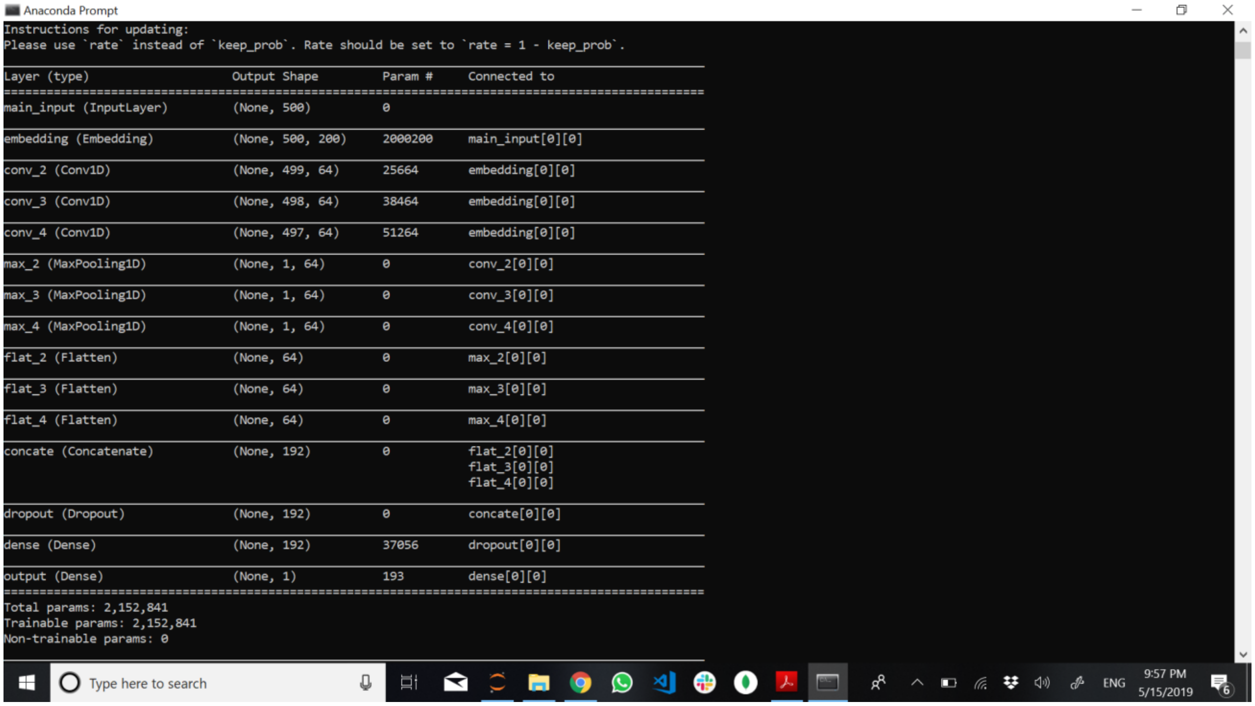
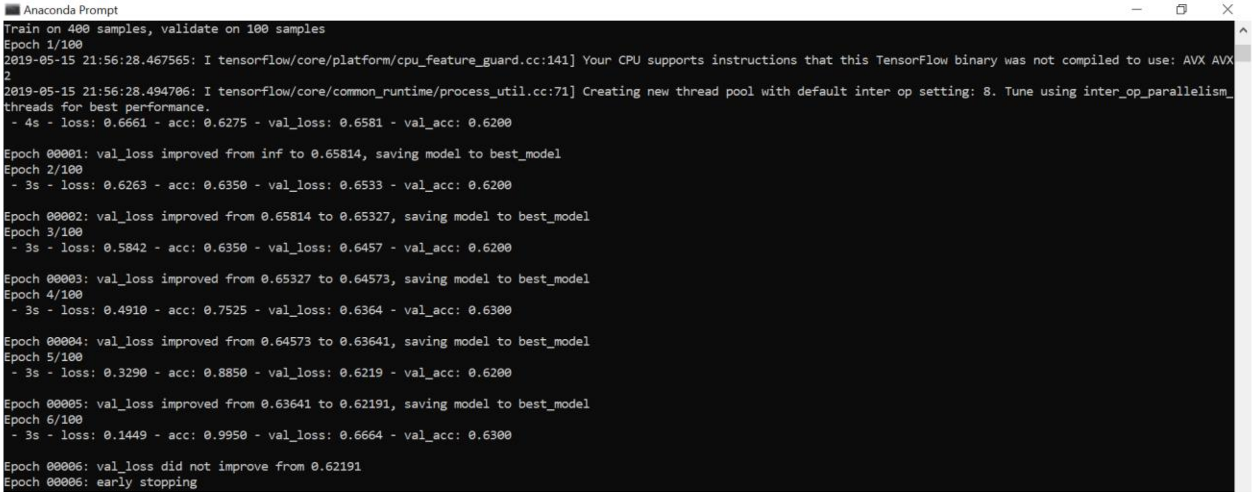
Assignment 7

I was not able to understand to question properly so I was not able to make model with two inputs. I made two different CNN left_cnn and right_cnn but I didn’t know how to concatenate them so I simply