3.Distllbert400000

January 7, 2022

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

```
Mounted at /content/drive
```

[ ]:
```
import os
import pathlib
from pathlib import Path
os.chdir("/content/drive/My Drive/Akarshan/BERT")
!ls -l
```

```
total 43071
drwx------ 2 root root     4096 Dec  3 16:27  clr
-rw------- 1 root root   488058 Dec 25 21:03  Compare.ipynb
-rw------- 1 root root   251068 Dec 26 21:29 'Copy of Distllbert400000.ipynb'
drwx------ 2 root root     4096 Dec  3 16:27  Data
-rw------- 1 root root  8306584 Dec 24 07:57  DBert1hk.hdf5
-rw------- 1 root root 12719136 Dec 24 07:57  DBert4hk.hdf5
-rw------- 1 root root   251068 Dec 26 21:28  Distllbert400000.ipynb
-rw------- 1 root root   476335 Dec 26 21:08 'EDA on results.ipynb'
drwx------ 2 root root     4096 Dec 18 07:14 'misc model'
-rw------- 1 root root    73707 Dec 25 10:58  model.png
drwx------ 2 root root     4096 Dec  3 16:27  papers
-rw------- 1 root root  8306584 Dec 19 08:56  Rbert4.hdf5
-rw------- 1 root root   203164 Dec 26 21:29  Retraining.ipynb
-rw------- 1 root root    86347 Dec 19 06:43  Roberta.ipynb
-rw------- 1 root root 12719160 Dec 25 10:35  SBert.hdf5
-rw------- 1 root root   203507 Dec 25 10:50  SciBert400k.ipynb
```

[ ]:
```
from psutil import virtual_memory
ram_gb = virtual_memory().total / 1e9
print('Your runtime has {:.1f} gigabytes of available RAM\n'.format(ram_gb))

if ram_gb < 20:
  print('Not using a high-RAM runtime')
else:
```

```
  print('You are using a high-RAM runtime!')
```

```
Your runtime has 27.3 gigabytes of available RAM

You are using a high-RAM runtime!
```

[ ]:
```
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
  print('Not connected to a GPU')
else:
  print(gpu_info)
```

```
Sun Dec 26 21:30:33 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 495.44       Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   34C    P0    27W / 250W |      0MiB / 16280MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

[ ]:
```
!pip install transformers
!pip install pympler
!pip install tensorflow_addons
```

```
Collecting transformers
  Downloading transformers-4.15.0-py3-none-any.whl (3.4 MB)
     || 3.4 MB 8.6 MB/s
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7
/dist-packages (from transformers) (21.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7
/dist-packages (from transformers) (2019.12.20)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
```

```
packages (from transformers) (2.23.0)
Collecting sacremoses
  Downloading sacremoses-0.0.46-py3-none-any.whl (895 kB)
     || 895 kB 65.7 MB/s
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-
packages (from transformers) (3.4.0)
Collecting pyyaml>=5.1
  Downloading PyYAML-6.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manyl
inux_2_12_x86_64.manylinux2010_x86_64.whl (596 kB)
     || 596 kB 64.9 MB/s
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7
/dist-packages (from transformers) (1.19.5)
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface_hub-0.2.1-py3-none-any.whl (61 kB)
     || 61 kB 589 kB/s
Collecting tokenizers<0.11,>=0.10.1
  Downloading tokenizers-0.10.3-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_6
4.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (3.3 MB)
     || 3.3 MB 60.1 MB/s
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7
/dist-packages (from transformers) (4.62.3)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7
/dist-packages (from transformers) (4.8.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.7/dist-packages (from huggingface-
hub<1.0,>=0.1.0->transformers) (3.10.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from packaging>=20.0->transformers)
(3.0.6)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata->transformers) (3.6.0)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7
/dist-packages (from requests->transformers) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->transformers) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7
/dist-packages (from requests->transformers) (2021.10.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (7.1.2)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.15.0)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.1.0)
Installing collected packages: pyyaml, tokenizers, sacremoses, huggingface-hub,
transformers
  Attempting uninstall: pyyaml
```

