1. 下载模型

> 点击链接： https://huggingface.co/models， 选择 bert-base-uncased 模型，并学习 bert 模型结构（参考资料见“BERT 学习资料.pdf”）。
> 如何下载和在本地使用 Bert 预训练模型： https://blog.csdn.net/weixin_38481963/article/details/110535583

Bert 预训练模型的下载有许多方式，比如从github官网上下载（官网下载的是tensorflow版本的），还可以从源码中找到下载链接，然后手动下载，还可以从huggingface中下载。

1.1. git LFS安装

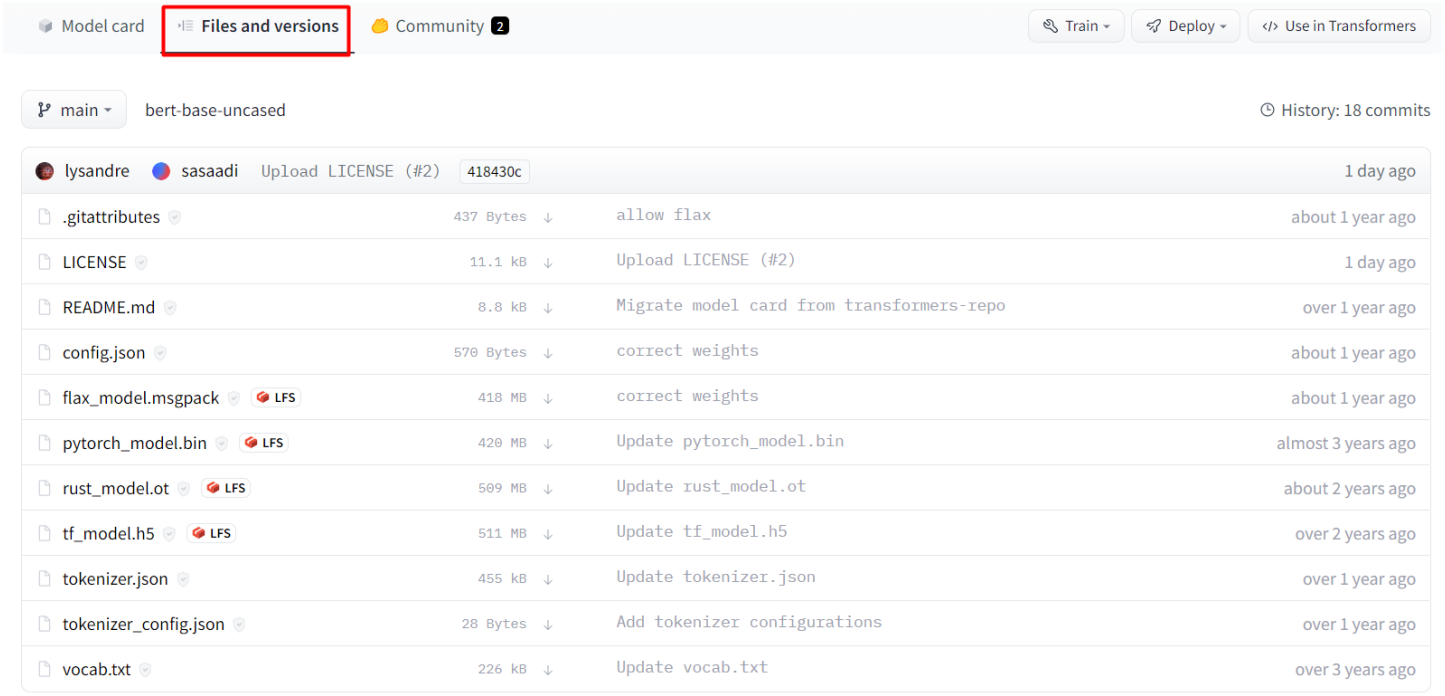
LFS是Large File Storage的缩写，用了帮助git管理大的文件。原理：不同于git每次保存diff，对于git来说，如果是模型或者一些设计大文件，改变一点，对于仓库来说会增加很大的体积，不一会就能几个G。对于git lfs来说，在使用git lfs track命令后，git push的时候，git lfs会截取要管理的大文件，并将其传至git lfs的服务器中，从而减小仓库的体积。

```
curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
sudo apt-get install git-lfs
git lfs install
```

```
[redacted]$ sudo apt-get install git-lfs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  git-lfs
0 upgraded, 1 newly installed, 0 to remove and 95 not upgraded.
Need to get 7,168 kB of archives.
After this operation, 15.6 MB of additional disk space will be used.
Get:1 https://packagecloud.io/github/git-lfs/ubuntu bionic/main amd64 git-lfs amd64 3.2.0 [7,168 kB]
Fetched 7,168 kB in 2s (3,824 kB/s)
Selecting previously unselected package git-lfs.
(Reading database ... 271147 files and directories currently installed.)
Preparing to unpack .../git-lfs_3.2.0_amd64.deb ...
Unpacking git-lfs (3.2.0) ...
Setting up git-lfs (3.2.0) ...
Git LFS initialized.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
[redacted]$ git lfs install
Git LFS initialized.
```

