

Assignment 4

Assignment Date	28 October 2022
Student Name	D Aparna
Student Roll no.	510919104003
Maximum Marks	2 Marks

1. Download the Dataset: -

Download the dataset

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

Mounted at /content/drive

2. Import required library

Import the required library

```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import LabelEncoder
    from tensorflow.keras.models import Model
    from tensorflow.keras.layers import LSTM, Activation, Dense, Dropout, Input, Embedding
    from tensorflow.keras.optimizers import RMSprop
    from tensorflow.keras.preprocessing.text import Tokenizer
    from tensorflow.keras.preprocessing import sequence
    from tensorflow.keras.utils import to_categorical
    from tensorflow.keras.callbacks import EarlyStopping
```

3. Read dataset and do pre-processing

Read the dataset

```
[ ] import csv
    with open('/content/drive/MyDrive/Colab Notebooks/spam.csv', 'r') as csvfile:
        reader = csv.reader(csvfile)

[ ] df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/spam.csv', encoding='latin-1')
    df
```

	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
...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will i_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows x 5 columns

Pre processing

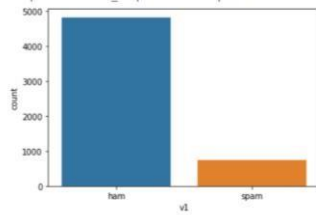
```
[ ] df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],axis=1,inplace=True)
    df.info()
```

```
<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
```

Count the spam and ham

```
[ ] sns.countplot(df.v1)
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument is x
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f475b28f590>



```
[ ] X = df.v2
Y = df.v1
le = LabelEncoder()
Y = le.fit_transform(Y)
Y = Y.reshape(-1,1)
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.20)
max_words = 10000
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-(Hidden Layers), Output)

Create Model and add Layers (LSTM,Dense-(Hidden Layers), Output)

```
[ ] def RNN():
    inputs = Input(name='inputs',shape=[max_len])
    layer = Embedding(max_words,50,input_length=max_len)(inputs)
    layer = LSTM(128)(layer)
    layer = Dense(256,name='FC1')(layer)
    layer = Activation('relu')(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1,name='out_layer')(layer)
    layer = Activation('tanh')(layer)
    model = Model(inputs=inputs,outputs=layer)
    return model
```

6. Compile the Model

Compile the Model

```
model = RNN()
model.summary()
model.compile(loss='binary_crossentropy',optimizer=RMSprop(),metrics=['accuracy','mse','mae'])
```

Model: "model_2"

Layer (type)	Output Shape	Param #
inputs (InputLayer)	[(None, 150)]	0
embedding_2 (Embedding)	(None, 150, 50)	50000
lstm_2 (LSTM)	(None, 128)	91648
FC1 (Dense)	(None, 256)	33024
activation_4 (Activation)	(None, 256)	0
dropout_2 (Dropout)	(None, 256)	0
out_layer (Dense)	(None, 1)	257
activation_5 (Activation)	(None, 1)	0

=====
Total params: 174,929
Trainable params: 174,929
Non-trainable params: 0

7. Fit the Model

Fit the Model

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

Epoch 1/10
28/28 [=====] - 16s 568ms/step - loss: 13.2345 - accuracy: 0.1321 - mse: 0.8679 - mae: 0.8679 - val_loss: 13.2149 - val_accuracy: 0.1334 - val_mse: 0.8666 - va
Epoch 2/10
28/28 [=====] - 27s 967ms/step - loss: 13.2345 - accuracy: 0.1321 - mse: 0.8679 - mae: 0.8679 - val_loss: 13.2149 - val_accuracy: 0.1334 - val_mse: 0.8666 - va
Epoch 3/10
28/28 [=====] - 15s 536ms/step - loss: 13.2345 - accuracy: 0.1321 - mse: 0.8679 - mae: 0.8679 - val_loss: 13.2149 - val_accuracy: 0.1334 - val_mse: 0.8666 - va
Epoch 4/10
28/28 [=====] - 15s 533ms/step - loss: 13.2345 - accuracy: 0.1321 - mse: 0.8679 - mae: 0.8679 - val_loss: 13.2149 - val_accuracy: 0.1334 - val_mse: 0.8666 - va
Epoch 5/10
28/28 [=====] - 15s 536ms/step - loss: 13.2345 - accuracy: 0.1321 - mse: 0.8679 - mae: 0.8679 - val_loss: 13.2149 - val_accuracy: 0.1334 - val_mse: 0.8666 - va
Epoch 6/10
28/28 [=====] - 15s 532ms/step - loss: 13.2345 - accuracy: 0.1321 - mse: 0.8679 - mae: 0.8679 - val_loss: 13.2149 - val_accuracy: 0.1334 - val_mse: 0.8666 - va
Epoch 7/10
28/28 [=====] - 15s 538ms/step - loss: 13.2345 - accuracy: 0.1321 - mse: 0.8679 - mae: 0.8679 - val_loss: 13.2149 - val_accuracy: 0.1334 - val_mse: 0.8666 - va
Epoch 8/10
28/28 [=====] - 16s 556ms/step - loss: 13.2345 - accuracy: 0.1321 - mse: 0.8679 - mae: 0.8679 - val_loss: 13.2149 - val_accuracy: 0.1334 - val_mse: 0.8666 - va
Epoch 9/10
28/28 [=====] - 16s 570ms/step - loss: 13.2345 - accuracy: 0.1321 - mse: 0.8679 - mae: 0.8679 - val_loss: 13.2149 - val_accuracy: 0.1334 - val_mse: 0.8666 - va
Epoch 10/10
28/28 [=====] - 15s 532ms/step - loss: 13.2345 - accuracy: 0.1321 - mse: 0.8679 - mae: 0.8679 - val_loss: 13.2149 - val_accuracy: 0.1334 - val_mse: 0.8666 - va
<keras.callbacks.History at 0x7f475002f650>

8. Save The Model

Save the Model

```
[ ] model.save("/assignment4_model.h5")
```

9. Test The Model

Test The Model

```
[ ] test_sequences = tok.texts_to_sequences(X_test)
    test_sequences_matrix = sequence.pad_sequences(test_sequences,maxlen=max_len)
    accr = model.evaluate(test_sequences_matrix,Y_test)
    print('Test set\n Loss: {:.3f}\n Accuracy:{:.3f}'.format(accr[0],accr[1]))
```

35/35 [=====] - 5s 129ms/step - loss: 13.0747 - accuracy: 0.1426 - mse: 0.8575 - mae: 0.8578
Test set
Loss: 13.075
Accuracy:0.143

```
[ ] from tensorflow.keras.models import load_model
    m2 = load_model("/assignment4_model.h5")
    m2.evaluate(test_sequences_matrix,Y_test)
```

35/35 [=====] - 3s 79ms/step - loss: 0.0989 - accuracy: 0.9865 - mse: 0.0216 - mae: 0.1009
[0.09894020110368729,
0.9865471124649048,
0.021568873897194862,
0.10085303336381912]