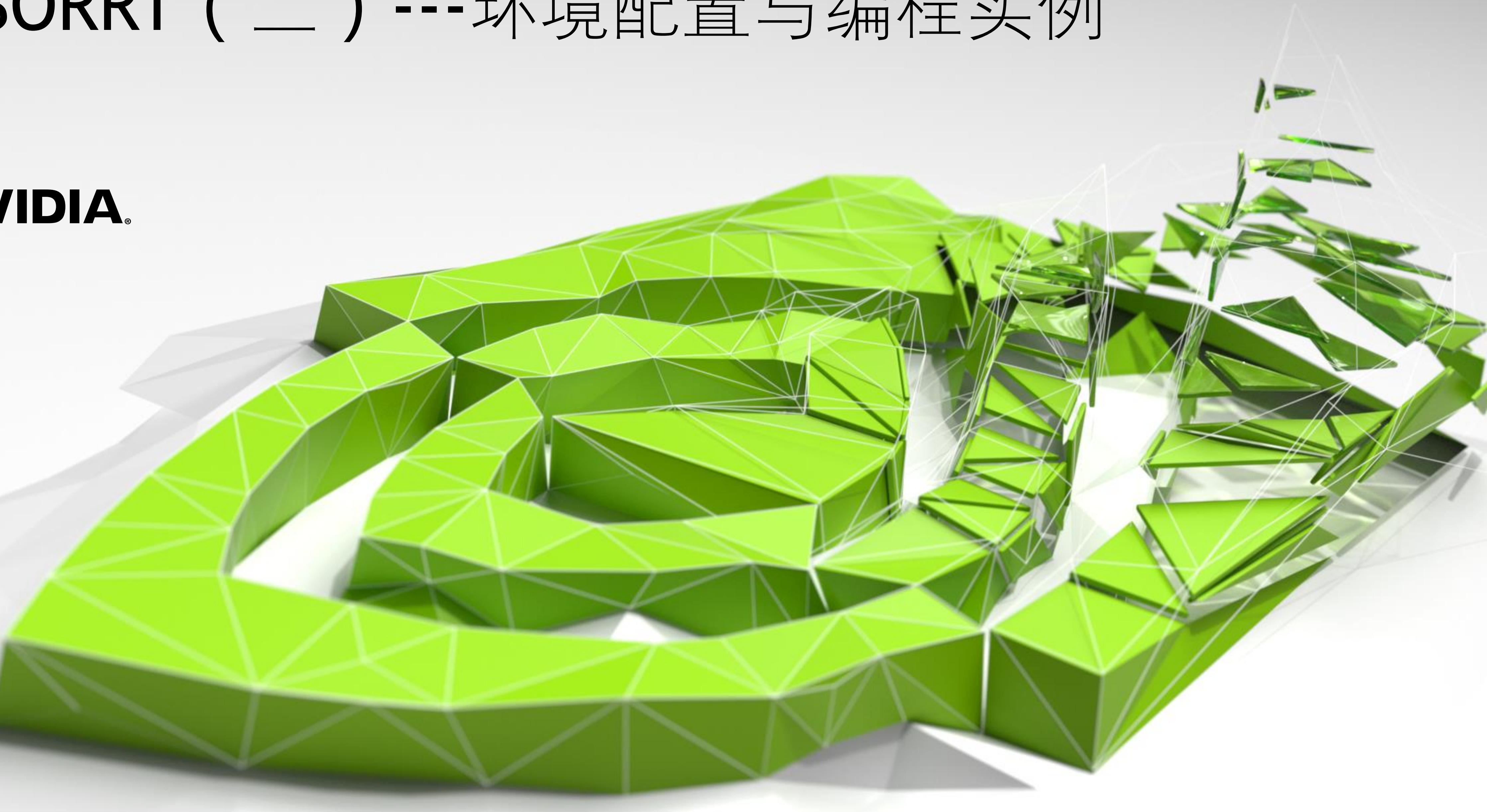
