Assignment-1: Neural Networks

Esha Reddy Emani

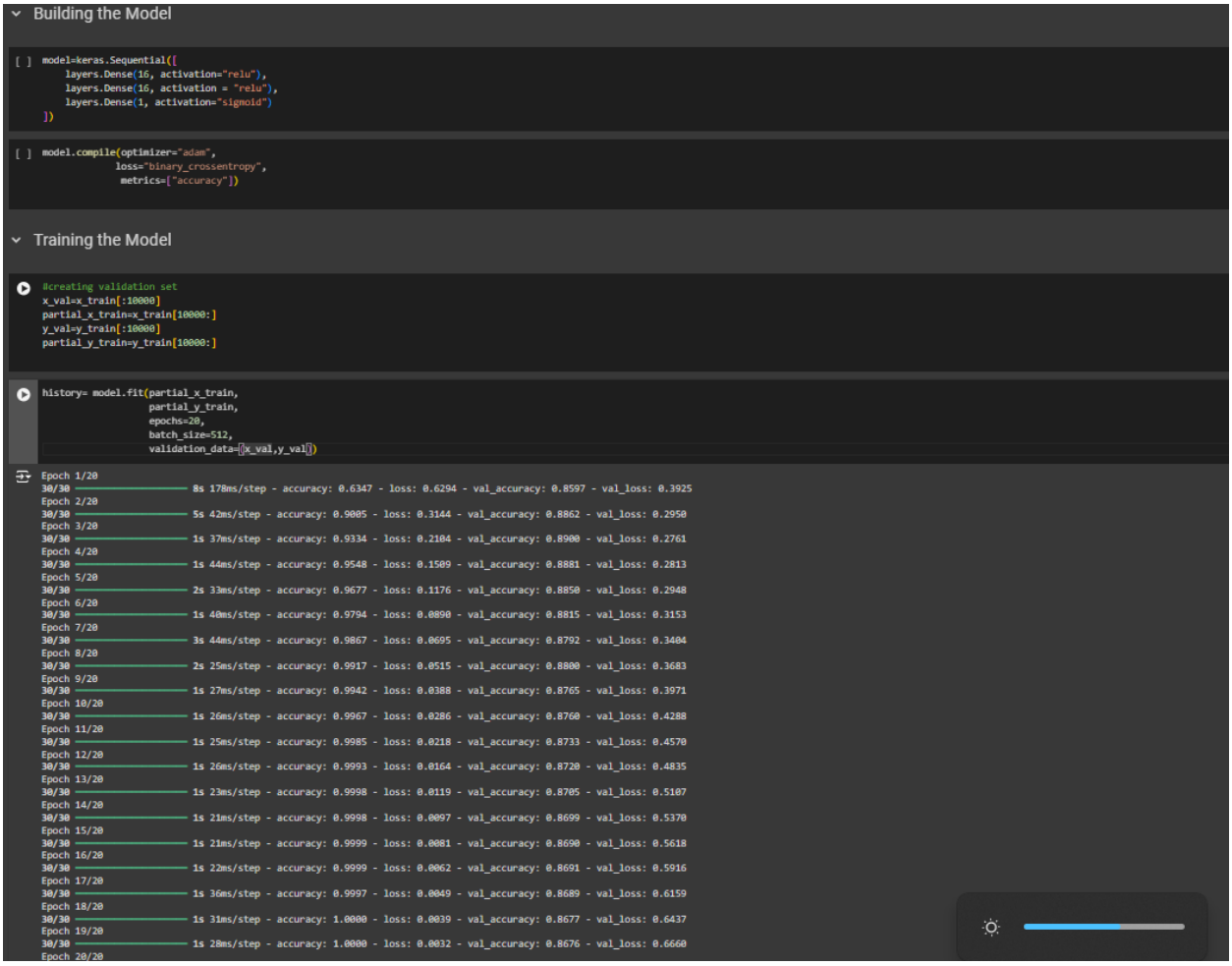
Introduction:

This assignment involves modifying a neural network model for the IMDB dataset by experimenting with different configurations, such as adjusting hidden layers, hidden units, loss functions, and activation functions. You will also apply techniques like regularization and dropout to improve the model's performance and analyze how these changes impact validation and test accuracy.

