Assignment_4

October 31, 2022

1 Import required library

[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 Importing NLTK libraries

[2]:
```
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.
```

3 Read dataset and do pre-processing

[3]:
```
df = pd.read_csv('/content/spam.csv',delimiter=',',encoding='latin-1')
df.head()
```

[3]:
```
     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
```

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

[5]:
```
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
```

[6]:
```
df.groupby(['v1']).size()
```

[6]:
```
v1
ham     4825
spam     747
dtype: int64
```

[7]:
```
#Label Encoding Required Column
X = df.v2
Y = df.v1
le = LabelEncoder()
Y = le.fit_transform(Y)
Y = Y.reshape(-1,1)
```

[8]:
```
# Test and train data split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.15)
```

[9]:
```
# 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)
```

4 Create Model

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

[10]:
```
#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)
```

6 Compile the Model

[11]:
```
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
_________________________________________________________________
```

7 Fit the Model

[12]:
```
model.fit(sequences_matrix,Y_train,batch_size=128,epochs=30,validation_split=0.
 ↪2)
```

```
Epoch 1/30
30/30 [==============================] - 10s 31ms/step - loss: 0.3366 -
accuracy: 0.8714 - val_loss: 0.1693 - val_accuracy: 0.9631
Epoch 2/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0826 - accuracy:
0.9797 - val_loss: 0.0743 - val_accuracy: 0.9821
Epoch 3/30
```

```
30/30 [==============================] - 0s 13ms/step - loss: 0.0481 - accuracy:
0.9865 - val_loss: 0.0554 - val_accuracy: 0.9810
Epoch 4/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0375 - accuracy:
0.9908 - val_loss: 0.0566 - val_accuracy: 0.9810
Epoch 5/30
30/30 [==============================] - 0s 12ms/step - loss: 0.0272 - accuracy:
0.9923 - val_loss: 0.0646 - val_accuracy: 0.9810
Epoch 6/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0227 - accuracy:
0.9939 - val_loss: 0.0466 - val_accuracy: 0.9852
Epoch 7/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0178 - accuracy:
0.9947 - val_loss: 0.0478 - val_accuracy: 0.9852
Epoch 8/30
30/30 [==============================] - 0s 12ms/step - loss: 0.0116 - accuracy:
0.9982 - val_loss: 0.0542 - val_accuracy: 0.9863
Epoch 9/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0099 - accuracy:
0.9974 - val_loss: 0.0589 - val_accuracy: 0.9873
Epoch 10/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0062 - accuracy:
0.9987 - val_loss: 0.0679 - val_accuracy: 0.9863
Epoch 11/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0061 - accuracy:
0.9989 - val_loss: 0.0672 - val_accuracy: 0.9852
Epoch 12/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0042 - accuracy:
0.9992 - val_loss: 0.0760 - val_accuracy: 0.9863
Epoch 13/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0035 - accuracy:
0.9992 - val_loss: 0.0919 - val_accuracy: 0.9852
Epoch 14/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0044 - accuracy:
0.9987 - val_loss: 0.0943 - val_accuracy: 0.9831
Epoch 15/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0026 - accuracy:
0.9995 - val_loss: 0.0913 - val_accuracy: 0.9852
Epoch 16/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0020 - accuracy:
0.9995 - val_loss: 0.1078 - val_accuracy: 0.9852
Epoch 17/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0029 - accuracy:
0.9995 - val_loss: 0.1030 - val_accuracy: 0.9810
Epoch 18/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0034 - accuracy:
0.9995 - val_loss: 0.1003 - val_accuracy: 0.9852
Epoch 19/30
```

```
30/30 [==============================] - 0s 13ms/step - loss: 0.0019 - accuracy:
0.9997 - val_loss: 0.1058 - val_accuracy: 0.9852
Epoch 20/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0020 - accuracy:
0.9997 - val_loss: 0.1184 - val_accuracy: 0.9831
Epoch 21/30
30/30 [==============================] - 0s 12ms/step - loss: 0.0021 - accuracy:
0.9997 - val_loss: 0.1254 - val_accuracy: 0.9810
Epoch 22/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0019 - accuracy:
0.9997 - val_loss: 0.1391 - val_accuracy: 0.9810
Epoch 23/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0017 - accuracy:
0.9997 - val_loss: 0.1219 - val_accuracy: 0.9852
Epoch 24/30
30/30 [==============================] - 0s 14ms/step - loss: 0.0013 - accuracy:
0.9997 - val_loss: 0.1378 - val_accuracy: 0.9831
Epoch 25/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0014 - accuracy:
0.9997 - val_loss: 0.1379 - val_accuracy: 0.9831
Epoch 26/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0020 - accuracy:
0.9997 - val_loss: 0.1362 - val_accuracy: 0.9831
Epoch 27/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0020 - accuracy:
0.9995 - val_loss: 0.1377 - val_accuracy: 0.9821
Epoch 28/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0013 - accuracy:
0.9997 - val_loss: 0.1461 - val_accuracy: 0.9800
Epoch 29/30
30/30 [==============================] - 0s 14ms/step - loss: 0.0016 - accuracy:
0.9997 - val_loss: 0.2142 - val_accuracy: 0.9778
Epoch 30/30
30/30 [==============================] - 0s 13ms/step - loss: 0.0013 - accuracy:
0.9997 - val_loss: 0.1591 - val_accuracy: 0.9810
```

[12]:
```
<keras.callbacks.History at 0x7ff4fc42ea50>
```

8 Save The Model

[ ]:
```
model.save("Ai_Spam_Identifier")
```

9 Test The Model

[14]:
```
test_sequences = tok.texts_to_sequences(X_test)
test_sequences_matrix = sequence.pad_sequences(test_sequences,maxlen=max_len)
```

[15]:
```
accuracy = model.evaluate(test_sequences_matrix,Y_test)
print('Accuracy: {:0.3f}'.format(accuracy[1]))
```

```
27/27 [==============================] - 0s 6ms/step - loss: 0.1540 - accuracy:
0.9856
Accuracy: 0.986
```

[21]:
```
y_pred = model.predict(test_sequences_matrix)
print(y_pred[25:35].round(3))
```

```
27/27 [==============================] - 0s 5ms/step
[[0.]
 [0.]
 [0.]
 [0.]
 [0.]
 [0.]
 [0.]
 [0.]
 [0.]
 [0.]]
```

[20]:
```
print(Y_test[25:30])
```

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