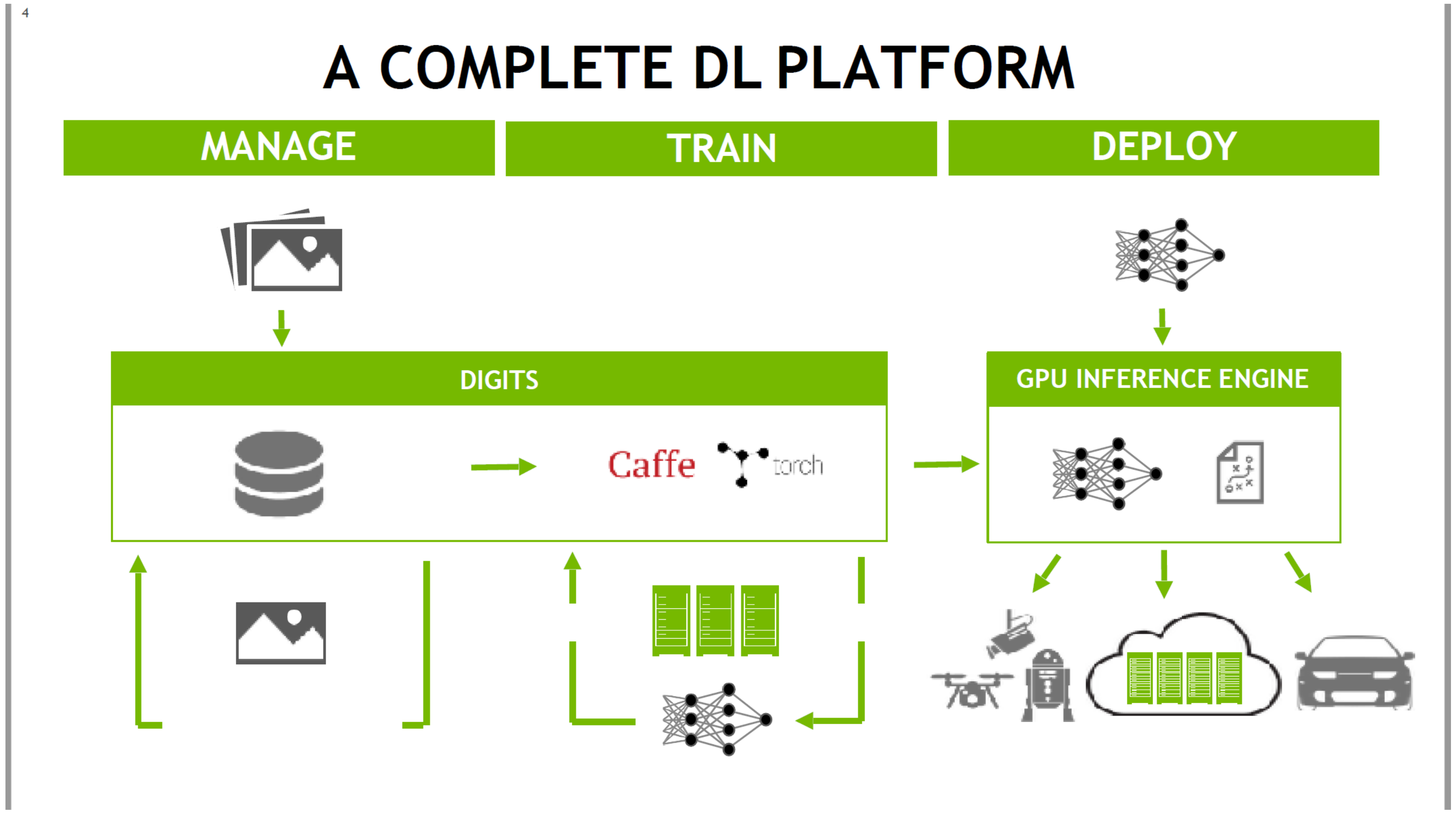
# TENSORRT（二）---环境配置与编程实例



