Assignment – 4

Assignment Date	15 October 2022
Student Name	SASIKUMAR.S
Student Roll Number	820319205032
Maximum Marks	2 Marks

Importing Model building libraries

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
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
```

Importing NLTK libraries

```
import csv
import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
STOPWORDS = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
```

Reading dataset and preprocessing

```
from google.colab import drive
drive.mount('/content/gdrive')
```

Mounted at /content/gdrive

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

OUTPUT:

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

  Unnamed: 3 Unnamed: 4
0        NaN        NaN
1        NaN        NaN
2        NaN        NaN
3        NaN        NaN
4        NaN        NaN
```

```
from google.colab import drive
drive.mount('/content/gdrive')


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
```

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

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

```
#Label Encoding Required Column
X = df.v2
Y = df.v1
```

```
le = preprocessing.LabelEncoder()
Y = le.fit_transform(Y)
Y = Y.reshape(-1,1)

# Test and train data split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.15)

# Tokenisation function
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)
```

Create Model

Add layers (LSTM ,Dense-(HiddenLayers),Ouput)

```
#LSTM model
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)

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

OUTPUT:

```
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
_________________________________________________________________
```

```
model.fit(sequences_matrix,Y_train,batch_size=128,epochs=25,validation_split=
0.2)
```

OUTPUT:

```
Epoch 1/25
30/30 [==============================] - 11s 274ms/step - loss: 0.3376 -
accuracy: 0.8659 - val_loss: 0.1359 - val_accuracy: 0.9789
Epoch 2/25
30/30 [==============================] - 9s 317ms/step - loss: 0.0902 -
accuracy: 0.9762 - val_loss: 0.0471 - val_accuracy: 0.9863
Epoch 3/25
30/30 [==============================] - 8s 256ms/step - loss: 0.0435 -
accuracy: 0.9863 - val_loss: 0.0401 - val_accuracy: 0.9863
Epoch 4/25
30/30 [==============================] - 8s 256ms/step - loss: 0.0322 -
accuracy: 0.9886 - val_loss: 0.0431 - val_accuracy: 0.9884
Epoch 5/25
30/30 [==============================] - 8s 259ms/step - loss: 0.0256 -
accuracy: 0.9923 - val_loss: 0.0563 - val_accuracy: 0.9863
Epoch 6/25
30/30 [==============================] - 10s 333ms/step - loss: 0.0176 -
accuracy: 0.9947 - val_loss: 0.1245 - val_accuracy: 0.9789
Epoch 7/25
30/30 [==============================] - 8s 257ms/step - loss: 0.0180 -
accuracy: 0.9952 - val_loss: 0.0629 - val_accuracy: 0.9863
Epoch 8/25
30/30 [==============================] - 8s 255ms/step - loss: 0.0116 -
accuracy: 0.9960 - val_loss: 0.0646 - val_accuracy: 0.9852
Epoch 9/25
30/30 [==============================] - 8s 257ms/step - loss: 0.0097 -
accuracy: 0.9968 - val_loss: 0.0945 - val_accuracy: 0.9842
Epoch 10/25
30/30 [==============================] - 8s 259ms/step - loss: 0.0078 -
```

```
accuracy: 0.9982 - val_loss: 0.0975 - val_accuracy: 0.9842
Epoch 11/25
30/30 [==============================] - 8s 253ms/step - loss: 0.0048 -
accuracy: 0.9992 - val_loss: 0.1190 - val_accuracy: 0.9863
Epoch 12/25
30/30 [==============================] - 8s 256ms/step - loss: 0.0041 -
accuracy: 0.9987 - val_loss: 0.1252 - val_accuracy: 0.9852
Epoch 13/25
30/30 [==============================] - 8s 260ms/step - loss: 0.0058 -
accuracy: 0.9984 - val_loss: 0.1310 - val_accuracy: 0.9852
Epoch 14/25
30/30 [==============================] - 8s 261ms/step - loss: 0.0025 -
accuracy: 0.9992 - val_loss: 0.1534 - val_accuracy: 0.9842
Epoch 15/25
30/30 [==============================] - 8s 257ms/step - loss: 0.0033 -
accuracy: 0.9987 - val_loss: 0.1294 - val_accuracy: 0.9842
Epoch 16/25
30/30 [==============================] - 8s 256ms/step - loss: 0.0018 -
accuracy: 0.9997 - val_loss: 0.1495 - val_accuracy: 0.9852
Epoch 17/25
30/30 [==============================] - 8s 259ms/step - loss: 0.0016 -
accuracy: 0.9997 - val_loss: 0.1561 - val_accuracy: 0.9873
Epoch 18/25
30/30 [==============================] - 8s 253ms/step - loss: 0.0014 -
accuracy: 0.9995 - val_loss: 0.1704 - val_accuracy: 0.9842
Epoch 19/25
30/30 [==============================] - 8s 256ms/step - loss: 0.0023 -
accuracy: 0.9992 - val_loss: 0.1624 - val_accuracy: 0.9863
Epoch 20/25
30/30 [==============================] - 8s 259ms/step - loss: 0.0019 -
accuracy: 0.9997 - val_loss: 0.1580 - val_accuracy: 0.9863
Epoch 21/25
30/30 [==============================] - 8s 255ms/step - loss: 0.0017 -
accuracy: 0.9997 - val_loss: 0.1654 - val_accuracy: 0.9852
Epoch 22/25
30/30 [==============================] - 8s 258ms/step - loss: 0.0022 -
accuracy: 0.9995 - val_loss: 0.1918 - val_accuracy: 0.9863
Epoch 23/25
30/30 [==============================] - 8s 257ms/step - loss: 0.0013 -
accuracy: 0.9997 - val_loss: 0.2004 - val_accuracy: 0.9852
Epoch 24/25
30/30 [==============================] - 8s 255ms/step - loss: 0.0018 -
accuracy: 0.9997 - val_loss: 0.1895 - val_accuracy: 0.9831
Epoch 25/25
30/30 [==============================] - 8s 258ms/step - loss: 0.0015 -
accuracy: 0.9997 - val_loss: 0.1820 - val_accuracy: 0.9852

<keras.callbacks.History at 0x7ff68a3bcf10>
```

```
model.save("AI_Spam_Identifier")
```

```
WARNING:absl:Function `_wrapped_model` contains input name(s) InputLayer with
unsupported characters which will be renamed to inputlayer in the SavedModel.
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.
```

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

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

```
27/27 [==============================] - 1s 21ms/step - loss: 0.1908 -
accuracy: 0.9833
Accuracy: 0.983
```

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

OUTPUT:

```
27/27 [==============================] - 1s 22ms/step
[[1.   ]
 [1.   ]
 [0.   ]
 [0.   ]
 [1.   ]
 [1.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [0.   ]
 [1.   ]
 [0.001]
 [0.   ]
 [0.   ]]
```

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

OUTPUT:

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