LSTM on Donors Choose dataset

In [1]:

```
import warnings
warnings.filterwarnings("ignore")
import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer
import re
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
from gensim.models import Word2Vec
from gensim.models import KeyedVectors
import pickle
from tqdm import tqdm
import os
from chart_studio.plotly import plotly
import plotly.offline as offline
import plotly.graph_objs as go
offline.init_notebook_mode()
from collections import Counter
```

Loading Data

In [0]:

```
data = pd.read_csv('preprocessed_data.csv')
data.head(2)
```

Out[0]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_s
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL

2 rows × 29 columns

In [0]:

```
data = data.drop(['id','teacher_id','project_submitted_datetime','project_grade_categor
y','project_resource_summary',
            'project_title','essay','titles_sw','essays_sw','project_essay_1',
            'project_essay_2','project_essay_3','project_essay_4', 'preprocessed_title
s',
           'sentimental_score','preprocessed_title_word_count'], axis=1)
data.head(3)
```

Out[0]:

	Unnamed: 0	teacher_prefix	school_state	teacher_number_of_previously_posted_pr
0	160221	Mrs.	IN	0
1	140945	Mr.	FL	7
2	21895	Ms.	AZ	1

In [0]:

```
data.shape
```

Out[0]:

```
(109248, 13)
```

In [0]:

```
data.to_csv('data.csv')
```

In [0]:

```
data = pd.read_csv('data.csv')
print(data.shape)
```

```
(109248, 14)
```

In [0]:

```
data.head(2)
```

Out[0]:

	Unnamed: 0	Unnamed: 0.1	teacher_prefix	school_state	teacher_number_of_previously
0	0	160221	Mrs.	IN	0
1	1	140945	Mr.	FL	7

In [0]:

```
data['project_is_approved'].value_counts()
```

Out[0]:

```
1    92706
0    16542
Name: project_is_approved, dtype: int64
```

In [0]:

```
y = data['project_is_approved'].values
X = data.drop(['project_is_approved'], axis=1)
X.head(1)
```

Out[0]:

	Unnamed: 0	Unnamed: 0.1	teacher_prefix	school_state	teacher_number_of_previously
0	0	160221	Mrs.	IN	0

Concatenating numerical features

In [0]:

```
X['Numerical_features'] = X['teacher_number_of_previously_posted_projects']+ X['price']
+ X['quantity']+ X['Numerical digits in summary']
X.head(2)
```

Out[0]:

	Unnamed: 0	Unnamed: 0.1	teacher_prefix	school_state	teacher_number_of_previousl
0	0	160221	Mrs.	IN	0
1	1	140945	Mr.	FL	7

In [0]:

```
X.shape
```

Out[0]:

```
(109248, 14)
```

Preprocessing teacher_prefix column as it has special characters & empty values

In [0]:

```
X['teacher_prefix'].value_counts()
```

Out[0]:

```
Mrs.       57269
Ms.        38955
Mr.        10648
Teacher     2360
Dr.           13
none           3
Name: teacher_prefix, dtype: int64
```

In [0]:

```
X['teacher_prefix']=X['teacher_prefix'].replace('none','Mrs.')
```

In [0]:

```
X['teacher_prefix'].value_counts()
```

Out[0]:

```
Mrs.       57272
Ms.        38955
Mr.        10648
Teacher     2360
Dr.           13
Name: teacher_prefix, dtype: int64
```

In [0]:

```
X['teacher_prefix'].unique()
```

Out[0]:

```
array(['Mrs.', 'Mr.', 'Ms.', 'Teacher', 'Dr.'], dtype=object)
```

In [0]:

```
#Replace special characters
X['teacher_prefix'] = X['teacher_prefix'].str.replace('.','')
X.head(2)
```

Out[0]:

	Unnamed: 0	Unnamed: 0.1	teacher_prefix	school_state	teacher_number_of_previously
0	0	160221	Mrs	IN	0
1	1	140945	Mr	FL	7

Splitting data into Train and cross validation(or test): Stratified Sampling

In [0]:

```
# train test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, stratify=y)
X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=0.20, strat
ify=y_train)
```

In [0]:

```
print(X_train.shape, y_train.shape)
print(X_cv.shape, y_cv.shape)
print(X_test.shape, y_test.shape)
```

```
(61178, 14) (61178,)
(15295, 14) (15295,)
(32775, 14) (32775,)
```

Model -1

1.1 Encoding text & other features

In [0]:

```
import warnings
warnings.filterwarnings("ignore")
from collections import defaultdict
import matplotlib.pyplot as plt
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Dropout, LSTM, BatchNormalization,concatenate,Flatten,Embeddin
g,Dense,Dropout,MaxPooling2D,Reshape
from keras.models import Sequential
from keras import Model,Input
from keras.layers.convolutional import Conv2D,Conv1D
import keras.backend as k
from sklearn.metrics import roc_auc_score
import tensorflow as tf
import keras
from keras.initializers import he_normal,glorot_normal
from keras.regularizers import l1,l2
from keras.callbacks import Callback, EarlyStopping, ModelCheckpoint,LearningRateSchedu
ler
from time import time
from tensorflow.python.keras.callbacks import TensorBoard
from IPython.display import SVG, display
from keras.preprocessing.text import one_hot
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers.embeddings import Embedding
```

Finding vocabulary for each feature

In [0]:

```
def unique(column):
    counter= CountVectorizer(lowercase=False) #columns like teacher_prefix have upperca
se letters
    matrix= counter.fit_transform(column.values)
    return matrix
```

In [0]:

```
essay_count= unique(X_train['preprocessed_essays'])
state_count= unique(X_train['school_state'])
grade_count= unique(X_train['preprocessed_project_grade_category'])
subject_cat_count= unique(X_train['clean_categories'])
subject_subcat_count= unique(X_train['clean_subcategories'])
teacher_prefix_count= unique(X_train['teacher_prefix'])
print('Essay:',essay_count.shape)
print('State:',state_count.shape)
print('Grade:',grade_count.shape)
print('Category:',subject_cat_count.shape)
print('Subcategory:',subject_subcat_count.shape)
print('Teacher prefix:',teacher_prefix_count.shape)
```

```
Essay: (61178, 44985)
State: (61178, 51)
Grade: (61178, 4)
Category: (61178, 9)
Subcategory: (61178, 30)
Teacher prefix: (61178, 5)
```

Converting pandas numerical features column to a ndarray

In [0]:

```
train_rem_inp= X_train['Numerical_features'].values
cv_rem_inp= X_cv['Numerical_features'].values
test_rem_inp= X_test['Numerical_features'].values
```

In [0]:

```
def encoder(feature):
    t = Tokenizer()
    t.fit_on_texts(feature)
    vocab_size = len(t.word_index) + 1
    # integer encode the documents
    encoded_docs = t.texts_to_sequences(feature)
    return encoded_docs,vocab_size,t
```

In [0]:

```
def padding(encoded_docs,max_length):
    padded_docs = pad_sequences(encoded_docs, maxlen=max_length, padding='post')
    return padded_docs
```

Encoding & padding essay

In [0]:

```
#train data
docs,vocab,t1=encoder(X_train.preprocessed_essays)
print(vocab)
```

```
45014
```

In [0]:

```
train_essay_padded = padding(docs,500)
print(train_essay_padded.shape)
print(train_essay_padded[5])
```

```
(61178, 500)
[   1  934   52  414   16  713 2169 3835   16    2   34   14  845  184
  129    5 1135 1426 1108  247  109 1166  338   61   22   67  189  519
   95   50   61   17    2  402   54  136   70  728   81   58   46  466
  364  687   58 2485  718  354  550   26   62  139    9  247    4   60
  221    1   25   78   99  878   39  445 1006   24  496  839  248  137
 7749 2622  160 2181  306    1  147  124  576 1444    1    9  129  179
  114 1282  679   34  406  403 1459 2520 8126   10  523  170   23  248
   18   15   10   71  188   50  179  169  470 7478  182 3193  217 1110
  248   18   21   15  147  124 1845   90  445 2376 2404  172   49   93
  179  114 1103 1068  958  149  114  443   42   18  530 4712    4   18
  456    1  194   12  650   10  179  318 4088    2  158    2   76  445
  163  243 5060 5357  243  813   22 5684 2133  243    4  414  362 1095
 8441  410  499 1891 1915  470 1047 8442   85    3 2800 1064  718  904
   32 5297  101   29   71   50  236  169 7478  480  149 3853  290   32
    8    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0]
```

In [0]:

```
#cv data
docs = t1.texts_to_sequences(X_cv.preprocessed_essays)
cv_essay_padded = padding(docs,500)
print(cv_essay_padded.shape)
print(cv_essay_padded[5])
```

```
(15295, 500)
[   4   53  567   53    7    1   56   36  181 1302  632   72   52    4
   10  301   94 9503  161  118  210    2 2944 2175  163 5770 2157   67
    4  304  132  241 2876 1018  217  464  118   36  396  133  241   94
   29  236  202   94   36  202 2564    6   29  236 3128 5042 1102    4
    1   94   29  236   23  422   85    1   37 2197  422 5918  468  497
   90  760  483  895   11   90   63    1   31   65  216  119  448  283
 1213   85  294    1   37  265   80 6397  305  463  987   13   11 3097
    1   33  626  113   76   42   32 1290  167 5137  613  132   90  116
 7346  894  241 1896   74    3 1076   28   11  197 1158    3    4   13
    2    6   52  222  625  216  357  537  463  167  785  730    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0]
```

In [0]:

```
#test data
docs = t1.texts_to_sequences(X_test.preprocessed_essays)
test_essay_padded = padding(docs,500)
print(test_essay_padded.shape)
print(test_essay_padded[5])
```

```
(32775, 500)
[    1  1040  2146   809    51   109    79     5    23    99    14     4
   895  2044   254  1174     1    89   140   671   109   297    14  3183
    45     1    57    88    19   278   130    31    88    19    74   242
     2   934   765    64   155     1   153   470    68   118    62  1240
    94  2087    77   122   158     1     9     7   115    31   179  1044
   518   249     6    62     1   179    55    20   194   217   464    10
   301   929   518   139    62     1   299     5    20   380   415   383
    31   819  1586    39   518  1445  3636    20   217   436   317    52
   247  2338   467    39   882   518  5724    48  1007   518     1   892
  2048  2582   163  4050   819   149 12929   163   496   312    40   518
   179    55  1121    42     6   456    28     2  2768   354   163  4050
   819     8     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0]
```

Encoding & padding school_state

In [0]:

```
#train data
docs,vocab,t=encoder(X_train.school_state)
print(vocab)
train_state_padded = padding(docs,1)
print(train_state_padded.shape)
print(train_state_padded[5])
```

```
52
(61178, 1)
[27]
```

In [0]:

```
#cv data
docs = t.texts_to_sequences(X_cv.school_state)
cv_state_padded = padding(docs,1)
print(cv_state_padded.shape)
print(cv_state_padded[5])
```

```
(15295, 1)
[6]
```

In [0]:

```
#test data
docs = t.texts_to_sequences(X_test.school_state)
test_state_padded = padding(docs,1)
print(test_state_padded.shape)
print(test_state_padded[5])
```

```
(32775, 1)
[12]
```

Encoding & padding project_grade_categories

In [0]:

```
#train data
docs,vocab,t=encoder(X_train.preprocessed_project_grade_category)
print(vocab)
train_grade_padded = padding(docs,1)
print(train_grade_padded.shape)
print(train_grade_padded[5])
```

```
10
(61178, 1)
[3]
```

In [0]:

```
#cv data
docs = t.texts_to_sequences(X_cv.preprocessed_project_grade_category)
cv_grade_padded = padding(docs,1)
print(cv_grade_padded.shape)
print(cv_grade_padded[5])
```

```
(15295, 1)
[3]
```

In [0]:

```
#test data
docs = t.texts_to_sequences(X_test.preprocessed_project_grade_category)
test_grade_padded = padding(docs,1)
print(test_grade_padded.shape)
print(test_grade_padded[5])
```

```
(32775, 1)
[3]
```

Encoding & padding clean_categories

In [0]:

```
#train data
docs,vocab,t=encoder(X_train.clean_categories)
print(vocab)
train_cat_padded = padding(docs,10)
print(train_cat_padded.shape)
print(train_cat_padded[5])
```

```
16
(61178, 10)
[1 2 3 4 0 0 0 0 0 0]
```

In [0]:

```
#cv data
docs = t.texts_to_sequences(X_cv.clean_categories)
cv_cat_padded = padding(docs,10)
print(cv_cat_padded.shape)
print(cv_cat_padded[5])
```

```
(15295, 10)
[1 2 0 0 0 0 0 0 0 0]
```

In [0]:

```
#test data
docs = t.texts_to_sequences(X_test.clean_categories)
test_cat_padded = padding(docs,10)
print(test_cat_padded.shape)
print(test_cat_padded[5])
```

```
(32775, 10)
[3 4 8 0 0 0 0 0 0 0]
```

Encoding & padding clean_subcategories

In [0]:

```
#train data
docs,vocab,t=encoder(X_train.clean_subcategories)
print(vocab)
train_subcat_padded = padding(docs,10)
print(train_subcat_padded.shape)
print(train_subcat_padded[5])
```

```
38
(61178, 10)
[3 4 2 0 0 0 0 0 0 0]
```

In [0]:

```
#cv data
docs = t.texts_to_sequences(X_cv.clean_subcategories)
cv_subcat_padded = padding(docs,10)
print(cv_subcat_padded.shape)
print(cv_subcat_padded[5])
```

```
(15295, 10)
[11  1  0  0  0  0  0  0  0  0]
```

In [0]:

```
#test data
docs = t.texts_to_sequences(X_test.clean_subcategories)
test_subcat_padded = padding(docs,10)
print(test_subcat_padded.shape)
print(test_subcat_padded[5])
```

```
(32775, 10)
[ 7 19 20  0  0  0  0  0  0  0]
```

Encoding & padding teacher_prefix

In [0]:

```
#train data
docs,vocab,t=encoder(X_train.teacher_prefix)
print(vocab)
train_prefix_padded = padding(docs,1)
print(train_prefix_padded.shape)
print(train_prefix_padded[5])
```

```
6
(61178, 1)
[2]
```

In [0]:

```
#cv data
docs = t.texts_to_sequences(X_cv.teacher_prefix)
cv_prefix_padded = padding(docs,1)
print(cv_prefix_padded.shape)
print(cv_prefix_padded[5])
```

```
(15295, 1)
[1]
```

In [0]:

```
#test data
docs = t.texts_to_sequences(X_test.teacher_prefix)
test_prefix_padded = padding(docs,1)
print(test_prefix_padded.shape)
print(test_prefix_padded[5])
```

```
(32775, 1)
[1]
```

1.2 Converting class labels to vectors using one-hot encoding

In [0]:

```
from keras.utils import to_categorical
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
y_cv = to_categorical(y_cv)
```

1.3 Saving all tensors for further use

In [0]:

```
#https://www.geeksforgeeks.org/numpy-save/
np.save('train_essay_padded', train_essay_padded)
np.save('cv_essay_padded', cv_essay_padded)
np.save('test_essay_padded', test_essay_padded)

np.save('train_state_padded', train_state_padded)
np.save('cv_state_padded', cv_state_padded)
np.save('test_state_padded', test_state_padded)

np.save('train_grade_padded', train_grade_padded)
np.save('cv_grade_padded', cv_grade_padded)
np.save('test_grade_padded', test_grade_padded)

np.save('train_cat_padded', train_cat_padded)
np.save('cv_cat_padded', cv_cat_padded)
np.save('test_cat_padded', test_cat_padded)

np.save('train_subcat_padded', train_subcat_padded)
np.save('cv_subcat_padded', cv_subcat_padded)
np.save('test_subcat_padded', test_subcat_padded)

np.save('train_prefix_padded', train_prefix_padded)
np.save('cv_prefix__padded', cv_prefix_padded)
np.save('test_prefix_padded', test_prefix_padded)

np.save('train_rem_inp', train_rem_inp)
np.save('cv_rem_inp', cv_rem_inp)
np.save('test_rem_inp', test_rem_inp)

np.save('y_train', y_train)
np.save('y_test', y_test)
np.save('y_cv', y_cv)
```

In [0]:

```
#Loading the tensors
train_essay_padded= np.load('/content/train_essay_padded.npy')
cv_essay_padded= np.load('/content/cv_essay_padded.npy')
test_essay_padded= np.load('/content/test_essay_padded.npy')

train_state_padded= np.load('/content/train_state_padded.npy')
cv_state_padded= np.load('/content/cv_state_padded.npy')
test_state_padded= np.load('/content/test_state_padded.npy')

train_grade_padded= np.load('/content/train_grade_padded.npy')
cv_grade_padded= np.load('/content/cv_grade_padded.npy')
test_grade_padded= np.load('/content/test_grade_padded.npy')

train_cat_padded= np.load('/content/train_cat_padded.npy')
cv_cat_padded= np.load('/content/cv_cat_padded.npy')
test_cat_padded= np.load('/content/test_cat_padded.npy')

train_subcat_padded= np.load('/content/train_subcat_padded.npy')
cv_subcat_padded= np.load('/content/cv_subcat_padded.npy')
test_subcat_padded= np.load('/content/test_subcat_padded.npy')

train_prefix_padded= np.load('/content/train_prefix_padded.npy')
cv_prefix_padded= np.load('/content/cv_prefix__padded.npy')
test_prefix_padded= np.load('/content/test_prefix_padded.npy')

train_rem_inp= np.load('/content/train_rem_inp.npy')
cv_rem_inp= np.load('/content/cv_rem_inp.npy')
test_rem_inp= np.load('/content/test_rem_inp.npy')

y_train= np.load('/content/y_train.npy')
y_test= np.load('/content/y_test.npy')
y_cv= np.load('/content/y_cv.npy')
```

1.4 Loading the pre-trained glove model

In [0]:

```
with open('glove_vectors', 'rb') as f:
    glove = pickle.load(f)
print ("Done.",len(glove)," words loaded!")
```

```
Done. 51510  words loaded!
```

In [0]:

```
type(glove)
```

Out[0]:

```
dict
```

1.5 Defining the performance metric[ROC]

In [0]:

```
def auc( y_true, y_pred ) :
    score = tf.py_func( lambda y_true, y_pred : roc_auc_score( y_true, y_pred, average=
'macro', sample_weight=None).astype('float32'),
                        [y_true, y_pred],'float32',stateful=True,name='sklearnAUC')
    return score
```

1.6 Creating the 2D Embedding matrix using Glove vectors

In [0]:

```
#Credits: https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-ke
ras/

embedding_matrix = np.zeros((45014, 300))
for word, i in t1.word_index.items():
    embedding_vector = glove.get(word)
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector
```

In [0]:

```
np.save('embedding_matrix',embedding_matrix)
```

In [0]:

```
embedding_matrix= np.load('/content/embedding_matrix(1).npy')
```

1.7 LSTM model

https://i.imgur.com/w395Yk9.png (https://i.imgur.com/w395Yk9.png)

In [0]:

```
import warnings
warnings.filterwarnings("ignore")
from collections import defaultdict
import matplotlib.pyplot as plt
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Dropout, LSTM, BatchNormalization,concatenate,Flatten,Embeddin
g,Dense,Dropout,MaxPooling2D,Reshape
from keras.models import Sequential
from keras import Model,Input
from keras.layers.convolutional import Conv2D,Conv1D
import keras.backend as k
from sklearn.metrics import roc_auc_score
import tensorflow as tf
import keras
from keras.initializers import he_normal,glorot_normal
from keras.regularizers import l1,l2
from keras.callbacks import Callback, EarlyStopping, ModelCheckpoint,LearningRateSchedu
ler
from time import time
from tensorflow.python.keras.callbacks import TensorBoard
from IPython.display import SVG, display
from keras.layers import LeakyReLU
```

1.7.1 Architecture

In [0]:

```
import keras.backend as K
K.clear_session()
```

In [0]:

```
#essay
input1 = Input(shape=(500,))
i1 = Embedding(input_dim=45014,output_dim= 300,input_length=500,weights=[embedding_matr
ix],trainable=False)(input1)
i1 = Dropout(0.5)(i1)
i1 = LSTM(128,kernel_initializer='he_normal',recurrent_dropout=0.5,kernel_regularizer=l
2(0.001),return_sequences=True)(i1)
i1= LeakyReLU(alpha = 0.5)(i1)
f1 = Flatten()(i1)

#school_state
input2 = Input(shape=(1,))
i2 = Embedding(input_dim= 52,output_dim= 2,input_length=1)(input2)
f2 = Flatten()(i2)

#project grade category
input3 = Input(shape=(1,))
i3 = Embedding(input_dim= 10,output_dim= 2,input_length=1)(input3)
f3 = Flatten()(i3)

#clean_categories
input4 = Input(shape=(10,))
i4 = Embedding(input_dim=16,output_dim= 2,input_length=10)(input4)
f4 = Flatten()(i4)

#clean_subcategories
input5 = Input(shape=(10,))
i5 = Embedding(input_dim= 38,output_dim= 2,input_length=10)(input5)
f5 = Flatten()(i5)

#teacher_prefix
input6 = Input(shape=(1,))
i6 = Embedding(input_dim= 6,output_dim= 2,input_length=1)(input6)
f6 = Flatten()(i6)

#concatenated numerical features
input7 = Input(shape=(1,))
i7 = Dense(16,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.
001))(input7)

#concatenating all the inputs
concat = concatenate([f1,f2,f3,f4,f5,f6,i7])


l = Dense(128,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.
001))(concat)
l = Dropout(0.5)(l)
l = Dense(64,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.0
01))(l)
l = Dropout(0.5)(l)
l = BatchNormalization()(l)
l = Dense(32,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.0
01))(l)
l = Dropout(0.5)(l)
output = Dense(2, activation = 'softmax')(l)
```

In [15]:

```
# create model with seven inputs
model = Model(inputs=[input1,input2,input3,input4,input5,input6,input7], outputs=[outpu
t])
model.summary()
```

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_8 (InputLayer)            (None, 500)          0
__________________________________________________________________________________________________
embedding_7 (Embedding)         (None, 500, 300)     13504200    input_8[0][0]
__________________________________________________________________________________________________
dropout_5 (Dropout)             (None, 500, 300)     0           embedding_7[0][0]
__________________________________________________________________________________________________
lstm_2 (LSTM)                   (None, 500, 128)     219648      dropout_5[0][0]
__________________________________________________________________________________________________
input_9 (InputLayer)            (None, 1)            0
__________________________________________________________________________________________________
input_10 (InputLayer)           (None, 1)            0
__________________________________________________________________________________________________
input_11 (InputLayer)           (None, 10)           0
__________________________________________________________________________________________________
input_12 (InputLayer)           (None, 10)           0
__________________________________________________________________________________________________
input_13 (InputLayer)           (None, 1)            0
__________________________________________________________________________________________________
leaky_re_lu_2 (LeakyReLU)       (None, 500, 128)     0           lstm_2[0][0]
__________________________________________________________________________________________________
embedding_8 (Embedding)         (None, 1, 2)         104         input_9[0][0]
__________________________________________________________________________________________________
embedding_9 (Embedding)         (None, 1, 2)         20          input_10[0][0]
__________________________________________________________________________________________________
embedding_10 (Embedding)        (None, 10, 2)        32          input_11[0][0]
__________________________________________________________________________________________________
embedding_11 (Embedding)        (None, 10, 2)        76          input_12[0][0]
__________________________________________________________________________________________________
embedding_12 (Embedding)        (None, 1, 2)         12          input_13[0][0]
__________________________________________________________________________________________________
```

```
input_14 (InputLayer)           (None, 1)            0
__________________________________________________________________________________________________
flatten_7 (Flatten)             (None, 64000)        0           leaky_re_lu_2[0][0]
__________________________________________________________________________________________________
flatten_8 (Flatten)             (None, 2)            0           embedding_8[0][0]
__________________________________________________________________________________________________
flatten_9 (Flatten)             (None, 2)            0           embedding_9[0][0]
__________________________________________________________________________________________________
flatten_10 (Flatten)            (None, 20)           0           embedding_10[0][0]
__________________________________________________________________________________________________
flatten_11 (Flatten)            (None, 20)           0           embedding_11[0][0]
__________________________________________________________________________________________________
flatten_12 (Flatten)            (None, 2)            0           embedding_12[0][0]
__________________________________________________________________________________________________
dense_6 (Dense)                 (None, 16)           32          input_14[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 64062)        0           flatten_7[0][0]
                                                                 flatten_8[0][0]
                                                                 flatten_9[0][0]
                                                                 flatten_10[0][0]
                                                                 flatten_11[0][0]
                                                                 flatten_12[0][0]
                                                                 dense_6[0][0]
__________________________________________________________________________________________________
dense_7 (Dense)                 (None, 128)          8200064     concatenate_2[0][0]
__________________________________________________________________________________________________
dropout_6 (Dropout)             (None, 128)          0           dense_7[0][0]
__________________________________________________________________________________________________
dense_8 (Dense)                 (None, 64)           8256        dropout_6[0][0]
__________________________________________________________________________________________________
dropout_7 (Dropout)             (None, 64)           0           dense_8[0][0]
```

```
_________________________________________________________________
_____________________________
batch_normalization_2 (BatchNor (None, 64)           256         dropout
_7[0][0]
_________________________________________________________________
_____________________________
dense_9 (Dense)                 (None, 32)           2080        batch_n
ormalization_2[0][0]
_________________________________________________________________
_____________________________
dropout_8 (Dropout)             (None, 32)           0           dense_9
[0][0]
_________________________________________________________________
_____________________________
dense_10 (Dense)                (None, 2)            66          dropout
_8[0][0]
=================================================================
==========================
Total params: 21,934,846
Trainable params: 8,430,518
Non-trainable params: 13,504,328
_________________________________________________________________
_____________________________
```

In [0]:

```
model.load_weights("/content/weights2it.best.hdf5")
```

In [0]:

```
model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.Adam(lr=0.001
,decay = 1e-4),metrics=[auc])
```

1.7.2 Checkpointing the model and creating the callback list

In [0]:

```
#https://machinelearningmastery.com/check-point-deep-learning-models-keras/


from keras.callbacks import ModelCheckpoint
from keras.callbacks import CSVLogger
import matplotlib.pyplot as plt
from tensorflow.python.keras.callbacks import TensorBoard
from keras.callbacks import TensorBoard

tensorboard = TensorBoard(log_dir='logs'.format(time()))
filepath="weights.best.hdf5"
checkpoints = ModelCheckpoint(filepath, monitor='val_auc', verbose=1, save_best_only=Tr
ue, mode='max')
train_results = CSVLogger('train_results.log') #storing the training results in a panda
s dataframe
callbacks_list = [checkpoints, tensorboard, train_results]
```

In [19]:

```
#trial 3
from keras.callbacks import ModelCheckpoint
from keras.callbacks import CSVLogger
import matplotlib.pyplot as plt
from tensorflow.python.keras.callbacks import TensorBoard
from keras.callbacks import TensorBoard
import tensorflow as tf
import datetime
import keras
from tensorboardcolab import *
from keras.callbacks import ReduceLROnPlateau

#https://github.com/taomanwai/tensorboardcolab/
tbc=TensorBoardColab()
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2,
                              patience=1, min_lr=0.002,verbose = 1)

filepath="weights3it.best.hdf5"
checkpoints = ModelCheckpoint(filepath, monitor='val_auc', verbose=1, save_best_only=True, mode='max')
train_results = CSVLogger('train_results_3.log') #storing the training results in a pandas dataframe
callbacks_list = [checkpoints, TensorBoardColabCallback(tbc), train_results]
```

```
Wait for 8 seconds...
TensorBoard link:
https://6bfd7f6c.ngrok.io
```

1.7.3 Fitting the model in batches

In [0]:

```
# finding the class weights before fitting the model
from sklearn.utils import compute_class_weight
class_wts = compute_class_weight("balanced", classes= np.unique(y),y=y)
print(class_wts)
```

```
[3.30214001 0.58921753]
```

In [0]:

```
np.save('class_wts', class_wts)
```

In [0]:

```
class_wts= np.load('class_wts.npy')
```

In [0]:

```
history=model.fit([train_essay_padded,train_state_padded,train_grade_padded,train_cat_p
added,train_subcat_padded,
           train_prefix_padded,train_rem_inp], y_train, nb_epoch=30,verbose=1,batch_siz
e=600,
          validation_data=([cv_essay_padded,cv_state_padded,cv_grade_padded,cv_cat_padd
ed,cv_subcat_padded,cv_prefix_padded,cv_rem_inp],y_cv),
          callbacks =callbacks_list,class_weight = class_wts)
```

```
Train on 61178 samples, validate on 15295 samples
Epoch 1/30
61178/61178 [==============================] - 1141s 19ms/step - loss: 1.6
096 - auc: 0.5066 - val_loss: 1.0287 - val_auc: 0.5733

Epoch 00001: val_auc improved from -inf to 0.57332, saving model to weight
s.best.hdf5
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/callb
acks/tensorboard_v1.py:343: The name tf.Summary is deprecated. Please use
tf.compat.v1.Summary instead.

