**The below excel is the summary of all models**

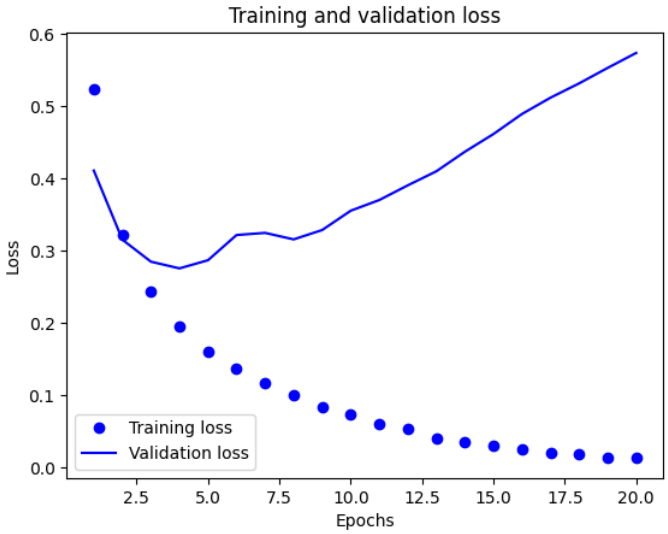
| # | | layers | nodes accordingly | epochs | batch | optimal stop learning point(around) | optimal train loss(around) | optimal validation loss(around) | optimal validation accuracy(around) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | model from textbook | 3 | 16,16,1 | 20 | 512 | 5 epochs | 0.1602 | 0.2866 | 0.8849 |
| 2 | first model from myself | 4 | 16,16,16,1 | 20 | 512 | 5 epochs | 0.1578 | 0.2874 | 0.8831 |
| 3 | second model from myself | 3 | 64,64,1 | 20 | 512 | 3 epochs | 0.231 | 0.276 | 0.8891 |
| 4 | third model from myself, please be advised, mse loss function is deployed instead of binary_crossentropy | 3 | 16,16,1 | 20 | 512 | 5 epochs | 0.0615 | 0.0844 | 0.8878 |
| 5 | fourth model is all the same like second model from myself, but replace relu with tanh as the activation function | 3 | 64,64,1 | 20 | 512 | 3 epochs | 0.2044 | 0.2766 | 0.8877 |
| 6 | model with dropout | 3 | 16,16,1 | 20 | 512 | 6 epochs | 0.2843 | 0.2758 | 0.8903 |

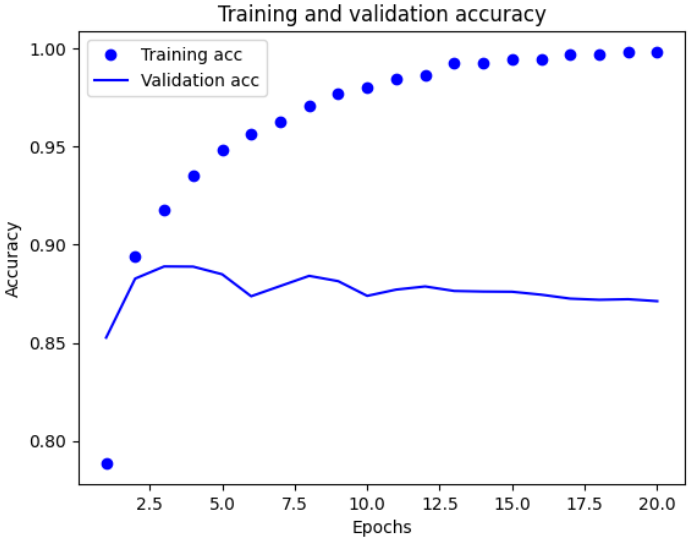
Based on my data, the model 6 which is with dropout add-in provides the highest accuracy as 0.8903 for validation data. I will suggest building the model with dropout add-in.

**Original model set-up from textbook as**

Three layers with 16,16,1 node(s) and relu, relu, sigmoid as the activation functions. Also, the model chooses optimizer as rmsprop, loss function as binary_crossentropy, and metrics as accuracy.

The plots for training and validation loss and accuracy as below:




As the graph shows the training loss keeps going down but with validation loss goes down and turns up at around 5 epochs. This indicates that 5 epochs could be the optimal end-learn point, with more epochs model starting overfitting.

