CS499 Project 5

Group Members:

Junshi Wei (jw2372)

Rui Li (rl726)

Alex Lacy (al2428)

Repository URL:

https://github.com/Alex-Lacy/CS499-Project-5

Group Evaluations

Giving v Receiving->	Junshi	Rui	Alex
Junshi	100	100	100
Rui	105	95	100
Alex	105	100	95

As a brief explanation for our group evaluation table, the first column represents the person assigning the scores for that row, while the first row represents the people who are being judged.

Introduction

This project was similar to previous two projects relating to neural networks. However, in this project, we used unique regularization parameters instead of just using number of epochs or iterations. The regularization parameters we chose to go with were the number of hidden units and the number of hidden layers.

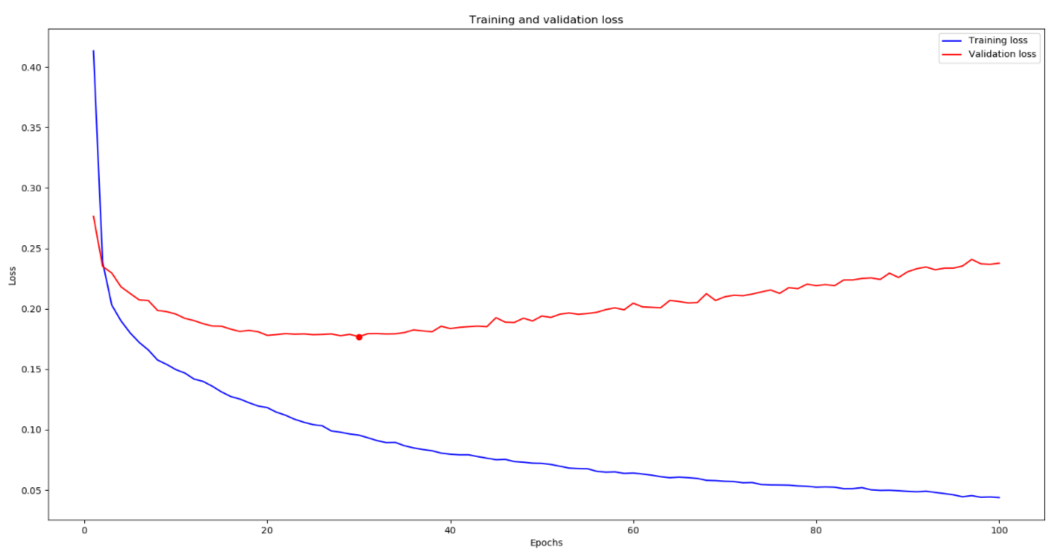
First, we found the optimum number of hidden layers. Each of the following graphs represents a different number of hidden layers. The blue line on all graphs represents the training loss, while the red line represents validation loss. On each, a point emphasizes the minimum value of the validation loss. Because the training loss is constantly decreasing, the minimum is always the far right of the graph, and we did not draw a point to emphasize it.

Because we tested with quite a different number of parameters, there are quite a few different graphs. Under each graph we label it and describe what parameter was used, what the minimum number of epochs was, and what the lowest loss achieved with that parameter was.

Hidden Layers

The following graphs demonstrate the effectiveness of different numbers of hidden layers. Each number of hidden layers has 10 hidden units.

