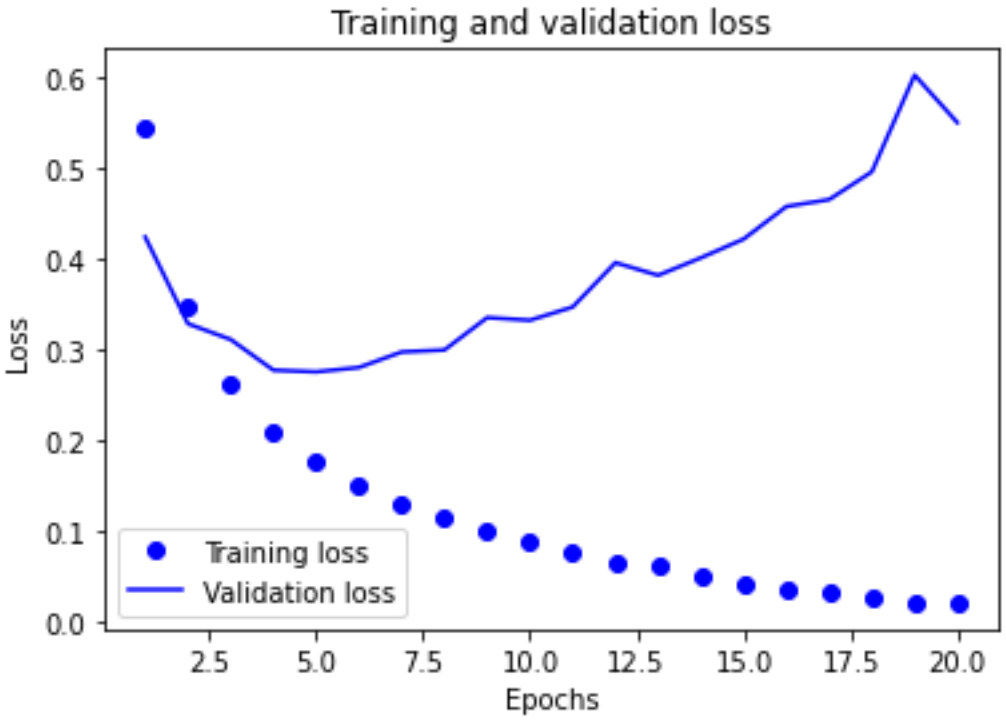
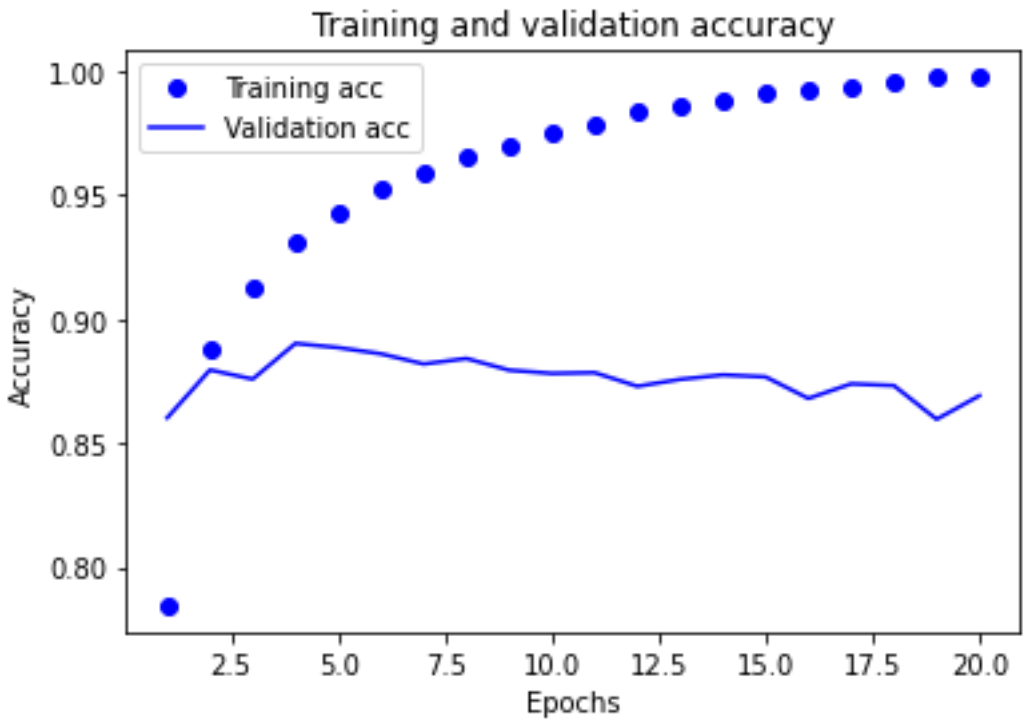
Overview report on the project

