

Phase 2 Model Training Documentation

ST1506: DSDA FINAL YEAR PROJECT

COMPANY PROFILING TEXT MINING

August 2021

Synopsis

The purpose of this documentation is to demonstrate the steps a user should take to successfully run the ipython file for model training.

This documentation is for data scientists who want to see what the process of training the model is, and possibly to use their own data to train the model.

Table of Contents

Running The Notebook - GPU	4
Running The Notebook – TPU	6

Running The Notebook - GPU

1. Upload “NLP_Bert.ipynb” to Google Collaboratory (i.e. Google Colab).
2. Under the Runtime tab, change runtime type to “GPU” and click “SAVE”. This will allow for faster running times when running the notebook so that time can be saved.

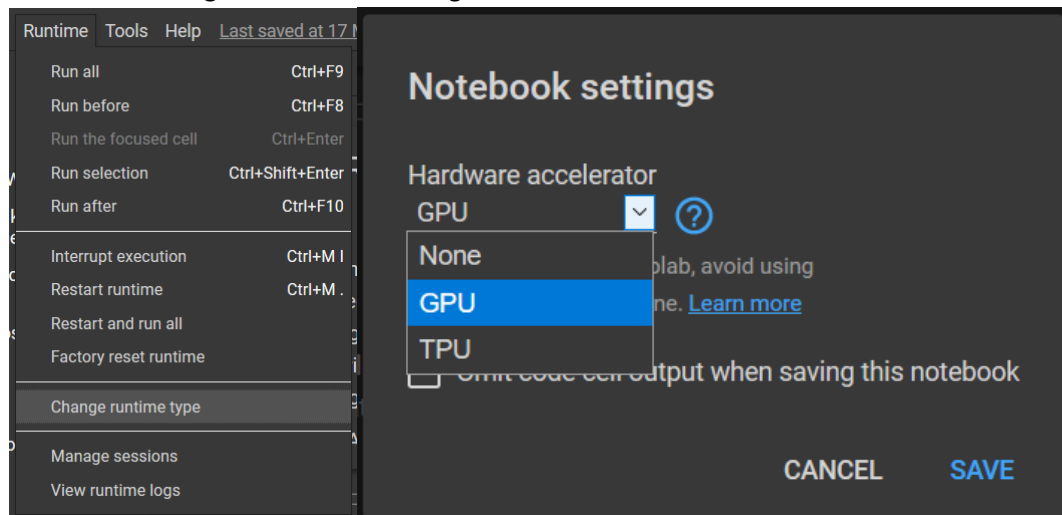


Figure 1: Changing Runtime to GPU

3. Connect to Colab runtime by clicking on the connect button found on the top right bar if not already converted. You should see the following if the connection is successful.

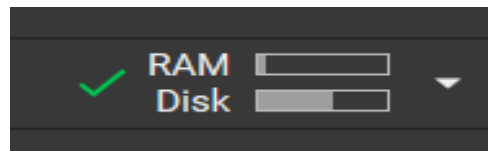


Figure 2: Ensuring runtime is connected

4. On the left tab, click on the icon of a folder and upload “clean_dataset.xlsx”, “sector_master_definition.xlsx” and “val_dataset.xlsx”. Once uploaded you should see that the files are in the temporary working directory in Google Colab.

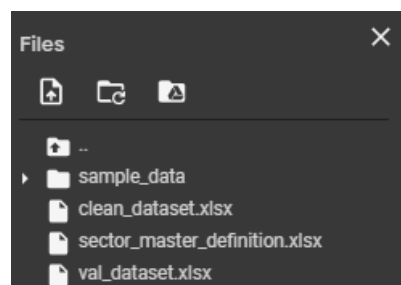


Figure 3: Uploaded excel data files

5. Now, the notebook is ready to run. Run the notebook cells one at a time starting from the start of Section 1, labelled as “Data Importing” until the end of Section 4, labelled as “BERT Model”.

