CSE6250: Big Data Analytics in Healthcare Homework 5

George Zhou (GTID: 903520176)

October 30, 2019

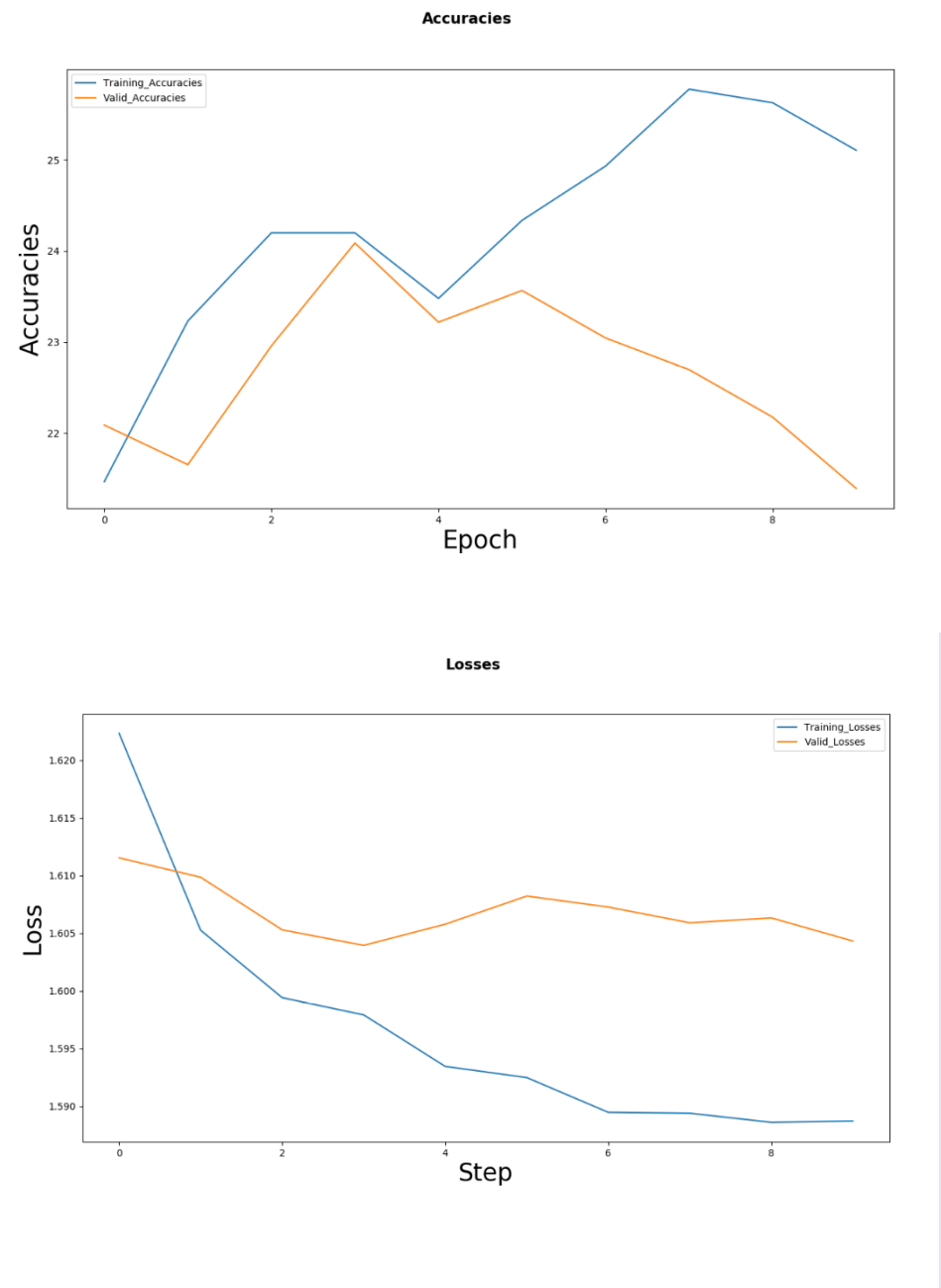
1 Epileptic Seizure Classification

1.2 Multi-layer Perceptron

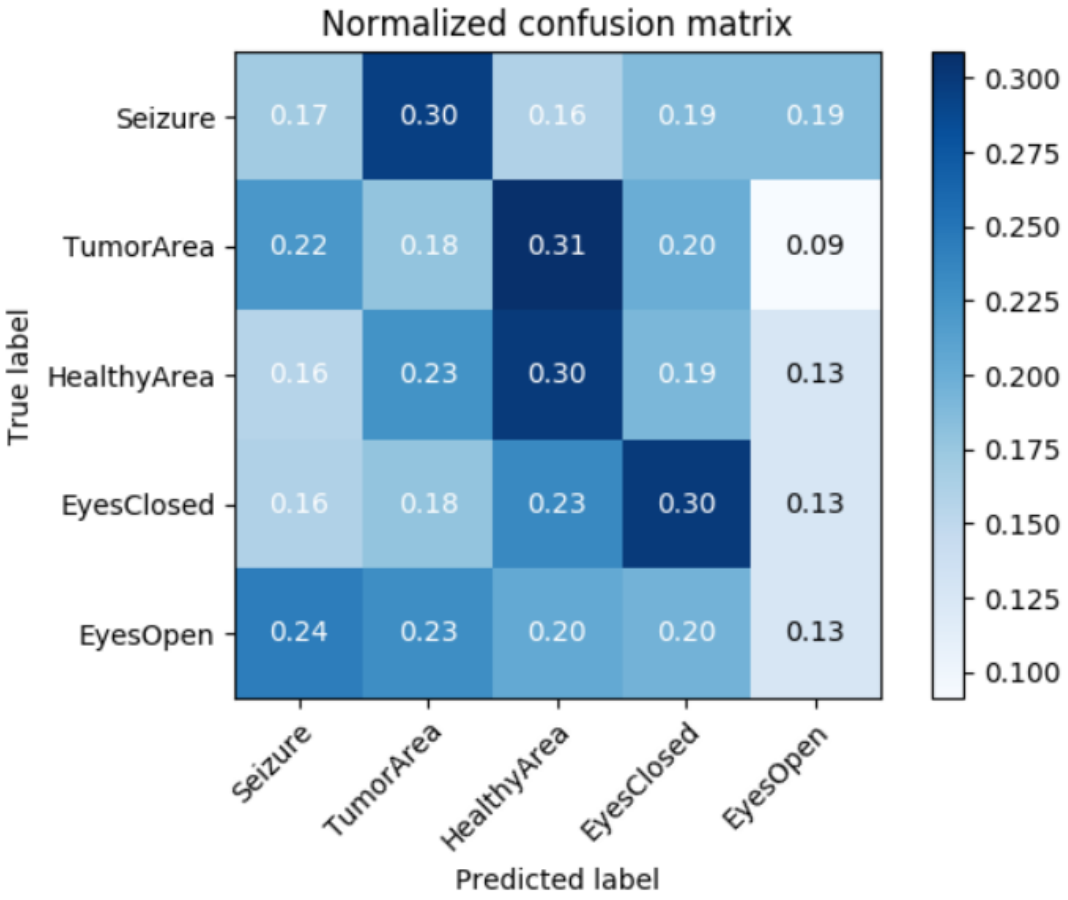
b.
Calculate the number of ”trainable” parameters in the model with providing the cal-culation details. How many floating-point computation will occur when a new single data point comes in to the model? You can make your own assumptions on the number of computations made by each elementary arithmetic, e.g., add/subtraction/multiplication/division/negation/exponent take 1 operation, etc. [5 points]
Trainable parameters are 178*16 (weights) + 16(bias) + 16*5(weights) + 5(bias) = 2949 parameters

