

```
!pip install modelzoo-client[transformers]
```

```
    Downloading click-7.1-py2.py3-none-any.whl (82 kB)
    |████████| 82 kB 859 kB/s
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from
Collecting yaspin==0.16.0
    Downloading yaspin-0.16.0-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: termcolor==1.1.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (f
Collecting names==0.3.0
    Downloading names-0.3.0.tar.gz (789 kB)
    |████████| 789 kB 47.1 MB/s
Requirement already satisfied: torch in /usr/local/lib/python3.7/dist-packages (from
Collecting transformers>=2.10.0
    Downloading transformers-4.18.0-py3-none-any.whl (4.0 MB)
    |████████| 4.0 MB 39.8 MB/s
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages
Collecting sacremoses
    Downloading sacremoses-0.0.49-py3-none-any.whl (895 kB)
    |████████| 895 kB 46.6 MB/s
Collecting huggingface-hub<1.0,>=0.1.0
    Downloading huggingface_hub-0.5.1-py3-none-any.whl (77 kB)
    |████████| 77 kB 3.5 MB/s
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7/dist-pa
Collecting tokenizers!=0.11.3,<0.13,>=0.11.1
    Downloading tokenizers-0.11.6-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_6
    |████████| 6.5 MB 46.4 MB/s
Collecting pyyaml>=5.1
    Downloading PyYAML-6.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux
    |████████| 596 kB 56.1 MB/s
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: urllib3!=1.25.0,!>=1.25.1,<1.26,>=1.21.1 in /usr/local
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from s
Building wheels for collected packages: names
  Building wheel for names (setup.py) ... done
  Created wheel for names: filename=names-0.3.0-py3-none-any.whl size=803699 sha256=
  Stored in directory: /root/.cache/pip/wheels/05/ea/68/92f6b0669e478af9b7c3c524520d

Successfully built names
Installing collected packages: pyyaml, click, yaspin, tokenizers, sacremoses, names,
Attempting uninstall: pyyaml
  Found existing installation: PyYAML 3.13
  Uninstalling PyYAML-3.13:
    Successfully uninstalled PyYAML-3.13
  Attempting uninstall: click
```

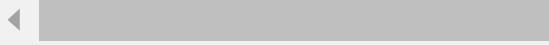
```
Attempting uninstall. Click
```

```
Found existing installation: click 7.1.2
```

```
Uninstalling click-7.1.2:
```

```
Successfully uninstalled click-7.1.2
```

```
Successfully installed click-7.1 colorama-0.4.3 huggingface-hub-0.5.1 modelzoo-clien
```



```
import tensorflow as tf  
tf.test.gpu_device_name()
```

```
'/device:GPU:0'
```

```
import transformers
```

```
!pip install tensorflow-addons  
!pip install tensorflow
```

```
Collecting tensorflow-addons
```

```
  Downloading tensorflow_addons-0.16.1-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.1 MB)  
|██████████| 1.1 MB 10.1 MB/s
```

```
Requirement already satisfied: typeguard>=2.7 in /usr/local/lib/python3.7/dist-packages  
Installing collected packages: tensorflow-addons
```

```
Successfully installed tensorflow-addons-0.16.1
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (2.8.0)
```

```
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: keras<2.9,>=2.8.0rc0 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.7/dist-packages
```

```
Collecting tf-estimator-nightly==2.8.0.dev2021122109
```

```
  Downloading tf_estimator_nightly-2.8.0.dev2021122109-py2.py3-none-any.whl (462 kB)  
|██████████| 462 kB 6.5 MB/s
```

```
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: gast>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from tf-estimator-nightly==2.8.0.dev2021122109)
```

```
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (from tf-estimator-nightly==2.8.0.dev2021122109)
```

```
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from tf-estimator-nightly==2.8.0.dev2021122109)
```

```
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (from tf-estimator-nightly==2.8.0.dev2021122109)
```

```
Requirement already satisfied: tensorboard<2.9,>=2.8 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.7/dist-packages (from tf-estimator-nightly==2.8.0.dev2021122109)
```

```
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: flatbuffers>=1.12 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: tensorflow-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from tensorflow<2.8.0.dev2021122109)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from tensorflow<2.8.0.dev2021122109)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages
Installing collected packages: tf-estimator-nightly
Successfully installed tf-estimator-nightly-2.8.0.dev2021122109
```

```
pip install -U bert-serving-server bert-serving-client
```

```
Collecting bert-serving-server
  Downloading bert_serving_server-1.10.0-py3-none-any.whl (61 kB)
    |██████████| 61 kB 277 kB/s
Collecting bert-serving-client
  Downloading bert_serving_client-1.10.0-py2.py3-none-any.whl (28 kB)
Requirement already satisfied: termcolor>=1.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pyzmq>=17.1.0 in /usr/local/lib/python3.7/dist-packages (from bert-servi
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from bert-servi
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from bert-servi
Collecting GPUUtil>=1.3.0
  Downloading GPUUtil-1.4.0.tar.gz (5.5 kB)
Building wheels for collected packages: GPUUtil
  Building wheel for GPUUtil (setup.py) ... done
  Created wheel for GPUUtil: filename=GPUUtil-1.4.0-py3-none-any.whl size=7411 sha256=b1ef
  Stored in directory: /root/.cache/pip/wheels/6e/f8/83/534c52482d6da64622ddbf72cd93c35c
Successfully built GPUUtil
Installing collected packages: GPUUtil, bert-serving-server, bert-serving-client
Successfully installed GPUUtil-1.4.0 bert-serving-client-1.10.0 bert-serving-server-1.10.0
```

```
# Importing necessary Libraries:  
import pandas as pd  
# from transformers import TFBertModel, BertTokenizer  
seed_value = 29  
import os  
os.environ['PYTHONHASHSEED'] = str(seed_value)  
import random  
random.seed(seed_value)  
import numpy as np  
np.random.seed(seed_value)  
np.set_printoptions(precision=2)
```

```

import tensorflow as tf
tf.random.set_seed(seed_value)
# import tensorflow_addons as tfa
import tensorflow.keras as keras
# import tensorflow.keras.layers as layers
# from tensorflow.keras.callbacks import ModelCheckpoint
import re
import matplotlib.pyplot as plt
from sklearn.metrics import auc, roc_curve

# Uploading dataset to google collab
from google.colab import files
uploaded = files.upload()