```
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed huggingface-hub-0.2.1 pyyaml-6.0 sacremoses-0.0.46
tokenizers-0.10.3 transformers-4.15.0
Collecting pympler
  Downloading Pympler-1.0.1-py3-none-any.whl (164 kB)
     || 164 kB 7.3 MB/s
Installing collected packages: pympler
Successfully installed pympler-1.0.1
Collecting tensorflow_addons
  Downloading tensorflow_addons-0.15.0-cp37-cp37m-
manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.1 MB)
     || 1.1 MB 6.7 MB/s
Requirement already satisfied: typeguard>=2.7 in /usr/local/lib/python3.7
/dist-packages (from tensorflow_addons) (2.7.1)
Installing collected packages: tensorflow-addons
Successfully installed tensorflow-addons-0.15.0
```

[ ]:
```
import numpy as np
import pickle
import pandas as pd
import pickle
import time
import matplotlib.pyplot as plt
import seaborn as sns
from pympler import asizeof
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import transformers
from transformers import pipeline
from tensorflow.keras.layers import concatenate
from transformers import TFAutoModel, AutoTokenizer,␣
 ↪AutoConfig,TFAutoModelForSequenceClassification
from tensorflow.keras.callbacks import ModelCheckpoint
from clr import clr_callback
import tensorflow_addons as tfa
```

[ ]:
```
csvfile = 'Data//data.csv'
dropna = 'Data//datadropna.csv'
sent_data_file = 'Data//sent_data.csv'
label_file = 'Data//label.csv'
vocab_file = 'Data//vocab_tr_w.txt'
```

[ ]:
```
df = pd.read_csv(dropna,usecols = ['SBE','Label'])
# df.dropna(inplace=True)
print(df.head())
```

```
print(df.shape)
```

```
   Label                                                SBE
0      1  To facilitate an easier notation throughout th...
1      0  Therefore _MATH_ defines a special order of ti...
2      0  This is important since only _MATH_ is the rea...
3      0  Note that in all contour time-integrals we ess...
4      0  Theorem _REF_ proves the equivalence of ensemb...
(1189321, 2)
```

0.1 Generating Embeddings

[ ]:
```
# Hyperparameters form paper

epoch = 30
patience = 10
lr = 1e-6
batch_size = 32
vocab = 30526 #will have to retrain Bert so not using
MAX_LEN = 128 #not enough ram for 256
```

[ ]:
```
model_name = 'distilbert-base-uncased'
config = AutoConfig.from_pretrained(model_name,trianing =False, num_labels=2 )
config.output_hidden_states = False

BERT = TFAutoModel.from_pretrained(model_name,config = config)

tokenizer = AutoTokenizer.from_pretrained(model_name,
                                          do_lower_case=True,
                                          use_fast=True,
                                          max_length=MAX_LEN,
                                          truncation=True,
                                          pad_to_max_length=True)

pipe = pipeline('feature-extraction', model=BERT,
                tokenizer=tokenizer,device=1)
```

```
Downloading:   0%|          | 0.00/483 [00:00<?, ?B/s]


Downloading:   0%|          | 0.00/347M [00:00<?, ?B/s]


Some layers from the model checkpoint at distilbert-base-uncased were not used
when initializing TFDistilBertModel: ['vocab_transform', 'vocab_layer_norm',
'activation_13', 'vocab_projector']
- This IS expected if you are initializing TFDistilBertModel from the checkpoint
of a model trained on another task or with another architecture (e.g.
```

```
initializing a BertForSequenceClassification model from a BertForPreTraining
model).
- This IS NOT expected if you are initializing TFDistilBertModel from the
checkpoint of a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).
All the layers of TFDistilBertModel were initialized from the model checkpoint
at distilbert-base-uncased.
If your task is similar to the task the model of the checkpoint was trained on,
you can already use TFDistilBertModel for predictions without further training.

Downloading:   0%|          | 0.00/28.0 [00:00<?, ?B/s]


Downloading:   0%|          | 0.00/226k [00:00<?, ?B/s]


Downloading:   0%|          | 0.00/455k [00:00<?, ?B/s]
```

[ ]:
```
batch=50
df = df.iloc[300000:400000,:]
step = int(df.shape[0]/batch)
step
```

[ ]:
```
2000
```

[ ]:
```
#### getting embedding vectors as bert output ###
# pipe returns embeddings for every token in a sent
# so features[x][0] is of shape (y,768) with y tokens in xth sentance
# taking the mean for y tokens give the embedding for the xth sent in total
# saving a batch of features as feature_matrix with 768 zeors as head
import pickle
import time
count = 50+50+50
for part in range(batch):
  i = part+count
  strt = time.time()
  indx = step*part
  indy = step*(part+1)
  # print(indx,indy)
  feature_matrix = array = np.empty(768, dtype=object)
  lst = []
  features = np.array(pipe(df['SBE'].iloc[indx:indy].to_list()))

  for idx in range(np.shape(features)[0]):
    sent_mean = np.mean(features[idx][0],axis =0)
    lst.append(sent_mean)
  # print(np.shape(lst))
  feature_matrix= np.array(lst)
  # print(np.shape(feature_matrix))
```

```
  # print(feature_matrix)

  with open('Data//embeddingsBr//embeddings'+str(i),'wb') as f:
    pickle.dump(feature_matrix,f)

  print(f'Part {part+1} of {batch} done. in {(time.time()-strt)/60:.2f} min')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:17:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences
(which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray
```