Steps:

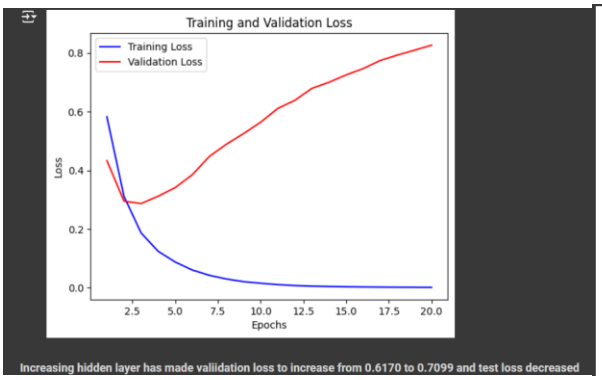
The Original Model:

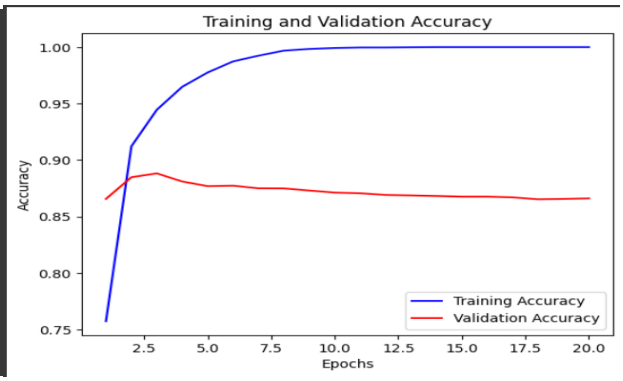
The neural network model was trained on the IMDB dataset with two hidden layers of 16 units each and used the ReLU activation function and binary cross-entropy as the loss function. The model was trained for 20 epochs with a validation set, and initially, validation accuracy increased while validation loss decreased. However, after a few epochs, the model started to overfit, as seen by the widening gap between training and validation performance. By the end of training, validation accuracy plateaued at around 86.7%, while validation loss increased, reaching 0.6877. The final test accuracy was 85.65%, with a test loss of 0.7522, indicating overfitting despite high training accuracy.



Step 1: Experimenting with different numbers of hidden layers, excluding the use of two hidden layers as done previously.

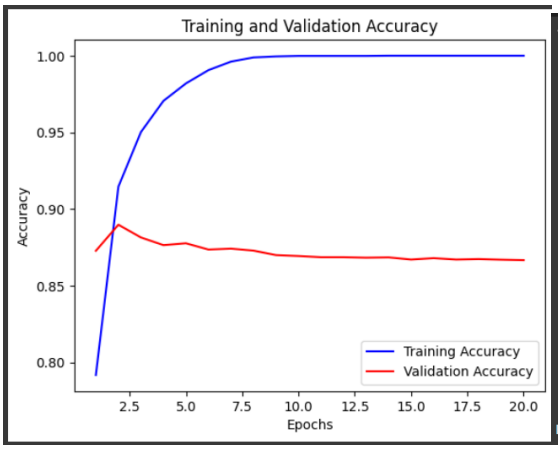
The modified neural network model with three hidden layers of 16 units each was trained for 20 epochs on the IMDB dataset. Initially, the model showed improvements in accuracy, reaching a validation accuracy of around 87% in the early epochs. However, as training progressed, the model began to overfit, with validation loss increasing steadily while validation accuracy remained relatively flat. By the end of training, validation accuracy dropped slightly to 86.45%, and the final test accuracy was 85.20%, with a significantly higher test loss of 0.9162, indicating overfitting and a diminishing return from adding an extra hidden layer.

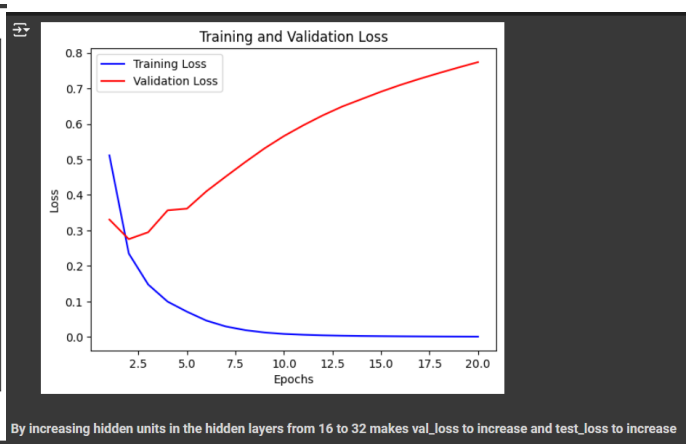




Step2: Experimented with using layers that have more or fewer hidden units, such as 32 units, 64 units, and others.

