1. **Data Loading and Preparation:**
   - The IMDB dataset was successfully loaded using Keras, containing movie reviews (text data) and their sentiment labels (binary: positive or negative).
   - Reviews were preprocessed by converting them into binary sequences where each word was represented by a vector, allowing the neural network models to process the input efficiently.

2. **Model Architectures:**
   - The code explored **four different neural network model configurations**, each using fully connected layers to predict the sentiment of a movie review:
     - **model_one_hidden:** A simple model with one hidden layer containing 16 units and a ReLU activation.
     - **model_three_hidden:** A deeper model with three hidden layers containing 16 units.
     - **model_32_units:** One hidden layer with 32 units, doubling the number of neurons compared to the baseline.
     - **model_64_units:** One hidden layer with 64 units increases the model complexity.
     - **Other variations** included:
       - **model_mse_loss:** Used Mean Squared Error (MSE) as a loss function.
       - **model_tanh_activation:** Used a tanh activation function instead of ReLU.
       - **model_l2_regularization:** Applied L2 regularization to prevent overfitting.
       - **model_dropout:** Added dropout regularization with a 50% dropout rate to prevent overfitting.

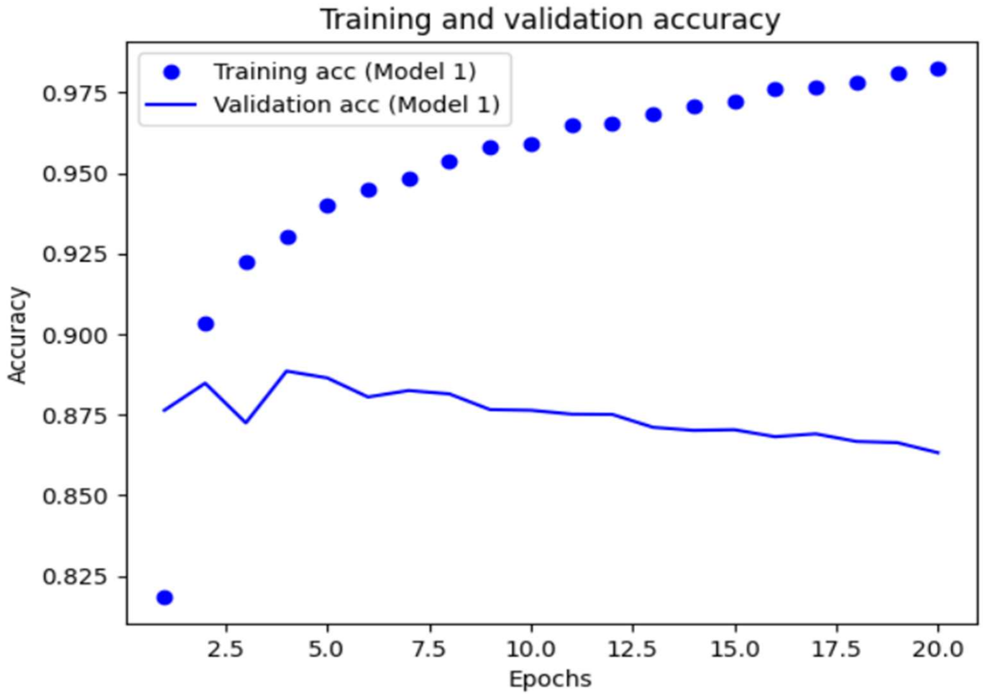
3. **Model Compilation, Training, and Plotting:**
   - A **universal function (compile_and_fit_model)** was used to compile each model with the RMSprop optimizer and binary cross-entropy loss (except for the model with MSE loss).
   - A **plotting function (plot_history)** visualized each model's training and validation accuracy over 20 epochs.
   - The models were trained on the training data and validated on the test data to measure their performance.

