DonorsChoose

DonorsChoose.org receives hundreds of thousands of project proposals each year for classroom projects in need of funding. Right now, a large number of volunteers is needed to manually screen each submission before it's approved to be posted on the DonorsChoose.org website.

Next year, DonorsChoose.org expects to receive close to 500,000 project proposals. As a result, there are three main problems they need to solve:

How to scale current manual processes and resources to screen 500,000 projects so that they can be posted as quickly and as efficiently as possible How to increase the consistency of project vetting across different volunteers to improve the experience for teachers How to focus volunteer time on the applications that need the most assistance The goal of the competition is to predict whether or not a DonorsChoose.org project proposal submitted by a teacher will be approved, using the text of project descriptions as well as additional metadata about the project, teacher, and school. DonorsChoose.org can then use this information to identify projects most likely to need further review before approval.

About the DonorsChoose Data Set

The `train.csv` data set provided by DonorsChoose contains the following features:

Feature	Description
`project_id`	A unique identifier for the proposed p **Example:** `p036502`
`project_title`	Title of the project. **Examples:** • `Art Will Make You Happy` • `First Grade Fun`

Feature	Description
`project_grade_category`	Grade level of students for which the project is targeted. One of the follow enumerated values: • `Grades PreK-2` • `Grades 3-5` • `Grades 6-8` • `Grades 9-12`
`project_subject_categories`	One or more (comma-separated) su categories for the project from the fo enumerated list of values: • `Applied Learning` • `Care & Hunger` • `Health & Sports` • `History & Civics` • `Literacy & Language` • `Math & Science` • `Music & The Arts` • `Special Needs` • `Warmth` **Examples:** • `Music & The Arts` • `Literacy & Language, Ma & Science`
`school_state`	State where school is located (Two-l U.S. postal code). **Example:** `WY`

Feature	Description
project_subject_subcategories	One or more (comma-separated) su subcategories for the project. **Exam** • `Literacy` • `Literature & Writing, Social Sciences`
project_resource_summary	An explanation of the resources nee the project. **Example:** • `My students need hands literacy materials to manage sensory needs!`
project_essay_1	First application essay*
project_essay_2	Second application essay*
project_essay_3	Third application essay*
project_essay_4	Fourth application essay*
project_submitted_datetime	Datetime when project application w submitted. **Example:** `2016-04-28 12:43:56.245`
teacher_id	A unique identifier for the teacher of proposed project. **Example:** `bdf8baa8fedef6bfeec7ae4ff1c`

Feature	Description
`teacher_prefix`	Teacher's title. One of the following enumerated values: • `nan` • `Dr.` • `Mr.` • `Mrs.` • `Ms.` • `Teacher.`
`teacher_number_of_previously_posted_projects`	Number of project applications previ submitted by the same teacher. **Exa** 2

* See the section **Notes on the Essay Data** for more details about these features.

Additionally, the `resources.csv` data set provides more data about the resources required for each project. Each line in this file represents a resource required by a project:

Feature	Description
`id`	A `project_id` value from the `train.csv` file. **Example:** `p036502`
`description`	Desciption of the resource. **Example:** `Tenor Saxophone Reeds, Box of 25`
`quantity`	Quantity of the resource required. **Example:** 3
`price`	Price of the resource required. **Example:** `9.95`

Note: Many projects require multiple resources. The `id` value corresponds to a `project_id` in train.csv, so you use it as a key to retrieve all resources needed for a project:

The data set contains the following label (the value you will attempt to predict):

Label	Description
`project_is_approved`	A binary flag indicating whether DonorsChoose approved the project. A value of `0` indicates the project was not approved, and a value of `1` indicates the project was approved.

Notes on the Essay Data

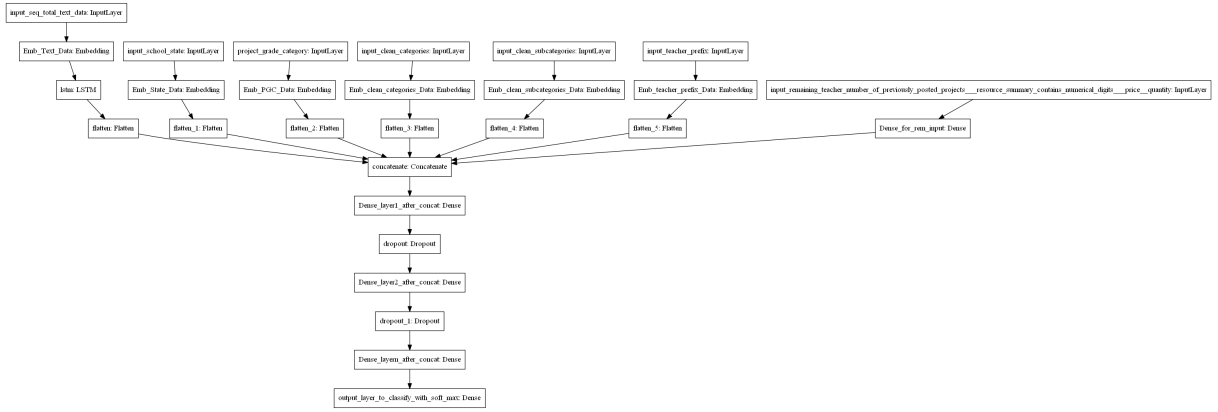
Prior to May 17, 2016, the prompts for the essays were as follows:

- __project_essay_1:__ "Introduce us to your classroom"
- __project_essay_2:__ "Tell us more about your students"
- __project_essay_3:__ "Describe how your students will use the materials you're requesting"
- __project_essay_3:__ "Close by sharing why your project will make a difference"

Starting on May 17, 2016, the number of essays was reduced from 4 to 2, and the prompts for the first 2 essays were changed to the following:

- __project_essay_1:__ "Describe your students: What makes your students special? Specific details about their background, your neighborhood, and your school are all helpful."
- __project_essay_2:__ "About your project: How will these materials make a difference in your students' learning and improve their school lives?"

For all projects with project_submitted_datetime of 2016-05-17 and later, the values of project_essay_3 and project_essay_4 will be NaN.



ref: https://i.imgur.com/w395Yk9.png

In [0]:
```
import numpy
from keras.datasets import imdb
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers.embeddings import Embedding
from keras.preprocessing import sequence
import pandas as pd
from keras.layers import Input, Embedding, LSTM, Dense,Concatenate,Drop
out
from keras.models import Model
import tensorflow as tf
from keras.layers import Input, Embedding, LSTM, Dense,Flatten
from keras.models import Model
import numpy as np
from sklearn.preprocessing import StandardScaler
from numpy import asarray
from numpy import zeros
from keras.layers.normalization import BatchNormalization
from keras.layers import Activation
from keras import regularizers
from keras import optimizers
from keras.optimizers import SGD
from google.colab import drive
from zipfile import ZipFile
```

```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split
import pickle
from keras.layers import LeakyReLU
from sklearn import preprocessing
from keras.layers.normalization import BatchNormalization
from keras.layers import Activation
from keras import regularizers
from keras.optimizers import SGD
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
```

```
Using TensorFlow backend.
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the `%tensorflow_version 1.x` magic: more info.

In [0]:
```
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remoun
t, call drive.mount("/content/drive", force_remount=True).
```

In [0]:
```
#archive  = ZipFile('/content/drive/My Drive/Assignments_DonorsChoose_2
018/glove.42B.300d.zip', 'r')
#data= archive.open('glove.42B.300d.txt','r')
```

In [0]:
```
#data = open('/content/drive/My Drive/glove_vectors2.txt', 'r')
```

In [0]:
```
project_data=pd.read_csv("/content/drive/My Drive/Assignments_DonorsCho
ose_2018/preprocessed_data.csv")
```

In [0]:
```
project_data.head(2)
```

Out[0]:

	school_state	teacher_prefix	project_grade_category	teacher_number_of_previously_p

	school_state	teacher_prefix	project_grade_category	teacher_number_of_previously_p
0	ca	mrs	grades_prek_2	53
1	ut	ms	grades_3_5	4

In [0]:
```
project_data_resources=pd.read_csv("/content/drive/My Drive/Assignments
_DonorsChoose_2018/resources.csv")
```

In [0]:
```
#price_data = project_data_resources.groupby('id').agg({'price':'sum',
 'quantity':'sum'}).reset_index()
#project_data = pd.merge(project_data, price_data, on='id', how='left')
```

In [0]:
```
project_data_resources.shape
```

Out[0]:
```
(1541272, 4)
```

In [0]:
```
project_data.head(2)
```

Out[0]:

	school_state	teacher_prefix	project_grade_category	teacher_number_of_previously_p

	school_state	teacher_prefix	project_grade_category	teacher_number_of_previously_p
0	ca	mrs	grades_prek_2	53
1	ut	ms	grades_3_5	4

In [0]:
```
x_train,X_test,y_train,Y_test=train_test_split(project_data,project_data['project_is_approved'],test_size=0.2,random_state=42)
X_train,X_cv,Y_train,Y_cv=train_test_split(x_train,y_train,test_size=0.2,random_state=42)
```

In [0]:
```
t = Tokenizer()
t.fit_on_texts(X_train['essay'])
vocab_size = len(t.word_index) + 1
# integer encode the documents
encoded_docs = t.texts_to_sequences(X_train['essay'])
print(encoded_docs)
# pad documents to a max length of 4 words
#max_length = 4
padded_docs = pad_sequences(encoded_docs, maxlen=300, padding='post')
print(padded_docs)
# load the whole embedding into memory
embeddings_index = dict()
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
```