```
Part 1 of 50 done. in 7.50 min
Part 2 of 50 done. in 6.19 min
Part 3 of 50 done. in 6.85 min
Part 4 of 50 done. in 7.48 min
Part 5 of 50 done. in 7.56 min
Part 6 of 50 done. in 7.16 min
Part 7 of 50 done. in 7.17 min
Part 8 of 50 done. in 7.14 min
Part 9 of 50 done. in 7.11 min
Part 10 of 50 done. in 6.65 min
Part 11 of 50 done. in 7.20 min
Part 12 of 50 done. in 7.23 min
Part 13 of 50 done. in 6.88 min
Part 14 of 50 done. in 6.65 min
Part 15 of 50 done. in 6.95 min
Part 16 of 50 done. in 6.96 min
Part 17 of 50 done. in 6.27 min
Part 18 of 50 done. in 6.43 min
Part 19 of 50 done. in 6.84 min
Part 20 of 50 done. in 7.41 min
Part 21 of 50 done. in 7.35 min
Part 22 of 50 done. in 7.31 min
Part 23 of 50 done. in 7.08 min
Part 24 of 50 done. in 7.02 min
Part 25 of 50 done. in 6.85 min
Part 26 of 50 done. in 7.20 min
Part 27 of 50 done. in 6.23 min
Part 28 of 50 done. in 6.11 min
Part 29 of 50 done. in 6.34 min
Part 30 of 50 done. in 6.31 min
Part 31 of 50 done. in 6.34 min
Part 32 of 50 done. in 6.41 min
Part 33 of 50 done. in 6.34 min
Part 34 of 50 done. in 6.29 min
Part 35 of 50 done. in 6.42 min
```

```
Part 36 of 50 done. in 6.31 min
Part 37 of 50 done. in 6.26 min
Part 38 of 50 done. in 6.36 min
Part 39 of 50 done. in 7.19 min
Part 40 of 50 done. in 7.74 min
Part 41 of 50 done. in 8.17 min
Part 42 of 50 done. in 8.45 min
Part 43 of 50 done. in 7.54 min
Part 44 of 50 done. in 6.83 min
Part 45 of 50 done. in 6.84 min
Part 46 of 50 done. in 7.26 min
Part 47 of 50 done. in 7.29 min
Part 48 of 50 done. in 7.48 min
Part 49 of 50 done. in 6.37 min
Part 50 of 50 done. in 6.26 min
```

[ ]:
```
num = len(os.listdir('Data//embeddingBr//'))

with open('Data//embeddingBr//embeddings'+str(0),'rb') as f:
    dataD = pickle.load(f)

for idx in range(1,num):

  with open('Data//embeddingBr//embeddings'+str(idx),'rb') as f:
    mat = pickle.load(f)
    dataD=np.concatenate([dataD,mat],axis=0)
```

[ ]:
```
np.shape(dataD)
```

[ ]:
```
(400000, 768)
```

[ ]:
```
datay = df.iloc[:400000,:]
```

[ ]:
```
train_text, temp_text, train_labels, temp_labels = train_test_split(dataD,␣
 ↪datay['Label'],

                                                                    ␣
 ↪random_state=2018,

                                                                     test_size=0.
 ↪3,

                                                                    ␣
 ↪stratify=datay['Label'])

# we will use temp_text and temp_labels to create validation and test set
val_text, test_text, val_labels, test_labels = train_test_split(temp_text,␣
 ↪temp_labels,

                                                                ␣
 ↪random_state=2018,

                                                                  test_size=0.5,
```

```
↪stratify=temp_labels)
```

[ ]:
```
train_labels = tf.keras.utils.to_categorical(train_labels)
val_labels = tf.keras.utils.to_categorical(val_labels)
test_labels = tf.keras.utils.to_categorical(test_labels)
```

[ ]:
```
train_data = tf.data.Dataset.from_tensor_slices((train_text, train_labels))
train_data = train_data.shuffle(5000).batch(128)