Epoch 2/30
61178/61178 [==============================] - 1131s 18ms/step - loss: 0.9
851 - auc: 0.5216 - val_loss: 0.7559 - val_auc: 0.5810

Epoch 00002: val_auc improved from 0.57332 to 0.58104, saving model to wei
ghts.best.hdf5
Epoch 3/30
61178/61178 [==============================] - 1131s 18ms/step - loss: 0.7
588 - auc: 0.5244 - val_loss: 0.6432 - val_auc: 0.5890

Epoch 00003: val_auc improved from 0.58104 to 0.58904, saving model to wei
ghts.best.hdf5
Epoch 4/30
61178/61178 [==============================] - 1147s 19ms/step - loss: 0.6
617 - auc: 0.5301 - val_loss: 0.5913 - val_auc: 0.5909

Epoch 00004: val_auc improved from 0.58904 to 0.59088, saving model to wei
ghts.best.hdf5
Epoch 5/30
61178/61178 [==============================] - 1126s 18ms/step - loss: 0.6
143 - auc: 0.5319 - val_loss: 0.5772 - val_auc: 0.5891

Epoch 00005: val_auc did not improve from 0.59088
Epoch 6/30
61178/61178 [==============================] - 1165s 19ms/step - loss: 0.5
824 - auc: 0.5443 - val_loss: 0.5529 - val_auc: 0.5940

Epoch 00006: val_auc improved from 0.59088 to 0.59400, saving model to wei
ghts.best.hdf5
Epoch 7/30
61178/61178 [==============================] - 1153s 19ms/step - loss: 0.5
670 - auc: 0.5404 - val_loss: 0.5408 - val_auc: 0.5903

Epoch 00007: val_auc did not improve from 0.59400
Epoch 8/30
61178/61178 [==============================] - 1139s 19ms/step - loss: 0.5
482 - auc: 0.5486 - val_loss: 0.5286 - val_auc: 0.5933

Epoch 00008: val_auc did not improve from 0.59400
Epoch 9/30
61178/61178 [==============================] - 1155s 19ms/step - loss: 0.5
376 - auc: 0.5473 - val_loss: 0.5197 - val_auc: 0.5964

Epoch 00009: val_auc improved from 0.59400 to 0.59642, saving model to wei
ghts.best.hdf5
Epoch 10/30
61178/61178 [==============================] - 1147s 19ms/step - loss: 0.5
286 - auc: 0.5511 - val_loss: 0.5227 - val_auc: 0.5907

Epoch 00010: val_auc did not improve from 0.59642
```

```
Epoch 11/30
61178/61178 [==============================] - 1159s 19ms/step - loss: 0.5
208 - auc: 0.5506 - val_loss: 0.5029 - val_auc: 0.5938

Epoch 00011: val_auc did not improve from 0.59642
Epoch 12/30
61178/61178 [==============================] - 1112s 18ms/step - loss: 0.5
090 - auc: 0.5559 - val_loss: 0.4969 - val_auc: 0.5951

Epoch 00012: val_auc did not improve from 0.59642
Epoch 13/30
61178/61178 [==============================] - 1107s 18ms/step - loss: 0.5
021 - auc: 0.5607 - val_loss: 0.4901 - val_auc: 0.5966

Epoch 00013: val_auc improved from 0.59642 to 0.59663, saving model to wei
ghts.best.hdf5
Epoch 14/30
61178/61178 [==============================] - 1099s 18ms/step - loss: 0.4
949 - auc: 0.5668 - val_loss: 0.4855 - val_auc: 0.5942

Epoch 00014: val_auc did not improve from 0.59663
Epoch 15/30
61178/61178 [==============================] - 1105s 18ms/step - loss: 0.4
899 - auc: 0.5673 - val_loss: 0.4807 - val_auc: 0.5960

Epoch 00015: val_auc did not improve from 0.59663
Epoch 16/30
61178/61178 [==============================] - 1104s 18ms/step - loss: 0.4
848 - auc: 0.5706 - val_loss: 0.4789 - val_auc: 0.5985

Epoch 00016: val_auc improved from 0.59663 to 0.59851, saving model to wei
ghts.best.hdf5
Epoch 17/30
61178/61178 [==============================] - 1099s 18ms/step - loss: 0.4
802 - auc: 0.5699 - val_loss: 0.4715 - val_auc: 0.5993

Epoch 00017: val_auc improved from 0.59851 to 0.59935, saving model to wei
ghts.best.hdf5
Epoch 18/30
61178/61178 [==============================] - 1100s 18ms/step - loss: 0.4
763 - auc: 0.5712 - val_loss: 0.4692 - val_auc: 0.5985

Epoch 00018: val_auc did not improve from 0.59935
Epoch 19/30
61178/61178 [==============================] - 1080s 18ms/step - loss: 0.4
723 - auc: 0.5710 - val_loss: 0.4646 - val_auc: 0.5991

Epoch 00019: val_auc did not improve from 0.59935
Epoch 20/30
61178/61178 [==============================] - 1066s 17ms/step - loss: 0.4
693 - auc: 0.5737 - val_loss: 0.4622 - val_auc: 0.6011

Epoch 00020: val_auc improved from 0.59935 to 0.60110, saving model to wei
ghts.best.hdf5
Epoch 21/30
61178/61178 [==============================] - 1082s 18ms/step - loss: 0.4
655 - auc: 0.5764 - val_loss: 0.4581 - val_auc: 0.6095

Epoch 00021: val_auc improved from 0.60110 to 0.60953, saving model to wei
ghts.best.hdf5
Epoch 22/30
```

```
61178/61178 [==============================] - 1066s 17ms/step - loss: 0.4
621 - auc: 0.5814 - val_loss: 0.4544 - val_auc: 0.6205

Epoch 00022: val_auc improved from 0.60953 to 0.62053, saving model to wei
ghts.best.hdf5
Epoch 23/30
61178/61178 [==============================] - 1054s 17ms/step - loss: 0.4
594 - auc: 0.5993 - val_loss: 0.4530 - val_auc: 0.6441

Epoch 00023: val_auc improved from 0.62053 to 0.64413, saving model to wei
ghts.best.hdf5
Epoch 24/30
61178/61178 [==============================] - 1059s 17ms/step - loss: 0.4
562 - auc: 0.6270 - val_loss: 0.4509 - val_auc: 0.6782

Epoch 00024: val_auc improved from 0.64413 to 0.67820, saving model to wei
ghts.best.hdf5
Epoch 25/30
61178/61178 [==============================] - 1055s 17ms/step - loss: 0.4
522 - auc: 0.6400 - val_loss: 0.4440 - val_auc: 0.6823

Epoch 00025: val_auc improved from 0.67820 to 0.68227, saving model to wei
ghts.best.hdf5
Epoch 26/30
61178/61178 [==============================] - 1054s 17ms/step - loss: 0.4
495 - auc: 0.6390 - val_loss: 0.4409 - val_auc: 0.6856

Epoch 00026: val_auc improved from 0.68227 to 0.68564, saving model to wei
ghts.best.hdf5
Epoch 27/30
61178/61178 [==============================] - 1057s 17ms/step - loss: 0.4
447 - auc: 0.6430 - val_loss: 0.4435 - val_auc: 0.6812

Epoch 00027: val_auc did not improve from 0.68564
Epoch 28/30
61178/61178 [==============================] - 1056s 17ms/step - loss: 0.4
452 - auc: 0.6334 - val_loss: 0.4454 - val_auc: 0.6831

Epoch 00028: val_auc did not improve from 0.68564
Epoch 29/30
61178/61178 [==============================] - 1072s 18ms/step - loss: 0.4
422 - auc: 0.6340 - val_loss: 0.4346 - val_auc: 0.6876

Epoch 00029: val_auc improved from 0.68564 to 0.68762, saving model to wei
ghts.best.hdf5
Epoch 30/30
61178/61178 [==============================] - 1073s 18ms/step - loss: 0.4
379 - auc: 0.6432 - val_loss: 0.4299 - val_auc: 0.6967

Epoch 00030: val_auc improved from 0.68762 to 0.69673, saving model to wei
ghts.best.hdf5
```

In [0]:

```
#2nd cycle of epochs
history1=model.fit([train_essay_padded,train_state_padded,train_grade_padded,train_cat_
padded,train_subcat_padded,
           train_prefix_padded,train_rem_inp], y_train, nb_epoch=20,verbose=1,batch_siz
e=600,
          validation_data=([cv_essay_padded,cv_state_padded,cv_grade_padded,cv_cat_padd
ed,cv_subcat_padded,cv_prefix_padded,cv_rem_inp],y_cv),
          callbacks =callbacks_list,class_weight = class_wts)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/keras/backe
nd/tensorflow_backend.py:422: The name tf.global_variables is deprecated.
Please use tf.compat.v1.global_variables instead.

Train on 61178 samples, validate on 15295 samples
Epoch 1/20
61178/61178 [==============================] - 1152s 19ms/step - loss: 0.4
122 - auc: 0.6871 - val_loss: 0.4069 - val_auc: 0.7235

Epoch 00001: val_auc improved from -inf to 0.72351, saving model to weight
s2it.best.hdf5
Epoch 2/20
61178/61178 [==============================] - 1146s 19ms/step - loss: 0.4
111 - auc: 0.6906 - val_loss: 0.4029 - val_auc: 0.7261

Epoch 00002: val_auc improved from 0.72351 to 0.72607, saving model to wei
ghts2it.best.hdf5
Epoch 3/20
61178/61178 [==============================] - 1156s 19ms/step - loss: 0.4
100 - auc: 0.6933 - val_loss: 0.4010 - val_auc: 0.7268

Epoch 00003: val_auc improved from 0.72607 to 0.72683, saving model to wei
ghts2it.best.hdf5
Epoch 4/20
61178/61178 [==============================] - 1156s 19ms/step - loss: 0.4
102 - auc: 0.6910 - val_loss: 0.4008 - val_auc: 0.7259

Epoch 00004: val_auc did not improve from 0.72683
Epoch 5/20
61178/61178 [==============================] - 1163s 19ms/step - loss: 0.4
069 - auc: 0.6975 - val_loss: 0.3983 - val_auc: 0.7270

Epoch 00005: val_auc improved from 0.72683 to 0.72699, saving model to wei
ghts2it.best.hdf5
Epoch 6/20
61178/61178 [==============================] - 1163s 19ms/step - loss: 0.4
078 - auc: 0.6941 - val_loss: 0.3986 - val_auc: 0.7279

Epoch 00006: val_auc improved from 0.72699 to 0.72787, saving model to wei
ghts2it.best.hdf5
Epoch 7/20
61178/61178 [==============================] - 1165s 19ms/step - loss: 0.4
062 - auc: 0.6967 - val_loss: 0.3969 - val_auc: 0.7283

Epoch 00007: val_auc improved from 0.72787 to 0.72830, saving model to wei
ghts2it.best.hdf5
Epoch 8/20
61178/61178 [==============================] - 1161s 19ms/step - loss: 0.4
051 - auc: 0.6985 - val_loss: 0.3970 - val_auc: 0.7308

Epoch 00008: val_auc improved from 0.72830 to 0.73080, saving model to wei
ghts2it.best.hdf5
Epoch 9/20
61178/61178 [==============================] - 1168s 19ms/step - loss: 0.4
038 - auc: 0.7008 - val_loss: 0.3979 - val_auc: 0.7292

Epoch 00009: val_auc did not improve from 0.73080
Epoch 10/20
61178/61178 [==============================] - 1163s 19ms/step - loss: 0.4
034 - auc: 0.7006 - val_loss: 0.3960 - val_auc: 0.7306
```

```
Epoch 00010: val_auc did not improve from 0.73080
Epoch 11/20
61178/61178 [==============================] - 1164s 19ms/step - loss: 0.4
031 - auc: 0.7017 - val_loss: 0.3946 - val_auc: 0.7317

Epoch 00011: val_auc improved from 0.73080 to 0.73167, saving model to wei
ghts2it.best.hdf5
Epoch 12/20
61178/61178 [==============================] - 1154s 19ms/step - loss: 0.4
030 - auc: 0.7020 - val_loss: 0.4051 - val_auc: 0.7349

Epoch 00012: val_auc improved from 0.73167 to 0.73489, saving model to wei
ghts2it.best.hdf5
Epoch 13/20
61178/61178 [==============================] - 1147s 19ms/step - loss: 0.4
026 - auc: 0.7014 - val_loss: 0.3948 - val_auc: 0.7313

Epoch 00013: val_auc did not improve from 0.73489
Epoch 14/20
61178/61178 [==============================] - 1155s 19ms/step - loss: 0.4
026 - auc: 0.6961 - val_loss: 0.4004 - val_auc: 0.7308

Epoch 00014: val_auc did not improve from 0.73489
Epoch 15/20
61178/61178 [==============================] - 1171s 19ms/step - loss: 0.4
008 - auc: 0.7036 - val_loss: 0.3965 - val_auc: 0.7335

Epoch 00015: val_auc did not improve from 0.73489
Epoch 16/20
61178/61178 [==============================] - 1156s 19ms/step - loss: 0.4
005 - auc: 0.7046 - val_loss: 0.3924 - val_auc: 0.7351

Epoch 00016: val_auc improved from 0.73489 to 0.73511, saving model to wei
ghts2it.best.hdf5
Epoch 17/20
61178/61178 [==============================] - 1156s 19ms/step - loss: 0.4
006 - auc: 0.7033 - val_loss: 0.3956 - val_auc: 0.7357

Epoch 00017: val_auc improved from 0.73511 to 0.73574, saving model to wei
ghts2it.best.hdf5
Epoch 18/20
61178/61178 [==============================] - 1159s 19ms/step - loss: 0.3
996 - auc: 0.7056 - val_loss: 0.3938 - val_auc: 0.7353

Epoch 00018: val_auc did not improve from 0.73574
Epoch 19/20
61178/61178 [==============================] - 1169s 19ms/step - loss: 0.3
992 - auc: 0.7077 - val_loss: 0.3922 - val_auc: 0.7340

Epoch 00019: val_auc did not improve from 0.73574
Epoch 20/20
61178/61178 [==============================] - 1159s 19ms/step - loss: 0.3
993 - auc: 0.7064 - val_loss: 0.3939 - val_auc: 0.7364

Epoch 00020: val_auc improved from 0.73574 to 0.73643, saving model to wei
ghts2it.best.hdf5
```

In [21]:

```
#3rd cycle of epochs
history1=model.fit([train_essay_padded,train_state_padded,train_grade_padded,train_cat_
padded,train_subcat_padded,
           train_prefix_padded,train_rem_inp], y_train, nb_epoch=20,verbose=1,batch_siz
e=500,
          validation_data=([cv_essay_padded,cv_state_padded,cv_grade_padded,cv_cat_padd
ed,cv_subcat_padded,cv_prefix_padded,cv_rem_inp],y_cv),
          callbacks =callbacks_list,class_weight = class_wts)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_
core/python/ops/math_grad.py:1424: where (from tensorflow.python.ops.array
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Pleas
e use tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:1020: The name tf.assign is deprecated. Please us
e tf.compat.v1.assign instead.

Train on 61178 samples, validate on 15295 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorboard
colab/core.py:49: The name tf.summary.FileWriter is deprecated. Please use
tf.compat.v1.summary.FileWriter instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callb
acks.py:1122: The name tf.summary.merge_all is deprecated. Please use tf.c
ompat.v1.summary.merge_all instead.

Epoch 1/20
61178/61178 [==============================] - 108s 2ms/step - loss: 0.413
2 - auc: 0.6820 - val_loss: 0.4008 - val_auc: 0.7365

Epoch 00001: val_auc improved from -inf to 0.73653, saving model to weight
s3it.best.hdf5
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorboard
colab/callbacks.py:51: The name tf.Summary is deprecated. Please use tf.co
mpat.v1.Summary instead.

Epoch 2/20
61178/61178 [==============================] - 109s 2ms/step - loss: 0.410
1 - auc: 0.6981 - val_loss: 0.4028 - val_auc: 0.7347

Epoch 00002: val_auc did not improve from 0.73653
Epoch 3/20
61178/61178 [==============================] - 107s 2ms/step - loss: 0.407
7 - auc: 0.7029 - val_loss: 0.3984 - val_auc: 0.7381

Epoch 00003: val_auc improved from 0.73653 to 0.73814, saving model to wei
ghts3it.best.hdf5
Epoch 4/20
61178/61178 [==============================] - 107s 2ms/step - loss: 0.407
9 - auc: 0.7016 - val_loss: 0.3974 - val_auc: 0.7343

Epoch 00004: val_auc did not improve from 0.73814
Epoch 5/20
61178/61178 [==============================] - 107s 2ms/step - loss: 0.403
0 - auc: 0.7107 - val_loss: 0.3961 - val_auc: 0.7350

Epoch 00005: val_auc did not improve from 0.73814
Epoch 6/20
61178/61178 [==============================] - 108s 2ms/step - loss: 0.441
8 - auc: 0.6512 - val_loss: 0.4241 - val_auc: 0.7267

Epoch 00006: val_auc did not improve from 0.73814
Epoch 7/20
61178/61178 [==============================] - 108s 2ms/step - loss: 0.428
0 - auc: 0.6873 - val_loss: 0.4115 - val_auc: 0.7317
```

```
Epoch 00007: val_auc did not improve from 0.73814
Epoch 8/20
61178/61178 [==============================] - 107s 2ms/step - loss: 0.415
5 - auc: 0.7027 - val_loss: 0.4013 - val_auc: 0.7329

Epoch 00008: val_auc did not improve from 0.73814
Epoch 9/20
61178/61178 [==============================] - 107s 2ms/step - loss: 0.411
9 - auc: 0.7027 - val_loss: 0.4004 - val_auc: 0.7381

Epoch 00009: val_auc did not improve from 0.73814
Epoch 10/20
61178/61178 [==============================] - 108s 2ms/step - loss: 0.407
3 - auc: 0.7079 - val_loss: 0.4014 - val_auc: 0.7375

Epoch 00010: val_auc did not improve from 0.73814
Epoch 11/20
61178/61178 [==============================] - 108s 2ms/step - loss: 0.406
7 - auc: 0.7065 - val_loss: 0.3974 - val_auc: 0.7400

Epoch 00011: val_auc improved from 0.73814 to 0.73999, saving model to wei
ghts3it.best.hdf5
Epoch 12/20
61178/61178 [==============================] - 108s 2ms/step - loss: 0.403
4 - auc: 0.7118 - val_loss: 0.3952 - val_auc: 0.7391

Epoch 00012: val_auc did not improve from 0.73999
Epoch 13/20
61178/61178 [==============================] - 109s 2ms/step - loss: 0.403
2 - auc: 0.7099 - val_loss: 0.3929 - val_auc: 0.7414

Epoch 00013: val_auc improved from 0.73999 to 0.74139, saving model to wei
ghts3it.best.hdf5
Epoch 14/20
61178/61178 [==============================] - 108s 2ms/step - loss: 0.401
2 - auc: 0.7104 - val_loss: 0.3922 - val_auc: 0.7394

Epoch 00014: val_auc did not improve from 0.74139
Epoch 15/20
61178/61178 [==============================] - 113s 2ms/step - loss: 0.401
7 - auc: 0.7050 - val_loss: 0.3900 - val_auc: 0.7408

Epoch 00015: val_auc did not improve from 0.74139
Epoch 16/20
61178/61178 [==============================] - 114s 2ms/step - loss: 0.400
0 - auc: 0.7118 - val_loss: 0.3914 - val_auc: 0.7366

Epoch 00016: val_auc did not improve from 0.74139
Epoch 17/20
61178/61178 [==============================] - 108s 2ms/step - loss: 0.400
2 - auc: 0.7107 - val_loss: 0.3918 - val_auc: 0.7413

Epoch 00017: val_auc did not improve from 0.74139
Epoch 18/20
61178/61178 [==============================] - 108s 2ms/step - loss: 0.400
4 - auc: 0.7090 - val_loss: 0.3908 - val_auc: 0.7406

Epoch 00018: val_auc did not improve from 0.74139
Epoch 19/20
61178/61178 [==============================] - 109s 2ms/step - loss: 0.399
```

```
1 - auc: 0.7159 - val_loss: 0.3932 - val_auc: 0.7412

Epoch 00019: val_auc did not improve from 0.74139
Epoch 20/20
61178/61178 [==============================] - 108s 2ms/step - loss: 0.399
0 - auc: 0.7141 - val_loss: 0.3915 - val_auc: 0.7451

Epoch 00020: val_auc improved from 0.74139 to 0.74506, saving model to wei
ghts3it.best.hdf5
```

1.8 Plots on training results

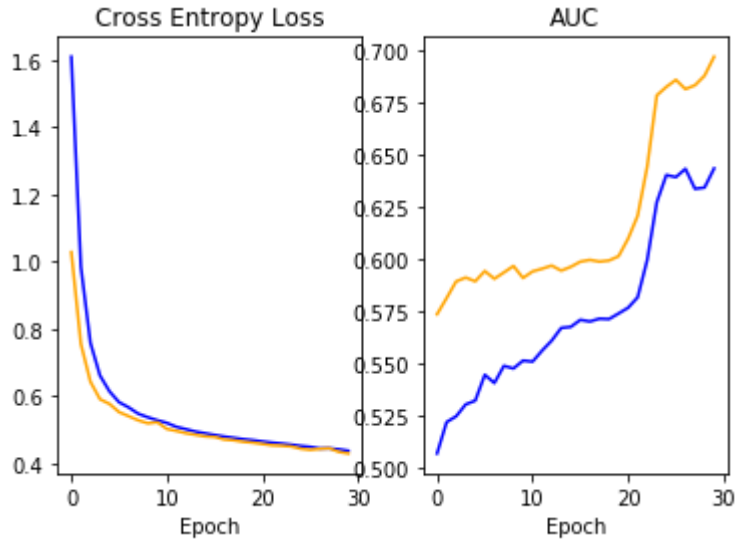
1.8.1 Loss & AUC plots for first 30 epochs

In [0]:

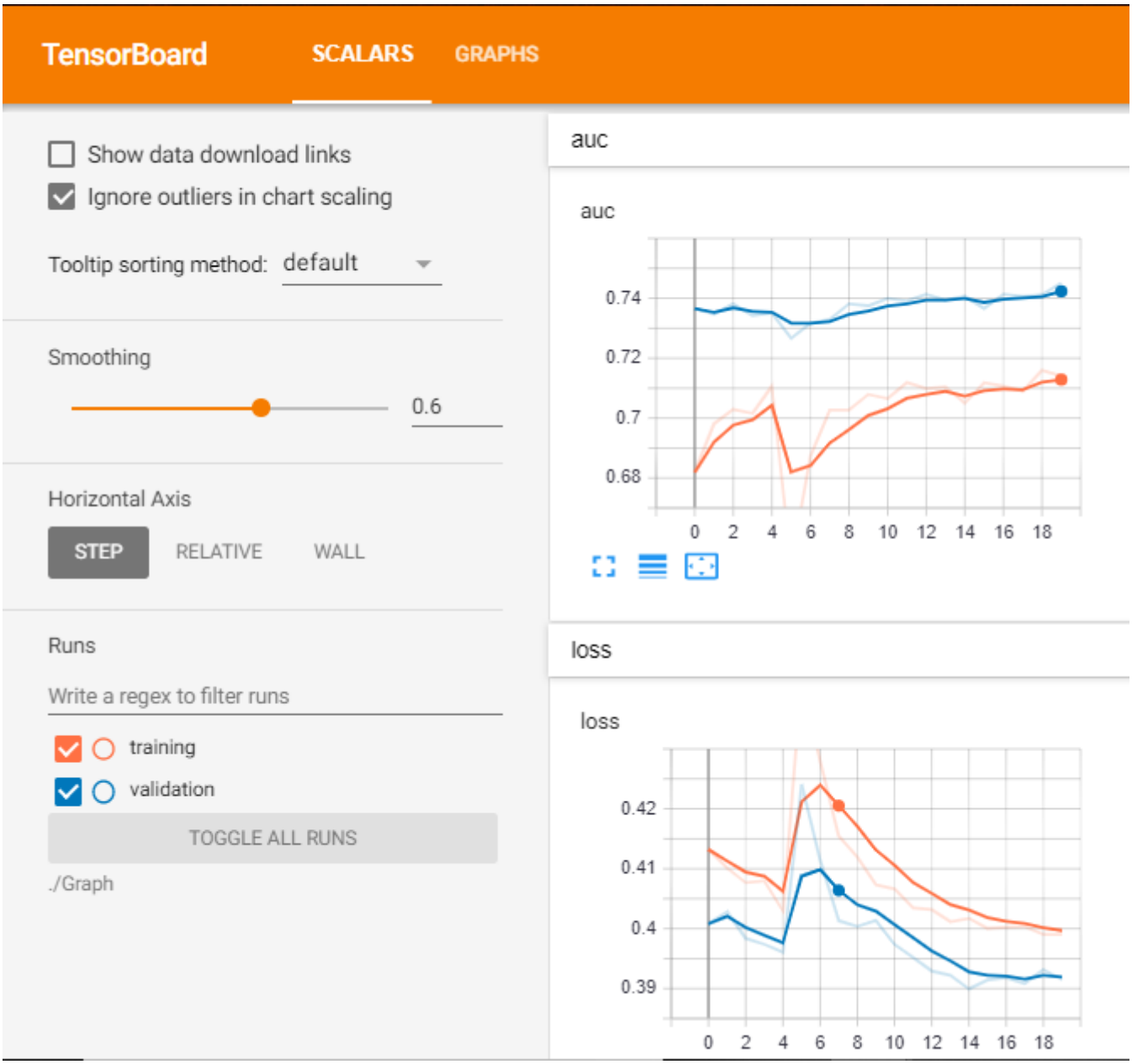
```
# function to plot epoch vs loss & epoch vs AUC
%matplotlib notebook
%matplotlib inline
from matplotlib import pyplot
def plot(history):
    # plot loss
    pyplot.subplot(121)
    pyplot.title('Cross Entropy Loss')
    pyplot.xlabel('Epoch')
    pyplot.plot(history.history['loss'], color='blue', label='train')
    pyplot.plot(history.history['val_loss'], color='orange', label='CV')
    # plot auc
    pyplot.subplot(122)
    pyplot.title('AUC')
    pyplot.xlabel('Epoch')
    pyplot.plot(history.history['auc'], color='blue', label='train')
    pyplot.plot(history.history['val_auc'], color='orange', label='CV')
```

In [0]:

```
plot(history)
```



1.8.2 Tensorboard image for the next 20 epochs



1.9 Results & Model Testing

In [25]:

```
train_results = model.evaluate([train_essay_padded,train_state_padded,train_grade_padde
d,train_cat_padded,
                                train_subcat_padded,train_prefix_padded,train_rem_inp],
y_train,
                               verbose=1,batch_size=600)
print('Train Loss: ',train_results[0])
print('Train AUC: ',train_results[1])
```

```
61178/61178 [==============================] - 31s 514us/step
Train Loss:  0.3868490646613763
Train AUC:  0.757658551906011
```

In [26]:

```
cv_results = model.evaluate([cv_essay_padded,cv_state_padded,cv_grade_padded,cv_cat_pad
ded,cv_subcat_padded,
                             cv_prefix_padded,cv_rem_inp],y_cv,verbose=1,batch_size=600
)
print('CV Loss: ',cv_results[0])
print('CV AUC: ',cv_results[1])
```

```
15295/15295 [==============================] - 8s 520us/step
CV Loss:  0.3915461766813346
CV AUC:  0.7450938297081866
```

In [22]:

```
test_results = model.evaluate([test_essay_padded,test_state_padded,test_grade_padded,te
st_cat_padded,
                               test_subcat_padded,test_prefix_padded,test_rem_inp],y_t
est,verbose=1,
                              batch_size=500)
print('Test Loss: ',test_results[0])
print('Test AUC: ',test_results[1])
```

```
32775/32775 [==============================] - 21s 626us/step
Test Loss:  0.3917855288944201
Test AUC:  0.7453525967037833
```

Model -2

Use the same model as above but for 'input_seq_total_text_data' give only some words in the sentance not all the words. Filter the words as below.

1. Train the TF-IDF on the Train data

2. Get the idf value for each word we have in the train data.

3. Remove the low idf value and high idf value words from our data. Do some analysis on the Idf values and based on those values choose the low and high threshold value. Because very frequent words and very very rare words don't give much information. (you can plot a box plots and take only the idf scores within IQR range and corresponding words)

4. Train the LSTM after removing the Low and High idf value words. (In model-1 Train on total data but in Model-2 train on data after removing some words based on IDF values)

2.1 Encoding text & other features

In [3]:

```
import warnings
warnings.filterwarnings("ignore")
from collections import defaultdict
import matplotlib.pyplot as plt
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Dropout, LSTM, BatchNormalization,concatenate,Flatten,Embeddin
g,Dense,Dropout,MaxPooling2D,Reshape
from keras.models import Sequential
from keras import Model,Input
from keras.layers.convolutional import Conv2D,Conv1D
import keras.backend as k
from sklearn.metrics import roc_auc_score
import tensorflow as tf
import keras
from keras.initializers import he_normal,glorot_normal
from keras.regularizers import l1,l2
from keras.callbacks import Callback, EarlyStopping, ModelCheckpoint,LearningRateSchedu
ler
from time import time
from tensorflow.python.keras.callbacks import TensorBoard
from IPython.display import SVG, display
from keras.preprocessing.text import one_hot
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers.embeddings import Embedding
```

```
Using TensorFlow backend.
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade (https://www.tensorflow.org/guide/migrate) now or ensure your notebook will continue to use TensorFlow 1.x via the `%tensorflow_version 1.x` magic:
more info (https://colab.research.google.com/notebooks/tensorflow_version.ipynb).

