Assignment : 14

In [1]:

```
%tensorflow_version 1.x
from keras.models import Sequential
import numpy as np
import keras
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical
from keras.layers import Dense, Input,
GlobalMaxPooling1D,Flatten,LSTM,concatenate,Dropout,CuDNNLSTM,SpatialDropout1D
from keras.layers import Conv1D, MaxPooling1D, Embedding,InputLayer,BatchNormalization
from keras.models import Model
from keras.initializers import Constant
from keras.regularizers import l2
#import tensorflow as tf
from keras import backend as K
import pandas as pd
from keras.layers import ReLU
```

```
TensorFlow 1.x selected.
```

```
Using TensorFlow backend.
```

In [0]:

```
from __future__ import print_function
import tqdm
import pickle
import pandas as pd
import numpy as np
import os
import sys
from keras.models import Sequential
from keras.layers import Dense,Input
from keras.layers import LSTM
from keras.layers.embeddings import Embedding
from keras.preprocessing import sequence
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
```

In [0]:

```
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import roc_auc_score
from scipy.sparse import hstack
from numpy import zeros
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Input
from keras.layers import Flatten
from keras.layers import Embedding
from keras.layers import LSTM,Bidirectional
from keras.layers.core import Dense, Dropout
from keras.models import Model, load_model
from keras.layers.normalization import BatchNormalization
from keras.callbacks import ReduceLROnPlateau
```

In [4]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6
qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%
b&response_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2
www.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly
ttps%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:
··········
Mounted at /content/drive
```

In [0]:

```
project_data=pd.read_csv("/content/drive/My Drive/prs.csv")
```

In [6]:

```
from sklearn.model_selection import train_test_split
# split the data set into train and test respectively 80% and 20%
y=project_data['project_is_approved']
project_data.drop(['project_is_approved'],axis=1, inplace=True)
x=project_data
X_temp,X_test,Y_temp,Y_test=train_test_split(x,y,test_size=0.2)
# split the data set into train and cv respectively 60% and 20%
X_train,X_cv,Y_train,Y_cv=train_test_split(X_temp,Y_temp,test_size=0.2)
print("Shape of Train data set X={} Y={}".format(X_train.shape,Y_train.shape))
print("Shape of Test data set X={} Y={}".format(X_test.shape,Y_test.shape))
print("Shape of CV data set X={} Y={}".format(X_cv.shape,Y_cv.shape))
```

```
Shape of Train data set X=(69918, 9) Y=(69918,)
Shape of Test data set X=(21850, 9) Y=(21850,)
Shape of CV data set X=(17480, 9) Y=(17480,)
```

In [0]:

```
y_train = to_categorical(Y_train)
y_cv = to_categorical(Y_cv)
y_test = to_categorical(Y_test)
```

In [0]:

```
len_essay=[]
for sentance1 in (X_train['essay'].values):
    len_essay.append(len(sentance1.split()))
num_essay=np.array(len_essay)
max_length=num_essay.max()
```

In [0]:

```
#text
tokenizer = Tokenizer()
tokenizer.fit_on_texts(X_train['essay'])
train_encoded_essays = tokenizer.texts_to_sequences(X_train['essay'])
train_padded_essays = pad_sequences(train_encoded_essays, maxlen=max_length,padding='post')

test_encoded_essays = tokenizer.texts_to_sequences(X_test['essay'])
test_padded_essays = pad_sequences(test_encoded_essays, maxlen=max_length,padding='post')


cv_encoded_essays = tokenizer.texts_to_sequences(X_cv['essay'])
cv_padded_essays = pad_sequences(cv_encoded_essays, maxlen=max_length,padding='post')

vocab_size = len(tokenizer.word_index) + 1
```

In [10]:

```
from sklearn.preprocessing import StandardScaler
#essay_stand = StandardScaler().fit(train_padded_essays)
tr_text__nor = train_padded_essays
cv_text__nor = cv_padded_essays
te_text__nor = test_padded_essays
print(tr_text__nor.shape)
```

```
(69918, 315)
```

In [0]:

```
#loading golve model
import pickle
with open('/content/drive/My Drive/glove_vectors', 'rb') as f:
    glove = pickle.load(f)
```

In [0]:

```
word_vector=np.zeros((vocab_size, 300))
for word, i in tokenizer.word_index.items():
    vector=glove.get(word)
    if vector is not None:
        word_vector[i]=vector
```

In [13]:

```
# check this one: https://www.youtube.com/watch?v=0HOqOcln3Z4&t=530s
# standardization sklearn: https://scikit-
learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html


# price_standardized = standardScalar.fit(X_train['price'].values)
# this will rise the error
# ValueError: Expected 2D array, got 1D array instead: array=[725.05 213.03 329.   ... 399.   287.
73   5.5 ].
# Reshape your data either using array.reshape(-1, 1)

price_scalar = StandardScaler()
tr_price_standardized=price_scalar.fit_transform(X_train['price'].values.reshape(-1,1)) # finding t
he mean and standard deviation of this data
print(f"Mean : {price_scalar.mean_[0]}, Standard deviation : {np.sqrt(price_scalar.var_[0])}")

# Now standardize the data with above maen and variance.
cv_price_standardized = price_scalar.transform(X_cv['price'].values.reshape(-1, 1))
te_price_standardized = price_scalar.transform(X_test['price'].values.reshape(-1, 1))
print(tr_price_standardized.shape)
```

```
Mean : 298.3500094396293, Standard deviation : 369.7265369517227
(69918, 1)
```

In [14]:

```
#teacher_number_of_previously_posted_projects
teacher_number_of_previously_posted_projects_scalar = StandardScaler()
tr_teacher_number_of_previously_posted_projects_standardized=teacher_number_of_previously_posted_pr
jects_scalar.fit_transform(X_train['teacher_number_of_previously_posted_projects'].values.reshape(
-1,1)) # finding the mean and standard deviation of this data
print(f"Mean : {teacher_number_of_previously_posted_projects_scalar.mean_[0]}, Standard deviation
: {np.sqrt(teacher_number_of_previously_posted_projects_scalar.var_[0])}")

# Now standardize the data with above maen and variance.
cv_teacher_number_of_previously_posted_projects_standardized =
teacher_number_of_previously_posted_projects_scalar.transform(X_cv['teacher_number_of_previously_po
ted_projects'].values.reshape(-1, 1))
te_teacher_number_of_previously_posted_projects_standardized =
teacher_number_of_previously_posted_projects_scalar.transform(X_test['teacher_number_of_previously_
osted_projects'].values.reshape(-1, 1))
print("\nShape of matrix after column standardization for
'teacher_number_of_previously_posted_projects'\nTrain data-{},\nCV data\t-{}\nTest data-{}".format
(tr_teacher_number_of_previously_posted_projects_standardized.shape,cv_teacher_number_of_previously
posted_projects_standardized.shape,te_teacher_number_of_previously_posted_projects_standardized.sh
ape))
```

```
Mean : 11.191495752166825, Standard deviation : 27.829357536069097

Shape of matrix after column standardization for 'teacher_number_of_previously_posted_projects'
Train data-(69918, 1),
```

```
CV data -(17480, 1)
Test data-(21850, 1)
```

In [0]:

```
#https://stackoverflow.com/posts/51734992/revisions
import tensorflow as tf
from sklearn.metrics import roc_auc_score

def auroc(y_true, y_pred):
    return tf.py_func(roc_auc_score, (y_true, y_pred), tf.double)
```

In [16]:

```
from sklearn import preprocessing
vocab_size_cat=len(X_train['clean_categories'].unique())+1
print(vocab_size_cat)
le = preprocessing.LabelEncoder()
le.fit(X_train['clean_categories'])
tr_categories_le=le.transform(X_train['clean_categories'])
X_test["clean_categories"] = X_test["clean_categories"].map(lambda s: ' ' if s not in le.classes_ e
lse s)
le.classes_ = np.append(le.classes_, ' ')
te_categories_le=le.transform(X_test['clean_categories'].values)
X_cv["clean_categories"] = X_cv["clean_categories"].map(lambda s: ' ' if s not in le.classes_ else
s)
cv_categories_le=le.transform(X_cv['clean_categories'].values)
```

```
52
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  import sys
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
```

In [17]:

```
from sklearn import preprocessing
vocab_size_state=len(X_train['school_state'].unique())+1
print(vocab_size_state)
le = preprocessing.LabelEncoder()
le.fit(X_train['school_state'])
tr_school_state_le=le.transform(X_train['school_state'])
X_test["school_state"] = X_test["school_state"].map(lambda s: ' ' if s not in le.classes_ else s)
le.classes_ = np.append(le.classes_, ' ')
te_school_state_le=le.transform(X_test['school_state'].values)
X_cv["school_state"] = X_cv["school_state"].map(lambda s: ' ' if s not in le.classes_ else s)
cv_school_state_le=le.transform(X_cv['school_state'].values)
```

```
52
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  import sys
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
```

In [18]:

```
from sklearn import preprocessing
vocab_size_te_pr=len(X_train['teacher_prefix'].unique())+1
print(vocab_size_te_pr)
le = preprocessing.LabelEncoder()
le.fit(X_train['teacher_prefix'])
tr_teacher_prefix_le=le.transform(X_train['teacher_prefix'])
X_test["teacher_prefix"] = X_test["teacher_prefix"].map(lambda s: ' ' if s not in le.classes_ else
s)
le.classes_ = np.append(le.classes_, ' ')
te_teacher_prefix_le=le.transform(X_test['teacher_prefix'].values)
X_cv["teacher_prefix"] = X_cv["teacher_prefix"].map(lambda s: ' ' if s not in le.classes_ else s)
cv_teacher_prefix_le=le.transform(X_cv['teacher_prefix'].values)
```

```
6
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  import sys
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
```

In [19]:

```
from sklearn import preprocessing
vocab_size_gra_cat=len(X_train['project_grade_category'].unique())+1
print(vocab_size_gra_cat)
le = preprocessing.LabelEncoder()
le.fit(X_train['project_grade_category'])
tr_project_grade_category_le=le.transform(X_train['project_grade_category'])
X_test["project_grade_category"] = X_test["project_grade_category"].map(lambda s: ' ' if s not in l
e.classes_ else s)
le.classes_ = np.append(le.classes_, ' ')
te_project_grade_category_le=le.transform(X_test['project_grade_category'].values)
X_cv["project_grade_category"] = X_cv["project_grade_category"].map(lambda s: ' ' if s not in le.cl
asses_ else s)
cv_project_grade_category_le=le.transform(X_cv['project_grade_category'].values)
```

```
5
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  import sys
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
```

In [20]:

```
from sklearn import preprocessing
vocab_size_sub_cat=len(X_train['clean_subcategories'].unique())+1
print(vocab_size_sub_cat)
le = preprocessing.LabelEncoder()
le.fit(X_train['clean_subcategories'])
tr_clean_subcategories_le=le.transform(X_train['clean_subcategories'])
X_test["clean_subcategories"] = X_test["clean_subcategories"].map(lambda s: ' ' if s not in le.clas
ses_ else s)
le.classes_ = np.append(le.classes_, ' ')
te_clean_subcategories_le=le.transform(X_test['clean_subcategories'].values)
X_cv["clean_subcategories"] = X_cv["clean_subcategories"].map(lambda s: ' ' if s not in le.classes_
else s)
cv_clean_subcategories_le=le.transform(X_cv['clean_subcategories'].values)
```

```
394
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  import sys
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
```

In [21]:

```
from numpy import hstack
tr_X_num=
hstack((tr_price_standardized,tr_teacher_number_of_previously_posted_projects_standardized))
cv_X_num=
hstack((cv_price_standardized,cv_teacher_number_of_previously_posted_projects_standardized))
te_X_num=
hstack((te_price_standardized,te_teacher_number_of_previously_posted_projects_standardized))
print(tr_X_num.shape)
print(cv_X_num.shape)
print(te_X_num.shape)
```

```
(69918, 2)
(17480, 2)
(21850, 2)
```

Model-1

In [0]:

```
#model one hot
e=64
K.clear_session()
#model = Sequential()
#total text data
input_total_text=Input(shape=(max_length,),name="input_seq_total_text_dat")
emedding_layer_total_text = Embedding(vocab_size,300, weights=[word_vector],
input_length=max_length, trainable=False,name="emb_text_data")(input_total_text)
lstm_layer_total_text=LSTM(64,kernel_initializer='glorot_normal',recurrent_dropout=0.5,return_seque
nces=True)(emedding_layer_total_text)
#lstm_layer_total_text= ReLU()(lstm_layer_total_text)
flat_layer_total_text=Flatten()(lstm_layer_total_text)

#project state
input_state=Input(shape=(1,),name="input_school_state")
emedding_layer_state = Embedding(vocab_size_state,26,name="emb_state_data")(input_state)
flat_layer_state=Flatten()(emedding_layer_state)
```

```
#project grade
input_grade=Input(shape=(1,),name="input_project_grade_category")
emedding_layer_grade = Embedding(vocab_size_gra_cat,3,name="emb_pgc_data")(input_grade)
flat_layer_grade=Flatten()(emedding_layer_grade)

#project categiries
input_categories=Input(shape=(1,),name="input_clean_categories")
emedding_layer_cat = Embedding(vocab_size_cat,26,name="emb_clean_categoriesa")(input_categories)
flat_layer_cat=Flatten()(emedding_layer_cat)

#project subcategiries
input_subcategories=Input(shape=(1,),name="input_clean_subcategories")
emedding_layer_subcat = Embedding(vocab_size_sub_cat,50,name="emb_clean_subcategoriesa")
(input_subcategories)
flat_layer_subcat=Flatten()(emedding_layer_subcat)

#Project teacher_prefix
input_teach_prefix=Input(shape=(1,),name="input_teach_prefix")
emedding_layer_teach_prefix = Embedding(vocab_size_te_pr,3,name="emb_teach_prefix_data")
(input_teach_prefix)
flat_layer_teach_prefix=Flatten()(emedding_layer_teach_prefix)

#number data price and teacher_number_of_previously_posted_projects
input_num=Input(shape=(2,),name="input_price_teacher_number_posted_project")
dense_layer_price_num_project=Dense(64,
activation='sigmoid',name="Dense_rem_input",kernel_initializer='he_normal',kernel_regularizer=l2(0
.001))(input_num)

concat_layer=concatenate(inputs=[flat_layer_total_text,flat_layer_cat,flat_layer_subcat,flat_layer_
state,flat_layer_teach_prefix,dense_layer_price_num_project],name="concatenate")
#BN_1=BatchNormalization()(concat_layer)
#dense layer 1
dense_layer_after_concat=Dense(32,name="Dense_layer_after_concat",kernel_initializer='he_normal',a
ctivation='relu',kernel_regularizer=l2(0.001))(concat_layer)
#dense_layer_after_concat=ReLU()(dense_layer_after_concat)
#dropout layer 1
dropout_1=Dropout(0.8,name="Dropout_1")(dense_layer_after_concat)
#dense layer 2
dense_layer_after_concat_2=Dense(16,name="Dense_layer_after_concat_2",kernel_initializer='he_normal
',activation='relu',kernel_regularizer=l2(0.001))(dense_layer_after_concat)
#dense_layer_after_concat_2=ReLU()(dense_layer_after_concat_2)
#dropout layer 2
dropout_2=Dropout(0.8,name="Dropout_2")(dense_layer_after_concat_2)
#dense layer 3
dense_layer_after_concat_3=Dense(8,name="Dense_layer_after_concat_3",kernel_initializer='he_normal'
,activation='relu',kernel_regularizer=l2(0.001))(dense_layer_after_concat_2)
#dense_layer_after_concat_3=ReLU()(dense_layer_after_concat_3)
dropout_3=Dropout(0.8,name="Dropout_3")(dense_layer_after_concat_3)

dense_layer_after_concat_4=Dense(16,name="Dense_layer_after_concat_4",kernel_initializer='he_normal
',activation='relu',kernel_regularizer=l2(0.001))(dense_layer_after_concat_3)
#dense_layer_after_concat_4=ReLU()(dense_layer_after_concat_4)

output_layer=Dense(2, activation='softmax',kernel_initializer='glorot_uniform')
(dense_layer_after_concat_4)

model=Model(inputs=[input_total_text,input_state,input_grade,input_categories,input_subcategories,i
nput_teach_prefix,input_num],outputs=output_layer)
```

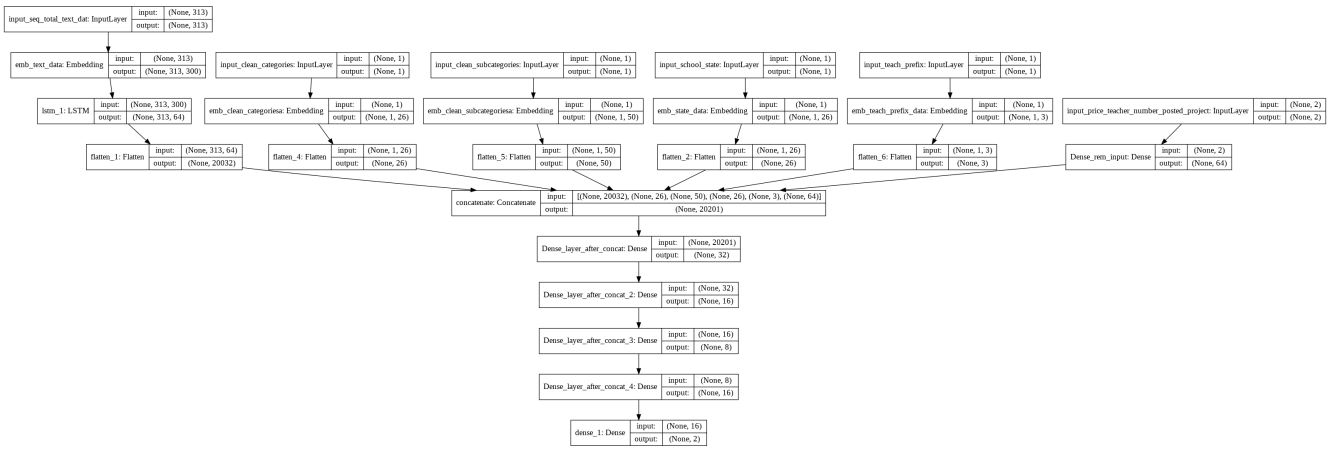
```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:107: The name tf.reset_default_graph is deprecated. P
lease use tf.compat.v1.reset_default_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:111: The name tf.placeholder_with_default is
deprecated. Please use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:66: The name tf.get_default_graph is deprecated. Plea
se use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:541: The name tf.placeholder is deprecated. Please us
e tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
```

```
packages/keras/backend/tensorflow_backend.py:4432: The name tf.random_uniform is deprecated. Pleas
e use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:190: The name tf.get_default_session is deprecated. P
lease use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:197: The name tf.ConfigProto is deprecated. Please us
e tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:203: The name tf.Session is deprecated. Please use tf
.compat.v1.Session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:207: The name tf.global_variables is deprecated. Plea
se use tf.compat.v1.global_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:216: The name tf.is_variable_initialized is
deprecated. Please use tf.compat.v1.is_variable_initialized instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:223: The name tf.variables_initializer is deprecated.
Please use tf.compat.v1.variables_initializer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:4479: The name tf.truncated_normal is deprecated. Ple
ase use tf.random.truncated_normal instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3733: calling dropout (from
tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future
version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
WARNING:tensorflow:Large dropout rate: 0.8 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate
instead of keep_prob. Please ensure that this is intended.
WARNING:tensorflow:Large dropout rate: 0.8 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate
instead of keep_prob. Please ensure that this is intended.
WARNING:tensorflow:Large dropout rate: 0.8 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate
instead of keep_prob. Please ensure that this is intended.
```

In [0]:

```
#https://machinelearningmastery.com/visualize-deep-learning-neural-network-model-keras/
from keras.utils.vis_utils import plot_model
plot_model(model, to_file='/content/drive/My Drive/model1.png', show_shapes=True, show_layer_names
=True)
```

Out[0]:



In [0]:

```
# summarize the model
print(model.summary())
```

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_seq_total_text_dat (Input (None, 313)          0
__________________________________________________________________________________________________
emb_text_data (Embedding)       (None, 313, 300)     14102100    input_seq_total_text_dat[0][0]
__________________________________________________________________________________________________
input_clean_categories (InputLa (None, 1)            0
__________________________________________________________________________________________________
input_clean_subcategories (Inpu (None, 1)            0
__________________________________________________________________________________________________
input_school_state (InputLayer) (None, 1)            0
__________________________________________________________________________________________________
input_teach_prefix (InputLayer) (None, 1)            0
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 313, 64)      93440       emb_text_data[0][0]
__________________________________________________________________________________________________
emb_clean_categoriesa (Embeddin (None, 1, 26)        1352        input_clean_categories[0][0]
__________________________________________________________________________________________________
emb_clean_subcategoriesa (Embed (None, 1, 50)        19650       input_clean_subcategories[0][0]
__________________________________________________________________________________________________
emb_state_data (Embedding)      (None, 1, 26)        1352        input_school_state[0][0]
__________________________________________________________________________________________________
emb_teach_prefix_data (Embeddin (None, 1, 3)         18          input_teach_prefix[0][0]
__________________________________________________________________________________________________
input_price_teacher_number_post (None, 2)            0
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 20032)        0           lstm_1[0][0]
__________________________________________________________________________________________________
flatten_4 (Flatten)             (None, 26)           0           emb_clean_categoriesa[0][0]
__________________________________________________________________________________________________
flatten_5 (Flatten)             (None, 50)           0           emb_clean_subcategoriesa[0][0]
__________________________________________________________________________________________________
flatten_2 (Flatten)             (None, 26)           0           emb_state_data[0][0]
__________________________________________________________________________________________________
flatten_6 (Flatten)             (None, 3)            0           emb_teach_prefix_data[0][0]
__________________________________________________________________________________________________
Dense_rem_input (Dense)         (None, 64)           192         input_price_teacher_number_posted
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 20201)        0           flatten_1[0][0]
                                                                 flatten_4[0][0]
                                                                 flatten_5[0][0]
                                                                 flatten_2[0][0]
                                                                 flatten_6[0][0]
                                                                 Dense_rem_input[0][0]
__________________________________________________________________________________________________
Dense_layer_after_concat (Dense (None, 32)           646464      concatenate[0][0]
__________________________________________________________________________________________________
Dense_layer_after_concat_2 (Den (None, 16)           528         Dense_layer_after_concat[0][0]
__________________________________________________________________________________________________
Dense_layer_after_concat_3 (Den (None, 8)            136         Dense_layer_after_concat_2[0][0]
__________________________________________________________________________________________________
Dense_layer_after_concat_4 (Den (None, 16)           144         Dense_layer_after_concat_3[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 2)            34          Dense_layer_after_concat_4[0][0]
==================================================================================================
Total params: 14,865,410
Trainable params: 763,310
Non-trainable params: 14,102,100
__________________________________________________________________________________________________
None
```

In [0]:

```
adam=keras.optimizers.Adam(lr=0.001)
model.compile(loss='categorical_crossentropy',optimizer='adam', metrics=[auroc])
```