Floating Point Computation:
178*16*3 (3 operations per layer) + 16*5*3 (again) + 16*3 (sigmoid) = 8832

c.Attach the learning curves for your MLP model in your report. [2 points]

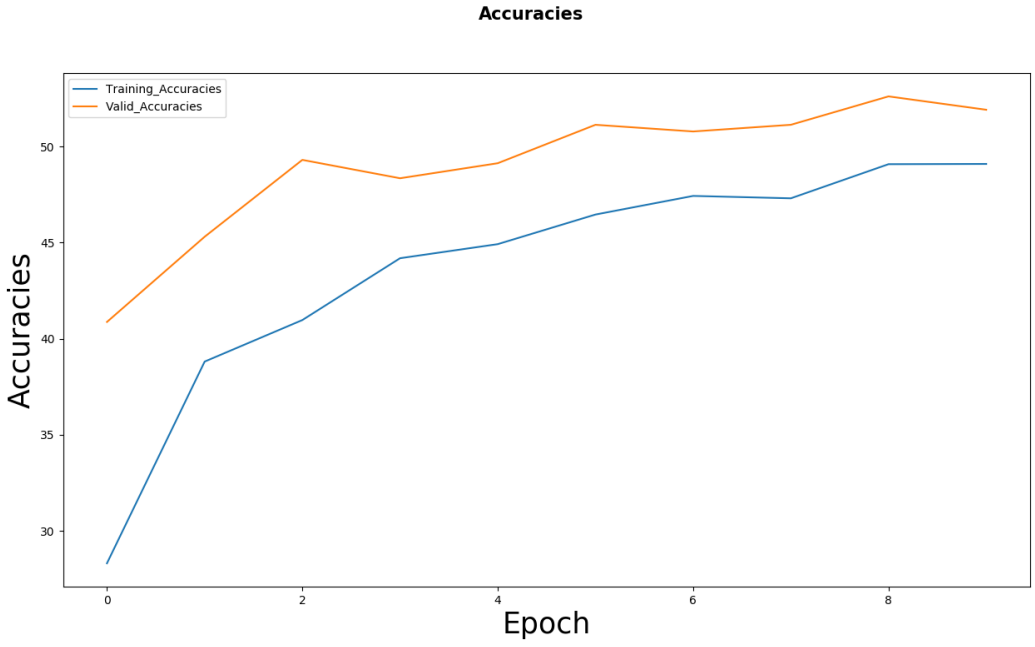


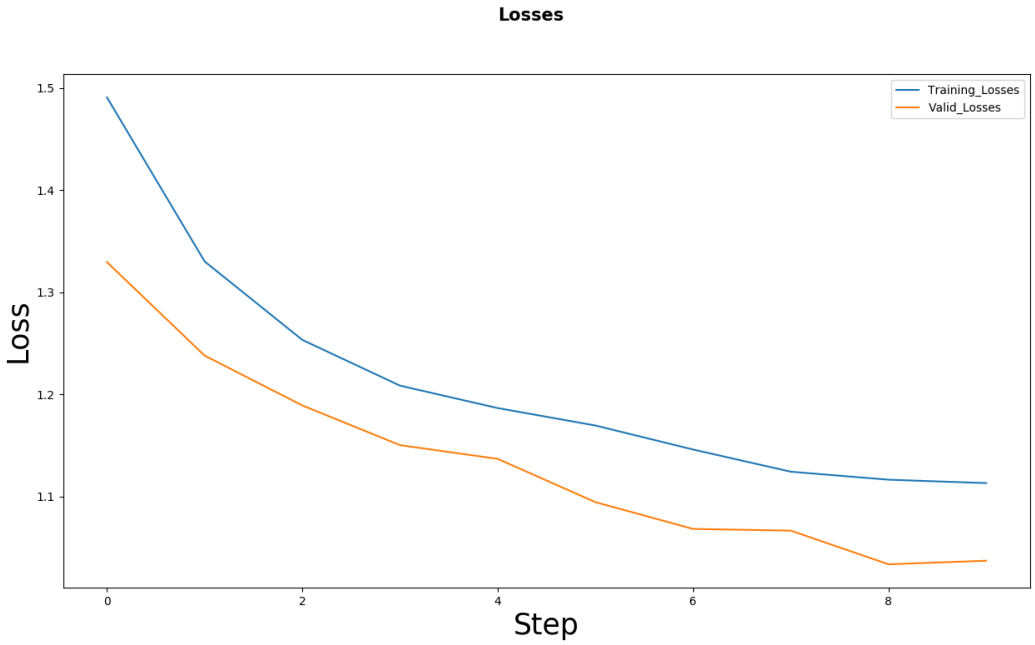
d.
Attach the confusion matrix for your MLP model in your report. [2 points]