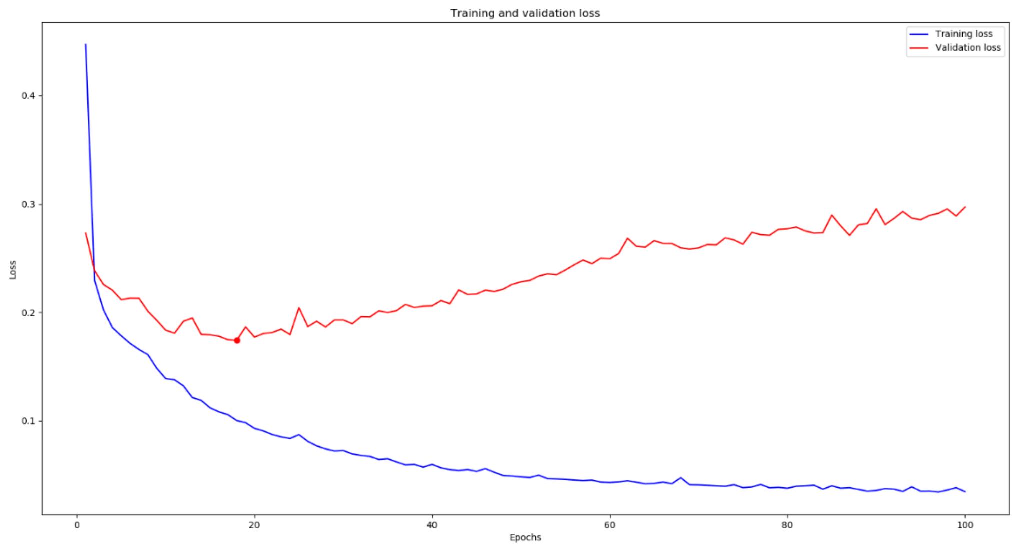
1 Hidden Layer



Best Epochs: 30

Best Loss: 0.17699145379596143

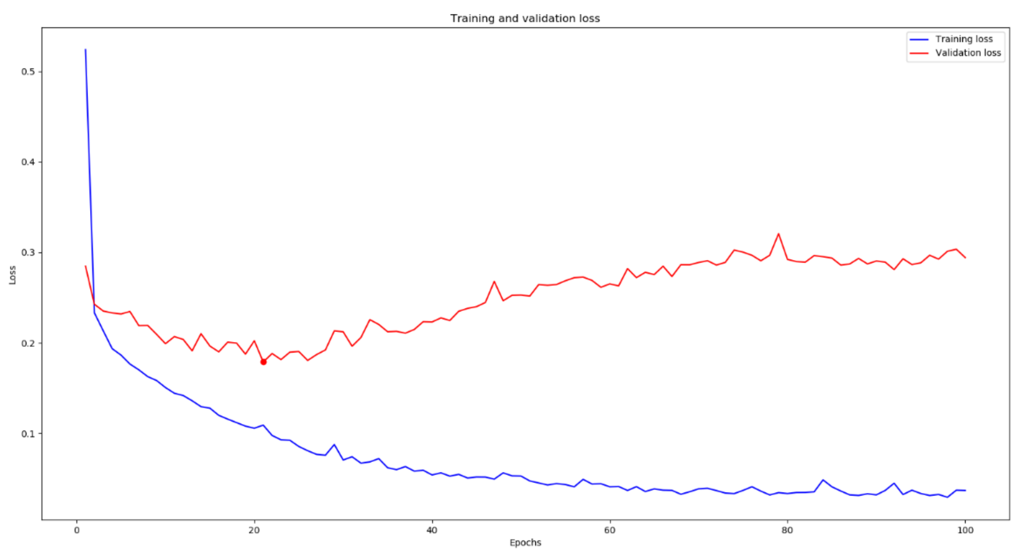
2 Hidden Layers



Best Epochs: 18

Best Loss: 0.17411653732295143

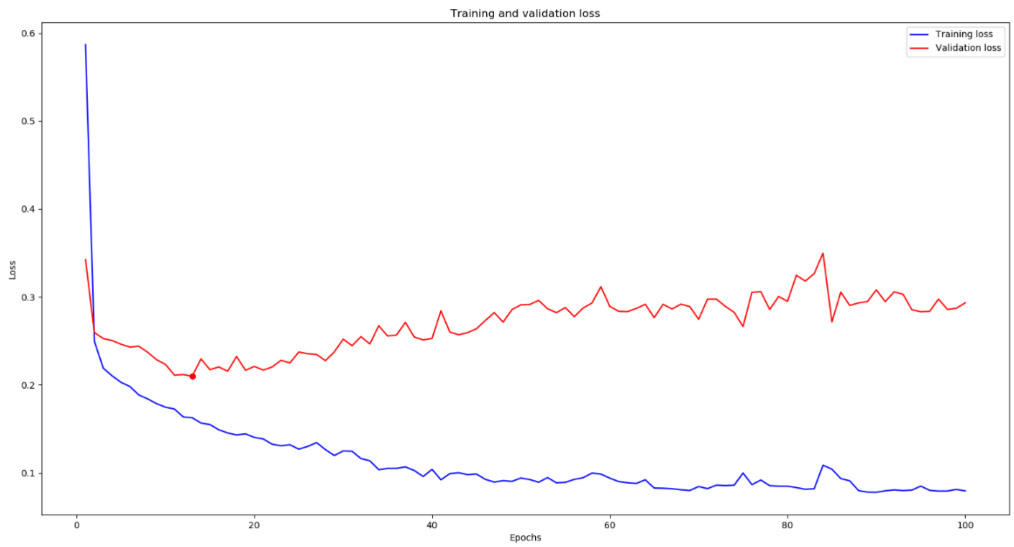
3 Hidden Layers



Best Epochs: 21

Best Loss: 0. 17883577342269086

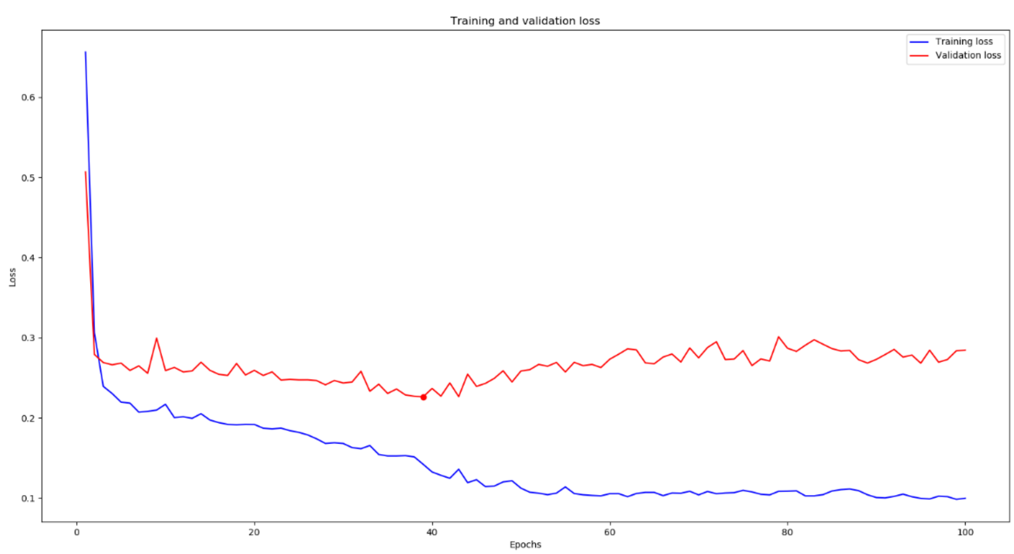
4 Hidden Layers



Best Epochs: 13

Best Loss: 0. 209762838403535

