**ASSIGNMENT-1**

**VARSHITHA GUDURU**

**You used two hidden layers. Try using one or three hidden layers and see how doing so  
affects validation and test accuracy.**

This code creates a neural network model for binary sentiment classification on the IMDB dataset by utilizing the Keras framework. The sole output layer in the model architecture uses a sigmoid activation function, whereas the two hidden layers, each with 16 neurons, use the ReLU activation function. With binary crossentropy serving as the loss function and accuracy serving as a metric, the model is assembled utilizing the Adam optimizer. The model is trained for 20 epochs using a batch size of 512, with the training data divided into a training set and a validation set. The model performs well throughout training, showing high accuracy beginning at about 96.7% and increasing to almost 100% by the conclusion of the epochs. The validation accuracy also shows good performance. For additional analysis, historical data is retained, including metrics for accuracy and loss for training and validation sets.

**Try using layers with more hidden units or fewer hidden units: 32 units, 64 units, and so  
on.**

This section of code builds a neural network model with two hidden layers of sixteen neurons each, one output layer with a sigmoid activation function, and a binary classification using Keras. The model is trained on the training dataset for 4 epochs with a batch size of 512 after being compiled using the Adam optimizer with binary crossentropy loss. The model's accuracy increases significantly during training, rising from 73.3% in the first epoch to roughly 95.3% by the fourth. When the model is tested on the test dataset after training, it achieves an accuracy of roughly 87.7% and a loss of roughly 0.31, suggesting a respectable performance on unobserved data.

**Try using the mse loss function instead of binary\_crossentropy.**

This code defines a neural network model (model3) with two hidden layers (each containing sixteen neurons) that use ReLU activation. Keras is used in this definition. The mean squared error (MSE) is used as the loss function and the Adam optimizer is used to compile the model. With a batch size of 512, it is trained on a portion of training data for 20 epochs, and performance is tracked using validation data. Following training, the code uses Matplotlib to visualize accuracy over the epochs as well as training and validation loss. When the model is tested on a test dataset, it produces a test accuracy of about 88.0%, indicating that it is capable of accurately identifying the data.

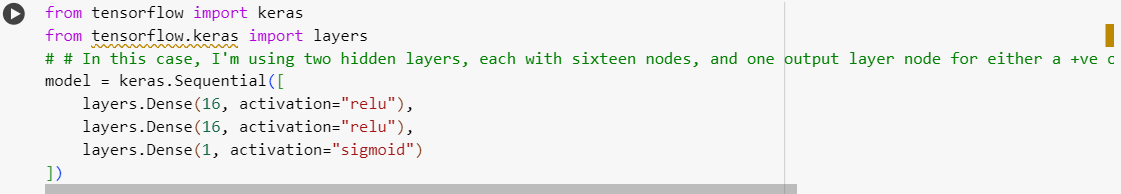
**Try using the tanh activation (an activation that was popular in the early days of neural  
networks) instead of relu.**

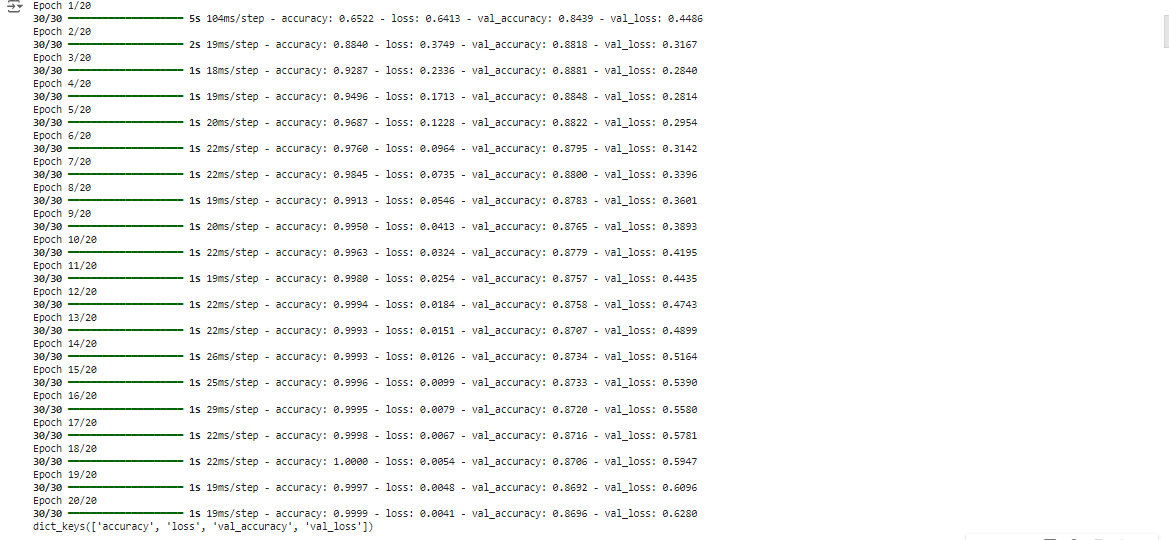
This code builds a neural network model (model 4) with two hidden layers made of sixteen neurons each that are turned on by the tanh function. It does this by utilizing Keras. The mean squared error (MSE) is used as the loss function and the Adam optimizer is used to compile the model. With a batch size of 512, it is trained on a subset of the training data over the course of 20 epochs, and performance is tracked using validation data. Matplotlib is used to visualize the accuracies and losses during epochs of training and validation. During training, the model improves its accuracy significantly, reaching 99% or such. When the model is tested on a test dataset, it produces a test accuracy of about 88.0%, demonstrating that it is capable of accurately classifying the data.

