作业第六章

作业内容是使用 TensorRT Python API 手动搭建 BERT 模型。同学们不需要从零 开始搭建，课程组已经搭好了一个模型结构框架，同学们进行相应的填充即可。

模型架构框架：https://github.com/shenlan2017/TensorRT

一. 文件信息

1. model2onnx.py 使用 pytorch 运行 bert 模型，生成 demo 输入输出和 onnx 模型
2. onnx2trt.py　将 onnx，使用 onnx-parser 转成 trt 模型，并 infer
3. builder.py　输入 onnx 模型，并进行转换
4. trt_helper.py　对 trt 的 api 进行封装，方便调用
5. calibrator.py int8 calibrator 代码
6. 基础款 LayerNormPlugin.zip　用于学习的 layer_norm_plugin

二. 模型信息

2.1 介绍

1. 标准 BERT 模型，12 层, hidden_size = 768
2. 不考虑 tokensizer 部分，输入是 ids，输出是 score
3. 为了更好的理解，降低作业难度，将 mask 逻辑去除，只支持 batch=1 的输入

BERT 模型可以实现多种 NLP 任务，作业选用了 fill-mask 任务的模型

```
输入：
The capital of France, [mask], contains the Eiffel Tower.

topk10 输出：
The capital of France, paris, contains the Eiffel Tower.
The capital of France, lyon, contains the Eiffel Tower.
The capital of France,, contains the Eiffel Tower.
The capital of France, tolilleulouse, contains the Eiffel Tower.
The capital of France, marseille, contains the Eiffel Tower.
The capital of France, orleans, contains the Eiffel Tower.
The capital of France, strasbourg, contains the Eiffel Tower.
The capital of France, nice, contains the Eiffel Tower.
The capital of France, cannes, contains the Eiffel Tower.
The capital of France, versailles, contains the Eiffel Tower.
```

2.2 输入输出信息

输入

1. input_ids[1,-1]：int 类型，input ids，从 BertTokenizer 获得
2. token_type_ids[1, -1]：int 类型，全 0
3. position_ids[1, -1]：int 类型，[0, 1, ..., len(input_ids) - 1]

输出

1. logit[1, -1, 768]

三. 作业内容

3.1 学习使用trt python api搭建网络

填充 trt_helper.py 中的空白函数（addLinear， addSoftmax 等）。学习使用 api 搭建网络的过程。

3.2 编写 plugin

trt 不支持 layer_norm 算子，编写 layer_norm plugin，并将算子添加到网络中，进行验证。

1. 及格：将 “基础款 LayerNormPlugin.zip”中实现的基础版 layer_norm 算子 插入到 trt_helper.pyaddLayerNorm 函数中。
2. 优秀：将整个 layer_norm 算子实现到一个 kernel 中，并插入到 trt_helper.py addLayerNorm 函数中。可以使用 testLayerNormPlugin.py 对合并后的 plugin 进行单元测试验证。
3. 进阶：在 2 的基础上进一步优化，线索见 https://www.bilibili.com/video/BV1i3411G7vN

3.3 观察 GELU 算子的优化过程

1. GELU 算子使用一堆基础算子堆叠实现的（详细见 trt_helper.py addGELU 函数），直观上感觉很分散，计算量比较大。
2. 但在实际 build 过程中，这些算子会被合并成一个算子。build 过程中需要设置 log 为 trt.Logger.VERBOSE，观察 build 过程。
3. 体会 trt 在转换过程中的加速优化操作。