After running multiple tests on the dataset I realized that there are multiple ways to modify and improve its performance. From changing the structure by adding or removing layers, adjusting the number of hiddent units or utilizing a different activation function.to better optimize the model.

Adding layers and increasing the number of hidden units can increase the complexity of the model and can improve performance, but can also lead to overfitting. Every activation function is better suited to a specific type problem, and your choice of usage will have an effect on the performance of the model.

Original unmodified dataset

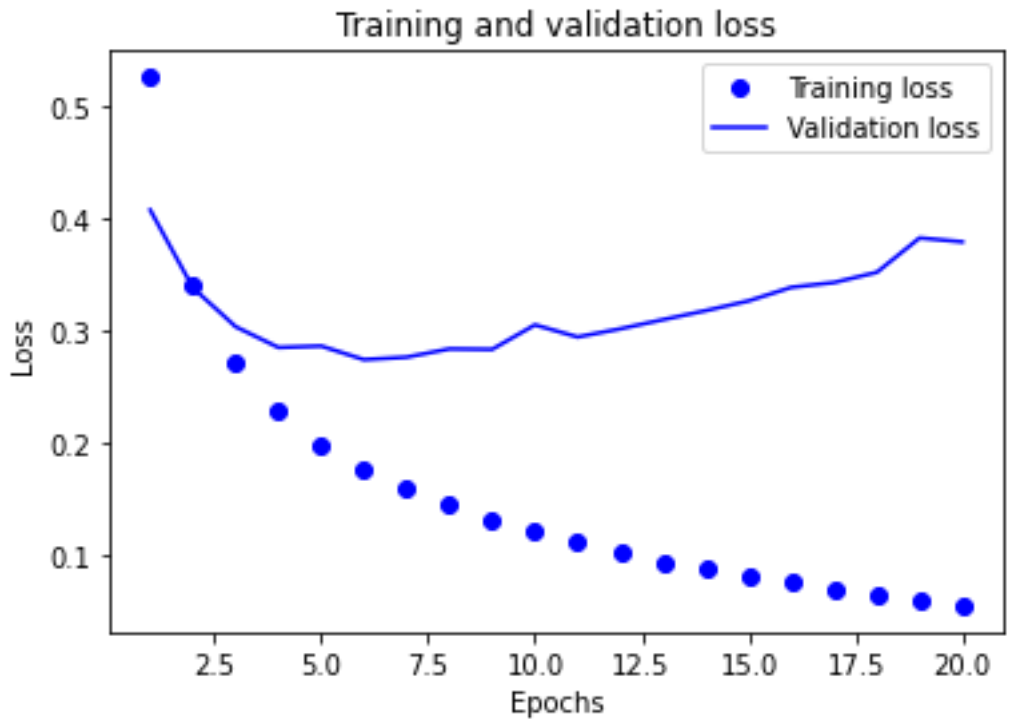