1.2. 模型文件下载

选择 bert-base-uncased 模型，点击链接（https://huggingface.co/bert-base-uncased），要下载的文件如下：



执行以下命令进行下载。

```
git lfs install
git clone https://huggingface.co/bert-base-uncased
```

```
[redacted] git clone https://huggingface.co/bert-base-uncased
Cloning into 'bert-base-uncased'...
remote: Enumerating objects: 55, done.
remote: Counting objects: 100% (55/55), done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 55 (delta 20), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (55/55), done.

Filtering content: 100% (4/4), 1.81 GiB | 10.32 MiB/s, done.
```

要耐心等待一段时间，文件在后台，悄悄下载。

下载完成后，文件详情如下图：

```
[redacted]/bert-base-uncased$ ls -lh
total 2.3G
-rw-rw-r-- 1 vcis1 vcis1 1.9M 6月   8 00:37 case_data.npz
-rw-rw-r-- 1 vcis1 vcis1  570 6月   8 00:31 config.json
-rw-rw-r-- 1 vcis1 vcis1 418M 6月   8 00:34 flax_model.msgpack
-rw-rw-r-- 1 vcis1 vcis1  12K 6月   8 00:31 LICENSE
-rw-rw-r-- 1 vcis1 vcis1 418M 6月   8 00:37 model.onnx
-rw-rw-r-- 1 vcis1 vcis1 421M 6月   8 00:34 pytorch_model.bin
-rw-rw-r-- 1 vcis1 vcis1 8.8K 6月   8 00:31 README.md
-rw-rw-r-- 1 vcis1 vcis1 510M 6月   8 00:34 rust_model.ot
-rw-rw-r-- 1 vcis1 vcis1 512M 6月   8 00:34 tf_model.h5
-rw-rw-r-- 1 vcis1 vcis1   28 6月   8 00:31 tokenizer_config.json
-rw-rw-r-- 1 vcis1 vcis1 456K 6月   8 00:31 tokenizer.json
-rw-rw-r-- 1 vcis1 vcis1 227K 6月   8 00:31 vocab.txt
```

2. 将模型转成 onnx 格式

> （1）不考虑 BertTokenizer 结构；
> （2）模型输入，格式为[batch_size, max_seq_len]，
> input_ids: [1, max_seq_len]
> token_type_ids: [1, max_seq_len] # 全 0
> input_mask: [1, max_seq_len]
> PS：固定 batch_size = 1，以降低作业难度。
> 模型转成 onnx 格式的实现代码见 Bertmodel 2 ONNX.py 文件 。
> 进阶任务： 使用 onnxruntime gpu 库，做 infer，得到运行时间 Tort，与后面的 trt 时间 进行对比。

2.1. 安装相应的库文件

```
# Install numpy, torch and transformers
pip install numpy
pip install torch
pip install transformers
pip install pycuda

# onnx依赖protobuf，所以需要先安装protobuf的库
sudo apt-get install libprotobuf-dev protobuf-compiler

# Install onnx related libraries
pip install onnx
pip install onnxcli
pip install onnx-simplifier
pip install onnxruntime-gpu
```