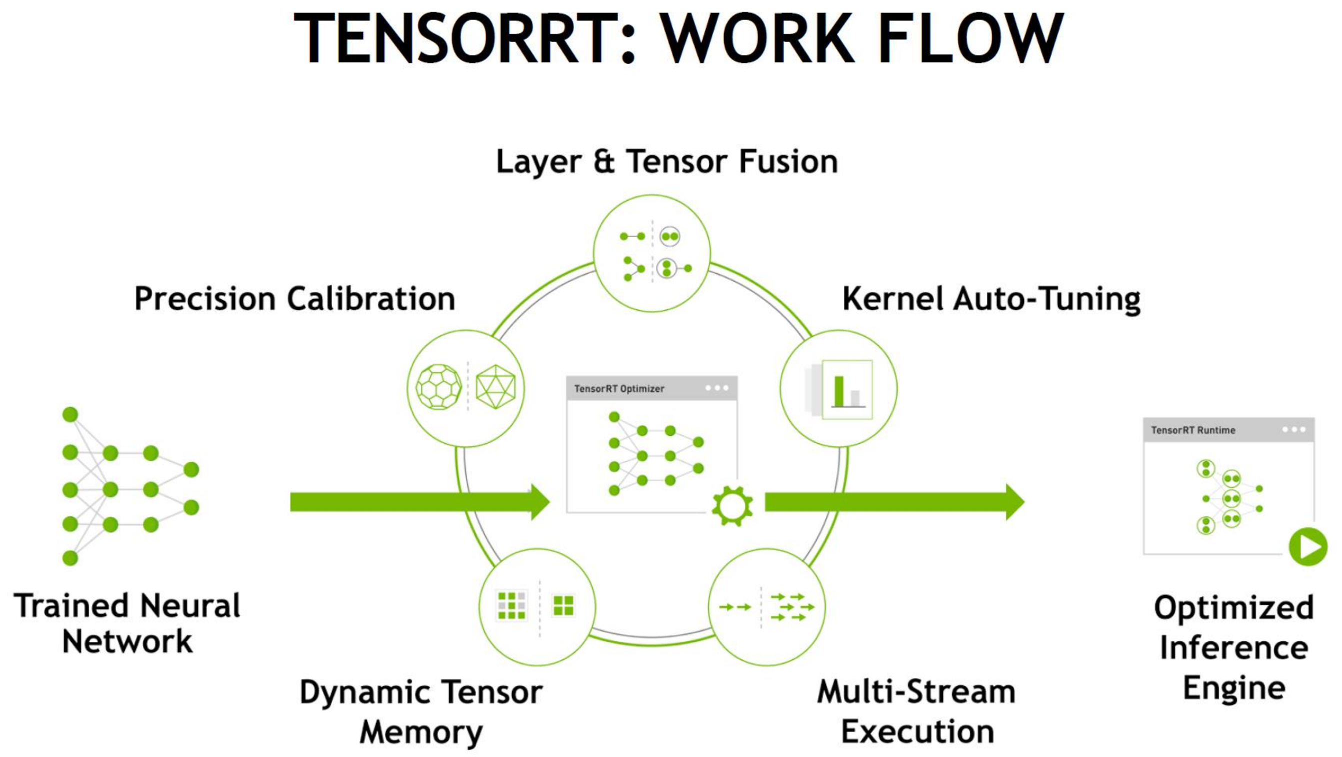
## 概述

- TensoRT回顾
- TensorRT环境配置与安装
- TensorRT编程模型
- TensoRT实例展示

## TensorRT回顾


4
A COMPLETE DL PLATFORM
MANAGE
TRAIN
DEPLOY
DIGITS
Caffe
torch
GPU INFERENCE ENGINE

# TensorRT回顾


TENSORRT: WORK FLOW
Layer & Tensor Fusion
Precision Calibration
Kernel Auto-Tuning
TensorRT Optimizer
TensorRT Runtime
Trained Neural Network
Dynamic Tensor Memory
Multi-Stream Execution
Optimized Inference Engine

## TensorRT环境配置与安装

1.CUDA

去官网下载cuda，此处使用的是cuda9.0版本的deb安装方式：

```
~$ mkdir Downloads
~$ cd Downloads
~$ wget http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_9.0.176-1_amd64.deb
~$ sudo dpkg -i cuda-repo-ubuntu1604_9.0.176-1_amd64.deb
~$ sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub
~$ sudo apt-get update
~$ sudo apt-get install cuda
```