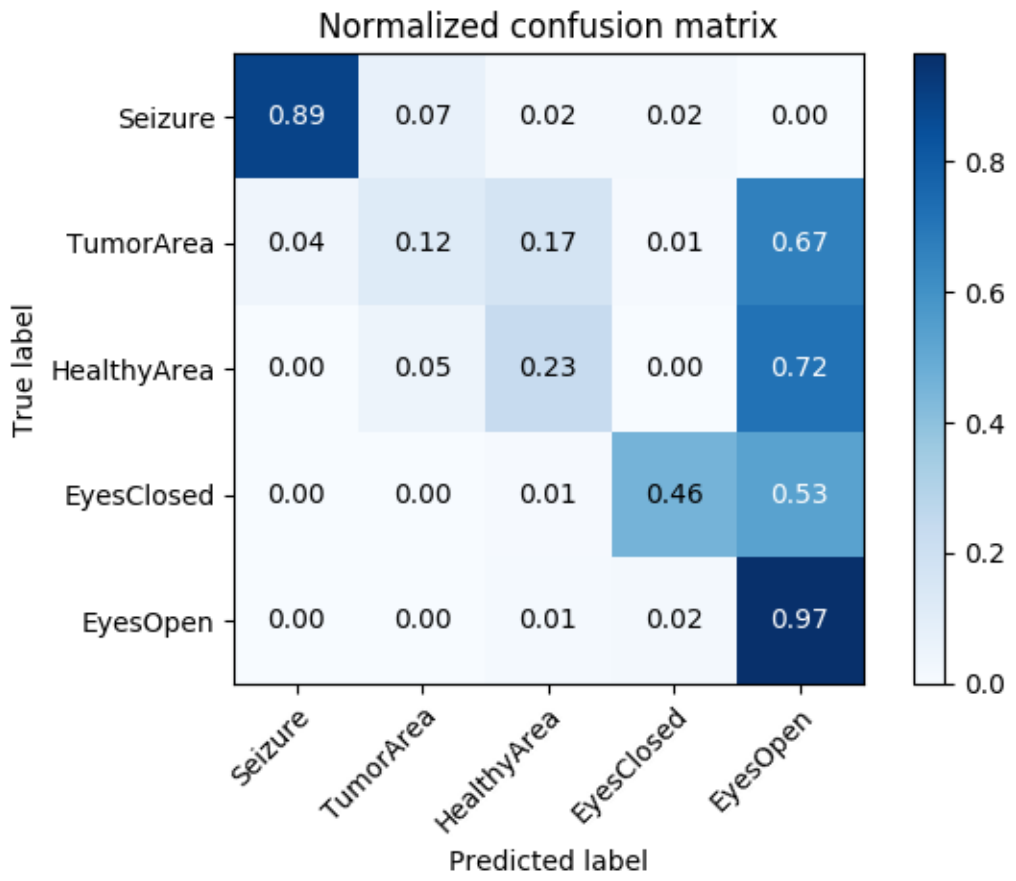


e.
Explain your architecture and techniques used. Briefly discuss about the result with
plots. [3 points]

I used 2 hidden layers with normalization in first layer and dropout of half the neurons in both layers as well as relu in both hidden layers. I tried with and without relu and it was very clear that relu was the main reason I got improved results in all the graphs. The other tweaks I did in this architecture did not improve results much at all.


Accuracies
Training_Accuracies
Valid_Accuracies
Accuracies
Epoch
50
45
40
35
30
0
2
4
6
8


Losses
Training_Losses
Valid_Losses
Loss
Step
1.5
1.4
1.3
1.2
1.1
0
2
4
6
8



1.3 Convolutional Neural Network (CNN)

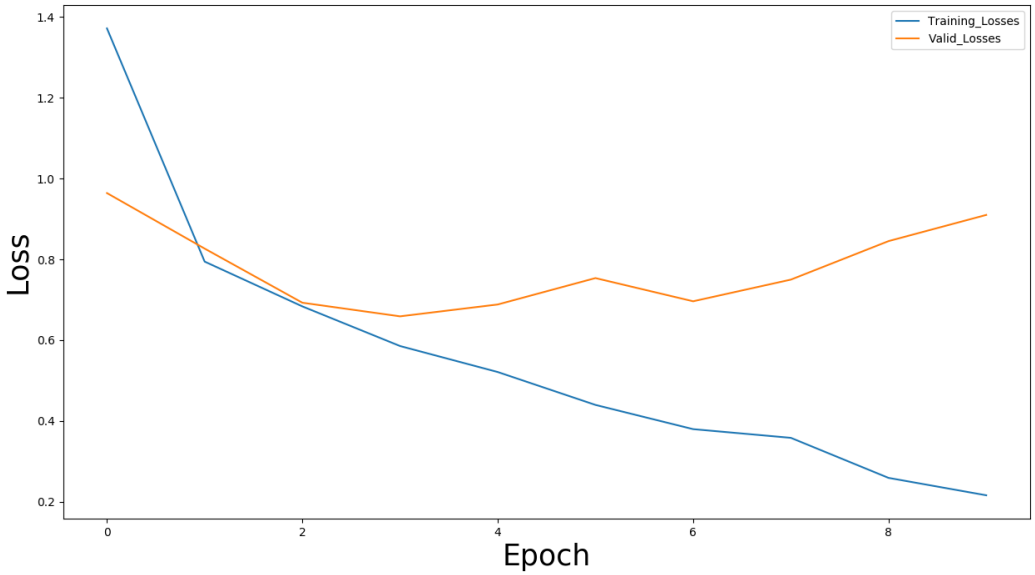
b.
Calculate the number of ”trainable” parameters in the model with providing the cal-
culation details. How many floating-point computation will occur when a new single data
point comes in to the model?
You can make your own assumptions on the number of
computations made by each elementary arithmetic, e.g., add/subtraction/multiplication/division/negation/exponent take 1 operation, etc.
[5 points]

For 1st conv1d layer, there are 1x5x6 weights and 6 bias = 36 parameters.
For 2nd conv1d layer there are 6x5x16 parameters and 16 bias = 496 parameters
For 1st FC layer, there are 16 x 41 x 128 weights and 128 bias = 84096 parameters
For 2nd FC layer, there are 128 x 5 weights and 5 bias = 645 parameters

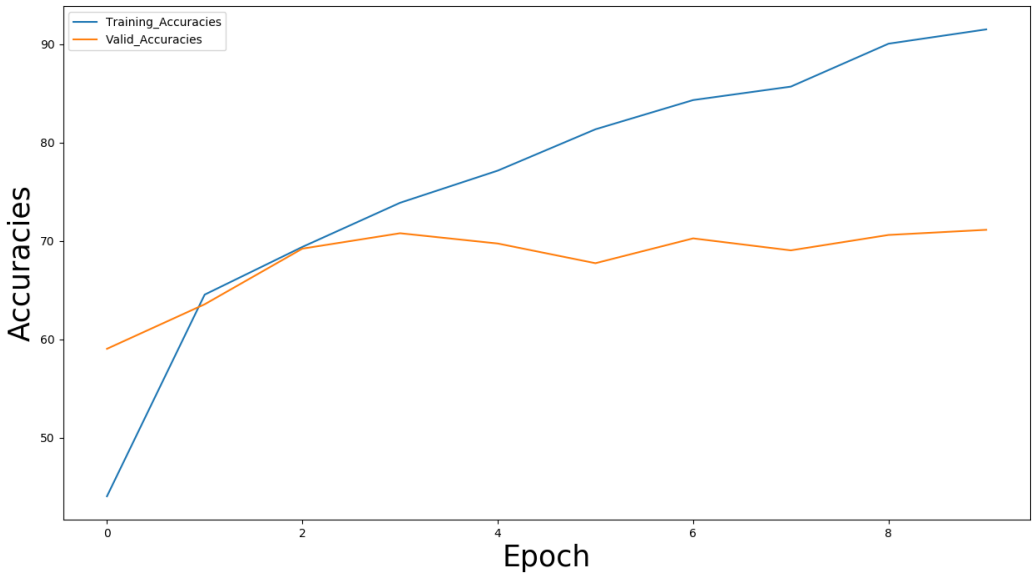
36 + 496 + 84096 + 645 parameters = 85273 parameters – pytorch says there are 85305 trainable parameters (I may be calling the wrong method though.)

