Thus, each review was represented as a fixed-length vector with each element denoting a dictionary word's index. Lastly, we transformed the integer representations into binary values using the one-hot encoding approach. As a result, the data produced a binary matrix where each row represented a review and each column included a dictionary term. We separated the dataset into training and testing sets in order to assess the effectiveness of our neural network model. The model was trained on a subset of the data, and its performance was then evaluated on new data.

**Approaches**: Next, we set the maximum word count and time for each review after importing the data. Next, we constructed a simple neural network model consisting of a single 16-unit hidden layer. The activation functions we used were relu and tanh; the loss function was binary Cross entropy with MSE; the optimizer was Adam Regularization; the hidden layer's parameters were dropout and hypertuned. Next, we examined the aforementioned techniques 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. Comparing the use of three hidden layers to that of one, we discovered that the latter improved test validity and accuracy.

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

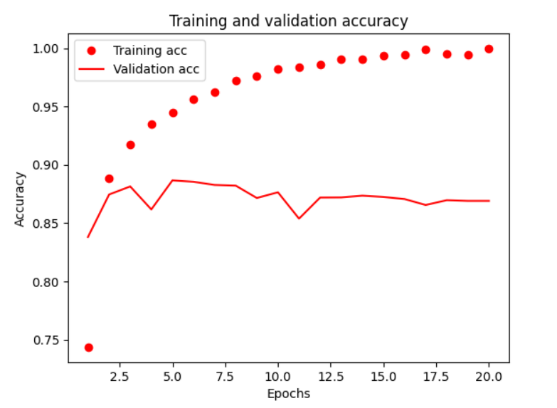
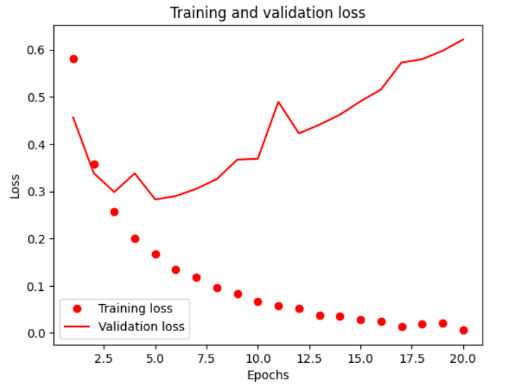
*Neural network with – 1-hidden layer,16-units , loss= binary crossentropy, activation= relu*

• Accuracy = 88.6%



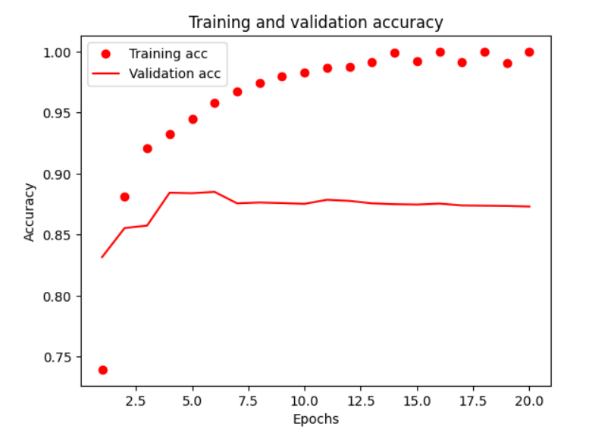
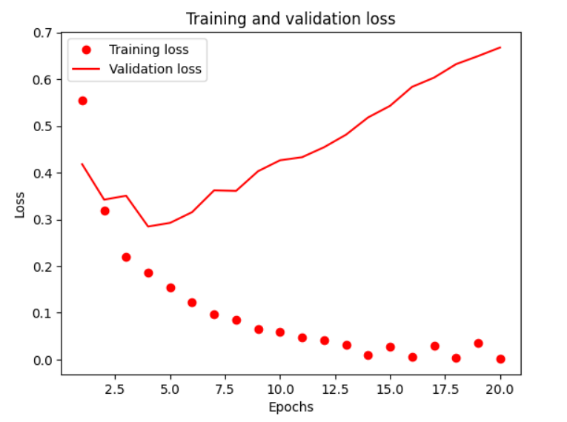
*Neural network with – 3-hidden layer,16-units , loss= binary crossentropy, activation= relu*