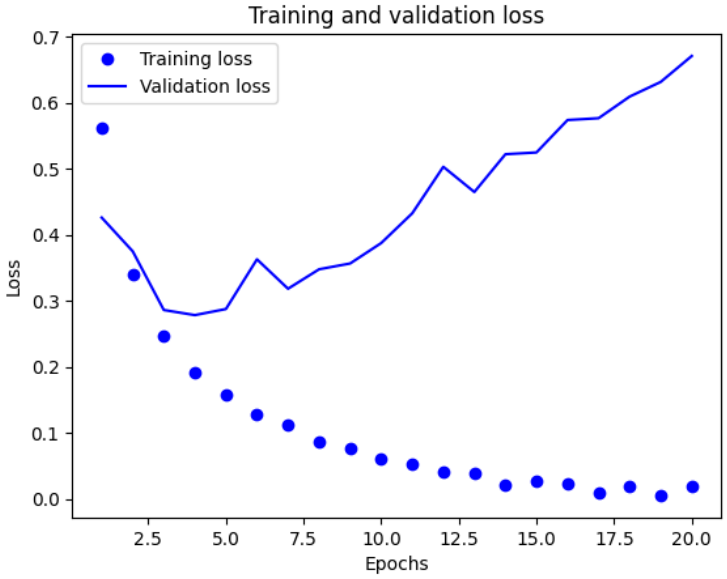
Plot about accuracy reflects the same idea that around 5 epochs the model hits optimal.

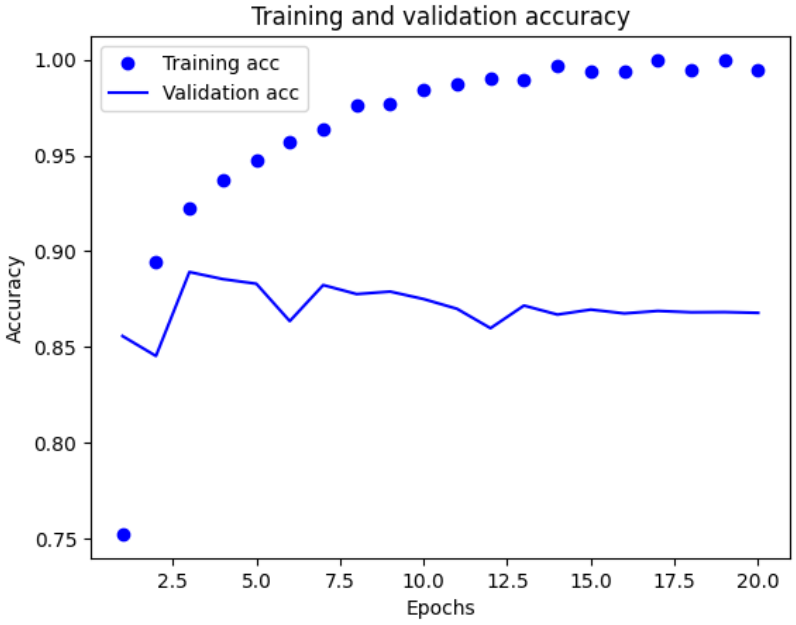
**Model set-up by myself**

**The first model from me is set as**

4 layers with 16, 16, 16, 1 node(s); relu, relu, relu and sigmoid as the activation functions.

The plots for training and validation loss and accuracy as below:



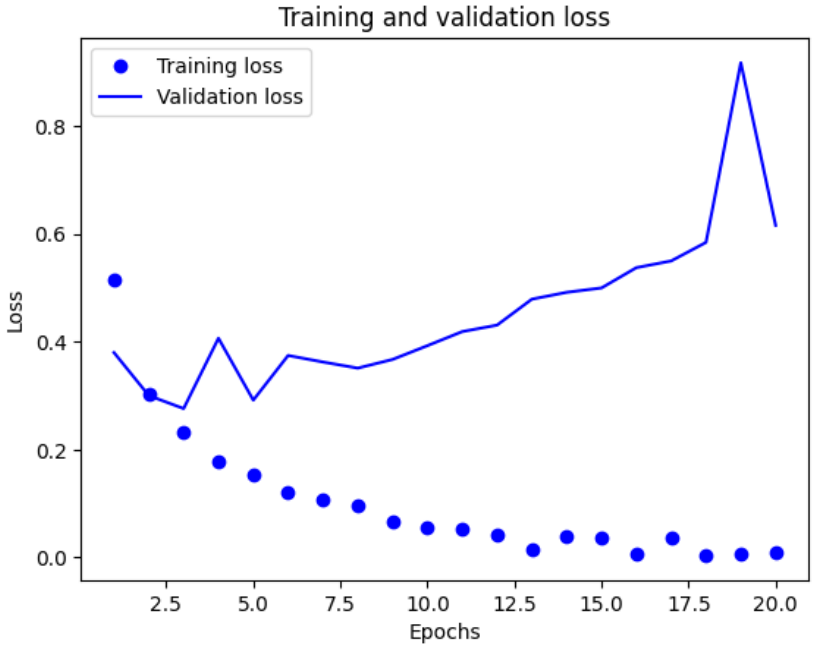

This plot indicates that optimal epochs for learning is around 5 with regarding minimum validation loss.

