SUMMARY REPORT

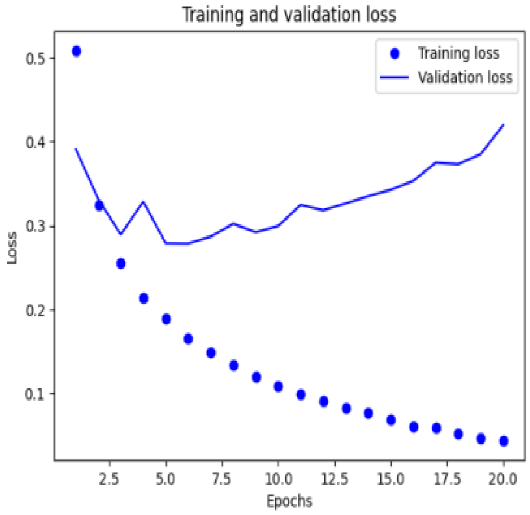
ASSIGNMENT-2

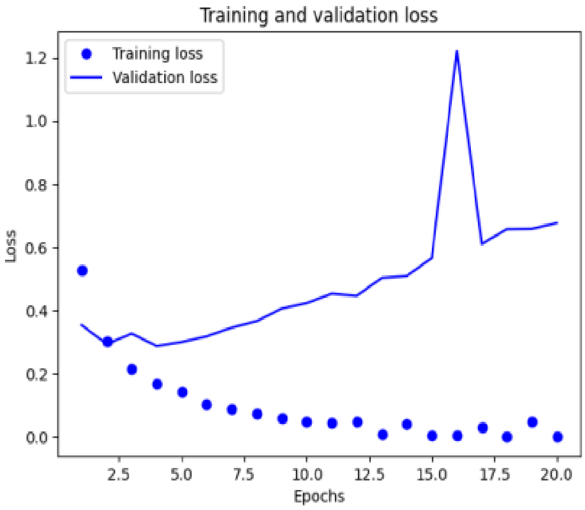
Introduction:

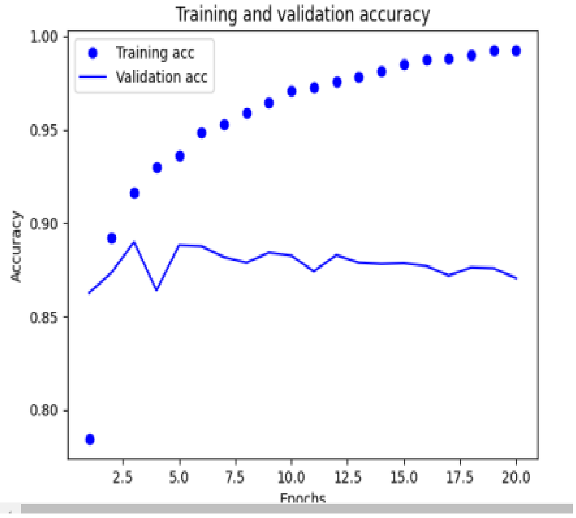
In this exercise, different neural network configurations were tested on the IMDB dataset, including changes to hidden layers, hidden units, activation functions, and loss functions. Despite employing regularization techniques like L2 regularization and Dropout, the model showed signs of overfitting after several epochs, as the validation accuracy plateaued and the validation loss began to rise.

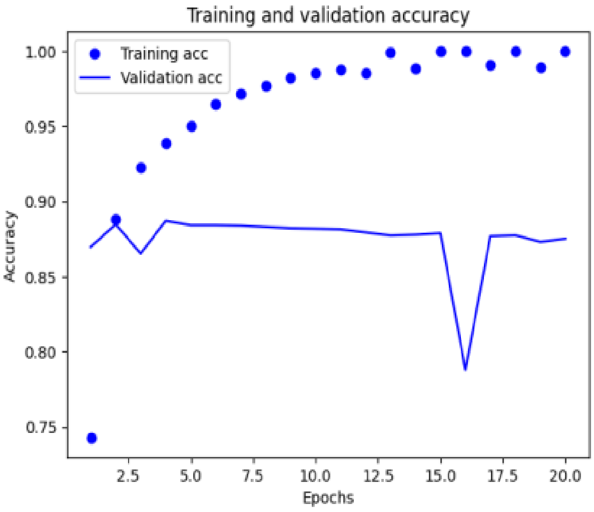
1. You used two hidden layers. Try using one or three hidden layers and see how doing so affects validation and test accuracy.