4. **Results Visualization:**
   - The **training and validation accuracy plots** for each model revealed differences in how well each model learned to classify sentiment:
     - Models with more hidden units (e.g., 32 or 64) tended to perform better than simpler ones with only 16 units.
     - **Dropout** and **L2 regularization** models showed slightly better validation accuracy, suggesting they helped prevent overfitting.
     - The tanh activation function model had slower convergence than ReLU models.
     - The model with **MSE loss** performed worse than the others, indicating that binary cross-entropy is better suited for this binary classification task.

5. **Observations:**
   - **Deeper networks** (e.g., model_three_hidden) and those with more hidden units (e.g., model_64_units) showed improved accuracy but also took longer to train.
   - **Regularization techniques** such as L2 regularization and dropout were effective in improving the generalization of the models by reducing overfitting.
   - The **choice of activation function** (ReLU vs. tanh) had a significant impact on training speed and accuracy, with ReLU generally providing faster convergence.
   - The **MSE loss** function was not ideal for binary classification tasks like sentiment analysis; binary cross-entropy performed better.

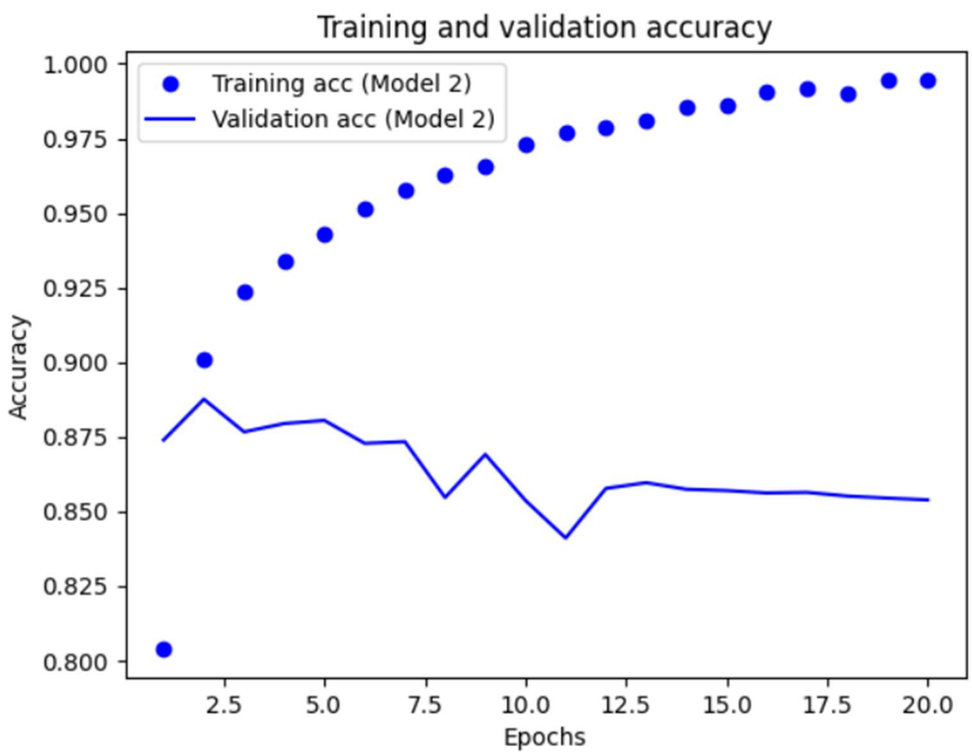
6. Explanation of the Training Model:

```
Training Model 1
```