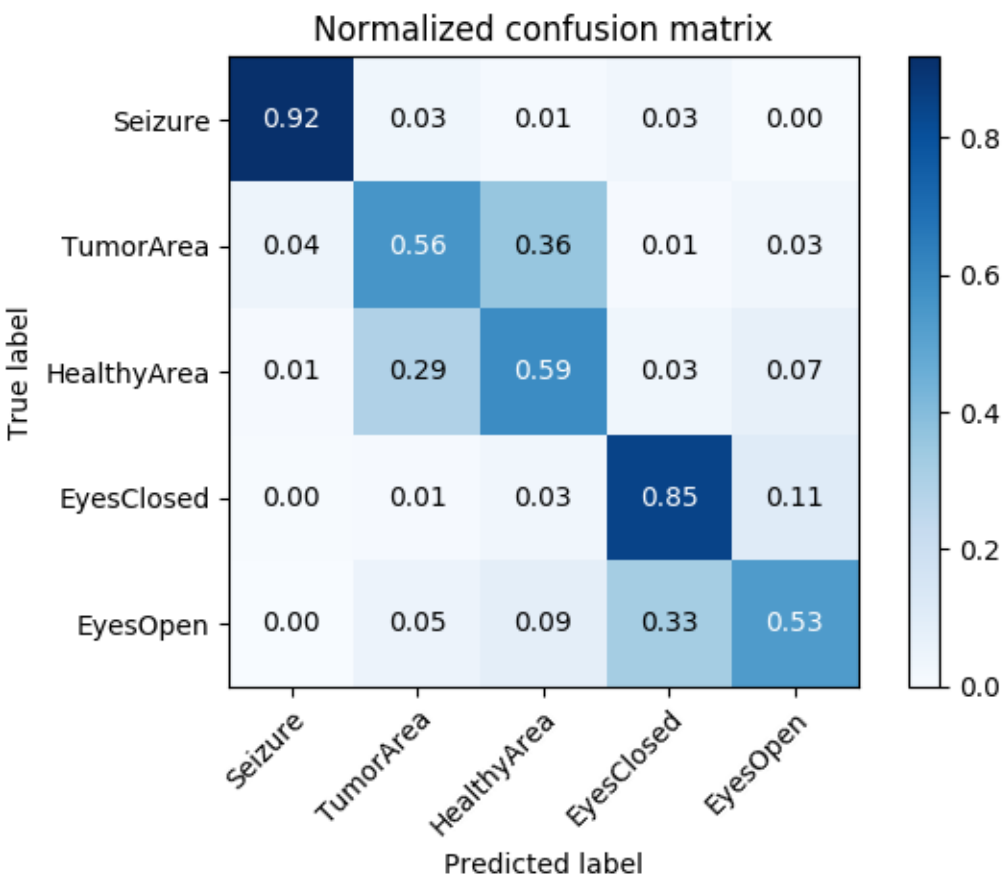
For 1st conv1d, 174*1*6*5*1*1(output *5)+174*1*6*2 (output * 2)= 7308 FP comp
1st maxpool, - 87*1*6*2*1 = 1312 FP computations,
For 1st FC layer, 128*(16*41) +128*2 = 84224 FP comp,
For 2nd FC layer, 5*128 = 640 FP comp,
1044+ 1312+ 42496+7308+84224+640 = 137024 FP comp.
c.
Plot and attach the learning curves and the confusion matrix for your CNN model in
your report. [2 points]