val_data = tf.data.Dataset.from_tensor_slices((val_text, val_labels))
val_data = val_data.shuffle(5000).batch(128)
```

[ ]:
```
input = tf.keras.layers.Input(shape=(768,), name='input_token', dtype='int32')
X = tf.keras.layers.Dense(768, activation='relu')(input)
X = tf.keras.layers.Dropout(0.2)(X)
X = tf.keras.layers.BatchNormalization()(X)
X = tf.keras.layers.BatchNormalization()(X)
X = tf.keras.layers.Dense(128, activation='relu')(X)
X = tf.keras.layers.Dropout(0.2)(X)
X = tf.keras.layers.BatchNormalization()(X)
X = tf.keras.layers.Dense(2, activation='softmax')(X)
model = tf.keras.Model(inputs=input, outputs = X)
```

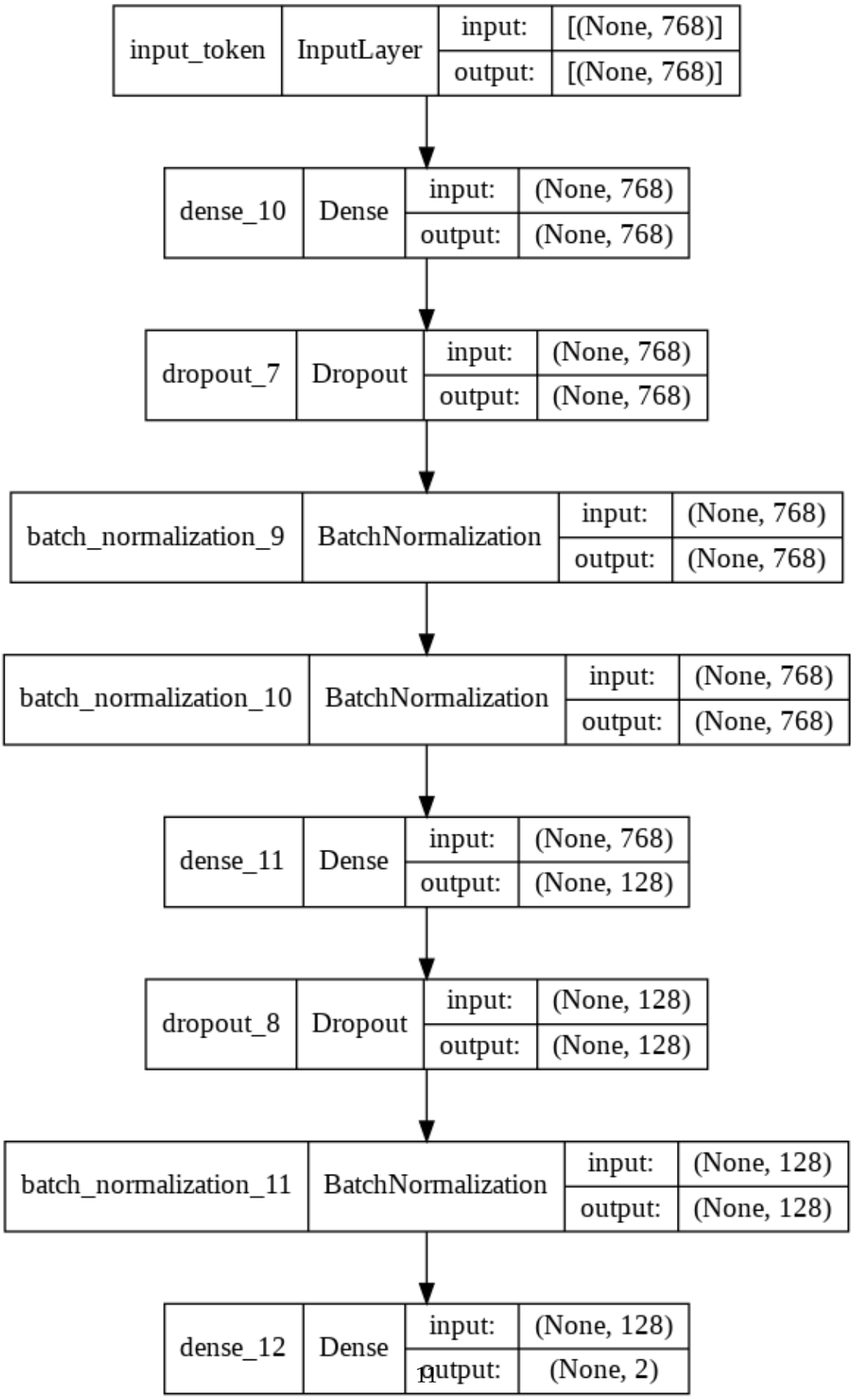
[ ]:
```
model.summary()
```

```
Model: "model_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_token (InputLayer)    [(None, 768)]             0

 dense_10 (Dense)            (None, 768)               590592

 dropout_7 (Dropout)         (None, 768)               0

 batch_normalization_9 (Batc  (None, 768)              3072
 hNormalization)

 batch_normalization_10 (Bat  (None, 768)              3072
 chNormalization)