2.1.1 Applying TFIDF vectorizer on essay

In [0]:

```
vectorizer = TfidfVectorizer()
tfidf = vectorizer.fit_transform(X_train['preprocessed_essay'])
print(tfidf.shape)
```

```
(61178, 44877)
```

In [0]:

```
idf_values = vectorizer.idf_
```

In [0]:

```
type(idf_values)
```

Out[0]:

```
numpy.ndarray
```

In [0]:

```
print(idf_values)
```

```
[ 7.26796908  5.91676604 11.32841209 ... 11.32841209 10.92294698
 11.32841209]
```

2.1.2 Analysis of the words

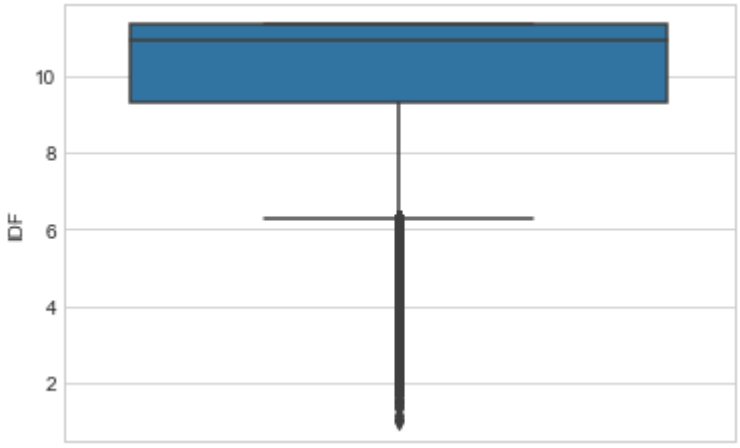
In [0]:

```
df_tfidf=pd.DataFrame(columns=['IDF'])
df_tfidf['IDF']=idf_values

sns.set_style("whitegrid")
sns.boxplot(y = 'IDF', data = df_tfidf)
```

Out[0]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2cf64546080>
```



- From the above box-plot, the 25th percentile is ~ 9.3, 50th percentile is ~ 11.0, 75th percentile is ~ 11.3
- Let us dive in deep into the idf values using percentiles

In [0]:

```
for i in range(0,101,5):
    print('{} percentile: {}'.format(i,np.percentile(idf_values,i)))
```

```
0 percentile: 1.007481454396812
5 percentile: 6.118925938959091
10 percentile: 7.367598922202934
15 percentile: 8.150358261452567
20 percentile: 8.802683447492257
25 percentile: 9.313509071258247
30 percentile: 9.71897417936641
35 percentile: 10.075649123305144
40 percentile: 10.412121359926356
45 percentile: 10.635264911240569
50 percentile: 10.922946983692347
55 percentile: 10.922946983692347
60 percentile: 10.922946983692347
65 percentile: 11.328412091800512
70 percentile: 11.328412091800512
75 percentile: 11.328412091800512
80 percentile: 11.328412091800512
85 percentile: 11.328412091800512
90 percentile: 11.328412091800512
95 percentile: 11.328412091800512
100 percentile: 11.328412091800512
```

In [0]:

```
for i in range(0,36,1):
    print('{} percentile: {}'.format(i,np.percentile(idf_values,i)))
```

```
0 percentile: 1.007481454396812
1 percentile: 4.0614172050061015
2 percentile: 4.880314150009966
3 percentile: 5.3777695392127844
4 percentile: 5.746797251118518
5 percentile: 6.118925938959091
6 percentile: 6.434310613960207
7 percentile: 6.714879667548734
8 percentile: 6.95896423933349
9 percentile: 7.16952900844084
10 percentile: 7.367598922202934
11 percentile: 7.555651153705873
12 percentile: 7.7174941791562865
13 percentile: 7.862676189000785
14 percentile: 8.014226087127986
15 percentile: 8.150358261452567
16 percentile: 8.28388965407709
17 percentile: 8.410641359716232
18 percentile: 8.55582336956073
19 percentile: 8.66970362309096
20 percentile: 8.802683447492257
21 percentile: 8.886065056431306
22 percentile: 9.025826998806465
23 percentile: 9.131187514464292
24 percentile: 9.18834592830424
25 percentile: 9.313509071258247
26 percentile: 9.382501942745199
27 percentile: 9.45660991489892
28 percentile: 9.536652622572456
29 percentile: 9.623663999562087
30 percentile: 9.71897417936641
31 percentile: 9.824334695024238
32 percentile: 9.824334695024238
33 percentile: 9.942117730680621
34 percentile: 9.942117730680621
35 percentile: 10.075649123305144
```

In [0]:

```
for i in np.arange(0,0.1,0.01):
    print('{} percentile: {}'.format(i,np.percentile(idf_values,i)))
```

```
0.0 percentile: 1.007481454396812
0.01 percentile: 1.4241428952977047
0.02 percentile: 1.7269058709895677
0.03 percentile: 1.854813213413192
0.04 percentile: 2.0051714749874994
0.05 percentile: 2.0462065800883633
0.06 percentile: 2.10594078681765
0.07 percentile: 2.1985668011223827
0.08 percentile: 2.3114348571610503
0.09 percentile: 2.348939254368712
```

- I shall consider words with idf values between 0.04th & 35th percentile as the words in IQR turned out to be very less

2.1.3 Extracting important words by excluding words having very low & very high idf values

In [0]:

```
features = vectorizer.get_feature_names()
sorted_idf = np.argsort(idf_values)
```

In [0]:

```
#getting the indices of words with idf's sorted in ascending order
index=sorted_idf
print(index)
```

```
[38474 26702 35061 ... 15428 32550 44876]
```

In [0]:

```
word_list=[]
for i in index:
    if idf_values[i]>=2.0051714749874994 and idf_values[i]<=10.075649123305144:
        word_list.append(features[i])
    else:
        continue;
```

In [0]:

```
print(len(word_list))
```

```
16312
```

- Therefore the number of distict words is 16312

2.1.4 Essay texts containing important words only

In [0]:

```
def update(data):
    corpus = []
    for text in tqdm(data):
        text = ' '.join(word for word in text.split() if word in word_list)
        corpus.append(text)
    return corpus
```

In [0]:

```
from datetime import datetime
start = datetime.now()
truncated_essay_train= update(X_train.preprocessed_essay)
np.save('truncated_essay_train', truncated_essay_train)
print('Time taken :', datetime.now() - start)
```

```
100%|████████████████████████████████████████████████████████████
████████████████████████████████████████████| 61178/61178 [14:24<
00:00, 51.11it/s]

Time taken : 0:14:47.959392
```

In [0]:

```
from datetime import datetime
start = datetime.now()
truncated_essay_cv= update(X_cv.preprocessed_essay)
np.save('truncated_essay_cv', truncated_essay_cv)
print('Time taken :', datetime.now() - start)
```

```
100%|████████████████████████████████████████████████████████████
████████████████████████████████████████████| 15295/15295 [03:18<
00:00, 77.06it/s]

Time taken : 0:03:20.002902
```

In [0]:

```
from datetime import datetime
start = datetime.now()
truncated_essay_test= update(X_test.preprocessed_essay)
np.save('truncated_essay_test', truncated_essay_test)
print('Time taken :', datetime.now() - start)
```

```
100%|████████████████████████████████████████████████████████████
████████████████████████████████████████████| 32775/32775 [06:55<
00:00, 78.91it/s]

Time taken : 0:06:59.000359
```

np.save('truncated_essay_train', truncated_essay_train) np.save('truncated_essay_cv', truncated_essay_cv) np.save('truncated_essay_test', truncated_essay_test)

In [0]:

```
truncated_essay_train= np.load('truncated_essay_train.npy')
truncated_essay_cv= np.load('truncated_essay_cv.npy')
truncated_essay_test= np.load('truncated_essay_test.npy')
```

In [0]:

```
len(truncated_essay_train)
```

Out[0]:

61178

In [0]:

```
#to get an estimate of review length of the truncatd essay columns
def review_length(data):
    count=[]
    c=[len(text.split()) for text in tqdm(data)]
    c=max(c)
    count.append(c)
    return max(count)
```

In [0]:

```
a=review_length(truncated_essay_train)
print(a)
```

```
100%|██████████████████████████████████████████████████████████████████████| 61178/61178 [00:00<00:00, 75015.58it/s]
275
```

In [0]:

```
df=pd.DataFrame(columns=['essay'])
df['essay']=truncated_essay_train
```

In [0]:

```
df.head(5)
```

Out[0]:

	essay
0	teach awesome group second graders motivated e...
1	areas district depends support community commu...
2	located fort worth lower middle neighborhood 5...
3	years taught 11th grade american literature en...
4	teach first grade title south carolina certain...

2.1.5 Finding vocabulary for each feature

In [0]:

```
def unique(column):
    counter= CountVectorizer(lowercase=False) #columns like teacher_prefix have upperca
se letters
    matrix= counter.fit_transform(column.values)
    return matrix
```

In [0]:

```
essay_count= unique(df['essay'])
state_count= unique(X_train['school_state'])
grade_count= unique(X_train['project_grade_category'])
subject_cat_count= unique(X_train['clean_categories'])
subject_subcat_count= unique(X_train['clean_subcategories'])
teacher_prefix_count= unique(X_train['teacher_prefix'])
print('Essay:',essay_count.shape)
print('State:',state_count.shape)
print('Grade:',grade_count.shape)
print('Category:',subject_cat_count.shape)
print('Subcategory:',subject_subcat_count.shape)
print('Teacher prefix:',teacher_prefix_count.shape)
```

```
Essay: (61178, 16312)
State: (61178, 51)
Grade: (61178, 4)
Category: (61178, 9)
Subcategory: (61178, 30)
Teacher prefix: (61178, 5)
```

2.1.6 Converting pandas numerical features column to a ndarray

In [0]:

```
train_rem_inp= X_train['Numerical_features'].values
cv_rem_inp= X_cv['Numerical_features'].values
test_rem_inp= X_test['Numerical_features'].values
```

In [0]:

```
def encoder(feature):
    t = Tokenizer()
    t.fit_on_texts(feature)
    vocab_size = len(t.word_index) + 1
    # integer encode the documents
    encoded_docs = t.texts_to_sequences(feature)
    return encoded_docs,vocab_size,t
```

In [0]:

```
def padding(encoded_docs,max_length):
    padded_docs = pad_sequences(encoded_docs, maxlen=max_length, padding='post')
    return padded_docs
```

2.1.7 Encoding & padding essay

In [0]:

```
#train data
docs,vocab,t1=encoder(truncated_essay_train)
print(vocab)
```

```
16313
```

In [0]:

```
train_essay_padded = padding(docs,500)
print(train_essay_padded.shape)
#print(train_essay_padded[0])
```

```
(61178, 500)
```

In [0]:

```
cv_essay_padded= np.load('/content/Corrected/cv_essay_padded.npy')
test_essay_padded= np.load('/content/Corrected/test_essay_padded.npy')
```

2.1.8 Encoding & padding school_state

In [0]:

```
#train data
docs,vocab,t=encoder(X_train.school_state)
print(vocab)
train_state_padded = padding(docs,1)
print(train_state_padded.shape)
print(train_state_padded[5])
```

```
52
(61178, 1)
[8]
```

In [0]:

```
#cv data
docs = t.texts_to_sequences(X_cv.school_state)
cv_state_padded = padding(docs,1)
print(cv_state_padded.shape)
print(cv_state_padded[5])
```

```
(15295, 1)
[41]
```

In [0]:

```
#test data
docs = t.texts_to_sequences(X_test.school_state)
test_state_padded = padding(docs,1)
print(test_state_padded.shape)
print(test_state_padded[5])
```

```
(32775, 1)
[19]
```

2.1.9 Encoding & padding project_grade_categories

In [0]:

```
#train data
docs,vocab,t=encoder(X_train.project_grade_category)
print(vocab)
train_grade_padded = padding(docs,1)
print(train_grade_padded.shape)
print(train_grade_padded[5])
```

```
10
(61178, 1)
[5]
```

In [0]:

```
#cv data
docs = t.texts_to_sequences(X_cv.project_grade_category)
cv_grade_padded = padding(docs,1)
print(cv_grade_padded.shape)
print(cv_grade_padded[5])
```

```
(15295, 1)
[7]
```

In [0]:

```
#test data
docs = t.texts_to_sequences(X_test.project_grade_category)
test_grade_padded = padding(docs,1)
print(test_grade_padded.shape)
print(test_grade_padded[5])
```

```
(32775, 1)
[5]
```

2.1.10 Encoding & padding clean_categories

In [0]:

```
#train data
docs,vocab,t=encoder(X_train.clean_categories)
print(vocab)
train_cat_padded = padding(docs,10)
print(train_cat_padded.shape)
print(train_cat_padded[5])
```

```
16
(61178, 10)
[3 4 0 0 0 0 0 0 0 0]
```

In [0]:

```
#cv data
docs = t.texts_to_sequences(X_cv.clean_categories)
cv_cat_padded = padding(docs,10)
print(cv_cat_padded.shape)
print(cv_cat_padded[5])
```

```
(15295, 10)
[7 0 0 0 0 0 0 0 0 0]
```

In [0]:

```
#test data
docs = t.texts_to_sequences(X_test.clean_categories)
test_cat_padded = padding(docs,10)
print(test_cat_padded.shape)
print(test_cat_padded[5])
```

```
(32775, 10)
[5 6 0 0 0 0 0 0 0 0]
```

2.1.11 Encoding & padding clean_subcategories

In [0]:

```
#train data
docs,vocab,t=encoder(X_train.clean_subcategories)
print(vocab)
train_subcat_padded = padding(docs,10)
print(train_subcat_padded.shape)
print(train_subcat_padded[5])
```

```
38
(61178, 10)
[7 0 0 0 0 0 0 0 0 0]
```

In [0]:

```
#cv data
docs = t.texts_to_sequences(X_cv.clean_subcategories)
cv_subcat_padded = padding(docs,10)
print(cv_subcat_padded.shape)
print(cv_subcat_padded[5])
```

```
(15295, 10)
[6 0 0 0 0 0 0 0 0 0]
```

In [0]:

```
#test data
docs = t.texts_to_sequences(X_test.clean_subcategories)
test_subcat_padded = padding(docs,10)
print(test_subcat_padded.shape)
print(test_subcat_padded[5])
```

```
(32775, 10)
[11 12  0  0  0  0  0  0  0  0]
```

2.1.12 Encoding & padding teacher_prefix

In [0]:

```
#train data
docs,vocab,t=encoder(X_train.teacher_prefix)
print(vocab)
train_prefix_padded = padding(docs,1)
print(train_prefix_padded.shape)
print(train_prefix_padded[5])
```

```
6
(61178, 1)
[1]
```

In [0]:

```
#cv data
docs = t.texts_to_sequences(X_cv.teacher_prefix)
cv_prefix_padded = padding(docs,1)
print(cv_prefix_padded.shape)
print(cv_prefix_padded[5])
```

```
(15295, 1)
[2]
```

In [0]:

```
#test data
docs = t.texts_to_sequences(X_test.teacher_prefix)
test_prefix_padded = padding(docs,1)
print(test_prefix_padded.shape)
print(test_prefix_padded[5])
```

```
(32775, 1)
[2]
```

2.2 Converting class labels to vectors using one-hot encoding

In [0]:

```
from keras.utils import to_categorical
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
y_cv = to_categorical(y_cv)
```

2.3 Saving all tensors for further use

In [0]:

```
#https://www.geeksforgeeks.org/numpy-save/
np.save('train_essay_padded', train_essay_padded)
np.save('cv_essay_padded', cv_essay_padded)
np.save('test_essay_padded', test_essay_padded)

np.save('train_state_padded', train_state_padded)
np.save('cv_state_padded', cv_state_padded)
np.save('test_state_padded', test_state_padded)

np.save('train_grade_padded', train_grade_padded)
np.save('cv_grade_padded', cv_grade_padded)
np.save('test_grade_padded', test_grade_padded)

np.save('train_cat_padded', train_cat_padded)
np.save('cv_cat_padded', cv_cat_padded)
np.save('test_cat_padded', test_cat_padded)

np.save('train_subcat_padded', train_subcat_padded)
np.save('cv_subcat_padded', cv_subcat_padded)
np.save('test_subcat_padded', test_subcat_padded)

np.save('train_prefix_padded', train_prefix_padded)
np.save('cv_prefix__padded', cv_prefix_padded)
np.save('test_prefix_padded', test_prefix_padded)

np.save('train_rem_inp', train_rem_inp)
np.save('cv_rem_inp', cv_rem_inp)
np.save('test_rem_inp', test_rem_inp)

np.save('y_train', y_train)
np.save('y_test', y_test)
np.save('y_cv', y_cv)
```

In [0]:

```
#Loading the tensors
train_essay_padded= np.load('/content/Data/train_essay_padded.npy')
#cv_essay_padded= np.load('/content/Data/cv_essay_padded.npy')
#test_essay_padded= np.load('/content/Data/test_essay_padded.npy')

train_state_padded= np.load('/content/Data/train_state_padded.npy')
cv_state_padded= np.load('/content/Data/cv_state_padded.npy')
test_state_padded= np.load('/content/Data/test_state_padded.npy')

train_grade_padded= np.load('/content/Data/train_grade_padded.npy')
cv_grade_padded= np.load('/content/Data/cv_grade_padded.npy')
test_grade_padded= np.load('/content/Data/test_grade_padded.npy')

train_cat_padded= np.load('/content/Data/train_cat_padded.npy')
cv_cat_padded= np.load('/content/Data/cv_cat_padded.npy')
test_cat_padded= np.load('/content/Data/test_cat_padded.npy')

train_subcat_padded= np.load('/content/Data/train_subcat_padded.npy')
cv_subcat_padded= np.load('/content/Data/cv_subcat_padded.npy')
test_subcat_padded= np.load('/content/Data/test_subcat_padded.npy')

train_prefix_padded= np.load('/content/Data/train_prefix_padded.npy')
cv_prefix_padded= np.load('/content/Data/cv_prefix__padded.npy')
test_prefix_padded= np.load('/content/Data/test_prefix_padded.npy')

train_rem_inp= np.load('/content/Data/train_rem_inp.npy')
cv_rem_inp= np.load('/content/Data/cv_rem_inp.npy')
test_rem_inp= np.load('/content/Data/test_rem_inp.npy')

y_train= np.load('/content/Data/y_train.npy')
y_test= np.load('/content/Data/y_test.npy')
y_cv= np.load('/content/Data/y_cv.npy')
```

2.4 Loading the pre-trained glove model

In [0]:

```
with open('glove_vectors', 'rb') as f:
    glove = pickle.load(f)
print ("Done.",len(glove)," words loaded!")
```

```
Done. 51510  words loaded!
```

In [0]:

```
type(glove)
```

Out[0]:

```
dict
```

2.5 Defining the performance metric[ROC]

In [0]:

```
def auc( y_true, y_pred ) :
    score = tf.py_func( lambda y_true, y_pred : roc_auc_score( y_true, y_pred, average=
'macro', sample_weight=None).astype('float32'),
                        [y_true, y_pred],'float32',stateful=True,name='sklearnAUC')
    return score
```

2.6 Creating the 2D Embedding matrix using Glove vectors

In [0]:

```
#Credits: https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-ke
ras/

embedding_matrix = np.zeros((16313, 300))
for word, i in t1.word_index.items():
    embedding_vector = glove.get(word)
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector
```

In [0]:

```
np.save('embedding_matrix',embedding_matrix)
```

In [0]:

```
embedding_matrix= np.load('/content/Data/embedding_matrix.npy')
```

In [5]:

```
import warnings
warnings.filterwarnings("ignore")
from collections import defaultdict
import matplotlib.pyplot as plt
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Dropout, LSTM, BatchNormalization,concatenate,Flatten,Embeddin
g,Dense,Dropout,MaxPooling2D,Reshape
from keras.models import Sequential
from keras import Model,Input
from keras.layers.convolutional import Conv2D,Conv1D
import keras.backend as k
from sklearn.metrics import roc_auc_score
import tensorflow as tf
import keras
from keras.initializers import he_normal,glorot_normal
from keras.regularizers import l1,l2
from keras.callbacks import Callback, EarlyStopping, ModelCheckpoint,LearningRateSchedu
ler
from time import time
from tensorflow.python.keras.callbacks import TensorBoard
from IPython.display import SVG, display
from keras.layers import LeakyReLU
```

```
Using TensorFlow backend.
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade (https://www.tensorflow.org/guide/migrate) now or ensure your notebook will continue to use TensorFlow 1.x via the `%tensorflow_version 1.x` magic:
more info (https://colab.research.google.com/notebooks/tensorflow_version.ipynb).

