import required library

import pandas as pd

import numpy as np

from keras import utils

import matplotlib.pyplot as plt

import seaborn as sns

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

%matplotlib inline

Read dataset

!unzip "/content/archive.zip"

Archive: /content/archive.zip

inflating: spam.csv

df = pd.read\_csv('spam.csv',delimiter=',',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 Ì\_ 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 × 5 columns

Pre processing

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

df

v1 v2

0 ham Go until jurong point, crazy.. Available only ...

1 ham Ok lar... Joking wif u oni...

2 spam Free entry in 2 a wkly comp to win FA Cup fina...

3 ham U dun say so early hor... U c already then say...

4 ham Nah I don't think he goes to usf, he lives aro...

... ... ...

5567 spam This is the 2nd time we have tried 2 contact u...

5568 ham Will Ì\_ b going to esplanade fr home?

5569 ham Pity, \* was in mood for that. So...any other s...

5570 ham The guy did some bitching but I acted like i'd...

5571 ham Rofl. Its true to its name

5572 rows × 2 columns

sns.countplot(df.v1,palette='Set3')

plt.xlabel('Label')

plt.title('Number of ham and spam messages')

/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 will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Text(0.5, 1.0, 'Number of ham and spam messages')

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=0.15)

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 = utils.pad\_sequences(sequences,maxlen=max\_len)

sequences\_matrix.shape

(4736, 150)

sequences\_matrix.ndim

2

sequences\_matrix = np.reshape(sequences\_matrix,(4736,150,1))

sequences\_matrix.ndim

3

Create model

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

from keras.layers import Embedding

model = Sequential()

Add layers(LSTM,Dense-(Hidden layers),output)

model.add(Embedding(max\_words,50,input\_length=max\_len))

model.add(LSTM(units=64,input\_shape = (sequences\_matrix.shape[1],1),return\_sequences=True))

model.add(LSTM(units=64,return\_sequences=True))

model.add(LSTM(units=64,return\_sequences=True))

model.add(LSTM(units=64))

model.add(Dense(units = 256,activation = 'relu'))

model.add(Dense(units = 1,activation = 'sigmoid'))

Compile the model

model.summary()

model.compile(loss='binary\_crossentropy',optimizer='adam',metrics=['accuracy'])

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

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

lstm (LSTM) (None, 150, 64) 29440

lstm\_1 (LSTM) (None, 150, 64) 33024

lstm\_2 (LSTM) (None, 150, 64) 33024

lstm\_3 (LSTM) (None, 64) 33024

dense (Dense) (None, 256) 16640

dense\_1 (Dense) (None, 1) 257

=================================================================

Total params: 195,409

Trainable params: 195,409

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fit the model

X = model.fit(sequences\_matrix,Y\_train,batch\_size=128,epochs=5,validation\_split=0.2)

X

Epoch 1/5

30/30 [==============================] - 43s 1s/step - loss: 0.4490 - accuracy: 0.8688 - val\_loss: 0.4257 - val\_accuracy: 0.8513

Epoch 2/5

30/30 [==============================] - 33s 1s/step - loss: 0.2615 - accuracy: 0.9092 - val\_loss: 0.1283 - val\_accuracy: 0.9610

Epoch 3/5

30/30 [==============================] - 35s 1s/step - loss: 0.0724 - accuracy: 0.9794 - val\_loss: 0.0896 - val\_accuracy: 0.9705

Epoch 4/5

30/30 [==============================] - 33s 1s/step - loss: 0.0529 - accuracy: 0.9857 - val\_loss: 0.0853 - val\_accuracy: 0.9778

Epoch 5/5

30/30 [==============================] - 33s 1s/step - loss: 0.0392 - accuracy: 0.9900 - val\_loss: 0.0836 - val\_accuracy: 0.9768

Save the model

model.save

>

Test the model

test\_sequences = tok.texts\_to\_sequences(X\_test)

test\_sequences\_matrix = utils.pad\_sequences(test\_sequences,maxlen=max\_len)

accr = model.evaluate(test\_sequences\_matrix,Y\_test)

27/27 [==============================] - 4s 88ms/step - loss: 0.0814 - accuracy: 0.9797

l = accr[0]

a =accr[1]

print('Test set\n Loss: {:0.3f}\n Accuracy: {:0.3f}'.format(l,a))

Test set

Loss: 0.081

Accuracy: 0.980mport required library

import pandas as pd

import numpy as np

from keras import utils

import matplotlib.pyplot as plt

import seaborn as sns

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

%matplotlib inline

Read dataset

!unzip "/content/archive.zip"

Archive: /content/archive.zip

inflating: spam.csv

df = pd.read\_csv('spam.csv',delimiter=',',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 Ì\_ 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 × 5 columns

Pre processing

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

df

v1 v2

0 ham Go until jurong point, crazy.. Available only ...