Losses

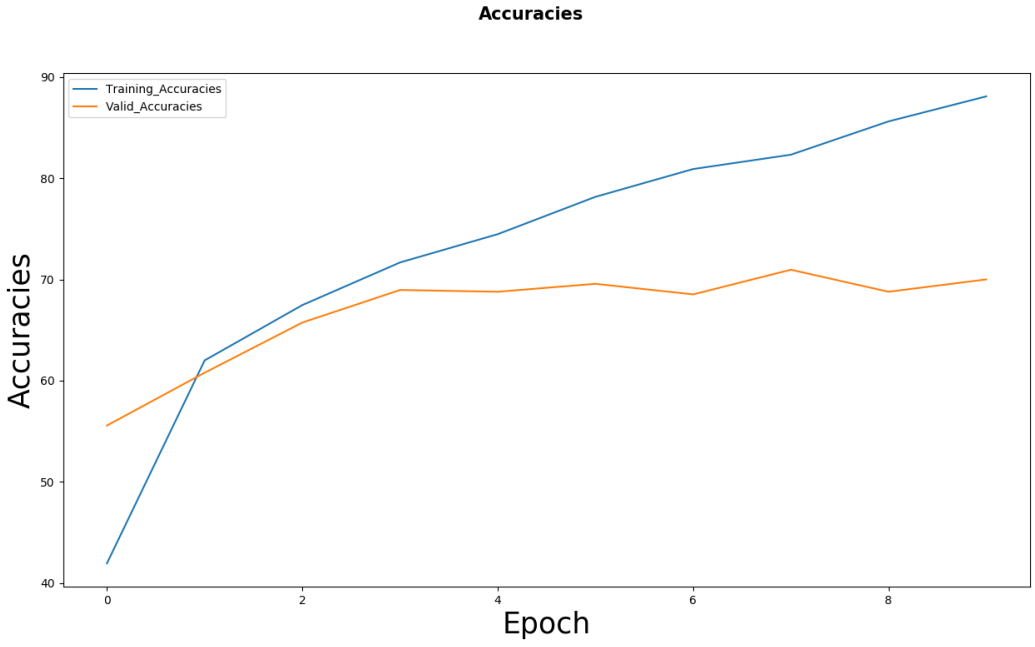


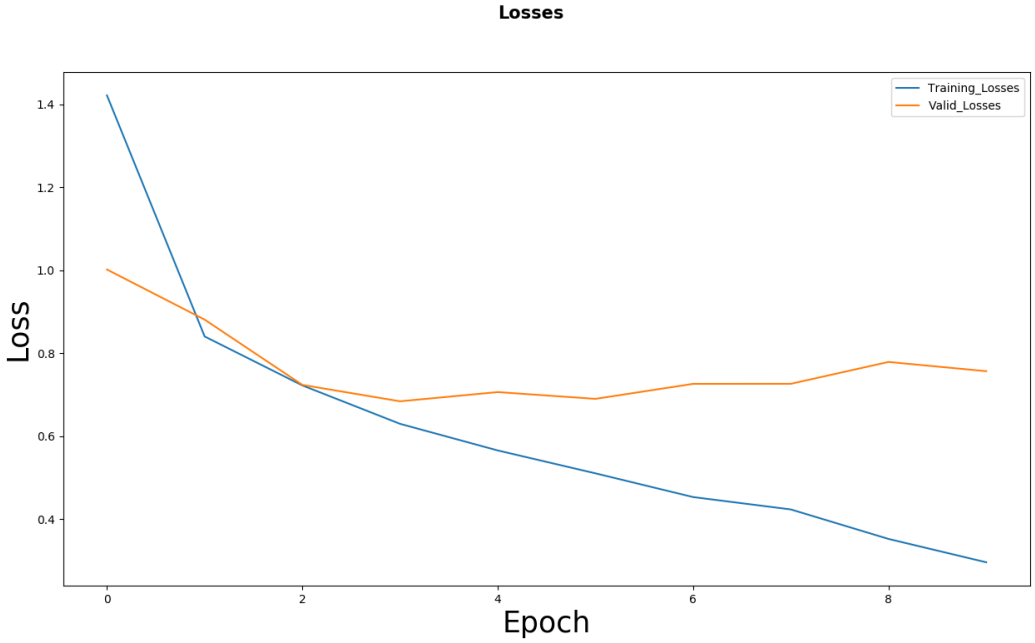
Accuracies

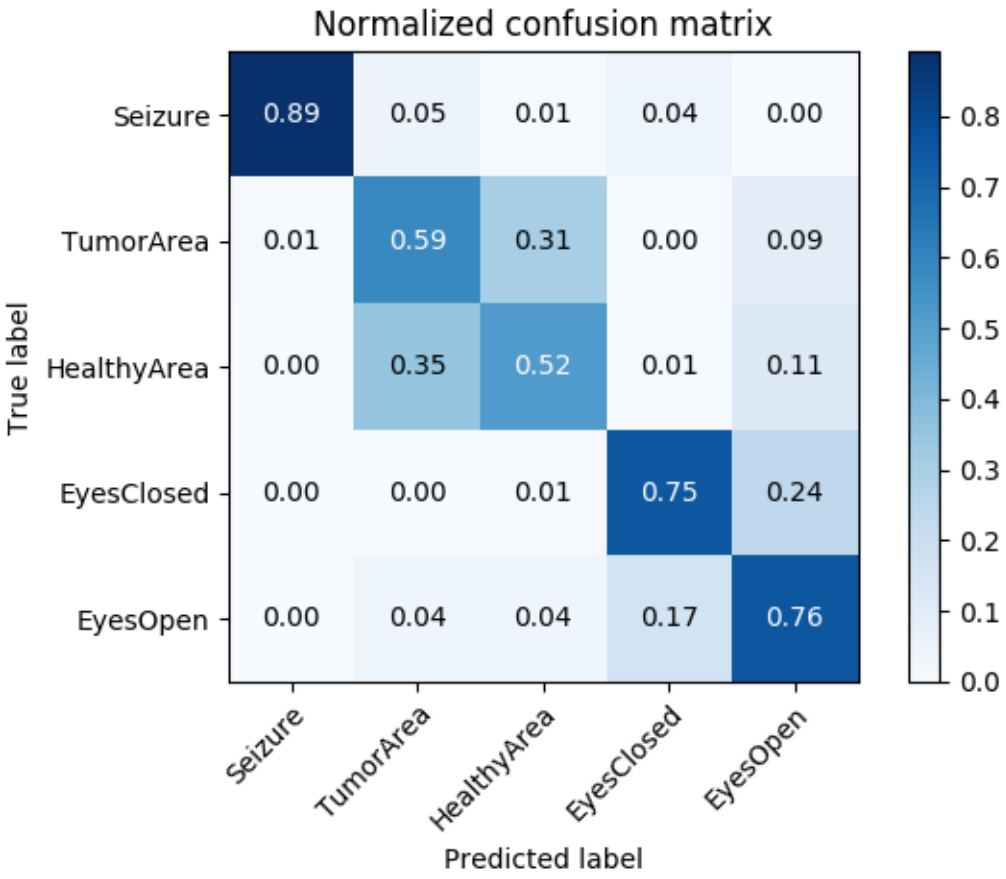




d.
Explain your architecture and techniques used. Briefly discuss about the result with
plots. [3 points]
Basically I just added dropout to last layer with about .1 dropped out to the dense layer. Validation loss definitely was more stable and improved.


Accuracies
Training_Accuracies
Valid_Accuracies
Accuracies
Epoch
90
80
70
60
50
40
0
2
4
6
8


Losses
Training_Losses
Valid_Losses
Loss
Epoch
1.4
1.2
1.0
0.8
0.6
0.4
0
2
4
6
8



1

1.4 Recurrent Neural Network (RNN)

b.

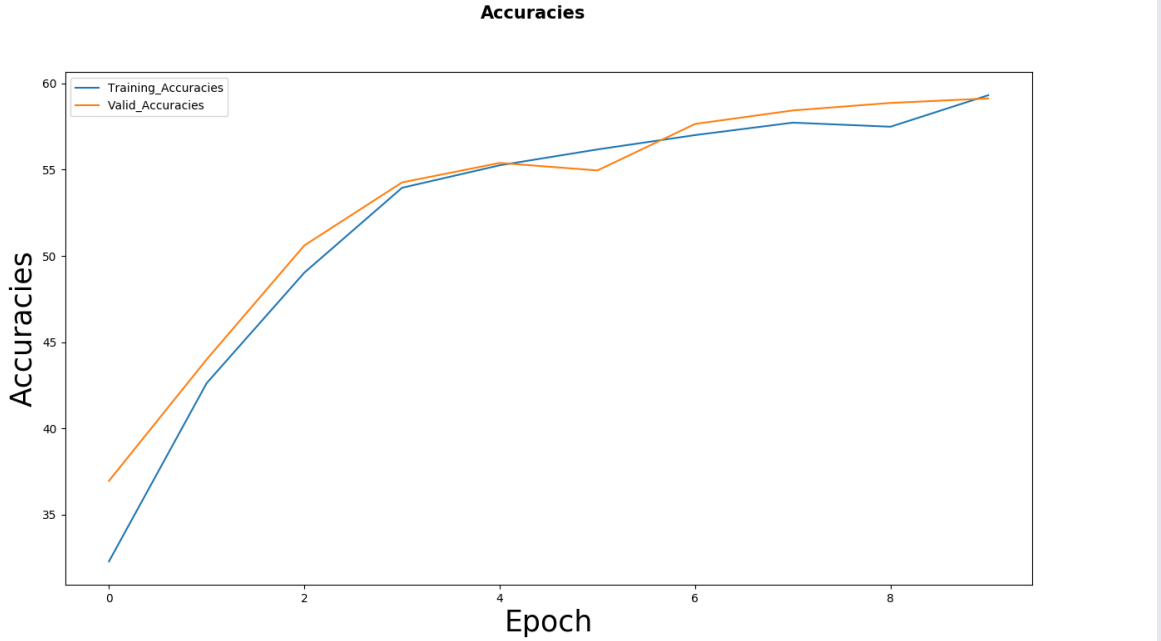
Calculate the number of ”trainable” parameters in the model with providing the cal-
culation details. How many floating-point computation will occur when a new single data
point comes in to the model?
You can make your own assumptions on the number of
computations made by each elementary arithmetic, e.g., add/subtraction/multiplication/division/negation/exponent take 1 operation, etc.
[5 points]