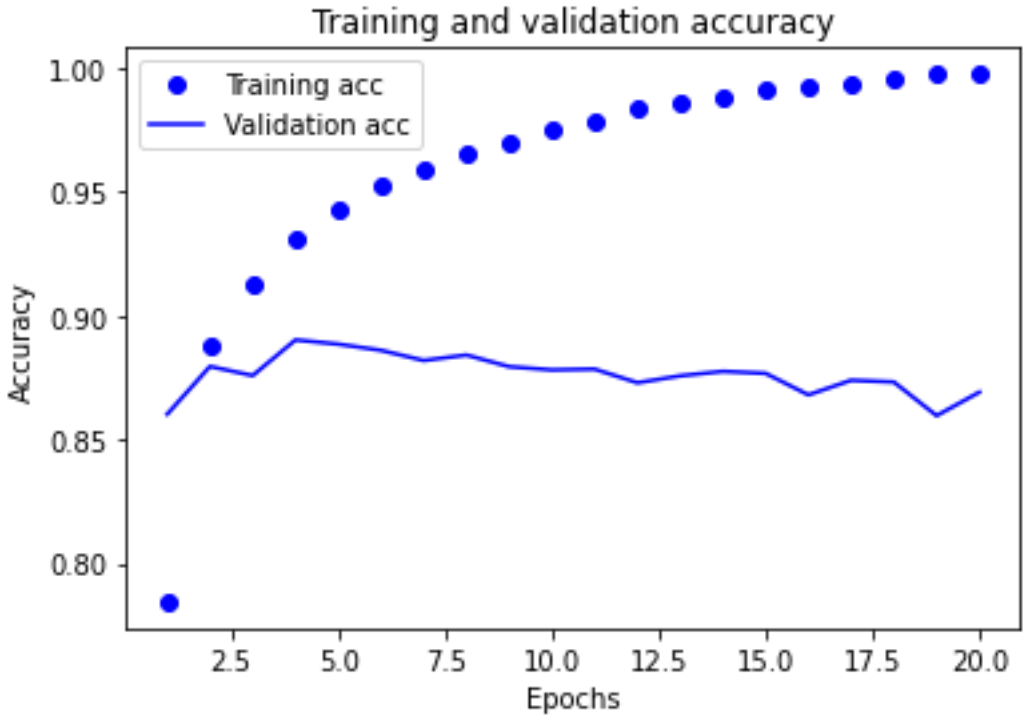


1. **Using varying amounts of hidden layers**

When adding more hidden layers. The complexity of the neural network increase and this leads to a better performance of the training data, but I realized that the more layers I added the greater the model was at risk of overfitting, because the model was becoming too complex.

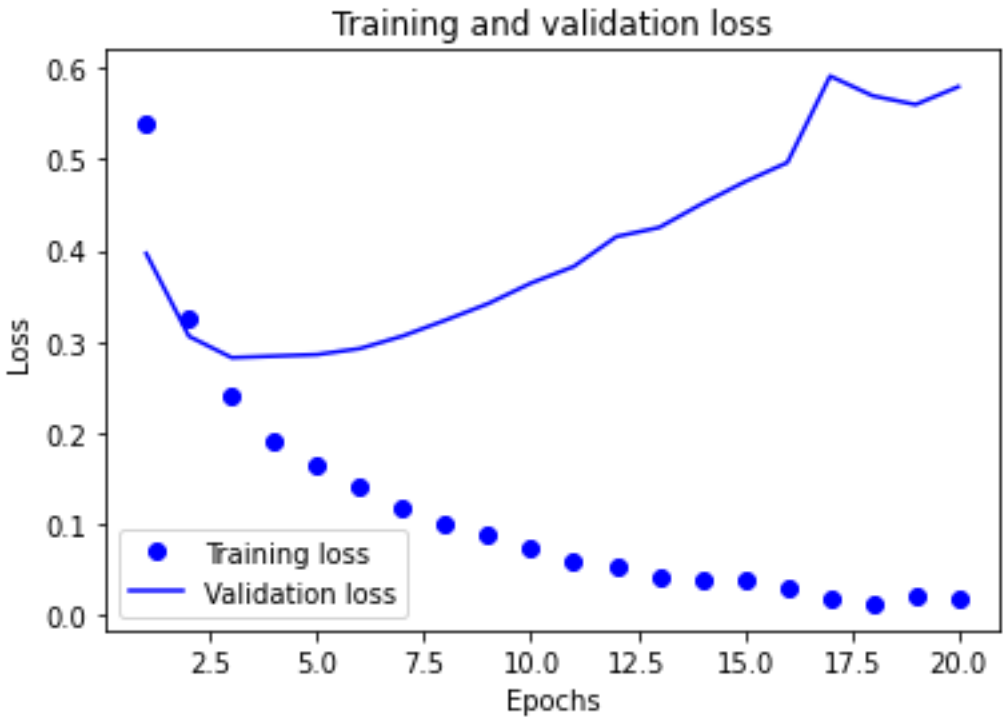
From my understanding the addition of hidden layers should be based on how complex is the project going to be, including the size of the training data and what resources the analyst will have available. In order to avoid overfitting, it's critical to use the correct regularization technique.