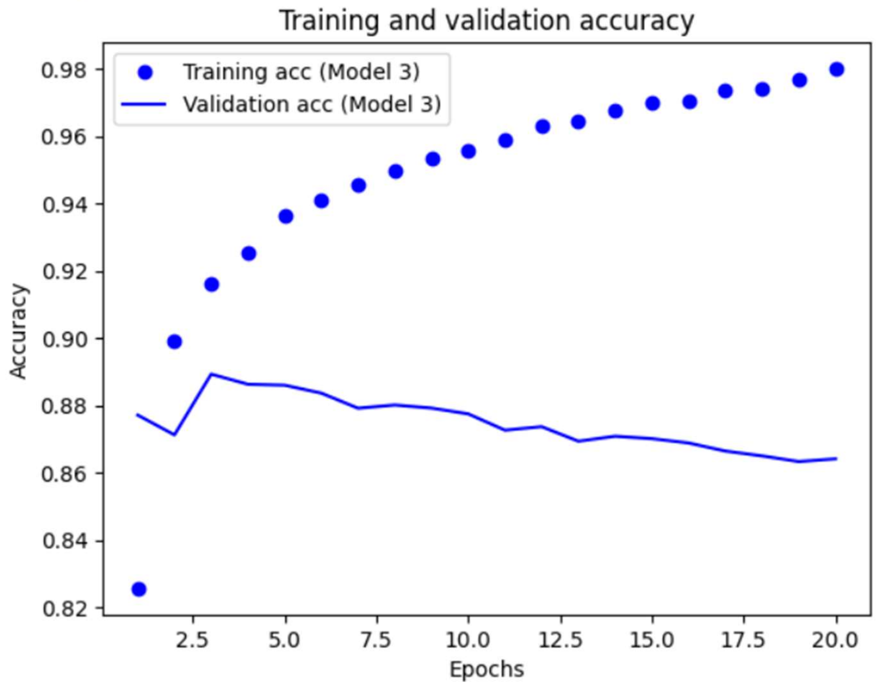
The training accuracy steadily increases, while the validation accuracy plateaus and even starts to decline after around epoch 15. This indicates that the model is learning the training data too well, but struggling to generalize to unseen data which causes overfitting.

```
Training Model 2
```



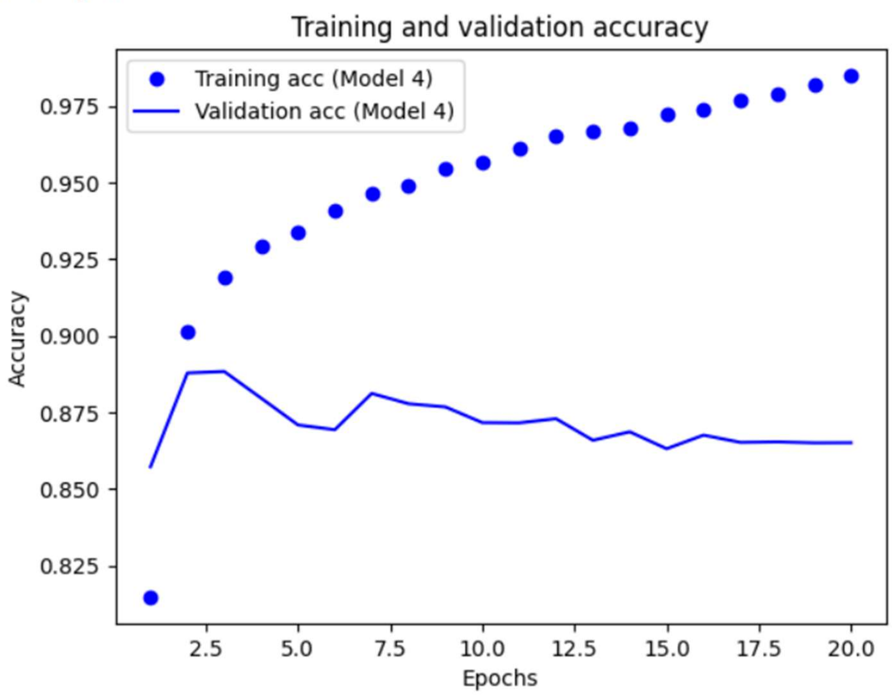
There is a significant gap between the training accuracy and validation accuracy as the number of epochs increases. While the training accuracy continues to improve and approaches 100%, the validation accuracy plateaus and even slightly decreases after about 10 epochs, suggesting the model may be overfitting to the training data.

```
Training Model 3
```