2.7 LSTM model

2.7.1 Architecture

In [0]:

```
import keras.backend as K
K.clear_session()
```

In [0]:

```
#essay
input1 = Input(shape=(500,))
i1 = Embedding(input_dim=16313,output_dim= 300,input_length=500,weights=[embedding_matr
ix],trainable=False)(input1)
i1 = Dropout(0.5)(i1)
i1 = LSTM(128,kernel_initializer='he_normal',recurrent_dropout=0.5,kernel_regularizer=l
2(0.001),return_sequences=True)(i1)
i1= LeakyReLU(alpha = 0.5)(i1)
f1 = Flatten()(i1)

#school_state
input2 = Input(shape=(1,))
i2 = Embedding(input_dim= 52,output_dim= 2,input_length=1)(input2)
f2 = Flatten()(i2)

#project grade category
input3 = Input(shape=(1,))
i3 = Embedding(input_dim= 10,output_dim= 2,input_length=1)(input3)
f3 = Flatten()(i3)

#clean_categories
input4 = Input(shape=(10,))
i4 = Embedding(input_dim=16,output_dim= 2,input_length=10)(input4)
f4 = Flatten()(i4)

#clean_subcategories
input5 = Input(shape=(10,))
i5 = Embedding(input_dim= 38,output_dim= 2,input_length=10)(input5)
f5 = Flatten()(i5)

#teacher_prefix
input6 = Input(shape=(1,))
i6 = Embedding(input_dim= 6,output_dim= 2,input_length=1)(input6)
f6 = Flatten()(i6)

#concatenated numerical features
input7 = Input(shape=(1,))
i7 = Dense(16,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.
001))(input7)

#concatenating all the inputs
concat = concatenate([f1,f2,f3,f4,f5,f6,i7])


l = Dense(128,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.
001))(concat)
l = Dropout(0.5)(l)
l = Dense(64,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.0
01))(l)
l = Dropout(0.5)(l)
l = BatchNormalization()(l)
l = Dense(32,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.0
01))(l)
l = Dropout(0.5)(l)
output = Dense(2, activation = 'softmax')(l)
```

In [34]:

```
# create model with seven inputs
model2 = Model(inputs=[input1,input2,input3,input4,input5,input6,input7], outputs=[outp
ut])
model2.summary()
```

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 500)          0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 500, 300)     4893900     input_1[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 500, 300)     0           embedding_1[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 500, 128)     219648      dropout_1[0][0]
__________________________________________________________________________________________________
input_2 (InputLayer)            (None, 1)            0
__________________________________________________________________________________________________
input_3 (InputLayer)            (None, 1)            0
__________________________________________________________________________________________________
input_4 (InputLayer)            (None, 10)           0
__________________________________________________________________________________________________
input_5 (InputLayer)            (None, 10)           0
__________________________________________________________________________________________________
input_6 (InputLayer)            (None, 1)            0
__________________________________________________________________________________________________
leaky_re_lu_1 (LeakyReLU)       (None, 500, 128)     0           lstm_1[0][0]
__________________________________________________________________________________________________
embedding_2 (Embedding)         (None, 1, 2)         104         input_2[0][0]
__________________________________________________________________________________________________
embedding_3 (Embedding)         (None, 1, 2)         20          input_3[0][0]
__________________________________________________________________________________________________
embedding_4 (Embedding)         (None, 10, 2)        32          input_4[0][0]
__________________________________________________________________________________________________
embedding_5 (Embedding)         (None, 10, 2)        76          input_5[0][0]
__________________________________________________________________________________________________
embedding_6 (Embedding)         (None, 1, 2)         12          input_6[0][0]
__________________________________________________________________________________________________
```

```
input_7 (InputLayer)            (None, 1)            0
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 64000)        0           leaky_r
e_lu_1[0][0]
__________________________________________________________________________________________________
flatten_2 (Flatten)             (None, 2)            0           embeddi
ng_2[0][0]
__________________________________________________________________________________________________
flatten_3 (Flatten)             (None, 2)            0           embeddi
ng_3[0][0]
__________________________________________________________________________________________________
flatten_4 (Flatten)             (None, 20)           0           embeddi
ng_4[0][0]
__________________________________________________________________________________________________
flatten_5 (Flatten)             (None, 20)           0           embeddi
ng_5[0][0]
__________________________________________________________________________________________________
flatten_6 (Flatten)             (None, 2)            0           embeddi
ng_6[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 16)           32          input_7
[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 64062)        0           flatten
_1[0][0]
                                                                 flatten
_2[0][0]
                                                                 flatten
_3[0][0]
                                                                 flatten
_4[0][0]
                                                                 flatten
_5[0][0]
                                                                 flatten
_6[0][0]
                                                                 dense_1
[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 128)          8200064     concate
nate_1[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 128)          0           dense_2
[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 64)           8256        dropout
_2[0][0]
__________________________________________________________________________________________________
dropout_3 (Dropout)             (None, 64)           0           dense_3
[0][0]
```

```
__________________________________________________________________________________
_____________________________
batch_normalization_1 (BatchNor (None, 64)           256         dropout
_3[0][0]
__________________________________________________________________________________
_____________________________
dense_4 (Dense)                 (None, 32)           2080        batch_n
ormalization_1[0][0]
__________________________________________________________________________________
_____________________________
dropout_4 (Dropout)             (None, 32)           0           dense_4
[0][0]
__________________________________________________________________________________
_____________________________
dense_5 (Dense)                 (None, 2)            66          dropout
_4[0][0]
==================================================================================
==========================
Total params: 13,324,546
Trainable params: 8,430,518
Non-trainable params: 4,894,028
__________________________________________________________________________________
_____________________________
```

In [0]:

```
model2.load_weights("/content/weights1it.best.hdf5")
```

In [0]:

```
model2.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.Adam(lr=0.00
1,decay = 1e-4),metrics=[auc])
```

2.7.2 Checkpointing the model and creating the callback list

In [37]:

```
from keras.callbacks import ModelCheckpoint
from keras.callbacks import CSVLogger
import matplotlib.pyplot as plt
from tensorflow.python.keras.callbacks import TensorBoard
from keras.callbacks import TensorBoard
import tensorflow as tf
import datetime
import keras
from tensorboardcolab import *
from keras.callbacks import ReduceLROnPlateau

#https://github.com/taomanwai/tensorboardcolab/
tbc=TensorBoardColab()
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2,
                              patience=1, min_lr=0.002,verbose = 1)

filepath="weights2it.best.hdf5"
checkpoints = ModelCheckpoint(filepath, monitor='val_auc', verbose=1, save_best_only=True, mode='max')
train_results = CSVLogger('train_results_2.log') #storing the training results in a pandas dataframe
callbacks_list = [checkpoints, TensorBoardColabCallback(tbc), train_results]
```

```
Wait for 8 seconds...
TensorBoard link:
https://de51267c.ngrok.io
```

2.7.3 Fitting the model in batches

In [0]:

```
# finding the class weights before fitting the model
from sklearn.utils import compute_class_weight
class_wts = compute_class_weight("balanced", classes= np.unique(y),y=y)
print(class_wts)
```

```
[3.30214001 0.58921753]
```

In [0]:

```
np.save('class_wts', class_wts)
```

In [0]:

```
class_wts= np.load('/content/Data/class_wts.npy')
```

In [17]:

```
history=model2.fit([train_essay_padded,train_state_padded,train_grade_padded,train_cat_
padded,train_subcat_padded,
           train_prefix_padded,train_rem_inp], y_train, nb_epoch=100,verbose=1,batch_si
ze=500,
          validation_data=([cv_essay_padded,cv_state_padded,cv_grade_padded,cv_cat_padd
ed,cv_subcat_padded,cv_prefix_padded,cv_rem_inp],y_cv),
          callbacks =callbacks_list,class_weight = class_wts)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_
core/python/ops/math_grad.py:1424: where (from tensorflow.python.ops.array
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Pleas
e use tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:1020: The name tf.assign is deprecated. Please us
e tf.compat.v1.assign instead.

Train on 61178 samples, validate on 15295 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorboard
colab/core.py:49: The name tf.summary.FileWriter is deprecated. Please use
tf.compat.v1.summary.FileWriter instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callb
acks.py:1122: The name tf.summary.merge_all is deprecated. Please use tf.c
ompat.v1.summary.merge_all instead.

Epoch 1/100
61178/61178 [==============================] - 156s 3ms/step - loss: 1.257
6 - auc: 0.5137 - val_loss: 0.7015 - val_auc: 0.5677

Epoch 00001: val_auc improved from -inf to 0.56769, saving model to weight
s1it.best.hdf5
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorboard
colab/callbacks.py:51: The name tf.Summary is deprecated. Please use tf.co
mpat.v1.Summary instead.

Epoch 2/100
61178/61178 [==============================] - 153s 3ms/step - loss: 0.670
2 - auc: 0.5220 - val_loss: 0.5711 - val_auc: 0.5769

Epoch 00002: val_auc improved from 0.56769 to 0.57691, saving model to wei
ghts1it.best.hdf5
Epoch 3/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.575
2 - auc: 0.5328 - val_loss: 0.5447 - val_auc: 0.5735

Epoch 00003: val_auc did not improve from 0.57691
Epoch 4/100
61178/61178 [==============================] - 153s 3ms/step - loss: 0.547
5 - auc: 0.5320 - val_loss: 0.5210 - val_auc: 0.5741

Epoch 00004: val_auc did not improve from 0.57691
Epoch 5/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.521
4 - auc: 0.5371 - val_loss: 0.5107 - val_auc: 0.5741

Epoch 00005: val_auc did not improve from 0.57691
Epoch 6/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.503
9 - auc: 0.5493 - val_loss: 0.4853 - val_auc: 0.5745

Epoch 00006: val_auc did not improve from 0.57691
Epoch 7/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.487
9 - auc: 0.5482 - val_loss: 0.4753 - val_auc: 0.5755
```

```
Epoch 00007: val_auc did not improve from 0.57691
Epoch 8/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.475
8 - auc: 0.5546 - val_loss: 0.4666 - val_auc: 0.5767

Epoch 00008: val_auc did not improve from 0.57691
Epoch 9/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.468
8 - auc: 0.5548 - val_loss: 0.4618 - val_auc: 0.5765

Epoch 00009: val_auc did not improve from 0.57691
Epoch 10/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.463
5 - auc: 0.5588 - val_loss: 0.4545 - val_auc: 0.5763

Epoch 00010: val_auc did not improve from 0.57691
Epoch 11/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.456
6 - auc: 0.5561 - val_loss: 0.4527 - val_auc: 0.5746

Epoch 00011: val_auc did not improve from 0.57691
Epoch 12/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.450
9 - auc: 0.5582 - val_loss: 0.4450 - val_auc: 0.5766

Epoch 00012: val_auc did not improve from 0.57691
Epoch 13/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.445
2 - auc: 0.5653 - val_loss: 0.4420 - val_auc: 0.5772

Epoch 00013: val_auc improved from 0.57691 to 0.57724, saving model to wei
ghts1it.best.hdf5
Epoch 14/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.443
1 - auc: 0.5612 - val_loss: 0.4384 - val_auc: 0.5786

Epoch 00014: val_auc improved from 0.57724 to 0.57857, saving model to wei
ghts1it.best.hdf5
Epoch 15/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.439
6 - auc: 0.5735 - val_loss: 0.4366 - val_auc: 0.5825

Epoch 00015: val_auc improved from 0.57857 to 0.58248, saving model to wei
ghts1it.best.hdf5
Epoch 16/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.437
8 - auc: 0.5689 - val_loss: 0.4343 - val_auc: 0.5833

Epoch 00016: val_auc improved from 0.58248 to 0.58329, saving model to wei
ghts1it.best.hdf5
Epoch 17/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.434
4 - auc: 0.5781 - val_loss: 0.4328 - val_auc: 0.5892

Epoch 00017: val_auc improved from 0.58329 to 0.58920, saving model to wei
ghts1it.best.hdf5
Epoch 18/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.437
5 - auc: 0.5648 - val_loss: 0.4456 - val_auc: 0.5732
```

```
Epoch 00018: val_auc did not improve from 0.58920
Epoch 19/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.437
4 - auc: 0.5590 - val_loss: 0.4356 - val_auc: 0.5746

Epoch 00019: val_auc did not improve from 0.58920
Epoch 20/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.433
4 - auc: 0.5556 - val_loss: 0.4301 - val_auc: 0.5748

Epoch 00020: val_auc did not improve from 0.58920
Epoch 21/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.430
2 - auc: 0.5692 - val_loss: 0.4297 - val_auc: 0.5743

Epoch 00021: val_auc did not improve from 0.58920
Epoch 22/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.430
3 - auc: 0.5618 - val_loss: 0.4285 - val_auc: 0.5744

Epoch 00022: val_auc did not improve from 0.58920
Epoch 23/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.428
5 - auc: 0.5656 - val_loss: 0.4291 - val_auc: 0.5749

Epoch 00023: val_auc did not improve from 0.58920
Epoch 24/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.427
8 - auc: 0.5687 - val_loss: 0.4308 - val_auc: 0.5758

Epoch 00024: val_auc did not improve from 0.58920
Epoch 25/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.466
5 - auc: 0.5697 - val_loss: 0.4726 - val_auc: 0.5782

Epoch 00025: val_auc did not improve from 0.58920
Epoch 26/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.458
4 - auc: 0.5760 - val_loss: 0.4503 - val_auc: 0.5787

Epoch 00026: val_auc did not improve from 0.58920
Epoch 27/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.449
5 - auc: 0.5704 - val_loss: 0.4504 - val_auc: 0.5794

Epoch 00027: val_auc did not improve from 0.58920
Epoch 28/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.442
4 - auc: 0.5695 - val_loss: 0.4398 - val_auc: 0.5756

Epoch 00028: val_auc did not improve from 0.58920
Epoch 29/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.436
4 - auc: 0.5644 - val_loss: 0.4353 - val_auc: 0.5748

Epoch 00029: val_auc did not improve from 0.58920
Epoch 30/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.433
2 - auc: 0.5667 - val_loss: 0.4320 - val_auc: 0.5749

Epoch 00030: val_auc did not improve from 0.58920
```

```
Epoch 31/100
61178/61178 [==============================] - 150s 2ms/step - loss: 0.430
5 - auc: 0.5693 - val_loss: 0.4296 - val_auc: 0.5760

Epoch 00031: val_auc did not improve from 0.58920
Epoch 32/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.432
2 - auc: 0.5707 - val_loss: 0.4327 - val_auc: 0.5747

Epoch 00032: val_auc did not improve from 0.58920
Epoch 33/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.430
9 - auc: 0.5716 - val_loss: 0.4293 - val_auc: 0.5745

Epoch 00033: val_auc did not improve from 0.58920
Epoch 34/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.430
3 - auc: 0.5685 - val_loss: 0.4283 - val_auc: 0.5760

Epoch 00034: val_auc did not improve from 0.58920
Epoch 35/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.428
8 - auc: 0.5683 - val_loss: 0.4283 - val_auc: 0.5745

Epoch 00035: val_auc did not improve from 0.58920
Epoch 36/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.427
5 - auc: 0.5727 - val_loss: 0.4270 - val_auc: 0.5747

Epoch 00036: val_auc did not improve from 0.58920
Epoch 37/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.428
5 - auc: 0.5707 - val_loss: 0.4341 - val_auc: 0.5749

Epoch 00037: val_auc did not improve from 0.58920
Epoch 38/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.430
9 - auc: 0.5718 - val_loss: 0.4284 - val_auc: 0.5746

Epoch 00038: val_auc did not improve from 0.58920
Epoch 39/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.428
1 - auc: 0.5706 - val_loss: 0.4270 - val_auc: 0.5741

Epoch 00039: val_auc did not improve from 0.58920
Epoch 40/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.426
5 - auc: 0.5708 - val_loss: 0.4257 - val_auc: 0.5742

Epoch 00040: val_auc did not improve from 0.58920
Epoch 41/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.426
4 - auc: 0.5717 - val_loss: 0.4255 - val_auc: 0.5737

Epoch 00041: val_auc did not improve from 0.58920
Epoch 42/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.425
8 - auc: 0.5702 - val_loss: 0.4247 - val_auc: 0.5739

Epoch 00042: val_auc did not improve from 0.58920
Epoch 43/100
```

```
61178/61178 [==============================] - 152s 2ms/step - loss: 0.427
3 - auc: 0.5716 - val_loss: 0.4279 - val_auc: 0.5740

Epoch 00043: val_auc did not improve from 0.58920
Epoch 44/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.426
3 - auc: 0.5728 - val_loss: 0.4263 - val_auc: 0.5734

Epoch 00044: val_auc did not improve from 0.58920
Epoch 45/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.425
2 - auc: 0.5720 - val_loss: 0.4247 - val_auc: 0.5737

Epoch 00045: val_auc did not improve from 0.58920
Epoch 46/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.425
4 - auc: 0.5727 - val_loss: 0.4265 - val_auc: 0.5741

Epoch 00046: val_auc did not improve from 0.58920
Epoch 47/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.426
5 - auc: 0.5677 - val_loss: 0.4251 - val_auc: 0.5738

Epoch 00047: val_auc did not improve from 0.58920
Epoch 48/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.425
2 - auc: 0.5703 - val_loss: 0.4255 - val_auc: 0.5742

Epoch 00048: val_auc did not improve from 0.58920
Epoch 49/100
61178/61178 [==============================] - 152s 2ms/step - loss: 0.425
5 - auc: 0.5729 - val_loss: 0.4246 - val_auc: 0.5749

Epoch 00049: val_auc did not improve from 0.58920
Epoch 50/100
61178/61178 [==============================] - 153s 3ms/step - loss: 0.429
6 - auc: 0.5735 - val_loss: 0.4359 - val_auc: 0.5747

Epoch 00050: val_auc did not improve from 0.58920
Epoch 51/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.451
0 - auc: 0.5674 - val_loss: 0.4644 - val_auc: 0.5744

Epoch 00051: val_auc did not improve from 0.58920
Epoch 52/100
61178/61178 [==============================] - 153s 3ms/step - loss: 0.457
9 - auc: 0.5715 - val_loss: 0.4494 - val_auc: 0.5748

Epoch 00052: val_auc did not improve from 0.58920
Epoch 53/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.446
2 - auc: 0.5729 - val_loss: 0.4455 - val_auc: 0.5749

Epoch 00053: val_auc did not improve from 0.58920
Epoch 54/100
61178/61178 [==============================] - 153s 3ms/step - loss: 0.456
9 - auc: 0.5725 - val_loss: 0.4643 - val_auc: 0.5738

Epoch 00054: val_auc did not improve from 0.58920
Epoch 55/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.475
```

```
3 - auc: 0.5704 - val_loss: 0.4810 - val_auc: 0.5744

Epoch 00055: val_auc did not improve from 0.58920
Epoch 56/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.503
8 - auc: 0.5716 - val_loss: 0.5205 - val_auc: 0.5760

Epoch 00056: val_auc did not improve from 0.58920
Epoch 57/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.518
8 - auc: 0.5718 - val_loss: 0.5162 - val_auc: 0.5759

Epoch 00057: val_auc did not improve from 0.58920
Epoch 58/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.515
4 - auc: 0.5703 - val_loss: 0.5140 - val_auc: 0.5792

Epoch 00058: val_auc did not improve from 0.58920
Epoch 59/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.512
8 - auc: 0.5715 - val_loss: 0.5126 - val_auc: 0.5766

Epoch 00059: val_auc did not improve from 0.58920
Epoch 60/100
61178/61178 [==============================] - 153s 3ms/step - loss: 0.510
8 - auc: 0.5694 - val_loss: 0.5074 - val_auc: 0.5759

Epoch 00060: val_auc did not improve from 0.58920
Epoch 61/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.508
0 - auc: 0.5737 - val_loss: 0.5063 - val_auc: 0.5778

Epoch 00061: val_auc did not improve from 0.58920
Epoch 62/100
61178/61178 [==============================] - 153s 3ms/step - loss: 0.505
5 - auc: 0.5722 - val_loss: 0.5040 - val_auc: 0.5784

Epoch 00062: val_auc did not improve from 0.58920
Epoch 63/100
61178/61178 [==============================] - 153s 2ms/step - loss: 0.503
2 - auc: 0.5727 - val_loss: 0.5013 - val_auc: 0.5788

Epoch 00063: val_auc did not improve from 0.58920
Epoch 64/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.501
0 - auc: 0.5725 - val_loss: 0.5008 - val_auc: 0.5763

Epoch 00064: val_auc did not improve from 0.58920
Epoch 65/100
61178/61178 [==============================] - 153s 3ms/step - loss: 0.498
9 - auc: 0.5727 - val_loss: 0.4976 - val_auc: 0.5773

Epoch 00065: val_auc did not improve from 0.58920
Epoch 66/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.497
2 - auc: 0.5759 - val_loss: 0.4959 - val_auc: 0.5769

Epoch 00066: val_auc did not improve from 0.58920
Epoch 67/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.495
1 - auc: 0.5726 - val_loss: 0.4955 - val_auc: 0.5758
```

```
Epoch 00067: val_auc did not improve from 0.58920
Epoch 68/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.493
8 - auc: 0.5722 - val_loss: 0.4945 - val_auc: 0.5751

Epoch 00068: val_auc did not improve from 0.58920
Epoch 69/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.491
9 - auc: 0.5736 - val_loss: 0.4913 - val_auc: 0.5754

Epoch 00069: val_auc did not improve from 0.58920
Epoch 70/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.490
2 - auc: 0.5747 - val_loss: 0.4889 - val_auc: 0.5745

Epoch 00070: val_auc did not improve from 0.58920
Epoch 71/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.488
4 - auc: 0.5719 - val_loss: 0.4874 - val_auc: 0.5741

Epoch 00071: val_auc did not improve from 0.58920
Epoch 72/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.486
3 - auc: 0.5753 - val_loss: 0.4860 - val_auc: 0.5749

Epoch 00072: val_auc did not improve from 0.58920
Epoch 73/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.485
4 - auc: 0.5745 - val_loss: 0.4855 - val_auc: 0.5746

Epoch 00073: val_auc did not improve from 0.58920
Epoch 74/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.483
4 - auc: 0.5768 - val_loss: 0.4827 - val_auc: 0.5743

Epoch 00074: val_auc did not improve from 0.58920
Epoch 75/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.481
8 - auc: 0.5731 - val_loss: 0.4807 - val_auc: 0.5748

Epoch 00075: val_auc did not improve from 0.58920
Epoch 76/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.481
2 - auc: 0.5756 - val_loss: 0.4827 - val_auc: 0.5749

Epoch 00076: val_auc did not improve from 0.58920
Epoch 77/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.480
5 - auc: 0.5755 - val_loss: 0.4811 - val_auc: 0.5743

Epoch 00077: val_auc did not improve from 0.58920
Epoch 78/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.478
6 - auc: 0.5761 - val_loss: 0.4806 - val_auc: 0.5743

Epoch 00078: val_auc did not improve from 0.58920
Epoch 79/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.478
2 - auc: 0.5704 - val_loss: 0.4771 - val_auc: 0.5744
```

```
Epoch 00079: val_auc did not improve from 0.58920
Epoch 80/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.476
4 - auc: 0.5706 - val_loss: 0.4763 - val_auc: 0.5756

Epoch 00080: val_auc did not improve from 0.58920
Epoch 81/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.474
4 - auc: 0.5749 - val_loss: 0.4738 - val_auc: 0.5746

Epoch 00081: val_auc did not improve from 0.58920
Epoch 82/100
61178/61178 [==============================] - 155s 3ms/step - loss: 0.474
2 - auc: 0.5720 - val_loss: 0.4737 - val_auc: 0.5735

Epoch 00082: val_auc did not improve from 0.58920
Epoch 83/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.472
1 - auc: 0.5738 - val_loss: 0.4713 - val_auc: 0.5772

Epoch 00083: val_auc did not improve from 0.58920
Epoch 84/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.470
1 - auc: 0.5745 - val_loss: 0.4700 - val_auc: 0.5752

Epoch 00084: val_auc did not improve from 0.58920
Epoch 85/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.469
1 - auc: 0.5741 - val_loss: 0.4680 - val_auc: 0.5768

Epoch 00085: val_auc did not improve from 0.58920
Epoch 86/100
61178/61178 [==============================] - 155s 3ms/step - loss: 0.467
6 - auc: 0.5750 - val_loss: 0.4666 - val_auc: 0.5764

Epoch 00086: val_auc did not improve from 0.58920
Epoch 87/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.466
7 - auc: 0.5768 - val_loss: 0.4657 - val_auc: 0.5769

Epoch 00087: val_auc did not improve from 0.58920
Epoch 88/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.465
6 - auc: 0.5730 - val_loss: 0.4640 - val_auc: 0.5799

Epoch 00088: val_auc did not improve from 0.58920
Epoch 89/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.464
5 - auc: 0.5748 - val_loss: 0.4658 - val_auc: 0.5806

Epoch 00089: val_auc did not improve from 0.58920
Epoch 90/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.465
1 - auc: 0.5749 - val_loss: 0.4653 - val_auc: 0.5751

Epoch 00090: val_auc did not improve from 0.58920
Epoch 91/100
61178/61178 [==============================] - 153s 3ms/step - loss: 0.463
5 - auc: 0.5702 - val_loss: 0.4621 - val_auc: 0.5752

Epoch 00091: val_auc did not improve from 0.58920
```

```
Epoch 92/100
61178/61178 [==============================] - 159s 3ms/step - loss: 0.480
5 - auc: 0.5757 - val_loss: 0.4846 - val_auc: 0.5780

Epoch 00092: val_auc did not improve from 0.58920
Epoch 93/100
61178/61178 [==============================] - 158s 3ms/step - loss: 0.484
5 - auc: 0.5753 - val_loss: 0.4848 - val_auc: 0.5763

Epoch 00093: val_auc did not improve from 0.58920
Epoch 94/100
61178/61178 [==============================] - 158s 3ms/step - loss: 0.483
2 - auc: 0.5760 - val_loss: 0.4821 - val_auc: 0.5775

Epoch 00094: val_auc did not improve from 0.58920
Epoch 95/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.482
0 - auc: 0.5766 - val_loss: 0.4806 - val_auc: 0.5768

Epoch 00095: val_auc did not improve from 0.58920
Epoch 96/100
61178/61178 [==============================] - 157s 3ms/step - loss: 0.480
4 - auc: 0.5760 - val_loss: 0.4804 - val_auc: 0.5760

Epoch 00096: val_auc did not improve from 0.58920
Epoch 97/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.479
5 - auc: 0.5746 - val_loss: 0.4787 - val_auc: 0.5758

Epoch 00097: val_auc did not improve from 0.58920
Epoch 98/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.478
6 - auc: 0.5759 - val_loss: 0.4770 - val_auc: 0.5764

Epoch 00098: val_auc did not improve from 0.58920
Epoch 99/100
61178/61178 [==============================] - 154s 3ms/step - loss: 0.477
4 - auc: 0.5749 - val_loss: 0.4765 - val_auc: 0.5766

Epoch 00099: val_auc did not improve from 0.58920
Epoch 100/100
61178/61178 [==============================] - 155s 3ms/step - loss: 0.476
3 - auc: 0.5746 - val_loss: 0.4776 - val_auc: 0.5762

Epoch 00100: val_auc did not improve from 0.58920
```

In [38]:

```
history=model2.fit([train_essay_padded,train_state_padded,train_grade_padded,train_cat_
padded,train_subcat_padded,
           train_prefix_padded,train_rem_inp], y_train, nb_epoch=80,verbose=1,batch_siz
e=500,
          validation_data=([cv_essay_padded,cv_state_padded,cv_grade_padded,cv_cat_padd
ed,cv_subcat_padded,cv_prefix_padded,cv_rem_inp],y_cv),
          callbacks =callbacks_list,class_weight = class_wts)
```

```
Train on 61178 samples, validate on 15295 samples
Epoch 1/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.434
4 - auc: 0.5850 - val_loss: 0.4303 - val_auc: 0.6190

Epoch 00001: val_auc improved from -inf to 0.61899, saving model to weight
s2it.best.hdf5
Epoch 2/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.430
7 - auc: 0.6420 - val_loss: 0.4331 - val_auc: 0.6892

Epoch 00002: val_auc improved from 0.61899 to 0.68916, saving model to wei
ghts2it.best.hdf5
Epoch 3/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.428
6 - auc: 0.6586 - val_loss: 0.4285 - val_auc: 0.6972

Epoch 00003: val_auc improved from 0.68916 to 0.69716, saving model to wei
ghts2it.best.hdf5
Epoch 4/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.424
9 - auc: 0.6751 - val_loss: 0.4173 - val_auc: 0.7057

Epoch 00004: val_auc improved from 0.69716 to 0.70571, saving model to wei
ghts2it.best.hdf5
Epoch 5/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.417
2 - auc: 0.6871 - val_loss: 0.4092 - val_auc: 0.7172

Epoch 00005: val_auc improved from 0.70571 to 0.71717, saving model to wei
ghts2it.best.hdf5
Epoch 6/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.411
9 - auc: 0.6952 - val_loss: 0.4090 - val_auc: 0.7195

Epoch 00006: val_auc improved from 0.71717 to 0.71947, saving model to wei
ghts2it.best.hdf5
Epoch 7/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.410
8 - auc: 0.6970 - val_loss: 0.4232 - val_auc: 0.7162

Epoch 00007: val_auc did not improve from 0.71947
Epoch 8/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.409
8 - auc: 0.7006 - val_loss: 0.3995 - val_auc: 0.7260

Epoch 00008: val_auc improved from 0.71947 to 0.72603, saving model to wei
ghts2it.best.hdf5
Epoch 9/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.407
5 - auc: 0.7038 - val_loss: 0.4038 - val_auc: 0.7267

Epoch 00009: val_auc improved from 0.72603 to 0.72666, saving model to wei
ghts2it.best.hdf5
Epoch 10/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.405
0 - auc: 0.7045 - val_loss: 0.3986 - val_auc: 0.7272

Epoch 00010: val_auc improved from 0.72666 to 0.72724, saving model to wei
ghts2it.best.hdf5
Epoch 11/80
```

```
61178/61178 [==============================] - 157s 3ms/step - loss: 0.404
3 - auc: 0.7065 - val_loss: 0.3962 - val_auc: 0.7300

Epoch 00011: val_auc improved from 0.72724 to 0.73005, saving model to wei
ghts2it.best.hdf5
Epoch 12/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.403
2 - auc: 0.7110 - val_loss: 0.3943 - val_auc: 0.7293

Epoch 00012: val_auc did not improve from 0.73005
Epoch 13/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.402
2 - auc: 0.7105 - val_loss: 0.3947 - val_auc: 0.7344

Epoch 00013: val_auc improved from 0.73005 to 0.73445, saving model to wei
ghts2it.best.hdf5
Epoch 14/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.400
5 - auc: 0.7155 - val_loss: 0.3941 - val_auc: 0.7346

Epoch 00014: val_auc improved from 0.73445 to 0.73464, saving model to wei
ghts2it.best.hdf5
Epoch 15/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.400
6 - auc: 0.7155 - val_loss: 0.3959 - val_auc: 0.7329

Epoch 00015: val_auc did not improve from 0.73464
Epoch 16/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.401
8 - auc: 0.7139 - val_loss: 0.3933 - val_auc: 0.7350

Epoch 00016: val_auc improved from 0.73464 to 0.73502, saving model to wei
ghts2it.best.hdf5
Epoch 17/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.401
2 - auc: 0.7159 - val_loss: 0.3916 - val_auc: 0.7354