 dense_11 (Dense)            (None, 128)               98432

 dropout_8 (Dropout)         (None, 128)               0

 batch_normalization_11 (Bat  (None, 128)              512
 chNormalization)
```

```
 dense_12 (Dense)            (None, 2)                 258

=================================================================
Total params: 695,938
Trainable params: 692,610
Non-trainable params: 3,328
_________________________________________________________________
```

[ ]:
```
from keras.utils.vis_utils import plot_model
plot_model(model, show_shapes=True, show_layer_names=True)
```

[ ]:


input_token
InputLayer
input: [(None, 768)]
output: [(None, 768)]
dense_10
Dense
input: (None, 768)
output: (None, 768)
dropout_7
Dropout
input: (None, 768)
output: (None, 768)
batch_normalization_9
BatchNormalization
input: (None, 768)
output: (None, 768)
batch_normalization_10
BatchNormalization
input: (None, 768)
output: (None, 768)
dense_11
Dense
input: (None, 768)
output: (None, 128)
dropout_8
Dropout
input: (None, 128)
output: (None, 128)
batch_normalization_11
BatchNormalization
input: (None, 128)
output: (None, 128)
dense_12
Dense
input: (None, 128)
output: (None, 2)

[ ]:
```
filepath="BERT5.hdf5"
checkpoint = ModelCheckpoint(filepath,␣
 ↪monitor='val_loss',verbose=1,save_best_only=True, mode='min')
ES =tf.keras.callbacks.
 ↪EarlyStopping(monitor="val_loss",patience=patience,verbose=1,mode="min",restore_best_weight
# pre = tf.keras.metrics.Precision()
f1 = tfa.metrics.F1Score(num_classes=2, average="macro")
callbacks_list = [checkpoint,ES]
model.compile(loss='binary_crossentropy', optimizer='adam' ,metrics=[f1])
```

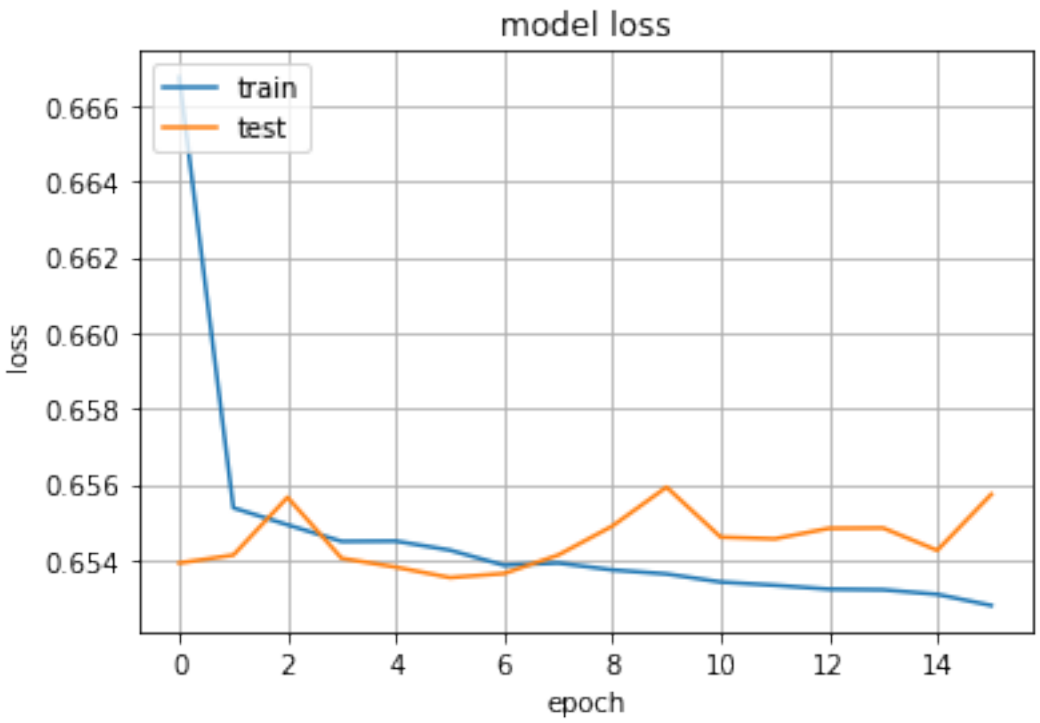
[ ]:
```
history = model.fit(train_data, validation_data=val_data,␣
 ↪epochs=epoch,verbose=1, callbacks = callbacks_list)
```