5 Hidden Layers



Best Epochs: 39

Best Loss: 0. 22621679098012978

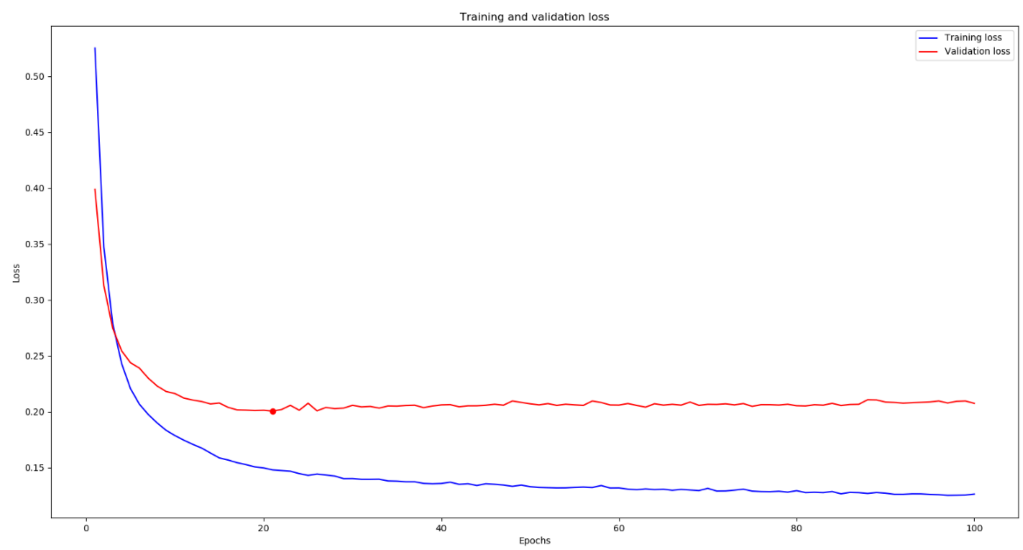
Hidden Layers Conclusion

From the data above, we can see that the best number of hidden layers to use on this dataset is 2 because it minimizes loss the most. Training the network using 2 hidden layers provides an accuracy value of 93.5%. We know this is a good learning value, because the baseline is only 60.6%.