1 hidden layer- 3 hidden layers-









One Hidden Layer

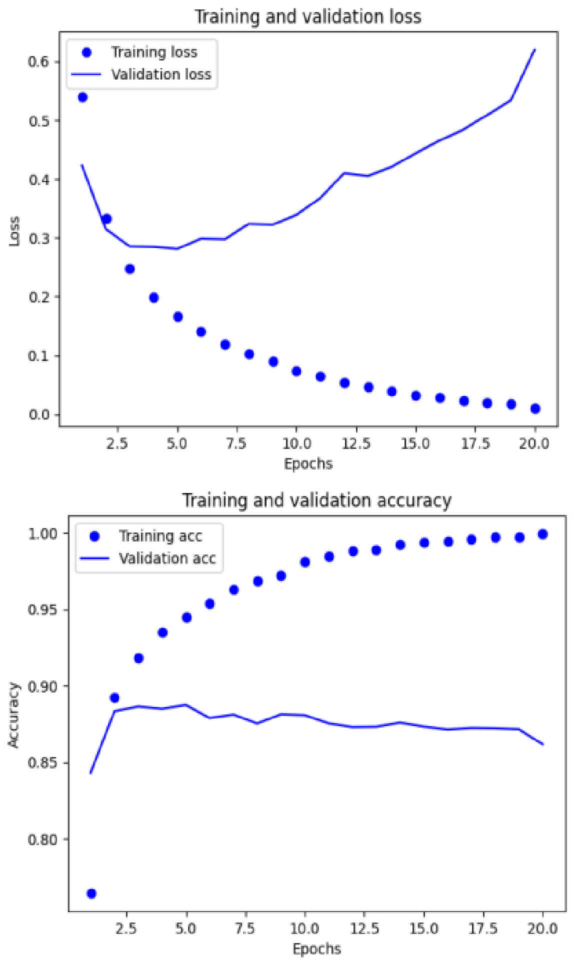
- Training Accuracy: By the final epoch, the model reached approximately 99.4% accuracy on the training set.
- Validation Accuracy: The validation accuracy topped out at around 88.9%.
- Training and Validation Loss: While the training loss consistently decreased, the validation loss showed some fluctuations, suggesting possible overfitting.

Three Hidden Layers

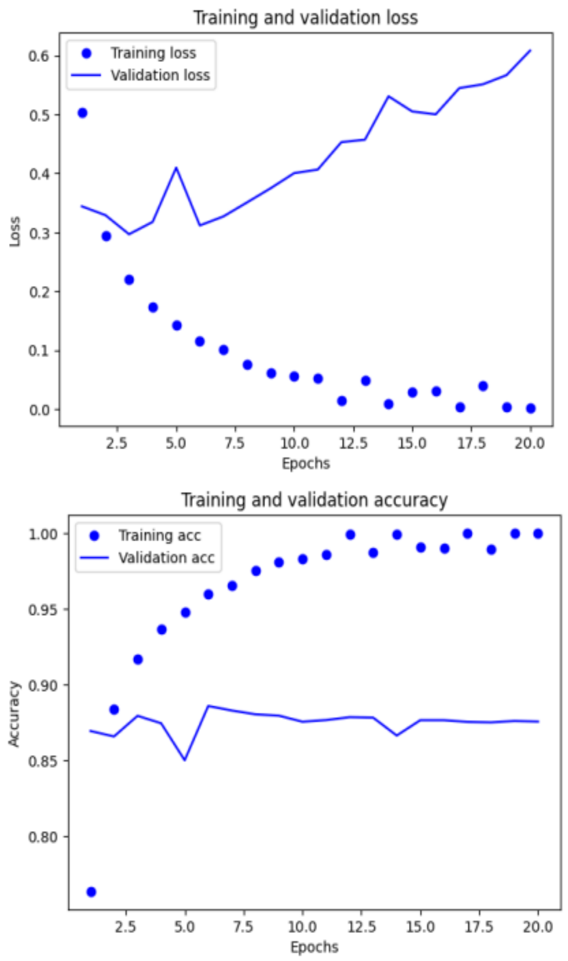
- Training Accuracy: The model achieved 99.9% accuracy, indicating a strong fit to the training data.
- Validation Accuracy: Despite this, the validation accuracy was slightly lower than the model with one hidden layer, reaching a peak of about 88.5%.
- Validation Loss: The validation loss became unstable, particularly near the end, suggesting the model may have overfit to the training data.

