Open in Colab

```
import numpy as np

import json
import tensorflow as tf

from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

!wget --no-check-certificate \
    https://storage.googleapis.com/laurencemoroney-blog.appspot.com/sarcasm.json \
    -O /tmp/sarcasm.json

vocab_size = 1000
embedding_dim = 16
max_length = 120
trunc_type='post'
padding_type='post'
oov_tok = "<OOV>"
training_size = 20000


with open("/tmp/sarcasm.json", 'r') as f:
    datastore = json.load(f)


sentences = []
labels = []
urls = []
for item in datastore:
    sentences.append(item['headline'])
    labels.append(item['is_sarcastic'])

training_sentences = sentences[0:training_size]
testing_sentences = sentences[training_size:]
training_labels = labels[0:training_size]
training_labels = labels[0:training_size]
testing_labels = labels[training_size:]

tokenizer = Tokenizer(num_words=vocab_size, oov_token=oov_tok)
tokenizer.fit_on_texts(training_sentences)

word_index = tokenizer.word_index

training_sequences = tokenizer.texts_to_sequences(training_sentences)
training_padded = pad_sequences(training_sequences, maxlen=max_length, padding=padding_type,

testing_sequences = tokenizer.texts_to_sequences(testing_sentences)
testing_padded = pad_sequences(testing_sequences, maxlen=max_length, padding=padding_type, tr

model = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size, embedding_dim, input_length=max_length),
    tf.keras.layers.Conv1D(128, 5, activation='relu'),
    tf.keras.layers.GlobalMaxPooling1D(),
    tf.keras.layers.Dense(24, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
model.summary()

num_epochs = 50

training_padded = np.array(training_padded)
training_labels = np.array(training_labels)
testing_padded = np.array(testing_padded)
testing_labels = np.array(testing_labels)

history = model.fit(training_padded, training_labels, epochs=num_epochs, validation_data=(tes
     --2020-09-20 10:04:13--  https://storage.googleapis.com/laurencemoroney-blog.appspot.cor
     Resolving storage.googleapis.com (storage.googleapis.com)... 108.177.97.128, 108.177.125
     Connecting to storage.googleapis.com (storage.googleapis.com)|108.177.97.128|:443... cor
     HTTP request sent, awaiting response... 200 OK
     Length: 5643545 (5.4M) [application/json]
     Saving to: '/tmp/sarcasm.json'

     /tmp/sarcasm.json   100%[===================>]   5.38M  --.-KB/s    in 0.07s

     2020-09-20 10:04:14 (75.8 MB/s) - '/tmp/sarcasm.json' saved [5643545/5643545]

     Model: "sequential"
     _________________________________________________________________
     Layer (type)                 Output Shape              Param #
     =================================================================
     embedding (Embedding)        (None, 120, 16)           16000
     _________________________________________________________________
     conv1d (Conv1D)              (None, 116, 128)          10368
     _________________________________________________________________
     global_max_pooling1d (Global (None, 128)               0
     _________________________________________________________________
     dense (Dense)                (None, 24)                3096
     _________________________________________________________________
     dense_1 (Dense)              (None, 1)                 25
     =================================================================
     Total params: 29,489
     Trainable params: 29,489
     Non-trainable params: 0
     _________________________________________________________________
     Epoch 1/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.4688 - accuracy: 0.767
     Epoch 2/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.3568 - accuracy: 0.840
     Epoch 3/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.3155 - accuracy: 0.861
     Epoch 4/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.2782 - accuracy: 0.880
     Epoch 5/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.2428 - accuracy: 0.897
     Epoch 6/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.2083 - accuracy: 0.915
     Epoch 7/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.1764 - accuracy: 0.931
     Epoch 8/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.1470 - accuracy: 0.943
     Epoch 9/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.1222 - accuracy: 0.955
     Epoch 10/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.1028 - accuracy: 0.962
     Epoch 11/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.0846 - accuracy: 0.970
     Epoch 12/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.0730 - accuracy: 0.974
     Epoch 13/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.0642 - accuracy: 0.977
     Epoch 14/50
     625/625 [==============================] - 4s 6ms/step - loss: 0.0555 - accuracy: 0.981
```

```
    Epoch 15/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0485 - accuracy: 0.9829
    Epoch 16/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0437 - accuracy: 0.9842
    Epoch 17/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0436 - accuracy: 0.9836
    Epoch 18/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0398 - accuracy: 0.9846
    Epoch 19/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0414 - accuracy: 0.9837
    Epoch 20/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0372 - accuracy: 0.9851
    Epoch 21/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0343 - accuracy: 0.9857
    Epoch 22/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0308 - accuracy: 0.9872
    Epoch 23/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0317 - accuracy: 0.9867
    Epoch 24/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0313 - accuracy: 0.9868
    Epoch 25/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0313 - accuracy: 0.9861
    Epoch 26/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0293 - accuracy: 0.9875
    Epoch 27/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0307 - accuracy: 0.9865
    Epoch 28/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0231 - accuracy: 0.9904
    Epoch 29/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0254 - accuracy: 0.9890
    Epoch 30/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0298 - accuracy: 0.9870
    Epoch 31/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0231 - accuracy: 0.9897
    Epoch 32/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0265 - accuracy: 0.9883
    Epoch 33/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0274 - accuracy: 0.9886
    Epoch 34/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0266 - accuracy: 0.9882
    Epoch 35/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0272 - accuracy: 0.9886
    Epoch 36/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0209 - accuracy: 0.9908
    Epoch 37/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0176 - accuracy: 0.9918
    Epoch 38/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0163 - accuracy: 0.9923
    Epoch 39/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0271 - accuracy: 0.9888
    Epoch 40/50
    625/625 [==============================] - 4s 6ms/step - loss: 0.0334 - accuracy: 0.9872
    Epoch 41/50
```

```
import matplotlib.pyplot as plt


def plot_graphs(history, string):
```