

NEURAL NETWORKS SUMMARY

GEETHIKA VULLI

811290653

Introduction:

The IMDb review dataset contains 50,000 movie reviews, 25,000 of which are labeled as "positive" or "good," and the remaining 25,000 as "negative." The present study tackles multiple strategies to enhance the performance of a neural network model using the IMDb dataset. An present neural network model will go through many alterations, including changes to the number of hidden layers, units, activation function, loss function, and regularization strategies like dropout. The outcomes will then be analysed.

Data Processing

To convert the raw text data from the IMDb review dataset into a format suitable for neural network training, we performed a few preprocessing operations. We only considered the top 10,000 most frequent terms in the dataset because included every word in the dataset would result in an extraordinarily high-dimensional input space. subsequently in order to convert the text reviews into integer representations, we mapped the terms in the top 10,000-word list to the corresponding indices using a dictionary.

In order to employ neural networks, we had to convert the integer representations to tensors. We shortened longer reviews and padded shorter ones with zeros to ensure that all reviews were the same length. Consequently, every review was represented as a fixed-length vector, where each element signified the index of a dictionary word.

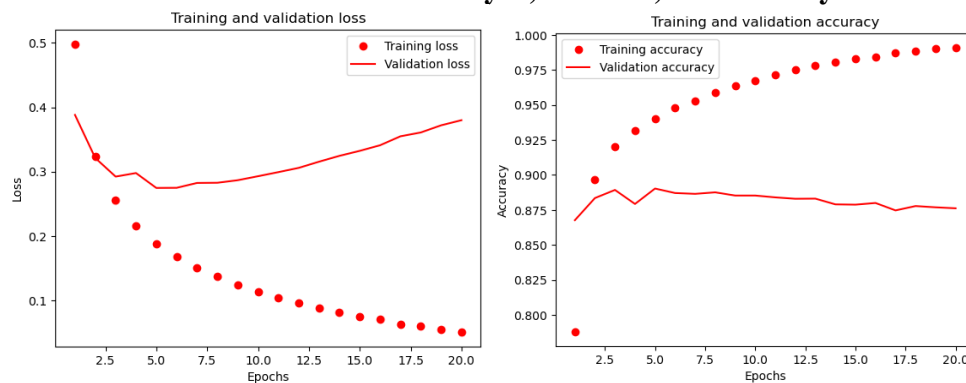
Finally, we used the technique of one-hot encoding to convert the integer representations into binary values. Therefore, a binary matrix emerged from the data, with a review for each row and a dictionary word for each column. We subdivided the dataset into testing and training sets with the objective to evaluate our neural network model's performance. We used a portion of data to train the model, then we tested its performance using fresh data.

Approaches:

We then imported the data and set the maximum word count and duration for each review. After that, we built a straightforward neural network model with 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, in an effort to increase the model's utility, we looked at the previously mentioned methods. Next, we created models with one, two, and three hidden layers by changing the quantity of hidden layers. We evaluated, contrasted, and trained the models using the test and training datasets. We found that adding three hidden layers enhanced test accuracy and validity when compared to using only one hidden layer..

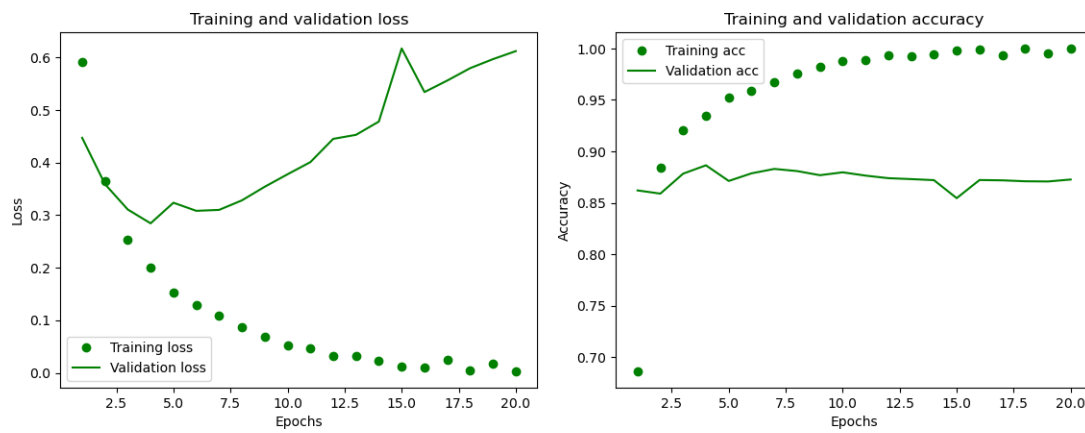
Below are the different approaches we used for validation and test accuracy:

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



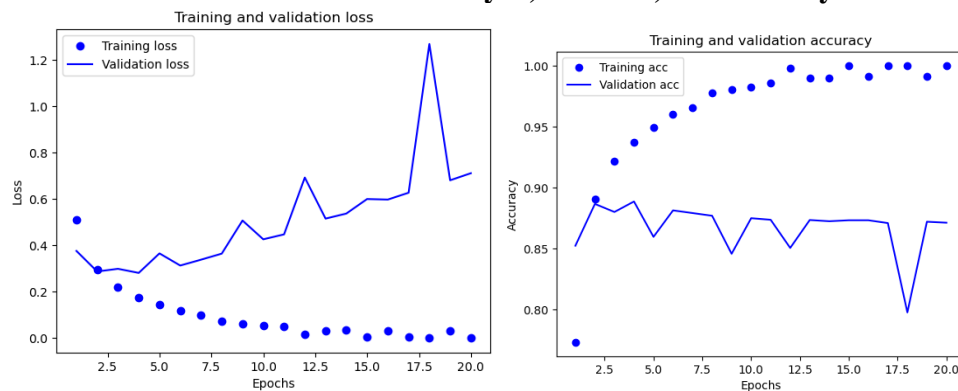
- Accuracy = 88.6%

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



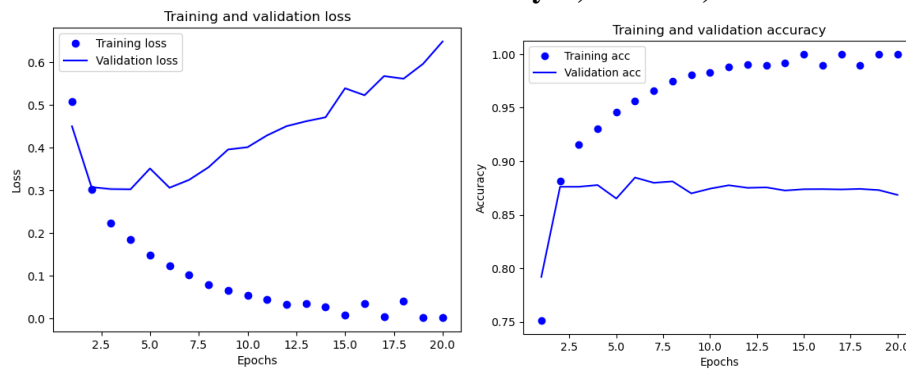
- **Accuracy=88.4%**

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



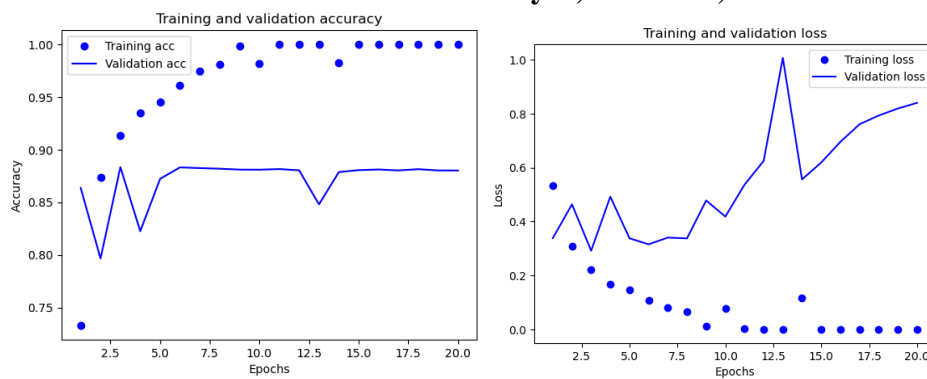
- **Accuracy=86.3%**

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



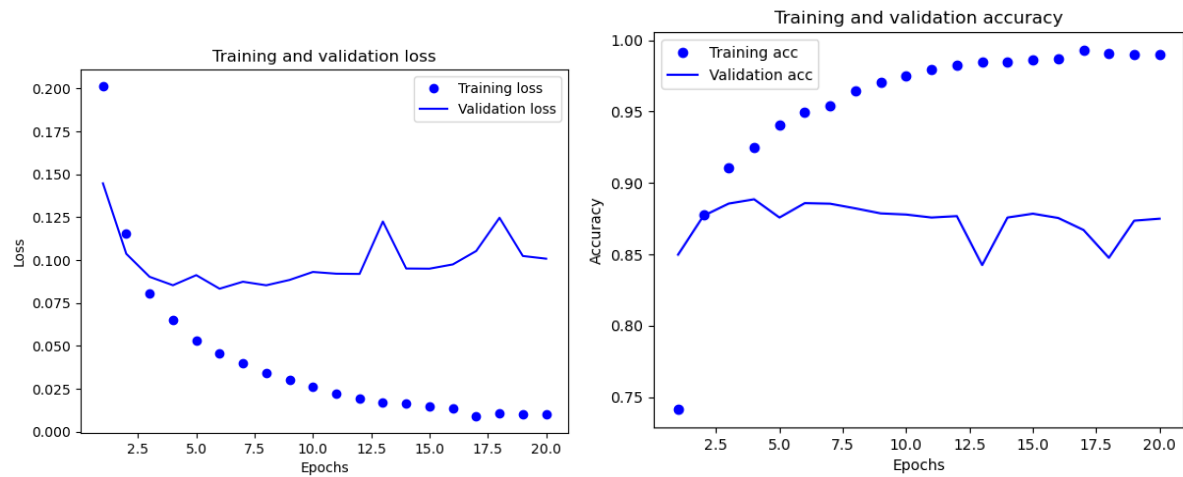
- Accuracy=86.1%,

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



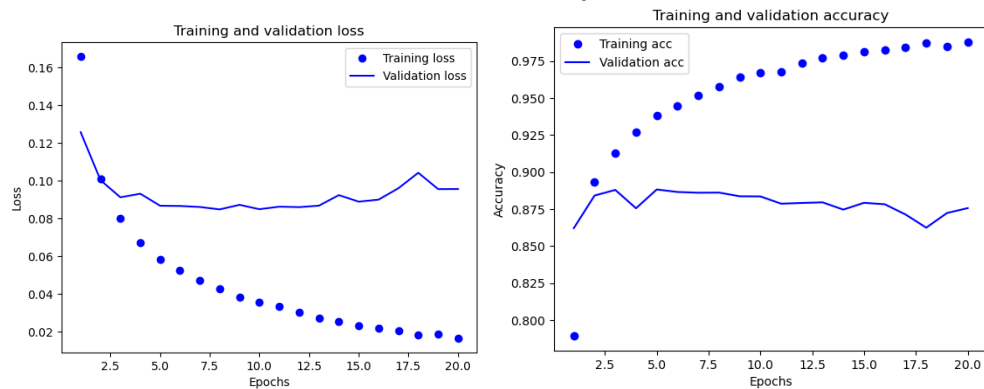