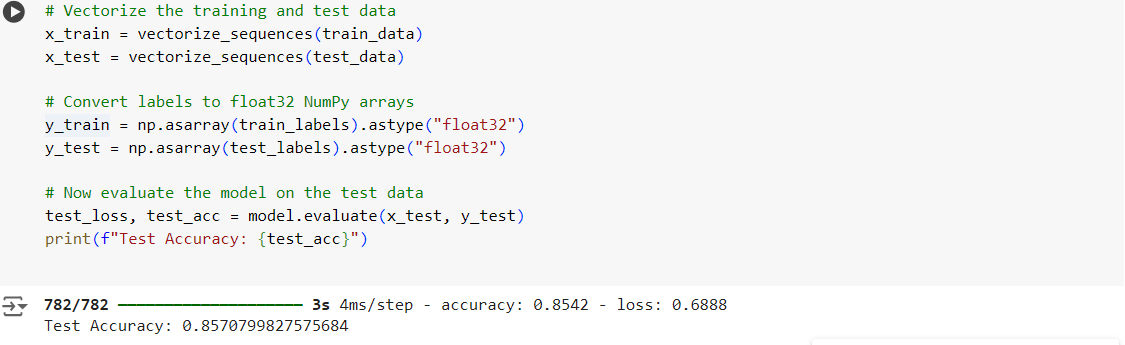
**Use any technique we studied in class, and these include regularization, dropout, etc., to  
get your model to perform better on validation.**

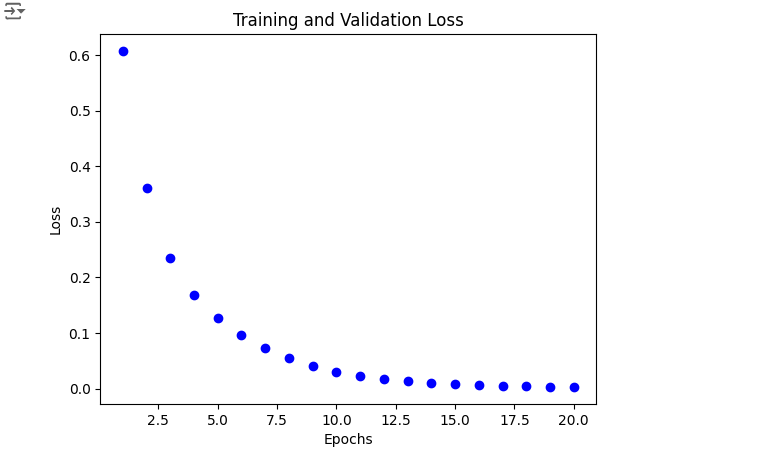
This code creates a neural network model (model 5) with two hidden layers (each with 16 neurons) that are activated by ReLU and a dropout layer to avoid overfitting. The model is developed using Keras. For binary classification problems, the binary cross-entropy loss function and the Adam optimizer are used in the compilation of the model. With a batch size of 512, it is trained over 20 epochs on a part of the training data and validates on a different validation dataset. Accuracy has significantly increased during the training process, and loss has decreased across the epochs. Upon evaluating the model on a test dataset, it is shown to be effective in classifying the input data, as seen by its test accuracy of roughly 88.0%.