```
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

```
[[  106    76 19532 ...     0     0     0]
 [   78    48  1097 ...     0     0     0]
 [    4     1    95 ...     0     0     0]
 ...
 [   25     3   921 ...     0     0     0]
 [    4     3   232 ...     0     0     0]
 [    2     8    40 ...     0     0     0]]
```

In [0]:
```
with open('/content/drive/My Drive/Assignments_DonorsChoose_2018/glove_vectors', 'rb') as f:
    model = pickle.load(f)
    glove_words =  set(model.keys())

# for train
embedding_matrix = np.zeros((vocab_size, 300))
for word, i in t.word_index.items():
    if word in glove_words:
        embedding_vector = model[word]
        embedding_matrix[i] = embedding_vector
```

In [0]:
```
encoded_docs_test = t.texts_to_sequences(X_test['essay'])
print(encoded_docs_test)
# pad documents to a max length of 4 words
#max_length = 4
padded_docs_test = pad_sequences(encoded_docs_test, maxlen=300, padding='post')
print(padded_docs_test)
# load the whole embedding into memory
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
```

```
to the client in order to avoid crashing it.

To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

```
[[   38  1502     1 ...     0     0     0]
 [21017   349  8120 ...     0     0     0]
 [   14    77   570 ...     0     0     0]
 ...
 [    2  1389     1 ...     0     0     0]
 [   14   273     3 ...     0     0     0]
 [ 2487  3241  6694 ...     0     0     0]]
```

In [0]:
```
encoded_docs_cv = t.texts_to_sequences(X_cv['essay'])
print(encoded_docs_cv)
# pad documents to a max length of 4 words
#max_length = 4
padded_docs_cv = pad_sequences(encoded_docs_cv, maxlen=300, padding='post')
print(padded_docs_cv)
# load the whole embedding into memory
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

```
[[    4     1  2097 ...     0     0     0]
 [  306  1340   321 ...     0     0     0]
 [   29    30    23 ...     0     0     0]
```

```
...
[   25     3    83 ...     0     0     0]
[  281   111 10497 ...     0     0     0]
[    2    49   112 ...     0     0     0]]
```

In [0]:
```
# Headline input: meant to receive sequences of 100 integers, between 1
 and 10000.
# Note that we can name any layer by passing it a "name" argument.
Input_Essay = Input(shape=(300,),  name='Input_Essay')

# This embedding layer will encode the input sequence
# into a sequence of dense 512-dimensional vectors.
embedding_essay = Embedding(vocab_size, 300, weights=[embedding_matrix
], input_length=300,trainable = False)(Input_Essay)

# A LSTM will transform the vector sequence into a single vector,
# containing information about the entire sequence

#lr= LeakyReLU(alpha = 0.3)(embedding_essay)
lstm_out = LSTM(300,return_sequences=True)(embedding_essay)
flatten_Essay = Flatten()(lstm_out)
```

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

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:66: The name tf.get_default_graph is deprec
ated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:4432: The name tf.random_uniform is depreca
ted. Please use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:190: The name tf.get_default_session is dep
recated. Please use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:197: The name tf.ConfigProto is deprecated.
```

```
Please use tf.compat.v1.ConfigProto instead.

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

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:207: The name tf.global_variables is deprec
ated. Please use tf.compat.v1.global_variables instead.

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

School_State

In [0]:
```
le_school_state = preprocessing.LabelEncoder()
Label_encoded_school_state=le_school_state.fit_transform(X_train['schoo
l_state'])
Label_encoded_school_state_cv=le_school_state.transform(X_cv['school_st
ate'])
Label_encoded_school_state_test=le_school_state.transform(X_test['schoo
l_state'])
```

In [0]:
```
padded_docs_school_state=Label_encoded_school_state
padded_docs_cv_school_state=Label_encoded_school_state_cv
padded_docs_test_school_state=Label_encoded_school_state_test
```