```
Epoch 1/30
2177/2188 [============================>.] - ETA: 0s - loss: 0.6668 - f1_score:
0.4900
Epoch 00001: val_loss improved from inf to 0.65392, saving model to BERT5.hdf5
2188/2188 [==============================] - 13s 5ms/step - loss: 0.6668 -
f1_score: 0.4900 - val_loss: 0.6539 - val_f1_score: 0.4801
Epoch 2/30
2187/2188 [============================>.] - ETA: 0s - loss: 0.6554 - f1_score:
0.4819
Epoch 00002: val_loss did not improve from 0.65392
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6554 -
f1_score: 0.4819 - val_loss: 0.6541 - val_f1_score: 0.5259
Epoch 3/30
2184/2188 [============================>.] - ETA: 0s - loss: 0.6550 - f1_score:
0.4825
Epoch 00003: val_loss did not improve from 0.65392
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6549 -
f1_score: 0.4827 - val_loss: 0.6557 - val_f1_score: 0.5259
Epoch 4/30
2184/2188 [============================>.] - ETA: 0s - loss: 0.6545 - f1_score:
0.4851
Epoch 00004: val_loss did not improve from 0.65392
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6545 -
f1_score: 0.4851 - val_loss: 0.6540 - val_f1_score: 0.5259
Epoch 5/30
2185/2188 [============================>.] - ETA: 0s - loss: 0.6545 - f1_score:
0.4840
Epoch 00005: val_loss improved from 0.65392 to 0.65381, saving model to
BERT5.hdf5
2188/2188 [==============================] - 12s 5ms/step - loss: 0.6545 -
f1_score: 0.4840 - val_loss: 0.6538 - val_f1_score: 0.5262
Epoch 6/30
2182/2188 [============================>.] - ETA: 0s - loss: 0.6543 - f1_score:
```

```
0.4847
Epoch 00006: val_loss improved from 0.65381 to 0.65353, saving model to
BERT5.hdf5
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6543 -
f1_score: 0.4848 - val_loss: 0.6535 - val_f1_score: 0.4411
Epoch 7/30
2178/2188 [============================>.] - ETA: 0s - loss: 0.6539 - f1_score:
0.4849
Epoch 00007: val_loss did not improve from 0.65353
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6539 -
f1_score: 0.4851 - val_loss: 0.6536 - val_f1_score: 0.4838
Epoch 8/30
2180/2188 [============================>.] - ETA: 0s - loss: 0.6540 - f1_score:
0.4866
Epoch 00008: val_loss did not improve from 0.65353
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6539 -
f1_score: 0.4867 - val_loss: 0.6541 - val_f1_score: 0.5262
Epoch 9/30
2180/2188 [============================>.] - ETA: 0s - loss: 0.6538 - f1_score:
0.4858
Epoch 00009: val_loss did not improve from 0.65353
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6537 -
f1_score: 0.4859 - val_loss: 0.6549 - val_f1_score: 0.5267
Epoch 10/30
2182/2188 [============================>.] - ETA: 0s - loss: 0.6536 - f1_score:
0.4866
Epoch 00010: val_loss did not improve from 0.65353
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6536 -
f1_score: 0.4866 - val_loss: 0.6559 - val_f1_score: 0.5260
Epoch 11/30
2187/2188 [============================>.] - ETA: 0s - loss: 0.6534 - f1_score:
0.4866
Epoch 00011: val_loss did not improve from 0.65353
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6534 -
f1_score: 0.4867 - val_loss: 0.6546 - val_f1_score: 0.5257
Epoch 12/30
2177/2188 [============================>.] - ETA: 0s - loss: 0.6533 - f1_score:
0.4874
Epoch 00012: val_loss did not improve from 0.65353
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6533 -
f1_score: 0.4876 - val_loss: 0.6546 - val_f1_score: 0.5243
Epoch 13/30
2186/2188 [============================>.] - ETA: 0s - loss: 0.6532 - f1_score:
0.4879
Epoch 00013: val_loss did not improve from 0.65353
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6532 -
f1_score: 0.4879 - val_loss: 0.6548 - val_f1_score: 0.5267
Epoch 14/30
```