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

Fewer hidden units(32 units)-



More hidden units(64 units)-



Fewer Hidden Units (16 units)

- Training Accuracy: The model attained a training accuracy of 99.98%.
- Validation Accuracy: The highest validation accuracy reached approximately 88.7%.
- Loss: The training loss steadily decreased, while the validation loss exhibited fluctuations, ending at about 0.6196.

More Hidden Units (64 units)

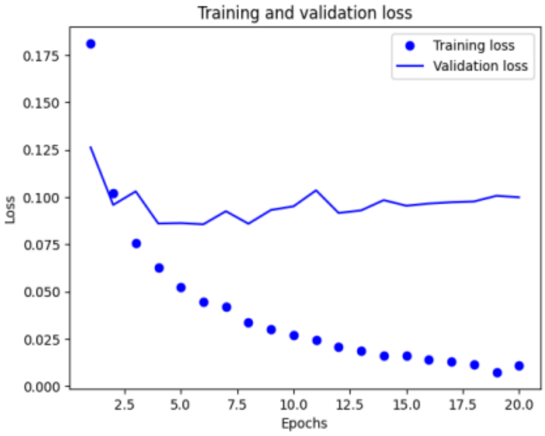
- Training Accuracy: Reached a training accuracy of 100% by the final epoch.
- Validation Accuracy: The validation accuracy peaked at around 88.6%, slightly lower than the model with fewer units.

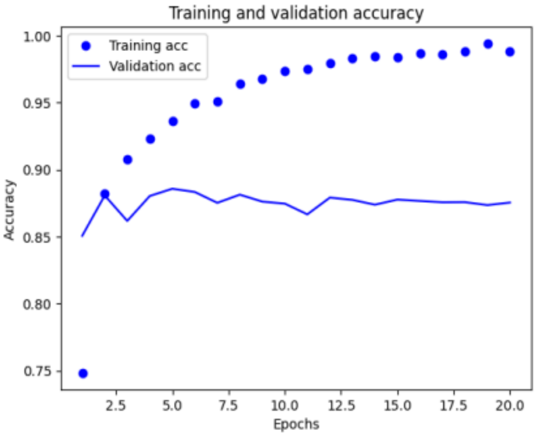
- Loss: The validation loss showed some instability, with a final loss value of around 0.6087.

Observations

1. Overfitting: Both models exhibited signs of overfitting, as evidenced by the gap between training and validation accuracy, particularly in the model with more units that achieved perfect training accuracy.
2. Loss Stability: The model with fewer hidden units demonstrated a slightly more stable loss trajectory compared to the model with more hidden units.

Try using the mse loss function instead of binary_crossentropy.





Training Results

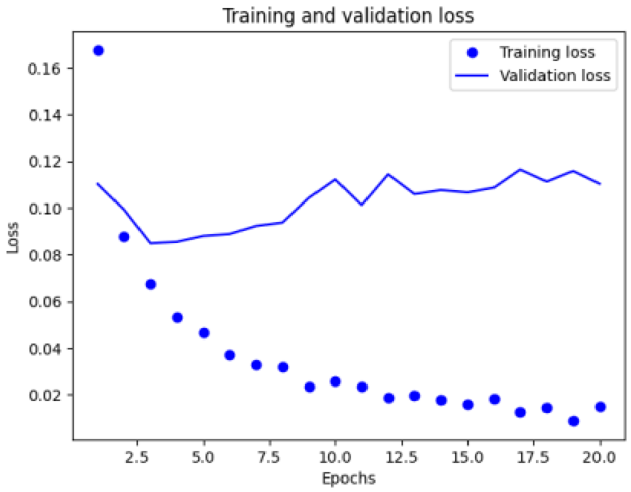
- Training Accuracy: The model reached a maximum training accuracy of about 99.4%.
- Validation Accuracy: The highest validation accuracy was roughly 88.8%.
- Training Loss: The training loss consistently decreased, finishing at around 0.0135 by the end of training.
- Validation Loss: The validation loss showed minor fluctuations, concluding at approximately 0.0999.Observations

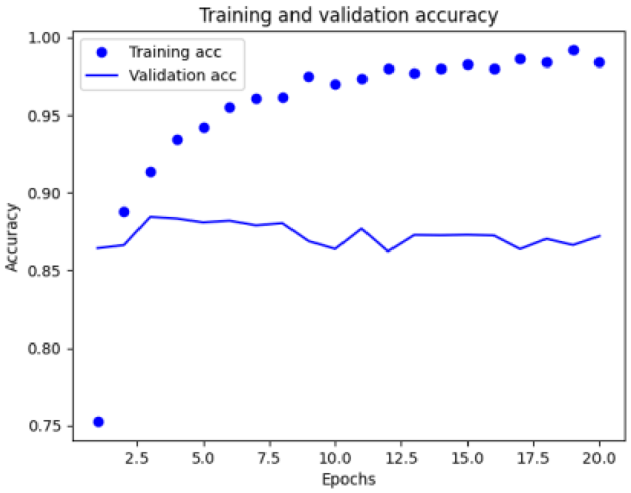
1.Performance with MSE: Although MSE is generally intended for regression tasks, it can be utilized for binary classification. However, its accuracy may not be as effective as that achieved with binary cross-entropy.