made model which will take one column as input and then train them. I have also made pretrained wordvectors for that. Here below is the model summary for just one column so I just got accuracy **64.0%**.

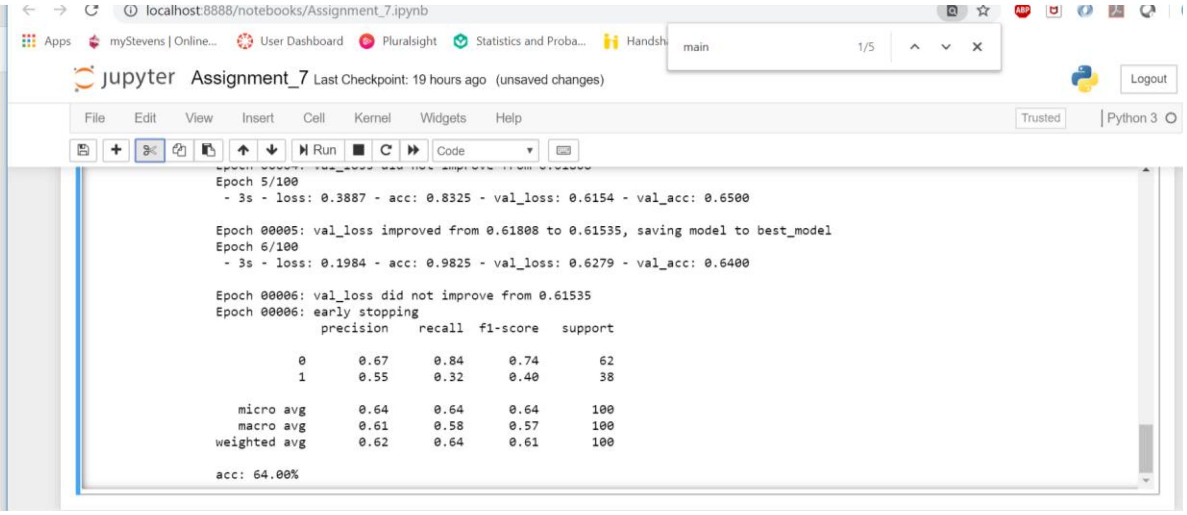
Please give me partial marks for my work if possible.