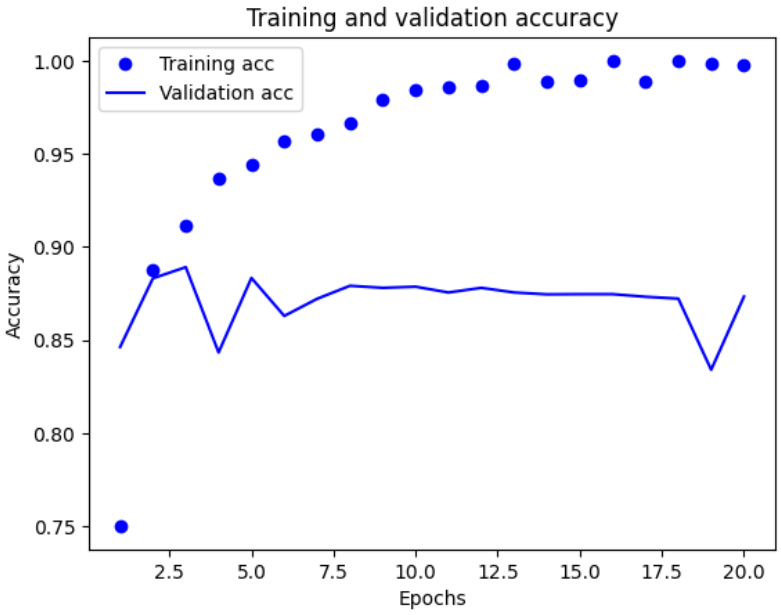
From the plot about accuracy, we can have the same line in conclusion that stop at around epochs 5 provides the maximum validation accuracy.

**The second model from me is set as**

3 layers with 64, 64, 1 node(s); relu, relu and sigmoid as the activation functions.

The plots for training and validation loss and accuracy as below:



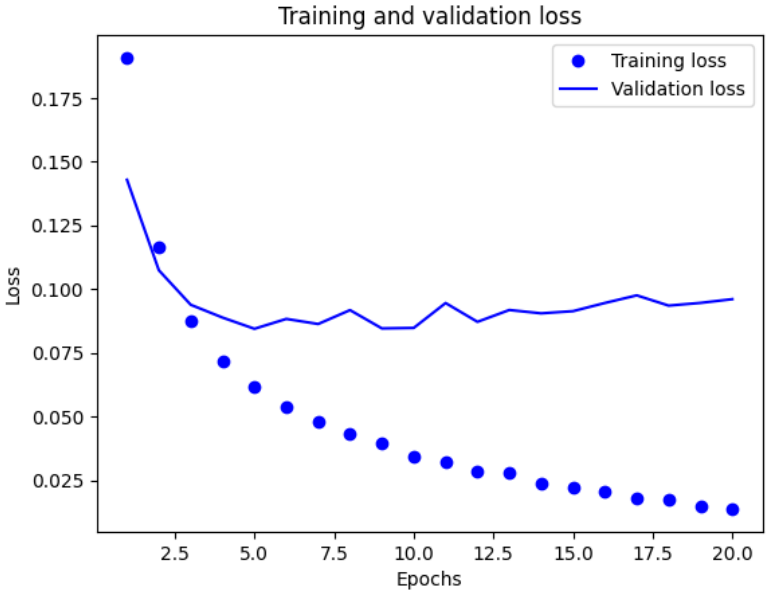



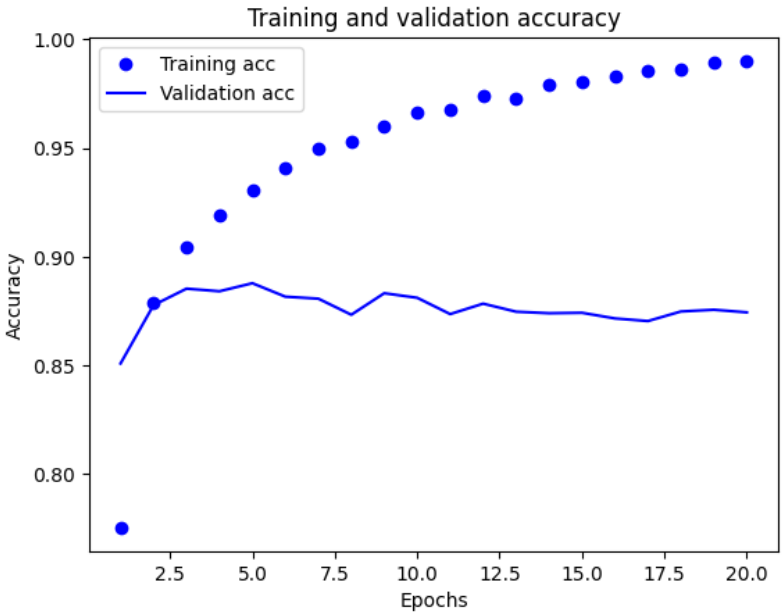

The plots indicates that 3 epochs are the optimal learning point.

**The third model from me is set as**

3 layers with 16, 16, 1 node(s); relu, relu and sigmoid as the activation functions. But loss function as MSE instead of binary_crossentropy.

The plots for training and validation loss and accuracy as below:



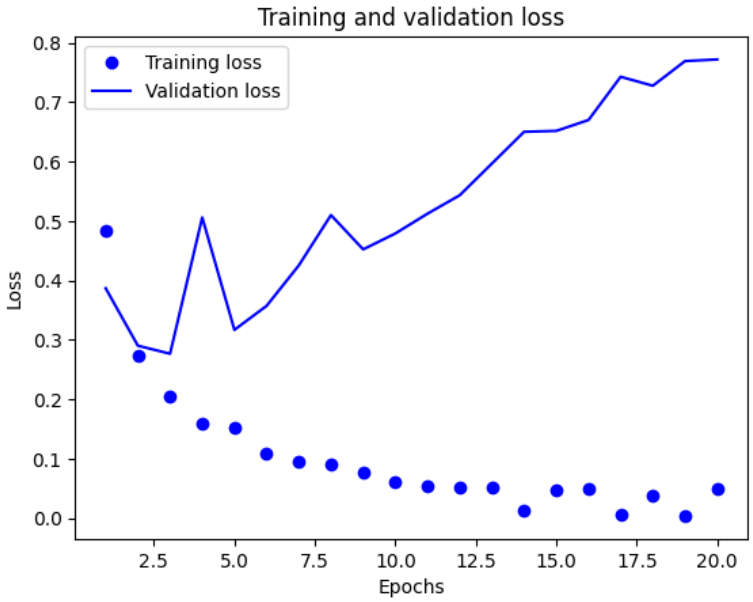



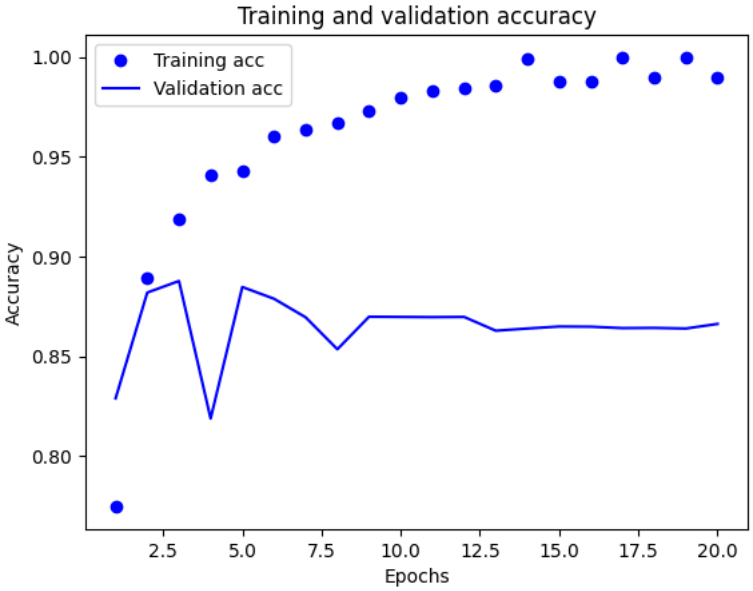

The interesting point for this model is that with the same training loss all the way goes down, the loss value of validation goes down and turn back at 5 epochs but the bounce of this turnback is not strong enough, this indicates that the model is overfitting and learning too much noise.

**The fourth model from me is set as**

3 layers with 64, 64, 1 node(s); tanh, tanh and sigmoid as the activation functions.

The plots for training and validation loss and accuracy as below:




With changing the activation function to tanh, these plots indicates that the optimal point is at 3 epochs. The result is like the second model, regardless which activation function has been used in the model.

The last model added dropout to check if the performance will better.