2.2. 模型转成 onnx 格式

使用老师提供的 Bertmodel 2 ONNX.py 文件，进行模型转换。

```
[redacted]/hw_5$ python Bertmodel_2_ONNX.py
2022-06-08 00:37:33.858949: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library libcudart.so.11.0
pytorch: 1.7.1+cu110
transformers: 4.19.2
Some weights of the model checkpoint at bert-base-uncased were not used when initializing BertForMaskedLM: ['cls.seq_relationship.weight', 'cls.seq_relationship.bias']
- This IS expected if you are initializing BertForMaskedLM from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertForMaskedLM from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).
==============model test==================
torch.Size([1, 16, 30522])
model test topk10 output:
The capital of France, paris, contains the Eiffel Tower.
The capital of France, lyon, contains the Eiffel Tower.
The capital of France, lille, contains the Eiffel Tower.
The capital of France, toulouse, 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.
Saving inputs and output to case_data.npz ...
tensor([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15]],
       dtype=torch.int32)
(1, 16)
[[  101  1996  3007  1997  2605  1010   103  1010  3397  1996  1041 13355
   2884  3578  1012   102]]
==================model2onnx======================
{'input_ids': tensor([[  101,  1996,  3007,  1997,  2605,  1010,   103,  1010,  3397,  1996,
         1041, 13355,  2884,  3578,  1012,   102]]), 'token_type_ids': tensor([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]), 'attention_mask': tensor([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]])}
/home/vcis1/anaconda3/envs/Wangjia01/lib/python3.8/site-packages/transformers/models/bert/modeling_bert.py:221: TracerWarning: Converting a tensor to a Python index might cause the trace to be incorrect. We can't record the data flow of Python values, so this value will be treated as a constant in the future. This means that the trace might not generalize to other inputs!
  position_ids = self.position_ids[:, past_key_values_length : seq_length + past_key_values_length]
Model exported at  bert-base-uncased/model.onnx
```

2.3. 使用 onnxruntime gpu 库，做 infer

创建对应的文件，进行测试。

```
[redacted]$ python bertmodel_onnx_infer.py
2022-06-09 00:40:21.923209: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library libcudart.so.11.0
pytorch: 1.7.1+cu110
onnxruntime version: 1.8.1
onnxruntime device: CPU
transformers: 4.19.2
/home/vcis1/.local/lib/python3.8/site-packages/onnxruntime/capi/onnxruntime_inference_collection.py:53: UserWarning: Specified provider 'CUDAExecutionProvider' is not in available provider names.Available providers: 'CPUExecutionProvider'
  warnings.warn("Specified provider '{}' is not in available provider names."
input_ids
attention_mask
token_type_ids
*xiaofeng* Tort: 0.015244722366333008
torch.Size([1, 16, 30522])
model test topk10 output:
The capital of France, paris, contains the Eiffel Tower.
The capital of France, lyon, contains the Eiffel Tower.
The capital of France, lille, contains the Eiffel Tower.
The capital of France, toulouse, 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.
```

3. 使用 onnxparser 将 onnx 模型转成 trt plan 模型

> 备注：建议使用 python api，不建议使用 trtexec，太黑盒，不利于学习。
>
> （1）下载 TensorRT： C++ api 直接使用库就行， python api 需要安装对应的 whil;

（2）使用 onn-simplifer 模型对 onnx 模型进行优化，得到 model-sim.onnx。需要进行 此步，否则后面的转换会失败。

onnxsim bert-base-uncased/model.onnx bert-base-uncased/model-sim.onnx --input-shape input_ids:1,12 token_type_ids:1,12 input_mask:1,12 --dynamic-inputshape

（3）调用 onnx parser python or c++ api，将 model-sim.onnx 转换成 model.plan;

（4）测速 。使用 c++ 或者 python api 编写测速代码，得到时间 Ttrt。 建议使用 c++ api，毕竟一般 上线都是用 c++。

可参考：https://hemanths933.medium.com/convert-onnx-bert-model-to-tensorrt-e809276b01b6

3.1. 下载 TensorRT，搭建环境

参照后续章节，进行安装测试。

3.2. 使用onnx-simplifer 模型对 onnx 模型进行优化

```
onnxsim model.onnx model-sim.onnx --input-shape input_ids:1,16 token_type_ids:1,16 attention_mask:1,16 --dynamic-input-shape
```

```
onnxsim model.
n_type_ids:1,16 attention_mask:1,16 --dynamic-input-shape
Simplifying...
Finish! Here is the difference:
```

	Original Model	Simplified Model
Add	178	178
Cast	53	1
Concat	48	48
Constant	290	0
Div	51	51
Erf	13	13
Gather	100	100
MatMul	98	98
Mul	53	53
Pow	26	26
ReduceMean	52	52
Reshape	48	48
Shape	97	97
Slice	1	1
Softmax	12	12
Sqrt	26	26
Sub	27	27
Transpose	122	48
Unsqueeze	185	99
Model Size	417.9MiB	507.3MiB