```

Choose Files clean_data.csv

- **clean_data.csv**(text/csv) - 36143407 bytes, last modified: 4/6/2022 - 100% done
Saving clean_data.csv to clean_data.csv

```

# Converting uploaded dataset to DataFrame
import io
data = pd.read_csv(io.BytesIO(uploaded['clean_data.csv']))
# Dataset is now stored in a Pandas Dataframe

```

data.head

		type	pc
0	INFJ	intj moment sportscenter top ten play prankswh...	
1	ENTP	finding lack post alarmingsex boring position ...	
2	INTP	good one course say know blessing cursedoes ab...	
3	INTJ	dear intp enjoyed conversation day esoteric ga...	
4	ENTJ	youre firedthats another silly misconception a...	
...
8670	ISFP	always think cat fi doms reason website become...	
8671	ENFP	soif thread already exists someplace else heck...	
8672	INTP	many question thing would take purple pill pic...	
8673	INFP	conflicted right come wanting child honestly m...	
8674	INFP	long since personalitycafe although seem chang...	

	type of encoding
0	8
1	3
2	11
3	10
4	2
...	...
8670	13
8671	1
8672	11
8673	9
8674	9

[8675 rows x 3 columns]>

```
# Pre-Processsing the data
import re
def text_preprocessing(text):
    text = text.lower()
    text = re.sub('.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text.encode('ascii', 'ignore').decode('ascii')
    if text.startswith("''"):
        text = text[1:-1]
    return text

# Pre-Processing the Labels from 1 column to 4 and converting categorical data into Numeric.
N_AXIS = 4
MAX_SEQ_LEN = 128
BERT_NAME = 'bert-base-uncased'
'''

EMOTIONAL AXES:
Introversion (I) - Extroversion (E)
Intuition (N) - Sensing (S)
Thinking (T) - Feeling (F)
Judging (J) - Perceiving (P)
'''
axes = ["I-E","N-S","T-F","J-P"]
classes = {"I":0, "E":1, # axis 1
           "N":0,"S":1, # axis 2
           "T":0, "F":1, # axis 3
           "J":0,"P":1} # axis 4

# Splitting data into test and train and pre-processing the input data so that it can be pass
train_n=6624
val_n=1024
test_n=1024
# data = pd.read_csv("mbti_1.csv")
data = data.sample(frac=1)
labels = []
print(data)
for personality in data["type"]:
    pers_vect = []
    for p in personality:
        pers_vect.append(classes[p])
    labels.append(pers_vect)
sentences = data["posts"].apply(str).apply(lambda x: text_preprocessing(x))
labels = np.array(labels, dtype="float32")
train_sentences = sentences[:train_n]
```

```
y_train = labels[:train_n]
val_sentences = sentences[train_n:train_n+val_n]
y_val = labels[train_n:train_n+val_n]
test_sentences = sentences[train_n+val_n:train_n+val_n+test_n]
y_test = labels[train_n+val_n:train_n+val_n+test_n]
```