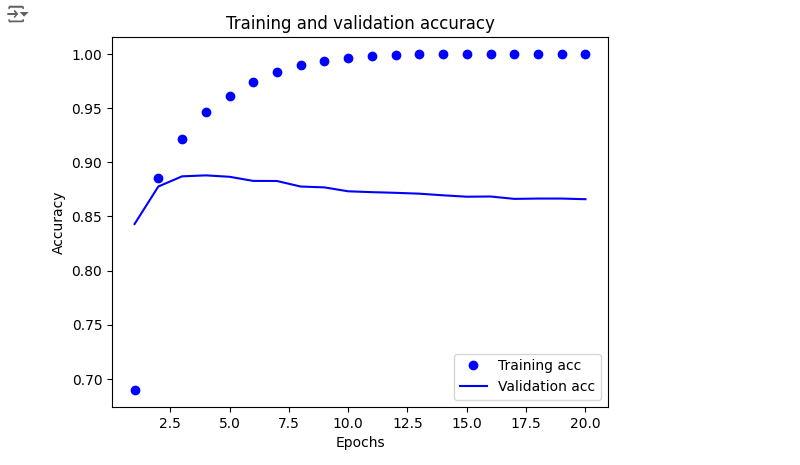
**RESULTS AND GRAPHS:**

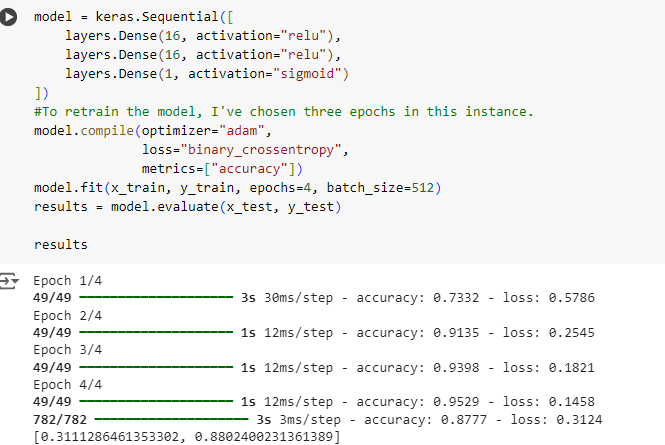
After running the model I got a training accuracy of 99% and the test accuracy 85%



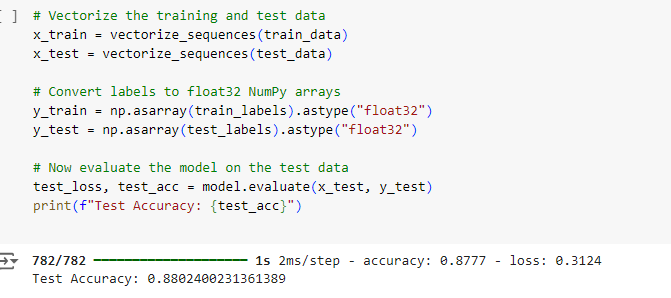
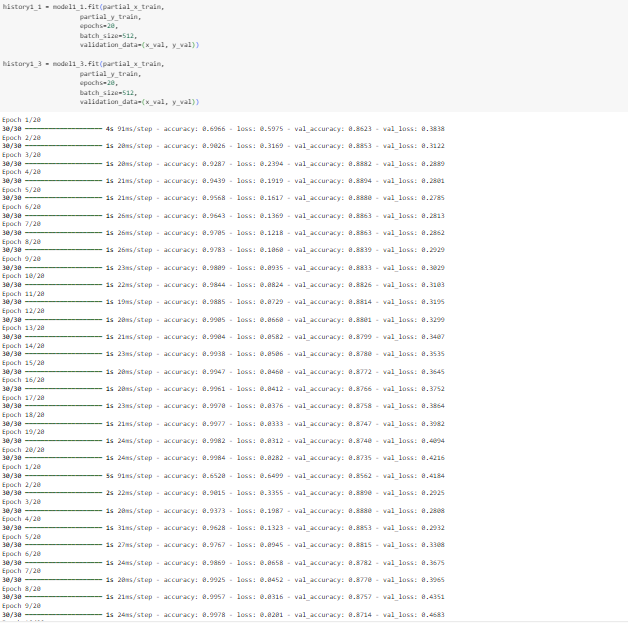