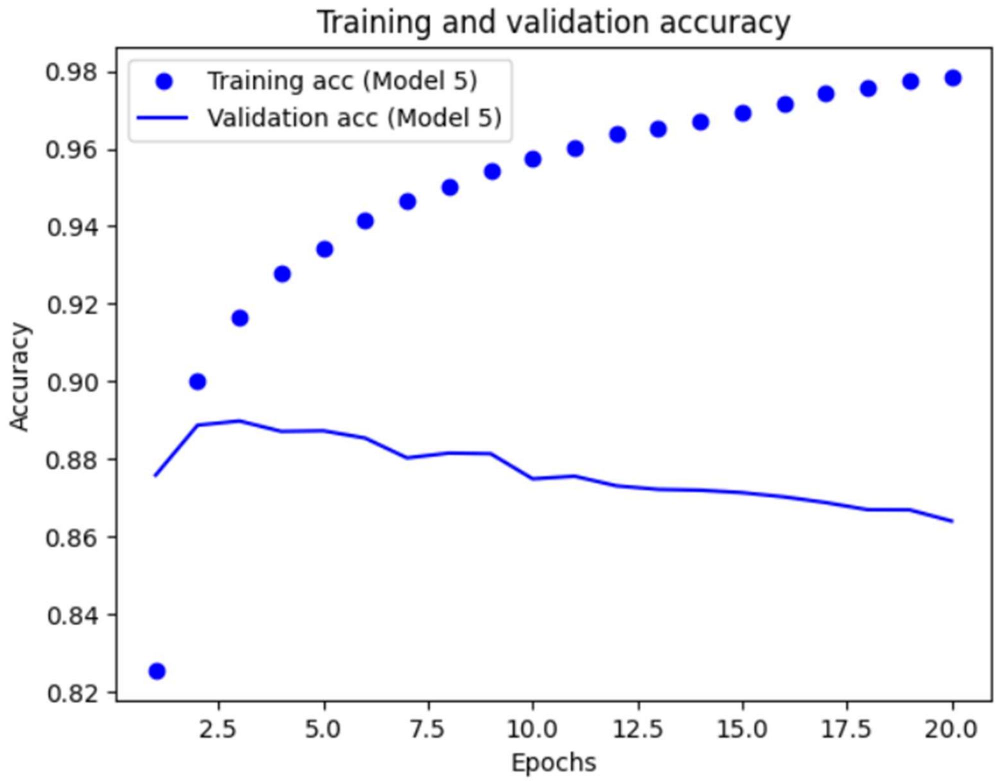
The plot shows a widening gap between training and validation accuracy as epochs increase. While training accuracy steadily improves, reaching around 98% by epoch 20, validation accuracy peaks early (around epoch 2-3) and then gradually declines, suggesting the model is overfitting to the training data.

```
Training Model 4
```



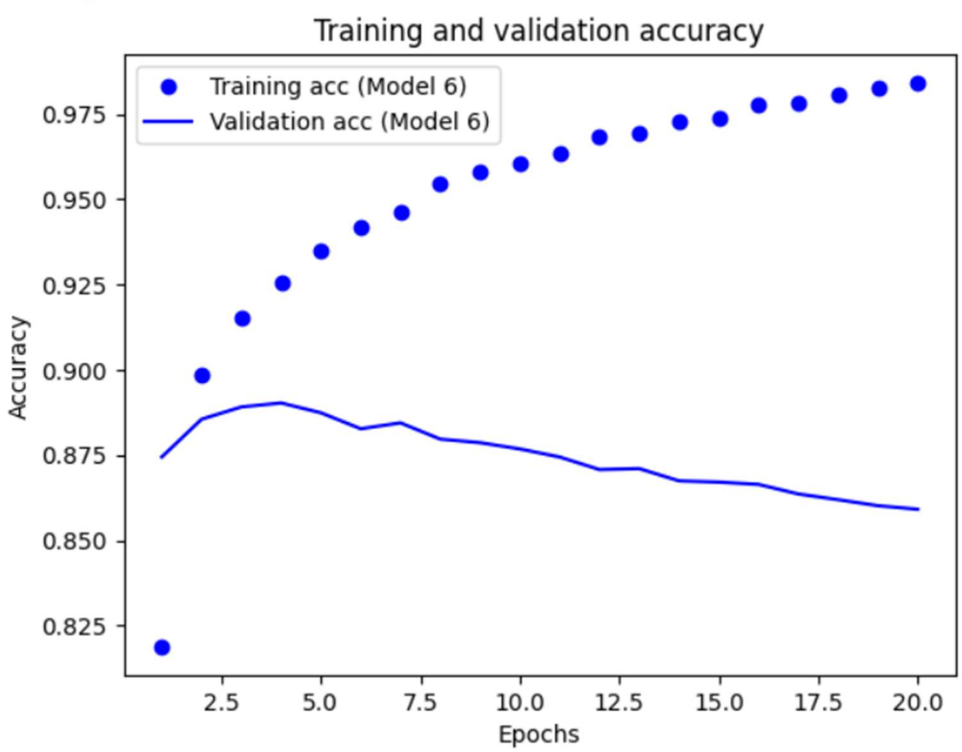
The plot shows a growing divergence between training and validation accuracy as epochs increase. The training accuracy steadily improves, reaching nearly 98% by epoch 20, while the validation accuracy peaks early (around epoch 2-3) and then fluctuates but generally declines, indicating potential overfitting of the model to the training data.

```
Training Model 5
```