	type	posts	\
4420	INFP	guess preparing wwww start whole country try...	
7570	ENTJ	like whenever start talking anything abstract ...	
2807	INFP	really strange fear shiny jewelry metal especi...	
463	ISTP	exactly cheerssame curiousahaha ditto picture m...	
3060	INFJ	may pop struggling perfectionism long time wou...	
...
920	INFP	excellent example explanation would able provi...	
864	INTP	thinking samewe sometimes mentioned paarthuna...	
808	ISTP	associate professional flight technologynormal...	
6380	INFJ	love bondi sorry lostmemories madekept itso ge...	
8149	INTJ	posted forgive already posted including classi...	

	type of encoding	
4420		9
7570		2
2807		9
463		15
3060		8
...		...
920		9
864		11
808		15
6380		8
8149		10

[8675 rows x 3 columns]

```
print(train_sentences.shape,y_train.shape)
```

(6624,) (6624, 4)

```
print(val_sentences.shape,y_val.shape)
```

(1024,) (1024, 4)

```
print(test_sentences.shape,y_test.shape)
```

(1024,) (1024, 4)

```
vocab_size = 10000
trunc_type = "post"
pad_type = "post"
oov_tok = "<OOV>"
maxlen = 1500
```

```
data['posts'] = [str(i) for i in data['posts'].values]

import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

tokenizer = Tokenizer(num_words=vocab_size, filters='!"#$%&()*+,-./:;=>?@[\\]^_`{|}~', lower=True)
tokenizer.fit_on_texts(data['posts'].values)

maxlen = 1500
train_sequences = tokenizer.texts_to_sequences(train_sentences)
train_padded = pad_sequences(train_sequences, maxlen = maxlen, truncating = trunc_type, padding = 'post')

val_sequences = tokenizer.texts_to_sequences(val_sentences)
val_padded = pad_sequences(val_sequences, maxlen = maxlen, truncating = trunc_type, padding = 'post')

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Bidirectional, Flatten, Dropout, Activation

def create_model():
    op = tf.keras.optimizers.Adam(learning_rate=0.00001)

    model = Sequential()
    model.add(Embedding(vocab_size, 256, input_length=maxlen))
    model.add(LSTM(60, return_sequences=True))
    model.add(GlobalMaxPool1D())
    model.add(Dropout(0.1))
    model.add(Dense(50, activation="relu"))
    model.add(Dropout(0.1))
    model.add(Dense(4, activation="softmax"))
    model.compile(loss='categorical_crossentropy', optimizer=op, metrics=['accuracy'])

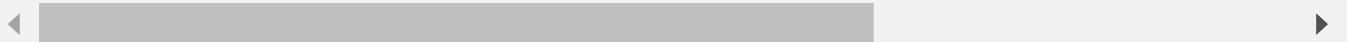
    return model

model = create_model()

history = model.fit(train_padded, y_train, epochs = 20, batch_size=64, verbose = 1, validation_data=(val_padded, y_val))

Epoch 1/20
104/104 [=====] - 13s 71ms/step - loss: 2.0998 - accuracy: 0.34
Epoch 2/20
104/104 [=====] - 7s 64ms/step - loss: 2.0846 - accuracy: 0.30
Epoch 3/20
104/104 [=====] - 7s 67ms/step - loss: 2.0679 - accuracy: 0.25
Epoch 4/20
104/104 [=====] - 7s 65ms/step - loss: 2.0450 - accuracy: 0.21
```

```
Epoch 5/20
