multi_class_text_classification

April 8, 2022

1 LSTM multiple classification of BBC articles

using this dataset and this guide

disciphering what type of article from the article's text

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

importing packages

[ ]:
```
from urllib.request import urlopen

import csv
import tensorflow as tf
import numpy as np
import pandas as pd

# tensorflow
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, LSTM, Dropout, Activation,␣
↪Embedding, Bidirectional
```

reading in stopwords these are common words that prodive little context or value in our case.

[ ]:
```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
STOPWORDS = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

1.1 setting model parameters and variables

it sets the neural network dimensions as well as how to truncate and select certain words for modelling on.

[ ]:
```
vocab_size = 5000 # make the top list of words (common words)
embedding_dim = 64
max_length = 200
trunc_type = 'post'
padding_type = 'post'
oov_tok = '<OOV>' # OOV = Out of Vocabulary
training_portion = .8
```

1.2 separating csv into articles and labels

list of each

reading in data reading it from a google server and saving locally to 'data/bbc-text.csv'

[ ]:
```
articles = []
labels = []

with open("/content/drive/MyDrive/Colab Notebooks/multi_class_BBC/data/bbc-text.
↪csv", 'r') as csvfile:
    reader = csv.reader(csvfile, delimiter=',')
    next(reader)
    for row in reader:
        labels.append(row[0])
        article = row[1]
        for word in STOPWORDS:
            token = ' ' + word + ' '
            article = article.replace(token, ' ')
            article = article.replace('  ', ' ')
        articles.append(article)
```

[ ]:
```
print(len(labels))
print(len(articles))
```

```
2225
2225
```

1.3 creating train test validation split

1.4 train test split with the articles

[ ]:
```
train_size = int(len(articles) * training_portion)

train_articles = articles[0: train_size]
train_labels = labels[0: train_size]
```

```
validation_articles = articles[train_size:]
validation_labels = labels[train_size:]
```

[ ]:
```
print("train_size", train_size)
print(f"train_articles {len(train_articles)}")
print("train_labels", len(train_labels))
print("validation_articles", len(validation_articles))
print("validation_labels", len(validation_labels))
```

```
train_size 1780
train_articles 1780
train_labels 1780
validation_articles 445
validation_labels 445
```

1.5 tokenizing words

only grabbing 5000 most common words. this is to help from grabbing unique nouns that seldom occur.

building tokenizer

[ ]:
```
tokenizer = Tokenizer(num_words = vocab_size, oov_token=oov_tok)
tokenizer.fit_on_texts(train_articles)
word_index = tokenizer.word_index
```

1.6 convert to sequencing

this converts a string into a squence of numbers that represent the word. this way the mdoel can put a number to the word and be usable.

[ ]:
```
# tokenizer = Tokenizer(num_words = vocab_size, oov_token=oov_tok)
# text_sequences = tokenizer.texts_to_sequences(["the cat sat on my table"])
# text_sequences
```

1.7 fitting tokenizer and padding

so they are all the same length, adding null to end of short sequences. can make sequences shorter or longer. it removes the end of a sentence if it is too long

[ ]:
```
train_sequences = tokenizer.texts_to_sequences(train_articles)
validation_sequences = tokenizer.texts_to_sequences(validation_articles)

train_padded = pad_sequences(train_sequences, maxlen=max_length,␣
 ↪padding=padding_type, truncating=trunc_type)
validation_padded = pad_sequences(validation_sequences, maxlen=max_length,␣
 ↪padding=padding_type, truncating=trunc_type)
```

1.8 tokenize data

[ ]:
```
label_tokenizer = Tokenizer()
label_tokenizer.fit_on_texts(labels)

training_label_seq = np.array(label_tokenizer.texts_to_sequences(train_labels))
validation_label_seq = np.array(label_tokenizer.
 ↪texts_to_sequences(validation_labels))
```

1.9 initialize the model

[ ]:
```
# create a model that uses a sequential bi directional keras model
model = Sequential()

model.add(Embedding(vocab_size, embedding_dim))
model.add(Dropout(0.5))
model.add(Bidirectional(LSTM(embedding_dim)))
model.add(Dense(64, activation='softmax'))

model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 embedding_2 (Embedding)      (None, None, 64)          320000

 dropout_2 (Dropout)          (None, None, 64)          0

 bidirectional_2 (Bidirectio  (None, 128)              66048
 nal)

 dense_2 (Dense)              (None, 64)                8256

=================================================================
Total params: 394,304
Trainable params: 394,304
Non-trainable params: 0
_________________________________________________________________
```

1.10 compile the model

[ ]:
```
opt = tf.keras.optimizers.Adam(lr=0.001, decay=1e-6)
model.compile(
    loss='sparse_categorical_crossentropy',
    optimizer=opt,
    metrics=['accuracy'],
)
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105:
UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)
```

1.11 train the model

[ ]:
```
num_epochs = 10
history = model.fit(train_padded, training_label_seq, epochs=num_epochs,␣
↪validation_data=(validation_padded, validation_label_seq), verbose=2)
```