```
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
main_input (InputLayer)         (None, 500)          0
__________________________________________________________________________________________________
embedding (Embedding)           (None, 500, 200)     2000200     main_input[0][0]
__________________________________________________________________________________________________
conv_2 (Conv1D)                 (None, 499, 64)      25664       embedding[0][0]
__________________________________________________________________________________________________
conv_3 (Conv1D)                 (None, 498, 64)      38464       embedding[0][0]
__________________________________________________________________________________________________
conv_4 (Conv1D)                 (None, 497, 64)      51264       embedding[0][0]
__________________________________________________________________________________________________
max_2 (MaxPooling1D)            (None, 1, 64)        0           conv_2[0][0]
__________________________________________________________________________________________________
max_3 (MaxPooling1D)            (None, 1, 64)        0           conv_3[0][0]
__________________________________________________________________________________________________
max_4 (MaxPooling1D)            (None, 1, 64)        0           conv_4[0][0]
__________________________________________________________________________________________________
flat_2 (Flatten)                (None, 64)           0           max_2[0][0]
__________________________________________________________________________________________________
flat_3 (Flatten)                (None, 64)           0           max_3[0][0]
__________________________________________________________________________________________________
flat_4 (Flatten)                (None, 64)           0           max_4[0][0]
__________________________________________________________________________________________________
concate (Concatenate)           (None, 192)          0           flat_2[0][0]
                                                                 flat_3[0][0]
                                                                 flat_4[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 192)          0           concate[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 192)          37056       dropout[0][0]
__________________________________________________________________________________________________
output (Dense)                  (None, 1)            193         dense[0][0]
==================================================================================================
Total params: 2,152,841
Trainable params: 2,152,841
Non-trainable params: 0
__________________________________________________________________________________________________
```



```
Train on 400 samples, validate on 100 samples
Epoch 1/100
2019-05-15 21:56:28.467565: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX
2
2019-05-15 21:56:28.494706: I tensorflow/core/common_runtime/process_util.cc:71] Creating new thread pool with default inter op setting: 8. Tune using inter_op_parallelism_
threads for best performance.
 - 4s - loss: 0.6661 - acc: 0.6275 - val_loss: 0.6581 - val_acc: 0.6200

Epoch 00001: val_loss improved from inf to 0.65814, saving model to best_model
Epoch 2/100
 - 3s - loss: 0.6263 - acc: 0.6350 - val_loss: 0.6533 - val_acc: 0.6200

Epoch 00002: val_loss improved from 0.65814 to 0.65327, saving model to best_model
Epoch 3/100
 - 3s - loss: 0.5842 - acc: 0.6350 - val_loss: 0.6457 - val_acc: 0.6200

Epoch 00003: val_loss improved from 0.65327 to 0.64573, saving model to best_model
Epoch 4/100
 - 3s - loss: 0.4910 - acc: 0.7525 - val_loss: 0.6364 - val_acc: 0.6300

Epoch 00004: val_loss improved from 0.64573 to 0.63641, saving model to best_model
Epoch 5/100
 - 3s - loss: 0.3290 - acc: 0.8850 - val_loss: 0.6219 - val_acc: 0.6200

Epoch 00005: val_loss improved from 0.63641 to 0.62191, saving model to best_model
Epoch 6/100
 - 3s - loss: 0.1449 - acc: 0.9950 - val_loss: 0.6664 - val_acc: 0.6300

Epoch 00006: val_loss did not improve from 0.62191
Epoch 00006: early stopping
```



```
Epoch 00004: val_loss did not improve from 0.61808
Epoch 5/100
 - 3s - loss: 0.3887 - acc: 0.8325 - val_loss: 0.6154 - val_acc: 0.6500

Epoch 00005: val_loss improved from 0.61808 to 0.61535, saving model to best_model
Epoch 6/100
 - 3s - loss: 0.1984 - acc: 0.9825 - val_loss: 0.6279 - val_acc: 0.6400

Epoch 00006: val_loss did not improve from 0.61535
Epoch 00006: early stopping
              precision    recall  f1-score   support

           0       0.67      0.84      0.74        62
           1       0.55      0.32      0.40        38

   micro avg       0.64      0.64      0.64       100
   macro avg       0.61      0.58      0.57       100
weighted avg       0.62      0.64      0.61       100

acc: 64.00%
```