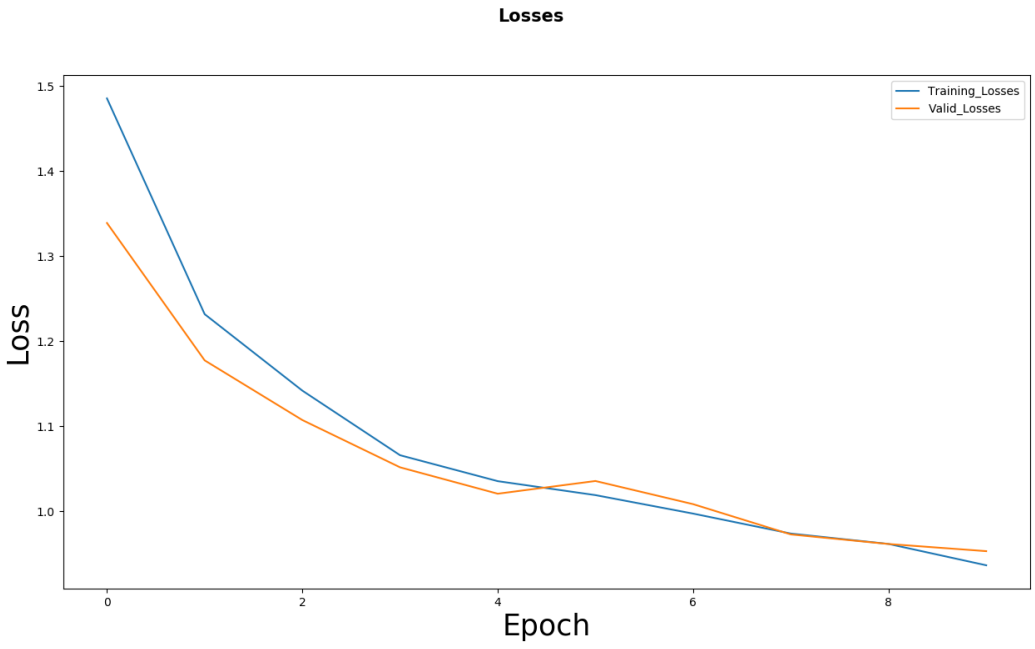
For GRU, 3*16*(16+1) weights + 3*16+3*16 bias = 912 parameters
For FC layer, 16*5 weights + 5 bias = 85 parameters
912+85 = 997 parameters. pytorch says there are 10217 trainable parameters(I might be using wrong function in pytorch though)
In GRU layer, 3*16*(16+1)*178*1*1*16*1*1 = 2323968 FP comp
In FC layer, 16*5 = 80 FP comp

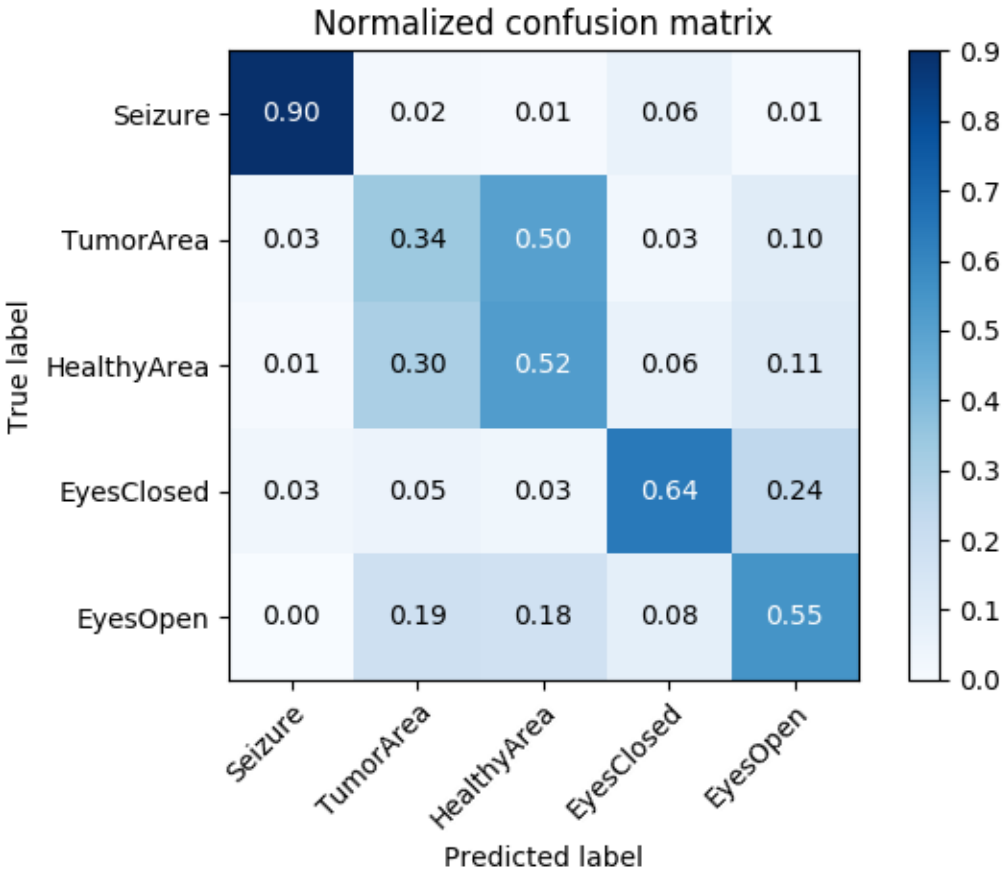
Total FP comp = 2323968+80 = 2324048

c.

Plot and attach the learning curves and the confusion matrix for your RNN model in
your report. [2 points]