In [0]:
```
vocab_size_school_state = le_school_state.classes_
```

In [0]:
```
vocab_size_school_state_len=len(vocab_size_school_state)
```

In [0]:
```
Input_School_State_1 = Input(shape=(1,),  name='Input_School_State')

embedding_school_state_1=Embedding(vocab_size_school_state_len, 32, inp
ut_length=1)(Input_School_State_1)
flatten_School_State_1 = Flatten()(embedding_school_state_1)
```

teacher_prefix

In [0]:
```
le_teacher_prefix = preprocessing.LabelEncoder()
Label_encoded_teacher_prefix=le_teacher_prefix.fit_transform(X_train['t
eacher_prefix'])
```

In [0]:
```
Label_encoded_teacher_prefix_cv=le_teacher_prefix.transform(X_cv['teach
er_prefix'])
```

In [0]:
```
Label_encoded_teacher_prefix_test=le_teacher_prefix.transform(X_test['t
eacher_prefix'])
```

In [0]:
```
padded_docs_teacher_prefix=Label_encoded_teacher_prefix
padded_docs_cv_teacher_prefix=Label_encoded_teacher_prefix_cv
padded_docs_test_teacher_prefix=Label_encoded_teacher_prefix_test
```

In [0]:
```
vocab_size_teacher_prefix_len=len(le_teacher_prefix.classes_)
```

In [0]:
```
Input_teacher_prefix_1 = Input(shape=(1,),  name='Input_teacher_prefix_
1')

embedding_teacher_prefix_1=Embedding(vocab_size_teacher_prefix_len, 32,
 input_length=1)(Input_teacher_prefix_1)
flatten_teacher_prefix_1 = Flatten()(embedding_teacher_prefix_1)
```

grade_category

In [0]:
```
le_grade_category = preprocessing.LabelEncoder()
```

```
Label_encoded_grade_category=le_grade_category.fit_transform(X_train['p
roject_grade_category'])
```

In [0]:
```
Label_encoded_grade_category_cv=le_grade_category.transform(X_cv['proje
ct_grade_category'])
```

In [0]:
```
Label_encoded_grade_category_test=le_grade_category.transform(X_test['p
roject_grade_category'])
vocab_size_grade_category_len=len(le_grade_category.classes_)
```

In [0]:
```
padded_docs_project_grade_category=Label_encoded_grade_category
padded_docs_cv_project_grade_category=Label_encoded_grade_category_cv
padded_docs_test_project_grade_category=Label_encoded_grade_category_te
st
```

In [0]:
```
Input_grade_category_1 = Input(shape=(1,),  name='Input_grade_category_
1')
embedding_grade_category_1=Embedding(vocab_size_grade_category_len, 32,
 input_length=1)(Input_grade_category_1)
flatten_grade_category_1 = Flatten()(embedding_grade_category_1)
```

clean_categories

In [0]:
```
le_clean_categories = preprocessing.LabelEncoder()
Label_encoded_clean_categories=le_clean_categories.fit_transform(X_trai
n['clean_categories'])
```

In [0]:
```
Label_encoded_clean_categories_cv=le_clean_categories.transform(X_cv['c
lean_categories'])
```

In [0]:
```
Label_encoded_clean_categories_test=le_clean_categories.transform(X_tes
t['clean_categories'])
```

In [0]:
```
padded_docs_clean_categories=Label_encoded_clean_categories
padded_docs_cv_clean_categories=Label_encoded_clean_categories_cv
```

```
padded_docs_test_clean_categories=Label_encoded_clean_categories_test
```

In [0]:
```
vocab_size_grade_category_len=len(le_clean_categories.classes_)
```

In [0]:
```
Input_clean_categories_1 = Input(shape=(1,),  name='Input_clean_categories_1')

embedding_clean_categories_1=Embedding(vocab_size_grade_category_len, 32, input_length=1)(Input_clean_categories_1)
flatten_clean_categories_1 = Flatten()(embedding_clean_categories_1)
```

clean_subcategories

In [0]:
```
clean_subcategories_dict=dict()
#clean_subcategories_dict.
i=1
for sent in X_train['clean_subcategories']:
    if sent not in clean_subcategories_dict:
        clean_subcategories_dict[sent]=i
        i=i+1
```

In [0]:
```
vocab_size_clean_subcategories=len(clean_subcategories_dict)
```

In [0]:
```
Label_clean_subcategories=[]
for sent in X_train['clean_subcategories']:
    try:
        Label_clean_subcategories.append(clean_subcategories_dict[sent])

    except:
        Label_clean_subcategories.append(0)

padded_docs_clean_subcategories = Label_clean_subcategories
```

In [0]:
```
padded_docs_clean_subcategories = np.asarray(padded_docs_clean_subcategories)
```