In [0]:

```
from keras.callbacks import ModelCheckpoint, EarlyStopping
#https://machinelearningmastery.com/check-point-deep-learning-models-keras/
filepath="/content/drive/My Drive/1epochs:{epoch:03d}.hdf5"
checkpoint_3 = ModelCheckpoint(filepath, monitor='val_auroc', verbose=1, mode='max',  save_best_onl
```

```
y=True  )
```

In [0]:

```
#https://github.com/taomanwai/tensorboardcolab/
from time import time
import keras
from tensorboardcolab import *
#https://github.com/taomanwai/tensorboardcolab/
tbc=TensorBoardColab()
```

```
Wait for 8 seconds...
TensorBoard link:
https://b690347c.ngrok.io
```

In [0]:

```
#reduce_lr_1 = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=1, min_lr=0.002,verbose
= 1)

callbacks_list = [checkpoint_3,TensorBoardColabCallback(tbc)]
```

In [0]:

```
history=model.fit([tr_text__nor,tr_school_state_le,tr_project_grade_category_le,tr_categories_le,t
r_clean_subcategories_le,tr_teacher_prefix_le,tr_X_num],y_train,epochs=20,batch_size=400,verbose=1
,validation_data=([cv_text__nor,cv_school_state_le,cv_project_grade_category_le,cv_categories_le,c
v_clean_subcategories_le,cv_teacher_prefix_le,cv_X_num], y_cv),callbacks=callbacks_list)
```

```
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/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/backend/tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Please us
e tf.compat.v1.assign_add instead.

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

Train on 69918 samples, validate on 17480 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorboardcolab/core.py:49: The na
me tf.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1122: The name t
f.summary.merge_all is deprecated. Please use tf.compat.v1.summary.merge_all instead.

Epoch 1/20
69918/69918 [==============================] - 124s 2ms/step - loss: 0.5961 - auroc: 0.6743 - val_
loss: 0.5452 - val_auroc: 0.7247

Epoch 00001: val_auroc improved from -inf to 0.72469, saving model to /content/drive/My
Drive/1epochs:001.hdf5
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorboardcolab/callbacks.py:51: T
he name tf.Summary is deprecated. Please use tf.compat.v1.Summary instead.

Epoch 2/20
69918/69918 [==============================] - 121s 2ms/step - loss: 0.5087 - auroc: 0.7422 - val_
loss: 0.4990 - val_auroc: 0.7384

Epoch 00002: val_auroc improved from 0.72469 to 0.73842, saving model to /content/drive/My
Drive/1epochs:002.hdf5
Epoch 3/20
69918/69918 [==============================] - 120s 2ms/step - loss: 0.4654 - auroc: 0.7651 - val_
loss: 0.4757 - val_auroc: 0.7429

Epoch 00003: val_auroc improved from 0.73842 to 0.74294, saving model to /content/drive/My
Drive/1epochs:003.hdf5
Epoch 4/20
69918/69918 [==============================] - 118s 2ms/step - loss: 0.4375 - auroc: 0.7831 - val_
loss: 0.4613 - val_auroc: 0.7391
```

loss: 0.4615 - val_auroc: 0.7391

Epoch 00004: val_auroc did not improve from 0.74294
Epoch 5/20
69918/69918 [==============================] - 118s 2ms/step - loss: 0.4115 - auroc: 0.8053 - val_
loss: 0.4568 - val_auroc: 0.7300

Epoch 00005: val_auroc did not improve from 0.74294
Epoch 6/20
69918/69918 [==============================] - 119s 2ms/step - loss: 0.3895 - auroc: 0.8316 - val_
loss: 0.4651 - val_auroc: 0.7223

Epoch 00006: val_auroc did not improve from 0.74294
Epoch 7/20
69918/69918 [==============================] - 121s 2ms/step - loss: 0.3596 - auroc: 0.8629 - val_
loss: 0.4844 - val_auroc: 0.7000

Epoch 00007: val_auroc did not improve from 0.74294
Epoch 8/20
69918/69918 [==============================] - 121s 2ms/step - loss: 0.3298 - auroc: 0.8955 - val_
loss: 0.5282 - val_auroc: 0.6863

Epoch 00008: val_auroc did not improve from 0.74294
Epoch 9/20
69918/69918 [==============================] - 122s 2ms/step - loss: 0.2976 - auroc: 0.9241 - val_
loss: 0.5809 - val_auroc: 0.6666

Epoch 00009: val_auroc did not improve from 0.74294
Epoch 10/20
69918/69918 [==============================] - 121s 2ms/step - loss: 0.2616 - auroc: 0.9496 - val_
loss: 0.6279 - val_auroc: 0.6559

Epoch 00010: val_auroc did not improve from 0.74294
Epoch 11/20
69918/69918 [==============================] - 120s 2ms/step - loss: 0.2268 - auroc: 0.9684 - val_
loss: 0.6970 - val_auroc: 0.6570

Epoch 00011: val_auroc did not improve from 0.74294
Epoch 12/20
69918/69918 [==============================] - 120s 2ms/step - loss: 0.1983 - auroc: 0.9798 - val_
loss: 0.7887 - val_auroc: 0.6411

Epoch 00012: val_auroc did not improve from 0.74294
Epoch 13/20
69918/69918 [==============================] - 117s 2ms/step - loss: 0.1738 - auroc: 0.9879 - val_
loss: 0.8738 - val_auroc: 0.6499

Epoch 00013: val_auroc did not improve from 0.74294
Epoch 14/20
69918/69918 [==============================] - 119s 2ms/step - loss: 0.1552 - auroc: 0.9916 - val_
loss: 0.9195 - val_auroc: 0.6471

Epoch 00014: val_auroc did not improve from 0.74294
Epoch 15/20
69918/69918 [==============================] - 120s 2ms/step - loss: 0.1439 - auroc: 0.9935 - val_
loss: 0.9896 - val_auroc: 0.6427

Epoch 00015: val_auroc did not improve from 0.74294
Epoch 16/20
69918/69918 [==============================] - 120s 2ms/step - loss: 0.1353 - auroc: 0.9951 - val_
loss: 1.0214 - val_auroc: 0.6459

Epoch 00016: val_auroc did not improve from 0.74294
Epoch 17/20
69918/69918 [==============================] - 120s 2ms/step - loss: 0.1265 - auroc: 0.9959 - val_
loss: 1.0723 - val_auroc: 0.6416

Epoch 00017: val_auroc did not improve from 0.74294
Epoch 18/20
69918/69918 [==============================] - 120s 2ms/step - loss: 0.1156 - auroc: 0.9968 - val_
loss: 1.1028 - val_auroc: 0.6402

Epoch 00018: val_auroc did not improve from 0.74294
Epoch 19/20
69918/69918 [==============================] - 119s 2ms/step - loss: 0.1097 - auroc: 0.9974 - val_
loss: 1.1201 - val_auroc: 0.6506

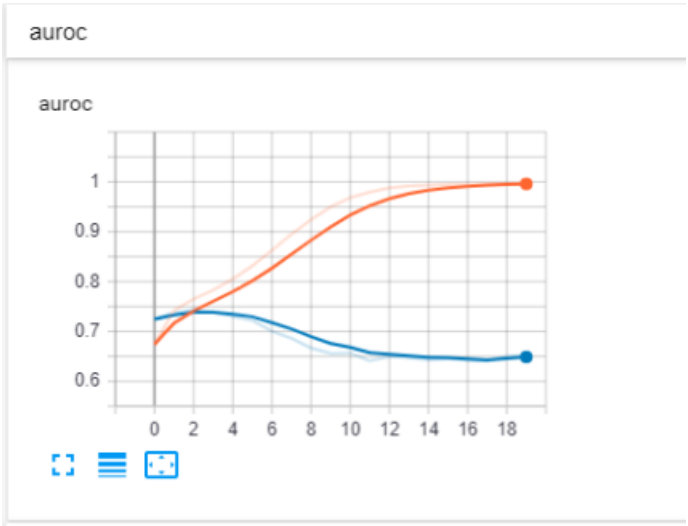
Epoch 00019: val_auroc did not improve from 0.74294

```
Epoch 00019: val_auroc did not improve from 0.74294
Epoch 20/20
69918/69918 [==============================] - 118s 2ms/step - loss: 0.1099 - auroc: 0.9972 - val_
loss: 1.1007 - val_auroc: 0.6528

Epoch 00020: val_auroc did not improve from 0.74294
```

In [0]:

```
from IPython.display import Image
Image('C:\\Users\\nnagari\\Downloads\\model_1_auc.PNG')
```

Out[0]:



In [0]:

```
model.load_weights('/content/drive/My Drive/1epochs:002.hdf5')
```

In [0]:

```
model.evaluate([te_text__nor,te_school_state_le,te_project_grade_category_le,te_categories_le,te_c
lean_subcategories_le,te_teacher_prefix_le,te_X_num],y_test,batch_size=100)
```

```
21850/21850 [==============================] - 36s 2ms/step
```

Out[0]:

```
[0.49255993580381713, 0.7532410295213476]
```

Model-2

In [22]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer_text = TfidfVectorizer(min_df=6,use_idf=True)
#Fitting train data and transforming  train ,cv and test to get idf values
tr_text_tfidf=tfidf_vectorizer_text.fit_transform(X_train['essay'])
cv_text_tfidf = tfidf_vectorizer_text.transform(X_cv['essay'])
te_text_tfidf = tfidf_vectorizer_text.transform(X_test['essay'])
print("Shape of matrix TFIDF Vectorizer on text \nTrain data-{},\nTest
data-{}".format(tr_text_tfidf.shape,te_text_tfidf.shape))
```

```
Shape of matrix TFIDF Vectorizer on text
Train data-(69918, 17213),
Test data-(21850, 17213)
```

In [0]:

```
#storing idf_values and feature_name
idf_feature=pd.DataFrame(tfidf_vectorizer_text.idf_,columns=['idf_values'])
idf_feature['feature_name']=tfidf_vectorizer_text.get_feature_names()
#sorting feature_name based on idf_values
idf_feature.sort_values(by=['idf_values'],ascending=False,inplace=True,axis=0)
```

In [24]:

```
#idf values and feature name
idf_feature[1995:2000]
```

Out[24]:

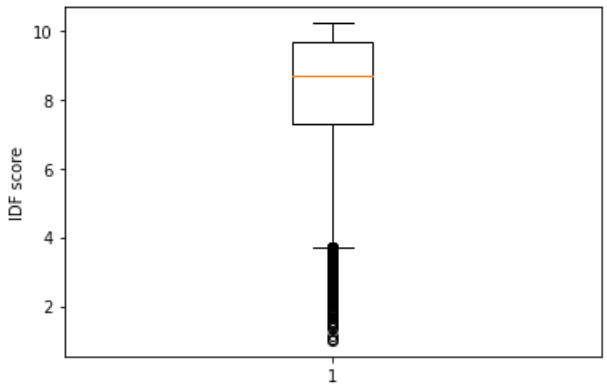
	idf_values	feature_name
10423	9.957868	numeral
10416	9.957868	nuisance
12803	9.957868	reptiles
8561	9.957868	kaplan
7993	9.957868	ingrain

In [25]:

```
plt.boxplot(idf_feature['idf_values'])
plt.ylabel("IDF score")
```

Out[25]:

```
Text(0, 0.5, 'IDF score')
```



In [26]:

```
for i in range(0,100+1,2):
    print("{}th percentile value = {}".format(i,np.percentile(idf_feature['idf_values'],[i])))
```