Epoch 00017: val_auc improved from 0.73502 to 0.73544, saving model to wei
ghts2it.best.hdf5
Epoch 18/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.400
0 - auc: 0.7182 - val_loss: 0.3942 - val_auc: 0.7356

Epoch 00018: val_auc improved from 0.73544 to 0.73561, saving model to wei
ghts2it.best.hdf5
Epoch 19/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.397
5 - auc: 0.7239 - val_loss: 0.3910 - val_auc: 0.7378

Epoch 00019: val_auc improved from 0.73561 to 0.73782, saving model to wei
ghts2it.best.hdf5
Epoch 20/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.398
2 - auc: 0.7185 - val_loss: 0.3932 - val_auc: 0.7372

Epoch 00020: val_auc did not improve from 0.73782
Epoch 21/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.399
9 - auc: 0.7218 - val_loss: 0.3935 - val_auc: 0.7373

Epoch 00021: val_auc did not improve from 0.73782
```

```
Epoch 22/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.399
7 - auc: 0.7168 - val_loss: 0.3901 - val_auc: 0.7376

Epoch 00022: val_auc did not improve from 0.73782
Epoch 23/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.397
8 - auc: 0.7221 - val_loss: 0.3913 - val_auc: 0.7393

Epoch 00023: val_auc improved from 0.73782 to 0.73934, saving model to wei
ghts2it.best.hdf5
Epoch 24/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.397
9 - auc: 0.7220 - val_loss: 0.3911 - val_auc: 0.7405

Epoch 00024: val_auc improved from 0.73934 to 0.74048, saving model to wei
ghts2it.best.hdf5
Epoch 25/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.396
7 - auc: 0.7249 - val_loss: 0.3911 - val_auc: 0.7398

Epoch 00025: val_auc did not improve from 0.74048
Epoch 26/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.395
7 - auc: 0.7270 - val_loss: 0.4031 - val_auc: 0.7395

Epoch 00026: val_auc did not improve from 0.74048
Epoch 27/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.397
9 - auc: 0.7215 - val_loss: 0.3924 - val_auc: 0.7412

Epoch 00027: val_auc improved from 0.74048 to 0.74124, saving model to wei
ghts2it.best.hdf5
Epoch 28/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.396
6 - auc: 0.7239 - val_loss: 0.3925 - val_auc: 0.7408

Epoch 00028: val_auc did not improve from 0.74124
Epoch 29/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.395
1 - auc: 0.7260 - val_loss: 0.3898 - val_auc: 0.7401

Epoch 00029: val_auc did not improve from 0.74124
Epoch 30/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.394
9 - auc: 0.7265 - val_loss: 0.3994 - val_auc: 0.7431

Epoch 00030: val_auc improved from 0.74124 to 0.74310, saving model to wei
ghts2it.best.hdf5
Epoch 31/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.395
2 - auc: 0.7290 - val_loss: 0.3888 - val_auc: 0.7415

Epoch 00031: val_auc did not improve from 0.74310
Epoch 32/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.394
6 - auc: 0.7287 - val_loss: 0.3904 - val_auc: 0.7412

Epoch 00032: val_auc did not improve from 0.74310
Epoch 33/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.392
```

```
7 - auc: 0.7324 - val_loss: 0.3910 - val_auc: 0.7394

Epoch 00033: val_auc did not improve from 0.74310
Epoch 34/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.394
5 - auc: 0.7270 - val_loss: 0.3933 - val_auc: 0.7436

Epoch 00034: val_auc improved from 0.74310 to 0.74357, saving model to wei
ghts2it.best.hdf5
Epoch 35/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.394
0 - auc: 0.7310 - val_loss: 0.3936 - val_auc: 0.7432

Epoch 00035: val_auc did not improve from 0.74357
Epoch 36/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.395
7 - auc: 0.7260 - val_loss: 0.3875 - val_auc: 0.7436

Epoch 00036: val_auc improved from 0.74357 to 0.74363, saving model to wei
ghts2it.best.hdf5
Epoch 37/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.394
6 - auc: 0.7278 - val_loss: 0.3868 - val_auc: 0.7437

Epoch 00037: val_auc improved from 0.74363 to 0.74371, saving model to wei
ghts2it.best.hdf5
Epoch 38/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.392
0 - auc: 0.7325 - val_loss: 0.3884 - val_auc: 0.7441

Epoch 00038: val_auc improved from 0.74371 to 0.74409, saving model to wei
ghts2it.best.hdf5
Epoch 39/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.393
0 - auc: 0.7308 - val_loss: 0.3871 - val_auc: 0.7445

Epoch 00039: val_auc improved from 0.74409 to 0.74445, saving model to wei
ghts2it.best.hdf5
Epoch 40/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.392
1 - auc: 0.7330 - val_loss: 0.3879 - val_auc: 0.7425

Epoch 00040: val_auc did not improve from 0.74445
Epoch 41/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.393
3 - auc: 0.7307 - val_loss: 0.3863 - val_auc: 0.7453

Epoch 00041: val_auc improved from 0.74445 to 0.74532, saving model to wei
ghts2it.best.hdf5
Epoch 42/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.392
4 - auc: 0.7342 - val_loss: 0.4025 - val_auc: 0.7422

Epoch 00042: val_auc did not improve from 0.74532
Epoch 43/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.393
7 - auc: 0.7295 - val_loss: 0.3858 - val_auc: 0.7447

Epoch 00043: val_auc did not improve from 0.74532
Epoch 44/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.391
```

```
4 - auc: 0.7309 - val_loss: 0.3861 - val_auc: 0.7465

Epoch 00044: val_auc improved from 0.74532 to 0.74654, saving model to wei
ghts2it.best.hdf5
Epoch 45/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.390
9 - auc: 0.7371 - val_loss: 0.3868 - val_auc: 0.7453

Epoch 00045: val_auc did not improve from 0.74654
Epoch 46/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.389
9 - auc: 0.7367 - val_loss: 0.3872 - val_auc: 0.7472

Epoch 00046: val_auc improved from 0.74654 to 0.74724, saving model to wei
ghts2it.best.hdf5
Epoch 47/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.390
9 - auc: 0.7357 - val_loss: 0.3914 - val_auc: 0.7442

Epoch 00047: val_auc did not improve from 0.74724
Epoch 48/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.391
8 - auc: 0.7334 - val_loss: 0.3883 - val_auc: 0.7459

Epoch 00048: val_auc did not improve from 0.74724
Epoch 49/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.392
1 - auc: 0.7334 - val_loss: 0.3865 - val_auc: 0.7452

Epoch 00049: val_auc did not improve from 0.74724
Epoch 50/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.389
6 - auc: 0.7361 - val_loss: 0.3844 - val_auc: 0.7468

Epoch 00050: val_auc did not improve from 0.74724
Epoch 51/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.389
3 - auc: 0.7372 - val_loss: 0.3871 - val_auc: 0.7463

Epoch 00051: val_auc did not improve from 0.74724
Epoch 52/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.389
6 - auc: 0.7372 - val_loss: 0.3860 - val_auc: 0.7471

Epoch 00052: val_auc did not improve from 0.74724
Epoch 53/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.390
6 - auc: 0.7348 - val_loss: 0.3869 - val_auc: 0.7464

Epoch 00053: val_auc did not improve from 0.74724
Epoch 54/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.390
2 - auc: 0.7370 - val_loss: 0.3869 - val_auc: 0.7482

Epoch 00054: val_auc improved from 0.74724 to 0.74822, saving model to wei
ghts2it.best.hdf5
Epoch 55/80
61178/61178 [==============================] - 157s 3ms/step - loss: 0.388
1 - auc: 0.7382 - val_loss: 0.3859 - val_auc: 0.7466

Epoch 00055: val_auc did not improve from 0.74822
```

```
Epoch 56/80
61178/61178 [==============================] - 156s 3ms/step - loss: 0.389
3 - auc: 0.7359 - val_loss: 0.3852 - val_auc: 0.7474

Epoch 00056: val_auc did not improve from 0.74822
Epoch 57/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.390
2 - auc: 0.7358 - val_loss: 0.3867 - val_auc: 0.7483

Epoch 00057: val_auc improved from 0.74822 to 0.74828, saving model to wei
ghts2it.best.hdf5
Epoch 58/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.387
9 - auc: 0.7407 - val_loss: 0.3853 - val_auc: 0.7466

Epoch 00058: val_auc did not improve from 0.74828
Epoch 59/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.388
5 - auc: 0.7398 - val_loss: 0.3850 - val_auc: 0.7468

Epoch 00059: val_auc did not improve from 0.74828
Epoch 60/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.389
8 - auc: 0.7371 - val_loss: 0.3834 - val_auc: 0.7484

Epoch 00060: val_auc improved from 0.74828 to 0.74845, saving model to wei
ghts2it.best.hdf5
Epoch 61/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.388
6 - auc: 0.7406 - val_loss: 0.3859 - val_auc: 0.7482

Epoch 00061: val_auc did not improve from 0.74845
Epoch 62/80
61178/61178 [==============================] - 153s 2ms/step - loss: 0.388
8 - auc: 0.7405 - val_loss: 0.3891 - val_auc: 0.7465

Epoch 00062: val_auc did not improve from 0.74845
Epoch 63/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.389
8 - auc: 0.7394 - val_loss: 0.3850 - val_auc: 0.7479

Epoch 00063: val_auc did not improve from 0.74845
Epoch 64/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.388
5 - auc: 0.7391 - val_loss: 0.3854 - val_auc: 0.7471

Epoch 00064: val_auc did not improve from 0.74845
Epoch 65/80
61178/61178 [==============================] - 154s 3ms/step - loss: 0.388
4 - auc: 0.7376 - val_loss: 0.3897 - val_auc: 0.7475

Epoch 00065: val_auc did not improve from 0.74845
Epoch 66/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.388
3 - auc: 0.7388 - val_loss: 0.3858 - val_auc: 0.7460

Epoch 00066: val_auc did not improve from 0.74845
Epoch 67/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.388
6 - auc: 0.7393 - val_loss: 0.3857 - val_auc: 0.7480
```

```
Epoch 00067: val_auc did not improve from 0.74845
Epoch 68/80
61178/61178 [==============================] - 153s 2ms/step - loss: 0.388
1 - auc: 0.7415 - val_loss: 0.3864 - val_auc: 0.7494

Epoch 00068: val_auc improved from 0.74845 to 0.74943, saving model to wei
ghts2it.best.hdf5
Epoch 69/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.388
8 - auc: 0.7392 - val_loss: 0.3847 - val_auc: 0.7468

Epoch 00069: val_auc did not improve from 0.74943
Epoch 70/80
61178/61178 [==============================] - 153s 2ms/step - loss: 0.389
5 - auc: 0.7379 - val_loss: 0.3862 - val_auc: 0.7477

Epoch 00070: val_auc did not improve from 0.74943
Epoch 71/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.387
8 - auc: 0.7408 - val_loss: 0.3850 - val_auc: 0.7478

Epoch 00071: val_auc did not improve from 0.74943
Epoch 72/80
61178/61178 [==============================] - 153s 3ms/step - loss: 0.387
8 - auc: 0.7408 - val_loss: 0.3840 - val_auc: 0.7469

Epoch 00072: val_auc did not improve from 0.74943
Epoch 73/80
61178/61178 [==============================] - 153s 2ms/step - loss: 0.386
3 - auc: 0.7443 - val_loss: 0.3846 - val_auc: 0.7476

Epoch 00073: val_auc did not improve from 0.74943
Epoch 74/80
61178/61178 [==============================] - 153s 2ms/step - loss: 0.387
6 - auc: 0.7401 - val_loss: 0.3838 - val_auc: 0.7501

Epoch 00074: val_auc improved from 0.74943 to 0.75014, saving model to wei
ghts2it.best.hdf5
Epoch 75/80
61178/61178 [==============================] - 153s 2ms/step - loss: 0.387
5 - auc: 0.7421 - val_loss: 0.3849 - val_auc: 0.7487

Epoch 00075: val_auc did not improve from 0.75014
Epoch 76/80
61178/61178 [==============================] - 152s 2ms/step - loss: 0.388
6 - auc: 0.7375 - val_loss: 0.3850 - val_auc: 0.7482

Epoch 00076: val_auc did not improve from 0.75014
Epoch 77/80
61178/61178 [==============================] - 153s 2ms/step - loss: 0.388
2 - auc: 0.7393 - val_loss: 0.3851 - val_auc: 0.7468

Epoch 00077: val_auc did not improve from 0.75014
Epoch 78/80
61178/61178 [==============================] - 153s 2ms/step - loss: 0.388
4 - auc: 0.7399 - val_loss: 0.3871 - val_auc: 0.7481

Epoch 00078: val_auc did not improve from 0.75014
Epoch 79/80
61178/61178 [==============================] - 152s 2ms/step - loss: 0.387
0 - auc: 0.7418 - val_loss: 0.3826 - val_auc: 0.7501
```

```
Epoch 00079: val_auc did not improve from 0.75014
Epoch 80/80
61178/61178 [==============================] - 152s 2ms/step - loss: 0.385
8 - auc: 0.7445 - val_loss: 0.3835 - val_auc: 0.7490

Epoch 00080: val_auc did not improve from 0.75014
```

2.8 Plots on training results

2.8.1 Loss & AUC plots

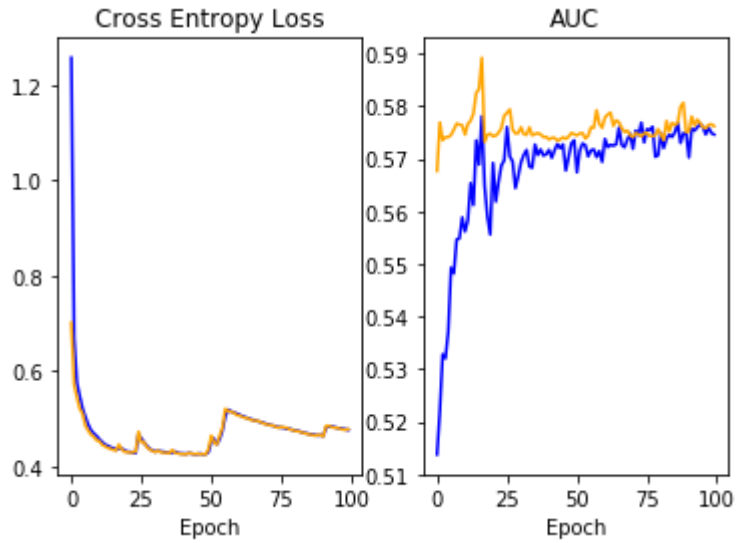
In [0]:

```
# function to plot epoch vs loss & epoch vs AUC
%matplotlib notebook
%matplotlib inline
from matplotlib import pyplot
def plot(history):
    # plot loss
    pyplot.subplot(121)
    pyplot.title('Cross Entropy Loss')
    pyplot.xlabel('Epoch')
    pyplot.plot(history.history['loss'], color='blue', label='train')
    pyplot.plot(history.history['val_loss'], color='orange', label='CV')
    # plot auc
    pyplot.subplot(122)
    pyplot.title('AUC')
    pyplot.xlabel('Epoch')
    pyplot.plot(history.history['auc'], color='blue', label='train')
    pyplot.plot(history.history['val_auc'], color='orange', label='CV')
```

Loss & AUC plots for the first 100 epochs

In [19]:

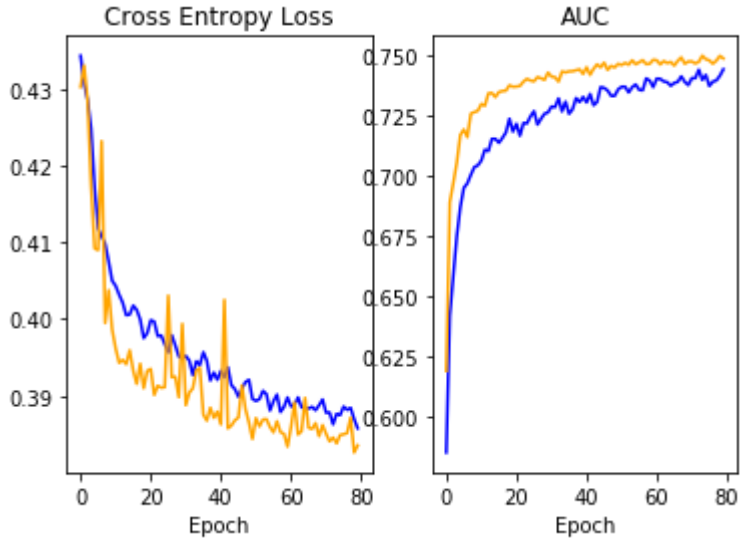
```
plot(history)
```



Loss & AUC plots for the next 80 epochs

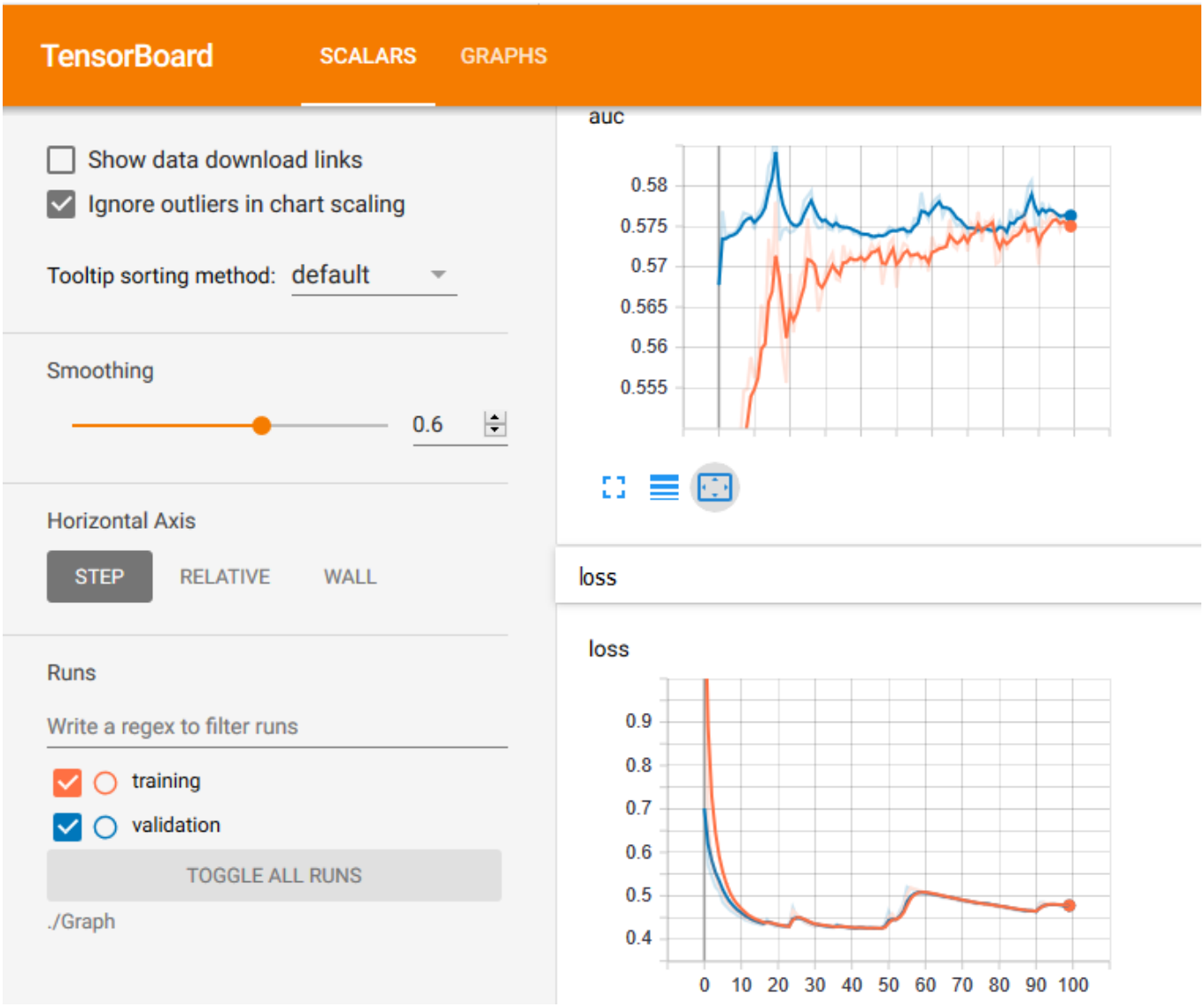
In [40]:

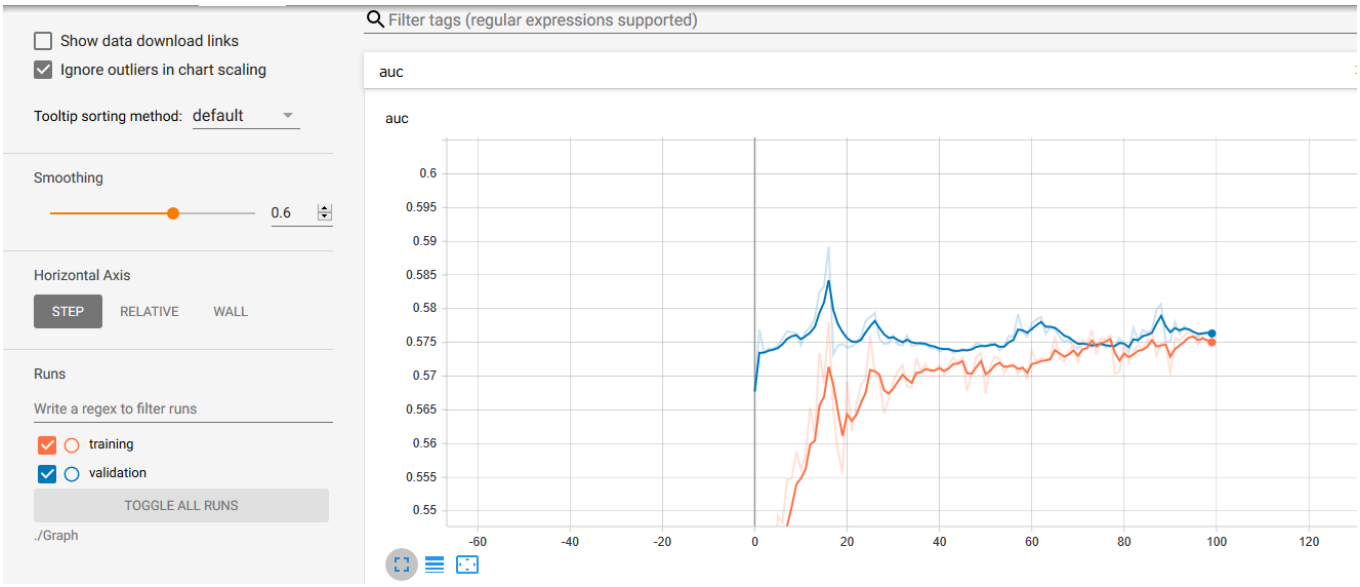
```
plot(history)
```

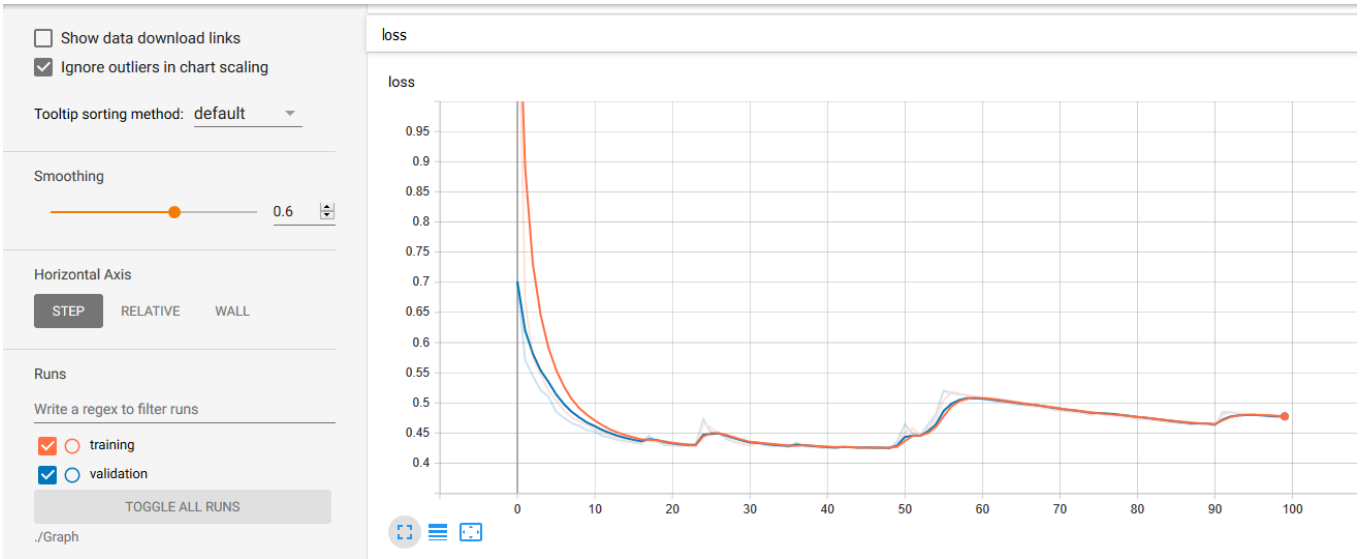


2.8.2 Tensorboard images

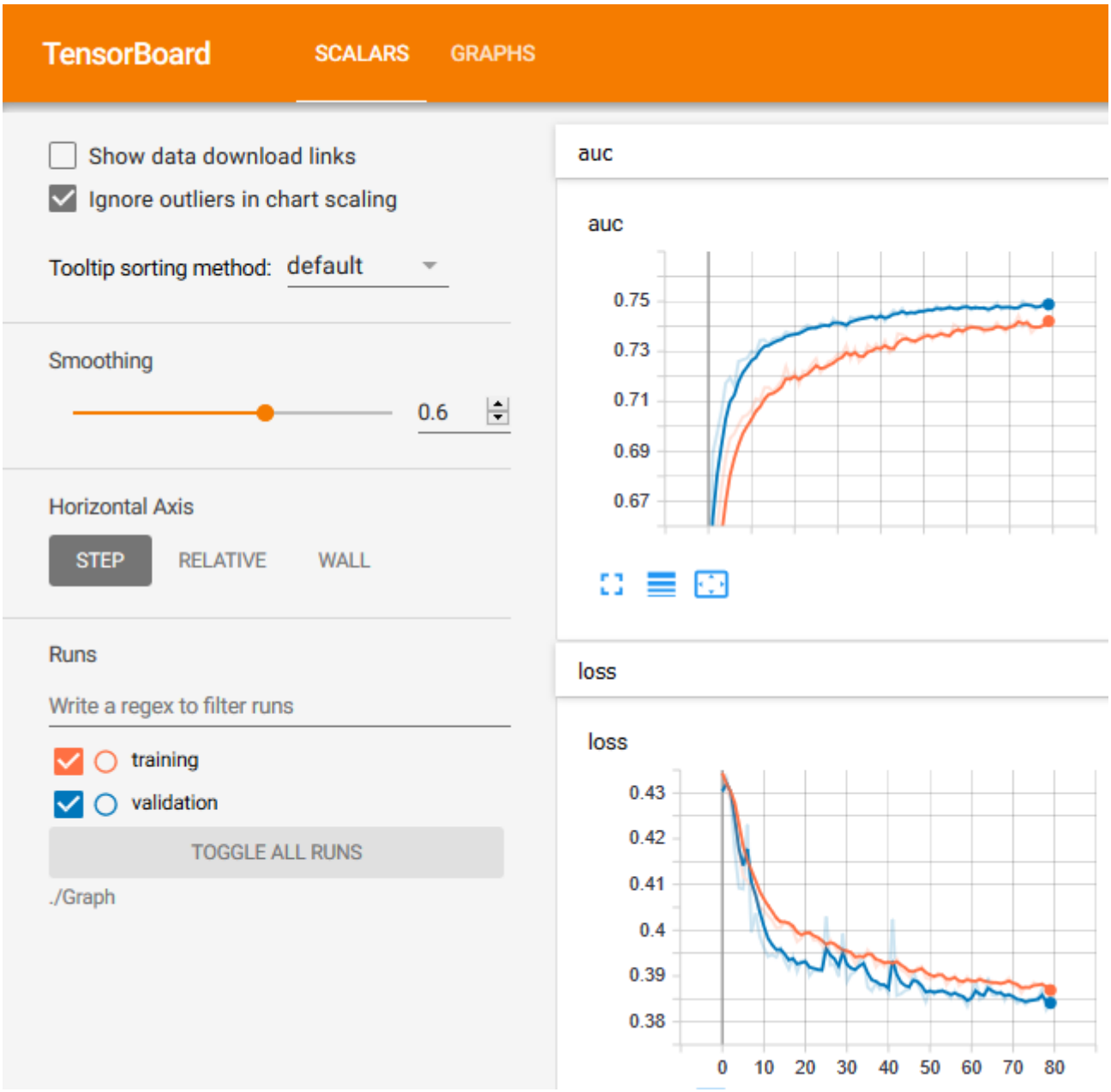
First 100 epochs

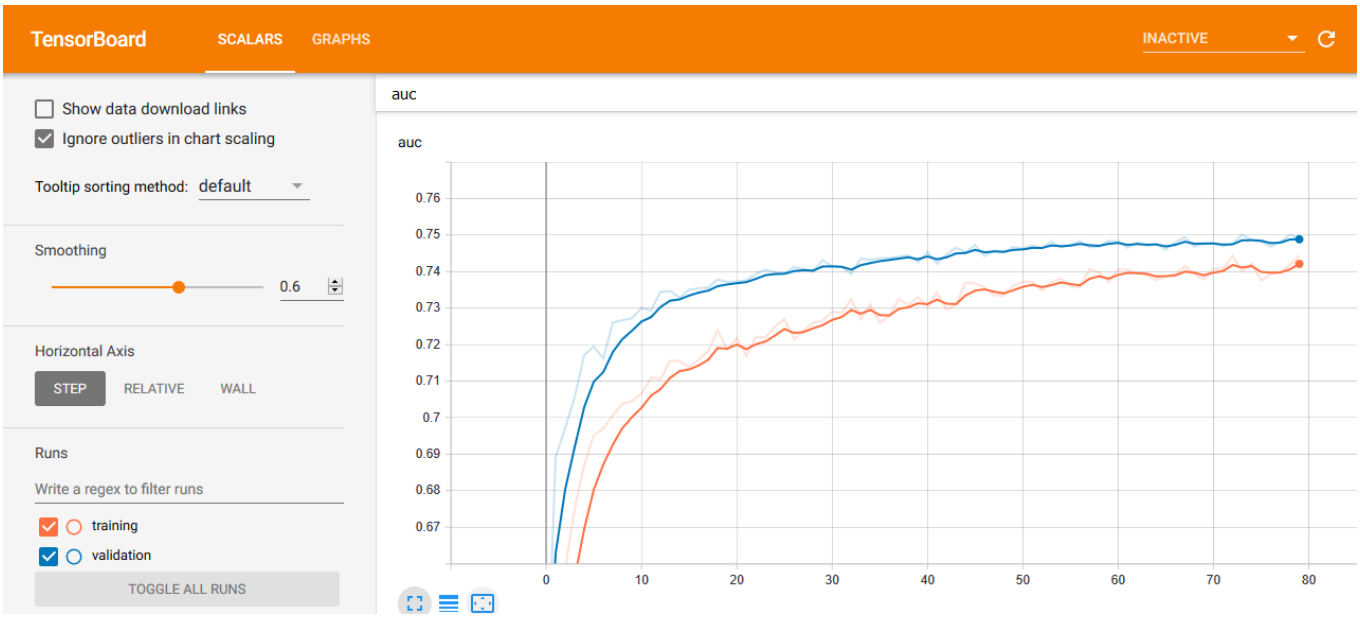

TensorBoard
SCALARS
GRAPHS
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
training
validation
TOGGLE ALL RUNS
./Graph
auc
0.58
0.575
0.57
0.565
0.56
0.555
loss
loss
0.9
0.8
0.7
0.6
0.5
0.4
0
10
20
30
40
50
60
70
80
90
100

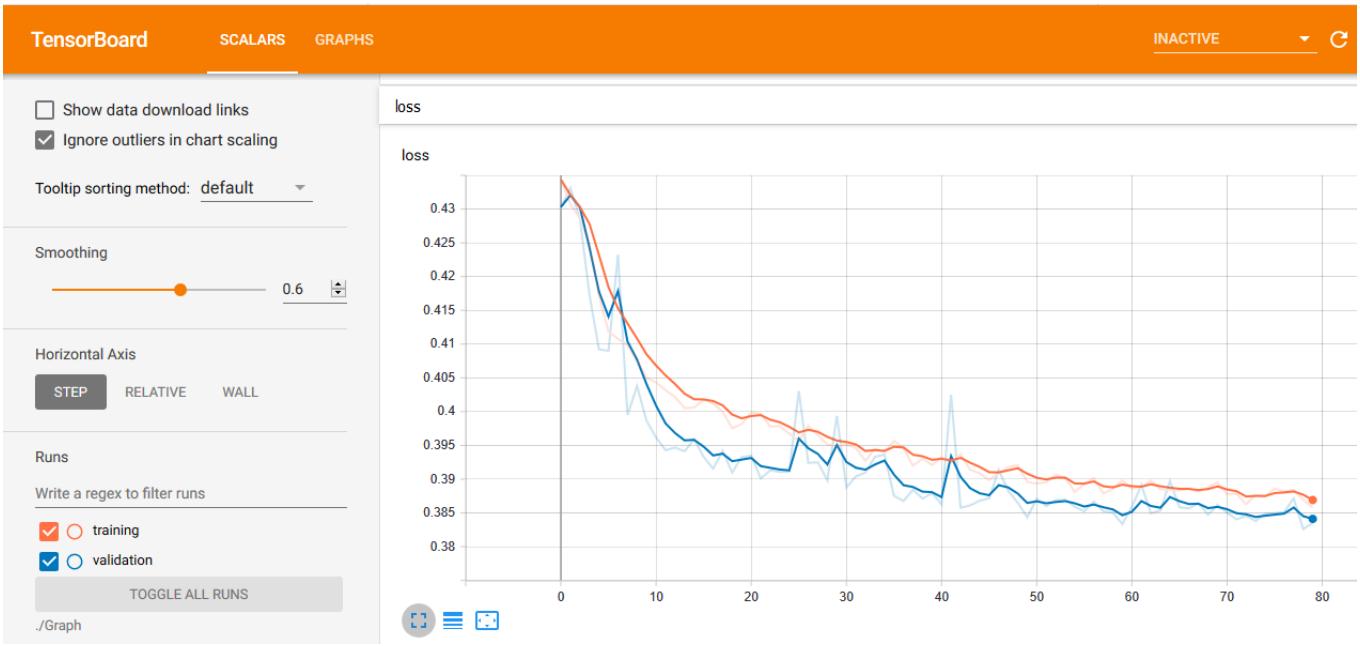




Next 80 epochs


TensorBoard
SCALARS
GRAPHS
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
training
validation
TOGGLE ALL RUNS
./Graph
auc
auc
0.75
0.73
0.71
0.69
0.67
loss
loss
0.43
0.42
0.41
0.4
0.39
0.38
0
10
20
30
40
50
60
70
80





2.9 Results & Model Testing

In [41]:

```
train_results = model2.evaluate([train_essay_padded,train_state_padded,train_grade_padd
ed,train_cat_padded,
                                 train_subcat_padded,train_prefix_padded,train_rem_inp],
y_train,
                                verbose=1,batch_size=500)
print('Train Loss: ',train_results[0])
print('Train AUC: ',train_results[1])
```

```
61178/61178 [==============================] - 59s 958us/step
Train Loss:  0.36955102657639677
Train AUC:  0.7752035285824126
```

In [42]:

```
cv_results = model2.evaluate([cv_essay_padded,cv_state_padded,cv_grade_padded,cv_cat_pa
dded,cv_subcat_padded,
                              cv_prefix_padded,cv_rem_inp],y_cv,verbose=1,batch_size=500
)
print('CV Loss: ',cv_results[0])
print('CV AUC: ',cv_results[1])
```

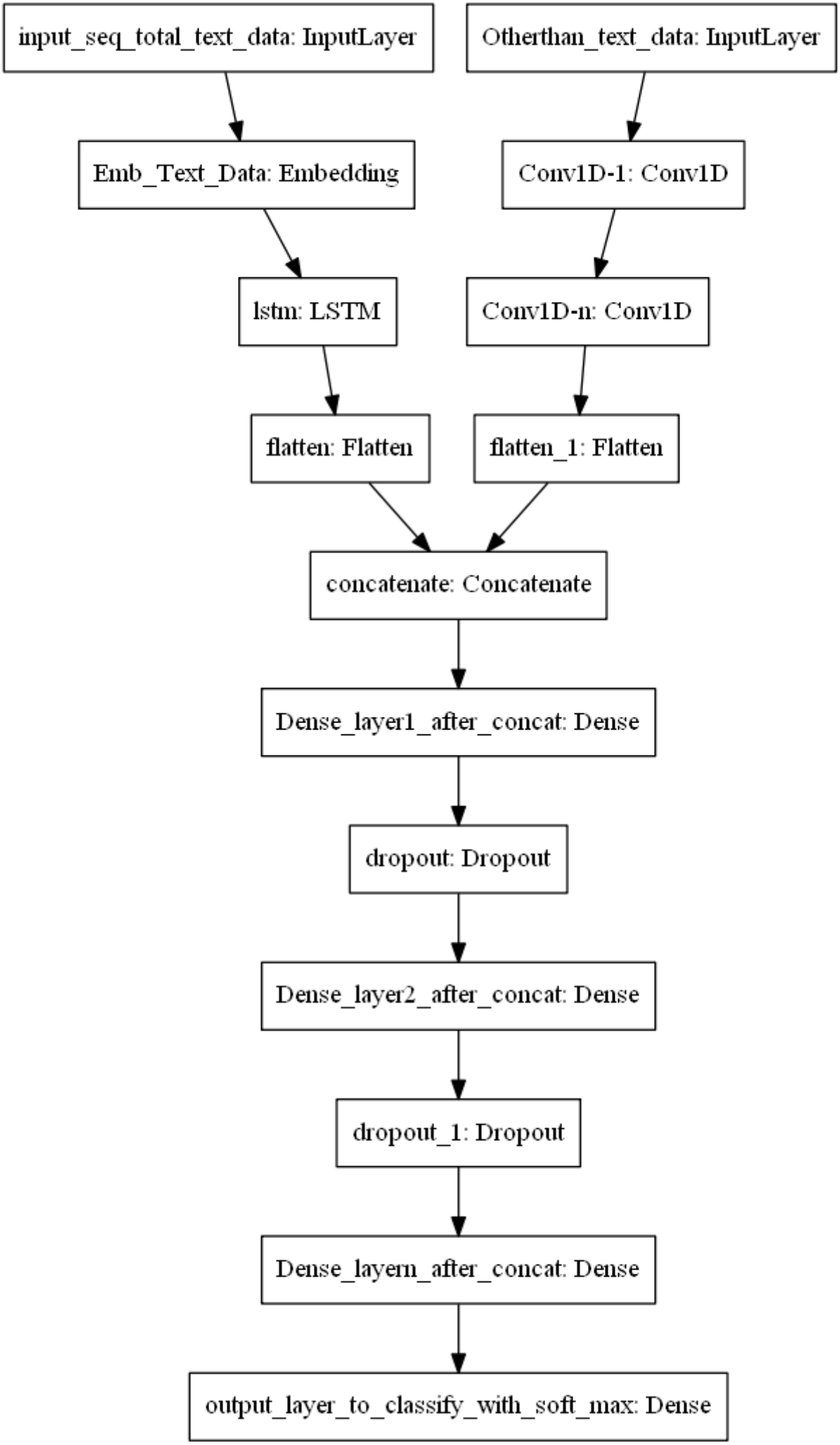
```
15295/15295 [==============================] - 15s 972us/step
CV Loss:  0.3835215083560588
CV AUC:  0.7490041170801435
```

In [43]:

```
test_results = model2.evaluate([test_essay_padded,test_state_padded,test_grade_padded,t
est_cat_padded,
                                test_subcat_padded,test_prefix_padded,test_rem_inp],y_t
est,verbose=1,
                               batch_size=500)
print('Test Loss: ',test_results[0])
print('Test AUC: ',test_results[1])
```

```
32775/32775 [==============================] - 32s 966us/step
Test Loss:  0.38155537489077046
Test AUC:  0.7521467415273507
```

Model-3


input_seq_total_text_data: InputLayer
Otherthan_text_data: InputLayer
Emb_Text_Data: Embedding
Conv1D-1: Conv1D
lstm: LSTM
Conv1D-n: Conv1D
flatten: Flatten
flatten_1: Flatten
concatenate: Concatenate
Dense_layer1_after_concat: Dense
dropout: Dropout
Dense_layer2_after_concat: Dense
dropout_1: Dropout
Dense_layern_after_concat: Dense
output_layer_to_classify_with_soft_max: Dense

ref: https://i.imgur.com/fkQ8nGo.png (https://i.imgur.com/fkQ8nGo.png)

- **input_seq_total_text_data**:

  . Use text column('essay'), and use the Embedding layer to get word vectors.

  . Use given predefined glove word vectors, don't train any word vectors.

  . Use LSTM that is given above, get the LSTM output and Flatten that output.

  . You are free to preprocess the input text as you needed.

- **Other_than_text_data**:

  . Convert all your Categorical values to onehot coded and then concatenate all these onehot vectors

  . Neumerical values and use CNN1D (https://keras.io/getting-started/sequential-model-guide/#sequence-classification-with-1d-convolutions) as shown in above figure.

  . You are free to choose all CNN parameters like kernel sizes, stride.

</pre>

In [2]:

```
import warnings
warnings.filterwarnings("ignore")
from collections import defaultdict
import matplotlib.pyplot as plt
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Dropout, LSTM, BatchNormalization,concatenate,Flatten,Embeddin
g,Dense,Dropout,MaxPooling2D,Reshape
from keras.models import Sequential
from keras import Model,Input
from keras.layers.convolutional import Conv2D,Conv1D
import keras.backend as k
from sklearn.metrics import roc_auc_score
import tensorflow as tf
import keras
from keras.initializers import he_normal,glorot_normal
from keras.regularizers import l1,l2
from keras.callbacks import Callback, EarlyStopping, ModelCheckpoint,LearningRateSchedu
ler
from time import time
from tensorflow.python.keras.callbacks import TensorBoard
from IPython.display import SVG, display
from keras.preprocessing.text import one_hot
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers.embeddings import Embedding
```

```
Using TensorFlow backend.
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade (https://www.tensorflow.org/guide/migrate) now or ensure your notebook will continue to use TensorFlow 1.x via the `%tensorflow_version 1.x` magic:
more info (https://colab.research.google.com/notebooks/tensorflow_version.ipynb).

Finding vocabulary for each feature

In [0]:

```
def unique(column):
    counter= CountVectorizer(lowercase=False) #columns like teacher_prefix have upperca
se letters
    matrix= counter.fit_transform(column.values)
    return matrix
```

In [0]:

```
essay_count= unique(X_train['preprocessed_essays'])
state_count= unique(X_train['school_state'])
grade_count= unique(X_train['preprocessed_project_grade_category'])
subject_cat_count= unique(X_train['clean_categories'])
subject_subcat_count= unique(X_train['clean_subcategories'])
teacher_prefix_count= unique(X_train['teacher_prefix'])
print('Essay:',essay_count.shape)
print('State:',state_count.shape)
print('Grade:',grade_count.shape)
print('Category:',subject_cat_count.shape)
print('Subcategory:',subject_subcat_count.shape)
print('Teacher prefix:',teacher_prefix_count.shape)
```

```
Essay: (61178, 44879)
State: (61178, 51)
Grade: (61178, 4)
Category: (61178, 9)
Subcategory: (61178, 30)
Teacher prefix: (61178, 5)
```

3.1 One-hot encoding of categorical features

School_state

In [0]:

```
vectorizer = CountVectorizer()
vectorizer.fit(X_train['school_state'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_state = vectorizer.transform(X_train['school_state'].values)
X_cv_state = vectorizer.transform(X_cv['school_state'].values)
X_test_state = vectorizer.transform(X_test['school_state'].values)

print("After vectorizations")
print(X_train_state.shape, y_train.shape)
print(X_cv_state.shape, y_cv.shape)
print(X_test_state.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)
```

```
After vectorizations
(61178, 51) (61178,)
(15295, 51) (15295,)
(32775, 51) (32775,)
['ak', 'al', 'ar', 'az', 'ca', 'co', 'ct', 'dc', 'de', 'fl', 'ga', 'hi',
'ia', 'id', 'il', 'in', 'ks', 'ky', 'la', 'ma', 'md', 'me', 'mi', 'mn', 'm
o', 'ms', 'mt', 'nc', 'nd', 'ne', 'nh', 'nj', 'nm', 'nv', 'ny', 'oh', 'o
k', 'or', 'pa', 'ri', 'sc', 'sd', 'tn', 'tx', 'ut', 'va', 'vt', 'wa', 'w
i', 'wv', 'wy']
====================================================================
==========================
```

Project grade category

In [0]:

```
#This step is to intialize a vectorizer with vocab from train data
#Ref: https://www.kaggle.com/shashank49/donors-choose-knn#Concatinating-all-features-(TFIDF)
from collections import Counter
my_counter = Counter()
for word in X_train['preprocessed_project_grade_category'].values:
    my_counter.update([word[i:i+14] for i in range(0, len(word),14)]) #https://www.geeksforgeeks.org/python-string-split/

# dict sort by value python: https://stackoverflow.com/a/613218/4084039
project_grade_category_dict = dict(my_counter)
sorted_project_grade_category_dict = dict(sorted(project_grade_category_dict.items(), key=lambda kv: kv[1]))
```

In [0]:

```
vectorizer = CountVectorizer(vocabulary=list(sorted_project_grade_category_dict.keys()), lowercase=False, binary=True,max_features=4)
vectorizer.fit(X_train['preprocessed_project_grade_category'].values) # fit has to happen only on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_grade = vectorizer.transform(X_train['preprocessed_project_grade_category'].values)
X_cv_grade = vectorizer.transform(X_cv['preprocessed_project_grade_category'].values)
X_test_grade = vectorizer.transform(X_test['preprocessed_project_grade_category'].values)

print("After vectorizations")
print(X_train_grade.shape, y_train.shape)
print(X_cv_grade.shape, y_cv.shape)
print(X_test_grade.shape, y_test.shape)
print(vectorizer.get_feature_names())
```

```
After vectorizations
(61178, 4) (61178,)
(15295, 4) (15295,)
(32775, 4) (32775,)
['grades_9_12', 'grades_6_8', 'grades_3_5', 'grades_prek_2']
```

Project category

In [0]:

```
vectorizer = CountVectorizer()
vectorizer.fit(X_train['clean_categories'].values) # fit has to happen only on train da
ta

# we use the fitted CountVectorizer to convert the text to vector
X_train_cat = vectorizer.transform(X_train['clean_categories'].values)
X_cv_cat = vectorizer.transform(X_cv['clean_categories'].values)
X_test_cat = vectorizer.transform(X_test['clean_categories'].values)

print("After vectorizations")
print(X_train_cat.shape, y_train.shape)
print(X_cv_cat.shape, y_cv.shape)
print(X_test_cat.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)
```

```
After vectorizations
(61178, 9) (61178,)
(15295, 9) (15295,)
(32775, 9) (32775,)
['appliedlearning', 'care_hunger', 'health_sports', 'history_civics', 'lit
eracy_language', 'math_science', 'music_arts', 'specialneeds', 'warmth']
============================================================================
==========================
```

Project subcategory

In [0]:

```
vectorizer = CountVectorizer()
vectorizer.fit(X_train['clean_subcategories'].values) # fit has to happen only on train
data

# we use the fitted CountVectorizer to convert the text to vector
X_train_subcat = vectorizer.transform(X_train['clean_subcategories'].values)
X_cv_subcat = vectorizer.transform(X_cv['clean_subcategories'].values)
X_test_subcat = vectorizer.transform(X_test['clean_subcategories'].values)

print("After vectorizations")
print(X_train_subcat.shape, y_train.shape)
print(X_cv_subcat.shape, y_cv.shape)
print(X_test_subcat.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)
```

```
After vectorizations
(61178, 30) (61178,)
(15295, 30) (15295,)
(32775, 30) (32775,)
['appliedsciences', 'care_hunger', 'charactereducation', 'civics_governmen
t', 'college_careerprep', 'communityservice', 'earlydevelopment', 'economi
cs', 'environmentalscience', 'esl', 'extracurricular', 'financialliterac
y', 'foreignlanguages', 'gym_fitness', 'health_lifescience', 'health_welln
ess', 'history_geography', 'literacy', 'literature_writing', 'mathematic
s', 'music', 'nutritioneducation', 'other', 'parentinvolvement', 'performi
ngarts', 'socialsciences', 'specialneeds', 'teamsports', 'visualarts', 'wa
rmth']
====================================================================
==========================
```

Teacher prefix

In [0]:

```
vectorizer = CountVectorizer()
vectorizer.fit(X_train['teacher_prefix'].values)

X_train_teacher = vectorizer.transform(X_train['teacher_prefix'].values)
X_cv_teacher = vectorizer.transform(X_cv['teacher_prefix'].values)
X_test_teacher = vectorizer.transform(X_test['teacher_prefix'].values)

print("After vectorizations")
print(X_train_teacher.shape, y_train.shape)
print(X_cv_teacher.shape, y_cv.shape)
print(X_test_teacher.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)
```

```
After vectorizations
(61178, 5) (61178,)
(15295, 5) (15295,)
(32775, 5) (32775,)
['dr', 'mr', 'mrs', 'ms', 'teacher']
====================================================================
==========================
```

3.2 Normalizing numerical features

In [0]:

```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(feature.values) will result in an error: Expected 2D array, got 1D arr
ay instead.
normalizer.fit(X_train['Numerical_features'].values.reshape(1,-1))

X_train_norm = normalizer.transform(X_train['Numerical_features'].values.reshape(-1,1))
X_cv_norm = normalizer.transform(X_cv['Numerical_features'].values.reshape(-1,1))
X_test_norm = normalizer.transform(X_test['Numerical_features'].values.reshape(-1,1))

print("After vectorizations")
print(X_train_norm.shape, y_train.shape)
print(X_cv_norm.shape, y_cv.shape)
print(X_test_norm.shape, y_test.shape)
print("="*100)
```

```
After vectorizations
(61178, 1) (61178,)
(15295, 1) (15295,)
(32775, 1) (32775,)
====================================================================================
==========================
```

3.3 Encoding & padding essay

In [0]:

```
def encoder(feature):
    t = Tokenizer()
    t.fit_on_texts(feature)
    vocab_size = len(t.word_index) + 1
    # integer encode the documents
    encoded_docs = t.texts_to_sequences(feature)
    return encoded_docs,vocab_size,t
```

In [0]:

```
def padding(encoded_docs,max_length):
    padded_docs = pad_sequences(encoded_docs, maxlen=max_length, padding='post')
    return padded_docs
```

In [0]:

```
#train data
docs,vocab,t1=encoder(X_train.preprocessed_essays)
print(vocab)
```

```
44908
```

In [0]:

```
#to get an estimate of review length of the truncatd essay column
essay_count=[]
for text in tqdm(X_train.preprocessed_essays.values):
    c=len(text.split()) #https://www.geeksforgeeks.org/python-program-to-count-words-in
-a-sentence/
    essay_count.append(c)
print(max(essay_count))
```

```
100%|██████████| 61178/61178 [00:00<00:00, 146023.87it/s]

315
```

In [0]:

```
train_essay_padded = padding(docs,500)
print(train_essay_padded.shape)
print(train_essay_padded[5])
```

```
(61178, 500)
[    1   658    42     2     1  1468   311   333    44  2985    21    13
  1548  9903   479  9734   718  1588     2   323    70   339   650  3615
  6026     2  1769  3916     7     1   118    54   137    71  1164  1299
  2043    29    46    13   337     2   102   517     7   186   115   189
     2  1508  2546   128   164   964    28   531     1   329  1430  1172
    59    42     2     4     5   228    59   986    60    19     1     5
   655  3623  3253   260  1773   829    45  7740   256   410  1758    10
    66   391   263   850  1111    28  5765    18    44   299    23   242
   677   113  1067     1   345   291     1    31   135  2986   530    14
  2183   209   443  3043 28799  3043  4610  4323  3043 10538  3043  9240
  3043  9087   366  1255  2533    28  5765   680    80    41   292   477
   733    28   531    18    44    21   299  1362   666     2   775  6356
     1   374    87    41    25   200   359   147    28   531  1342   281
  3201     8     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0]
```

In [0]:

```
#cv data
docs = t1.texts_to_sequences(X_cv.preprocessed_essays)
cv_essay_padded = padding(docs,500)
print(cv_essay_padded.shape)
print(cv_essay_padded[5])
```

```
(15295, 500)
[    4   883   296  2674   197  2352    28  2166  1557  1238  1454    58
   107  2470   264   422   818    57  1662   524  1662  6943  1662     1
  1529   512  1547   744    60  2301    57  6158  6594  5037     1   140
  1194   358   364   922  1322   331    64   198 10009   404   578  6394
   189   298  1462    40   241    23    36    14   280   430  2849   548
  4713    27    63   496   197   726    92    49   581  1204  2849    63
   521   315  1055   358   148  1141   755   226  5060   663     7     1
    68  5822  4713    20    58  1355  1355  1174   144    18    44    45
  1053 13665    44   125     4     1   160   663   292   106  1055   741
   226   119  2390   125  6950  2562   393  3554   714   889   674   896
    24     8     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0]
```

In [0]:

```
#test data
docs = t1.texts_to_sequences(X_test.preprocessed_essays)
test_essay_padded = padding(docs,500)
print(test_essay_padded.shape)
print(test_essay_padded[5])
```

```
(32775, 500)
[    4  4603   231  2741   733  1093   192   563  3640  2207   105   419
  2913    40    62    30    91   396  2319    44   338   207  1395     1
  1861  2675   382   354    83     5   498     7    95     9    52     7
    52    59    18    25   238  1360  1539   983   229     2   109     5
   144   293     1    76   221   926    76   240     9    12   107   613
    33  1249   173    22   780    72    42    76   164    60   633   943
   121    20     3    17   144    25   633   112     9   367   107    33
    21     9   267    33    95  1030   633   265  4894  1588    21    82
    90    27  4894  1588     6 11911   592    11    21   207   453    12
  2252  1862    90   350   259  1227   164   753   436     1   633    45
  1227    34   300   188    70   184     6   489     1    21     1     9
   858     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0]
```

3.4 Concatenating categorical & numerical features

In [0]:

```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack

X_train_rem = hstack((X_train_state, X_train_teacher, X_train_grade, X_train_cat, X_tra
in_subcat, X_train_norm)).todense()

X_cv_rem = hstack((X_cv_state, X_cv_teacher, X_cv_grade, X_cv_cat, X_cv_subcat, X_cv_no
rm)).todense()

X_test_rem = hstack((X_test_state, X_test_teacher, X_test_grade, X_test_cat, X_test_sub
cat, X_test_norm)).todense()

print("Final Data Matrix")
print(X_train_rem.shape, y_train.shape)
print(X_cv_rem.shape, y_train.shape)
print(X_test_rem.shape, y_train.shape)
```

```
Final Data Matrix
(61178, 100) (61178,)
(15295, 100) (61178,)
(32775, 100) (61178,)
```