104/104 [=====] - 7s 65ms/step - loss: 2.0127 - accuracy: 0.171
Epoch 6/20
104/104 [=====] - 8s 78ms/step - loss: 1.9700 - accuracy: 0.161
Epoch 7/20
104/104 [=====] - 7s 65ms/step - loss: 1.9207 - accuracy: 0.151
Epoch 8/20
104/104 [=====] - 7s 68ms/step - loss: 1.8969 - accuracy: 0.161
Epoch 9/20
104/104 [=====] - 7s 66ms/step - loss: 1.8929 - accuracy: 0.174
Epoch 10/20
104/104 [=====] - 7s 68ms/step - loss: 1.9137 - accuracy: 0.171
Epoch 11/20
104/104 [=====] - 7s 66ms/step - loss: 1.9296 - accuracy: 0.191
```



```
hist = pd.DataFrame(history.history)
print(hist['loss'],hist['val_loss'])
hist["epoch"] = history.epoch
# Plotting the Train Vs Val Error:
def plot_history():
    plt.figure()
    plt.xlabel('Epoch')
    plt.ylabel('categorical Crossentropy')
    plt.plot(hist['epoch'], hist['loss'], label='Train Error')
    plt.plot(hist['epoch'], hist['val_loss'], label='Val Error')
    plt.legend()
    # plt.ylim([0, 50])

plot_history()
```

```

0    2.099830
1    2.084620
2    2.067932
3    2.044987
4    2.012710
5    1.970003
6    1.920659
7    1.896900
8    1.892912
9    1.913661
10   1.929633
Name: loss, dtype: float64 0      2.086461
1    2.070916
2    2.052410
3    2.024101
4    1.984623
5    1.935466

maxlen = 1500
test_sequences = tokenizer.texts_to_sequences(test_sentences)
test_padded = pad_sequences(test_sequences, maxlen = maxlen, truncating = trunc_type, padding

predictions = model.predict(test_padded)
model.evaluate(test_padded, y_test, batch_size=32)

32/32 [=====] - 1s 23ms/step - loss: 1.9547 - accuracy: 0.1475
[1.9547337293624878, 0.1474609375]
  |          \  /  |  |

from tensorflow.keras.optimizers import Adam
def create_model():
    model = Sequential()
    model.add(Embedding(vocab_size, 256, input_length=maxlen))
        # model.add(SimpleRNN(EMBEDDING_VECTOR_LENGTH, dropout=DROPOUT, recurrent_dropout=DROPOUT))
        # model.add(GRU(EMBEDDING_VECTOR_LENGTH, dropout=DROPOUT, recurrent_dropout=DROPOUT))
    model.add(LSTM(50,dropout=0.1,recurrent_dropout=0.1,activation="sigmoid",kernel_initializer="he_normal"))
        # model.add(Bidirectional(LSTM(EMBEDDING_VECTOR_LENGTH, dropout=DROPOUT, recurrent_dropout=DROPOUT)))
    model.add(Dense(4, activation="sigmoid"))
    optimizer = Adam(lr=0.01, beta_1=0.9, beta_2=0.999, epsilon=1e-8)
    model.compile(loss="binary_crossentropy", optimizer=optimizer, metrics=["accuracy"])
    print(model.summary())

create_model()

WARNING:tensorflow:Layer lstm_1 will not use cuDNN kernels since it doesn't meet the criteria.
Model: "sequential_1"

-----  

Layer (type)           Output Shape        Param #
-----  

embedding_1 (Embedding) (None, 1500, 256)     2560000  

lstm_1 (LSTM)          (None, 50)            61400

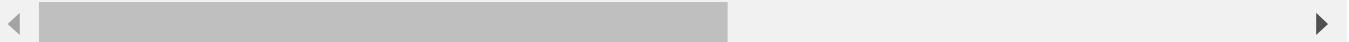
```