1.Data Importing
1.1. Check if TPU or GPU is utilized.
1.2. Load the libraries
1.3. Load the modules
1.4. Load the dataset
2.Exploratory Data Analysis
2.1. Get overview of dataset
2.2. Drop unnecessary columns
2.3. Filter rows with valid data
2.4. Get graphical overview of dataset
2.5. See examples of company description
3.Data Preprocessing
3.1. Removing \n and \t
3.2. Calculating the word length distribution
3.3. Populating Nan cells
3.4. Assigning tags
3.5. Text Tokenization, data preprocessing using BertTokenizerFast
3.6. Preprocessing data for test dataset.
3.7. Preparing data for BERT
4.BERT model
4.1. Training Models
4.2 Evaluating Models

Figure 4: List of Contents to run

While running cell 1.1., labelled “Check if TPU or GPU is utilized”, please ensure the cell output is as of below. Otherwise, check the Hardware accelerator used in the current session.

```
2.1. Check if TPU or GPU is utilized.

We need to check the if we are currently using TPU or GPU.

1 import tensorflow as tf
2
3 # tpu
4 if tf.test.gpu_device_name() == '':
5     print('Using TPU!')
6     resolver = tf.distribute.cluster_resolver.TPUClusterResolver()
7     tf.config.experimental_connect_to_cluster(resolver)
8     tf.tpu.experimental.initialize_tpu_system(resolver)
9     print("All devices: ", tf.config.list_logical_devices('TPU'))
10    tpu_usage = True # set tpu_usage bool to true for later usage
11    strategy = tf.distribute.TPUStrategy(resolver)
12 # gpu
13 else:
14     print('Using GPU!')
15     print("Num GPUs Available: ", len(tf.config.experimental.list_physical_devices('GPU')))
16    tpu_usage = False # set tpu_usage bool to true for later usage

Using GPU!
Num GPUs Available: 1
```

Figure 5: Sample Cell Output for Section 1.1.

6. To save the model, run Section 5: Saving Model of the notebook.

Running The Notebook – TPU

1. Similar to that of Section 1 of this documentation, upload “NLP_Bert.ipynb” to Google Collaboratory (i.e. Google Colab).
2. Under the Runtime tab, change runtime type to “TPU” and click “SAVE”. This will allow for even faster running times when running the notebook so that time can be saved.



Note: While using TPU may deliver the training result faster than GPU, it is impossible to export the trained model using TPU.