```
2180/2188 [============================>.] - ETA: 0s - loss: 0.6532 - f1_score:
0.4894
Epoch 00014: val_loss did not improve from 0.65353
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6532 -
f1_score: 0.4895 - val_loss: 0.6548 - val_f1_score: 0.5267
Epoch 15/30
2184/2188 [============================>.] - ETA: 0s - loss: 0.6531 - f1_score:
0.4877
Epoch 00015: val_loss did not improve from 0.65353
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6531 -
f1_score: 0.4878 - val_loss: 0.6543 - val_f1_score: 0.5257
Epoch 16/30
2182/2188 [============================>.] - ETA: 0s - loss: 0.6528 - f1_score:
0.4886
Epoch 00016: val_loss did not improve from 0.65353
Restoring model weights from the end of the best epoch: 6.
2188/2188 [==============================] - 11s 5ms/step - loss: 0.6528 -
f1_score: 0.4888 - val_loss: 0.6557 - val_f1_score: 0.5271
Epoch 00016: early stopping
```

[ ]:
```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.grid()
plt.show()
```



[ ]:
```
from keras.models import load_model
model = load_model("BERT5.hdf5")
```

[ ]:
```
test_data = tf.data.Dataset.from_tensor_slices((test_text))
test_data = test_data.shuffle(5000).batch(128)
```

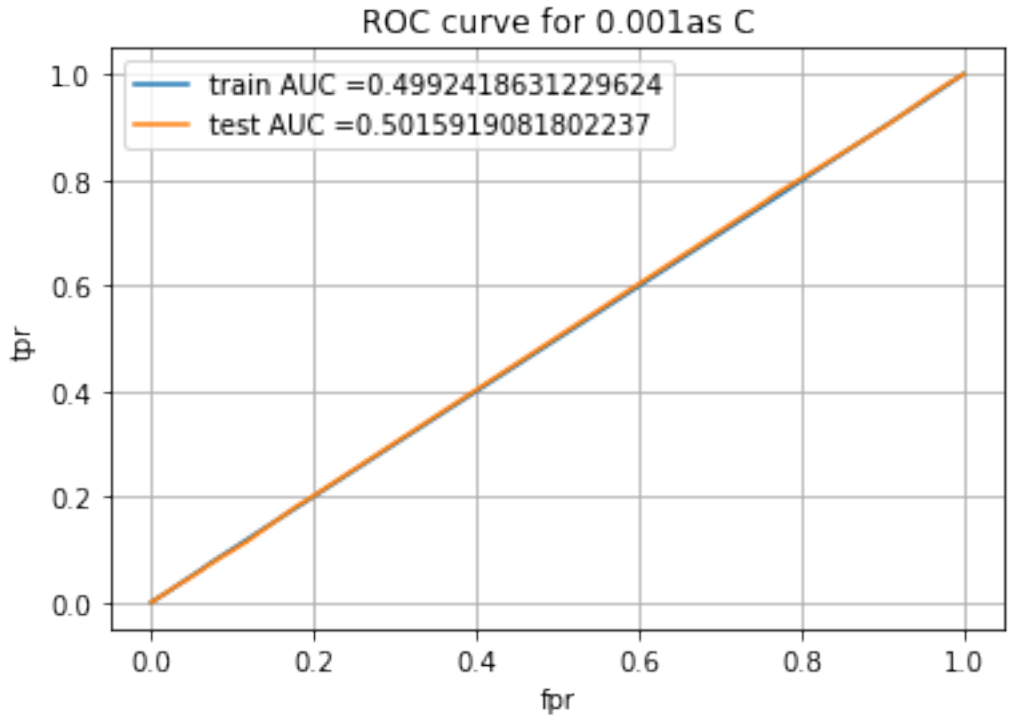
[ ]:
```
model.evaluate(test_data)
```

```
469/469 [==============================] - 2s 3ms/step - loss: 0.6543 -
f1_score: 0.4374
```

[ ]:
```
[0.6542547941207886, 0.43743130564689636]
```

[ ]:
```
y_pr_ts = model.predict(test_data)[:,0]
y_pred_tr = model.predict(train_data)[:,0]
y_ts = test_labels[:,0]
y_tr = train_labels[:,0]
from sklearn.metrics import␣
 ↪roc_curve,auc,confusion_matrix,accuracy_score,precision_score,recall_score,f1_score

train_fpr, train_tpr, tr_thresholds = roc_curve(y_tr, y_pred_tr)
test_fpr, test_tpr, te_thresholds = roc_curve(y_ts, y_pr_ts)
plt.plot(train_fpr, train_tpr, label="train AUC␣
 ↪="+str(auc(train_fpr,train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
```

```
plt.xlabel("fpr")
plt.ylabel("tpr")
plt.title('ROC curve for '+str (0.001)+'as C')
plt.legend()
plt.grid()
plt.show()
```



[ ]:
```
# This section of code where ever implemented is taken from sample kNN python␣
↪notebook

def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is very␣
↪high
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for␣
↪threshold", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
```

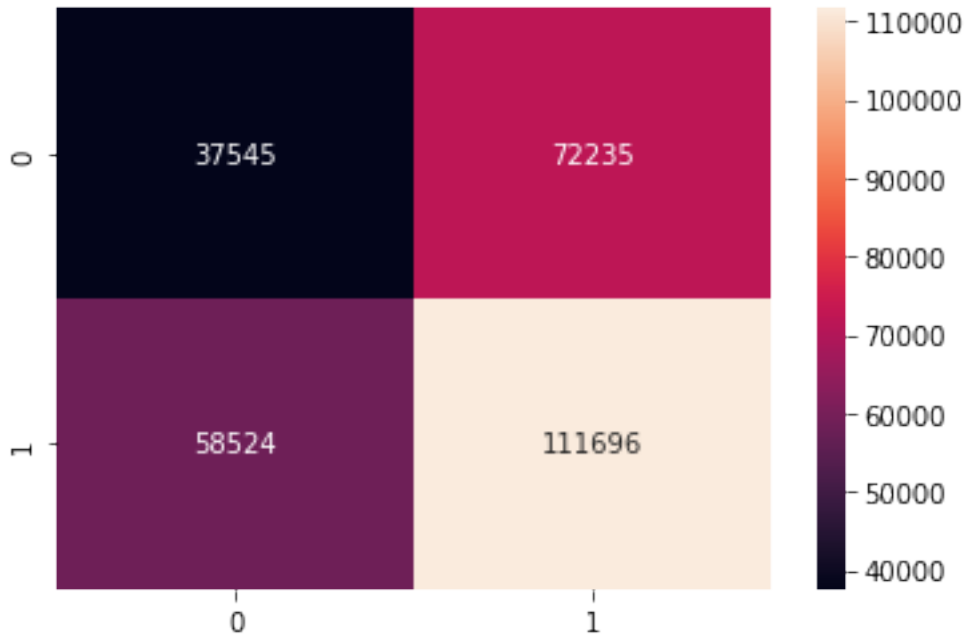
```
            predictions.append(0)
    return predictions


