```
Jupyter Notebook
Fake News Model Creation
(autosaved)
Current Kernel Logo
Python 3
File
Edit
View
Insert
Cell
Kernel
Widgets
Help

Orginal Notebook Created by CIEP / Global DDM COE
Nidhi Sawhney, Stojan Maleschlijski & Ian Henry

import FN_Utils
import imp
from FN_Utils import *
%matplotlib inline
0.19.1
Labels are the binary/fake flag - real (0), fake 1

labels = getBinaryLabels()

print('Total Records:',len(labels))
print('Total Fake: ',sum(labels))
print('First 10 labels',labels[:10])
Total Records: 39373
Total Fake:  12948
First 10 labels [1 0 1 0 0 0 0 0 1 1]

# Input parameters to shape data
vocab_size = 5000 # Total Vocab size - anything bigger is rare
seq_len = 40 # Number of Words per record
# trn_data includes all data
trn_data = formatTrainingData(getTrainingData(),vocab_size,seq_len)
39372
(39372, 40)
Show the first record of our Training Data

print(trn_data[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 4310  453 1112
   29  108   23   69   33  501 4999 4999 4999 4999]

factor = 0.80  # How much data is used for training vs validation

t_index = round(factor*len(labels))
print (t_index)

model = createBinaryModel(len(trn_data[0]), vocab_size)
31498
/Users/i049374/OneDrive - SAP
SE/Documents/HANA/Install/TensorFlow/Fake.News/FN_Utils.py:62: UserWarning: The
`dropout` argument is no longer support in `Embedding`. You can apply a
`keras.layers.SpatialDropout1D` layer right after the `Embedding` layer to get the
same behavior.
  dropout=0.2),
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 40, 32)            160000
_________________________________________________________________
spatial_dropout1d_1 (Spatial (None, 40, 32)            0
_________________________________________________________________
dropout_1 (Dropout)          (None, 40, 32)            0
_________________
```

```
conv1d_1 (Conv1D)            (None, 40, 64)            10304
_________________________________________________________________
dropout_2 (Dropout)          (None, 40, 64)            0
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 20, 64)            0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1280)              0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               128100
_________________________________________________________________
dropout_3 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 101
=================================================================
Total params: 298,505
Trainable params: 298,505
Non-trainable params: 0
_________________________________________________________________

model.fit(trn_data[:t_index], labels[:t_index],
validation_data=(trn_data[t_index:len(trn_data)], labels[t_index:len(trn_data)]),
epochs=4, batch_size=256)
Train on 31498 samples, validate on 7874 samples
Epoch 1/4
31498/31498 [==============================] - 4s - loss: 0.3854 - acc: 0.8218 -
val_loss: 0.2663 - val_acc: 0.9211
Epoch 2/4
31498/31498 [==============================] - 5s - loss: 0.1477 - acc: 0.9520 -
val_loss: 0.2632 - val_acc: 0.9291
Epoch 3/4
31498/31498 [==============================] - 5s - loss: 0.1019 - acc: 0.9650 -
val_loss: 0.2787 - val_acc: 0.9336
Epoch 4/4
31498/31498 [==============================] - 5s - loss: 0.0783 - acc: 0.9731 -
val_loss: 0.3478 - val_acc: 0.9270
<keras.callbacks.History at 0x1c280bb5f8>

model.save('./Models/FakeNews-v7.h5')

preds = model.predict_classes(trn_data[t_index:len(trn_data)])
f1_score(labels[t_index:len(trn_data)],preds,labels=list(set(labels)),average=None)
6912/7874 [=========================>....] - ETA: 0s
array([ 0.94707777,  0.88224452])

confusion_matrix(labels[t_index:len(trn_data)],preds)
array([[5145,  142],
       [ 433, 2154]])

# We could just to visualise our Word Vectors
import sklearn
import FN_Utils
from sklearn.manifold import TSNE
from matplotlib import pyplot as plt
tsne = TSNE(n_components=2, random_state=0)
Y = tsne.fit_transform(trn_data[50:150])

start=1; end=250
dat = Y[start:end]
plt.figure(figsize=(15,15))
plt.scatter(dat[:, 0], dat[:, 1])
for label, x, y in zip(getWords()[start:end], dat[:, 0], dat[:, 1]):
    plt.text(x,y,label, color=np.random.rand(3)*0.7,
                 fontsize=14)
plt.show()
62806

Validation of our model and predictions

#sum(preds)
```

```
print('First Record used for validation',t_index)
print('\nFirst 10 prediction from validation data \n',preds[:10])
print('\nTotal Predicticted as Fake',sum(preds))
First Record used for validation 31498

First 10 prediction from validation data
 [[0]
 [1]
 [1]
 [1]
 [0]
 [0]
 [0]
 [1]
 [0]
 [0]]

Total Predicticted as Fake [2296]

# Print out a specific row
print(trn_data[31505])
[   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0  161   45  145   30 4999  107 1112  947   25]

print(trn_data[31504:31505])
print(labels[31504])
[[   0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0  445 4999  400   41 3016  334 1806 1160 4999]]
0

model.predict(trn_data[31504:31506])
array([[ 0.00155476],
       [ 0.99999857]], dtype=float32)

print(preds[31504 - t_index])
model.predict_classes(trn_data[31504:31506])
[0]
2/2 [==============================] - 0s
array([[0],
       [1]], dtype=int32)
```