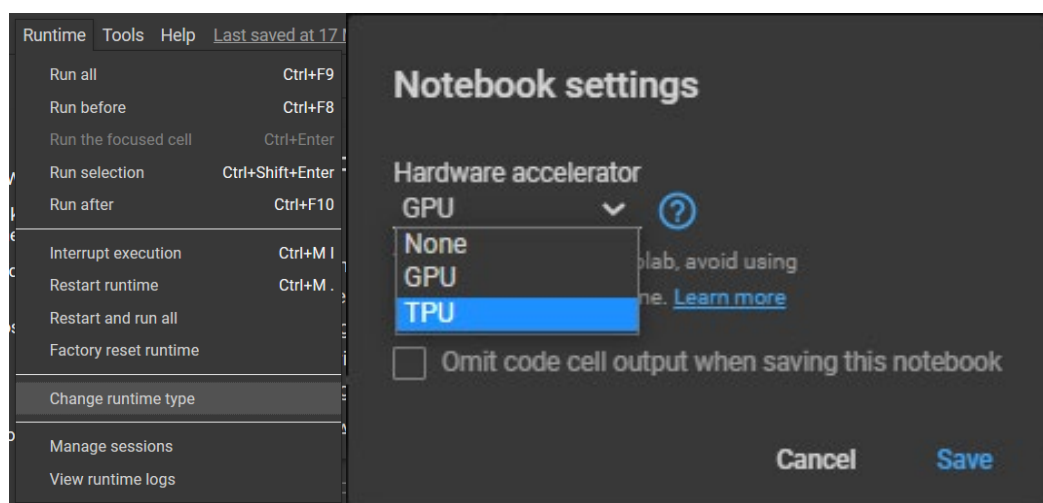


Figure 6: Changing Runtime to TPU

3. Repeat step 3 to 5 in Section 1: Running The Notebook – GPU.
4. While running cell 1.1., labelled “Check if TPU or GPU is utilized”, please ensure the cell output is as of below. Otherwise, check the Hardware accelerator used in the current session.

```
2.1. Check if TPU or GPU is utilized.
# We need to check if we are currently using TPU or GPU.

1 import tensorflow as tf
2
3 # GPU
4 if tf.test.gpu_device_name() != "":
5     print("Using GPU")
6     resolver = tf.distribute.cluster_resolver.TFClusterResolver()
7     tf.config.experimental_connect_to_cluster(resolver)
8     tf.distribute.experimental_init_system(resolver)
9     print('All devices: ', tf.config.list_logical_devices('GPU'))
10    tf.config.set_logical_device_configuration(tf.config.list_logical_devices('GPU'),
11        [tf.config.LogicalDeviceConfiguration('/dev/xnu000000000')])
12    strategy = tf.distribute.TFStrategy(resolver)
13
14 # CPU
15 print('Using CPU')
16 print('Num GPUs Available: ', len(tf.config.experimental_list_physical_devices('GPU'))
17    'log_device = false & set log_device true to test for later usage')

Using TPU
INFO:tensorflow:Initializing the TPU system: grpc://TPU:21.12.8476
INFO:tensorflow:Initializing the TPU system: grpc://TPU:21.12.8476
INFO:tensorflow:Clearing out eager caches
INFO:tensorflow:Finished initializing the system.
All devices: [LogicalDevice('/job:worker/replica:0/task:0/device:TPU:0', device_type='TPU'), LogicalDevice('/job:worker/replica:0/task:0/device:TPU:1', device_type='TPU'), LogicalDevice('/job:worker/replica:0/task:0/device:TPU:2', device_type='TPU'), LogicalDevice('/job:worker/replica:0/task:0/device:TPU:3', device_type='TPU'), LogicalDevice('/job:worker/replica:0/task:0/device:TPU:4', device_type='TPU'), LogicalDevice('/job:worker/replica:0/task:0/device:TPU:5', device_type='TPU'), LogicalDevice('/job:worker/replica:0/task:0/device:TPU:6', device_type='TPU'), LogicalDevice('/job:worker/replica:0/task:0/device:TPU:7', device_type='TPU'), LogicalDevice('/job:worker/replica:0/task:0/device:TPU:8', device_type='TPU'), LogicalDevice('/job:worker/replica:0/task:0/device:TPU:9', device_type='TPU')]
INFO:tensorflow:*** Num TPU Cores: 8
INFO:tensorflow:*** Num TPU workers: 1
INFO:tensorflow:*** Num TPU cores per worker: 8
INFO:tensorflow:*** Num TPU cores per worker: 8
INFO:tensorflow:*** Available Device: [device:TPU:0:replica:0/task:0/device:TPU:0]
INFO:tensorflow:*** Available Device: [device:TPU:1:replica:0/task:0/device:TPU:1]
INFO:tensorflow:*** Available Device: [device:TPU:2:replica:0/task:0/device:TPU:2]
INFO:tensorflow:*** Available Device: [device:TPU:3:replica:0/task:0/device:TPU:3]
INFO:tensorflow:*** Available Device: [device:TPU:4:replica:0/task:0/device:TPU:4]
INFO:tensorflow:*** Available Device: [device:TPU:5:replica:0/task:0/device:TPU:5]
INFO:tensorflow:*** Available Device: [device:TPU:6:replica:0/task:0/device:TPU:6]
INFO:tensorflow:*** Available Device: [device:TPU:7:replica:0/task:0/device:TPU:7]
INFO:tensorflow:*** Available Device: [device:TPU:8:replica:0/task:0/device:TPU:8]
INFO:tensorflow:*** Available Device: [device:TPU:9:replica:0/task:0/device:TPU:9]
```

Figure 7: Sample Cell Output for Section 1.1.