Hidden Units

For each of the following graphs, we used the number of hidden units as our parameter. To maintain consistency, we used 1 hidden layer for each of the tests. The format follows the previous section. For each segment of hidden units, we used a different power of 2 to test. We will simply list the flat number, but it is important to remember that these are all powers of 2.

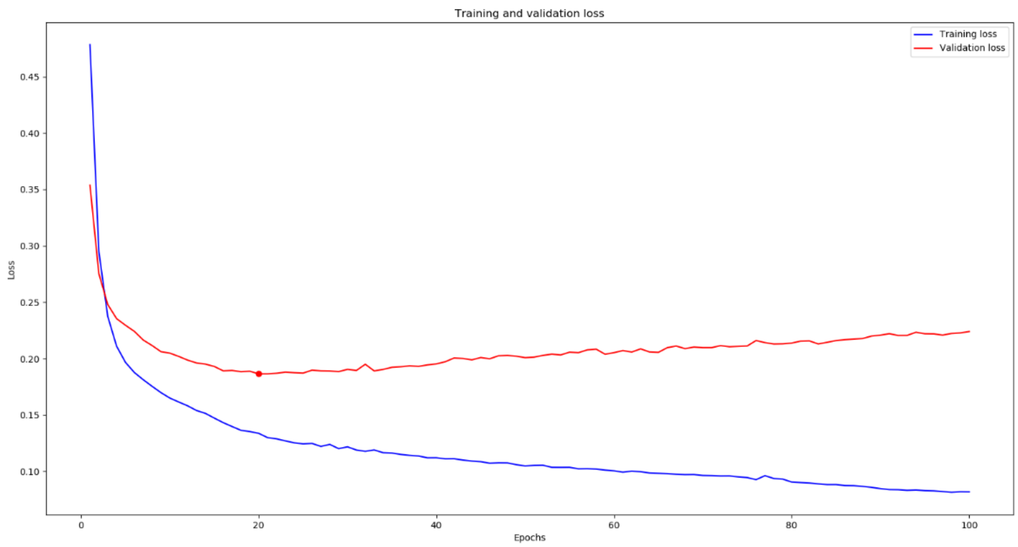
2 Hidden Units



Best Epochs: 21

Best Loss: 0. 20050883035760803

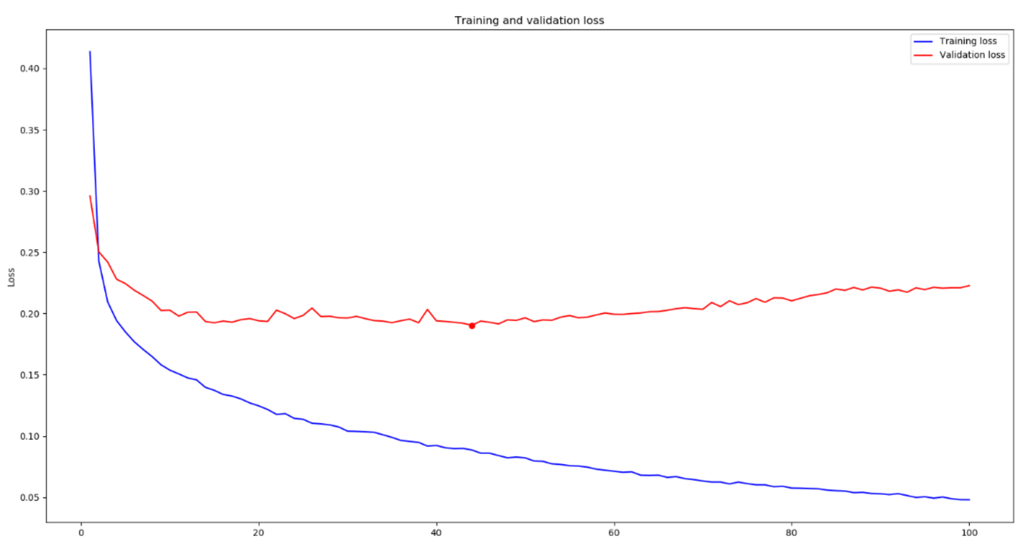
