```
import datetime
from packaging import version
from collections import Counter
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# TensorFlow and tf.keras
import tensorflow as tf
from tensorflow import keras
import tensorflow_datasets as tfds


%matplotlib inline
np.set_printoptions(precision=3, suppress=True)


def plot_graphs(history, metric):
  plt.plot(history.history[metric])
  plt.plot(history.history['val_'+metric], '')
  plt.xlabel("Epochs")
  plt.ylabel(metric)
  plt.legend([metric, 'val_'+metric])


#register  ag_news_subset so that tfds.load doesn't generate a checksum (mismatch)
!python -m tensorflow_datasets.scripts.download_and_prepare --register_checksums --

# https://www.tensorflow.org/datasets/splits
# The full `train` and `test` splits, interleaved together.
ri = tfds.core.ReadInstruction('train') + tfds.core.ReadInstruction('test')
dataset_all, info = tfds.load('ag_news_subset', with_info=True,  split=ri, as_super
```

```
    2021-08-09 17:08:57.834956: I tensorflow/stream_executor/platform/default/dso_
    I0809 17:09:00.013056 139638028969856 download_and_prepare.py:200] Running do
    ag_news_subset
    2021-08-09 17:09:00.020596: I tensorflow/core/platform/cloud/google_auth_prov
    2021-08-09 17:09:00.128816: I tensorflow/core/platform/cloud/google_auth_prov
    2021-08-09 17:09:00.208541: I tensorflow/core/platform/cloud/google_auth_prov
    I0809 17:09:00.282195 139638028969856 dataset_info.py:434] Load pre-computed
    2021-08-09 17:09:00.296452: I tensorflow/core/platform/cloud/google_auth_prov
    2021-08-09 17:09:00.504638: I tensorflow/core/platform/cloud/google_auth_prov
    I0809 17:09:00.721039 139638028969856 dataset_info.py:361] Load dataset info
    I0809 17:09:00.722968 139638028969856 download_and_prepare.py:138] download_a
    I0809 17:09:00.723343 139638028969856 dataset_builder.py:357] Generating data
    Downloading and preparing dataset ag_news_subset/1.0.0 (download: 11.24 MiB,
    2021-08-09 17:09:00.865821: I tensorflow/core/platform/cloud/google_auth_prov
    2021-08-09 17:09:00.969605: I tensorflow/core/platform/cloud/google_auth_prov
```

✓ 0s completed at 1:03 PM ● ×

```
Dl Size...: 0 MiB [00:08, ? MiB/s]

Extraction completed...: 0 file [00:08, ? file/s]
Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 1 MiB [00:08,  8.72s/ MiB]

Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 2 MiB [00:08,  8.72s/ MiB]

Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 3 MiB [00:08,  8.72s/ MiB]

Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 4 MiB [00:08,  8.72s/ MiB]

Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 5 MiB [00:08,  8.72s/ MiB]

Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 6 MiB [00:08,  8.72s/ MiB]

Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 7 MiB [00:08,  8.72s/ MiB]

Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 8 MiB [00:08,  8.72s/ MiB]

Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 9 MiB [00:08,  8.72s/ MiB]

Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 10 MiB [00:08,  8.72s/ MiB]

Dl Completed...:   0% 0/1 [00:08<?, ? url/s]
Dl Size...: 11 MiB [00:08,  8.72s/ MiB]
```

EDA

```
# classes dictionary
categories =dict(enumerate(info.features["label"].names))
categories
train_categories = [categories[label] for label in dataset_all.map(lambda text, lab
Counter(train_categories).most_common()
```

```
%%time
doc_sizes = []
corpus = []
for example, _ in dataset_all.as_numpy_iterator():
  enc_example = encoder(example)
  doc_sizes.append(len(enc_example))
  corpus+=list(enc_example.numpy())
```

```
    CPU times: user 11min 2s, sys: 1min 7s, total: 12min 10s
    Wall time: 10min 21s
```

```
# register  ag_news_subset so that tfds.load doesn't generate a checksum (mismatch)
!python -m tensorflow_datasets.scripts.download_and_prepare --register_checksums --

# Example Approaches to Split Data Set
# dataset, info = tfds.load('ag_news_subset', with_info=True,  split=['train[:]','t
dataset, info = tfds.load('ag_news_subset', with_info=True,  split=['train[:95%]','
# dataset, info = tfds.load('ag_news_subset', with_info=True,  split=['train[:11400
                          as_supervised=True)
train_dataset, validation_dataset, test_dataset = dataset
# train_dataset, test_dataset = dataset['train'],dataset['test']
```

```
    2021-08-09 17:22:15.569553: I tensorflow/stream_executor/platform/default/dso_
    I0809 17:22:17.631692 139639198746496 download_and_prepare.py:200] Running dow
    ag_news_subset
    I0809 17:22:17.632377 139639198746496 dataset_info.py:361] Load dataset info f
    I0809 17:22:17.633637 139639198746496 download_and_prepare.py:138] download_an
    I0809 17:22:17.633913 139639198746496 dataset_builder.py:299] Reusing dataset
    name: "ag_news_subset"
    description: "AG is a collection of more than 1 million news articles.\nNews a
    citation: "@misc{zhang2015characterlevel,\n    title={Character-level Convolu
    location {
      urls: "https://arxiv.org/abs/1509.01626"
    }
    splits {
      name: "test"
      shard_lengths: 7600
      num_bytes: 2226751
    }
```

```
test_dataset = test_dataset.batch(BATCH_SIZE).prefetch(tf.data.experimental.AUTOTUN
```

```
for example, label in train_dataset.take(2):
  print('texts: ', example.numpy()[:3])
  print()
  print('labels: ', label.numpy()[:3])
```

```
texts:  [b'The wreckage of a Detroit case about a "sleeper operational combat
 b'NOVEMBER 29, 2004 (IDG NEWS SERVICE) - Malicious hackers have compromised
 b'Russian Svetlana Kuznetsova came back twice from a break down in the secon

labels:  [0 3 1]
texts:  [b'Gossip-seekers eager to dig up dirt on David Beckham risk ending u
 b'NEW YORK -- New York boxing officials have suspended former heavyweight ch
 b'Reuters - The trial of Indonesian\\cleric Abu Bakar Bashir over charges re

labels:  [3 1 0]
```

```
VOCAB_SIZE=1000
encoder = tf.keras.layers.experimental.preprocessing.TextVectorization(
    max_tokens=VOCAB_SIZE)
encoder.adapt(train_dataset.map(lambda text, label: text))
```