```
0th percentile value = [1.00773879]
2th percentile value = [3.78169623]
4th percentile value = [4.55189367]
6th percentile value = [5.05071864]
8th percentile value = [5.40394448]
10th percentile value = [5.69675442]
12th percentile value = [5.97507605]
14th percentile value = [6.24614138]
16th percentile value = [6.47833891]
18th percentile value = [6.68703257]
20th percentile value = [6.87697805]
22th percentile value = [7.04914723]
24th percentile value = [7.20633282]
26th percentile value = [7.37596922]
28th percentile value = [7.51070181]
30th percentile value = [7.63330413]
32th percentile value = [7.76064355]
34th percentile value = [7.86463327]
36th percentile value = [7.98070544]
```

```
38th percentile value = [8.07755526]
40th percentile value = [8.18480079]
42th percentile value = [8.30494511]
44th percentile value = [8.41742309]
46th percentile value = [8.49153106]
48th percentile value = [8.59974465]
50th percentile value = [8.68935681]
52th percentile value = [8.78779688]
54th percentile value = [8.85925584]
56th percentile value = [8.97703888]
58th percentile value = [9.06405025]
60th percentile value = [9.11057027]
62th percentile value = [9.21065373]
64th percentile value = [9.26472095]
66th percentile value = [9.32187936]
68th percentile value = [9.44704251]
70th percentile value = [9.51603538]
72th percentile value = [9.51603538]
74th percentile value = [9.59014335]
76th percentile value = [9.67018606]
78th percentile value = [9.75719744]
80th percentile value = [9.75719744]
82th percentile value = [9.85250762]
84th percentile value = [9.85250762]
86th percentile value = [9.95786813]
88th percentile value = [9.95786813]
90th percentile value = [10.07565117]
92th percentile value = [10.07565117]
94th percentile value = [10.20918256]
96th percentile value = [10.20918256]
98th percentile value = [10.20918256]
100th percentile value = [10.20918256]
```

1. Selecting the features based on idf score in between 3.78348349 and 10.20918256

In [0]:

```
selected_feature=idf_feature[(idf_feature['idf_values']>=3.78348349)&(idf_feature['idf_values']<=10.20918256) ]
words_list=[]
words_list=selected_feature['feature_name'].to_list()
```

In [28]:

```
selected_feature.shape
```

Out[28]:

```
(16867, 2)
```

In [29]:

```
idf_feature.shape
```

Out[29]:

```
(17213, 2)
```

In [0]:

```
from tqdm import tqdm
def remove_word(sentences):
    sentences_with_imp_word=[]
    for sentance in tqdm(sentences.values):
        sent_word=sentance.split(' ')
        sent=' '.join(word for word in sent_word if word in words_list)
        sentences_with_imp_word.append(sent)
    return(sentences_with_imp_word)
```

In [31]:

```
train_essay_imp=remove_word(X_train['essay'])
test_essay_imp=remove_word(X_test['essay'])
cv_essay_imp=remove_word(X_cv['essay'])
```

```
100%|██████████| 69918/69918 [47:13<00:00, 24.68it/s]
100%|██████████| 21850/21850 [14:56<00:00, 24.38it/s]
100%|██████████| 17480/17480 [11:52<00:00, 24.52it/s]
```

In [0]:

```
len_essay=[]
for sentance1 in (train_essay_imp):
    len_essay.append(len(sentance1.split()))
num_essay=np.array(len_essay)
max_length=num_essay.max()
```

In [0]:

```
max_length=192
```

In [0]:

```
tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_essay_imp)
train_encoded_essays = tokenizer.texts_to_sequences(train_essay_imp)
train_padded_essays = pad_sequences(train_encoded_essays, maxlen=max_length,padding='post')

test_encoded_essays = tokenizer.texts_to_sequences(test_essay_imp)
test_padded_essays = pad_sequences(test_encoded_essays, maxlen=max_length,padding='post')


cv_encoded_essays = tokenizer.texts_to_sequences(cv_essay_imp)
cv_padded_essays = pad_sequences(cv_encoded_essays, maxlen=max_length,padding='post')

vocab_size = len(tokenizer.word_index) + 1
```

In [0]:

```
word_vector=np.zeros((vocab_size, 300))
for word, i in tokenizer.word_index.items():
    vector=glove.get(word)
    if vector is not None:
        word_vector[i]=vector
```

In [0]:

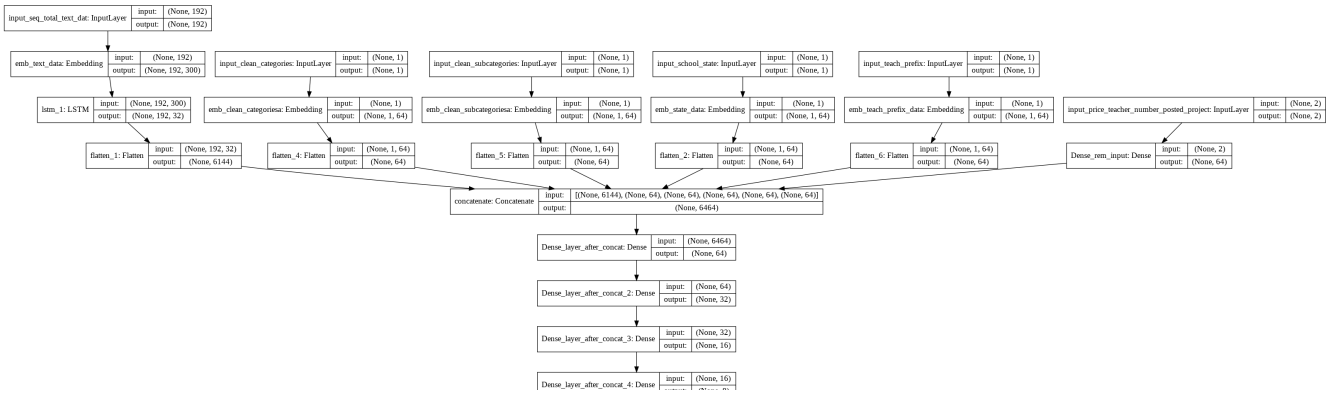
```
#model2 one hot
e=64
K.clear_session()
#model2 = Sequential()
#total text data
input_total_text=Input(shape=(max_length,),name="input_seq_total_text_dat")
emedding_layer_total_text = Embedding(vocab_size,300, weights=[word_vector],
input_length=max_length, trainable=False,name="emb_text_data")(input_total_text)
lstm_layer_total_text=LSTM(32,kernel_initializer='glorot_normal',recurrent_dropout=0.5,return_seque
nces=True)(emedding_layer_total_text)
#lstm_layer_total_text= ReLU()(lstm_layer_total_text)
flat_layer_total_text=Flatten()(lstm_layer_total_text)

#project state
input_state=Input(shape=(1,),name="input_school_state")
emedding_layer_state = Embedding(vocab_size_state,64,name="emb_state_data")(input_state)
flat_layer_state=Flatten()(emedding_layer_state)

#project grade
input_grade=Input(shape=(1,),name="input_project_grade_category")
emedding_layer_grade = Embedding(vocab_size_gra_cat,64,name="emb_pgc_data")(input_grade)
flat_layer_grade=Flatten()(emedding_layer_grade)

#project categiries
input_categories=Input(shape=(1,),name="input_clean_categories")
```

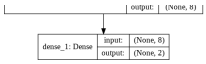
```
input_categories=Input(shape=(1,),name="input_clean_categories")
emedding_layer_cat = Embedding(vocab_size_cat,64,name="emb_clean_categoriesa")(input_categories)
flat_layer_cat=Flatten()(emedding_layer_cat)

#project subcategiries
input_subcategories=Input(shape=(1,),name="input_clean_subcategories")
emedding_layer_subcat = Embedding(vocab_size_sub_cat,64,name="emb_clean_subcategoriesa")
(input_subcategories)
flat_layer_subcat=Flatten()(emedding_layer_subcat)

#Project teacher_prefix
input_teach_prefix=Input(shape=(1,),name="input_teach_prefix")
emedding_layer_teach_prefix = Embedding(vocab_size_te_pr,64,name="emb_teach_prefix_data")
(input_teach_prefix)
flat_layer_teach_prefix=Flatten()(emedding_layer_teach_prefix)

#number data price and teacher_number_of_previously_posted_projects
input_num=Input(shape=(2,),name="input_price_teacher_number_posted_project")
dense_layer_price_num_project=Dense(64,
activation='sigmoid',name="Dense_rem_input",kernel_initializer='he_normal',kernel_regularizer=l2(0
.001))(input_num)

concat_layer=concatenate(inputs=[flat_layer_total_text,flat_layer_cat,flat_layer_subcat,flat_layer_
state,flat_layer_teach_prefix,dense_layer_price_num_project],name="concatenate")
#BN_1=BatchNormalization()(concat_layer)
#dense layer 1
dense_layer_after_concat=Dense(64,name="Dense_layer_after_concat",kernel_initializer='he_normal',a
ctivation='relu',kernel_regularizer=l2(0.001))(concat_layer)
#dense_layer_after_concat=ReLU()(dense_layer_after_concat)
#dropout layer 1
dropout_1=Dropout(0.5,name="Dropout_1")(dense_layer_after_concat)
#dense layer 2
dense_layer_after_concat_2=Dense(32,name="Dense_layer_after_concat_2",kernel_initializer='he_normal
',activation='relu',kernel_regularizer=l2(0.001))(dense_layer_after_concat)
#dense_layer_after_concat_2=ReLU()(dense_layer_after_concat_2)
#dropout layer 2
dropout_2=Dropout(0.5,name="Dropout_2")(dense_layer_after_concat_2)
#dense layer 3
dense_layer_after_concat_3=Dense(16,name="Dense_layer_after_concat_3",kernel_initializer='he_normal
',activation='relu',kernel_regularizer=l2(0.001))(dense_layer_after_concat_2)
#dense_layer_after_concat_3=ReLU()(dense_layer_after_concat_3)
dropout_3=Dropout(0.5,name="Dropout_3")(dense_layer_after_concat_3)

dense_layer_after_concat_4=Dense(8,name="Dense_layer_after_concat_4",kernel_initializer='he_normal'
,activation='relu',kernel_regularizer=l2(0.001))(dense_layer_after_concat_3)
#dense_layer_after_concat_4=ReLU()(dense_layer_after_concat_4)

output_layer=Dense(2, activation='softmax',kernel_initializer='glorot_uniform')
(dense_layer_after_concat_4)

model2=Model(inputs=[input_total_text,input_state,input_grade,input_categories,input_subcategories,
input_teach_prefix,input_num],outputs=output_layer)
```

In [0]:

```
#https://machinelearningmastery.com/visualize-deep-learning-neural-network-model-keras/
from keras.utils.vis_utils import plot_model
plot_model(model2, to_file='/content/drive/My Drive/model2.png', show_shapes=True,
show_layer_names=True)
```

Out[0]:





In [0]:

```
# summarize the model
print(model2.summary())
```

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_seq_total_text_dat (Input (None, 192)          0
__________________________________________________________________________________________________
emb_text_data (Embedding)       (None, 192, 300)     14208300    input_seq_total_text_dat[0][0]
__________________________________________________________________________________________________
input_clean_categories (InputLa (None, 1)            0
__________________________________________________________________________________________________
input_clean_subcategories (Inpu (None, 1)            0
__________________________________________________________________________________________________
input_school_state (InputLayer) (None, 1)            0
__________________________________________________________________________________________________
input_teach_prefix (InputLayer) (None, 1)            0
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 192, 32)      42624       emb_text_data[0][0]
__________________________________________________________________________________________________
emb_clean_categoriesa (Embeddin (None, 1, 64)        3328        input_clean_categories[0][0]
__________________________________________________________________________________________________
emb_clean_subcategoriesa (Embed (None, 1, 64)        25024       input_clean_subcategories[0][0]
__________________________________________________________________________________________________
emb_state_data (Embedding)      (None, 1, 64)        3328        input_school_state[0][0]
__________________________________________________________________________________________________
emb_teach_prefix_data (Embeddin (None, 1, 64)        384         input_teach_prefix[0][0]
__________________________________________________________________________________________________
input_price_teacher_number_post (None, 2)            0
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 6144)         0           lstm_1[0][0]
__________________________________________________________________________________________________
flatten_4 (Flatten)             (None, 64)           0           emb_clean_categoriesa[0][0]
__________________________________________________________________________________________________
flatten_5 (Flatten)             (None, 64)           0           emb_clean_subcategoriesa[0][0]
__________________________________________________________________________________________________
flatten_2 (Flatten)             (None, 64)           0           emb_state_data[0][0]
__________________________________________________________________________________________________
flatten_6 (Flatten)             (None, 64)           0           emb_teach_prefix_data[0][0]
__________________________________________________________________________________________________
Dense_rem_input (Dense)         (None, 64)           192         input_price_teacher_number_posted
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 6464)         0           flatten_1[0][0]
                                                                 flatten_4[0][0]
                                                                 flatten_5[0][0]
                                                                 flatten_2[0][0]
                                                                 flatten_6[0][0]
                                                                 Dense_rem_input[0][0]
__________________________________________________________________________________________________
Dense_layer_after_concat (Dense (None, 64)           413760      concatenate[0][0]
__________________________________________________________________________________________________
Dense_layer_after_concat_2 (Den (None, 32)           2080        Dense_layer_after_concat[0][0]
__________________________________________________________________________________________________
Dense_layer_after_concat_3 (Den (None, 16)           528         Dense_layer_after_concat_2[0][0]
__________________________________________________________________________________________________
Dense_layer_after_concat_4 (Den (None, 8)            136         Dense_layer_after_concat_3[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 2)            18          Dense_layer_after_concat_4[0][0]
==================================================================================================
Total params: 14,699,702
Trainable params: 491,402
Non-trainable params: 14,208,300
__________________________________________________________________________________________________
None
```

In [51]:

```
adam=keras.optimizers.Adam(lr=0.01)
```

```
model2.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['accuracy',auroc])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:793: The name t
f.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3576: The name tf.log is deprecated. Please use tf.ma
th.log instead.

WARNING:tensorflow:From <ipython-input-15-4a25250c5bd7>:5: py_func (from
tensorflow.python.ops.script_ops) is deprecated and will be removed in a future version.
Instructions for updating:
tf.py_func is deprecated in TF V2. Instead, there are two
    options available in V2.
    - tf.py_function takes a python function which manipulates tf eager
    tensors instead of numpy arrays. It's easy to convert a tf eager tensor to
    an ndarray (just call tensor.numpy()) but having access to eager tensors
    means `tf.py_function`s can use accelerators such as GPUs as well as
    being differentiable using a gradient tape.
    - tf.numpy_function maintains the semantics of the deprecated tf.py_func
    (it is not differentiable, and manipulates numpy arrays). It drops the
    stateful argument making all functions stateful.
```

In [0]:

```
from keras.callbacks import ModelCheckpoint, EarlyStopping
#https://machinelearningmastery.com/check-point-deep-learning-models-keras/
filepath="/content/drive/My Drive/model2-epochs:{epoch:03d}.hdf5"
checkpoint_3 = ModelCheckpoint(filepath, monitor='val_auroc', verbose=1, mode='max',save_best_only=
True)
```

In [56]:

```
#https://github.com/taomanwai/tensorboardcolab/
from time import time
import keras
from tensorboardcolab import *
#https://github.com/taomanwai/tensorboardcolab/
tbc=TensorBoardColab()
```

```
Wait for 8 seconds...
TensorBoard link:
https://3987b100.ngrok.io
```

In [0]:

```
callbacks_list = [checkpoint_3,TensorBoardColabCallback(tbc)]
```

In [0]:

```
history=model2.fit([train_padded_essays,tr_school_state_le,tr_project_grade_category_le,tr_categori
es_le,tr_clean_subcategories_le,tr_teacher_prefix_le,tr_X_num],y_train,epochs=20,batch_size=400,ve
rbose=1,validation_data=([cv_padded_essays,cv_school_state_le,cv_project_grade_category_le,cv_categ
ories_le,cv_clean_subcategories_le,cv_teacher_prefix_le,cv_X_num], y_cv),callbacks=callbacks_list)
```

```
Train on 69918 samples, validate on 17480 samples
Epoch 1/20
69918/69918 [==============================] - 76s 1ms/step - loss: 0.6134 - acc: 0.8462 - auroc:
0.6859 - val_loss: 0.5419 - val_acc: 0.8474 - val_auroc: 0.7143

Epoch 00001: saving model to /content/drive/My Drive/model2-epochs:001.hdf5
Epoch 2/20
69918/69918 [==============================] - 75s 1ms/step - loss: 0.5042 - acc: 0.8504 - auroc:
0.7345 - val_loss: 0.4902 - val_acc: 0.8480 - val_auroc: 0.7257

Epoch 00002: saving model to /content/drive/My Drive/model2-epochs:002.hdf5
Epoch 3/20
69918/69918 [==============================] - 75s 1ms/step - loss: 0.4607 - acc: 0.8522 - auroc:
0.7526 - val_loss: 0.4628 - val_acc: 0.8485 - val_auroc: 0.7255
```

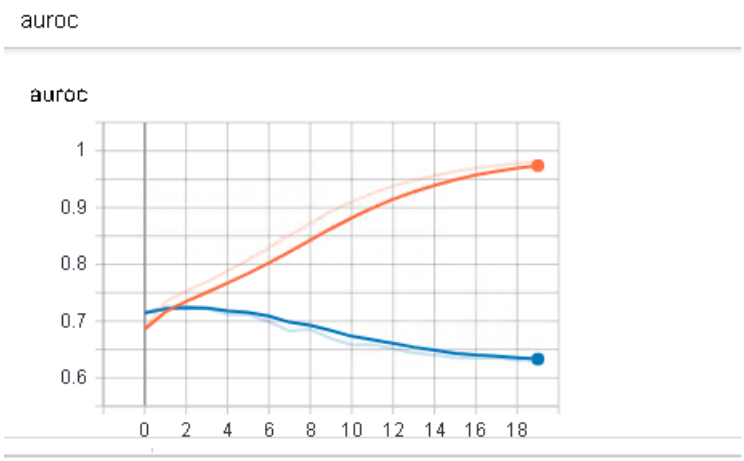
```
Epoch 00003: saving model to /content/drive/My Drive/model2-epochs:003.hdf5
Epoch 4/20
69918/69918 [==============================] - 75s 1ms/step - loss: 0.4313 - acc: 0.8560 - auroc:
0.7693 - val_loss: 0.4500 - val_acc: 0.8413 - val_auroc: 0.7226

Epoch 00004: saving model to /content/drive/My Drive/model2-epochs:004.hdf5
Epoch 5/20
69918/69918 [==============================] - 75s 1ms/step - loss: 0.4089 - acc: 0.8594 - auroc:
0.7883 - val_loss: 0.4463 - val_acc: 0.8491 - val_auroc: 0.7119

Epoch 00005: saving model to /content/drive/My Drive/model2-epochs:005.hdf5
Epoch 6/20
69918/69918 [==============================] - 75s 1ms/step - loss: 0.3888 - acc: 0.8648 - auroc:
0.8071 - val_loss: 0.4414 - val_acc: 0.8409 - val_auroc: 0.7115

Epoch 00006: saving model to /content/drive/My Drive/model2-epochs:006.hdf5
Epoch 7/20
69918/69918 [==============================] - 76s 1ms/step - loss: 0.3678 - acc: 0.8709 - auroc:
0.8295 - val_loss: 0.4521 - val_acc: 0.8438 - val_auroc: 0.6989

Epoch 00007: saving model to /content/drive/My Drive/model2-epochs:007.hdf5
Epoch 8/20
69918/69918 [==============================] - 75s 1ms/step - loss: 0.3485 - acc: 0.8779 - auroc:
0.8515 - val_loss: 0.4662 - val_acc: 0.8396 - val_auroc: 0.6825

Epoch 00008: saving model to /content/drive/My Drive/model2-epochs:008.hdf5
Epoch 9/20
69918/69918 [==============================] - 76s 1ms/step - loss: 0.3276 - acc: 0.8853 - auroc:
0.8727 - val_loss: 0.4798 - val_acc: 0.8348 - val_auroc: 0.6853

Epoch 00009: saving model to /content/drive/My Drive/model2-epochs:009.hdf5
Epoch 10/20
69918/69918 [==============================] - 76s 1ms/step - loss: 0.3076 - acc: 0.8947 - auroc:
0.8927 - val_loss: 0.5036 - val_acc: 0.8263 - val_auroc: 0.6693

Epoch 00010: saving model to /content/drive/My Drive/model2-epochs:010.hdf5
Epoch 11/20
69918/69918 [==============================] - 75s 1ms/step - loss: 0.2871 - acc: 0.9044 - auroc:
0.9104 - val_loss: 0.5253 - val_acc: 0.8288 - val_auroc: 0.6583

Epoch 00011: saving model to /content/drive/My Drive/model2-epochs:011.hdf5
Epoch 12/20
69918/69918 [==============================] - 74s 1ms/step - loss: 0.2678 - acc: 0.9128 - auroc:
0.9252 - val_loss: 0.5472 - val_acc: 0.8152 - val_auroc: 0.6585

Epoch 00012: saving model to /content/drive/My Drive/model2-epochs:012.hdf5
Epoch 13/20
69918/69918 [==============================] - 74s 1ms/step - loss: 0.2474 - acc: 0.9230 - auroc:
0.9380 - val_loss: 0.5897 - val_acc: 0.8035 - val_auroc: 0.6514

Epoch 00013: saving model to /content/drive/My Drive/model2-epochs:013.hdf5
Epoch 14/20
69918/69918 [==============================] - 74s 1ms/step - loss: 0.2324 - acc: 0.9302 - auroc:
0.9474 - val_loss: 0.6235 - val_acc: 0.8131 - val_auroc: 0.6439

Epoch 00014: saving model to /content/drive/My Drive/model2-epochs:014.hdf5
Epoch 15/20
69918/69918 [==============================] - 75s 1ms/step - loss: 0.2154 - acc: 0.9388 - auroc:
0.9563 - val_loss: 0.6543 - val_acc: 0.7866 - val_auroc: 0.6406

Epoch 00015: saving model to /content/drive/My Drive/model2-epochs:015.hdf5
Epoch 16/20
69918/69918 [==============================] - 74s 1ms/step - loss: 0.2042 - acc: 0.9438 - auroc:
0.9638 - val_loss: 0.6823 - val_acc: 0.7978 - val_auroc: 0.6354

Epoch 00016: saving model to /content/drive/My Drive/model2-epochs:016.hdf5
Epoch 17/20
69918/69918 [==============================] - 74s 1ms/step - loss: 0.1871 - acc: 0.9519 - auroc:
0.9698 - val_loss: 0.6962 - val_acc: 0.8064 - val_auroc: 0.6351

Epoch 00017: saving model to /content/drive/My Drive/model2-epochs:017.hdf5
Epoch 18/20
69918/69918 [==============================] - 73s 1ms/step - loss: 0.1793 - acc: 0.9554 - auroc:
0.9740 - val_loss: 0.7565 - val_acc: 0.7871 - val_auroc: 0.6356

Epoch 00018: saving model to /content/drive/My Drive/model2-epochs:018.hdf5
Epoch 19/20
```

```
69918/69918 [==============================] - 74s 1ms/step - loss: 0.1714 - acc: 0.9587 - auroc:
0.9776 - val_loss: 0.7850 - val_acc: 0.7821 - val_auroc: 0.6311

Epoch 00019: saving model to /content/drive/My Drive/model2-epochs:019.hdf5
Epoch 20/20
69918/69918 [==============================] - 74s 1ms/step - loss: 0.1592 - acc: 0.9642 - auroc:
0.9808 - val_loss: 0.7967 - val_acc: 0.7995 - val_auroc: 0.6295

Epoch 00020: saving model to /content/drive/My Drive/model2-epochs:020.hdf5
```