dense_2 (Dense)	(None, 4)	204
-----------------	-----------	-----

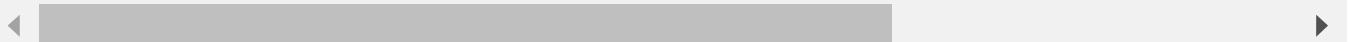
```
=====
Total params: 2,621,604
Trainable params: 2,621,604
Non-trainable params: 0
```

```
None
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105: UserWarning: The
super(Adam, self).__init__(name, **kwargs)
```



```
model.fit(train_padded, y_train, epochs=20, batch_size=128, verbose = 1, validation_data = (val_
Epoch 1/20
52/52 [=====] - 5s 94ms/step - loss: 1.9471 - accuracy: 0.2006
Epoch 2/20
52/52 [=====] - 4s 86ms/step - loss: 1.9517 - accuracy: 0.2065
Epoch 3/20
52/52 [=====] - 4s 85ms/step - loss: 1.9652 - accuracy: 0.2073
Epoch 4/20
52/52 [=====] - 4s 85ms/step - loss: 1.9756 - accuracy: 0.2115
<keras.callbacks.History at 0x7f60806ba710>
```



```
hist = pd.DataFrame(history.history)
print(hist['loss'],hist['val_loss'])
hist["epoch"] = history.epoch
# Plotting the Train Vs Val Error:
def plot_history():
    plt.figure()
    plt.xlabel('Epoch')
    plt.ylabel('Binary_Crossentropy')
    plt.plot(hist['epoch'], hist['loss'], label='Train Error')
    plt.plot(hist['epoch'], hist['val_loss'], label='Val Error')
    plt.legend()
    # plt.ylim([0, 50])

plot_history()
```

```

0      2.099830
1      2.084620
2      2.067932
3      2.044987
4      2.012710
5      1.970003
6      1.920659
7      1.896900
8      1.892912
9      1.913661
10     1.929633
Name: loss, dtype: float64 0      2.086461
1      2.070916
2      2.052410
3      2.024101
4      1.984623
5      1.935466
6      1.887340
7      1.873886
8      1.879149
9      1.890867
10     1.903480
Name: val_loss, dtype: float64

```



```

maxlen = 1500
test_sequences = tokenizer.texts_to_sequences(test_sentences)
test_padded = pad_sequences(test_sequences, maxlen = maxlen, truncating = trunc_type, padding = padding)

predictions = model.predict(test_padded)
model.evaluate(test_padded, y_test, batch_size=32)

32/32 [=====] - 1s 24ms/step - loss: 1.9940 - accuracy: 0.1475
[1.9939548969268799, 0.1474609375]

```

```

# This function prepares input to be fed into the BERT Model.
def prepare_bert_input(sentences, seq_len, bert_name):
    tokenizer = BertTokenizer.from_pretrained(bert_name)
    encodings = tokenizer(sentences.tolist(), truncation=True, padding='max_length',
                          max_length=seq_len)
    input = [np.array(encodings["input_ids"]), np.array(encodings["token_type_ids"]),
            np.array(encodings["attention_mask"])]
    return input

```

```
from transformers import TFBertModel, BertTokenizer
```

```
# Converting Input train,val and test data into a format that can be understand by the BERT M
X_train = prepare_bert_input(train_sentences, MAX_SEQ_LEN, BERT_NAME)
```

```
X_val = prepare_bert_input(val_sentences, MAX_SEQ_LEN, BERT_NAME)
X_test = prepare_bert_input(test_sentences, MAX_SEQ_LEN, BERT_NAME)
```

Downloading: 100% 226k/226k [00:00<00:00, 904kB/s]

Downloading: 100% 28.0/28.0 [00:00<00:00, 656B/s]

Downloading: 100% 570/570 [00:00<00:00, 15.2kB/s]

```
import tensorflow.keras.layers as layers
```

```
# Creating the BERT Model.
```

```
input_ids = layers.Input(shape=(MAX_SEQ_LEN,), dtype=tf.int32, name='input_ids')
input_type = layers.Input(shape=(MAX_SEQ_LEN,), dtype=tf.int32, name='token_type_ids')
input_mask = layers.Input(shape=(MAX_SEQ_LEN,), dtype=tf.int32, name='attention_mask')
inputs = [input_ids, input_type, input_mask]
bert = TFBertModel.from_pretrained(BERT_NAME)
bert_outputs = bert(inputs)
last_hidden_states = bert_outputs.last_hidden_state
avg = layers.GlobalAveragePooling1D()(last_hidden_states)
output = layers.Dense(N_AXIS, activation="sigmoid")(avg)
model = keras.Model(inputs=inputs, outputs=output)
model.summary()
```

Downloading: 100%

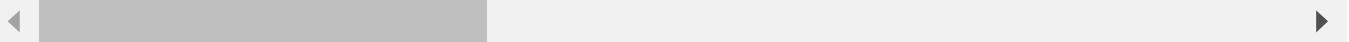
511M/511M [00:15<00:00, 35.7MB/s]