```
Epoch 1/10
56/56 - 9s - loss: 2.3694 - accuracy: 0.2287 - val_loss: 1.5971 - val_accuracy:
0.2270 - 9s/epoch - 169ms/step
Epoch 2/10
56/56 - 5s - loss: 1.5894 - accuracy: 0.2657 - val_loss: 1.5566 - val_accuracy:
0.2899 - 5s/epoch - 86ms/step
Epoch 3/10
56/56 - 5s - loss: 1.5171 - accuracy: 0.3331 - val_loss: 1.4713 - val_accuracy:
0.3618 - 5s/epoch - 87ms/step
Epoch 4/10
56/56 - 5s - loss: 1.2759 - accuracy: 0.5708 - val_loss: 1.1566 - val_accuracy:
0.7213 - 5s/epoch - 85ms/step
Epoch 5/10
56/56 - 5s - loss: 0.8431 - accuracy: 0.7989 - val_loss: 0.6924 - val_accuracy:
0.8404 - 5s/epoch - 86ms/step
Epoch 6/10
56/56 - 5s - loss: 0.4635 - accuracy: 0.9079 - val_loss: 0.4757 - val_accuracy:
0.8854 - 5s/epoch - 86ms/step
Epoch 7/10
56/56 - 5s - loss: 0.2276 - accuracy: 0.9758 - val_loss: 0.3123 - val_accuracy:
0.9281 - 5s/epoch - 86ms/step
Epoch 8/10
56/56 - 5s - loss: 0.1128 - accuracy: 0.9921 - val_loss: 0.2776 - val_accuracy:
0.9236 - 5s/epoch - 86ms/step
Epoch 9/10
56/56 - 5s - loss: 0.0887 - accuracy: 0.9893 - val_loss: 0.2619 - val_accuracy:
0.9056 - 5s/epoch - 86ms/step
Epoch 10/10
56/56 - 5s - loss: 0.0373 - accuracy: 1.0000 - val_loss: 0.1895 - val_accuracy:
0.9416 - 5s/epoch - 86ms/step
```

1.12 prediction

[ ]:
```

```

```
txt = ["blair prepares to name poll date tony blair is likely to name 5 may as␣
 ↪election day when parliament returns from its easter break  the bbc s␣
 ↪political editor has learned.  andrew marr says mr blair will ask the queen␣
 ↪on 4 or 5 april to dissolve parliament at the end of that week. mr blair has␣
 ↪so far resisted calls for him to name the day but all parties have stepped␣
 ↪up campaigning recently. downing street would not be drawn on the claim ␣
 ↪saying election timing was a matter for the prime minister.  a number 10␣
 ↪spokeswoman would only say:  he will announce an election when he wants to␣
 ↪announce an election.  the move will signal a frantic week at westminster as␣
 ↪the government is likely to try to get key legislation through parliament.␣
 ↪the government needs its finance bill  covering the budget plans  to be␣
 ↪passed before the commons closes for business at the end of the session on 7␣
 ↪april.  but it will also seek to push through its serious and organised␣
 ↪crime bill and id cards bill. mr marr said on wednesday s today programme: ␣
 ↪there s almost nobody at a senior level inside the government or in␣
 ↪parliament itself who doesn t expect the election to be called on 4 or 5␣
 ↪april.  as soon as the commons is back after the short easter recess  tony␣
 ↪blair whips up to the palace  asks the queen to dissolve parliament ... and␣
 ↪we re going.  the labour government officially has until june 2006 to hold␣
 ↪general election  but in recent years governments have favoured four-year␣
 ↪terms."]