In [0]:

```
from IPython.display import Image
Image('C:\\Users\\nnagari\\Downloads\\model_2_auc.PNG')
```

Out[0]:



In [0]:

```
model2.load_weights('/content/drive/My Drive/model2-epochs:002.hdf5')
```

In [64]:

```
model2.evaluate([test_padded_essays,te_school_state_le,te_project_grade_category_le,te_categories_l
e,te_clean_subcategories_le,te_teacher_prefix_le,te_X_num],y_test,batch_size=100)
```

```
21850/21850 [==============================] - 39s 2ms/step
```

Out[64]:

```
[0.45271887706946456, 0.8511212818682603, 0.7369468280421273]
```

Model-3

In [0]:

```
len_essay=[]
for sentance1 in (X_train['essay'].values):
    len_essay.append(len(sentance1.split()))
num_essay=np.array(len_essay)
max_length=num_essay.max()
```

In [0]:

```
max_length
```

Out[0]:

```
311
```

In [0]:

```
#text
tokenizer = Tokenizer()
tokenizer.fit_on_texts(X_train['essay'])
train_encoded_essays = tokenizer.texts_to_sequences(X_train['essay'])
train_padded_essays = pad_sequences(train_encoded_essays, maxlen=max_length,padding='post')

test_encoded_essays = tokenizer.texts_to_sequences(X_test['essay'])
test_padded_essays = pad_sequences(test_encoded_essays, maxlen=max_length,padding='post')


cv_encoded_essays = tokenizer.texts_to_sequences(X_cv['essay'])
cv_padded_essays = pad_sequences(cv_encoded_essays, maxlen=max_length,padding='post')

vocab_size = len(tokenizer.word_index) + 1
```

In [0]:

```
from sklearn.preprocessing import StandardScaler
#essay_stand = StandardScaler().fit(train_padded_essays)
tr_text__nor = train_padded_essays
cv_text__nor = cv_padded_essays
te_text__nor = test_padded_essays
print(tr_text__nor.shape)
```

```
(69918, 311)
```

In [0]:

```
#loading golve model
import pickle
with open('/content/drive/My Drive/glove_vectors', 'rb') as f:
    glove = pickle.load(f)
```

In [0]:

```
word_vector=np.zeros((vocab_size, 300))
for word, i in tokenizer.word_index.items():
    vector=glove.get(word)
    if vector is not None:
        word_vector[i]=vector
```

In [0]:

```
from collections import Counter
my_counter = Counter()
for word in project_data['clean_categories'].values:
    my_counter.update(word.split())

cat_dict = dict(my_counter)
sorted_cat_dict = dict(sorted(cat_dict.items(), key=lambda kv: kv[1]))
```

In [0]:

```
for word in project_data['clean_subcategories'].values:
    my_counter.update(word.split())

sub_cat_dict = dict(my_counter)
sorted_sub_cat_dict = dict(sorted(sub_cat_dict.items(), key=lambda kv: kv[1]))
```

In [0]:

```
# we use count vectorizer to convert the values into one hot encoded features
# Project subcategories
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_subcategories = CountVectorizer(vocabulary=list(sorted_sub_cat_dict.keys()), lowercase=
False, binary=True)

tr_sub_categories_one_hot=vectorizer_subcategories.fit_transform(X_train['clean_subcategories'].va
```

```
tr_sub_categories_one_hot=vectorizer_subcategories.fit_transform(X_train['clean_subcategories'].va
lues)
print(vectorizer_subcategories.get_feature_names())

cv_sub_categories_one_hot = vectorizer_subcategories.transform(X_cv['clean_subcategories'].values)
te_sub_categories_one_hot =
vectorizer_subcategories.transform(X_test['clean_subcategories'].values)

print(tr_sub_categories_one_hot.toarray()[0:2])
print("\nShape of matrix after one hot encodig for 'Project sub categories'\nTrain data-{},\nCV da
ta\t-{}\nTest data-{}".format(tr_sub_categories_one_hot.shape,cv_sub_categories_one_hot.shape,te_s
ub_categories_one_hot.shape))
```

```
['economics', 'communityservice', 'financialliteracy', 'parentinvolvement', 'extracurricular',
'civics_government', 'foreignlanguages', 'nutritioneducation', 'socialsciences', 'performingarts',
'charactereducation', 'teamsports', 'other', 'college_careerprep', 'warmth', 'care_hunger',
'music', 'history_geography', 'health_lifescience', 'earlydevelopment', 'esl', 'gym_fitness', 'env
ironmentalscience', 'history_civics', 'visualarts', 'health_wellness', 'music_arts',
'appliedsciences', 'appliedlearning', 'health_sports', 'literature_writing', 'specialneeds',
'mathematics', 'literacy', 'math_science', 'literacy_language']
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]]

Shape of matrix after one hot encodig for 'Project sub categories'
Train data-(69918, 36),
CV data -(17480, 36)
Test data-(21850, 36)
```

In [0]:

```
# we use count vectorizer to convert the values into one hot encoded features
# Project categories
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_categories = CountVectorizer(vocabulary=list(sorted_cat_dict.keys()),lowercase=False, b
inary=True)

tr_categories_one_hot=vectorizer_categories.fit_transform(X_train['clean_categories'].values)
print(vectorizer_categories.get_feature_names())

cv_categories_one_hot =vectorizer_categories.transform(X_cv['clean_categories'].values)
te_categories_one_hot =vectorizer_categories.transform(X_test['clean_categories'].values)

print(tr_categories_one_hot.toarray()[0:1])
print("\nShape of matrix after one hot encodig for 'Project categories'\nTrain data-{},\nCV data\t
-{}\nTest
data-{}".format(tr_categories_one_hot.shape,cv_categories_one_hot.shape,te_categories_one_hot.shape
))
```

```
['warmth', 'care_hunger', 'history_civics', 'music_arts', 'appliedlearning', 'specialneeds',
'health_sports', 'math_science', 'literacy_language']
[[0 0 0 0 0 0 0 0 1]]

Shape of matrix after one hot encodig for 'Project categories'
Train data-(69918, 9),
CV data -(17480, 9)
Test data-(21850, 9)
```

In [0]:

```
# you can do the similar thing with state, teacher_prefix and project_grade_category also
# we use count vectorizer to convert the values into one hot encoded features
#teacher_prefix
vectorizer_teacher_prefix = CountVectorizer(lowercase=False, binary=True)
tr_teacher_prefix_one_hot=vectorizer_teacher_prefix.fit_transform(X_train['teacher_prefix'].values
.astype('str'))
print(vectorizer_teacher_prefix.get_feature_names())

cv_teacher_prefix_one_hot =
vectorizer_teacher_prefix.transform(X_cv['teacher_prefix'].values.astype('str'))
te_teacher_prefix_one_hot =
vectorizer_teacher_prefix.transform(X_test['teacher_prefix'].values.astype('str'))

print(tr_teacher_prefix_one_hot.toarray()[0:1])
print("\nShape of matrix after one hot encodig for 'teacher_prefix'\nTrain data-{},\nCV data\t-{}\
```

```
nTest
data-{}".format(tr_teacher_prefix_one_hot.shape,cv_teacher_prefix_one_hot.shape,te_teacher_prefix_c
ne_hot.shape))
```

```
['dr', 'mr', 'mrs', 'ms', 'teacher']
[[0 0 1 0 0]]

Shape of matrix after one hot encodig for 'teacher_prefix'
Train data-(69918, 5),
CV data -(17480, 5)
Test data-(21850, 5)
```

In [0]:

```
```

In [0]:

```
# we use count vectorizer to convert the values into one hot encoded features
#school_state
vectorizer_school_state = CountVectorizer(lowercase=False, binary=True)
tr_school_state_one_hot=vectorizer_school_state.fit_transform(X_train['school_state'].values.astype
('str'))
print(vectorizer_school_state.get_feature_names())

cv_school_state_one_hot =
vectorizer_school_state.transform(X_cv['school_state'].values.astype('str'))
te_school_state_one_hot = vectorizer_school_state.transform(X_test['school_state'].values.astype('s
tr'))

print(tr_school_state_one_hot.toarray()[0:1])
print("\nShape of matrix after one hot encodig for 'teacher_prefix'\nTrain data-{},\nCV data\t-{}\
nTest
data-{}".format(tr_school_state_one_hot.shape,cv_school_state_one_hot.shape,te_school_state_one_hot
.shape))
```

```
['ak', 'al', 'ar', 'az', 'ca', 'co', 'ct', 'dc', 'de', 'fl', 'ga', 'hi', 'ia', 'id', 'il', 'in', 'k
s', 'ky', 'la', 'ma', 'md', 'me', 'mi', 'mn', 'mo', 'ms', 'mt', 'nc', 'nd', 'ne', 'nh', 'nj', 'nm',
'nv', 'ny', 'oh', 'ok', 'or', 'pa', 'ri', 'sc', 'sd', 'tn', 'tx', 'ut', 'va', 'vt', 'wa', 'wi', 'wv
', 'wy']
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]]

Shape of matrix after one hot encodig for 'teacher_prefix'
Train data-(69918, 51),
CV data -(17480, 51)
Test data-(21850, 51)
```

In [0]:

```
    # we use count vectorizer to convert the values into one hot encoded features
    #project_grade_category
    vectorizer_grade_category = CountVectorizer(lowercase=False, binary=True)

tr_grade_category_one_hot=vectorizer_grade_category.fit_transform(X_train['project_grade_category'
])
    print(vectorizer_grade_category.get_feature_names())

    cv_grade_category_one_hot = vectorizer_grade_category.transform(X_cv['project_grade_category']
)
    te_grade_category_one_hot =
vectorizer_grade_category.transform(X_test['project_grade_category'])

    print(tr_grade_category_one_hot.toarray()[0:1])
    print(cv_grade_category_one_hot.toarray()[0:1])
    print(te_grade_category_one_hot.toarray()[0:1])
    print("\nShape of matrix after one hot encodig for 'project_grade_category'\nTrain data-{},\nC
V data\t-{}\nTest data-{}".format(tr_grade_category_one_hot.shape,cv_grade_category_one_hot.shape,
te_grade_category_one_hot.shape))
```

```
['grades_3_5', 'grades_6_8', 'grades_9_12', 'grades_prek_2']
[[0 0 0 1]]
```

```
[[0 1 0 0]]
[[1 0 0 0]]

Shape of matrix after one hot encodig for 'project_grade_category'
Train data-(69918, 4),
CV data -(17480, 4)
Test data-(21850, 4)
```

