ASSIGNMENTS - 4

Assignment Date	9 October2022
Student Name	Anthony raj L
Team ID	PNT2022TMID25121
Maximum Marks	2 Marks

1.Import required library

In [1]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from keras.models import Model
from keras.layers import LSTM,Activation,Dense,Dropout,Input,Embedding
from keras.optimizers import RMSprop
from keras.preprocessing.text import Tokenizer
from keras_preprocessing import sequence
from keras.utils import to_categorical
from keras.models import load_model
```

2.Read dataset and do preprocessing

In [6]:

```
df=pd.read_csv('/content/spam.csv',delimiter=',',encoding='latin-1')
df.head()
```

Out[6]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

In [7]:

```
df.drop(['Unnamed: 2','Unnamed: 3','Unnamed: 4'],axis=1,inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   v1      5572 non-null   object
 1   v2      5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

In [

8]:

```
df.groupby(['v1']).size()
```

Out[8]:

```
v1
ham     4825
spam     747
dtype: int64
```

In [9]:

```
X=df.v2
Y=df.v1
le=LabelEncoder()
Y=le.fit_transform(Y)
Y=Y.reshape(-1,1)
```

In [10]:

```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.15)
```

In [11]:

```
max_words = 1000
max_len = 150
tok = Tokenizer(num_words=max_words)
tok.fit_on_texts(X_train)
sequences = tok.texts_to_sequences(X_train)
sequences_matrix = sequence.pad_sequences(sequences,maxlen=max_len)
```

3.Create Model and 4. Add layers (LSTM ,Dense-(Hidden Layers),Ouput)

In [12]:

```
inputs = Input(name='InputLayer',shape=[max_len])
layer = Embedding(max_words,50,input_length=max_len)(inputs)
layer = LSTM(64)(layer)
layer = Dense(256,name='FullyConnectedLayer1')(layer)
layer = Activation('relu')(layer)
layer = Dropout(0.5)(layer)
layer = Dense(1,name='OutputLayer')(layer)
layer = Activation('sigmoid')(layer)
```

5.Compile the model

In [1]:

3

```
model = Model(inputs=inputs,outputs=layer)
model.summary()
model.compile(loss='binary_crossentropy',optimizer=RMSprop(),metrics=['accuracy'])
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 InputLayer (InputLayer)     [(None, 150)]             0

 embedding (Embedding)       (None, 150, 50)           50000

 lstm (LSTM)                 (None, 64)                29440

 FullyConnectedLayer1 (Dense  (None, 256)              16640
 )

 activation (Activation)     (None, 256)               0

 dropout (Dropout)           (None, 256)               0

 OutputLayer (Dense)         (None, 1)                 257

 activation_1 (Activation)   (None, 1)                 0

=================================================================
Total params: 96,337
Trainable params: 96,337
Non-trainable params: 0
_________________________________________________________________
```

6.Fit the Model

In [1]:

4

```
model.fit(sequences_matrix,Y_train,batch_size=128,epochs=10,
          validation_split=0.2)
```

```
Epoch 1/10
30/30 [==============================] - 12s 288ms/step - loss: 0.3170 - acc
uracy: 0.8809 - val_loss: 0.0929 - val_accuracy: 0.9694
Epoch 2/10
30/30 [==============================] - 8s 261ms/step - loss: 0.0887 - accu
racy: 0.9789 - val_loss: 0.0548 - val_accuracy: 0.9916
Epoch 3/10
30/30 [==============================] - 9s 309ms/step - loss: 0.0490 - accu
racy: 0.9855 - val_loss: 0.0438 - val_accuracy: 0.9831
Epoch 4/10
30/30 [==============================] - 13s 426ms/step - loss: 0.0341 - acc
uracy: 0.9892 - val_loss: 0.0127 - val_accuracy: 0.9968
Epoch 5/10
30/30 [==============================] - 9s 304ms/step - loss: 0.0293 - accu
racy: 0.9913 - val_loss: 0.0154 - val_accuracy: 0.9968
Epoch 6/10
30/30 [==============================] - 8s 257ms/step - loss: 0.0210 - accu
racy: 0.9945 - val_loss: 0.0166 - val_accuracy: 0.9947
Epoch 7/10
30/30 [==============================] - 11s 375ms/step - loss: 0.0156 - acc
uracy: 0.9947 - val_loss: 0.0120 - val_accuracy: 0.9958
Epoch 8/10
30/30 [==============================] - 9s 303ms/step - loss: 0.0094 - accu
racy: 0.9966 - val_loss: 0.0084 - val_accuracy: 0.9968
Epoch 9/10
30/30 [==============================] - 8s 260ms/step - loss: 0.0078 - accu
racy: 0.9984 - val_loss: 0.0221 - val_accuracy: 0.9895
Epoch 10/10
30/30 [==============================] - 8s 262ms/step - loss: 0.0038 - accu
racy: 0.9992 - val_loss: 0.0135 - val_accuracy: 0.9968
```

Out[14]:

```
<keras.callbacks.History at 0x7fb201587f10>
```

7.Save the Model

In [18]:

```
model.save("model_1")
```

```
WARNING:absl:Function `_wrapped_model` contains input name(s) InputLayer wit
h unsupported characters which will be renamed to inputlayer in the SavedMod
el.
WARNING:absl:Found untraced functions such as lstm_cell_layer_call_fn, lstm_
cell_layer_call_and_return_conditional_losses while saving (showing 2 of 2).
These functions will not be directly callable after loading.
```

8.Test the Model

In [1]:

9

```
test_sequences=tok.texts_to_sequences(X_test)
test_sequences_matrix =sequence.pad_sequences(test_sequences,maxlen=max_len)
```

In [20]:

```
accuracy = model.evaluate(test_sequences_matrix,Y_test)
print('Accuracy: {:0.3f}'.format(accuracy[1]))
```

```
27/27 [==============================] - 1s 24ms/step - loss: 0.1139 - accur
acy: 0.9833
Accuracy: 0.983
```

In [21]:

```
y_pred =model.predict(test_sequences_matrix)
print(y_pred[25:40].round(3))
```

```
27/27 [==============================] - 1s 23ms/step
[[0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.999]
 [0.   ]
 [1.   ]]
```

In [23]:

```
print(Y_test[25:40])
```

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