Sequence Pruning Operational Manual

Someone who does not want to reveal his name

June 2020

1 Requirements

1.1 Hardware Environment & Operating System

All of our experiments are tested with a personal computer with one GeForce GTX 1080 Ti Graphics Card and 64G RAM. Be aware that you must have a graphics card with more than 11G video memory to carry on following experiments, otherwise you have to decrease the batch size somehow.

Our code is tested with Windows 10. It should be fine when running with Linux.

1.2 Python Environment & required Libraries

We use python 3.6 in our experiments, and we noticed that python 3.7 may encounter some error when installing required libraries. Required Libraries are listed in table 1:

Transformers is a library made by hugging face team, it can be download through pip or `https://github.com/huggingface/transformers`

Tensorflow and **Pandas** are used in LSTM related experiments, if reader cares only about experiments of BERT, then it is not compulsory to have them installed.

2 Preparation work

2.1 Download & Tokenize Your Data

We have uploaded our data to `http://www.mediafire.com/file/w8ug8zs92ddjf5d/data.rar/file`, reader can directly download it and unzip all data to project directory. After the uncompressing, the project directory should look like figure 1.

Use PrepareAgnews4Lstm.py, PrepareGlue4Lstm.py, PrepareGlue4Bert.py to tokenize all data for training and evaluation. Those scripts need no parameter, so you can directly run them and wait for them to finish their processing (it will take minutes).

After the tokenization, make sure you have a cache directory with processed dataset. it should look like figure 2.

Library name	version
Pytorch	1.3
Tensorflow	2.0
Scikit-learn	0.21
Tqdm	4.32.2
Transformers	2.3
Pandas	0.24
Numpy	1.16

Table 1: Required Libraries

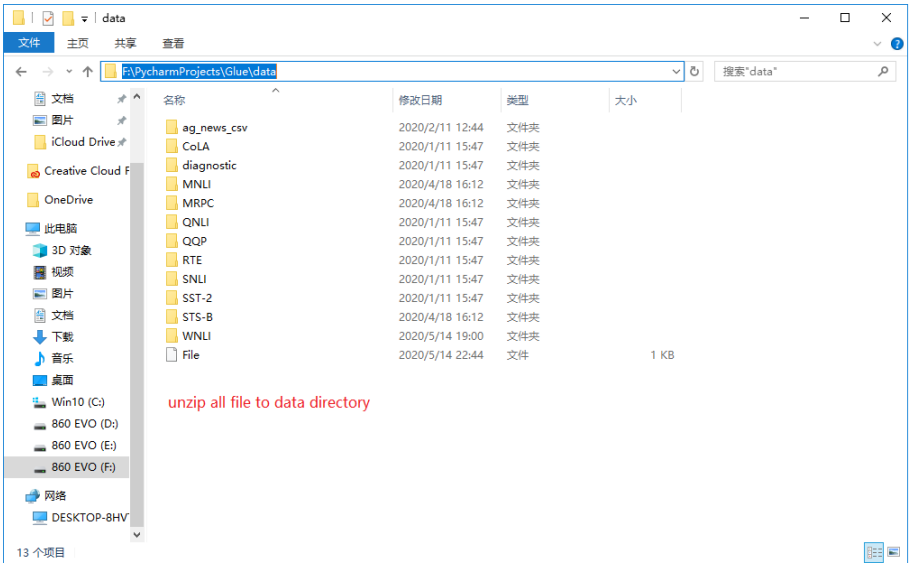

Figure 1: Download data from given url and unzip them into data/

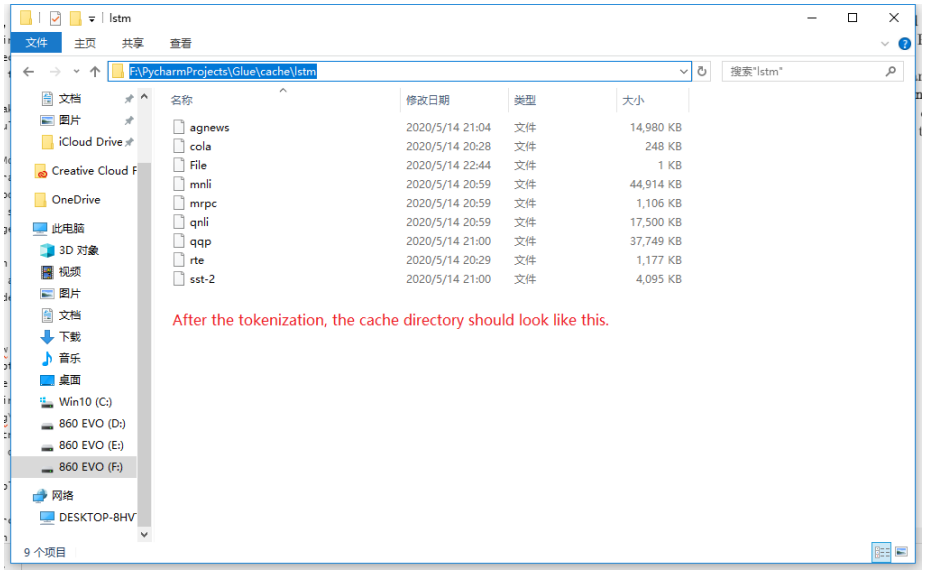

Figure 2: Use tokenization script to finish preprocessing.