In [0]:

```
# check this one: https://www.youtube.com/watch?v=0HOqOcln3Z4&t=530s
# standardization sklearn: https://scikit-
learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
from sklearn.preprocessing import StandardScaler

# price_standardized = standardScalar.fit(X_train['price'].values)
# this will rise the error
# ValueError: Expected 2D array, got 1D array instead: array=[725.05 213.03 329.   ... 399.   287.
73   5.5 ].
# Reshape your data either using array.reshape(-1, 1)

price_scalar = StandardScaler()
tr_price_standardized=price_scalar.fit_transform(X_train['price'].values.reshape(-1,1)) # finding t
he mean and standard deviation of this data
print(f"Mean : {price_scalar.mean_[0]}, Standard deviation : {np.sqrt(price_scalar.var_[0])}")

# Now standardize the data with above maen and variance.
cv_price_standardized = price_scalar.transform(X_cv['price'].values.reshape(-1, 1))
te_price_standardized = price_scalar.transform(X_test['price'].values.reshape(-1, 1))
```

```
Mean : 298.17651634772164, Standard deviation : 359.92483517322444
```

In [0]:

```
#teacher_number_of_previously_posted_projects
teacher_number_of_previously_posted_projects_scalar = StandardScaler()
tr_teacher_number_of_previously_posted_projects_standardized=teacher_number_of_previously_posted_pr
jects_scalar.fit_transform(X_train['teacher_number_of_previously_posted_projects'].values.reshape(
-1,1)) # finding the mean and standard deviation of this data
print(f"Mean : {teacher_number_of_previously_posted_projects_scalar.mean_[0]}, Standard deviation
: {np.sqrt(teacher_number_of_previously_posted_projects_scalar.var_[0])}")

# Now standardize the data with above maen and variance.
cv_teacher_number_of_previously_posted_projects_standardized =
teacher_number_of_previously_posted_projects_scalar.transform(X_cv['teacher_number_of_previously_po
ted_projects'].values.reshape(-1, 1))
te_teacher_number_of_previously_posted_projects_standardized =
teacher_number_of_previously_posted_projects_scalar.transform(X_test['teacher_number_of_previously_
osted_projects'].values.reshape(-1, 1))
print("\nShape of matrix after column standardization for
'teacher_number_of_previously_posted_projects'\nTrain data-{},\nCV data\t-{}\nTest data-{}".format
(tr_teacher_number_of_previously_posted_projects_standardized.shape,cv_teacher_number_of_previously
posted_projects_standardized.shape,te_teacher_number_of_previously_posted_projects_standardized.sh
ape))
```

```
Mean : 11.247475614291027, Standard deviation : 28.000278112533355

Shape of matrix after column standardization for 'teacher_number_of_previously_posted_projects'
Train data-(69918, 1),
CV data -(17480, 1)
Test data-(21850, 1)
```

In [0]:

```
#merging all columns
from scipy.sparse import hstack
tr_other_text=hstack((tr_school_state_one_hot,tr_grade_category_one_hot,tr_categories_one_hot,tr_s
ub_categories_one_hot,tr_teacher_prefix_one_hot,tr_teacher_number_of_previously_posted_projects_sta
ardized,tr_price_standardized)).todense()
cv_other_text=hstack((cv_school_state_one_hot,cv_grade_category_one_hot,cv_categories_one_hot,cv_s
ub_categories_one_hot,cv_teacher_prefix_one_hot,cv_teacher_number_of_previously_posted_projects_sta
ardized,cv_price_standardized)).todense()
te_other_text=hstack((te_school_state_one_hot,te_grade_category_one_hot,te_categories_one_hot,te_s
```

```
ub_categories_one_hot,te_teacher_prefix_one_hot,te_teacher_number_of_previously_posted_projects_sta
ardized,te_price_standardized)).todense()
```

In [0]:

```
tr_other_text = np.array(tr_other_text).reshape(69918,107,1)
cv_other_text = np.array(cv_other_text).reshape(17480, 107,1)
te_other_text = np.array(te_other_text).reshape(21850, 107,1)
```

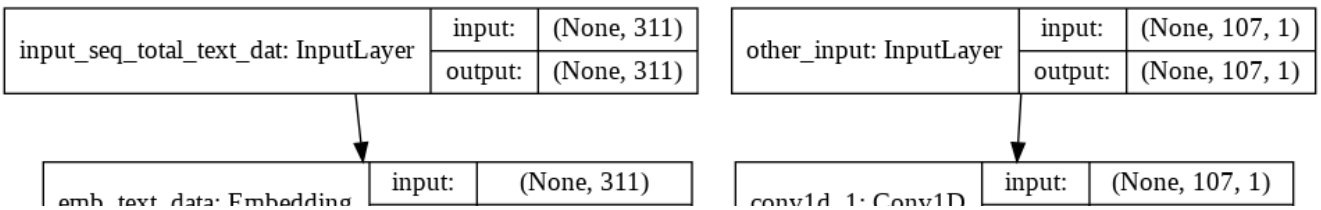
In [0]:

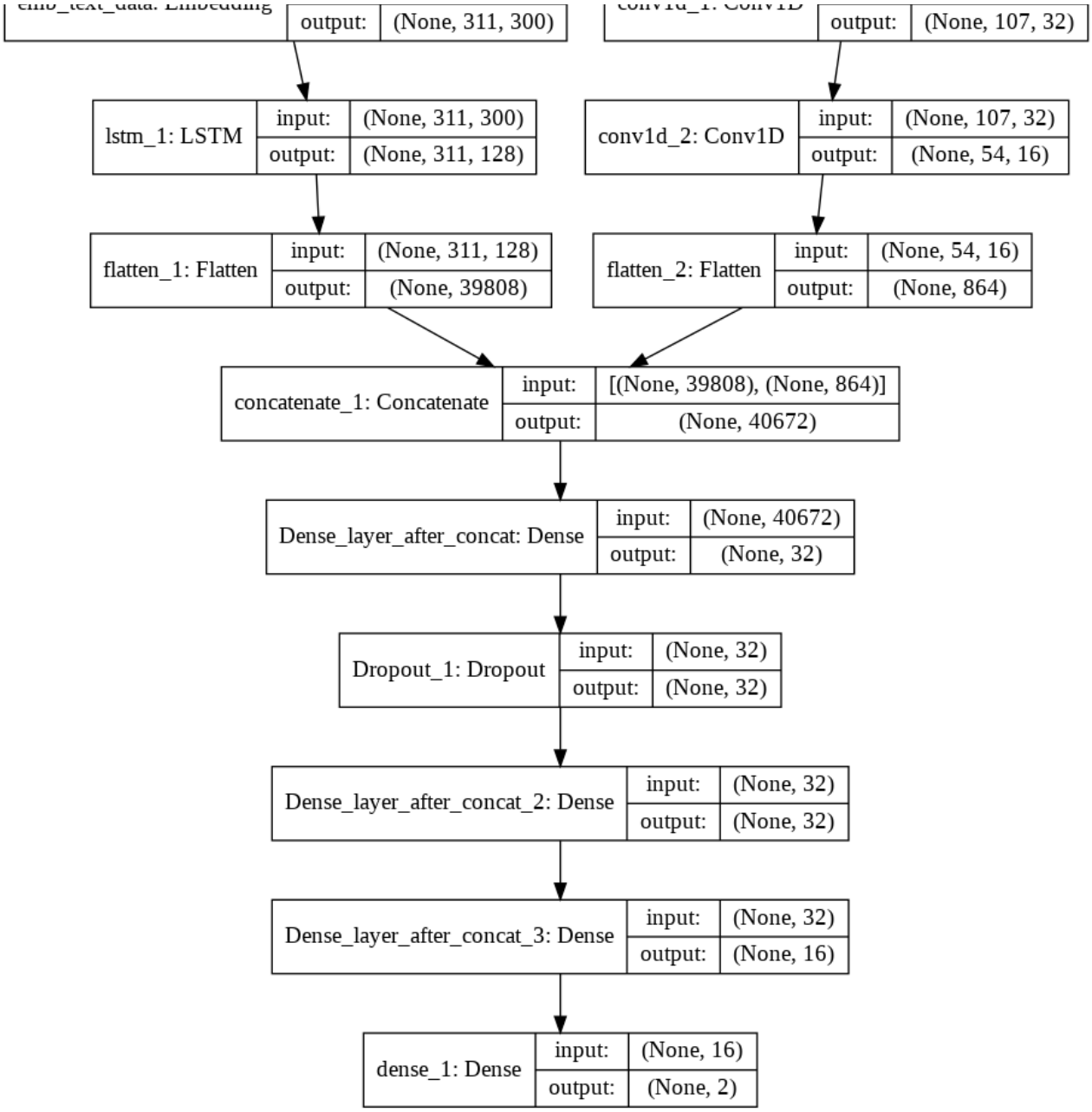
```
from keras import backend as K
K.clear_session()
#total text data
input_total_text=Input(shape=(max_length,),name="input_seq_total_text_dat")
emedding_layer_total_text = Embedding(vocab_size,300, weights=[word_vector],
input_length=max_length, trainable=False,name="emb_text_data")(input_total_text)
lstm_layer_total_text=LSTM(128,kernel_initializer='glorot_normal',return_sequences=True)
(emedding_layer_total_text)
#lstm_layer_total_text= ReLU()(lstm_layer_total_text)
flat_layer_total_text=Flatten()(lstm_layer_total_text)

other_input = Input(shape=(tr_other_text.shape[1],1), name='other_input')
#emedding_layer_other=Embedding(input_dim=tr_other_text.shape[1],output_dim=400,
input_length=tr_other_text.shape[1])(other_input)
conv_layer_other=Conv1D(32,7, activation='relu',padding='same',kernel_initializer='glorot_normal',s
trides=1)(other_input)
conv_layer_other=Conv1D(16,3, activation='relu',padding='same',kernel_initializer='glorot_normal',s
trides=2)(conv_layer_other)

#pool_layer_other=MaxPooling1D(pool_size=3)(other_input)
flat_layer_other = Flatten()(conv_layer_other)



concat_layer = concatenate([flat_layer_total_text,flat_layer_other])

dense_layer_after_concat=Dense(32,name="Dense_layer_after_concat",kernel_initializer='he_normal',a
ctivation='relu',kernel_regularizer=l2(0.0001))(concat_layer)
#dense_layer_after_concat=ReLU()(dense_layer_after_concat)
#dropout layer 1
dropout_1=Dropout(0.5,name="Dropout_1")(dense_layer_after_concat)
#dense layer 2
dense_layer_after_concat_2=Dense(32,name="Dense_layer_after_concat_2",kernel_initializer='he_normal
',activation='relu',kernel_regularizer=l2(0.0001))(dropout_1)
#dense_layer_after_concat_2=ReLU()(dense_layer_after_concat_2)
#dropout layer 2
#dropout_2=Dropout(0.5,name="Dropout_2")(dense_layer_after_concat_2)
#dense layer 3
dense_layer_after_concat_3=Dense(16,name="Dense_layer_after_concat_3",kernel_initializer='he_normal
',activation='relu',kernel_regularizer=l2(0.0001))(dense_layer_after_concat_2)
#dense_layer_after_concat_3=ReLU()(dense_layer_after_concat_3)
output_layer=Dense(2, activation='softmax',kernel_initializer='glorot_uniform')
(dense_layer_after_concat_3)

model3=Model(inputs=[input_total_text,other_input],outputs=output_layer)
```

In [0]:

```
#https://machinelearningmastery.com/visualize-deep-learning-neural-network-model-keras/
from keras.utils.vis_utils import plot_model
plot_model(model3, to_file='/content/drive/My Drive/model3.png', show_shapes=True,
show_layer_names=True)
```

Out[0]:





In [0]:

```
# summarize the model
print(model3.summary())
```

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_seq_total_text_dat (Input (None, 320)          0
__________________________________________________________________________________________________
other_input (InputLayer)        (None, 107, 1)       0
__________________________________________________________________________________________________
emb_text_data (Embedding)       (None, 320, 300)     14132700    input_seq_total_text_dat[0][0]
__________________________________________________________________________________________________
conv1d_1 (Conv1D)               (None, 107, 32)      256         other_input[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 320, 128)     219648      emb_text_data[0][0]
__________________________________________________________________________________________________
conv1d_2 (Conv1D)               (None, 54, 16)       1552        conv1d_1[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 40960)        0           lstm_1[0][0]
__________________________________________________________________________________________________
flatten_2 (Flatten)             (None, 864)          0           conv1d_2[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 41824)        0           flatten_1[0][0]
                                                                 flatten_2[0][0]
```

```
Dense_layer_after_concat (Dense (None, 32)            1338400     concatenate_1[0][0]
_____________________________________________________________________________________
Dropout_1 (Dropout)             (None, 32)            0           Dense_layer_after_concat[0][0]
_____________________________________________________________________________________
Dense_layer_after_concat_2 (Den (None, 32)            1056        Dropout_1[0][0]
_____________________________________________________________________________________
Dense_layer_after_concat_3 (Den (None, 16)            528         Dense_layer_after_concat_2[0][0]
_____________________________________________________________________________________
dense_1 (Dense)                 (None, 2)             34          Dense_layer_after_concat_3[0][0]
=====================================================================================================
Total params: 15,694,174
Trainable params: 1,561,474
Non-trainable params: 14,132,700
_____________________________________________________________________________________
None
```

In [0]:

```
import tensorflow as tf
from sklearn.metrics import roc_auc_score
def auc( y_true, y_pred ) :
    score = tf.py_func( lambda y_true, y_pred : roc_auc_score( y_true, y_pred).astype('float32'),
                        [y_true, y_pred],
                        'float32',
                        stateful=True,
                        name='sklearnAUC' )
    return score
```

In [0]:

```
adam=keras.optimizers.Adam(lr=10,decay = 1e-4)
model3.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['accuracy',auc])
```

In [0]:

```
from keras.callbacks import ModelCheckpoint, EarlyStopping
#https://machinelearningmastery.com/check-point-deep-learning-models-keras/
filepath="/content/drive/My Drive/model3-epochs:{epoch:03d}.hdf5"
checkpoint_3 = ModelCheckpoint(filepath, monitor='val_auc', verbose=1,save_best_only=True, mode='ma
x')
```

In [0]:

```
#https://github.com/taomanwai/tensorboardcolab/
from time import time
import keras
from tensorboardcolab import *
#https://github.com/taomanwai/tensorboardcolab/
tbc=TensorBoardColab()
```

```
Wait for 8 seconds...
TensorBoard link:
https://2f3c78e8.ngrok.io
```

In [0]:

```
callbacks_list = [checkpoint_3,TensorBoardColabCallback(tbc)]

history=model3.fit([tr_text__nor,tr_other_text],y_train,epochs=20,batch_size=2000,verbose=1,validat
ion_data=([cv_text__nor,cv_other_text], y_cv),callbacks=callbacks_list,class_weight = "balanced")
```

```
Train on 69918 samples, validate on 17480 samples
Epoch 1/20
69918/69918 [==============================] - 21s 301us/step - loss: 0.4888 - acc: 0.8400 - auc:
0.5679 - val_loss: 0.4446 - val_acc: 0.8507 - val_auc: 0.6654

Epoch 00001: val_auc improved from -inf to 0.66538, saving model to /content/drive/My
Drive/model3-epochs:001.hdf5
Epoch 2/20
69918/69918 [==============================] - 20s 289us/step - loss: 0.4193 - acc: 0.8492 - auc:
```

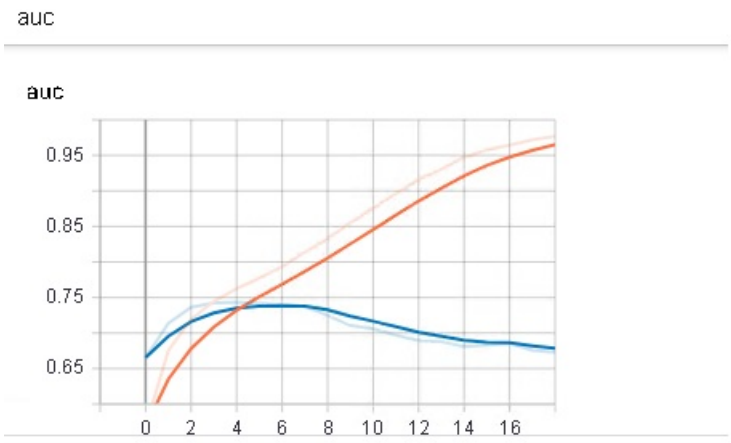
```
0.6754 - val_loss: 0.4275 - val_acc: 0.8509 - val_auc: 0.7137

Epoch 00002: val_auc improved from 0.66538 to 0.71367, saving model to /content/drive/My
Drive/model3-epochs:002.hdf5
Epoch 3/20
69918/69918 [==============================] - 20s 290us/step - loss: 0.4007 - acc: 0.8498 - auc:
0.7194 - val_loss: 0.4049 - val_acc: 0.8519 - val_auc: 0.7361

Epoch 00003: val_auc improved from 0.71367 to 0.73611, saving model to /content/drive/My
Drive/model3-epochs:003.hdf5
Epoch 4/20
69918/69918 [==============================] - 20s 289us/step - loss: 0.3891 - acc: 0.8510 - auc:
0.7445 - val_loss: 0.4027 - val_acc: 0.8524 - val_auc: 0.7423

Epoch 00004: val_auc improved from 0.73611 to 0.74226, saving model to /content/drive/My
Drive/model3-epochs:004.hdf5
Epoch 5/20
69918/69918 [==============================] - 20s 288us/step - loss: 0.3809 - acc: 0.8527 - auc:
0.7625 - val_loss: 0.4084 - val_acc: 0.8546 - val_auc: 0.7435

Epoch 00005: val_auc improved from 0.74226 to 0.74351, saving model to /content/drive/My
Drive/model3-epochs:005.hdf5
Epoch 6/20
69918/69918 [==============================] - 20s 288us/step - loss: 0.3704 - acc: 0.8577 - auc:
0.7774 - val_loss: 0.4116 - val_acc: 0.8538 - val_auc: 0.7421

Epoch 00006: val_auc did not improve from 0.74351
Epoch 7/20
69918/69918 [==============================] - 20s 288us/step - loss: 0.3622 - acc: 0.8601 - auc:
0.7933 - val_loss: 0.4136 - val_acc: 0.8543 - val_auc: 0.7394

Epoch 00007: val_auc did not improve from 0.74351
Epoch 8/20
69918/69918 [==============================] - 20s 288us/step - loss: 0.3503 - acc: 0.8650 - auc:
0.8136 - val_loss: 0.4058 - val_acc: 0.8534 - val_auc: 0.7375

Epoch 00008: val_auc did not improve from 0.74351
Epoch 9/20
69918/69918 [==============================] - 20s 287us/step - loss: 0.3385 - acc: 0.8695 - auc:
0.8327 - val_loss: 0.4084 - val_acc: 0.8522 - val_auc: 0.7244

Epoch 00009: val_auc did not improve from 0.74351
Epoch 10/20
69918/69918 [==============================] - 20s 287us/step - loss: 0.3216 - acc: 0.8771 - auc:
0.8551 - val_loss: 0.4131 - val_acc: 0.8517 - val_auc: 0.7105

Epoch 00010: val_auc did not improve from 0.74351
Epoch 11/20
69918/69918 [==============================] - 20s 289us/step - loss: 0.3063 - acc: 0.8819 - auc:
0.8753 - val_loss: 0.4317 - val_acc: 0.8422 - val_auc: 0.7063

Epoch 00011: val_auc did not improve from 0.74351
Epoch 12/20
69918/69918 [==============================] - 20s 289us/step - loss: 0.2878 - acc: 0.8901 - auc:
0.8961 - val_loss: 0.4529 - val_acc: 0.8365 - val_auc: 0.6970

Epoch 00012: val_auc did not improve from 0.74351
Epoch 13/20
69918/69918 [==============================] - 20s 290us/step - loss: 0.2698 - acc: 0.8974 - auc:
0.9159 - val_loss: 0.4608 - val_acc: 0.8494 - val_auc: 0.6892

Epoch 00013: val_auc did not improve from 0.74351
Epoch 14/20
69918/69918 [==============================] - 20s 287us/step - loss: 0.2518 - acc: 0.9048 - auc:
0.9304 - val_loss: 0.4889 - val_acc: 0.8318 - val_auc: 0.6879

Epoch 00014: val_auc did not improve from 0.74351
Epoch 15/20
69918/69918 [==============================] - 20s 289us/step - loss: 0.2299 - acc: 0.9141 - auc:
0.9474 - val_loss: 0.5185 - val_acc: 0.8367 - val_auc: 0.6810

Epoch 00015: val_auc did not improve from 0.74351
Epoch 16/20
69918/69918 [==============================] - 20s 287us/step - loss: 0.2142 - acc: 0.9245 - auc:
0.9578 - val_loss: 0.5669 - val_acc: 0.8360 - val_auc: 0.6833

Epoch 00016: val_auc did not improve from 0.74351
```

```
Epoch 17/20
69918/69918 [==============================] - 20s 287us/step - loss: 0.2052 - acc: 0.9293 - auc:
0.9645 - val_loss: 0.6074 - val_acc: 0.8289 - val_auc: 0.6850

Epoch 00017: val_auc did not improve from 0.74351
Epoch 18/20
69918/69918 [==============================] - 20s 287us/step - loss: 0.1919 - acc: 0.9350 - auc:
0.9718 - val_loss: 0.6432 - val_acc: 0.8134 - val_auc: 0.6758

Epoch 00018: val_auc did not improve from 0.74351
Epoch 19/20
69918/69918 [==============================] - 20s 288us/step - loss: 0.1793 - acc: 0.9427 - auc:
0.9769 - val_loss: 0.6771 - val_acc: 0.8100 - val_auc: 0.6728

Epoch 00019: val_auc did not improve from 0.74351
Epoch 20/20
69918/69918 [==============================] - 20s 289us/step - loss: 0.1714 - acc: 0.9481 - auc:
0.9804 - val_loss: 0.7164 - val_acc: 0.8200 - val_auc: 0.6704

Epoch 00020: val_auc did not improve from 0.74351
```

In [0]:

```
from IPython.display import Image
Image('C:\\Users\\nnagari\\Downloads\\model_3_auc.png')
```

Out[0]:



In [0]:

```
model3.load_weights('/content/drive/My Drive/model3-epochs:005.hdf5')
```

In [0]:

```
model3.evaluate([te_text__nor,te_other_text],y_test,batch_size=100)
```

```
21850/21850 [==============================] - 35s 2ms/step
```

Out[0]:

```
[0.4116582988465132, 0.8510297487747751, 0.7457665919711988]
```

In [72]:

```
# Please compare all your models using Prettytable library
from prettytable import PrettyTable

table = PrettyTable()

table.field_names = ["Models", "Train","CV","Test"]

table.add_row(["Model-1", "0.7651","0.7429","0.7532"])
table.add_row(["Model-2", "0.7345","0.7252","0.7369"])
table.add_row(["Model-3", "0.7625", "0.7435","0.7457"])
```

```
print(table)
```

```
+---------+--------+--------+--------+
|  Models | Train  |   CV   |  Test  |
+---------+--------+--------+--------+
| Model-1 | 0.7651 | 0.7429 | 0.7532 |
| Model-2 | 0.7345 | 0.7252 | 0.7369 |
| Model-3 | 0.7625 | 0.7435 | 0.7457 |
+---------+--------+--------+--------+
```