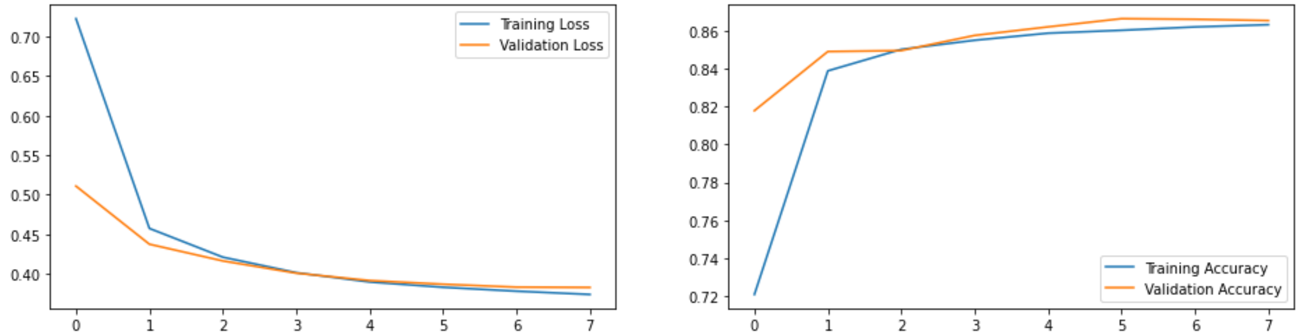
```
num_classes = 4
model = tf.keras.Sequential([
                              encoder
                              ,tf.keras.layers.Embedding(input_dim=len(encoder.get_
                              ,output_dim=64
                                # Use masking to handle the variable sequence lengt
                              ,mask_zero=True)
                              ,tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(6
    Epoch 6/200
    1782/1782 [==============================] - 30s 17ms/step - loss: 0.3827 - a
    Epoch 7/200
    1782/1782 [==============================] - 32s 18ms/step - loss: 0.3777 - a
    Epoch 8/200
    1782/1782 [==============================] - 32s 18ms/step - loss: 0.3736 - a
```

```
test_loss, test_acc = model.evaluate(test_dataset)

print('Test Loss: {}'.format(test_loss))
print('Test Accuracy: {}'.format(test_acc))
```

```
    119/119 [==============================] - 1s 10ms/step - loss: 0.4034 - accu
    Test Loss: 0.4034428000450134
    Test Accuracy: 0.8501315712928772
```

```
history_dict = history.history
history_dict.keys()

history_df=pd.DataFrame(history_dict)
history_df.tail(10).round(3)
```

```
plt.show()
```



```
preds = model.predict(test_dataset)
```

```
preds.shape
```

```
              ,loss=tf.keras.losses.SparseCategoricalCrossentropy() # if we set fro
              ,metrics=['accuracy'])
```

Simple RNN significantly increases the amount of time needed to train the model for approximately the same performance

```
history = model.fit(train_dataset
                    ,epochs = 200
                    ,validation_data=validation_dataset
                    ,callbacks=[tf.keras.callbacks.EarlyStopping(monitor='val_accur
                    )
```

1	[illegible]	[illegible]	[illegible]	[illegible]
2	0.421	0.851	0.437	0.847
3	0.399	0.858	0.404	0.859
4	0.387	0.862	0.405	0.859
5	0.376	0.864	0.393	0.862