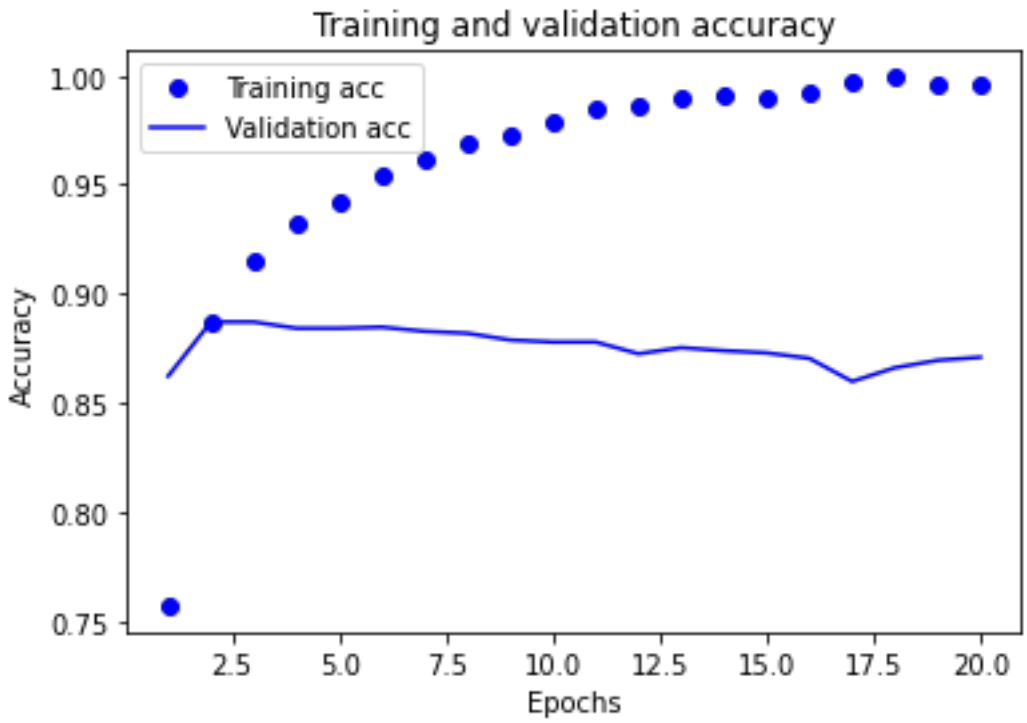

Training and validation loss
Training loss
Validation loss
Loss
Epochs
0.5
0.4
0.3
0.2
0.1
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Training and validation accuracy
Training acc
Validation acc
Accuracy
Epochs
1.00
0.95
0.90
0.85
0.80
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

2. **Using varying numbers of hidden units**

I started off using 16 units and worked my way up until the performance on the validation began to downgrade. I realized the number of units had a similar effect to the hidden layers. With 16 units the model was underfitting.
Which also concludes the number of the units should also be related to the complexity and size of the training data to ensure a perfect model.