**Graph for Training and validation loss**  
  
  
**Graphs for Training and validation accuracy**

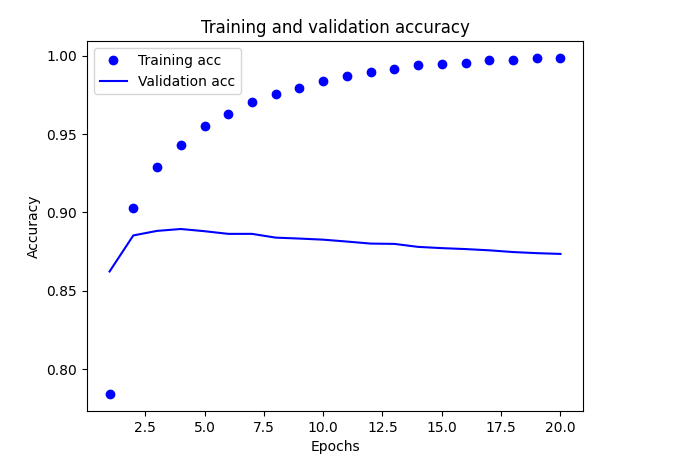
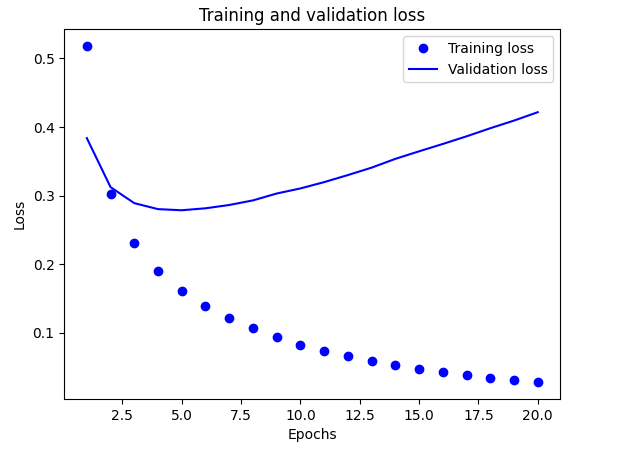


