CPSC 501 Assignment 2

Report

Part 3

Data decision

As there is only a small amount of data, I thought it would need sufficient testing data to get consistent result in the first place. I decided to split train-test data as 75-25, as that would still give more than 100 testing data and over 300 training data so it trains properly.

Processing data and the model before the changes

To begin fitting the model, I needed to convert the text column 'famhist' in the dataset, as it has string values. In order to do that, I used pandas groupby.ngroup() method to give each unique text value an integer value, so that the rows with the same text value will be applied the same numerical value. (Source: https://pandas.pydata.org/docs/reference/api/pandas.core.groupby.GroupBy.ngroup.html)

For the model, I re-used the code from Main Stage 3, which had a hidden layer of 400 nodes and a dropout layer with a dropout rate of 0.2, and 100 epochs for fitting the model. Using the exact model in this dataset, I got a training accuracy of 74-78% and a testing accuracy of 65-70, with around 7-8% difference in average.

Dealing with the overfitting problem

First of all, I added a dropout layer right after the input layer and a batch normalization layer right before the existing hidden layer. Using https://www.tensorflow.org/tutorials/keras/overfit_and_underfit as the source, I followed its "Tiny model" setup and reduced the number of neurons in the hidden layer to 16, leaving the dropout layer alone. I tweaked around with the drop rates with both dropout layers to find the optimal result.

Result of the changes

With the dropout layer of 0.25 drop rate in the beginning, the hidden layer having 16 neurons followed by batch normalization and then the dropout layer of 0.1 drop rate, the model's result with 100 epochs was as following:

- Training accuracy: 66-70
- Testing accuracy: 66-71
- Average difference: 0.195% difference in the average of 20 runs