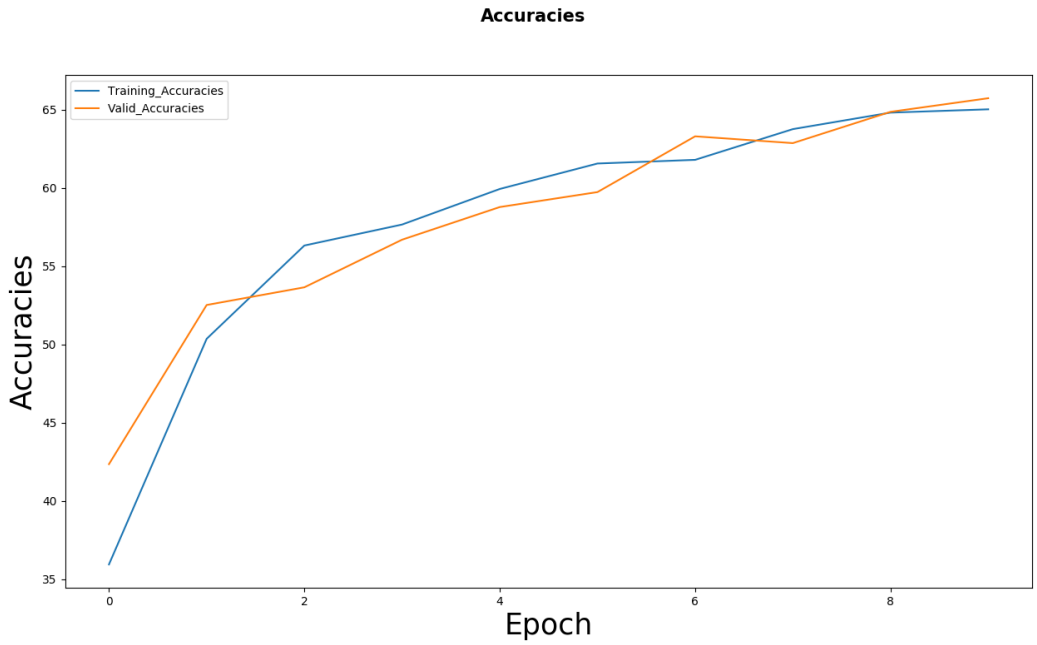


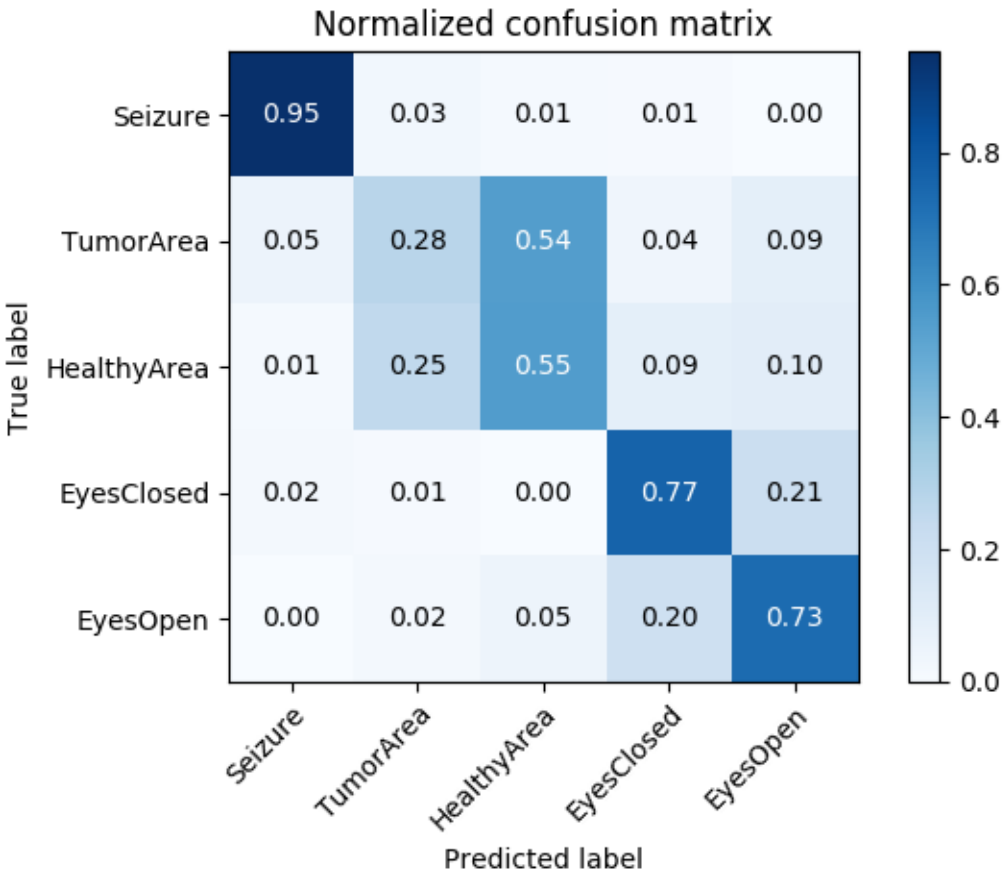


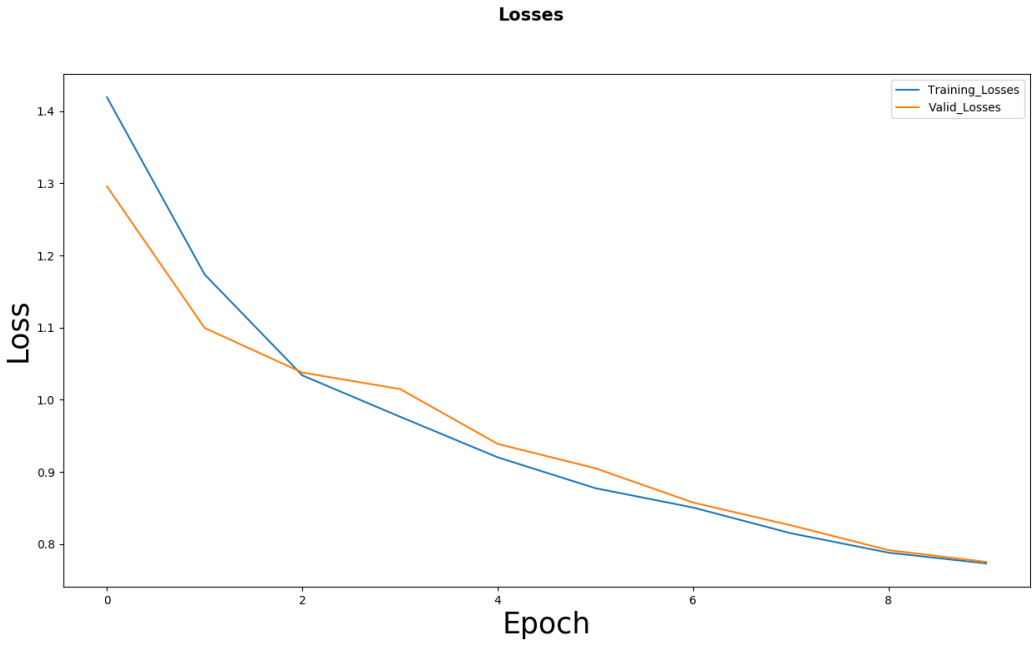


d.
Explain your architecture and techniques used. Briefly discuss about the result with

plots. [3 points]
I basically used an extra relu activation function and 2 layers instead of just 1 GRU layer. This change improved all three graphs for accuracy, loss and confusion matrix for both training and validation set.