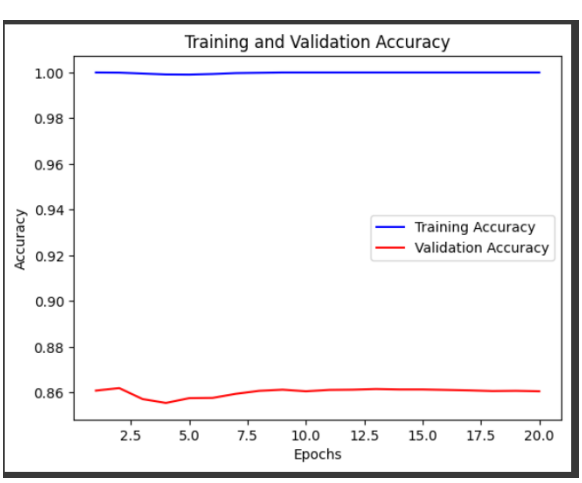
In this step, a neural network model is created with two hidden layers, each containing 32 units using the ReLU activation function, and an output layer using the sigmoid activation function. The model is compiled with the Adam optimizer and binary cross-entropy loss function, and trained on the IMDB dataset for 20 epochs with a batch size of 512. The training includes validation data to monitor performance during training. The results show that the model achieves a validation accuracy of around 87%, with a test accuracy of 85.37% and a test loss of 0.8160, indicating some overfitting as the loss increases over time.

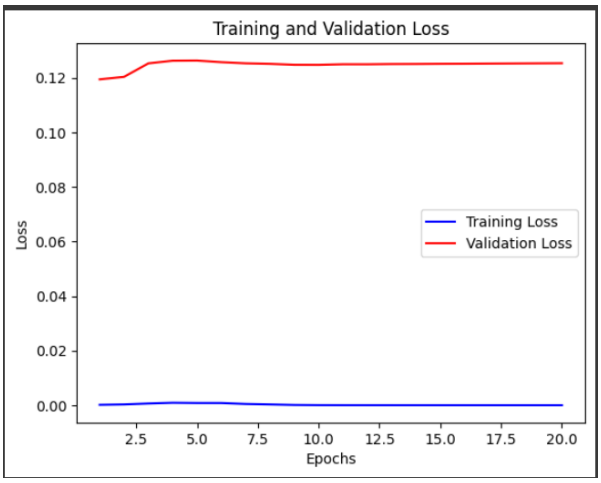




Step3: Experiment with the MSE loss function in place of binary_crossentropy.

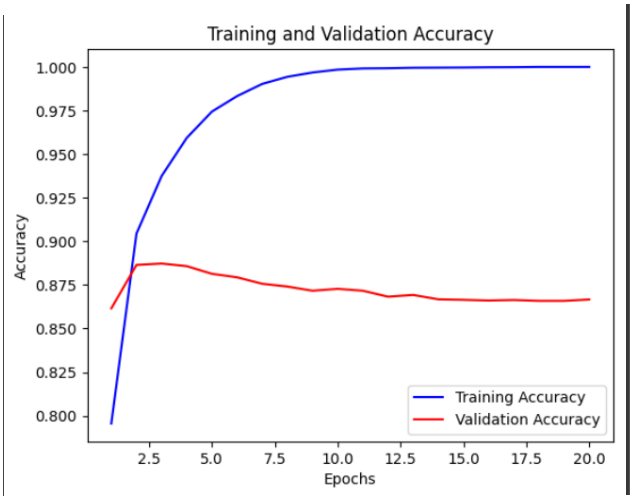
The neural network model's loss function is changed from binary cross-entropy to mean squared error (MSE). The model is compiled with the Adam optimizer and trained for 20 epochs with a batch size of 512 on the IMDB dataset. The training process includes validation data to monitor performance. The model achieves a validation accuracy of around 86% with a validation loss that stabilizes at approximately 0.123 after multiple epochs. When evaluated on the test set, the model achieves an accuracy of 84.68% with a loss of 0.1355. Overall, using the MSE loss function results in slightly lower validation accuracy and a higher loss compared to binary cross-entropy.