In [0]:
```
#Label_encoded_clean_subcategories_test
padded_docs_cv_clean_subcategories=[]
Label_encoded_clean_subcategories_cv=[]
for sent in X_cv['clean_subcategories']:
    try:
        Label_encoded_clean_subcategories_cv.append(clean_subcategories
_dict[sent])

    except:
        Label_encoded_clean_subcategories_cv.append(0)

padded_docs_cv_clean_subcategories=Label_encoded_clean_subcategories_cv
```

In [0]:
```
padded_docs_cv_clean_subcategories = np.asarray(padded_docs_cv_clean_su
bcategories)
```

In [0]:
```
#Label_encoded_clean_subcategories_test
padded_docs_test_clean_subcategories=[]
Label_encoded_clean_subcategories_test=[]
for sent in X_test['clean_subcategories']:
    try:
        Label_encoded_clean_subcategories_test.append(clean_subcategori
es_dict[sent])

    except:
        Label_encoded_clean_subcategories_test.append(0)


padded_docs_test_clean_subcategories=Label_encoded_clean_subcategories_
test
```

In [0]:
```
padded_docs_clean_subcategories = np.asarray(padded_docs_clean_subcateg
ories)
padded_docs_test_clean_subcategories=np.asarray(padded_docs_test_clean_
subcategories)
```

In [0]:
```

```

In [0]:
```
Input_clean_subcategories_1 = Input(shape=(1,), name='Input_clean_subca
tegories_1')

embedding_clean_subcategories_1=Embedding(vocab_size_clean_subcategorie
s, 28, input_length=1)(Input_clean_subcategories_1)
flatten_clean_subcategories_1 = Flatten()(embedding_clean_subcategories
_1)
```

teacher_number_of_previously_posted_projects

In [0]:
```
rem_input_train = np.concatenate((X_train['price'].values.reshape(-1,1
),X_train['teacher_number_of_previously_posted_projects'].values.reshap
e(-1,1)), axis=1)
rem_input_cv =    np.concatenate((X_cv['price'].values.reshape(-1,1),X_
cv['teacher_number_of_previously_posted_projects'].values.reshape(-1,1
)), axis=1)
rem_input_test = np.concatenate((X_test['price'].values.reshape(-1,1),X
_test['teacher_number_of_previously_posted_projects'].values.reshape(-1
,1)), axis=1)
```

In [0]:
```
#from sklearn.preprocessing import StandardScalar
mms = StandardScaler().fit(rem_input_train)
rem_input_train_norm = mms.transform(rem_input_train)
rem_input_cv_norm = mms.transform(rem_input_cv)
rem_input_test_norm = mms.transform(rem_input_test)
```

In [0]:
```
remaining_input = Input(shape=(2,), name='remaining_input')
dense_1 = Dense(1, activation='relu',kernel_initializer="uniform")(rema
ining_input)
```

In [0]:
```
y = Concatenate()([flatten_Essay, flatten_School_State_1,flatten_teache
r_prefix_1,flatten_grade_category_1,flatten_clean_categories_1,flatten_
clean_subcategories_1,dense_1])
```

In [0]:
```
y = Dense(64, activation='relu',kernel_initializer="he_normal",kernel_r
egularizer=regularizers.l2(0.001))(y)
```

```
y=  Dropout(0.3)(y)


y = Dense(32, activation='relu',kernel_initializer="he_normal",kernel_r
egularizer=regularizers.l2(0.001))(y)
y=  Dropout(0.3)(y)
y = BatchNormalization()(y)

y = Dense(20, activation='relu',kernel_initializer="he_normal",kernel_r
egularizer=regularizers.l2(0.001))(y)
y=  Dropout(0.3)(y)

y = Dense(20, activation='relu',kernel_initializer="he_normal",kernel_r
egularizer=regularizers.l2(0.001))(y)
y=  Dropout(0.3)(y)









main_output = Dense(1, activation='sigmoid', name='main_output')(y)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:4479: The name tf.truncated_normal is depre
cated. Please use tf.random.truncated_normal instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:148: The name tf.placeholder_with_default i
s deprecated. Please use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:3733: calling dropout (from tensorflow.pyth
on.ops.nn_ops) with keep_prob is deprecated and will be removed in a fu
ture version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate =
1 - keep_prob`.
```

In [0]:
```
def auc(y_true,y_pred):
    return tf.py_func(roc_auc_score, (y_true, y_pred), tf.double)
```

