**Rashadul Hasan Badhon**

**#801341298**

**HW#1**

**Gitlink:** [**https://github.com/RHBadhon/ECGR\_5106/tree/main/HW%231**](https://github.com/RHBadhon/ECGR_5106/tree/main/HW%231)

Q1

**a.** Train the model from scratch (with randomized parameters) and plot the results (training loss and accuracy, validation accuracy) after 20 epochs. Does your network need more epochs for full training? Do you observe overfitting? Make sure to save the trained parameters and model. Report and plot your training and validation results. Report precision, recall, F1 score, and confusion matrix.

Ans: From the figure it’s obvious that overfitting has affected the model heavily so increasing epoch is not going to solve the problem. I got a training accuracy of 45 % and validation accuracy of 53 %. Precision, recall, f1 score for each class is attached below:

|  |
| --- |
|  |
|  |
|  |

**b.** Explore the complexity of the network by increasing its width and depth. How do the training and validation results change? Compare them against the baseline. Do you see any overfitting?

Ans: Previously the model had 3 hidden layers now it’s increased to 5 hidden layers. Although it has less overfitting, the model complexity is still not enough to get good accuracy. Due to fully connected layers, we can only get a limited amount performance increase for large or complex dataset. We also increased epochs from 20 to 30 so the accuracy improves as well:

|  |
| --- |
|  |
|  |
|  |
|  |

Q2

**a.** Build a multi-perceptron network that regresses the housing price (based on 20%, 80% split). Use the same number of features we did in the lecture without on-hot encoding. Please plot the training and validation results and report final accuracy and model complexity

Ans: The model is very simple with few layers hence the accuracy wasn’t very high. Final validation rMSE was 5.64.

|  |
| --- |
|  |

**b.** Build a multi-perceptron network that regresses the housing price (based on 20%, 80% split). Use the same number of features we did in the lecture, but this time also add on-hot encoding. Please plot the training and validation results and report the final accuracy and model complexity. Do you see the meaningful changes against 2.b.

Ans: One problem at this stage was that some columns were either all 0’s or having a standard deviation of 0. So, I excluded one column “ExterQual” and added epsilon e^-8 to the standard deviation to avoid getting NaN value. This slightly increased the rMSE to 5.65.

|  |
| --- |
|  |

**c. I**ncrease the complexity of the network for problem 2. b and compare your results against 2.b.

Ans: 3 additional layers were added for the model and executed.

|  |
| --- |
|  |