**Summary:**

**Aim:** The main objective of the neural network model is to optimize a neural network for the analysis on the IMDB dataset. Here as I have considered a 2500 IMDb reviews for my model. So, I have considered few libraries for neural network like TensorFlow as it has a good support and implementation.

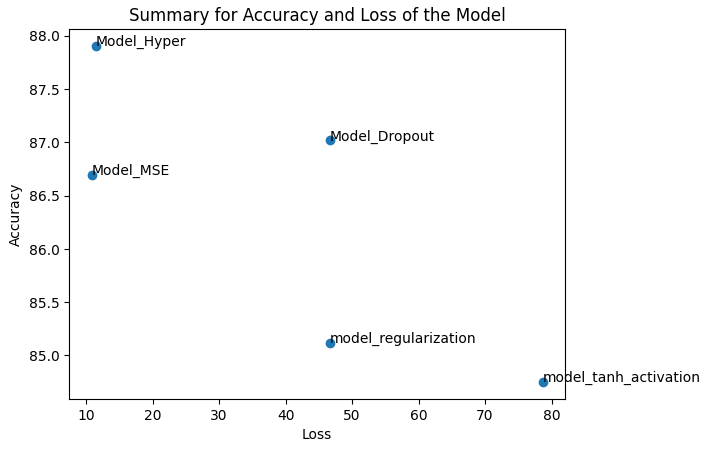
My proposed solution to the problem focused on first analyzing the importance of the Keras Sequential model that is mainly a stack of layers for constructing neural networks. An important idea of importing the essential features of layers, Dense, Dropout, and Regularizers using TensorFlow.keras helped in designing the neural network.

In order to compare the results, I built models with neural networks of two, three, and six layers to train with 16 hidden neurons in the first two-layer model, 64 neurons in the three-layer model, and 64 neurons in the six-layer model. One interesting finding was that whichever number of layers we stack, the gains are limited after a certain point. The creation of the Sequential model with model = keras.Sequential() defines the required architecture which is a model with layers that includes input, hidden, output. The meaning of the tanh activation function (model.add(Dense(64, activation="tanh")) is to created 64 neurons in the layer to work with the vectors.

The Dropout layer (model.add(Dropout(0.5))) is useful in avoiding the overfitting problem of neural networks as it randomly drops out neurons. By setting it to 0.5, what it means is that we have actually pruned out the network to half the measure though it only undergoes this process.

Despite attempts to incorporate L1 and L2 norms into regularization techniques to increase the performance, these were not successful and might even have led to a decreased performance. This means that the model might have reached a stage of self-saturation in the sense that the top five validation accuracy could have reached the peak value of 85% to 86%

When assigning loss evaluation criteria the use of mean squared error (MSE) instead of binary\_crossentropy was adopted and its performance increased yields values of a lower validation loss than achieved by binary\_crossentropy. At some point ReLU was the best activation function instead of sigmoid and tanh because ReLU solve the vanishing gradient problem. However, as for the case of tanh it has the similar results to ReLU based on the results of the training errors of the networks.



The above figure gives the overall accuracy and loss of the model.

We can observe the differences in the model performance by looking into training accuracy and validation accuracy. I am considering Tanh activation, model regularization, model mse, model dropout and model hyper.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Models | Training Accuracy | Validation Accuracy |  |  |  |
| MSE Model | 86.6% | 87.7% |  |  |  |
| Tanh Activation Model | 84.7% | 86.3% |  |  |  |
| Adam Optimizer | 86.12% | 87.03% |  |  |  |
| Regularization | 85.11% | 86.98% |  |  |  |
| Model Hyper | 87.09% | 88.03% |  |  |  |
| Drop out | 87.01% | 88.04% |  |  |  |