Keras exercise

In this exercise you will be creating a Keras model by loading a data set, preprocessing input data, building a Sequential Keras model and compiling the model with a training configuration. Afterwards, you train your model on the training data and evaluate it on the test set. To finish this exercise, you will past the accuracy of your model to the Coursera grader.

This notebook is tested in IBM Watson Studio under python 3.6

Data

For this exercise we will use the Reuters newswire dataset. This dataset consists of 11,228 newswires from the Reuters news agency. Each wire is encoded as a sequence of word indexes, just as in the IMDB data we encountered in lecture 5 of this series. Moreover, each wire is categorised into one of 46 topics, which will serve as our label. This dataset is available through the Keras API.

Goal

We want to create a Multi-layer perceptron (MLP) using Keras which we can train to classify news items into the specified 46 topics.

Instructions

We start by installing and importing everything we need for this exercise:

In [1]:
```
!pip install tensorflow==2.2.0rc0
```

```
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes ins
tead
  from cryptography.utils import int_from_bytes
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/secretstorage/util.py:25: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
Collecting tensorflow==2.2.0rc0
  Downloading tensorflow-2.2.0rc0-cp37-cp37m-manylinux2010_x86_64.whl (515.9 MB)
     |████████████████████████████████| 515.9 MB 40 kB/s s eta 0:00:01
Requirement already satisfied: opt-einsum>=2.3.2 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (3.1.0)
Requirement already satisfied: wrapt>=1.11.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (1.12.1)
Requirement already satisfied: numpy<2.0,>=1.16.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (1.18.5)
Requirement already satisfied: keras-preprocessing>=1.1.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (1.1.0)
Requirement already satisfied: tensorboard<2.2.0,>=2.1.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (2.1.0)
Requirement already satisfied: grpcio>=1.8.6 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (1.27.2)
Requirement already satisfied: termcolor>=1.1.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (1.1.0)
Collecting scipy==1.4.1; python_version >= "3"
  Downloading scipy-1.4.1-cp37-cp37m-manylinux1_x86_64.whl (26.1 MB)
     |████████████████████████████████| 26.1 MB 51.3 MB/s eta 0:00:01
Requirement already satisfied: absl-py>=0.7.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (0.9.0)
Requirement already satisfied: h5py<2.11.0,>=2.10.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (2.10.0)
Collecting gast==0.3.3
  Downloading gast-0.3.3-py2.py3-none-any.whl (9.7 kB)
Requirement already satisfied: six>=1.12.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (1.15.0)
Requirement already satisfied: tensorflow-estimator<2.2.0,>=2.1.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (2.1.0)
Requirement already satisfied: protobuf>=3.8.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (3.12.3)
Requirement already satisfied: google-pasta>=0.1.8 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (0.2.0)
Requirement already satisfied: wheel>=0.26; python_version >= "3" in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorflow==2.2.0rc0) (0.34.2)
Collecting astunparse==1.6.3
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: requests<3,>=2.21.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (2.2
4.0)
Requirement already satisfied: markdown>=2.6.8 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (3.1.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorboard<2.2.0,>=2.1.0->tensorflow==2.
2.0rc0) (0.4.1)
Requirement already satisfied: google-auth<2,>=1.6.3 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (1.
22.0)
Requirement already satisfied: setuptools>=41.0.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (47.3.
1.post20200622)
Requirement already satisfied: werkzeug>=0.11.15 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (1.0.1)
Requirement already satisfied: chardet<4,>=3.0.2 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from requests<3,>=2.21.0->tensorboard<2.2.0,>=2.1.0->tensorfl
ow==2.2.0rc0) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from requests<3,>=2.21.0->tensorboard<
2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (1.25.9)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from requests<3,>=2.21.0->tensorboard<2.2.0,>=2.1.0->tensorf
low==2.2.0rc0) (2020.12.5)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from requests<3,>=2.21.0->tensorboard<2.2.0,>=2.1.0->tensorflow==
2.2.0rc0) (2.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2
2.0,>=2.1.0->tensorflow==2.2.0rc0) (1.3.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from google-auth<2,>=1.6.3->tensorboard<2.2.0,>=2.1.0->te
nsorflow==2.2.0rc0) (0.2.8)
Requirement already satisfied: aiohttp<4.0.0dev,>=3.6.2; python_version >= "3.6" in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from google-auth<2,>=1.6.3->t
ensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (3.6.2)
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3.5" in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from google-auth<2,>=1.6.3->tensorboard<
2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (4.6)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from google-auth<2,>=1.6.3->tensorboard<2.2.0,>=2.1.0->t
ensorflow==2.2.0rc0) (4.1.1)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1
->tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (3.1.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from pyasn1-modules>=0.2.1->google-auth<2,>=1.6.3->tensorb
oard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (0.4.8)
Requirement already satisfied: yarl<2.0,>=1.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from aiohttp<4.0.0dev,>=3.6.2; python_version >= "3.6"->google-a
uth<2,>=1.6.3->tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (1.5.1)
Requirement already satisfied: async-timeout<4.0,>=3.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from aiohttp<4.0.0dev,>=3.6.2; python_version >= "3.6"-
>google-auth<2,>=1.6.3->tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (3.0.1)
Requirement already satisfied: multidict<5.0,>=4.5 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from aiohttp<4.0.0dev,>=3.6.2; python_version >= "3.6"->goo
gle-auth<2,>=1.6.3->tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (4.7.6)
Requirement already satisfied: attrs>=17.3.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from aiohttp<4.0.0dev,>=3.6.2; python_version >= "3.6"->google-au
th<2,>=1.6.3->tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (19.3.0)
Requirement already satisfied: typing-extensions>=3.7.4; python_version < "3.8" in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from yarl<2.0,>=1.0->aiohttp<
4.0.0dev,>=3.6.2; python_version >= "3.6"->google-auth<2,>=1.6.3->tensorboard<2.2.0,>=2.1.0->tensorflow==2.2.0rc0) (3.7.4.2)
Installing collected packages: scipy, gast, astunparse, tensorflow
  Attempting uninstall: scipy
    Found existing installation: scipy 1.5.0
    Uninstalling scipy-1.5.0:
      Successfully uninstalled scipy-1.5.0
  Attempting uninstall: gast
    Found existing installation: gast 0.2.2
    Uninstalling gast-0.2.2:
      Successfully uninstalled gast-0.2.2
  Attempting uninstall: tensorflow
    Found existing installation: tensorflow 2.1.0
```

```
    Uninstalling tensorflow-2.1.0:
      Successfully uninstalled tensorflow-2.1.0
Successfully installed astunparse-1.6.3 gast-0.3.3 scipy-1.4.1 tensorflow-2.2.0rc0
```