4 Hidden Units



Best Epochs: 20

Best Loss: 0. 18649843866121893

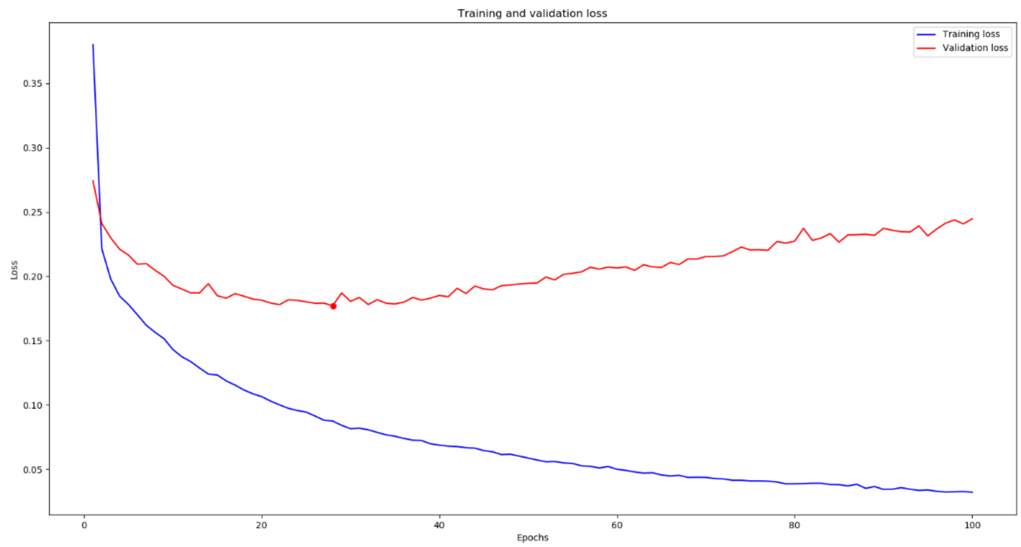
8 Hidden Units



Best Epochs: 44

Best Loss: 0. 19036341998837422

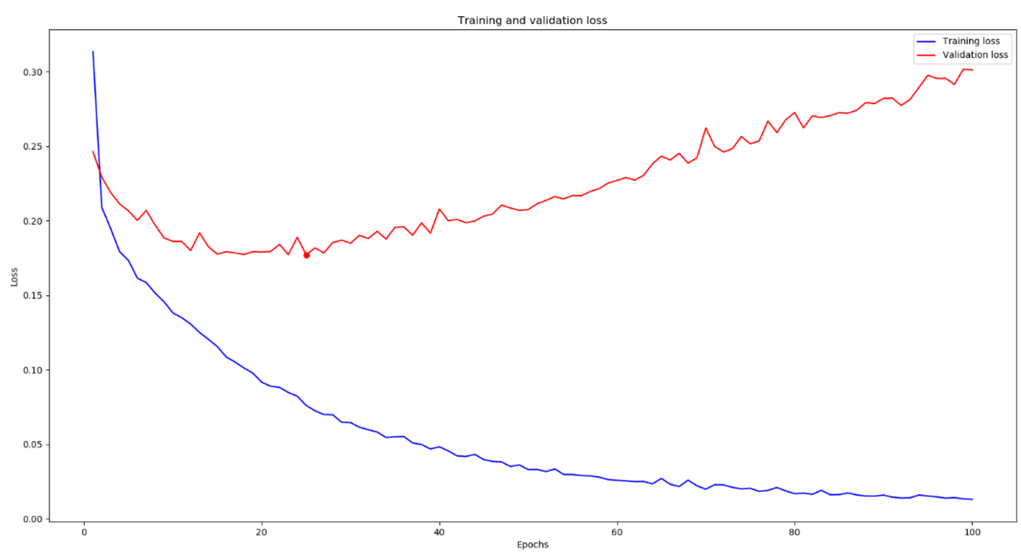
16 Hidden Units



Best Epochs: 28

Best Loss: 0. 177123482406366

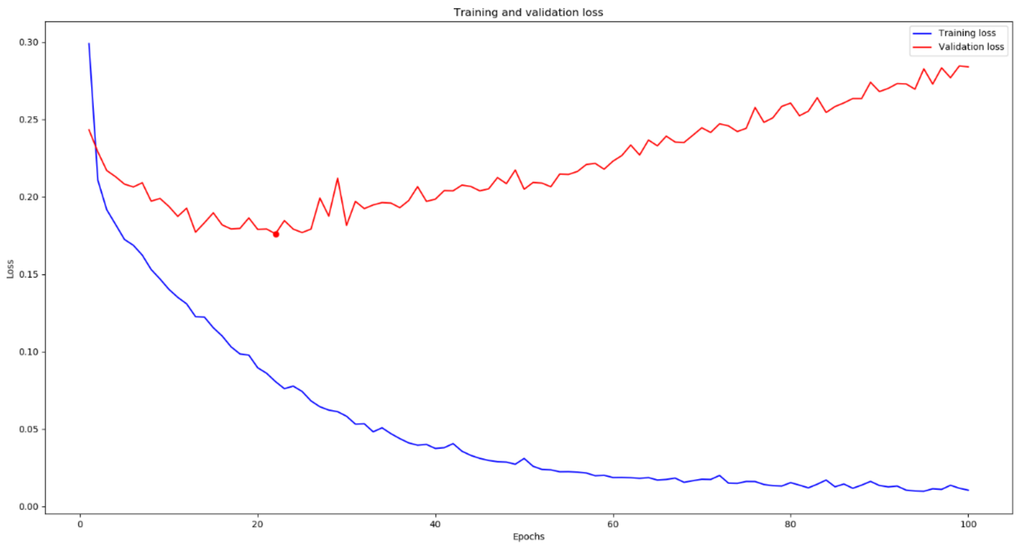
32 Hidden Units



Best Epochs: 25

Best Loss: 0. 1771945816903319

64 Hidden Units



Best Epochs: 22

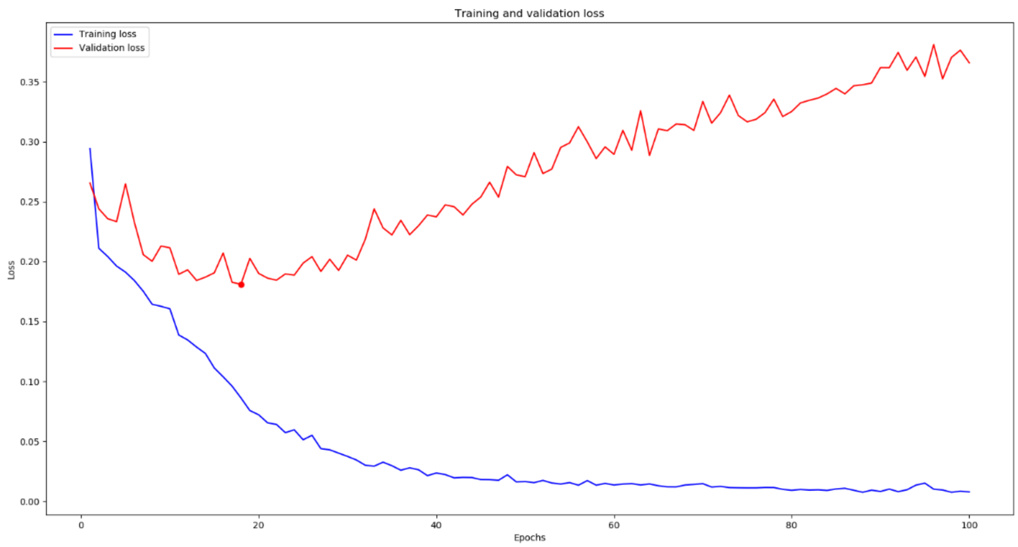
Best Loss: 0. 17604231783419055

128 Hidden Units

Best Epochs: 18