## TensorRT环境配置与安装

接下来，在环境变量中添加CUDA：

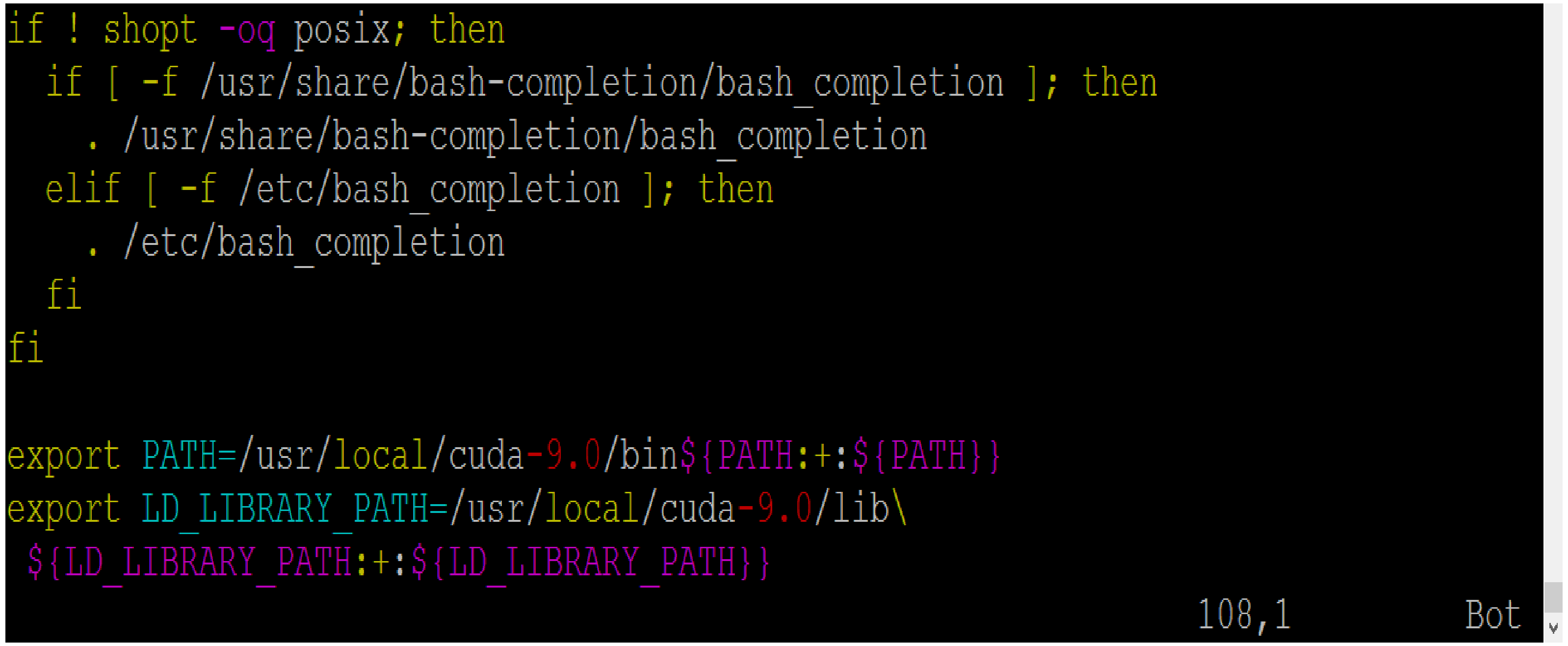
~$ sudo vim ~/.bashrc

在最后一行添加cuda的安装路径：

export PATH=/usr/local/cuda-9.0/bin${PATH:+:${PATH}}

export LD_LIBRARY_PATH=/usr/local/cuda-9.0/lib64\

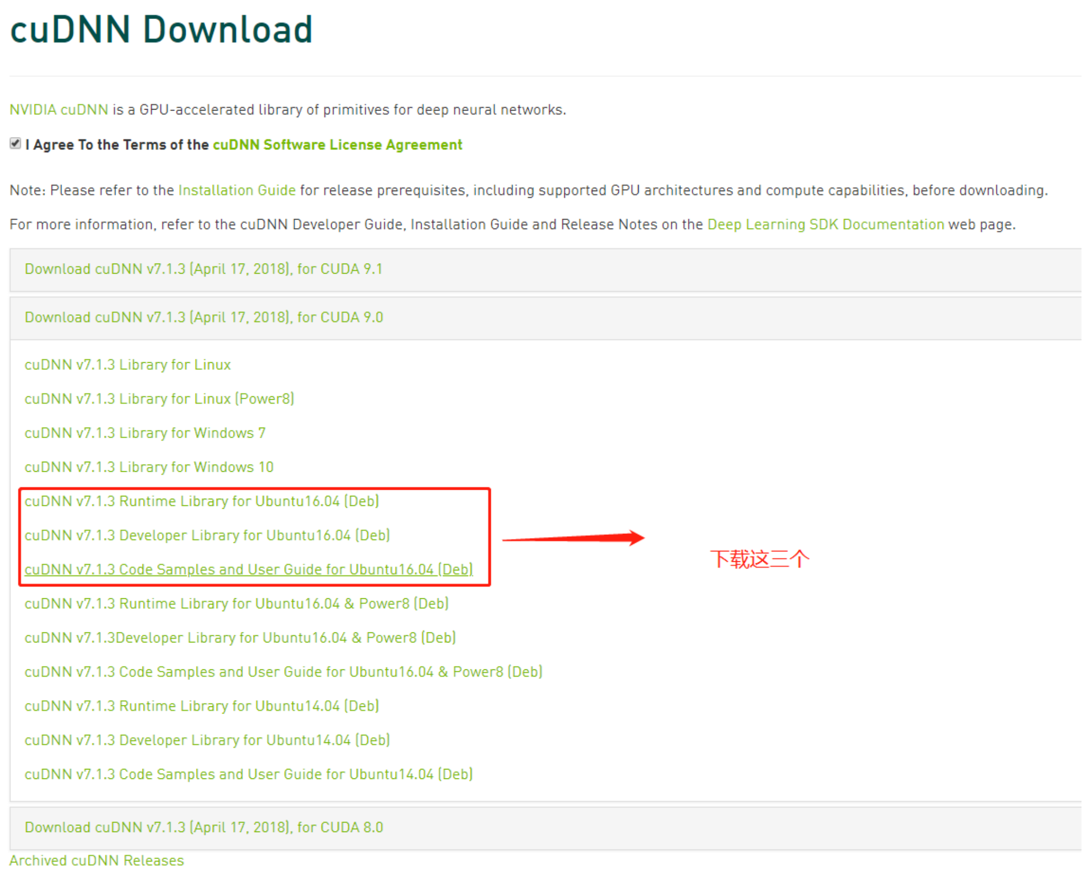
${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}



```
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export PATH=/usr/local/cuda-9.0/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-9.0/lib\
 ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
                                                        108,1         Bot
```

## TensorRT环境配置与安装

2.安装cudnn：
下载cudnn （本系统是ubuntu 16.04+cuda 9.0）：
下载地址：https://developer.nvidia.com/rdp/cudnn-download



cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

☑ I Agree To the Terms of the cuDNN Software License Agreement

Note: Please refer to the Installation Guide for release prerequisites, including supported GPU architectures and compute capabilities, before downloading.