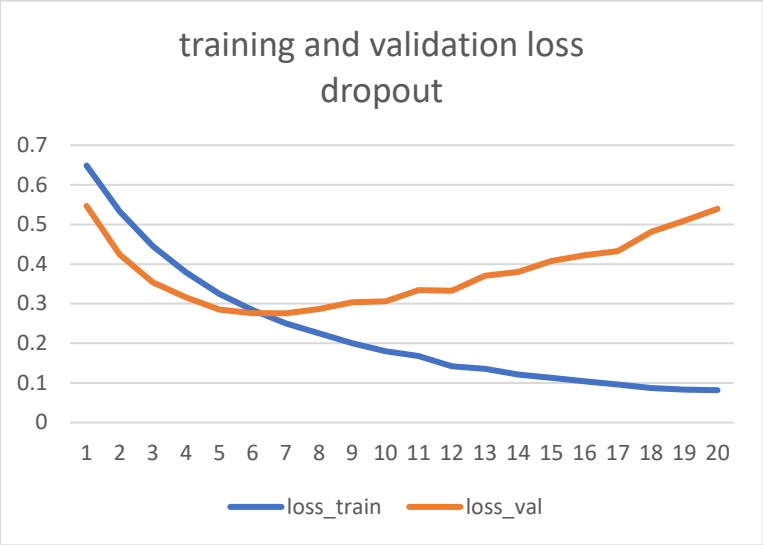
Model is set up as

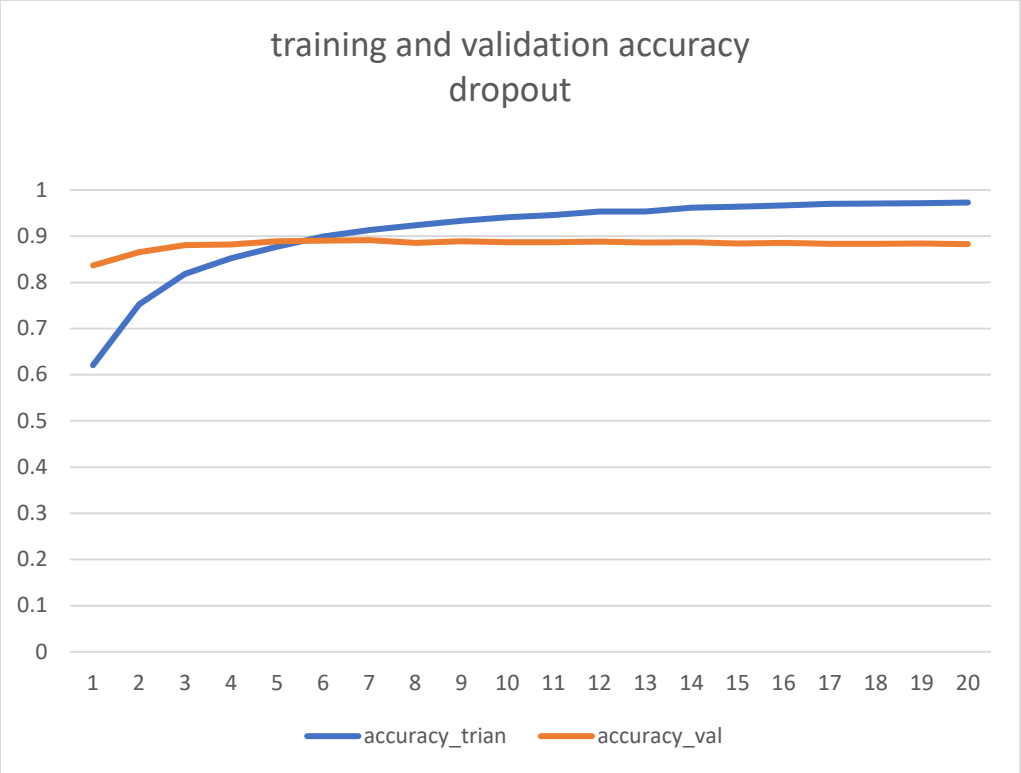
3 layers with 16,16,1 node(s), relu, relu, relu as the activation function.

Details about setting dropout as below:

```
model = keras.Sequential([
    layers.Dense(16, activation="relu"),
    layers.Dropout(0.5),
    layers.Dense(16, activation="relu"),
    layers.Dropout(0.5),
    layers.Dense(1, activation="sigmoid")
```

The plots for training and validation loss and accuracy as below:






Loss value plot indicates that the optimal epochs is around 6 and the validation accuracy line is flat, and I can only tell the optimal numbers from the output in python, the optimal accuracy is 0.8903.