Best Loss: 0. 17369902464179227

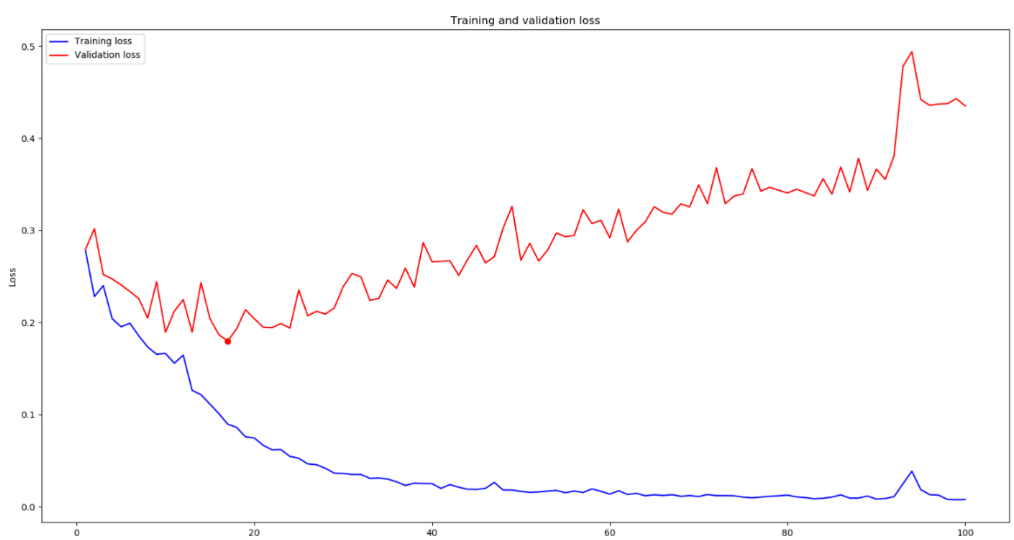
256 Hidden Units



Best Epochs: 18

Best Loss: 0. 18106134819440514

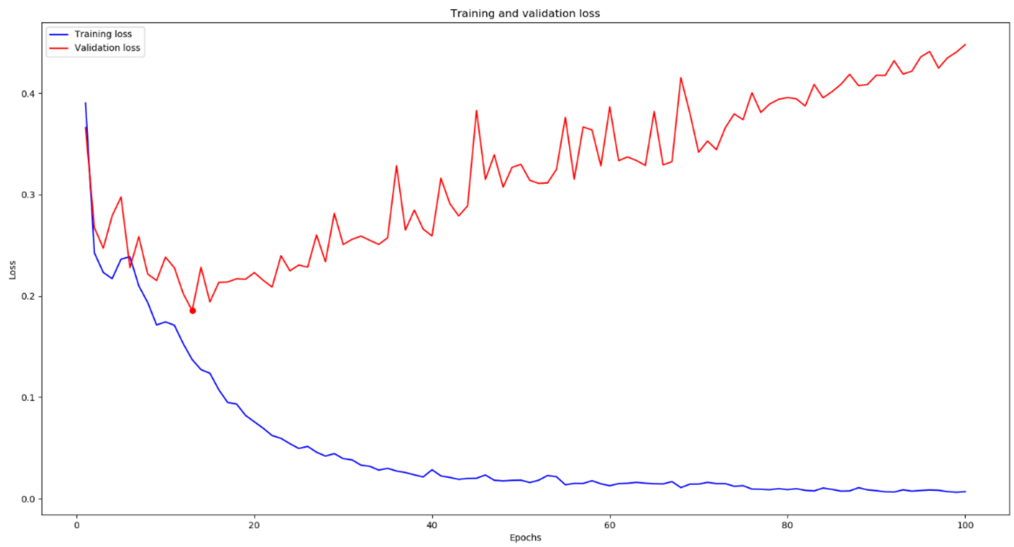
512 Hidden Units



Best Epochs: 17

Best Loss: 0. 17988540629599808

1024 Hidden Units



Best Epochs: 13

Best Loss: 0. 18548974423127482

Hidden Units Conclusion

From the data above, we can see that the best number of hidden units to use on this dataset (with one hidden layer) is 128, or 2^6, because it minimizes loss the most. Training the network using 128 hidden units provides an accuracy value of 93.9%. We know this is a good learning value, because the baseline is only 60.6%.

Final Conclusion

Using both of these methods resulted in prediction accuracies far exceeding the baseline predictions for this dataset. Using each algorithm separately resulted in accuracy values of 93.5% accuracy and 93.9% accuracy, which are both very close to each other. A much more complicated version of this project would include finding both values at the same time (or finding on then the other).