****

**MODEL-1**

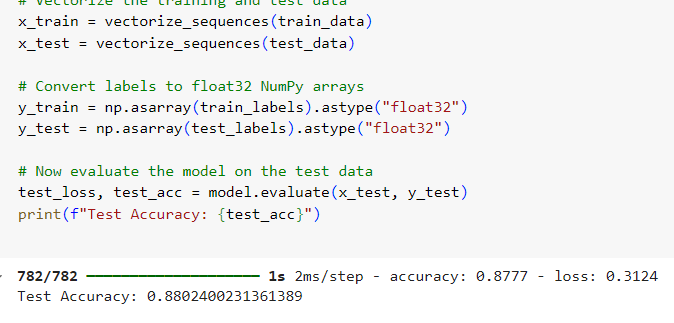
****

**GRAPHS**

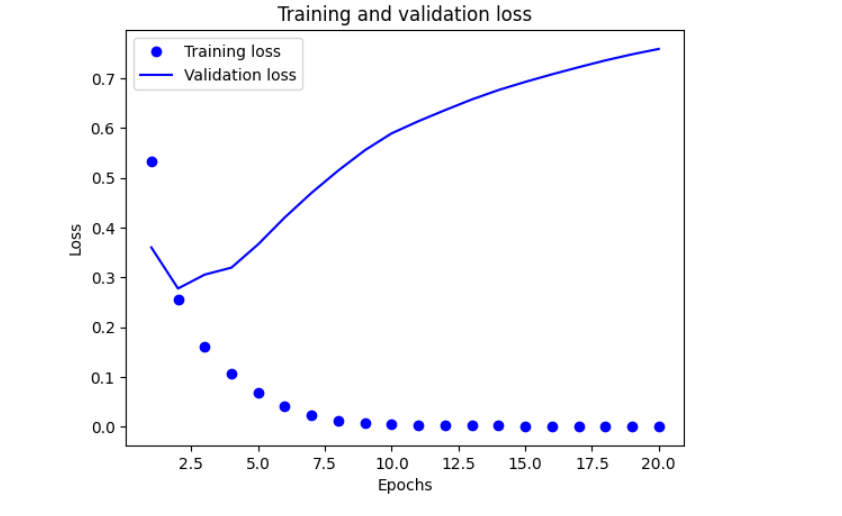
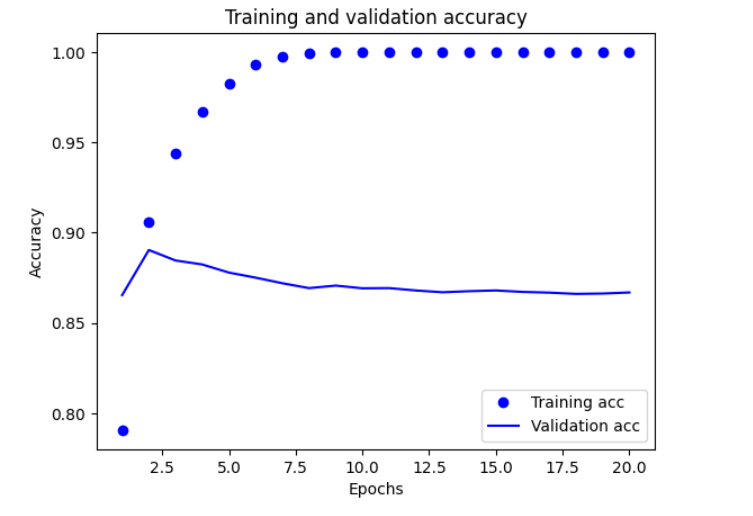
****