In [0]:
```
learning_rate = 0.001
decay_rate = learning_rate / 10
#optimizers.Adam,optimizers.Nadam,optimizers.RMSprop
#from keras.optimizers import Adam
model = Model(inputs=[Input_Essay,Input_School_State_1,Input_teacher_pr
efix_1,Input_grade_category_1,Input_clean_categories_1,Input_clean_subc
ategories_1,remaining_input], outputs=[main_output])
#adam = optimizers.Adam(lr=0.001, beta_1=0.9, beta_2=0.999, amsgrad=Fal
se)
#lr=0.01,
#decay=decay_rate
adam = optimizers.Adam(lr=0.001)
model.compile(loss="binary_crossentropy", optimizer=adam, metrics=[auc
])
```

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

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:3657: The name tf.log is deprecated. Please
use tf.math.log instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorfl
ow_core/python/ops/nn_impl.py:183: where (from tensorflow.python.ops.ar
ray_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 <ipython-input-55-1ce08cb8f48c>:2: py_func (fro
m tensorflow.python.ops.script_ops) is deprecated and will be removed i
n 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 te
nsor to
```

```
    an ndarray (just call tensor.numpy()) but having access to eager te
nsors
    means `tf.py_function`s can use accelerators such as GPUs as well a
s
    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]:
```
model.summary()
```

```
Model: "model_1"
__________________________________________________________________________
_______________________
Layer (type)                 Output Shape              Param #     Connec
ted to
==========================================================================
=======================
Input_Essay (InputLayer)     (None, 300)               0

__________________________________________________________________________
_______________________
embedding_1 (Embedding)      (None, 300, 300)          14217900    Input_
Essay[0][0]
__________________________________________________________________________
_______________________
Input_School_State (InputLayer) (None, 1)              0

__________________________________________________________________________
_______________________
Input_teacher_prefix_1 (InputLa (None, 1)              0

__________________________________________________________________________
_______________________
Input_grade_category_1 (InputLa (None, 1)              0
```

```
__________________________________________________________________________
_____________________________
Input_clean_categories_1 (Input (None, 1)            0

__________________________________________________________________________
_____________________________
Input_clean_subcategories_1 (In (None, 1)            0

__________________________________________________________________________
_____________________________
lstm_1 (LSTM)                   (None, 300, 300)     721200      embedd
ing_1[0][0]
__________________________________________________________________________
_____________________________
embedding_2 (Embedding)         (None, 1, 32)        1632        Input_
School_State[0][0]
__________________________________________________________________________
_____________________________
embedding_3 (Embedding)         (None, 1, 32)        160         Input_
teacher_prefix_1[0][0]
__________________________________________________________________________
_____________________________
embedding_4 (Embedding)         (None, 1, 32)        128         Input_
grade_category_1[0][0]
__________________________________________________________________________
_____________________________
embedding_5 (Embedding)         (None, 1, 32)        1632        Input_
clean_categories_1[0][0]
__________________________________________________________________________
_____________________________
embedding_6 (Embedding)         (None, 1, 28)        10864       Input_
clean_subcategories_1[0][0]
__________________________________________________________________________
_____________________________
remaining_input (InputLayer)    (None, 2)            0

__________________________________________________________________________
_____________________________
```

```
flatten_1 (Flatten)             (None, 90000)        0           lstm_1
[0][0]
__________________________________________________________________________
_____
flatten_2 (Flatten)             (None, 32)           0           embedd
ing_2[0][0]
__________________________________________________________________________
_____
flatten_3 (Flatten)             (None, 32)           0           embedd
ing_3[0][0]
__________________________________________________________________________
_____
flatten_4 (Flatten)             (None, 32)           0           embedd
ing_4[0][0]
__________________________________________________________________________
_____
flatten_5 (Flatten)             (None, 32)           0           embedd
ing_5[0][0]
__________________________________________________________________________
_____
flatten_6 (Flatten)             (None, 28)           0           embedd
ing_6[0][0]
__________________________________________________________________________
_____
dense_1 (Dense)                 (None, 1)            3           remain
ing_input[0][0]
__________________________________________________________________________
_____
concatenate_1 (Concatenate)     (None, 90157)        0           flatte
n_1[0][0]
                                                                 flatte
n_2[0][0]
                                                                 flatte
n_3[0][0]
                                                                 flatte
n_4[0][0]
                                                                 flatte
n_5[0][0]
                                                                 flatte
```

```
n_6[0][0]
                                                                    dense_
1[0][0]
__________________________________________________________________________
__________________________________
dense_2 (Dense)              (None, 64)            5770112     concat
enate_1[0][0]
__________________________________________________________________________
__________________________________
dropout_1 (Dropout)          (None, 64)            0           dense_
2[0][0]
__________________________________________________________________________
__________________________________
dense_3 (Dense)              (None, 32)            2080        dropou
t_1[0][0]
__________________________________________________________________________
__________________________________
dropout_2 (Dropout)          (None, 32)            0           dense_
3[0][0]
__________________________________________________________________________
__________________________________
batch_normalization_1 (BatchNor (None, 32)            128         dropou
t_2[0][0]
__________________________________________________________________________
__________________________________
dense_4 (Dense)              (None, 20)            660         batch_
normalization_1[0][0]
__________________________________________________________________________
__________________________________
dropout_3 (Dropout)          (None, 20)            0           dense_
4[0][0]
__________________________________________________________________________
__________________________________
dense_5 (Dense)              (None, 20)            420         dropou
t_3[0][0]
__________________________________________________________________________
__________________________________
dropout_4 (Dropout)          (None, 20)            0           dense_
5[0][0]
```

```
_____
main_output (Dense)             (None, 1)            21          dropou
t_4[0][0]
==================================================================
==========================
Total params: 20,726,940
Trainable params: 6,508,976
Non-trainable params: 14,217,964
_____
_____
```