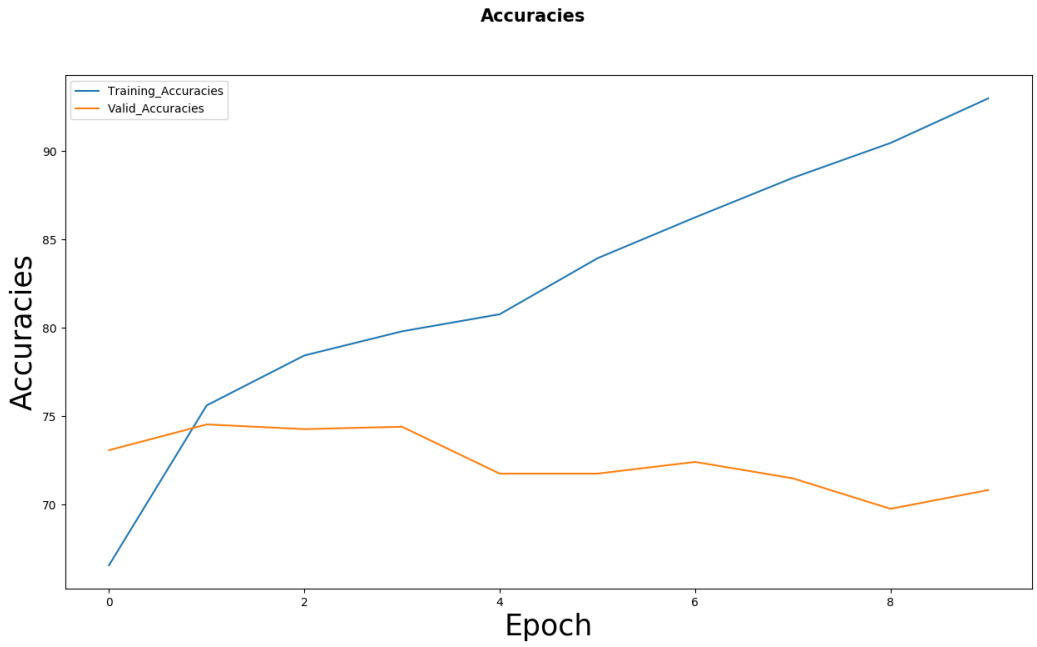


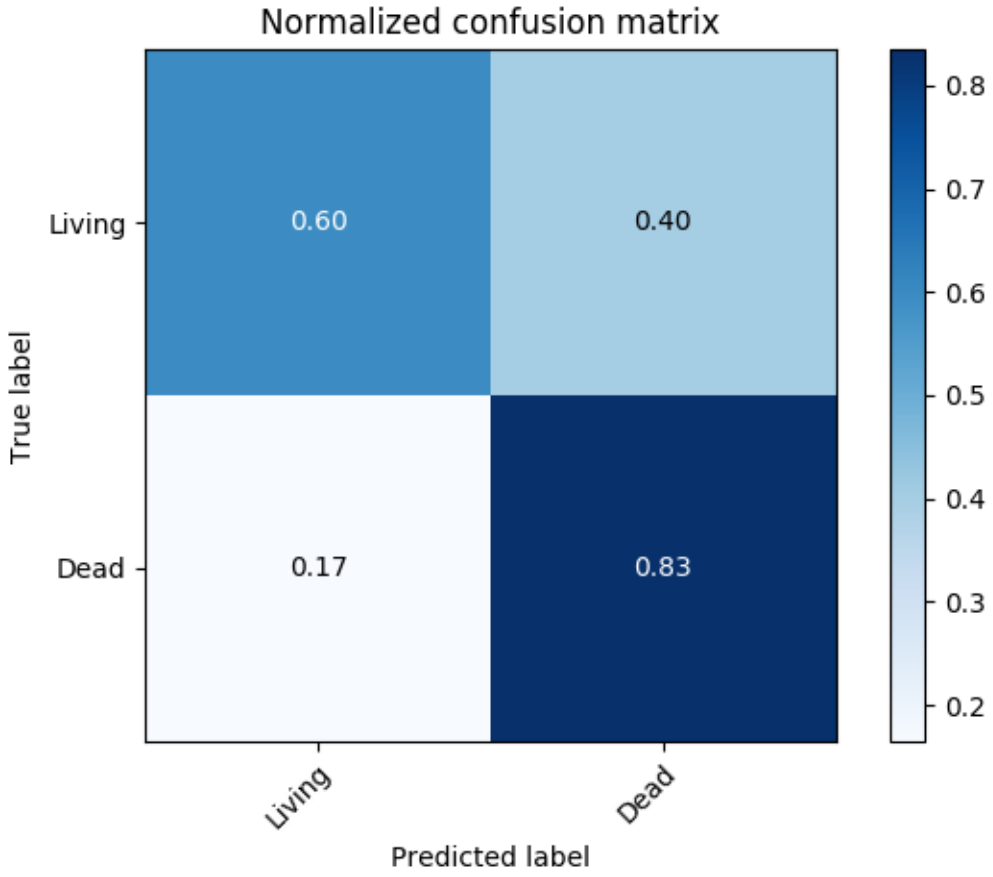
2 Mortality Prediction with RNN

2.3 Building Model

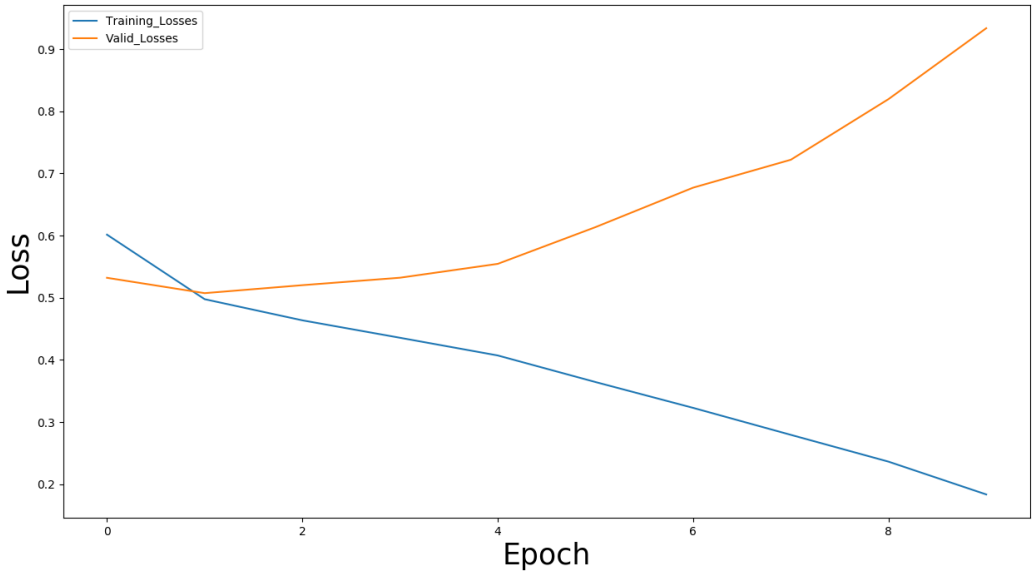
b.
Explain your architecture and techniques used. Briefly discuss about the result with
plots. [5 points]

I used 2 layers instead of 1 and added a leakyrelu with a negative slope of 0.02 to improve results upon the default network given in the instructions. I also included Xavier initialization for the last dense layer and dropout of 0.5 for GRU layer. The model does pretty well for training set confusion matrix shows high prediction to true ratios. Validation is going off though as seen in loss graph. Early stopping would help here.




Normalized confusion matrix
Living
Dead
True label
0.60
0.40
0.17
0.83
Living
Dead
Predicted label
0.8
0.7
0.6
0.5
0.4
0.3
0.2

Losses


Training_Losses
Valid_Losses
Loss
Epoch
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
2
4
6
8