seq = tokenizer.texts_to_sequences(txt)
padded = pad_sequences(seq, maxlen=max_length)
pred = model.predict(padded)
labels = ['sport', 'bussiness', 'politics', 'tech', 'entertainment']

print(pred)
print(np.argmax(pred))
print(labels[np.argmax(pred)-1])
```

```
[[5.4528948e-07 6.5186606e-03 1.4443640e-03 9.8001456e-01 1.1763926e-02
  2.1246016e-04 3.5696254e-07 3.1074489e-07 2.0138960e-07 1.3819903e-06
  3.8346673e-07 6.4541047e-07 6.5637869e-07 7.0535361e-07 7.3257024e-07
  3.7014624e-07 7.3992658e-07 6.1169266e-07 8.7923189e-07 9.1063566e-07
  9.4149959e-07 4.3945184e-07 1.9908578e-06 2.6611062e-06 4.3266598e-07
  5.8000234e-07 5.7347364e-07 6.3086316e-07 1.2377124e-06 6.8401158e-07
  9.9870692e-07 7.7494400e-07 8.3759943e-07 2.6616098e-07 4.2281624e-07
  2.5781383e-06 4.1506357e-07 1.3966390e-06 1.3550534e-06 3.8476381e-07
  6.5075727e-07 4.2661222e-07 2.3443522e-06 6.7789739e-07 3.2119479e-07
  6.3699940e-07 4.3632062e-07 3.6425479e-07 4.0196355e-07 8.0936167e-07
  3.4867159e-07 9.1323778e-07 4.6004203e-07 6.5606014e-07 6.8236506e-07
  1.9187301e-07 3.0250646e-07 2.4071926e-06 7.3494874e-07 5.4471110e-07
  7.3294416e-07 7.2370761e-07 5.8908881e-07 5.7747775e-07]]
3
politics
```

```
[ ]: txt = ["call to save manufacturing jobs the trades union congress (tuc) is␣
      ↪calling on the government to stem job losses in manufacturing firms by␣
      ↪reviewing the help it gives companies.  the tuc said in its submission␣
      ↪before the budget that action is needed because of 105 000 jobs lost from␣
      ↪the sector over the last year. it calls for better pensions  child care␣
      ↪provision and decent wages. the 36-page submission also urges the government␣
      ↪to examine support other european countries provide to industry. tuc general␣
      ↪secretary brendan barber called for  a commitment to policies that will make␣
      ↪a real difference to the lives of working people.    greater investment in␣
      ↪childcare strategies and the people delivering that childcare will increases␣
      ↪the options available to working parents   he said.  a commitment to our␣
      ↪public services and manufacturing sector ensures that we can continue to␣
      ↪compete on a global level and deliver the frontline services that this␣
      ↪country needs.  he also called for  practical measures  to help pensioners ␣
      ↪especially women who he said  are most likely to retire in poverty . the␣
      ↪submission also calls for decent wages and training for people working in␣
      ↪the manufacturing sector."]

     seq = tokenizer.texts_to_sequences(txt)
     padded = pad_sequences(seq, maxlen=max_length)
     pred = model.predict(padded)
     labels = ['sport', 'bussiness', 'politics', 'tech', 'entertainment']

     print(pred)
     print(np.argmax(pred))
     print(labels[np.argmax(pred)-1])
```

```
[[4.64055120e-06 1.55212230e-03 9.33732808e-01 2.80778445e-02
  3.55434828e-02 5.81650762e-04 4.75775369e-06 6.87960619e-06
  2.80046493e-06 8.46164130e-06 5.07016966e-06 9.05422985e-06
  5.83948167e-06 7.34853802e-06 8.57344367e-06 5.13397754e-06
  1.08496679e-05 2.87510898e-06 7.50055369e-06 1.11082845e-05
  1.04783094e-05 1.20740388e-05 1.22860683e-05 1.72111158e-05
  5.44817794e-06 5.48764228e-06 9.63362891e-06 9.56845543e-06
  4.37447079e-06 5.35981553e-06 4.16320017e-06 6.55491976e-06
  1.37061270e-05 3.50176970e-06 6.60892647e-06 1.85855679e-05
  9.53131439e-06 7.06927085e-06 1.97946738e-05 1.07271326e-05
  1.06321995e-05 5.14648264e-06 1.94502227e-05 3.98282236e-06
  3.89698062e-06 6.76175068e-06 4.07554990e-06 6.34912703e-06
  6.44042666e-06 3.16389487e-05 2.10076837e-06 5.19309879e-06
  6.97474115e-06 3.75193440e-06 1.12264779e-05 3.92491665e-06
  8.46710554e-06 2.35666648e-05 4.93998732e-06 8.61276294e-06
  1.22561105e-05 6.55060785e-06 1.07944015e-05 1.24697544e-05]]
2
bussiness
```