3.3. 将 onnx 模型转成 trt plan 模型

调用 onnx parser python or c++ api，将 model-sim.onnx 转换成 trt plan 模型。（**PS：实现该文件为作业的关键考核内容，请大家认真完成作业。**）

```
[redacted]$ python onnx_to_trt.py
[06/08/2022-01:53:47] [TRT] [W] onnx2trt_utils.cpp:366: Your ONNX model has been generated with INT64 weights, while TensorRT does not native
INT32.
[06/08/2022-01:53:50] [TRT] [W] Output type must be INT32 for shape outputs
[06/08/2022-01:53:50] [TRT] [W] Output type must be INT32 for shape outputs
[06/08/2022-01:53:50] [TRT] [W] Output type must be INT32 for shape outputs
[06/08/2022-01:53:50] [TRT] [W] Output type must be INT32 for shape outputs
network.num_layers: 1982
onnx_to_trt.py:31: DeprecationWarning: Use build_serialized_network instead.
  engine = builder.build_engine(network, config=config)
[06/08/2022-01:53:52] [TRT] [W] TensorRT was linked against cuBLAS/cuBLASLt 11.6.5 but loaded cuBLAS/cuBLASLt 11.2.0
[06/08/2022-01:56:08] [TRT] [W] TensorRT was linked against cuBLAS/cuBLASLt 11.6.5 but loaded cuBLAS/cuBLASLt 11.2.0
Saved model to bert-base-uncased/engine.trt
```

3.4. 测速

使用 c++ 或者 python api 编写测速代码，得到时间 Ttrt。

```
[redacted]5$ python3 trt_infer.py
2022-06-08 02:03:38.167284: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library libcudart.so.11.0
[06/08/2022-02:03:41] [TRT] [I] [MemUsageChange] Init CUDA: CPU +456, GPU +0, now: CPU 562, GPU 512 (MiB)
[06/08/2022-02:03:42] [TRT] [I] Loaded engine size: 1017 MiB
[06/08/2022-02:03:42] [TRT] [V] Using cublasLt as a tactic source
[06/08/2022-02:03:42] [TRT] [W] TensorRT was linked against cuBLAS/cuBLASLt 11.6.5 but loaded cuBLAS/cuBLASLt 11.2.0
[06/08/2022-02:03:42] [TRT] [I] [MemUsageChange] Init cuBLAS/cuBLASLt: CPU +630, GPU +262, now: CPU 2719, GPU 1282 (MiB)
[06/08/2022-02:03:42] [TRT] [V] Using cuDNN as a tactic source
[06/08/2022-02:03:43] [TRT] [I] [MemUsageChange] Init cuDNN: CPU +663, GPU +268, now: CPU 3382, GPU 1550 (MiB)
[06/08/2022-02:03:43] [TRT] [V] Deserialization required 1186417 microseconds.
[06/08/2022-02:03:43] [TRT] [I] [MemUsageChange] TensorRT-managed allocation in engine deserialization: CPU +0, GPU +507, now: CPU 0, GPU 507 (MiB)
[06/08/2022-02:03:43] [TRT] [V] Using cublasLt as a tactic source
[06/08/2022-02:03:43] [TRT] [W] TensorRT was linked against cuBLAS/cuBLASLt 11.6.5 but loaded cuBLAS/cuBLASLt 11.2.0
[06/08/2022-02:03:43] [TRT] [I] [MemUsageChange] Init cuBLAS/cuBLASLt: CPU +0, GPU +10, now: CPU 2364, GPU 1542 (MiB)
[06/08/2022-02:03:43] [TRT] [V] Using cuDNN as a tactic source
[06/08/2022-02:03:43] [TRT] [I] [MemUsageChange] Init cuDNN: CPU +0, GPU +8, now: CPU 2364, GPU 1550 (MiB)
[06/08/2022-02:03:43] [TRT] [V] Total per-runner device persistent memory is 0
[06/08/2022-02:03:43] [TRT] [V] Total per-runner host persistent memory is 192
[06/08/2022-02:03:43] [TRT] [V] Allocated activation device memory of size 24205312
[06/08/2022-02:03:44] [TRT] [I] [MemUsageChange] TensorRT-managed allocation in IExecutionContext creation: CPU +0, GPU +23, now: CPU 0, GPU 530 (MiB)
*xiaofeng* Ttrt: 0.0025420188903808594
```

4. 参考：TensorRT环境搭建

首先，根据需要的版本以及 CUDA版本，系统版本来选择下载的文件（链接：https://developer.nvidia.com/nvidia-tensorrt-download）。

NVIDIA TensorRT Download

NVIDIA TensorRT is a high-performance deep learning inference optimizer and runtime for deep learning applications.