• Accuracy = 88.4%



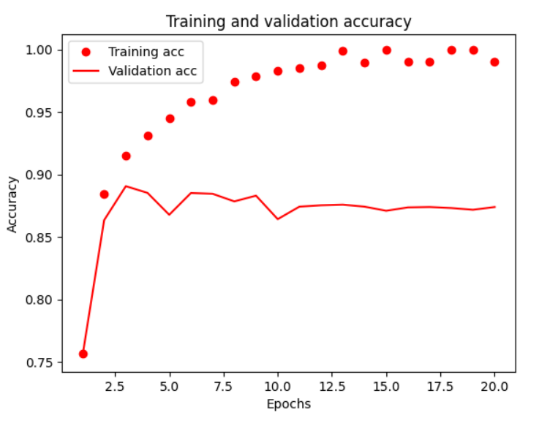
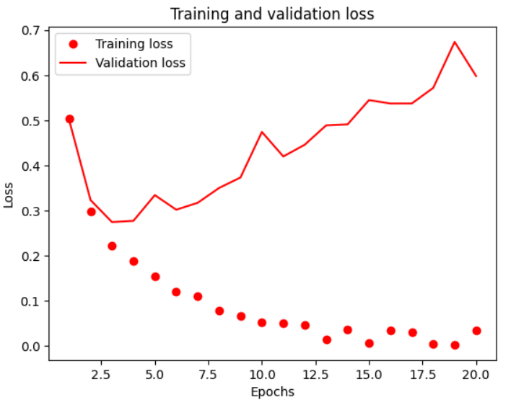
*Neural network with – 3-hidden layer,32-units , loss= binary crossentropy, activation= relu*

• Accuracy = 86.8%



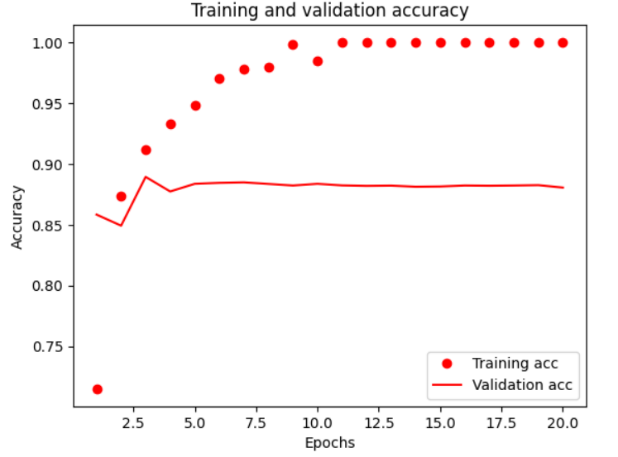
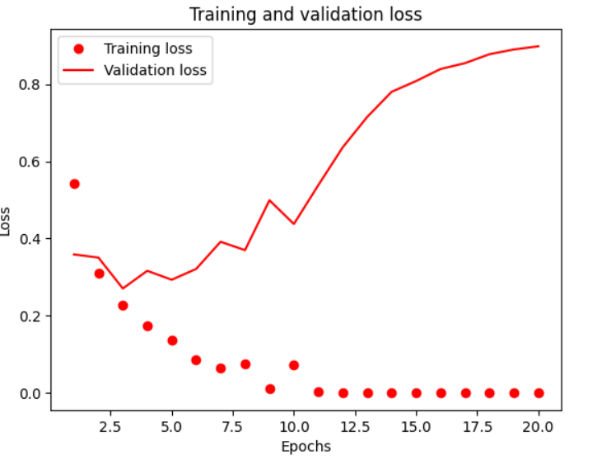
*Neural network with – 2-hidden layer,64-units , loss= binarcrossentropy, activation= relu*

