Vincent La

903178639-vla6-hw5

# 1.2 Multi Layer Perceptron

## 1.2.b

*Calculate the number of "trainable" parameters in the model with providing the calculation details. How many floating-point computation will occur when a new single data point comes in to the model?*

*Note that on Calculating the Number of “trainable parameters”, the following Piazza Posts were consulted:*

1. [*https://piazza.com/class/jjjilbkqk8m1r4?cid=928*](https://piazza.com/class/jjjilbkqk8m1r4?cid=928)
2. [*https://piazza.com/class/jjjilbkqk8m1r4?cid=928*](https://piazza.com/class/jjjilbkqk8m1r4?cid=928)

Note that the number of “trainable parameters” come from Weights, biases, etc. which are being trained/learned in the model.

We can calculate this using code (note that formatting taken from this SO [post](https://stackoverflow.com/questions/53035690/how-to-test-python-code-responding-to-another-post)):

|  |
| --- |
| print(sum(p.numel() for p in best\_model.parameters() if p.requires\_grad)) |

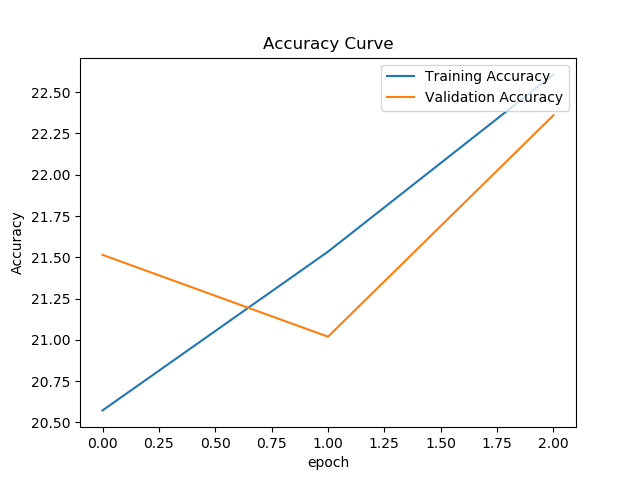
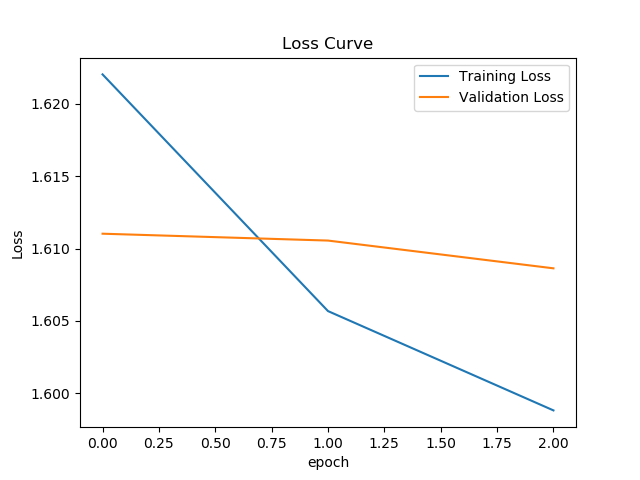
**This returns 2,965 trainable parameters.**

For the number of floating point operations, note that a sigmoid function is 4 operations. Furthermore, a dot product between two vectors of size n returns 2n -1 operations, since we have n multiplications and n-1 summations.

Thus, if there are 2,965 trainable parameters, we have close to ~20,000 floating point operations.

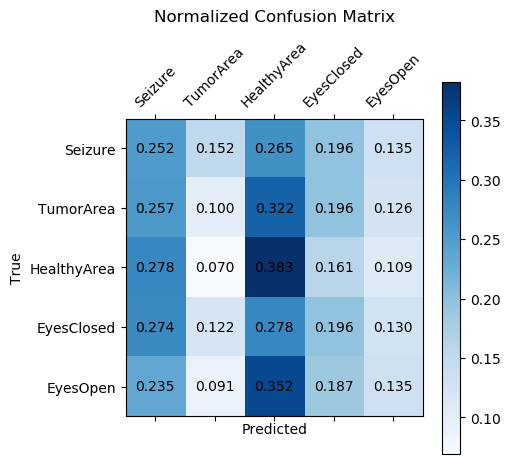
## 1.2.c

Learning Curves for MLP:



## 1.2.d

Confusion Matrix for MLP



## 1.2.e