1 ham Ok lar... Joking wif u oni...

2 spam Free entry in 2 a wkly comp to win FA Cup fina...

3 ham U dun say so early hor... U c already then say...

4 ham Nah I don't think he goes to usf, he lives aro...

... ... ...

5567 spam This is the 2nd time we have tried 2 contact u...

5568 ham Will Ì\_ b going to esplanade fr home?

5569 ham Pity, \* was in mood for that. So...any other s...

5570 ham The guy did some bitching but I acted like i'd...

5571 ham Rofl. Its true to its name

5572 rows × 2 columns

sns.countplot(df.v1,palette='Set3')

plt.xlabel('Label')

plt.title('Number of ham and spam messages')

/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 will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Text(0.5, 1.0, 'Number of ham and spam messages')

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=0.15)

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 = utils.pad\_sequences(sequences,maxlen=max\_len)

sequences\_matrix.shape

(4736, 150)

sequences\_matrix.ndim

2

sequences\_matrix = np.reshape(sequences\_matrix,(4736,150,1))

sequences\_matrix.ndim

3

Create model

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

from keras.layers import Embedding

model = Sequential()

Add layers(LSTM,Dense-(Hidden layers),output)

model.add(Embedding(max\_words,50,input\_length=max\_len))

model.add(LSTM(units=64,input\_shape = (sequences\_matrix.shape[1],1),return\_sequences=True))

model.add(LSTM(units=64,return\_sequences=True))

model.add(LSTM(units=64,return\_sequences=True))

model.add(LSTM(units=64))

model.add(Dense(units = 256,activation = 'relu'))

model.add(Dense(units = 1,activation = 'sigmoid'))

Compile the model

model.summary()

model.compile(loss='binary\_crossentropy',optimizer='adam',metrics=['accuracy'])

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

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

lstm (LSTM) (None, 150, 64) 29440

lstm\_1 (LSTM) (None, 150, 64) 33024

lstm\_2 (LSTM) (None, 150, 64) 33024

lstm\_3 (LSTM) (None, 64) 33024

dense (Dense) (None, 256) 16640

dense\_1 (Dense) (None, 1) 257

=================================================================

Total params: 195,409

Trainable params: 195,409

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fit the model

X = model.fit(sequences\_matrix,Y\_train,batch\_size=128,epochs=5,validation\_split=0.2)

X

Epoch 1/5

30/30 [==============================] - 43s 1s/step - loss: 0.4490 - accuracy: 0.8688 - val\_loss: 0.4257 - val\_accuracy: 0.8513

Epoch 2/5

30/30 [==============================] - 33s 1s/step - loss: 0.2615 - accuracy: 0.9092 - val\_loss: 0.1283 - val\_accuracy: 0.9610

Epoch 3/5

30/30 [==============================] - 35s 1s/step - loss: 0.0724 - accuracy: 0.9794 - val\_loss: 0.0896 - val\_accuracy: 0.9705

Epoch 4/5

30/30 [==============================] - 33s 1s/step - loss: 0.0529 - accuracy: 0.9857 - val\_loss: 0.0853 - val\_accuracy: 0.9778

Epoch 5/5

30/30 [==============================] - 33s 1s/step - loss: 0.0392 - accuracy: 0.9900 - val\_loss: 0.0836 - val\_accuracy: 0.9768

Save the model

model.save

>

Test the model

test\_sequences = tok.texts\_to\_sequences(X\_test)

test\_sequences\_matrix = utils.pad\_sequences(test\_sequences,maxlen=max\_len)

accr = model.evaluate(test\_sequences\_matrix,Y\_test)

27/27 [==============================] - 4s 88ms/step - loss: 0.0814 - accuracy: 0.9797

l = accr[0]

a =accr[1]

print('Test set\n Loss: {:0.3f}\n Accuracy: {:0.3f}'.format(l,a))

Test set

Loss: 0.081

Accuracy: 0.980mport required library

import pandas as pd

import numpy as np

from keras import utils

import matplotlib.pyplot as plt

import seaborn as sns

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

%matplotlib inline

Read dataset

!unzip "/content/archive.zip"

Archive: /content/archive.zip

inflating: spam.csv

df = pd.read\_csv('spam.csv',delimiter=',',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 Ì\_ 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 × 5 columns

Pre processing

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

df

v1 v2

0 ham Go until jurong point, crazy.. Available only ...

