Assignment_4

October 30, 2022

Harshath.M
732119104028

```
[118]: import numpy as np
       import pandas as pd
```

```
[57]: df= pd.read_csv("/spam.csv" , encoding='latin-1')
```

```
[17]: df.head()
```

```
[17]:       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
```

```
[58]: df=df.drop(["Unnamed: 2","Unnamed: 3","Unnamed: 4"],axis=1)
```

```
[19]: 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
```

```
[59]: df["v1"].unique()
```

```
[59]: array(['ham', 'spam'], dtype=object)
```

```python
[60]: from sklearn.preprocessing import LabelEncoder
      le=LabelEncoder()
      le.fit(df["v1"].unique())
      df["v1"] = le.transform(df["v1"])
```

```python
[55]: y = df["v1"].values
```

```python
[61]: from sklearn.feature_extraction.text import CountVectorizer

      cv = CountVectorizer()
      x = cv.fit_transform(df['v2']).toarray()
      x
```

```
[61]: array([[0, 0, 0, …, 0, 0, 0],
             [0, 0, 0, …, 0, 0, 0],
             [0, 0, 0, …, 0, 0, 0],
             …,
             [0, 0, 0, …, 0, 0, 0],
             [0, 0, 0, …, 0, 0, 0],
             [0, 0, 0, …, 0, 0, 0]])
```

```python
[62]: x.shape
```

```
[62]: (5572, 8672)
```

```python
[65]: x = np.reshape(x,(5572, 8672,1))
```

```python
[66]: x.shape
```

```
[66]: (5572, 8672, 1)
```

```python
[67]: from tensorflow.keras.models import Sequential
      from tensorflow.keras.layers import Dense,LSTM
```

```python
[86]: model = Sequential()
      model.add(LSTM(10,input_shape=(8672,1)))
      model.add(Dense(2,activation='relu'))
      model.add(Dense(1,activation='sigmoid'))
```

```python
[87]: model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

```python
[88]: model.fit(x,y,epochs=5)
```

```
      Epoch 1/5
      175/175 [==============================] - 412s 2s/step - loss: 0.4721 -
      accuracy: 0.8656
      Epoch 2/5
```

```
175/175 [==============================] - 406s 2s/step - loss: 0.3948 -
accuracy: 0.8659
Epoch 3/5
175/175 [==============================] - 406s 2s/step - loss: 0.3941 -
accuracy: 0.8659
Epoch 4/5
175/175 [==============================] - 411s 2s/step - loss: 0.3947 -
accuracy: 0.8659
Epoch 5/5
175/175 [==============================] - 403s 2s/step - loss: 0.3944 -
accuracy: 0.8659
```

[88]: `<keras.callbacks.History at 0x7f7a527f0210>`

[89]: `model.save('spam_ham_lstm.h5')`

[113]: `x_test= x[:20]`

[114]: `y_pred = model.predict(x_test)`

```
1/1 [==============================] - 1s 1s/step
```

[115]: `y[:20]`

[115]: `array([0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1])`