TensorRT works across all NVIDIA GPUs using the CUDA platform. The following files are for use for Linux servers and workstati
NVIDIA recommends Tesla V100, P100, P4, and P40 GPUs for production deployment.

Ethical AI

NVIDIA's platforms and application frameworks enable developers to build a wide array of AI applications. Consider potential alg
deployed. Work with the model's developer to ensure that it meets the requirements for the relevant industry and use case; that
to understand error rates, confidence intervals, and results; and that the model is being used under the conditions and in the ma

Available Versions

- TensorRT 8
- TensorRT 7
- TensorRT 6
- TensorRT 5
- TensorRT 4
- TensorRT 3
- TensorRT 2
- TensorRT 1

这里选择TensorRT8.x来进行讲解。

4.1. C++ API 环境搭建

NVIDIA TensorRT 8.x Download

NVIDIA TensorRT is a platform for high performance deep learning inference.

TensorRT works across all NVIDIA GPUs using the CUDA platform. NVIDIA recommends A100, A30, A10 and T4 GPUs for production deployment.

☑ **I Agree To the Terms of the NVIDIA TensorRT License Agreement**
Please download the version compatible with your development environment.

TensorRT 8.4 EA

TensorRT 8.2 GA Update 3

Documentation

- Online Documentation

TensorRT 8.2 GA Update 3 for x86_64 Architecture

Debian, RPM, and TAR Install Packages for Linux

- TensorRT 8.2 GA Update 3 for Linux x86_64 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 TAR Package
- TensorRT 8.2 GA Update 3 for Ubuntu 20.04 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 DEB local repo Package
- TensorRT 8.2 GA Update 3 for Ubuntu 18.04 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 DEB local repo Package
- TensorRT 8.2 GA Update 3 for CentOS / RedHat 7 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 RPM local repo Package
- TensorRT 8.2 GA Update 3 for CentOS / RedHat 8 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 RPM local repo Package

然后，安装TensorRT安装包，同时注意自己选择的系统版本、CUDA版本、TensorRT版本。

```
sudo dpkg -i nv-tensorrt-repo-ubuntu1804-cuda11.4-trt8.2.4.2-ga-20220324_1-1_amd64.deb
sudo apt-key add /var/nv-tensorrt-repo-ubuntu1804-cuda11.4-trt8.2.4.2-ga-20220324/7fa2af80.pub
sudo apt-get update
sudo apt-get install tensorrt
```

安装时，如果出现以下问题：

```
(base) vcis1@vcis1:~/Downloads$ sudo apt-get install tensorrt
Reading package lists... Done
Building dependency tree
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
 tensorrt : Depends: libnvinfer8 (= 8.2.4-1+cuda11.4) but it is not going to be installed
            Depends: libnvinfer-plugin8 (= 8.2.4-1+cuda11.4) but it is not going to be installed
            Depends: libnvparsers8 (= 8.2.4-1+cuda11.4) but it is not going to be installed
            Depends: libnvonnxparsers8 (= 8.2.4-1+cuda11.4) but it is not going to be installed
            Depends: libnvinfer-bin (= 8.2.4-1+cuda11.4) but it is not going to be installed
            Depends: libnvinfer-dev (= 8.2.4-1+cuda11.4) but it is not going to be installed
            Depends: libnvinfer-plugin-dev (= 8.2.4-1+cuda11.4) but it is not going to be installed
            Depends: libnvparsers-dev (= 8.2.4-1+cuda11.4) but it is not going to be installed
            Depends: libnvonnxparsers-dev (= 8.2.4-1+cuda11.4) but it is not going to be installed
            Depends: libnvinfer-samples (= 8.2.4-1+cuda11.4) but it is not going to be installed
            Depends: libnvinfer-doc (= 8.2.4-1+cuda11.4) but it is not going to be installed
E: Unable to correct problems, you have held broken packages.
```