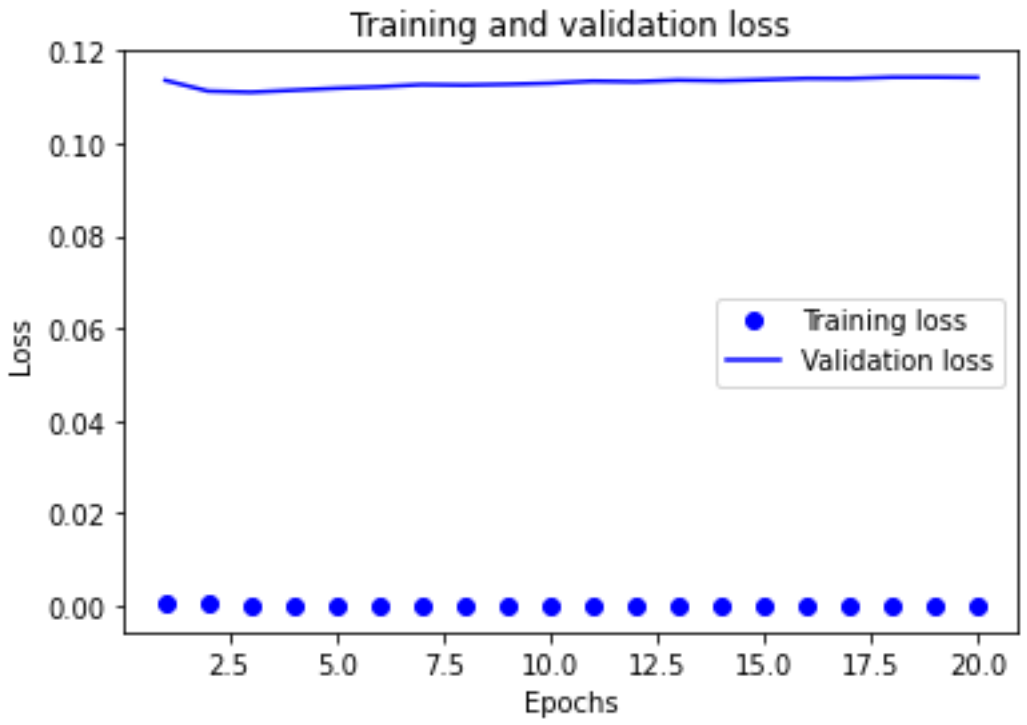


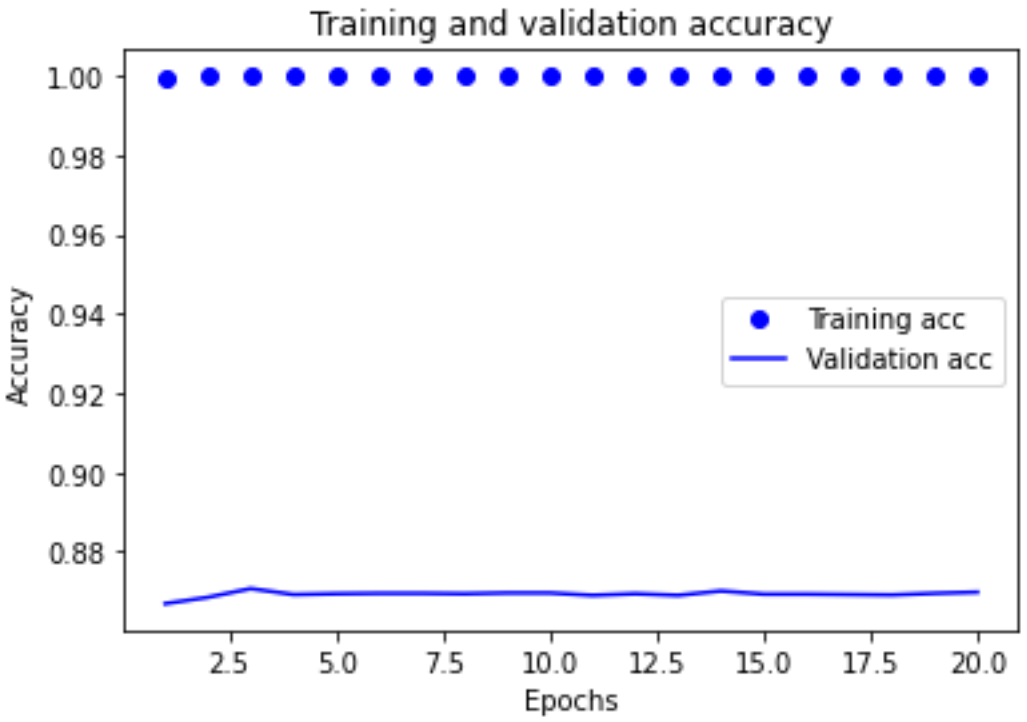


3. **<u>Using MSE as a loss function</u>**

Using the MSE loss function was not ideal for this dataset because MSE is mostly designed for regression problems where the objective is to predict a continuous value. The data set we are