• Accuracy = 86.7%



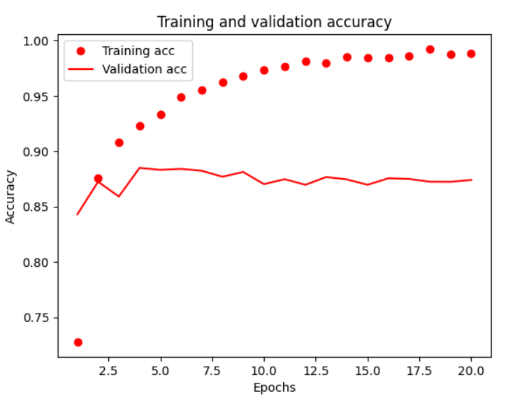
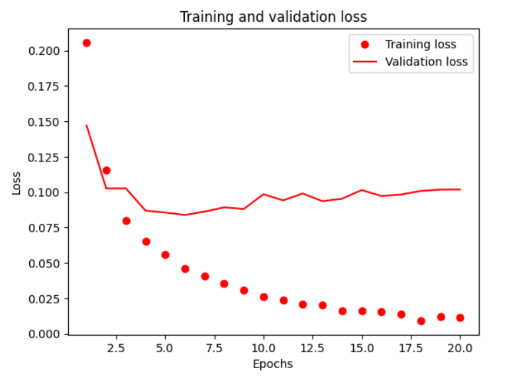
*Neural network with – 3-hidden layer,128-units ,loss=binarcrossentropy, activation= relu*

• Accuracy=84.6%



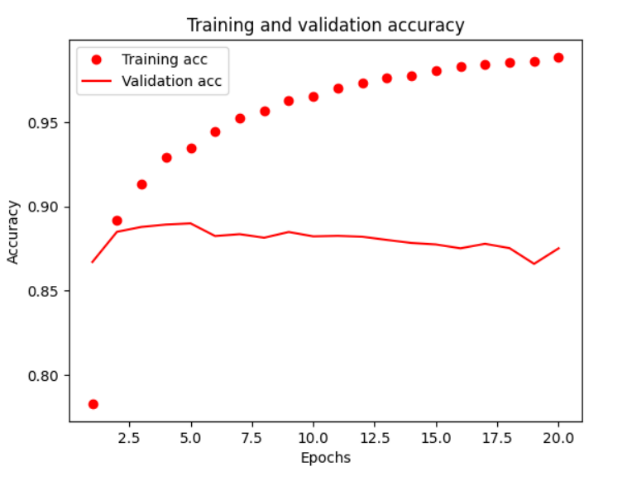
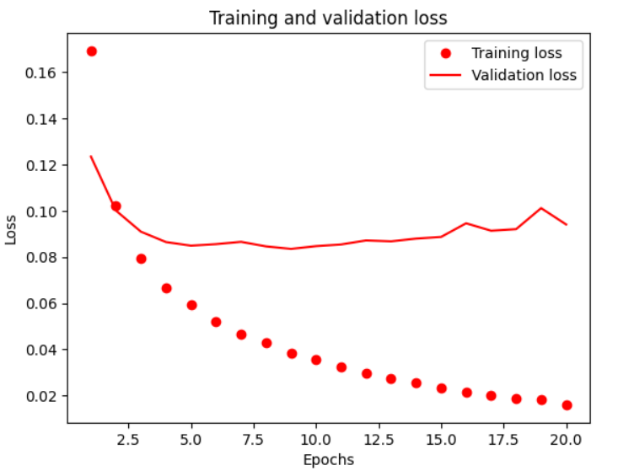
*Neural network with – 3-hidden layer,16-units ,loss=MSE , activation= relu*