进入/var/nv-tensorrt-repo-ubuntu1804-cuda11.4-trt8.2.4.2-ga-20220324/文件夹，依次安装各个安装包。

```
(base) vcis1@vcis1:/var/nv-tensorrt-repo-ubuntu1804-cuda11.4-trt8.2.4.2-ga-20220324$ ls
7fa2af80.pub                                libnvinfer-plugin8_8.2.4-1+cuda11.4_amd64.deb     Packages.gz
graphsurgeon-tf_8.2.4-1+cuda11.4_amd64.deb  libnvinfer-plugin-dev_8.2.4-1+cuda11.4_amd64.deb  python3-libnvinfer_8.2.4-1+cuda11.4_amd64.deb
libcudnn8_8.2.1.32-1+cuda11.3_amd64.deb     libnvinfer-samples_8.2.4-1+cuda11.4_all.deb       python3-libnvinfer-dev_8.2.4-1+cuda11.4_amd64.deb
libcudnn8-dev_8.2.1.32-1+cuda11.3_amd64.deb libnvonnxparsers8_8.2.4-1+cuda11.4_amd64.deb      Release
libnvinfer8_8.2.4-1+cuda11.4_amd64.deb      libnvonnxparsers-dev_8.2.4-1+cuda11.4_amd64.deb   Release.gpg
libnvinfer-bin_8.2.4-1+cuda11.4_amd64.deb   libnvparsers8_8.2.4-1+cuda11.4_amd64.deb          tensorrt_8.2.4.2-1+cuda11.4_amd64.deb
libnvinfer-dev_8.2.4-1+cuda11.4_amd64.deb   libnvparsers-dev_8.2.4-1+cuda11.4_amd64.deb       uff-converter-tf_8.2.4-1+cuda11.4_amd64.deb
libnvinfer-doc_8.2.4-1+cuda11.4_all.deb     onnx-graphsurgeon_8.2.4-1+cuda11.4_amd64.deb
```

如果安装过程中，出现以下问题，需要去下载相应的库文件（链接：https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/）。要有耐心，会出现层层套娃的现象。

```
(base) vcis1@vcis1:/var/nv-tensorrt-repo-ubuntu1804-cuda11.4-trt8.2.4.2-ga-20220324$ sudo dp
(Reading database ... 256997 files and directories currently installed.)
Preparing to unpack libnvinfer8_8.2.4-1+cuda11.4_amd64.deb ...
Unpacking libnvinfer8 (8.2.4-1+cuda11.4) over (8.2.4-1+cuda11.4) ...
dpkg: dependency problems prevent configuration of libnvinfer8:
 libnvinfer8 depends on libcublas.so.11 | libcublas-11-1 | libcublas-11-0; however:
  Package libcublas.so.11 is not installed.
  Package libcublas-11-1 is not installed.
  Package libcublas-11-0 is not installed.

dpkg: error processing package libnvinfer8 (--install):
 dependency problems - leaving unconfigured
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
Errors were encountered while processing:
 libnvinfer8
```

```
(base) vcis1@vcis1:/var/nv-tensorrt-repo-ubuntu1804-cuda11.4-trt8.2.4.2-ga-20220324$ sudo dpkg -i libnvinfer-samples_8.2.4-1+cuda11.4_all.deb
Selecting previously unselected package libnvinfer-samples.
(Reading database ... 258106 files and directories currently installed.)
Preparing to unpack libnvinfer-samples_8.2.4-1+cuda11.4_all.deb ...
Unpacking libnvinfer-samples (8.2.4-1+cuda11.4) ...
dpkg: dependency problems prevent configuration of libnvinfer-samples:
 libnvinfer-samples depends on libcudart.so.11.0-dev | cuda-cudart-dev-11-1 | cuda-cudart-dev-11-0 | cuda-cudart-cross-amd64-11-4; however:
  Package libcudart.so.11.0-dev is not installed.
  Package cuda-cudart-dev-11-1 is not installed.
  Package cuda-cudart-dev-11-0 is not installed.
  Package cuda-cudart-cross-amd64-11-4 is not installed.
 libnvinfer-samples depends on cuda-nvcc-11-1 | cuda-nvcc-11-2 | cuda-nvcc-11-3 | cuda-nvcc-11-4 | cuda-nvcc-11-0; however:
  Package cuda-nvcc-11-1 is not installed.
  Package cuda-nvcc-11-2 is not installed.
  Package cuda-nvcc-11-3 is not installed.
  Package cuda-nvcc-11-4 is not installed.
  Package cuda-nvcc-11-0 is not installed.

dpkg: error processing package libnvinfer-samples (--install):
 dependency problems - leaving unconfigured
Errors were encountered while processing:
 libnvinfer-samples
```