3.5 Converting class labels to vectors using one-hot encoding

In [0]:

```
from keras.utils import to_categorical
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
y_cv = to_categorical(y_cv)
```

3.6 Saving all tensors for further use

In [0]:

```
#https://www.geeksforgeeks.org/numpy-save/
np.save('train_essay_padded', train_essay_padded)
np.save('cv_essay_padded', cv_essay_padded)
np.save('test_essay_padded', test_essay_padded)

np.save('X_train_rem',X_train_rem)
np.save('X_cv_rem', X_cv_rem)
np.save('X_test_rem', X_test_rem)

np.save('y_train', y_train)
np.save('y_test', y_test)
np.save('y_cv', y_cv)
```

In [0]:

```
#Loading the tensors
train_essay_padded= np.load('/content/Saved files_cycle1/train_essay_padded.npy')
cv_essay_padded= np.load('/content/Saved files_cycle1/cv_essay_padded.npy')
test_essay_padded= np.load('/content/Saved files_cycle1/test_essay_padded.npy')

X_train_rem= np.load('/content/Saved files_cycle1/X_train_rem.npy')
X_cv_rem= np.load('/content/Saved files_cycle1/X_cv_rem.npy')
X_test_rem= np.load('/content/Saved files_cycle1/X_test_rem.npy')

y_train= np.load('/content/Saved files_cycle1/y_train.npy')
y_test= np.load('/content/Saved files_cycle1/y_test.npy')
y_cv= np.load('/content/Saved files_cycle1/y_cv.npy')
```

3.7 Loading the pre-trained glove model

In [0]:

```
with open('/content/Data/glove_vectors', 'rb') as f:
    glove = pickle.load(f)
print ("Done.",len(glove)," words loaded!")
```

```
Done. 51510  words loaded!
```

In [0]:

```
type(glove)
```

Out[0]:

```
dict
```

3.8 Defining the performance metric[ROC]

In [0]:

```
def auc( y_true, y_pred ) :
    score = tf.py_func( lambda y_true, y_pred : roc_auc_score( y_true, y_pred, average=
'macro', sample_weight=None).astype('float32'),
                        [y_true, y_pred],'float32',stateful=True,name='sklearnAUC')
    return score
```

3.9 LSTM model

3.9.1 Creating the 2D Embedding matrix using Glove vectors

In [0]:

```
#Credits: https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/

embedding_matrix = np.zeros((44908, 300))
for word, i in t1.word_index.items():
    embedding_vector = glove.get(word)
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector
```

In [0]:

```
np.save('embedding_matrix',embedding_matrix)
```

In [0]:

```
embedding_matrix= np.load('/content/Saved files_cycle1/embedding_matrix.npy')
```

In [0]:

```
import warnings
warnings.filterwarnings("ignore")
from collections import defaultdict
import matplotlib.pyplot as plt
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Dropout, LSTM, BatchNormalization,concatenate,Flatten,Embedding,Dense,Dropout,MaxPooling2D,Reshape
from keras.models import Sequential
from keras import Model,Input
from keras.layers.convolutional import Conv2D,Conv1D
import keras.backend as k
from sklearn.metrics import roc_auc_score
import tensorflow as tf
import keras
from keras.initializers import he_normal,glorot_normal
from keras.regularizers import l1,l2
from keras.callbacks import Callback, EarlyStopping, ModelCheckpoint,LearningRateScheduler
from time import time
from tensorflow.python.keras.callbacks import TensorBoard
from IPython.display import SVG, display
from keras.layers import LeakyReLU
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.layers import Embedding
from keras.layers import Conv1D, GlobalAveragePooling1D, MaxPooling1D
```

3.9.2 Reshaping the remaining data

In [0]:

```
#https://docs.scipy.org/doc/numpy/reference/generated/numpy.resize.html
X_train_rem = np.resize(X_train_rem, (X_train_rem.shape[0],X_train_rem.shape[1],1))
X_cv_rem = np.resize(X_cv_rem, (X_cv_rem.shape[0],X_cv_rem.shape[1],1))
X_test_rem = np.resize(X_test_rem, (X_test_rem.shape[0],X_test_rem.shape[1],1))
```

In [31]:

```
print(X_train_rem.shape)
```

```
(61178, 100, 1)
```

3.9.3 Architecture

In [0]:

```
import keras.backend as K
K.clear_session()
```

In [0]:

```
#essay
input1 = Input(shape=(500,))
i1 = Embedding(input_dim=44908,output_dim= 300,input_length=500,weights=[embedding_matr
ix],trainable=False)(input1)
i1 = Dropout(0.5)(i1)
i1 = LSTM(128,kernel_initializer='he_normal',recurrent_dropout=0.5,kernel_regularizer=l
2(0.001),return_sequences=True)(i1)
i1= LeakyReLU(alpha = 0.5)(i1)
f1 = Flatten()(i1)

#remaining inputs
#ref: https://keras.io/getting-started/sequential-model-guide/#sequence-classification-
with-1d-convolutions
input2 = Input(shape=(100,1))
i2 = Conv1D(64, 3, activation='relu',strides=1,kernel_initializer=he_normal())(input2)
i2 = Conv1D(64, 3, activation='relu',strides=1,kernel_initializer=he_normal())(i2)
i2 = MaxPooling1D(3)(i2)
f2 = Flatten()(i2)

#concatenating input1 & input2
concat = concatenate([f1,f2])


l = Dense(128,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.
001))(concat)
l = Dropout(0.5)(l)
l = Dense(64,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.0
01))(l)
l = Dropout(0.5)(l)
l = BatchNormalization()(l)
l = Dense(32,activation='relu',kernel_initializer=he_normal(),kernel_regularizer=l2(0.0
01))(l)
l = Dropout(0.5)(l)
output = Dense(2, activation = 'softmax')(l)
```

In [34]:

```
# create model with seven inputs
model3 = Model(inputs=[input1,input2], outputs=[output])
model3.summary()
```

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 500)          0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 500, 300)     13472400    input_1[0][0]
__________________________________________________________________________________________________
input_2 (InputLayer)            (None, 100, 1)       0
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 500, 300)     0           embedding_1[0][0]
__________________________________________________________________________________________________
conv1d_1 (Conv1D)               (None, 98, 64)       256         input_2[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 500, 128)     219648      dropout_1[0][0]
__________________________________________________________________________________________________
conv1d_2 (Conv1D)               (None, 96, 64)       12352       conv1d_1[0][0]
__________________________________________________________________________________________________
leaky_re_lu_1 (LeakyReLU)       (None, 500, 128)     0           lstm_1[0][0]
__________________________________________________________________________________________________
max_pooling1d_1 (MaxPooling1D)  (None, 32, 64)       0           conv1d_2[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 64000)        0           leaky_re_lu_1[0][0]
__________________________________________________________________________________________________
flatten_2 (Flatten)             (None, 2048)         0           max_pooling1d_1[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 66048)        0           flatten_1[0][0]
                                                                 flatten_2[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 128)          8454272     concatenate_1[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 128)          0           dense_1[0][0]
```

```
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256      dropout_2[0][0]
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0         dense_2[0][0]
_________________________________________________________________
batch_normalization_1 (BatchNor (None, 64)             256       dropout_3[0][0]
_________________________________________________________________
dense_3 (Dense)              (None, 32)                2080      batch_normalization_1[0][0]
_________________________________________________________________
dropout_4 (Dropout)          (None, 32)                0         dense_3[0][0]
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 66        dropout_4[0][0]
=================================================================
Total params: 22,169,586
Trainable params: 8,697,058
Non-trainable params: 13,472,528
_________________________________________________________________
```

In [0]:

```
model3.load_weights("/content/weights1it.best.hdf5")
```

In [0]:

```
model3.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.Adam(lr=0.001,decay = 1e-4),metrics=[auc])
```

3.9.4 Checkpointing the model and creating the callback list

In [37]:

```
from keras.callbacks import ModelCheckpoint
from keras.callbacks import CSVLogger
import matplotlib.pyplot as plt
from tensorflow.python.keras.callbacks import TensorBoard
from keras.callbacks import TensorBoard
import tensorflow as tf
import datetime
import keras
from tensorboardcolab import *
from keras.callbacks import ReduceLROnPlateau

#https://github.com/taomanwai/tensorboardcolab/
tbc=TensorBoardColab()
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2,
                              patience=1, min_lr=0.002,verbose = 1)

filepath="weights2it.best.hdf5"
checkpoints = ModelCheckpoint(filepath, monitor='val_auc', verbose=1, save_best_only=True, mode='max')
train_results = CSVLogger('train_results_2.log') #storing the training results in a pandas dataframe
callbacks_list = [checkpoints, TensorBoardColabCallback(tbc), train_results]
```

```
Wait for 8 seconds...
TensorBoard link:
https://8acb5581.ngrok.io
```

3.9.5 Fitting the model in batches

In [0]:

```
# finding the class weights before fitting the model
from sklearn.utils import compute_class_weight
class_wts = compute_class_weight("balanced", classes= np.unique(y),y=y)
print(class_wts)
```

```
[3.30214001 0.58921753]
```

In [0]:

```
np.save('class_wts', class_wts)
```

In [0]:

```
class_wts= np.load('/content/Saved files_cycle1/class_wts.npy')
```

In [18]:

```
#2nd cycle of epoch size=100
history=model3.fit([train_essay_padded,X_train_rem], y_train, nb_epoch=100,verbose=1,ba
tch_size=500,
          validation_data=([cv_essay_padded,X_cv_rem],y_cv),
          callbacks =callbacks_list,class_weight = class_wts)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_
core/python/ops/math_grad.py:1424: where (from tensorflow.python.ops.array
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Pleas
e use tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backe
nd/tensorflow_backend.py:1020: The name tf.assign is deprecated. Please us
e tf.compat.v1.assign instead.

Train on 61178 samples, validate on 15295 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorboard
colab/core.py:49: The name tf.summary.FileWriter is deprecated. Please use
tf.compat.v1.summary.FileWriter instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callb
acks.py:1122: The name tf.summary.merge_all is deprecated. Please use tf.c
ompat.v1.summary.merge_all instead.

Epoch 1/100
61178/61178 [==============================] - 144s 2ms/step - loss: 1.718
5 - auc: 0.5033 - val_loss: 0.9219 - val_auc: 0.5215

Epoch 00001: val_auc improved from -inf to 0.52153, saving model to weight
s1it.best.hdf5
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorboard
colab/callbacks.py:51: The name tf.Summary is deprecated. Please use tf.co
mpat.v1.Summary instead.

Epoch 2/100
61178/61178 [==============================] - 140s 2ms/step - loss: 0.866
6 - auc: 0.5115 - val_loss: 0.6839 - val_auc: 0.5030

Epoch 00002: val_auc did not improve from 0.52153
Epoch 3/100
61178/61178 [==============================] - 138s 2ms/step - loss: 0.660
5 - auc: 0.5333 - val_loss: 0.6268 - val_auc: 0.4662

Epoch 00003: val_auc did not improve from 0.52153
Epoch 4/100
61178/61178 [==============================] - 139s 2ms/step - loss: 0.624
2 - auc: 0.5337 - val_loss: 0.5558 - val_auc: 0.5868

Epoch 00004: val_auc improved from 0.52153 to 0.58681, saving model to wei
ghts1it.best.hdf5
Epoch 5/100
61178/61178 [==============================] - 140s 2ms/step - loss: 0.559
6 - auc: 0.5467 - val_loss: 0.5640 - val_auc: 0.6613

Epoch 00005: val_auc improved from 0.58681 to 0.66130, saving model to wei
ghts1it.best.hdf5
Epoch 6/100
61178/61178 [==============================] - 141s 2ms/step - loss: 0.570
0 - auc: 0.5613 - val_loss: 0.5579 - val_auc: 0.6480

Epoch 00006: val_auc did not improve from 0.66130
Epoch 7/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.513
```

```
5 - auc: 0.5873 - val_loss: 0.5077 - val_auc: 0.6906

Epoch 00007: val_auc improved from 0.66130 to 0.69059, saving model to wei
ghts1it.best.hdf5
Epoch 8/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.490
5 - auc: 0.6299 - val_loss: 0.4891 - val_auc: 0.7040

Epoch 00008: val_auc improved from 0.69059 to 0.70403, saving model to wei
ghts1it.best.hdf5
Epoch 9/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.472
8 - auc: 0.6591 - val_loss: 0.4719 - val_auc: 0.7104

Epoch 00009: val_auc improved from 0.70403 to 0.71043, saving model to wei
ghts1it.best.hdf5
Epoch 10/100
61178/61178 [==============================] - 141s 2ms/step - loss: 0.474
8 - auc: 0.6630 - val_loss: 0.4716 - val_auc: 0.7175

Epoch 00010: val_auc improved from 0.71043 to 0.71748, saving model to wei
ghts1it.best.hdf5
Epoch 11/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.460
1 - auc: 0.6836 - val_loss: 0.4611 - val_auc: 0.7268

Epoch 00011: val_auc improved from 0.71748 to 0.72676, saving model to wei
ghts1it.best.hdf5
Epoch 12/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.445
2 - auc: 0.6930 - val_loss: 0.4482 - val_auc: 0.7351

Epoch 00012: val_auc improved from 0.72676 to 0.73509, saving model to wei
ghts1it.best.hdf5
Epoch 13/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.437
5 - auc: 0.6984 - val_loss: 0.4417 - val_auc: 0.7388

Epoch 00013: val_auc improved from 0.73509 to 0.73879, saving model to wei
ghts1it.best.hdf5
Epoch 14/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.432
0 - auc: 0.7016 - val_loss: 0.4309 - val_auc: 0.7392

Epoch 00014: val_auc improved from 0.73879 to 0.73919, saving model to wei
ghts1it.best.hdf5
Epoch 15/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.425
6 - auc: 0.7065 - val_loss: 0.4243 - val_auc: 0.7452

Epoch 00015: val_auc improved from 0.73919 to 0.74518, saving model to wei
ghts1it.best.hdf5
Epoch 16/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.425
0 - auc: 0.7046 - val_loss: 0.4272 - val_auc: 0.7459

Epoch 00016: val_auc improved from 0.74518 to 0.74593, saving model to wei
ghts1it.best.hdf5
Epoch 17/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.422
7 - auc: 0.7061 - val_loss: 0.4283 - val_auc: 0.7450
```

```
Epoch 00017: val_auc did not improve from 0.74593
Epoch 18/100
61178/61178 [==============================] - 141s 2ms/step - loss: 0.419
5 - auc: 0.7043 - val_loss: 0.4169 - val_auc: 0.7423

Epoch 00018: val_auc did not improve from 0.74593
Epoch 19/100
61178/61178 [==============================] - 141s 2ms/step - loss: 0.414
5 - auc: 0.7135 - val_loss: 0.4244 - val_auc: 0.7445

Epoch 00019: val_auc did not improve from 0.74593
Epoch 20/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.414
4 - auc: 0.7117 - val_loss: 0.4097 - val_auc: 0.7437

Epoch 00020: val_auc did not improve from 0.74593
Epoch 21/100
61178/61178 [==============================] - 141s 2ms/step - loss: 0.412
9 - auc: 0.7117 - val_loss: 0.4169 - val_auc: 0.7493

Epoch 00021: val_auc improved from 0.74593 to 0.74932, saving model to wei
ghts1it.best.hdf5
Epoch 22/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.410
9 - auc: 0.7138 - val_loss: 0.4122 - val_auc: 0.7480

Epoch 00022: val_auc did not improve from 0.74932
Epoch 23/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.410
1 - auc: 0.7147 - val_loss: 0.4072 - val_auc: 0.7474

Epoch 00023: val_auc did not improve from 0.74932
Epoch 24/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.410
1 - auc: 0.7128 - val_loss: 0.4066 - val_auc: 0.7481

Epoch 00024: val_auc did not improve from 0.74932
Epoch 25/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.407
6 - auc: 0.7190 - val_loss: 0.4017 - val_auc: 0.7479

Epoch 00025: val_auc did not improve from 0.74932
Epoch 26/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.407
1 - auc: 0.7172 - val_loss: 0.3995 - val_auc: 0.7522

Epoch 00026: val_auc improved from 0.74932 to 0.75216, saving model to wei
ghts1it.best.hdf5
Epoch 27/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.407
9 - auc: 0.7146 - val_loss: 0.4060 - val_auc: 0.7473

Epoch 00027: val_auc did not improve from 0.75216
Epoch 28/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.405
7 - auc: 0.7180 - val_loss: 0.4006 - val_auc: 0.7500

Epoch 00028: val_auc did not improve from 0.75216
Epoch 29/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.404
```

```
4 - auc: 0.7183 - val_loss: 0.4039 - val_auc: 0.7468

Epoch 00029: val_auc did not improve from 0.75216
Epoch 30/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.404
8 - auc: 0.7170 - val_loss: 0.3934 - val_auc: 0.7513

Epoch 00030: val_auc did not improve from 0.75216
Epoch 31/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.404
3 - auc: 0.7170 - val_loss: 0.4024 - val_auc: 0.7501

Epoch 00031: val_auc did not improve from 0.75216
Epoch 32/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.403
3 - auc: 0.7194 - val_loss: 0.4039 - val_auc: 0.7476

Epoch 00032: val_auc did not improve from 0.75216
Epoch 33/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.403
5 - auc: 0.7206 - val_loss: 0.3965 - val_auc: 0.7528

Epoch 00033: val_auc improved from 0.75216 to 0.75281, saving model to wei
ghts1it.best.hdf5
Epoch 34/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.403
0 - auc: 0.7212 - val_loss: 0.4029 - val_auc: 0.7523

Epoch 00034: val_auc did not improve from 0.75281
Epoch 35/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.402
6 - auc: 0.7208 - val_loss: 0.3937 - val_auc: 0.7528

Epoch 00035: val_auc did not improve from 0.75281
Epoch 36/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.401
5 - auc: 0.7209 - val_loss: 0.3925 - val_auc: 0.7504

Epoch 00036: val_auc did not improve from 0.75281
Epoch 37/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.400
2 - auc: 0.7237 - val_loss: 0.3952 - val_auc: 0.7523

Epoch 00037: val_auc did not improve from 0.75281
Epoch 38/100
61178/61178 [==============================] - 141s 2ms/step - loss: 0.401
5 - auc: 0.7223 - val_loss: 0.3937 - val_auc: 0.7519

Epoch 00038: val_auc did not improve from 0.75281
Epoch 39/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.401
0 - auc: 0.7224 - val_loss: 0.4035 - val_auc: 0.7524

Epoch 00039: val_auc did not improve from 0.75281
Epoch 40/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.401
0 - auc: 0.7240 - val_loss: 0.3937 - val_auc: 0.7517

Epoch 00040: val_auc did not improve from 0.75281
Epoch 41/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.403
```

```
0 - auc: 0.7196 - val_loss: 0.3988 - val_auc: 0.7489

Epoch 00041: val_auc did not improve from 0.75281
Epoch 42/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.401
1 - auc: 0.7234 - val_loss: 0.3931 - val_auc: 0.7515

Epoch 00042: val_auc did not improve from 0.75281
Epoch 43/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.400
7 - auc: 0.7236 - val_loss: 0.3918 - val_auc: 0.7540

Epoch 00043: val_auc improved from 0.75281 to 0.75399, saving model to wei
ghts1it.best.hdf5
Epoch 44/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.401
3 - auc: 0.7227 - val_loss: 0.3966 - val_auc: 0.7529

Epoch 00044: val_auc did not improve from 0.75399
Epoch 45/100
61178/61178 [==============================] - 141s 2ms/step - loss: 0.401
1 - auc: 0.7249 - val_loss: 0.3959 - val_auc: 0.7538

Epoch 00045: val_auc did not improve from 0.75399
Epoch 46/100
61178/61178 [==============================] - 141s 2ms/step - loss: 0.401
0 - auc: 0.7240 - val_loss: 0.3979 - val_auc: 0.7544

Epoch 00046: val_auc improved from 0.75399 to 0.75441, saving model to wei
ghts1it.best.hdf5
Epoch 47/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.400
5 - auc: 0.7256 - val_loss: 0.3912 - val_auc: 0.7499

Epoch 00047: val_auc did not improve from 0.75441
Epoch 48/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.400
1 - auc: 0.7252 - val_loss: 0.3954 - val_auc: 0.7567

Epoch 00048: val_auc improved from 0.75441 to 0.75668, saving model to wei
ghts1it.best.hdf5
Epoch 49/100
61178/61178 [==============================] - 141s 2ms/step - loss: 0.401
0 - auc: 0.7247 - val_loss: 0.3932 - val_auc: 0.7552

Epoch 00049: val_auc did not improve from 0.75668
Epoch 50/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.399
7 - auc: 0.7259 - val_loss: 0.4028 - val_auc: 0.7535

Epoch 00050: val_auc did not improve from 0.75668
Epoch 51/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.398
1 - auc: 0.7288 - val_loss: 0.3943 - val_auc: 0.7556

Epoch 00051: val_auc did not improve from 0.75668
Epoch 52/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.399
2 - auc: 0.7265 - val_loss: 0.3963 - val_auc: 0.7521

Epoch 00052: val_auc did not improve from 0.75668
```

```
Epoch 53/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.400
6 - auc: 0.7251 - val_loss: 0.3909 - val_auc: 0.7552

Epoch 00053: val_auc did not improve from 0.75668
Epoch 54/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.400
2 - auc: 0.7277 - val_loss: 0.3978 - val_auc: 0.7549

Epoch 00054: val_auc did not improve from 0.75668
Epoch 55/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.398
9 - auc: 0.7268 - val_loss: 0.3962 - val_auc: 0.7537

Epoch 00055: val_auc did not improve from 0.75668
Epoch 56/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.398
8 - auc: 0.7289 - val_loss: 0.3995 - val_auc: 0.7525

Epoch 00056: val_auc did not improve from 0.75668
Epoch 57/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.399
1 - auc: 0.7293 - val_loss: 0.3928 - val_auc: 0.7542

Epoch 00057: val_auc did not improve from 0.75668
Epoch 58/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.399
0 - auc: 0.7291 - val_loss: 0.3956 - val_auc: 0.7546

Epoch 00058: val_auc did not improve from 0.75668
Epoch 59/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.398
4 - auc: 0.7278 - val_loss: 0.3911 - val_auc: 0.7572

Epoch 00059: val_auc improved from 0.75668 to 0.75718, saving model to wei
ghts1it.best.hdf5
Epoch 60/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.398
8 - auc: 0.7267 - val_loss: 0.4051 - val_auc: 0.7552

Epoch 00060: val_auc did not improve from 0.75718
Epoch 61/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.398
2 - auc: 0.7309 - val_loss: 0.3933 - val_auc: 0.7565

Epoch 00061: val_auc did not improve from 0.75718
Epoch 62/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.398
2 - auc: 0.7291 - val_loss: 0.3935 - val_auc: 0.7579

Epoch 00062: val_auc improved from 0.75718 to 0.75787, saving model to wei
ghts1it.best.hdf5
Epoch 63/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.399
1 - auc: 0.7285 - val_loss: 0.3921 - val_auc: 0.7573

Epoch 00063: val_auc did not improve from 0.75787
Epoch 64/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.399
1 - auc: 0.7268 - val_loss: 0.3880 - val_auc: 0.7548
```

```
Epoch 00064: val_auc did not improve from 0.75787
Epoch 65/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.399
4 - auc: 0.7270 - val_loss: 0.3956 - val_auc: 0.7558

Epoch 00065: val_auc did not improve from 0.75787
Epoch 66/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.397
9 - auc: 0.7307 - val_loss: 0.3995 - val_auc: 0.7556

Epoch 00066: val_auc did not improve from 0.75787
Epoch 67/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.398
3 - auc: 0.7291 - val_loss: 0.3956 - val_auc: 0.7561

Epoch 00067: val_auc did not improve from 0.75787
Epoch 68/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.396
0 - auc: 0.7352 - val_loss: 0.3905 - val_auc: 0.7550

Epoch 00068: val_auc did not improve from 0.75787
Epoch 69/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.398
2 - auc: 0.7290 - val_loss: 0.4002 - val_auc: 0.7543

Epoch 00069: val_auc did not improve from 0.75787
Epoch 70/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.396
3 - auc: 0.7351 - val_loss: 0.3910 - val_auc: 0.7553

Epoch 00070: val_auc did not improve from 0.75787
Epoch 71/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.398
7 - auc: 0.7311 - val_loss: 0.3996 - val_auc: 0.7569

Epoch 00071: val_auc did not improve from 0.75787
Epoch 72/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.396
8 - auc: 0.7338 - val_loss: 0.4016 - val_auc: 0.7576

Epoch 00072: val_auc did not improve from 0.75787
Epoch 73/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.397
9 - auc: 0.7348 - val_loss: 0.4023 - val_auc: 0.7588

Epoch 00073: val_auc improved from 0.75787 to 0.75876, saving model to wei
ghts1it.best.hdf5
Epoch 74/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.397
5 - auc: 0.7334 - val_loss: 0.4063 - val_auc: 0.7569

Epoch 00074: val_auc did not improve from 0.75876
Epoch 75/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.396
8 - auc: 0.7362 - val_loss: 0.4005 - val_auc: 0.7563

Epoch 00075: val_auc did not improve from 0.75876
Epoch 76/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.396
4 - auc: 0.7351 - val_loss: 0.3929 - val_auc: 0.7562
```

```
Epoch 00076: val_auc did not improve from 0.75876
Epoch 77/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.395
7 - auc: 0.7384 - val_loss: 0.3959 - val_auc: 0.7532

Epoch 00077: val_auc did not improve from 0.75876
Epoch 78/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.396
4 - auc: 0.7380 - val_loss: 0.3935 - val_auc: 0.7554

Epoch 00078: val_auc did not improve from 0.75876
Epoch 79/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.396
7 - auc: 0.7359 - val_loss: 0.4033 - val_auc: 0.7594

Epoch 00079: val_auc improved from 0.75876 to 0.75945, saving model to wei
ghts1it.best.hdf5
Epoch 80/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.397
7 - auc: 0.7357 - val_loss: 0.3942 - val_auc: 0.7583

Epoch 00080: val_auc did not improve from 0.75945
Epoch 81/100
61178/61178 [==============================] - 142s 2ms/step - loss: 0.396
2 - auc: 0.7376 - val_loss: 0.3949 - val_auc: 0.7573

Epoch 00081: val_auc did not improve from 0.75945
Epoch 82/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.396
8 - auc: 0.7384 - val_loss: 0.3988 - val_auc: 0.7558

Epoch 00082: val_auc did not improve from 0.75945
Epoch 83/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.397
6 - auc: 0.7366 - val_loss: 0.4161 - val_auc: 0.7554

Epoch 00083: val_auc did not improve from 0.75945
Epoch 84/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.397
5 - auc: 0.7389 - val_loss: 0.3969 - val_auc: 0.7551

Epoch 00084: val_auc did not improve from 0.75945
Epoch 85/100
61178/61178 [==============================] - 143s 2ms/step - loss: 0.397
2 - auc: 0.7401 - val_loss: 0.3970 - val_auc: 0.7570

Epoch 00085: val_auc did not improve from 0.75945
Epoch 86/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.396
7 - auc: 0.7388 - val_loss: 0.3918 - val_auc: 0.7535

Epoch 00086: val_auc did not improve from 0.75945
Epoch 87/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.396
6 - auc: 0.7405 - val_loss: 0.3952 - val_auc: 0.7565