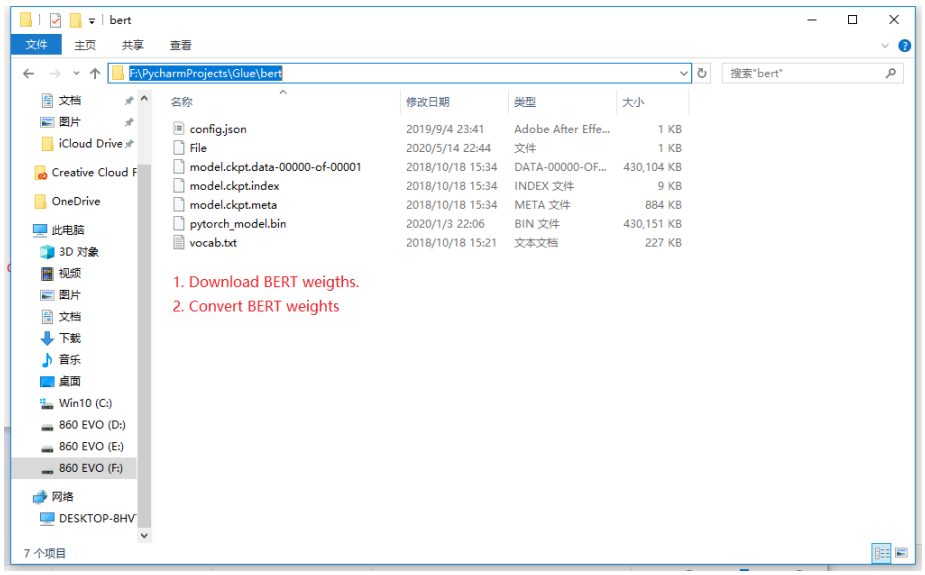

Figure 3: Donwload BERT model from given url, use conversation script to convert weights from tensorflow towards pytorch.(do not change output model name here)

2.2 Prepare BERT Model

You should download pre-trained BERT weights from `https://github.com/google-research/bert`. Notice that only 12-layer $BERT_{Base}$ is support for now, for 24-layer $BERT_{Large}$ you have to manually change the pruning schedule (see section 3).

There is no limit of which type of $BERT_{Base}$ model you are going to use, cased or uncased are all acceptable (Bert for other language is also fine). Unzip downloaded weights to bert/ directory, it should be look like figure 3 shows.

Converting tensorflow weights towards pytorch is necessary with our code. We offer a script named ConvertBert4Torch.py(downloaded form hugging face team) to make the conversation. This script needs 3 parameters: –tf_checkpoint_path indicates path to the TensorFlow checkpoint; –bert_config_file indicates the path of config json file corresponding to the pre-trained BERT model; –pytorch_dump_path indicates the path to the output PyTorch model.

We present a running example below(do not change output model name here):

python ConvertBert4Torch.py
–tf_checkpoint_path bert/model.ckpt
–pytorch_dump_path cache/pytorch_model.bin
–bert_config_file bert/config.json

After the conversation, you should have a file named pytorch_model.bin under bert/, it should look like figure 3.

Overwrite transformers library file is necessary however, cause we built our system based on their implementation of BERT. We have uploaded our modification towards their Repository, while they have not responded yet, which means you have to overwrite the library file transformers/modeling_bert.py with our file (provided with our project) ModifiedModelingBert.py. In case of unexpected error, you may want to backup your modeling_bert.py firstly.

3 Run Evaluation Scripts

3.1 Run & configure LSTM with sequence pruning

Use RunLstm4Glue.py, RunLstm4Agnews.py to test LSTM with sequence pruning. Again they don't need any parameter, so you can directly run them to see the evaluation result. Besides the running record will be saved at

records/LSTM/. We believe there is only one hyper parameter needs to be tuned somehow, which is the pruning rate used by sequence pruner.

To modify pruning rate, reader have to manually change the value of **target_prune_rate** in RunLstm4Glue.py and RunLstm4Agnews.py.

3.2 Run & configure BERT with sequence pruning

Use RunBert4Glue.py to test BERT with sequence pruning. Fortunately it doesn't require any parameter either, so you can directly run it and see the evaluation result. The running record will be saved at records/BERT/.

We have designed a delicate pruning schedule for BERT, to modify it reader must change the value of pruning_schedule in RunBert4Glue.py. It is a list containing pruning rate for each BERT layer, notice that the length of pruning schedule must match the number of BERT layers. Each value inside pruning_schedule denotes a pruning rate for corresponding BERT layer, pruning rate should be set between 0 and 1.

Notice that if pruning rate is set as 0, actually no pruner will be established for corresponding layer, otherwise a pruner is going to be built for this layer. Pruner will introduce extra computational and memory cost, we recommend to minimize the number of pruner.

We haven't support inject pruners with different pruning methods yet.