如果安装过程中，出现以下问题，需要先去安装对应的库文件。

```
(base) vcis1@vcis1:/var/nv-tensorrt-repo-ubuntu1804-cuda11.4-trt8.2.4.2-ga-20220324$ sudo dpkg -i libnvinfer-bin_8.2.4-1+cuda11.4_amd64.deb
Selecting previously unselected package libnvinfer-bin.
(Reading database ... 257011 files and directories currently installed.)
Preparing to unpack libnvinfer-bin_8.2.4-1+cuda11.4_amd64.deb ...
Unpacking libnvinfer-bin (8.2.4-1+cuda11.4) ...
dpkg: dependency problems prevent configuration of libnvinfer-bin:
 libnvinfer-bin depends on libnvinfer-plugin8 (= 8.2.4-1+cuda11.4); however:
  Package libnvinfer-plugin8 is not installed.
 libnvinfer-bin depends on libnvparsers8 (= 8.2.4-1+cuda11.4); however:
  Package libnvparsers8 is not installed.
 libnvinfer-bin depends on libnvonnxparsers8 (= 8.2.4-1+cuda11.4); however:
  Package libnvonnxparsers8 is not installed.

dpkg: error processing package libnvinfer-bin (--install):
 dependency problems - leaving unconfigured
Errors were encountered while processing:
 libnvinfer-bin
```

```
(base) vcis1@vcis1:/var/nv-tensorrt-repo-ubuntu1804-cuda11.4-trt8.2.4.2-ga-20220324$ sudo dpkg -i libnvinfer-bin_8.2.4-1+cuda11.4_amd64.deb
(Reading database ... 258094 files and directories currently installed.)
Preparing to unpack libnvinfer-bin_8.2.4-1+cuda11.4_amd64.deb ...
Unpacking libnvinfer-bin (8.2.4-1+cuda11.4) over (8.2.4-1+cuda11.4) ...
dpkg: dependency problems prevent configuration of libnvinfer-bin:
 libnvinfer-bin depends on libnvparsers8 (= 8.2.4-1+cuda11.4); however:
  Package libnvparsers8 is not installed.
 libnvinfer-bin depends on libnvonnxparsers8 (= 8.2.4-1+cuda11.4); however:
  Package libnvonnxparsers8 is not installed.

dpkg: error processing package libnvinfer-bin (--install):
 dependency problems - leaving unconfigured
Errors were encountered while processing:
 libnvinfer-bin
```

安装完成后，可以进行测试

```
sudo cp -r /usr/src/tensorrt/ ~/
cd ~/tensorrt/samples
sudo make
```

```
cd ../bin
./sample_mnist
```

运行效果如下：

```
[04/30/2022-23:04:51] [I] [TRT] Loaded engine size: 1 MiB
[04/30/2022-23:04:51] [W] [TRT] TensorRT was linked against cuBLAS/cuBLASLt 11.6.5 but loaded cuBLAS/cuBLASLt 11.2.0
[04/30/2022-23:04:51] [I] [TRT] [MemUsageChange] Init cuBLAS/cuBLASLt: CPU +1, GPU +8, now: CPU 2835, GPU 13667 (MiB)
[04/30/2022-23:04:51] [I] [TRT] [MemUsageChange] Init cuDNN: CPU +0, GPU +8, now: CPU 2835, GPU 13675 (MiB)
[04/30/2022-23:04:51] [I] [TRT] [MemUsageChange] TensorRT-managed allocation in engine deserialization: CPU +0, GPU +1, now: CPU 0, GPU 1 (MiB)
[04/30/2022-23:04:51] [W] [TRT] TensorRT was linked against cuBLAS/cuBLASLt 11.6.5 but loaded cuBLAS/cuBLASLt 11.2.0
[04/30/2022-23:04:51] [I] [TRT] [MemUsageChange] Init cuBLAS/cuBLASLt: CPU +1, GPU +8, now: CPU 2678, GPU 13639 (MiB)
[04/30/2022-23:04:51] [I] [TRT] [MemUsageChange] Init cuDNN: CPU +0, GPU +8, now: CPU 2678, GPU 13647 (MiB)
[04/30/2022-23:04:51] [I] [TRT] [MemUsageChange] TensorRT-managed allocation in IExecutionContext creation: CPU +0, GPU +2, now: CPU 0, GPU 3 (MiB)
[04/30/2022-23:04:51] [I] Input:
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@+ @@@@@@@@@@@@@@
@@@@@@@@@@@. @@@@@@@@@@@@@@
@@@@@@@@@@@- @@@@@@@@@@@@@@
@@@@@@@@@@#  @@@@@@@@@@@@@@
@@@@@@@@@@#  *@@@@@@@@@@@@@
@@@@@@@@@@   :@@@@@@@@@@@@@
@@@@@@@@@@= .@@@@@@@@@@@@@@
@@@@@@@@@@#  %@@@@@@@@@@@@@
@@@@@@@@@@% .@@@@@@@@@@@@@@
@@@@@@@@@@%  %@@@@@@@@@@@@@
@@@@@@@@@@%  %@@@@@@@@@@@@@
@@@@@@@@@@@= +@@@@@@@@@@@@@
@@@@@@@@@@@* -@@@@@@@@@@@@@
@@@@@@@@@@@*  @@@@@@@@@@@@@
@@@@@@@@@@@@  @@@@@@@@@@@@@
@@@@@@@@@@@@  *@@@@@@@@@@@@
@@@@@@@@@@@@  *@@@@@@@@@@@@
@@@@@@@@@@@@  *@@@@@@@@@@@@
@@@@@@@@@@@@  *@@@@@@@@@@@@
@@@@@@@@@@@@* @@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@

[04/30/2022-23:04:51] [I] Output:
0:
1: **********
2:
3:
4:
5:
6:
7:
8:
9:

&&&& PASSED TensorRT.sample_mnist [TensorRT v8204] # ../bin/sample_mnist
```