In [0]:
```
model_1=model.fit([padded_docs, padded_docs_school_state,padded_docs_te
acher_prefix,padded_docs_project_grade_category,padded_docs_clean_categ
ories,padded_docs_clean_subcategories,rem_input_train_norm],Y_train,epo
chs=10 ,batch_size=512,validation_data=([padded_docs_cv,padded_docs_cv_
school_state,padded_docs_cv_teacher_prefix,padded_docs_cv_project_grade
_category,padded_docs_cv_clean_categories,padded_docs_cv_clean_subcateg
ories,rem_input_cv_norm], Y_cv))
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:1033: The name tf.assign_add is deprecated.
Please use tf.compat.v1.assign_add instead.

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

Train on 69918 samples, validate on 17480 samples
Epoch 1/10
69918/69918 [==============================] - 65s 930us/step - loss:
0.8545 - auc: 0.5316 - val_loss: 0.6359 - val_auc: 0.6506
Epoch 2/10
69918/69918 [==============================] - 64s 914us/step - loss:
0.6163 - auc: 0.6026 - val_loss: 0.5799 - val_auc: 0.6856
Epoch 3/10
69918/69918 [==============================] - 64s 909us/step - loss:
0.5346 - auc: 0.6644 - val_loss: 0.5495 - val_auc: 0.7317
Epoch 4/10
69918/69918 [==============================] - 64s 919us/step - loss:
```

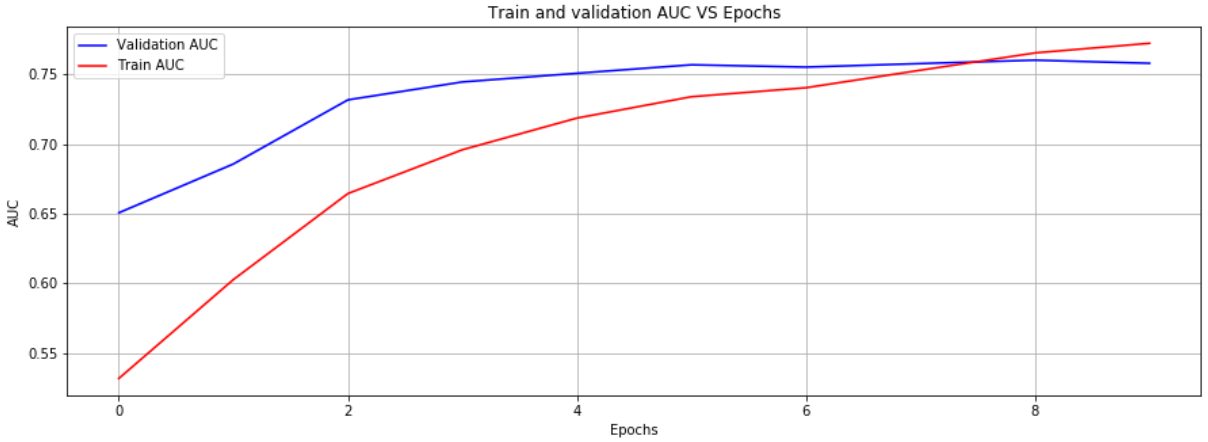
```
0.4971 - auc: 0.6958 - val_loss: 0.5087 - val_auc: 0.7445
Epoch 5/10
69918/69918 [==============================] - 64s 919us/step - loss:
0.4721 - auc: 0.7186 - val_loss: 0.4814 - val_auc: 0.7508
Epoch 6/10
69918/69918 [==============================] - 64s 911us/step - loss:
0.4500 - auc: 0.7339 - val_loss: 0.4465 - val_auc: 0.7569
Epoch 7/10
69918/69918 [==============================] - 64s 910us/step - loss:
0.4498 - auc: 0.7404 - val_loss: 0.4403 - val_auc: 0.7552
Epoch 8/10
69918/69918 [==============================] - 64s 910us/step - loss:
0.4312 - auc: 0.7530 - val_loss: 0.4296 - val_auc: 0.7577
Epoch 9/10
69918/69918 [==============================] - 64s 918us/step - loss:
0.4193 - auc: 0.7654 - val_loss: 0.4266 - val_auc: 0.7602
Epoch 10/10
69918/69918 [==============================] - 64s 909us/step - loss:
0.4147 - auc: 0.7723 - val_loss: 0.4263 - val_auc: 0.7579
```