2.Validation Stability: The validation accuracy stayed fairly consistent, but there were no significant improvements throughout the epochs.

3.Overfitting: Similar to previous models, this model exhibits signs of overfitting, as indicated by a considerably higher training accuracy compared to validation accuracy.

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





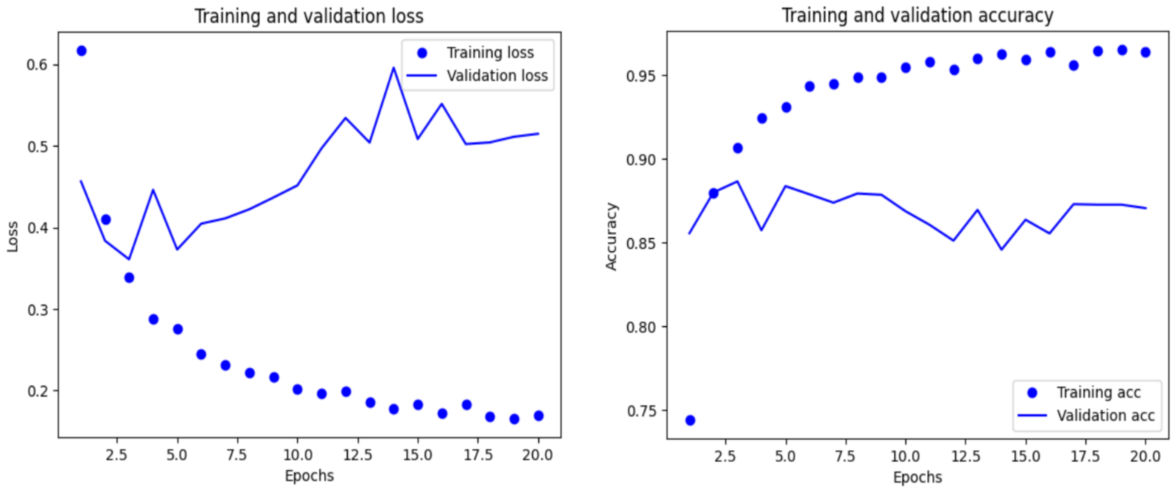
Training Results

- Training Accuracy: Reached a peak accuracy of around 99.2%.
- Validation Accuracy: The best validation accuracy was about 88.1%.
- Training Loss: Training loss decreased consistently, reaching around 0.0229 by the end.
- Validation Loss: The final validation loss was approximately 0.1103.

Observations

1. Tanh vs. ReLU: The tanh activation function can alleviate problems related to dying neurons more effectively than ReLU, as it produces outputs within the range of -1 to 1. However, its performance may still be affected by how the inputs are initialized and scaled.
2. Validation Stability: The validation accuracy showed minor fluctuations, similar to previous models, indicating a consistent performance but room for improvement.
3. Overfitting: While the training accuracy remains high, the validation accuracy does not follow suit as closely, suggesting some overfitting may still be occurring.

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



Training Results

- Training Accuracy: Peaked at around 97.3%.
- Validation Accuracy: Best validation accuracy reached about 88.0%.
- Training Loss: Decreased steadily, finishing at around 0.1583.
- Validation Loss: Ended at approximately 0.5147.

Observations

1. Dropout Effect: The use of dropout seems to stabilize the validation accuracy compared to previous models, though it did not result in a notable enhancement in validation performance.

2. Validation Fluctuations: The validation loss exhibits some fluctuations, indicating that the model continues to experience overfitting despite the use of dropout and regularization.

3. L2 Regularization: This technique has contributed to reducing the loss and sustaining a satisfactory validation accuracy; however, further adjustments (such as modifying the regularization strength) may lead to improved outcomes.

Overall, models with both fewer and more hidden units exhibited similar performance, showing no significant advantages from increasing the number of hidden units, though all models experienced overfitting. The model using MSE as the loss function had slightly inferior performance compared to the one using binary cross-entropy, while the model with the tanh activation function performed similarly to the one using relu.