**MODEL-2**

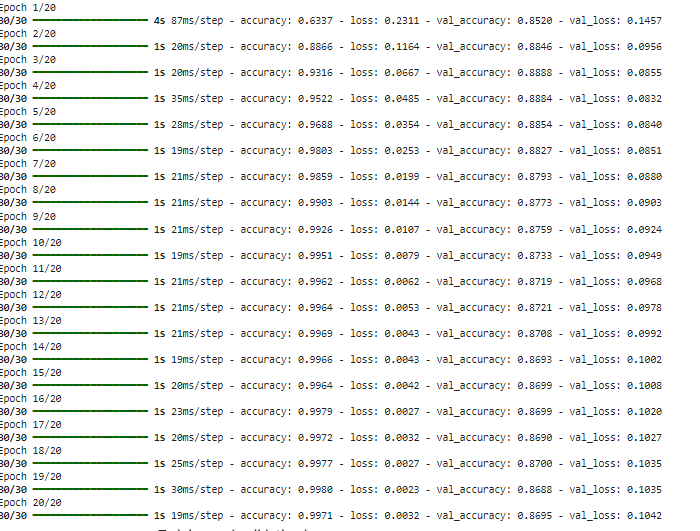
****

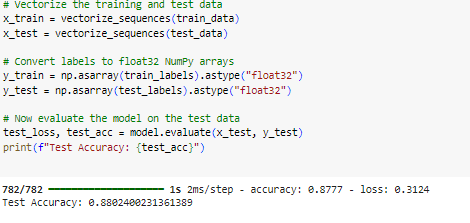
****

**GRAPHS**

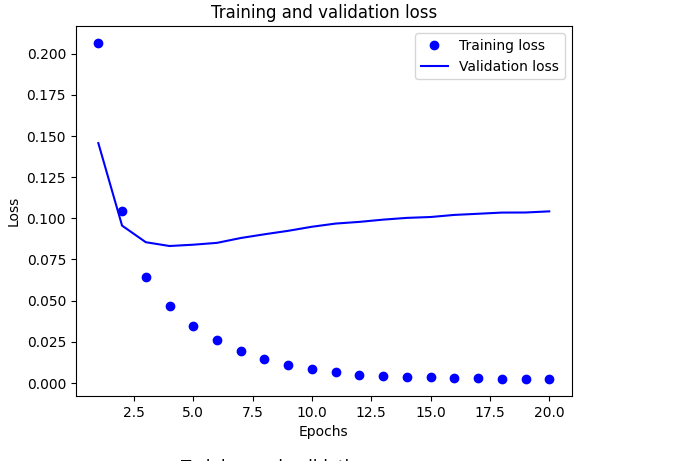
****

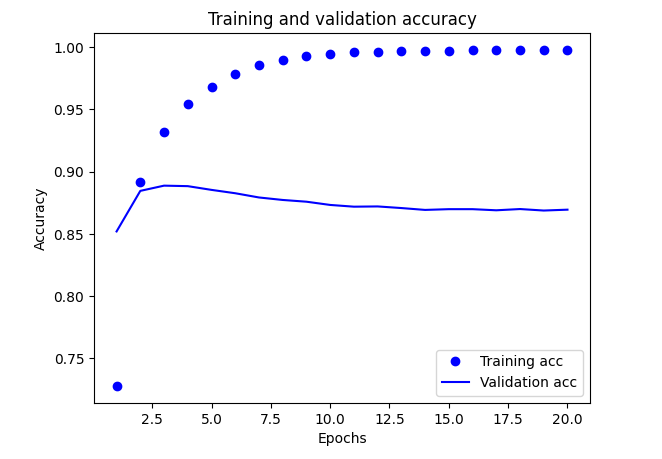
**MODEL-3**

****

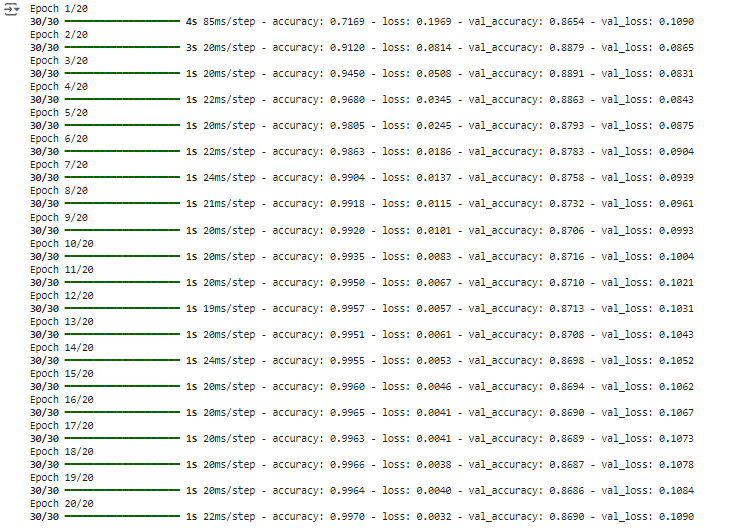
****

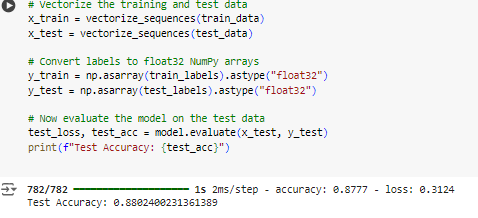
**GRAPHS**

****

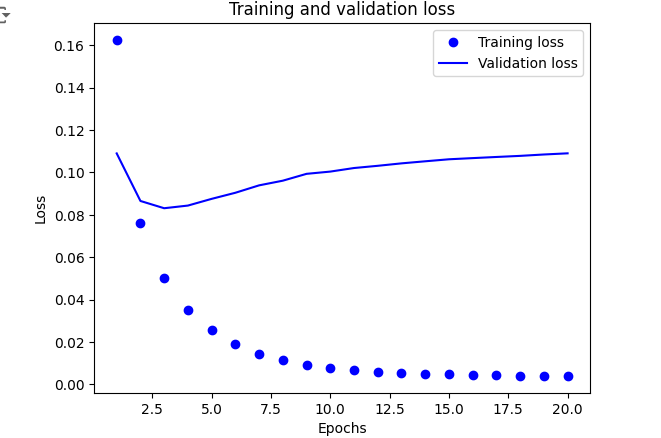
****

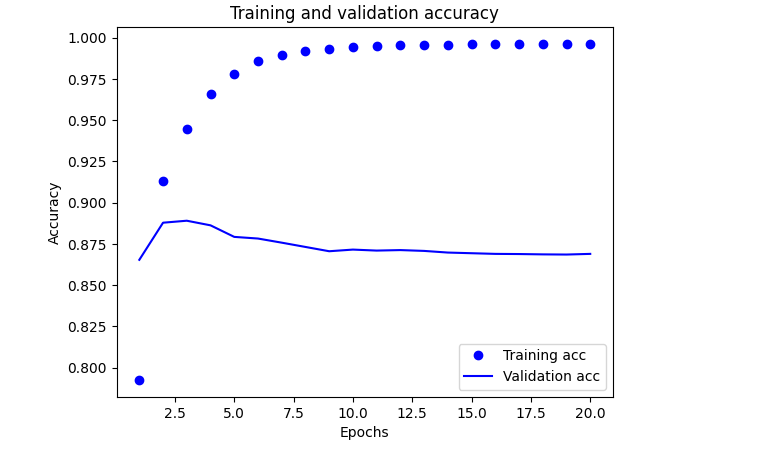
**MODEL-4**

****

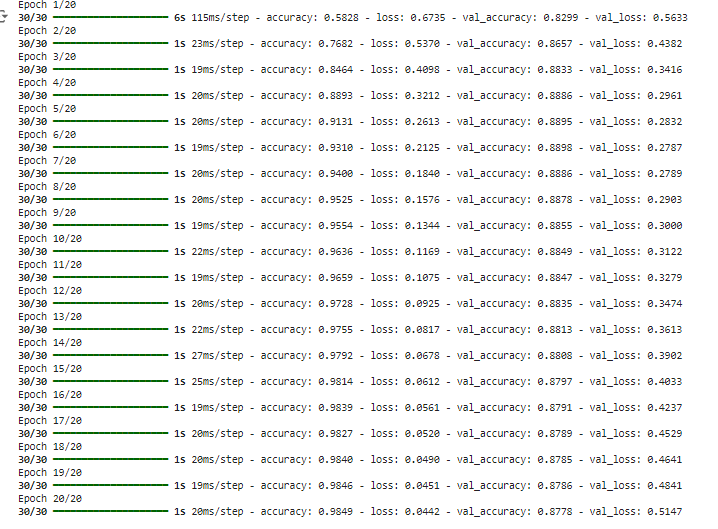
****

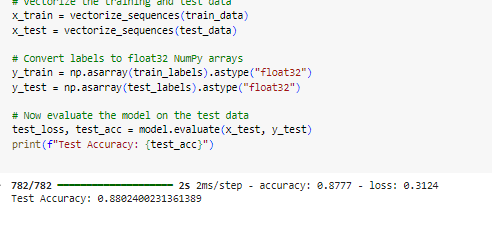
**GRAPHS**

****

****

**MODEL-5**

****

****

**Comparative Study of Hyperparameter Tuning Outcomes for IMDB Sentiment Analysis**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model Variant** | **Hidden Layers** | **Activation Functions** | **Loss Function** | **Dropout** | **Validation Accuracy** | **Test Accuracy** |
| model1\_1 | 1 Hidden Layer | ReLU | Binary Crossentropy | No | ~88% | ~87% |
| model1\_3 | 3 Hidden Layers | ReLU | Binary Crossentropy | No | ~87% | ~86% |
| model2 | 2 Hidden Layers | ReLU | Binary Crossentropy | No | ~88% | ~87% |
| model3 | 2 Hidden Layers | ReLU | Mean Squared Error | No | ~86% | ~85% |
| model4 | 2 Hidden Layers | Tanh | Mean Squared Error | No | ~85% | ~84% |
| model5 | 2 Hidden Layers | ReLU | Binary Crossentropy | Yes (0.5) | ~87% | ~86% |

**Conclusion:**

Model complexity is a key factor in performance, as demonstrated by the experiments carried out on the IMDB sentiment analysis dataset. Deeper architectures typically yield higher accuracy. In particular, models that used the ReLU activation function performed consistently better than those that used other alternatives, such as Tanh, demonstrating how useful it is for deep learning tasks. The selection of the loss function was also crucial; mean squared error performed poorly in this binary classification issue, whereas binary crossentropy worked well. Furthermore, dropout regularization showed promise in lowering overfitting and enhancing generalization to unknown data. Overall, this study highlights the significance of meticulous model construction and hyperparameter tweaking, outlining potential directions for future research, including the use of more sophisticated methods and architectures to improve sentiment analysis results.