

Project 2 Report

Code deliverables:

The program iterates over all components and prints out the test accuracy at each step:

```
Number of components 6 , Accuracy achieved: 0.90
Number of components 7 , Accuracy achieved: 0.94
Number of components 8 , Accuracy achieved: 0.87
Number of components 9 , Accuracy achieved: 0.90
Number of components 10 , Accuracy achieved: 0.90
Number of components 11 , Accuracy achieved: 0.87
Number of components 12 , Accuracy achieved: 0.87
Number of components 13 , Accuracy achieved: 0.87
Number of components 14 , Accuracy achieved: 0.87
Number of components 15 , Accuracy achieved: 0.87
```

Figure 2: Iterating for various components, with Accuracy

```
-----
Number of components for Max accuracy 7 , Max Accuracy achieved: 0.94
Confusion matrix for max accuracy:
[[ 95  2]
 [  2 109]]
```

Figure 1: Components that achieve the max accuracy with

It also prints out the number of components that achieve the max accuracy, the accuracy achieved and the confusion matrix corresponding to the max accuracy.

Analysis:

As per the highest accuracy that we were able to achieve, I believe the chances of surviving the minefield are low. The reason for that, despite having a high accuracy (94 – test and 98 - combined), is because the Cobra shaped submarine can barely stand even 1 mine explosion. So, if we're not 100% correct, we'd be doomed. More importantly, the confusion matrix prints out 2 false positives – which is fine as we would miss the rocks thinking them as mines, but the 2 False negatives – are a serious threat for the submarine as those would be actual mines that can destroy the submarine. The general shape of the plot is like this because the accuracy of prediction increases drastically initially as we increase the number of components up to a certain limit (7 in our case), because we're introducing new correlated data. After that the more components we add it keeps on introducing noise in our prediction accuracy and ends up increasing the false positive and negative values. I iterated over various combinations of max_iterations and hidden layer sizes to find the best accuracy with the lowest false negatives. The value of max_iterations also started showing very less impact after a limit and I changed the random seed to false so that the randomization doesn't impact my calculations.