Some layers from the model checkpoint at bert-base-uncased were not used when initializing the model.
- This IS expected if you are initializing TFBertModel from the checkpoint of a model trained on a different task.
- This IS NOT expected if you are initializing TFBertModel from the checkpoint of a model trained on the same task as the one you are using now.
All the layers of TFBertModel were initialized from the model checkpoint at bert-base-uncased.
If your task is similar to the task the model of the checkpoint was trained on, you can initialize TFBertModel with
Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 128)]	0	[]
token_type_ids (InputLayer)	[(None, 128)]	0	[]

```
import tensorflow_addons as tfa
tf_bert_model (TFBertModel)      TFBertModelOutputWithTfOutput 109482240  ['input_ids', 'token_type_ids']
from tensorflow.keras.callbacks import ModelCheckpoint
# Defining Model Parameters:
max_epochs = 7
batch_size = 32
opt = tfa.optimizers.RectifiedAdam(learning_rate=3e-5)
loss = keras.losses.BinaryCrossentropy()
best_weights_file = "weights.h5"
auc = keras.metrics.AUC(multi_label=True, curve="ROC")
m_ckpt = ModelCheckpoint(best_weights_file, monitor='val_'+auc.name, mode='max', verbose=1,
                         save_weights_only=True, save_best_only=True)
model.compile(loss=loss, optimizer=opt, metrics=[auc, keras.metrics.BinaryAccuracy()])
# Fitting the Model:
history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=max_epochs,
    batch_size=batch_size,
    callbacks=[m_ckpt],
    verbose=1
)
Epoch 1/7
WARNING:tensorflow:Gradients do not exist for variables ['tf_bert_model/bert/pooler/dense/kernel', 'tf_bert_model/bert/pooler/dense/bias'].
WARNING:tensorflow:Gradients do not exist for variables ['tf_bert_model/bert/pooler/dense/kernel', 'tf_bert_model/bert/pooler/dense/bias'].
207/207 [=====] - ETA: 0s - loss: 0.5751 - auc: 0.5667 - binary_accuracy: 0.69912
Epoch 1: val_auc improved from -inf to 0.69912, saving model to weights.h5
207/207 [=====] - 207s 913ms/step - loss: 0.5751 - auc: 0.5667 - binary_accuracy: 0.69912
Epoch 2/7
207/207 [=====] - ETA: 0s - loss: 0.4863 - auc: 0.7525 - binary_accuracy: 0.8077
Epoch 2: val_auc improved from 0.69912 to 0.78468, saving model to weights.h5
207/207 [=====] - 197s 951ms/step - loss: 0.4863 - auc: 0.7525 - binary_accuracy: 0.78468
Epoch 3/7
207/207 [=====] - ETA: 0s - loss: 0.4440 - auc: 0.8077 - binary_accuracy: 0.8421
```

```
Epoch 3: val_auc improved from 0.78468 to 0.80093, saving model to weights.h5
207/207 [=====] - 198s 955ms/step - loss: 0.4440 - auc: 0.8077
Epoch 4/7
207/207 [=====] - ETA: 0s - loss: 0.4097 - auc: 0.8413 - binary
Epoch 4: val_auc did not improve from 0.80093
207/207 [=====] - 196s 945ms/step - loss: 0.4097 - auc: 0.8413
Epoch 5/7
207/207 [=====] - ETA: 0s - loss: 0.3612 - auc: 0.8821 - binary
Epoch 5: val_auc improved from 0.80093 to 0.80792, saving model to weights.h5
207/207 [=====] - 197s 953ms/step - loss: 0.3612 - auc: 0.8821
Epoch 6/7
207/207 [=====] - ETA: 0s - loss: 0.2953 - auc: 0.9227 - binary
Epoch 6: val_auc did not improve from 0.80792
207/207 [=====] - 196s 947ms/step - loss: 0.2953 - auc: 0.9227
Epoch 7/7
207/207 [=====] - ETA: 0s - loss: 0.2218 - auc: 0.9580 - binary
Epoch 7: val_auc did not improve from 0.80792
207/207 [=====] - 196s 947ms/step - loss: 0.2218 - auc: 0.9580
```



```
hist = pd.DataFrame(history.history)
```

```
print(hist['loss'],hist['val_loss'])
```

```
0    0.575080
1    0.486283
2    0.444014
3    0.409681
4    0.361189
5    0.295250
6    0.221820
Name: loss, dtype: float64 0    0.525129
1    0.476843
2    0.473904
3    0.461003
4    0.469215
5    0.505816
6    0.583501
Name: val_loss, dtype: float64
```