• Accuracy = 86.5%



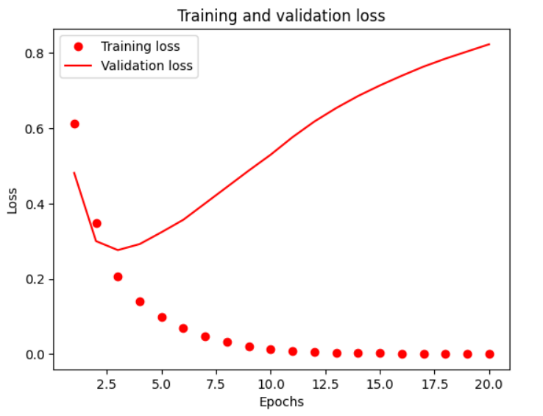
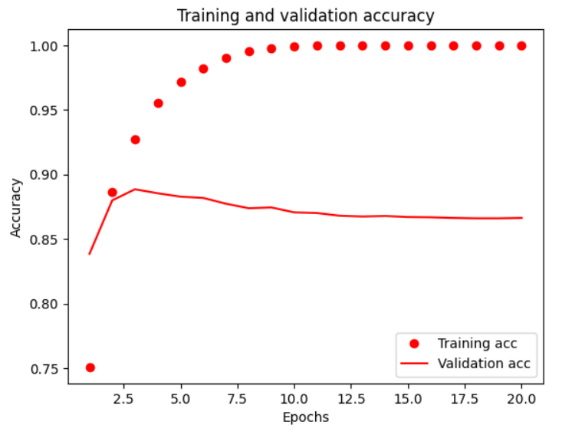
*Neural network with – 1-hidden layer,16-units ,loss=MSE , activation = tanh*

• Accuracy=86.7%



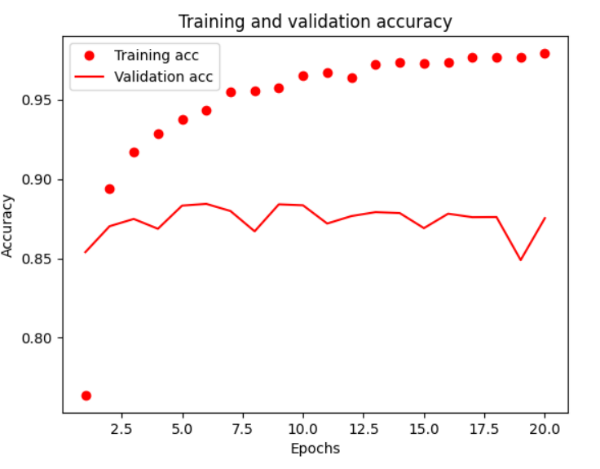
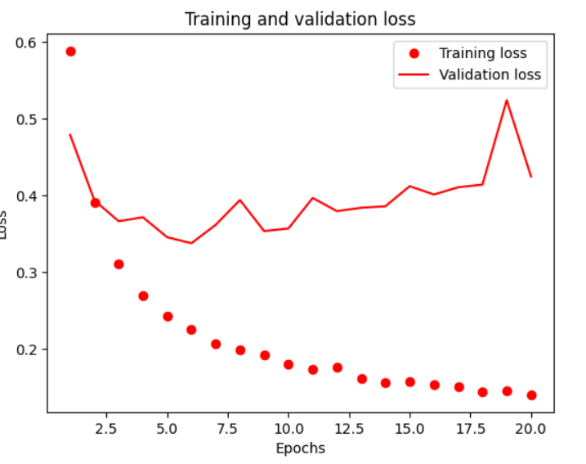
*Neural network with – 3-hidden layer,16-units ,loss=binary crossentropy, activation = relu, optimizer = adam*

• Accuracy = 85.6%

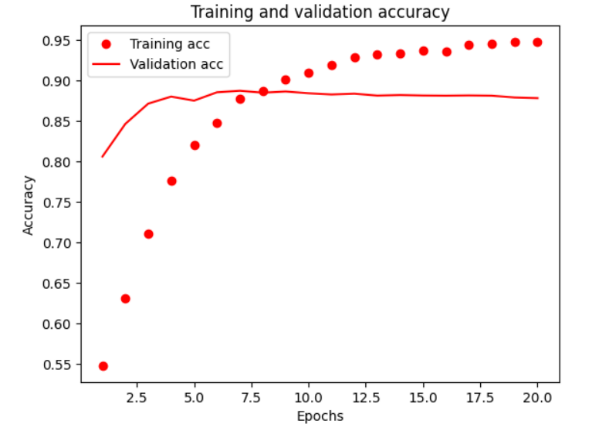
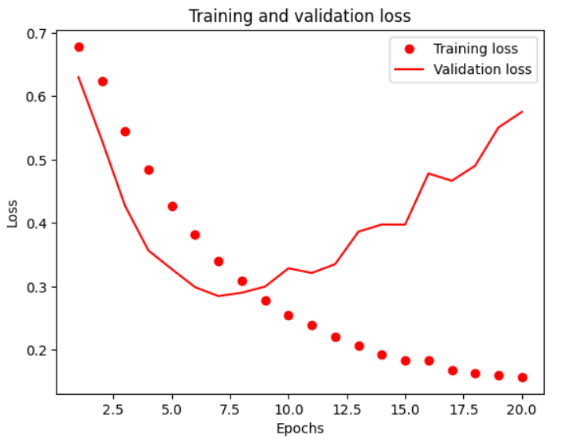
*Neural network with – 2-hidden layer,16-units ,loss=binary Cross entropy , activation=relu, optimizer=rmsprop(regularization)*