In [2]:
```
import tensorflow as tf
if not tf.__version__ == '2.2.0-rc0':
    print(tf.__version__)
    raise ValueError('please upgrade to TensorFlow 2.2.0-rc0, or restart your Kernel (Kernel->Restart & Clear Output)')
```

IMPORTANT! => Please restart the kernel by clicking on "Kernel"->"Restart and Clear Outout" and wait until all output disapears. Then your changes are beeing picked up

As you can see, we use Keras' Sequential model with only two types of layers: Dense and Dropout. We also specify a random seed to make our results reproducible. Next, we load the Reuters data set:

In [3]:
```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.utils import to_categorical
seed = 1337
np.random.seed(seed)
from tensorflow.keras.datasets import reuters

max_words = 1000
(x_train, y_train), (x_test, y_test) = reuters.load_data(num_words=max_words,
                                                         test_split=0.2,
                                                         seed=seed)
num_classes = np.max(y_train) + 1  # 46 topics
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/reuters.npz
2113536/2110848 [==============================] - 0s 0us/step
```

Note that we cap the maximum number of words in a news item to 1000 by specifying the *num_words* key word. Also, 20% of the data will be test data and we ensure reproducibility by setting our random seed.

Our training features are still simply sequences of indexes and we need to further preprocess them, so that we can plug them into a *Dense* layer. For this we use a *Tokenizer* from Keras' text preprocessing module. This tokenizer will take an index sequence and map it to a vector of length *max_words=1000*. Each of the 1000 vector positions corresponds to one of the words in our newswire corpus. The output of the tokenizer has a 1 at the i-th position of the vector, if the word corresponding to i is in the description of the newswire, and 0 otherwise. Even if this word appears multiple times, we still just put a 1 into our vector, i.e. our tokenizer is binary. We use this tokenizer to transform both train and test features:

In [4]:
```
from tensorflow.keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer(num_words=max_words)
x_train = tokenizer.sequences_to_matrix(x_train, mode='binary')
x_test = tokenizer.sequences_to_matrix(x_test, mode='binary')
```

1. Exercise part: label encoding

Use to_categorical, as we did in the lectures, to transform both *y_train* and *y_test* into one-hot encoded vectors of length *num_classes*:

In [5]:
```
y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)
```

2. Exercise part: model definition

Next, initialise a Keras *Sequential* model and add three layers to it:

```
Layer: Add a *Dense* layer with in input_shape=(max_words,), 512 output units and "relu" activation.
Layer: Add a *Dropout* layer with dropout rate of 50%.
Layer: Add a *Dense* layer with num_classes output units and "softmax" activation.
```

In [6]:
```
model = Sequential()  # Instantiate sequential model
model.add(Dense(512, activation='relu', input_shape=(max_words,))) # Add first layer. Make sure to specify input shape
model.add(Dropout(0.5)) # Add second layer
model.add(Dense(num_classes, activation='softmax')) # Add third layer
```

3. Exercise part: model compilation

As the next step, we need to compile our Keras model with a training configuration. Compile your model with "categorical_crossentropy" as loss function, "adam" as optimizer and specify "accuracy" as evaluation metric.
NOTE: In case you get an error regarding h5py, just restart the kernel and start from scratch

In [7]:
```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

4. Exercise part: model training and evaluation

Next, define the batch_size for training as 32 and train the model for 5 epochs on *x_train* and *y_train* by using the *fit* method of your model. Then calculate the score for your trained model by running *evaluate* on *x_test* and *y_test* with the same batch size as used in *fit*.

In [10]:
```
batch_size = 32
model.fit(x_train, y_train, batch_size=batch_size, epochs=5, valication_data=(x_test, y_test))
score = model.evaluate(x_test, y_test, verbose=0)
```

```
Epoch 1/5
281/281 [==============================] - 3s 12ms/step - loss: 0.7728 - accuracy: 0.8152
Epoch 2/5
281/281 [==============================] - 3s 10ms/step - loss: 0.5530 - accuracy: 0.8668
Epoch 3/5
281/281 [==============================] - 3s 10ms/step - loss: 0.4185 - accuracy: 0.8986
Epoch 4/5
281/281 [==============================] - 3s 10ms/step - loss: 0.3409 - accuracy: 0.9139
Epoch 5/5
281/281 [==============================] - 3s 10ms/step - loss: 0.2925 - accuracy: 0.9226
```

If you have done everything as specified, in particular set the random seed as we did above, your test accuracy should be around 80%

In [11]:
```
score[1]
```

Out[11]: 0.7951914668083191

Congratulations, now it's time to submit your result to the Coursera grader by executing the following cells (Programming Assingment, Week2).

We have to install a little library in order to submit to coursera

In [12]:
```
!rm -f rklib.py
!wget https://raw.githubusercontent.com/IBM/coursera/master/rklib.py
```

```
--2021-04-30 01:25:01--  https://raw.githubusercontent.com/IBM/coursera/master/rklib.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.108.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
```

```
Length: 2540 (2.5K) [text/plain]
Saving to: ‘rklib.py’

rklib.py            100%[===================>]   2.48K  --.-KB/s    in 0s

2021-04-30 01:25:01 (21.8 MB/s) - ‘rklib.py’ saved [2540/2540]
```

Please provide your email address and obtain a submission token (secret) on the grader's submission page in coursera, then execute the cell

In [13]:
```
from rklib import submit
import json

key = "XbAMqtjdEeepUgo7OOVwng"
part = "HCvcp"
email = "tjamesbu@gmail.com"
token = "eqlZAJelfPrwG2mt" #you can obtain it from the grader page on Coursera (have a look here if you need more information on how to obtain the token https://youtu.be/GcDo0Rwe0

submit(email, token, 'XbAMqtjdEeepUgo7OOVwng', part, [part], json.dumps(score[1]*100))
```

```
Submission successful, please check on the coursera grader page for the status
------------------------
{"elements":[{"itemId":"ozVf2","id":"tE4j0qhMEeecqgpT6QjMdA~ozVf2~4q0biqlSEeua6ApSOL_w2Q","courseId":"tE4j0qhMEeecqgpT6QjMdA"}],"paging":{},"linked":{}}
------------------------
```

In [ ]:
```
```