

Import the necessary libraries

```
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 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 pad_sequences
from keras.utils import to_categorical
from keras.callbacks import EarlyStopping
```

Read dataset and do pre-processing

Read dataset

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

Mounted at /content/drive

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

	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

Preprocessing the Dataset

```
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
```

```
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 = pad_sequences(sequences,maxlen=max_len)
```

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

```
inputs = Input(name='inputs',shape=[max_len])
layer = Embedding(max_words,50,input_length=max_len)(inputs)
layer = LSTM(64)(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('sigmoid')(layer)
model = Model(inputs=inputs,outputs=layer)
model.summary()
```

Model: "model_1"

Layer (type)	Output Shape	Param #
=====		
inputs (InputLayer)	[(None, 150)]	0
embedding_1 (Embedding)	(None, 150, 50)	50000
lstm_1 (LSTM)	(None, 64)	29440
FC1 (Dense)	(None, 256)	16640
activation_2 (Activation)	(None, 256)	0
dropout_1 (Dropout)	(None, 256)	0
out_layer (Dense)	(None, 1)	257

activation_3 (Activation) (None, 1) 0

```
=====
Total params: 96,337
Trainable params: 96,337
Non-trainable params: 0
=====
```

Compile the Model

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

Train and Fit the Model

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

```
Epoch 1/10
30/30 [=====] - 11s 293ms/step - loss: 0.3269 - accuracy: 0
Epoch 2/10
30/30 [=====] - 8s 274ms/step - loss: 0.0788 - accuracy: 0.9
Epoch 3/10
30/30 [=====] - 8s 271ms/step - loss: 0.0408 - accuracy: 0.9
Epoch 4/10
30/30 [=====] - 8s 271ms/step - loss: 0.0319 - accuracy: 0.9
Epoch 5/10
30/30 [=====] - 10s 332ms/step - loss: 0.0223 - accuracy: 0
Epoch 6/10
30/30 [=====] - 8s 277ms/step - loss: 0.0161 - accuracy: 0.9
Epoch 7/10
30/30 [=====] - 8s 271ms/step - loss: 0.0132 - accuracy: 0.9
Epoch 8/10
30/30 [=====] - 8s 272ms/step - loss: 0.0092 - accuracy: 0.9
Epoch 9/10
30/30 [=====] - 10s 346ms/step - loss: 0.0066 - accuracy: 0
Epoch 10/10
30/30 [=====] - 8s 270ms/step - loss: 0.0113 - accuracy: 0.9
<keras.callbacks.History at 0x7fbb15657550>
```

Save The Model

```
model.save('sms_classifier.h5')
```

Preprocessing the Test Dataset

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

Testing the Model

```
accr = model.evaluate(test_sequences_matrix,Y_test)
```

27/27 [=====] - 1s 24ms/step - loss: 0.0752 - accuracy: 0.98



```
print('Test set\n Loss: {:.3f}\n Accuracy: {:.3f}'.format(accr[0],accr[1]))
```

```
Test set
Loss: 0.075
Accuracy: 0.983
```

[Colab paid products](#) - [Cancel contracts here](#)