• Accuracy = 84%



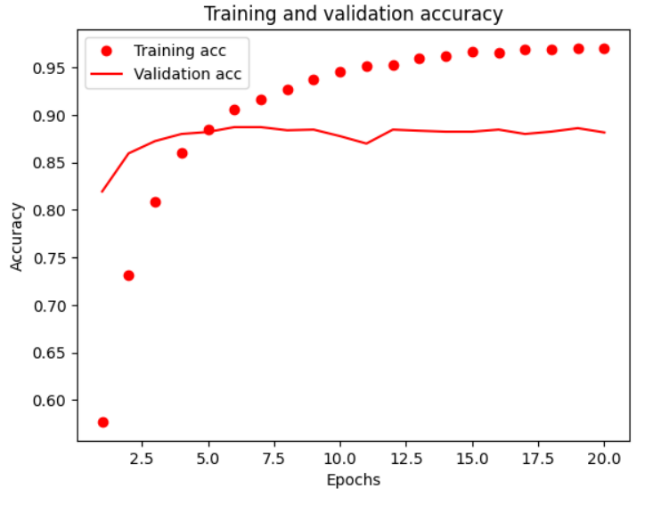
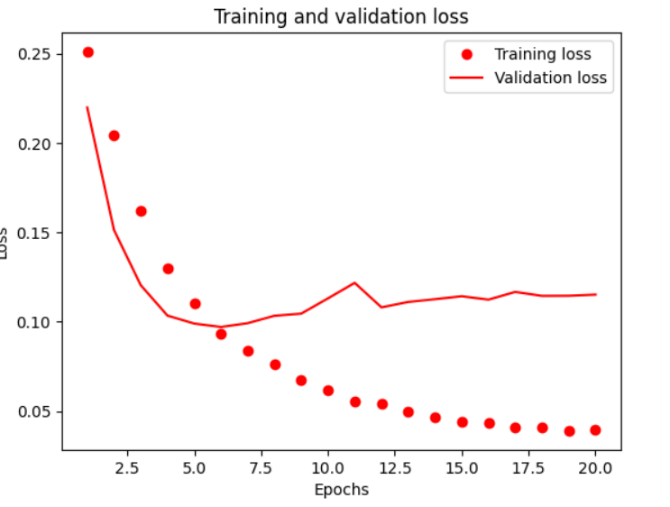
*Neural network with – 3-hidden layer,16-units ,loss=binary Cross entropy , activation=relu, optimizer=rmsprop(regularization),dropout=0.5*

• Accuracy=87.1%



*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**:

Finally, in order to avoid overfitting, we attempted dropout regularization. Dropout layers were used to create our new model, which has training and test datasets. We discovered that using dropout regularization improved the validation accuracy in compared to the baseline model. It follows that different neural network model modifications should have different accuracy and loss functions. The Model Hyper yielded the best accuracy and loss, suggesting that the IMDB dataset would benefit from the inclusion of three thick layers with a dropout rate of 0.5. The MSE loss function showed the lowest loss value when compared to binary cross-entropy. The vanishing gradient problem lowers the precision of the tanh activation function. It was demonstrated that the Adam optimizer function may be used to compute the model effectively. The graphic below illustrates the many models that are employed together with their accuracy and validation loss performance, making it easier for us to understand each model. With the lowest loss value, Model MSE is less accurate than Model Hyper. The Model Regularization shows poor accuracy when compared to other models. As a result, we can say that the Model Hyper performs the best among all the models examined.