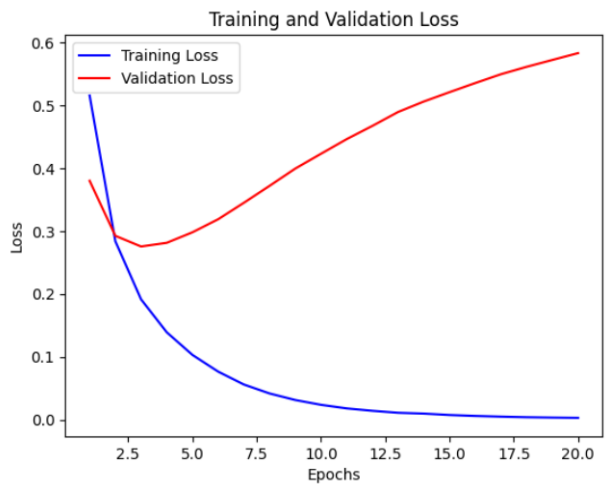




Step4: Attempt to utilize the tanh activation function in place of relu.

A neural network model named model4 is created using Keras, featuring two hidden layers with 16 units each and a "tanh" activation function. The model is compiled with the Adam optimizer and binary crossentropy as the loss function. It is trained for 20 epochs on a training dataset with a batch size of 512, while monitoring validation accuracy and loss. The training results show that the model achieved high accuracy and low loss during training. Finally, the model is evaluated on a test dataset, resulting in an accuracy of 85.09% and a loss of 0.6721. While the accuracy decreased marginally, the loss improved significantly, indicating that the "tanh" activation function may lead to a more stable model performance despite a slight drop in classification accuracy. Overall, both models demonstrate comparable performance, but the second model shows better loss values.

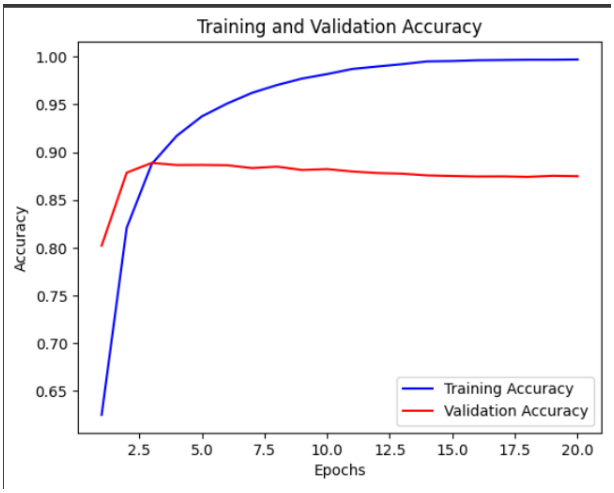




Step5: Apply any of the techniques we've learned in class, such as regularization or dropout, to enhance your model's performance on the validation set.

We were able to add Dropout to the original model, when comparing model5 to the original model, notable differences in performance and architecture emerge. Model5 utilizes two dense layers with 16 neurons each and incorporates dropout layers to reduce overfitting, while the original model featured two dense layers with 32 neurons. In terms of training accuracy, the original model

achieved nearly 100%, whereas model5 peaked at around 99.72%, suggesting better generalization. Validation accuracy improved with model5, reaching approximately 87.53%, compared to the original model's peak of 88.90%. Additionally, model5 maintained a more stable validation loss, indicating better performance in generalization, while the original model's loss increased despite high training accuracy. Ultimately, model5 demonstrated slightly better test accuracy at 86.20% versus the original model's 85.37%, highlighting the benefits of using dropout for enhanced validation performance.



S No	Method	Training Accuracy	Validation Accuracy	Test Accuracy
1	Original Method	1	0.866	0.8534
2	Adding hidden layer	1	0.8662	0.853
3	32 Hidden units	1	0.8667	0.852
4	Using mse	1	0.8606	0.844
5	Using tanh	1	0.8667	0.852
6.	Using Dropout	1	0.87	0.86

Conclusion:

In conclusion, the introduction of dropout layers in model5 significantly improved its validation performance and generalization capabilities compared to the original model. While both models achieved high training accuracy, model5 demonstrated a more stable validation accuracy and a modestly lower validation loss, indicating it was less prone to overfitting. The slightly higher test accuracy of model5 further emphasizes the effectiveness of incorporating dropout as a regularization technique. Overall, these results underscore the importance of employing strategies like dropout to enhance model robustness and performance on unseen data.