working with is a binary classification and the objective is to predict whether the rating for a movie is positive or negative.

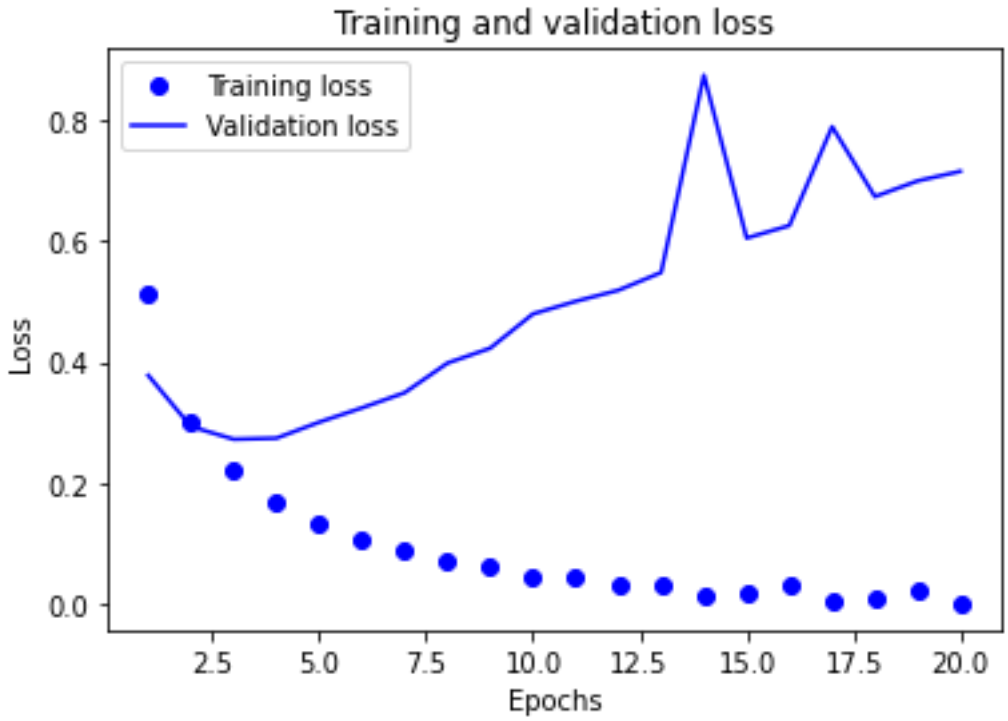
That's why the Binary cross-entropy was the better option as the as the loss function because it is designed for problems that only have two classes.