1 ham Ok lar... Joking wif u oni...

2 spam Free entry in 2 a wkly comp to win FA Cup fina...

3 ham U dun say so early hor... U c already then say...

4 ham Nah I don't think he goes to usf, he lives aro...

... ... ...

5567 spam This is the 2nd time we have tried 2 contact u...

5568 ham Will Ì\_ b going to esplanade fr home?

5569 ham Pity, \* was in mood for that. So...any other s...

5570 ham The guy did some bitching but I acted like i'd...

5571 ham Rofl. Its true to its name

5572 rows × 2 columns

sns.countplot(df.v1,palette='Set3')

plt.xlabel('Label')

plt.title('Number of ham and spam messages')

/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 will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Text(0.5, 1.0, 'Number of ham and spam messages')

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=0.15)

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 = utils.pad\_sequences(sequences,maxlen=max\_len)

sequences\_matrix.shape

(4736, 150)

sequences\_matrix.ndim

2

sequences\_matrix = np.reshape(sequences\_matrix,(4736,150,1))

sequences\_matrix.ndim

3

Create model

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

from keras.layers import Embedding

model = Sequential()

Add layers(LSTM,Dense-(Hidden layers),output)

model.add(Embedding(max\_words,50,input\_length=max\_len))

model.add(LSTM(units=64,input\_shape = (sequences\_matrix.shape[1],1),return\_sequences=True))

model.add(LSTM(units=64,return\_sequences=True))

model.add(LSTM(units=64,return\_sequences=True))

model.add(LSTM(units=64))

model.add(Dense(units = 256,activation = 'relu'))

model.add(Dense(units = 1,activation = 'sigmoid'))

Compile the model

model.summary()

model.compile(loss='binary\_crossentropy',optimizer='adam',metrics=['accuracy'])

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

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

lstm (LSTM) (None, 150, 64) 29440

lstm\_1 (LSTM) (None, 150, 64) 33024

lstm\_2 (LSTM) (None, 150, 64) 33024

lstm\_3 (LSTM) (None, 64) 33024

dense (Dense) (None, 256) 16640

dense\_1 (Dense) (None, 1) 257

=================================================================

Total params: 195,409

Trainable params: 195,409

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fit the model

X = model.fit(sequences\_matrix,Y\_train,batch\_size=128,epochs=5,validation\_split=0.2)

X

Epoch 1/5

30/30 [==============================] - 43s 1s/step - loss: 0.4490 - accuracy: 0.8688 - val\_loss: 0.4257 - val\_accuracy: 0.8513

Epoch 2/5

30/30 [==============================] - 33s 1s/step - loss: 0.2615 - accuracy: 0.9092 - val\_loss: 0.1283 - val\_accuracy: 0.9610

Epoch 3/5

30/30 [==============================] - 35s 1s/step - loss: 0.0724 - accuracy: 0.9794 - val\_loss: 0.0896 - val\_accuracy: 0.9705

Epoch 4/5

30/30 [==============================] - 33s 1s/step - loss: 0.0529 - accuracy: 0.9857 - val\_loss: 0.0853 - val\_accuracy: 0.9778

Epoch 5/5

30/30 [==============================] - 33s 1s/step - loss: 0.0392 - accuracy: 0.9900 - val\_loss: 0.0836 - val\_accuracy: 0.9768

Save the model

model.save

>

Test the model

test\_sequences = tok.texts\_to\_sequences(X\_test)

test\_sequences\_matrix = utils.pad\_sequences(test\_sequences,maxlen=max\_len)

accr = model.evaluate(test\_sequences\_matrix,Y\_test)

27/27 [==============================] - 4s 88ms/step - loss: 0.0814 - accuracy: 0.9797

l = accr[0]

a =accr[1]

print('Test set\n Loss: {:0.3f}\n Accuracy: {:0.3f}'.format(l,a))

Test set

Loss: 0.081

Accuracy: 0.980v