In [0]:
```
import pickle
filename = '/content/drive/My Drive/model/donor_model1.pkl'
saved_model = pickle.dump(model,open('/content/drive/My Drive/donor_mod
el1.pkl','wb'))
```

In [0]:
```
model_history=model.predict([padded_docs_test,padded_docs_test_school_s
tate,padded_docs_test_teacher_prefix,padded_docs_test_project_grade_cat
egory,padded_docs_test_clean_categories,padded_docs_test_clean_subcateg
ories,rem_input_test_norm])
```

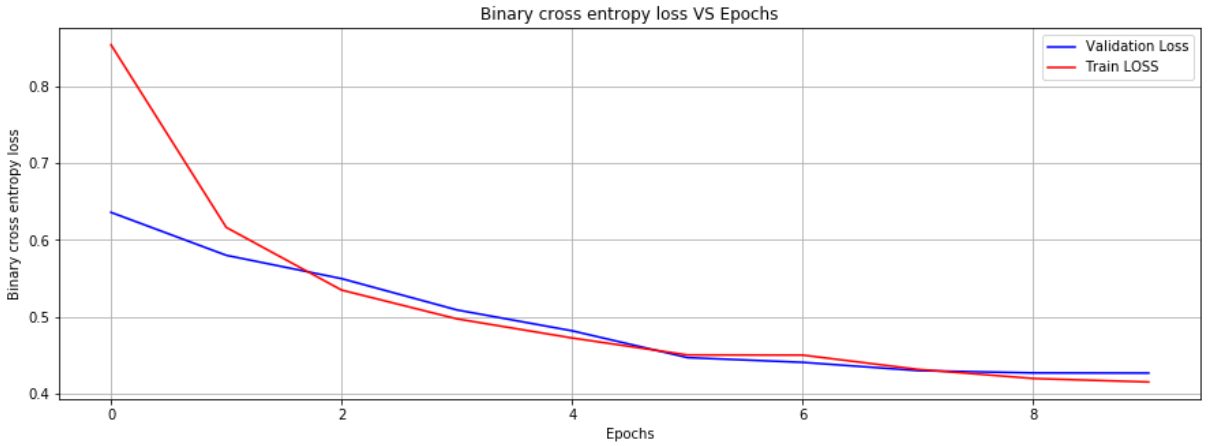
In [0]:
```
x = list(range(10))
vy = model_1.history['auc']
ty = model_1.history['val_auc']
plt.figure(figsize=(15,5))
plt.plot(x, ty, 'b', label="Validation AUC")
plt.plot(x, vy, 'r', label="Train AUC")
plt.xlabel('Epochs')
```

```
plt.ylabel('AUC')
plt.title('\nTrain and validation AUC VS Epochs')
plt.legend()
plt.grid()
plt.show()
```



In [0]:
```
import matplotlib.pyplot as plt

x = list(range(10))
vy = model_1.history['loss']
ty = model_1.history['val_loss']
plt.figure(figsize=(15,5))
plt.plot(x, ty, 'b', label="Validation Loss")
plt.plot(x, vy, 'r', label="Train LOSS")
plt.xlabel('Epochs')
plt.ylabel('Binary cross entropy loss')
plt.title('\nBinary cross entropy loss VS Epochs')
plt.legend()
plt.grid()
plt.show()
```



In [0]:
```
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
fpr, tpr, _ = roc_curve(Y_test,  model_history)
auc = roc_auc_score(Y_test, model_history)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
#plt.xlabel('Epochs')
plt.ylabel('Test AUC')
plt.show()
```