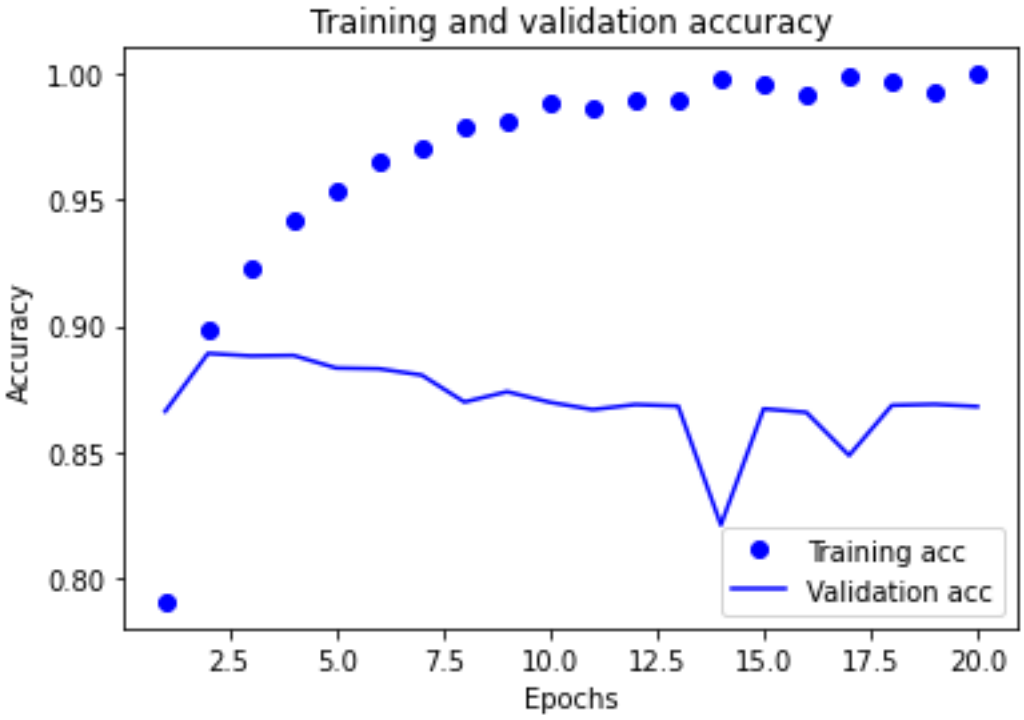




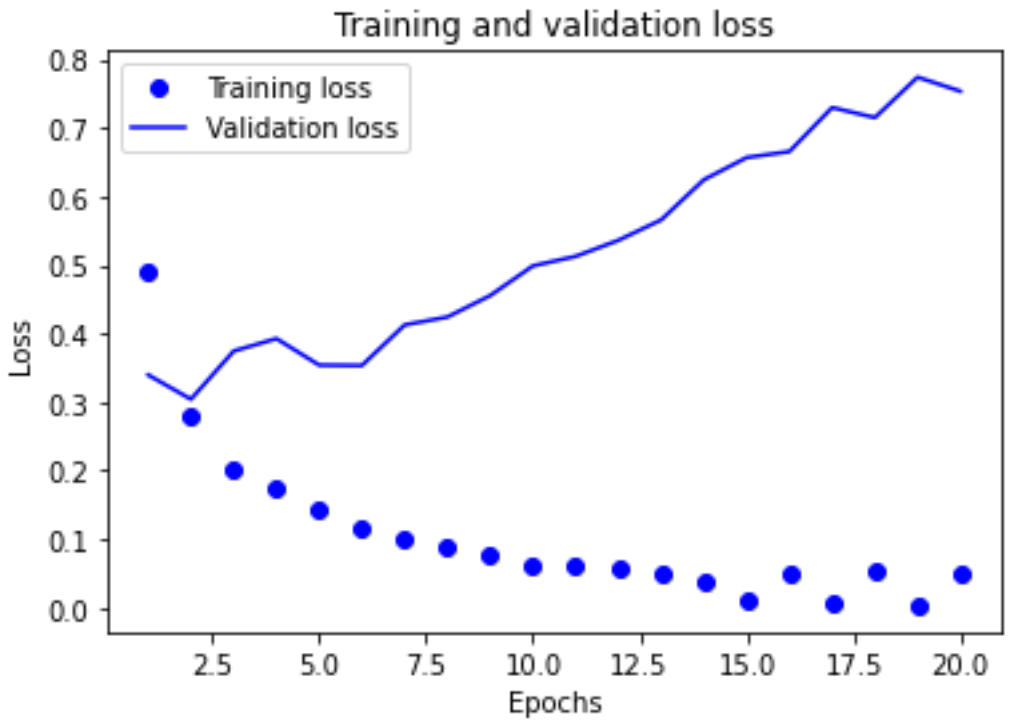
4. **<u>Using the Tanh function with varying amounts of hidden units</u>**