Epoch 00087: val_auc did not improve from 0.75945
Epoch 88/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.395
3 - auc: 0.7426 - val_loss: 0.4007 - val_auc: 0.7565
```

```
Epoch 00088: val_auc did not improve from 0.75945
Epoch 89/100
61178/61178 [==============================] - 145s 2ms/step - loss: 0.397
0 - auc: 0.7404 - val_loss: 0.3926 - val_auc: 0.7569

Epoch 00089: val_auc did not improve from 0.75945
Epoch 90/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.397
2 - auc: 0.7393 - val_loss: 0.3992 - val_auc: 0.7562

Epoch 00090: val_auc did not improve from 0.75945
Epoch 91/100
61178/61178 [==============================] - 145s 2ms/step - loss: 0.395
5 - auc: 0.7444 - val_loss: 0.3956 - val_auc: 0.7573

Epoch 00091: val_auc did not improve from 0.75945
Epoch 92/100
61178/61178 [==============================] - 145s 2ms/step - loss: 0.395
3 - auc: 0.7448 - val_loss: 0.3954 - val_auc: 0.7550

Epoch 00092: val_auc did not improve from 0.75945
Epoch 93/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.395
0 - auc: 0.7454 - val_loss: 0.3978 - val_auc: 0.7533

Epoch 00093: val_auc did not improve from 0.75945
Epoch 94/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.397
2 - auc: 0.7411 - val_loss: 0.4067 - val_auc: 0.7539

Epoch 00094: val_auc did not improve from 0.75945
Epoch 95/100
61178/61178 [==============================] - 145s 2ms/step - loss: 0.396
4 - auc: 0.7435 - val_loss: 0.4028 - val_auc: 0.7543

Epoch 00095: val_auc did not improve from 0.75945
Epoch 96/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.395
2 - auc: 0.7466 - val_loss: 0.4010 - val_auc: 0.7525

Epoch 00096: val_auc did not improve from 0.75945
Epoch 97/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.396
5 - auc: 0.7477 - val_loss: 0.3983 - val_auc: 0.7528

Epoch 00097: val_auc did not improve from 0.75945
Epoch 98/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.397
4 - auc: 0.7452 - val_loss: 0.3949 - val_auc: 0.7519

Epoch 00098: val_auc did not improve from 0.75945
Epoch 99/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.395
0 - auc: 0.7504 - val_loss: 0.3974 - val_auc: 0.7560

Epoch 00099: val_auc did not improve from 0.75945
Epoch 100/100
61178/61178 [==============================] - 144s 2ms/step - loss: 0.396
2 - auc: 0.7479 - val_loss: 0.3970 - val_auc: 0.7550

Epoch 00100: val_auc did not improve from 0.75945
```

In [39]:

```
#2nd cycle of epoch size=50
history1=model3.fit([train_essay_padded,X_train_rem], y_train, nb_epoch=50,verbose=1,ba
tch_size=500,
          validation_data=([cv_essay_padded,X_cv_rem],y_cv),
          callbacks =callbacks_list,class_weight = class_wts)
```

```
Train on 61178 samples, validate on 15295 samples
Epoch 1/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.405
5 - auc: 0.7276 - val_loss: 0.3900 - val_auc: 0.7576

Epoch 00001: val_auc improved from -inf to 0.75761, saving model to weight
s2it.best.hdf5
Epoch 2/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.403
9 - auc: 0.7268 - val_loss: 0.3979 - val_auc: 0.7540

Epoch 00002: val_auc did not improve from 0.75761
Epoch 3/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.402
9 - auc: 0.7277 - val_loss: 0.3949 - val_auc: 0.7554

Epoch 00003: val_auc did not improve from 0.75761
Epoch 4/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.402
1 - auc: 0.7267 - val_loss: 0.3995 - val_auc: 0.7578

Epoch 00004: val_auc improved from 0.75761 to 0.75777, saving model to wei
ghts2it.best.hdf5
Epoch 5/50
61178/61178 [==============================] - 145s 2ms/step - loss: 0.401
7 - auc: 0.7265 - val_loss: 0.3936 - val_auc: 0.7559

Epoch 00005: val_auc did not improve from 0.75777
Epoch 6/50
61178/61178 [==============================] - 145s 2ms/step - loss: 0.401
5 - auc: 0.7278 - val_loss: 0.3928 - val_auc: 0.7558

Epoch 00006: val_auc did not improve from 0.75777
Epoch 7/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.401
5 - auc: 0.7281 - val_loss: 0.3953 - val_auc: 0.7576

Epoch 00007: val_auc did not improve from 0.75777
Epoch 8/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.400
8 - auc: 0.7274 - val_loss: 0.4067 - val_auc: 0.7557

Epoch 00008: val_auc did not improve from 0.75777
Epoch 9/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.400
9 - auc: 0.7287 - val_loss: 0.3903 - val_auc: 0.7571

Epoch 00009: val_auc did not improve from 0.75777
Epoch 10/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.400
3 - auc: 0.7296 - val_loss: 0.3932 - val_auc: 0.7567

Epoch 00010: val_auc did not improve from 0.75777
Epoch 11/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.401
7 - auc: 0.7281 - val_loss: 0.3956 - val_auc: 0.7582

Epoch 00011: val_auc improved from 0.75777 to 0.75820, saving model to wei
ghts2it.best.hdf5
Epoch 12/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.400
```

```
5 - auc: 0.7279 - val_loss: 0.3934 - val_auc: 0.7558

Epoch 00012: val_auc did not improve from 0.75820
Epoch 13/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.400
6 - auc: 0.7271 - val_loss: 0.4012 - val_auc: 0.7550

Epoch 00013: val_auc did not improve from 0.75820
Epoch 14/50
61178/61178 [==============================] - 148s 2ms/step - loss: 0.401
6 - auc: 0.7272 - val_loss: 0.4001 - val_auc: 0.7566

Epoch 00014: val_auc did not improve from 0.75820
Epoch 15/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.400
7 - auc: 0.7287 - val_loss: 0.3946 - val_auc: 0.7564

Epoch 00015: val_auc did not improve from 0.75820
Epoch 16/50
61178/61178 [==============================] - 148s 2ms/step - loss: 0.400
1 - auc: 0.7309 - val_loss: 0.3920 - val_auc: 0.7563

Epoch 00016: val_auc did not improve from 0.75820
Epoch 17/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.400
1 - auc: 0.7309 - val_loss: 0.3935 - val_auc: 0.7555

Epoch 00017: val_auc did not improve from 0.75820
Epoch 18/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.399
5 - auc: 0.7330 - val_loss: 0.3918 - val_auc: 0.7540

Epoch 00018: val_auc did not improve from 0.75820
Epoch 19/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.398
0 - auc: 0.7343 - val_loss: 0.3971 - val_auc: 0.7551

Epoch 00019: val_auc did not improve from 0.75820
Epoch 20/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.398
3 - auc: 0.7318 - val_loss: 0.3989 - val_auc: 0.7544

Epoch 00020: val_auc did not improve from 0.75820
Epoch 21/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.398
3 - auc: 0.7326 - val_loss: 0.4054 - val_auc: 0.7562

Epoch 00021: val_auc did not improve from 0.75820
Epoch 22/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.398
7 - auc: 0.7312 - val_loss: 0.3936 - val_auc: 0.7555

Epoch 00022: val_auc did not improve from 0.75820
Epoch 23/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.399
2 - auc: 0.7310 - val_loss: 0.3957 - val_auc: 0.7560

Epoch 00023: val_auc did not improve from 0.75820
Epoch 24/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.399
9 - auc: 0.7325 - val_loss: 0.4001 - val_auc: 0.7560
```

```
Epoch 00024: val_auc did not improve from 0.75820
Epoch 25/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.399
2 - auc: 0.7338 - val_loss: 0.3994 - val_auc: 0.7566

Epoch 00025: val_auc did not improve from 0.75820
Epoch 26/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.398
4 - auc: 0.7364 - val_loss: 0.3970 - val_auc: 0.7587

Epoch 00026: val_auc improved from 0.75820 to 0.75865, saving model to wei
ghts2it.best.hdf5
Epoch 27/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.399
1 - auc: 0.7341 - val_loss: 0.3912 - val_auc: 0.7555

Epoch 00027: val_auc did not improve from 0.75865
Epoch 28/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.397
7 - auc: 0.7358 - val_loss: 0.3926 - val_auc: 0.7566

Epoch 00028: val_auc did not improve from 0.75865
Epoch 29/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.399
0 - auc: 0.7357 - val_loss: 0.3940 - val_auc: 0.7576

Epoch 00029: val_auc did not improve from 0.75865
Epoch 30/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.397
6 - auc: 0.7370 - val_loss: 0.3969 - val_auc: 0.7587

Epoch 00030: val_auc improved from 0.75865 to 0.75871, saving model to wei
ghts2it.best.hdf5
Epoch 31/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.397
5 - auc: 0.7368 - val_loss: 0.3919 - val_auc: 0.7574

Epoch 00031: val_auc did not improve from 0.75871
Epoch 32/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.396
0 - auc: 0.7384 - val_loss: 0.4076 - val_auc: 0.7576

Epoch 00032: val_auc did not improve from 0.75871
Epoch 33/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.397
7 - auc: 0.7359 - val_loss: 0.3994 - val_auc: 0.7562

Epoch 00033: val_auc did not improve from 0.75871
Epoch 34/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.397
0 - auc: 0.7381 - val_loss: 0.3925 - val_auc: 0.7568

Epoch 00034: val_auc did not improve from 0.75871
Epoch 35/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.397
6 - auc: 0.7384 - val_loss: 0.3996 - val_auc: 0.7575

Epoch 00035: val_auc did not improve from 0.75871
Epoch 36/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.397
```

```
2 - auc: 0.7371 - val_loss: 0.3942 - val_auc: 0.7553

Epoch 00036: val_auc did not improve from 0.75871
Epoch 37/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.398
1 - auc: 0.7379 - val_loss: 0.3969 - val_auc: 0.7588

Epoch 00037: val_auc improved from 0.75871 to 0.75883, saving model to wei
ghts2it.best.hdf5
Epoch 38/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.396
4 - auc: 0.7397 - val_loss: 0.3948 - val_auc: 0.7588

Epoch 00038: val_auc did not improve from 0.75883
Epoch 39/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.396
1 - auc: 0.7408 - val_loss: 0.3971 - val_auc: 0.7548

Epoch 00039: val_auc did not improve from 0.75883
Epoch 40/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.397
9 - auc: 0.7365 - val_loss: 0.3938 - val_auc: 0.7563

Epoch 00040: val_auc did not improve from 0.75883
Epoch 41/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.395
6 - auc: 0.7426 - val_loss: 0.3912 - val_auc: 0.7576

Epoch 00041: val_auc did not improve from 0.75883
Epoch 42/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.395
0 - auc: 0.7423 - val_loss: 0.4042 - val_auc: 0.7583

Epoch 00042: val_auc did not improve from 0.75883
Epoch 43/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.397
1 - auc: 0.7400 - val_loss: 0.3902 - val_auc: 0.7588

Epoch 00043: val_auc did not improve from 0.75883
Epoch 44/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.396
4 - auc: 0.7417 - val_loss: 0.3935 - val_auc: 0.7575

Epoch 00044: val_auc did not improve from 0.75883
Epoch 45/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.395
6 - auc: 0.7413 - val_loss: 0.3887 - val_auc: 0.7554

Epoch 00045: val_auc did not improve from 0.75883
Epoch 46/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.394
7 - auc: 0.7432 - val_loss: 0.3959 - val_auc: 0.7565

Epoch 00046: val_auc did not improve from 0.75883
Epoch 47/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.396
3 - auc: 0.7438 - val_loss: 0.4000 - val_auc: 0.7551

Epoch 00047: val_auc did not improve from 0.75883
Epoch 48/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.396
```

```
4 - auc: 0.7408 - val_loss: 0.3933 - val_auc: 0.7567

Epoch 00048: val_auc did not improve from 0.75883
Epoch 49/50
61178/61178 [==============================] - 146s 2ms/step - loss: 0.395
6 - auc: 0.7421 - val_loss: 0.4050 - val_auc: 0.7575

Epoch 00049: val_auc did not improve from 0.75883
Epoch 50/50
61178/61178 [==============================] - 147s 2ms/step - loss: 0.395
1 - auc: 0.7455 - val_loss: 0.3934 - val_auc: 0.7574

Epoch 00050: val_auc did not improve from 0.75883
```

- The second cycle did not improve the AUC.

3.10 Plots on training results

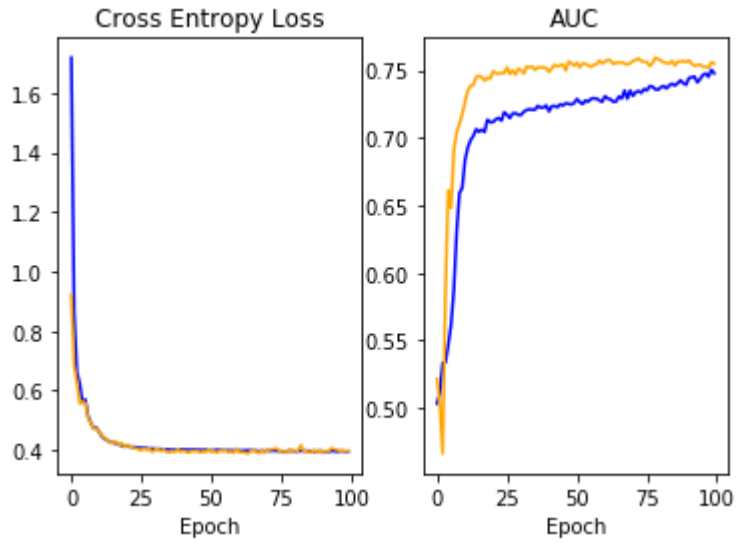
3.10.1 Loss & AUC plots

In [0]:

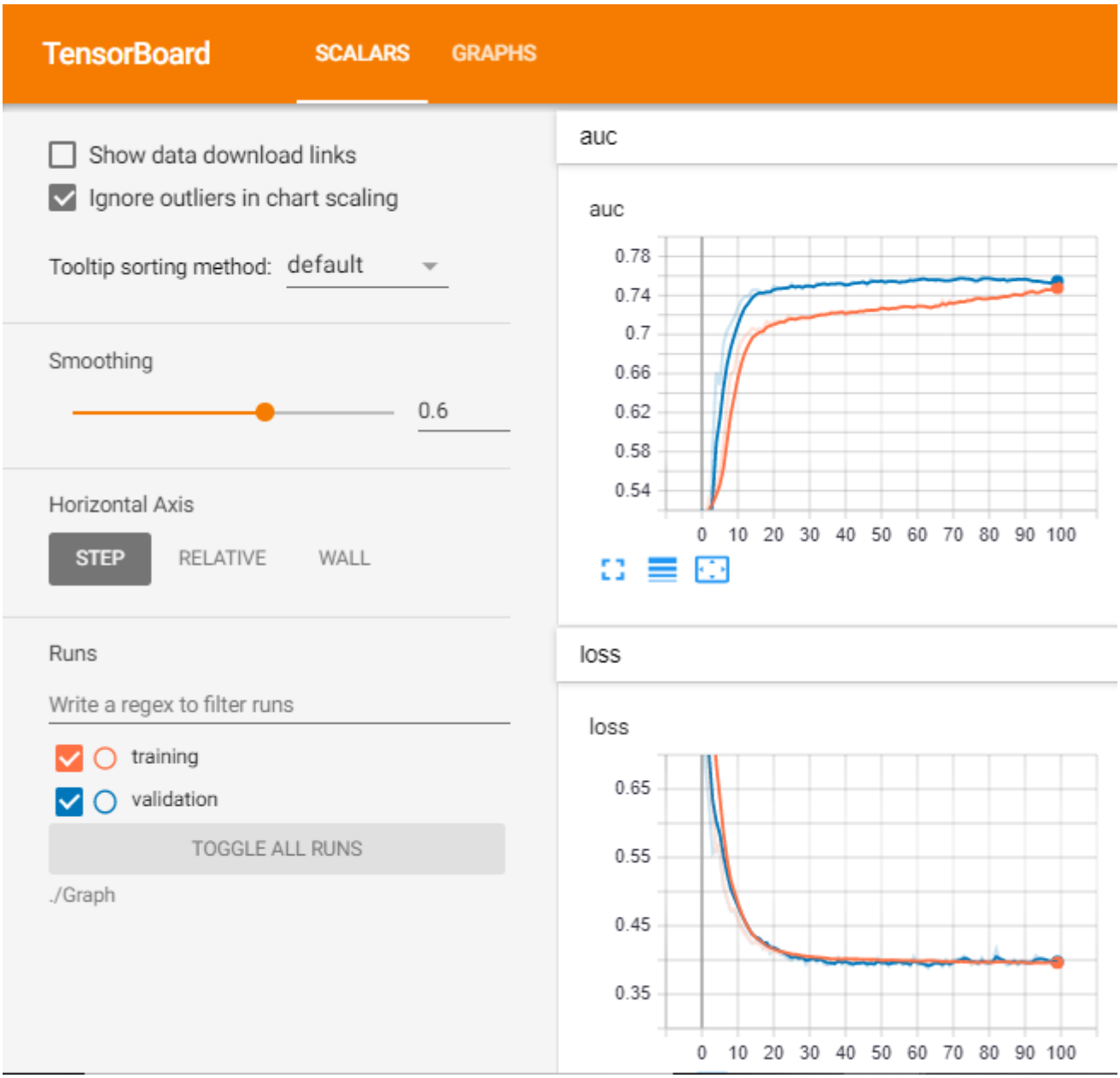
```
# function to plot epoch vs loss & epoch vs AUC
%matplotlib notebook
%matplotlib inline
from matplotlib import pyplot
def plot(history):
    # plot loss
    pyplot.subplot(121)
    pyplot.title('Cross Entropy Loss')
    pyplot.xlabel('Epoch')
    pyplot.plot(history.history['loss'], color='blue', label='train')
    pyplot.plot(history.history['val_loss'], color='orange', label='CV')
    # plot auc
    pyplot.subplot(122)
    pyplot.title('AUC')
    pyplot.xlabel('Epoch')
    pyplot.plot(history.history['auc'], color='blue', label='train')
    pyplot.plot(history.history['val_auc'], color='orange', label='CV')
```

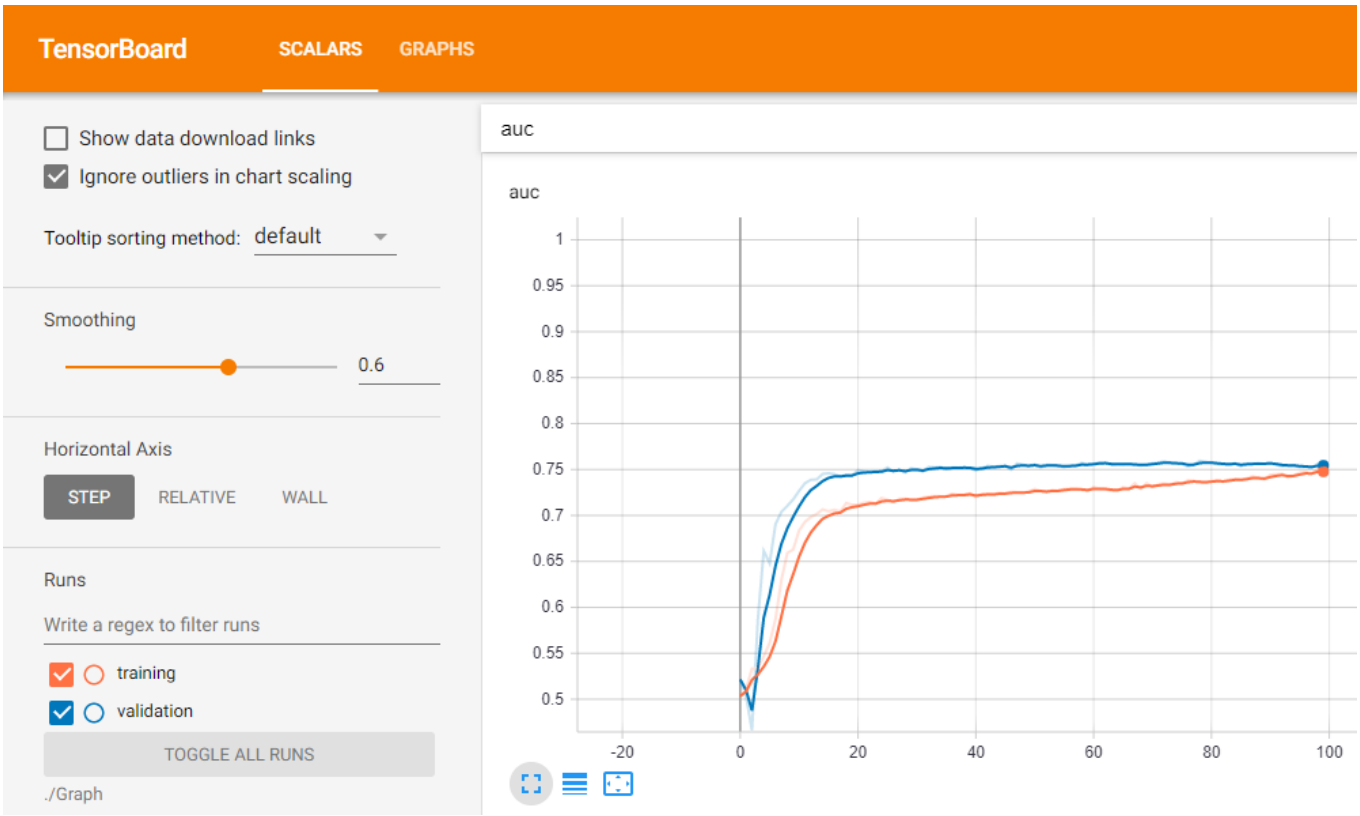
In [20]:

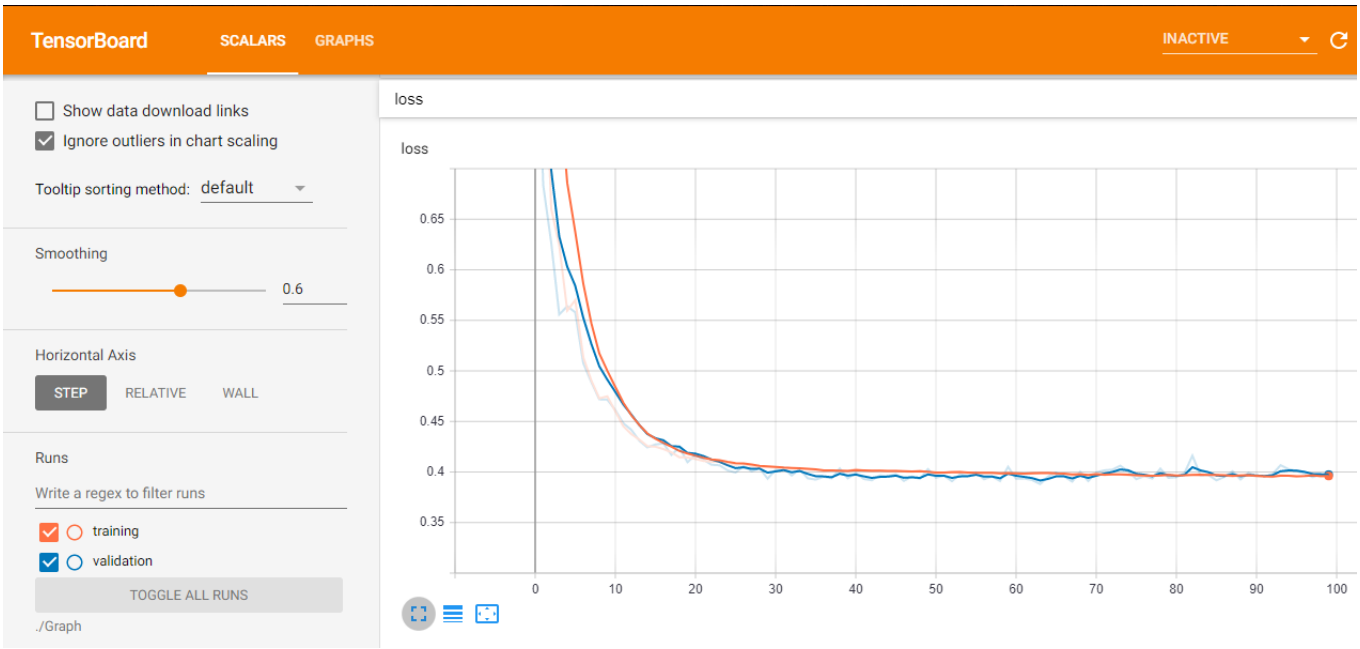
```
plot(history)
```



3.10.2 Tensorboard images


TensorBoard
SCALARS
GRAPHS
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
training
validation
TOGGLE ALL RUNS
./Graph
auc
auc
0.78
0.74
0.7
0.66
0.62
0.58
0.54
0 10 20 30 40 50 60 70 80 90 100
loss
loss
0.65
0.55
0.45
0.35
0 10 20 30 40 50 60 70 80 90 100





3.11 Results & Model Testing

In [22]:

```
train_results = model3.evaluate([train_essay_padded,X_train_rem],y_train,
                                verbose=1,batch_size=500)
print('Train Loss: ',train_results[0])
print('Train AUC: ',train_results[1])
```

```
61178/61178 [==============================] - 51s 836us/step
Train Loss:  0.381318802457718
Train AUC:  0.7961345330957783
```

In [23]:

```
cv_results = model3.evaluate([cv_essay_padded,X_cv_rem],y_cv,verbose=1,batch_size=500)
print('CV Loss: ',cv_results[0])
print('CV AUC: ',cv_results[1])
```

```
15295/15295 [==============================] - 13s 840us/step
CV Loss:  0.3969620003560744
CV AUC:  0.7550055366514866
```

In [21]:

```
test_results = model3.evaluate([test_essay_padded,X_test_rem],y_test,verbose=1,
                               batch_size=500)
print('Test Loss: ',test_results[0])
print('Test AUC: ',test_results[1])
```

```
32775/32775 [==============================] - 28s 845us/step
Test Loss:  0.4020882885161472
Test AUC:  0.7411375541345114
```

4.0 Summary

4.1 Loss

In [1]:

```
#Ref: http://zetcode.com/python/prettytable/

from prettytable import PrettyTable
x=PrettyTable()
x.field_names=["Model","Train loss","CV loss","Test loss"]

x.add_row(["Model-1","0.3868","0.3915","0.3918"])
x.add_row(["Model-2","0.3696","0.3835","0.3816"])
x.add_row(["Model-3","0.3813","0.3969","0.4020"])

print(x)
```

```
+---------+------------+---------+-----------+
|  Model  | Train loss | CV loss | Test loss |
+---------+------------+---------+-----------+
| Model-1 |   0.3868   |  0.3915 |   0.3918  |
| Model-2 |   0.3696   |  0.3835 |   0.3816  |
| Model-3 |   0.3813   |  0.3969 |   0.4020  |
+---------+------------+---------+-----------+
```

4.2 AUC

In [2]:

```
from prettytable import PrettyTable
x=PrettyTable()
x.field_names=["Model","Train AUC","CV AUC","Test AUC"]

x.add_row(["Model-1","0.76","0.75","0.75"])
x.add_row(["Model-2","0.78","0.75","0.75"])
x.add_row(["Model-3","0.80","0.76","0.75"])

print(x)
```

```
+---------+-----------+--------+----------+
|  Model  | Train AUC | CV AUC | Test AUC |
+---------+-----------+--------+----------+
| Model-1 |    0.76   |  0.75  |   0.75   |
| Model-2 |    0.78   |  0.75  |   0.75   |
| Model-3 |    0.80   |  0.76  |   0.75   |
+---------+-----------+--------+----------+
```