- Accuracy=82.9%

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



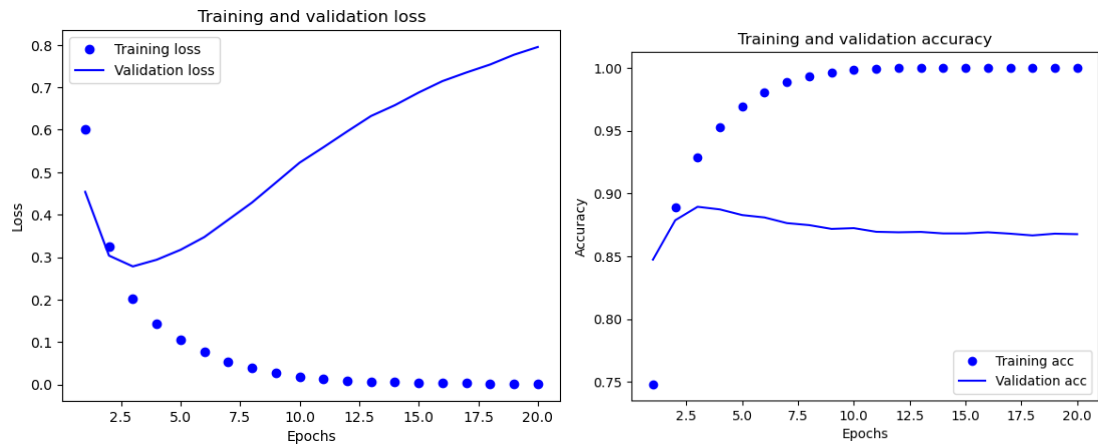
- Accuracy = 86.5%

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



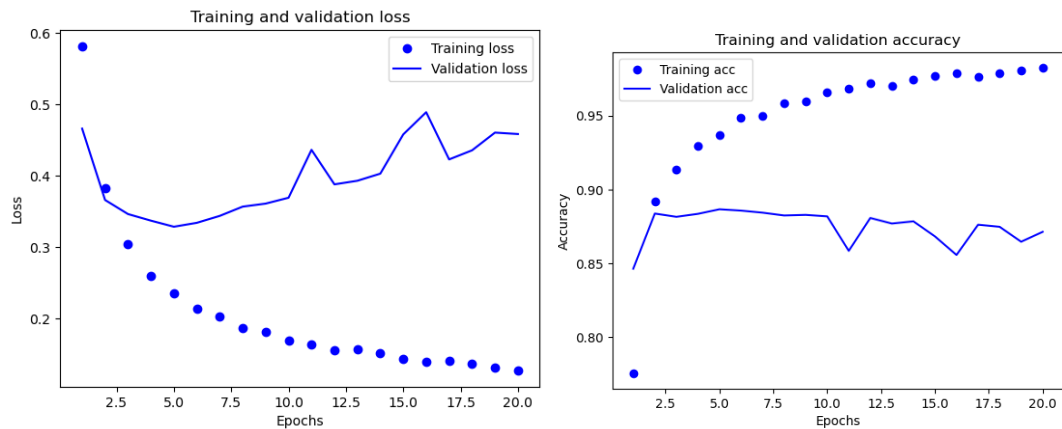
- Accuracy=86.9%

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



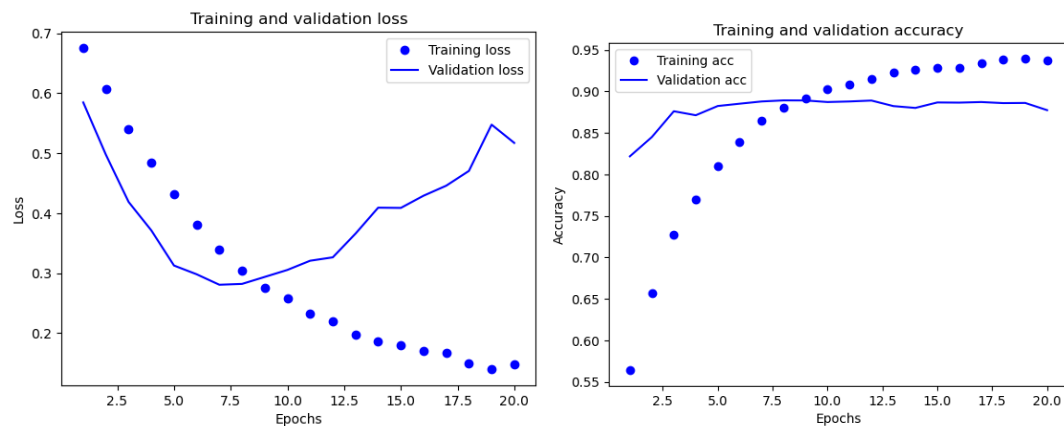
- Accuracy = 85.9%

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



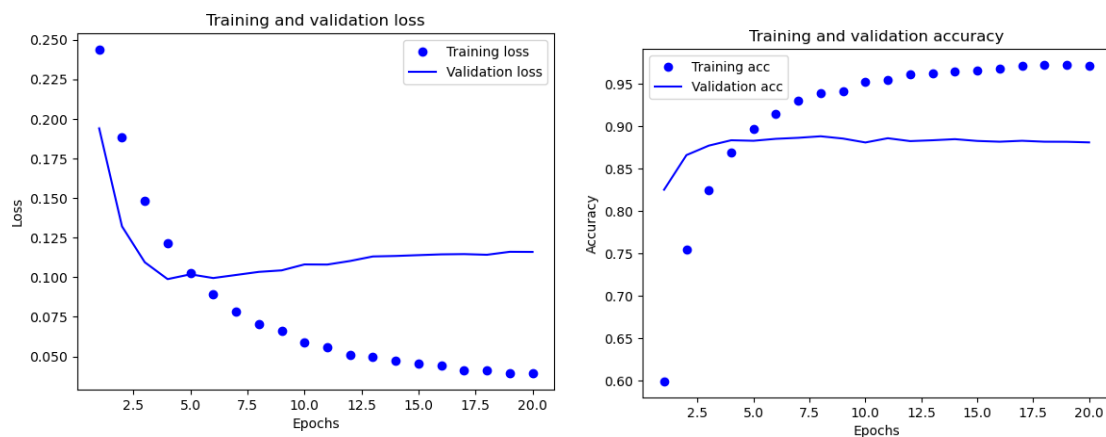
- Accuracy = 86.1%

**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))**



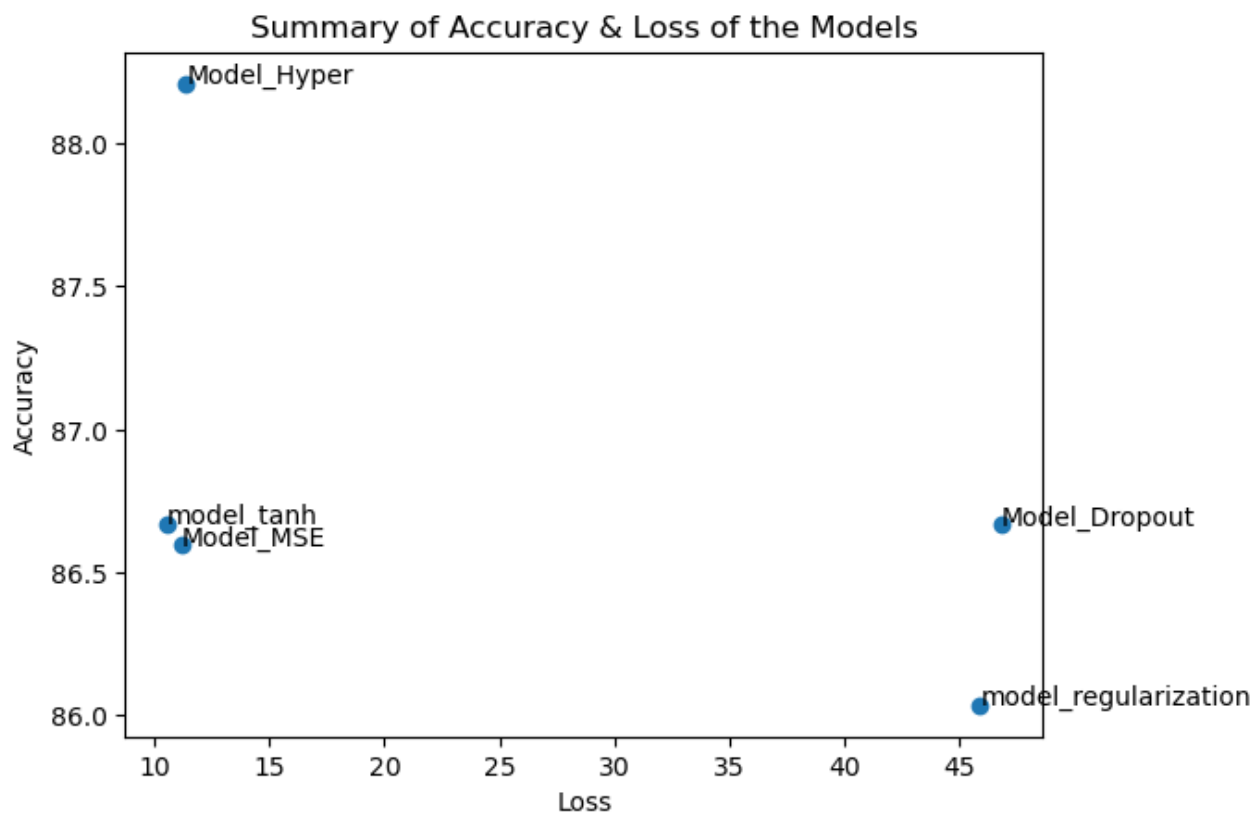
- Accuracy=88.2%

Conclusion:

At last, we tried dropout regularization to prevent overfitting. Our new model with training and test datasets was made utilizing dropout layers. In contrast to the baseline model, we found that the validation accuracy increased with the use of dropout regularization. Thus, varying neural network model alterations are assumed to have various accuracy and loss functions.

The Model Hyper produced the best accuracy and loss, indicating that three thick layers with a dropout rate of 0.5 may be used to provide the best results for the IMDB dataset. Compared to binary cross-entropy, the MSE loss function exhibited the lowest loss value. Tanh activation function accuracy is reduced as a result of the vanishing gradient issue. It was shown that the model could be computed efficiently using the Adam optimizer function.

Below image shows about different models used and their performance of accuracy and validation loss in the model, which helps us to easily know about each model.



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

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