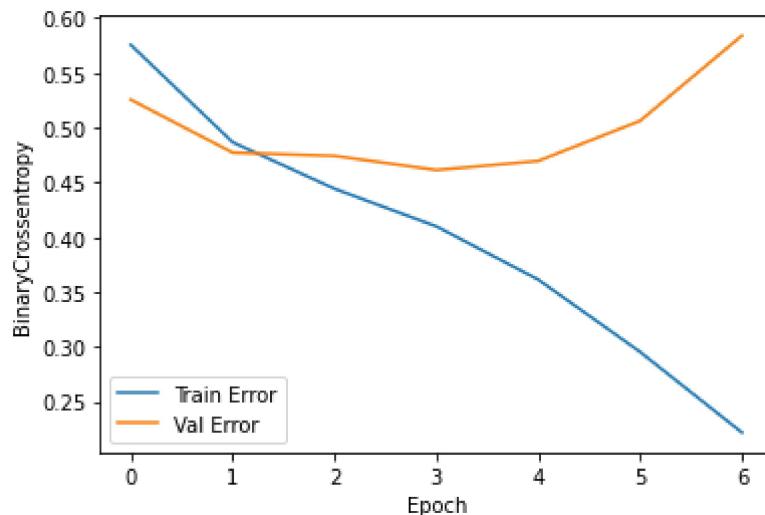
```
hist["epoch"] = history.epoch
```

```
# Plotting the Train Vs Val Error:
```

```
def plot_history():
    plt.figure()
    plt.xlabel('Epoch')
    plt.ylabel('BinaryCrossentropy')
    plt.plot(hist['epoch'], hist['loss'], label='Train Error')
    plt.plot(hist['epoch'], hist['val_loss'], label='Val Error')
    plt.legend()
```

```
# plt.ylim([0, 50])
```

```
plot_history()
```



```
loss = keras.losses.BinaryCrossentropy()
best_weights_file = "weights.h5"
model.load_weights(best_weights_file)
opt = tfa.optimizers.RectifiedAdam(learning_rate=3e-5)
model.compile(loss=loss, optimizer=opt, metrics=[keras.metrics.AUC(multi_label=True, curve="R",
                                                               keras.metrics.BinaryAccuracy())])
predictions = model.predict(X_test)
model.evaluate(X_test, y_test, batch_size=32)

32/32 [=====] - 12s 273ms/step - loss: 0.5019 - auc_1: 0.7945 -
[0.5019081234931946, 0.7945380210876465, 0.78125]
```



```
X_train[0].shape
```

```
(6624, 128)
```

```
y_train.shape
```

```
(6624, 4)
```

```
X_test[0].shape
```

```
(1024, 128)
```

```
predictions
```

```
array([[0.22, 0.05, 0.26, 0.3 ],
       [0.01, 0.03, 0.96, 0.97],
```

```
[0.39, 0.03, 0.07, 0.77],  
...,  
[0.11, 0.65, 0.2 , 0.07],  
[0.32, 0.34, 0.92, 0.62],  
[0.12, 0. , 0.03, 0.91]], dtype=float32)  
  
# Testing the Resukts using Random Text Input.  
s1 = "I like studying deep learning, playing football and my guitar, "  
     "and I love visit foreign cities all over the world."  
sentences = np.asarray([s1])  
enc_sentences = prepare_bert_input(sentences, MAX_SEQ_LEN, BERT_NAME)  
predictions = model.predict(enc_sentences)  
for sentence, pred in zip(sentences, predictions):  
    pred_axis = []  
    mask = (pred > 0.5).astype(bool)  
    for i in range(len(mask)):  
        if mask[i]:  
            pred_axis.append(axes[i][2])  
        else:  
            pred_axis.append(axes[i][0])  
    print('-- comment: '+sentence.replace("\n", "").strip() +  
          '\n-- personality: '+str(pred_axis) +  
          '\n-- scores:'+str(pred))  
  
-- comment: I like studying deep learning, playing football and my guitar, and I love vi  
-- personality: ['I', 'N', 'F', 'P']  
-- scores:[0.01 0.34 0.51 0.78]
```