print('test')
best_ts_thres = find_best_threshold(te_thresholds, test_fpr, test_tpr)

print('train')
best_tr_thres = find_best_threshold(tr_thresholds, train_fpr, train_tpr)
```

```
test
the maximum value of tpr*(1-fpr) 0.22691762915098726 for threshold 0.715
train
the maximum value of tpr*(1-fpr) 0.2244170848827466 for threshold 0.721
```

[ ]:
```
print('Train Confusion Matrix')
cm2 = pd.DataFrame(confusion_matrix(y_tr, predict_with_best_t(y_pred_tr,␣
 ↪best_tr_thres)), range(2),range(2))
sns.heatmap(cm2, annot=True,fmt='g')
```

```
Train Confusion Matrix
```

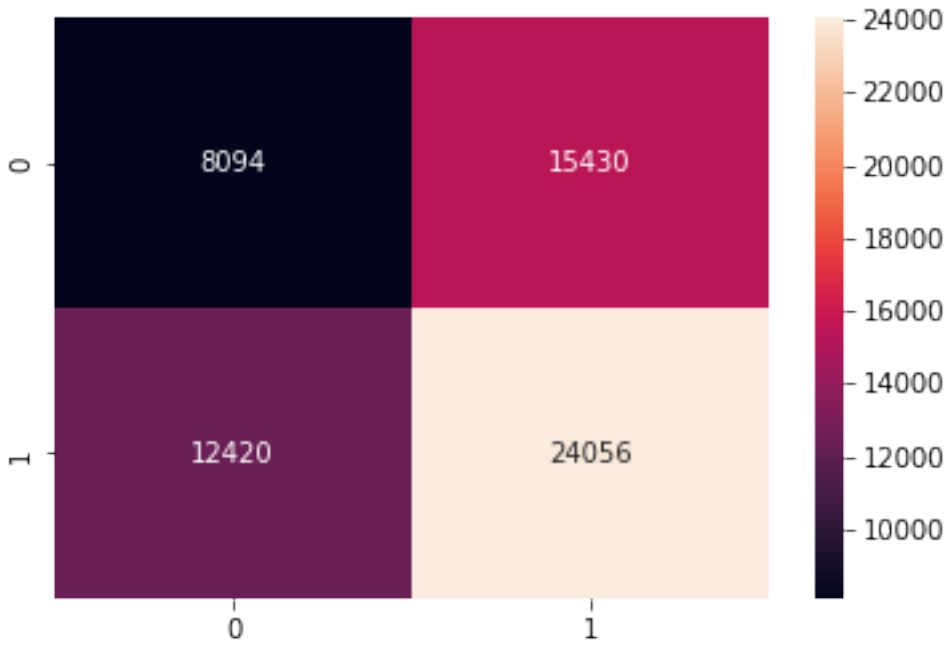
[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2404235ed0>
```



[ ]:
```
print('Test Confusion Matrix')
cm2 = pd.DataFrame(confusion_matrix(y_ts, predict_with_best_t(y_pr_ts,␣
 ↪best_ts_thres)), range(2),range(2))
sns.heatmap(cm2, annot=True,fmt='g')
```

```
Test Confusion Matrix
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2404606f90>
```



[ ]:
```
acc=accuracy_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100
ps=precision_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100
rc=recall_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100
f1=f1_score(y_ts, predict_with_best_t(y_pr_ts, best_ts_thres))*100

print("Accuracy on test set: %0.2f%%"%(acc))
print("Precision on test set: %0.2f%%"%(ps))
print("recall score on test set: %0.2f%%"%(rc))
print("f1 score on test set: %0.2f%%"%(f1))
```

```
Accuracy on test set: 53.58%
Precision on test set: 60.92%
recall score on test set: 65.95%
f1 score on test set: 63.34%
```