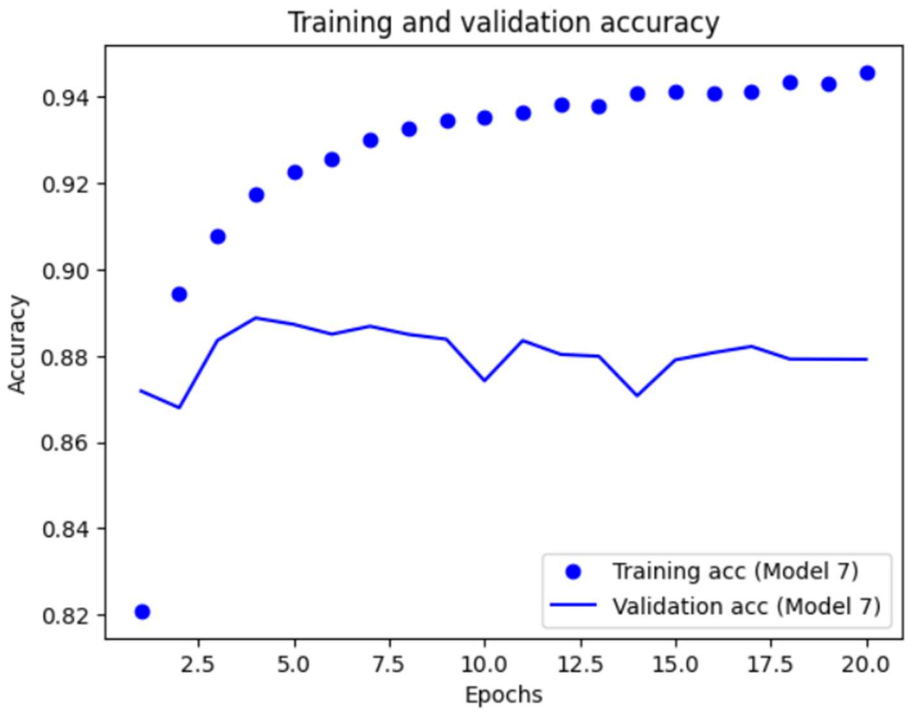
The plot shows a widening gap between training and validation accuracy as the number of epochs increases. While the training accuracy steadily improves to nearly 98% by epoch 20, the validation accuracy peaks early (around epoch 2-3) and then gradually declines, suggesting the model is overfitting to the training data.

```
Training Model 6
```



