

**Bert** 主要包括有两个步骤:

**1、Pre-training:** 训练开销非常的大, 因为模型包括一个 12-24 层的 Transformer, 数据涵盖了整个维基百科和书的数据, 以及 1M 次的更新迭代。需要 4~16 个云端 TPUs 花费 4 天的时间。不过好在对于每种语言来说, 这样的训练都是一劳永逸的, 因为这是一个相当通用而且效果极佳的模型。

**2、fine-tuning:** 训练起来则方便的多, 在 GPU 上大概只需要数个小时即可。这主要是在一些特定任务上对 Bert 参数的微调。

Github 地址: <https://github.com/google-research/bert>

已经发布的代码介绍:

- **BERT-Base, Uncased:** 12-layer, 768-hidden, 12-heads, **110M** parameters
- **BERT-Large, Uncased:** 24-layer, 1024-hidden, 16-heads, **340M** parameters
- **BERT-Base, Cased:** 12-layer, 768-hidden, 12-heads, **110M** parameters
- **BERT-Large, Cased:** 24-layer, 1024-hidden, 16-heads, **340M** parameters
- **BERT-Base, Multilingual Cased (New, recommended):** 104 languages, 12-layer, 768-hidden, 12-heads, **110M** parameters
- **BERT-Base, Multilingual Uncased (Orig, not recommended) (Not recommended, use Multilingual Cased instead):** 102 languages, 12-layer, 768-hidden, 12-heads, **110M** parameters
- **BERT-Base, Chinese:** Chinese Simplified and Traditional, 12-layer, 768-hidden, 12-heads, **110M** parameters

**注释:** *uncased* 表示将数据里面的所有字母都变成了小写, 例如 John Smith 变成了 john smith, 同时还去掉了例如德语法语里面的音调符号。*Cased* 则保留了语言的以上信息。*Multilingual* 表示覆盖多语种, *layer* 表示 Transformer 层

每一个文件包\*\***.zip**

 里都包含以下三个部分:

- 1、一个 TensorFlow checkpoint (bert\_model.ckpt) 里面有预训练的所有权重, 一共 3 个文件;
- 2、一个 vocab.txt 文件, 是 wordpiece 到 word id 的对应关系。
- 3、一个 bert\_config.json 文件, 它指定了模型的超参数。

## Fine-tuning

因为模型较大，即使是 BERT-Base 也需要 12GB 的 RAM 的 GPU 来运行默认的超参数。否则很容易溢出。

## 案例：1

Sentence (and sentence-pair) 语义判别分类任务：

```
Quality #1 ID #2 ID #1 String #2 String
1 702976 702977 Amrozi accused his brother , whom he called " the witness " , of deliberately distorting his evidence . Referring to him as only " the witness " , Amrozi accused his brother of deliberately
0 2108705 2108831 Yucaipa owned Dominick 's before selling the chain to Safeway in 1998 for $ 2.5 billion . Yucaipa bought Dominick 's in 1998 for $ 693 million and sold it to Safeway for $ 1.8 billion in
1 1330391 1330521 They had published an advertisement on the Internet on June 10 , offering the cargo for sale , he added . On June 10 , the ship 's owners had published an advertisement on the Internet ,
0 3344667 3344668 Around 0335 GMT , Tab shares were up 19 cents , or 4.4 % , at $ 4.56 , having earlier set a record high of $ 4.57 . Tab shares jumped 20 cents , or 4.6 % , to set a record closing high
1 1236820 1236712 The stock rose $ 2.11 , or about 11 percent , to close Friday at $ 21.51 on the New York Stock Exchange . PG & E Corp. shares jumped $ 1.63 or 8 percent to $ 21.03 on the New York Stock E
```

两个句子，各有一个 ID，label 是 1/0 分别代表两个句子的语义是否一致。

训练脚本编写：

```
export BERT_BASE_DIR=/work4/caiyq/bert/model/uncased_L-12_H-768_A-12
export GLUE_DIR=/work4/caiyq/data/glue_data
python run_classifier.py \
  --task_name=MRPC \
  --do_train=true \
  --do_eval=true \
  --data_dir=$GLUE_DIR/MRPC \
  --vocab_file=$BERT_BASE_DIR/vocab.txt \
  --bert_config_file=$BERT_BASE_DIR/bert_config.json \
  --init_checkpoint=$BERT_BASE_DIR/bert_model.ckpt \
  --max_seq_length=128 \
  --train_batch_size=4 \ #调得过大很容易内存溢出
  --learning_rate=2e-5 \
  --num_train_epochs=3.0 \
  --output_dir=/work4/caiyq/bert/output/mrpc_output/
```

You should see output like this:

```
***** Eval results *****
eval_accuracy = 0.845588
eval_loss = 0.505248
global_step = 343
loss = 0.505248
*****
```

预测脚本编写：

```
export BERT_BASE_DIR=/work4/caiyq/bert/model/uncased_L-12_H-768_A-12
export GLUE_DIR=/work4/caiyq/data/glue_data
export export TRAINED_CLASSIFIER=/work4/caiyq/bert/output/mrpc_output
python run_classifier.py \
  --task_name=MRPC \
  --do_predict=true \
```

```
--data_dir=$GLUE_DIR/MRPC \  
--vocab_file=$BERT_BASE_DIR/vocab.txt \  
--bert_config_file=$BERT_BASE_DIR/bert_config.json \  
--init_checkpoint=$TRAINED_CLASSIFIER \  
#会自动去路径文件夹里面寻找 model.ckpt 文件  
--max_seq_length=128 \  
--output_dir=/work4/caiyq/bert/output/Pre_mrpc_output/
```