4.2. Python API 环境搭建

TensorRT 8.2 GA Update 3

Documentation

- Online Documentation

TensorRT 8.2 GA Update 3 for x86_64 Architecture

Debian, RPM, and TAR Install Packages for Linux

- TensorRT 8.2 GA Update 3 for Linux x86_64 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 TAR Package
- TensorRT 8.2 GA Update 3 for Ubuntu 20.04 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 DEB local repo Package
- TensorRT 8.2 GA Update 3 for Ubuntu 18.04 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 DEB local repo Package
- TensorRT 8.2 GA Update 3 for CentOS / RedHat 7 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 RPM local repo Package
- TensorRT 8.2 GA Update 3 for CentOS / RedHat 8 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 RPM local repo Package
- TensorRT 8.2 GA Update 3 for Linux x86_64 and CUDA 10.2 TAR Package
- TensorRT 8.2 GA Update 3 for Ubuntu 18.04 and CUDA 10.2 DEB local repo Package
- TensorRT 8.2 GA Update 3 for CentOS / RedHat 7 and CUDA 10.2 RPM local repo Package
- TensorRT 8.2 GA Update 3 for CentOS / RedHat 8 and CUDA 10.2 RPM local repo Package

Zip Packages for Windows

- TensorRT 8.2 GA Update 3 for Windows 10 and CUDA 11.0, 11.1, 11.2, 11.3, 11.4 and 11.5 ZIP Package
- TensorRT 8.2 GA Update 3 for Windows 10 and CUDA 10.2 ZIP Package

下载获取的文件 TensorRT-8.2.4.2.Linux.x86_64-gnu.cuda-11.4.cudnn8.2.tar.gz , 执行命令解压文件。

```
tar -xvzf TensorRT-8.2.4.2.Linux.x86_64-gnu.cuda-11.4.cudnn8.2.tar.gz
cd TensorRT-8.2.4.2/python
ls
```

```
(base) vcis1@vcis1:~/Downloads/TensorRT-8.2.4.2/python$ ls
tensorrt-8.2.4.2-cp36-none-linux_x86_64.whl  tensorrt-8.2.4.2-cp38-none-linux_x86_64.whl
tensorrt-8.2.4.2-cp37-none-linux_x86_64.whl  tensorrt-8.2.4.2-cp39-none-linux_x86_64.whl
```

安装对应Python 版本的文件。

```
pip install tensorrt-8.2.4.2-cp38-none-linux_x86_64.whl
```

安装完成后，进行测试验证。

```
[redacted]:~/Downloads/TensorRT-8.2.4.2/python$ python
Python 3.8.0 | packaged by conda-forge | (default, Nov 22 2019, 19:11:38)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorrt
>>> print (tensorrt.__version__)
8.2.4.2
>>>
```

更多细节及使用说明，参照如下路径下的文档。

```
vcis1@vcis1:~/Downloads/TensorRT-8.2.4.2/doc/pdf$ ls
TensorRT-Developer-Guide.pdf     TensorRT-Quick-Start-Guide.pdf     TensorRT-SLA.pdf
TensorRT-Installation-Guide.pdf  TensorRT-Sample-Support-Guide.pdf  TensorRT-Support-Matrix-Guide.pdf
```