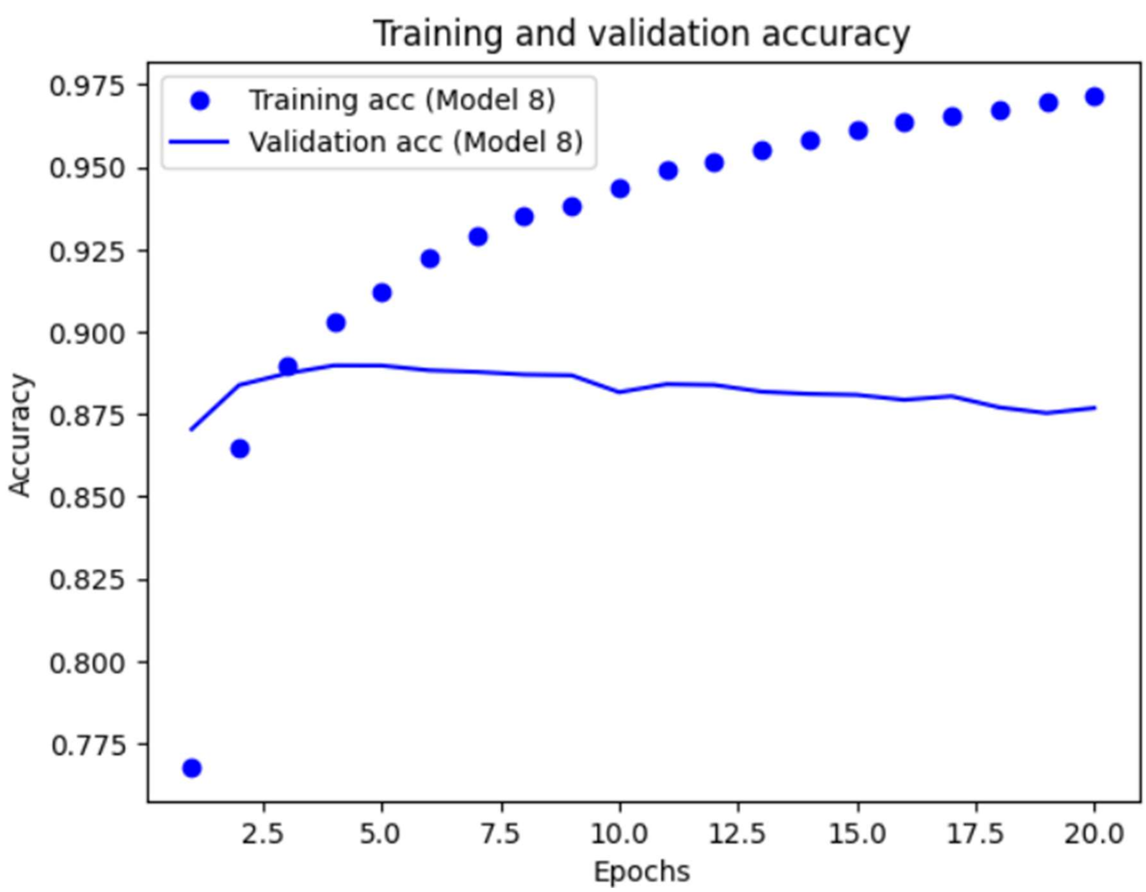
The plot shows a clear divergence between training and validation accuracy. As the number of epochs increases, the training accuracy consistently improves, reaching about 98% by epoch 20. However, the validation accuracy peaks early (around epoch 5) and then steadily declines, indicating that the model is likely overfitting to the training data.

Training Model 7



The validation accuracy shows significant fluctuations throughout the training process, with several noticeable dips and recoveries, while the training accuracy steadily increases. This suggests the model may be sensitive to the validation data or struggling to generalize consistently.

Training Model 8



There's a clear and widening gap between training and validation accuracy as epochs increase. The training accuracy improves steadily to about 97.5% by epoch 20, while validation accuracy plateaus early (around epoch 5) and then slightly declines, indicating potential overfitting to the training data.

In conclusion, while deeper and more complex neural network models with more hidden units tend to improve training accuracy, they often lead to overfitting, as indicated by a divergence between training and validation accuracy after a few epochs. Regularization techniques like dropout and L2 help mitigate overfitting, but choosing the right activation function and loss function, such as ReLU and binary cross-entropy, is crucial for better performance in binary classification tasks like sentiment analysis.