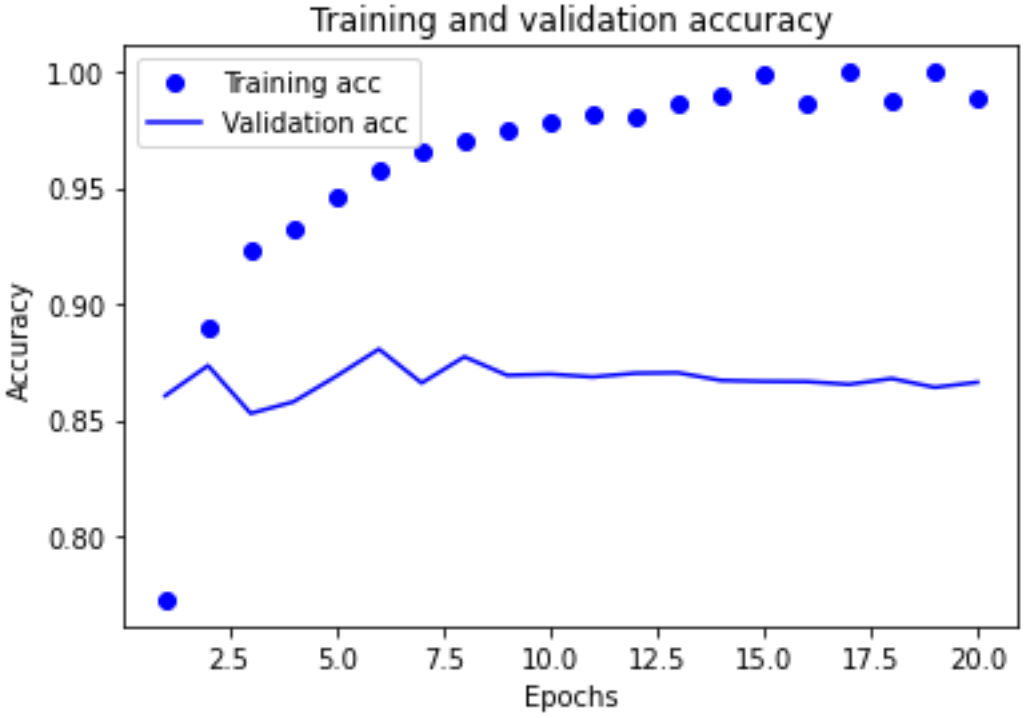
After using the Tanh function with both 16 and 32 bits, I found relu to be more efficient.





Using the tanh loss function with 32 bits


Training and validation loss
Training loss
Validation loss
Loss
Epochs
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Training and validation accuracy
Training acc
Validation acc
Accuracy
Epochs
0.80
0.85
0.90
0.95
1.00
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

Summary table of the hyperparameters

Model	Activation function	Loss function	Number of Hidden layers	Number of hidden units	Accuracy percentage
1	Relu	Binary	1	16	88.84
2	Relu	Binary	2	32	88.34
3	Relu	Mse	2	32	88.58
4	Relu	Mse	3	32	87.20
5	Tanh	Mse	2	16	87.81
6	Tanh	Binary	3	64	86.84
7	Relu	Binary	2	64	88.26

Overall Summary

The best performing models used the relu activation function because of its capability to compute quickly and efficiently. The Tanh activation function wasn't as effective as those models suffered from vanishing gradients. The models that were designed with the MSE loss function were the worst performing, with the one exception of the binary_crossentropy involving the tanh activation function, as that loss function is not designed for classification models that only have two classes.

Changing the number of layers from one to two showed a higher level of accuracy, but changing the layers from two to three didn't have too much of an impact on the performances of the models.