For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the Deep Learning SDK Documentation web page.

Download cuDNN v7.1.3 (April 17, 2018), for CUDA 9.1

Download cuDNN v7.1.3 (April 17, 2018), for CUDA 9.0

cuDNN v7.1.3 Library for Linux
cuDNN v7.1.3 Library for Linux (Power8)
cuDNN v7.1.3 Library for Windows 7
cuDNN v7.1.3 Library for Windows 10
cuDNN v7.1.3 Runtime Library for Ubuntu16.04 (Deb)
cuDNN v7.1.3 Developer Library for Ubuntu16.04 (Deb)
cuDNN v7.1.3 Code Samples and User Guide for Ubuntu16.04 (Deb)
cuDNN v7.1.3 Runtime Library for Ubuntu16.04 & Power8 (Deb)
cuDNN v7.1.3Developer Library for Ubuntu16.04 & Power8 (Deb)
cuDNN v7.1.3 Code Samples and User Guide for Ubuntu16.04 & Power8 (Deb)
cuDNN v7.1.3 Runtime Library for Ubuntu14.04 (Deb)
cuDNN v7.1.3 Developer Library for Ubuntu14.04 (Deb)
cuDNN v7.1.3 Code Samples and User Guide for Ubuntu14.04 (Deb)

Download cuDNN v7.1.3 (April 17, 2018), for CUDA 8.0
Archived cuDNN Releases

## TensorRT环境配置与安装

安装并测试：

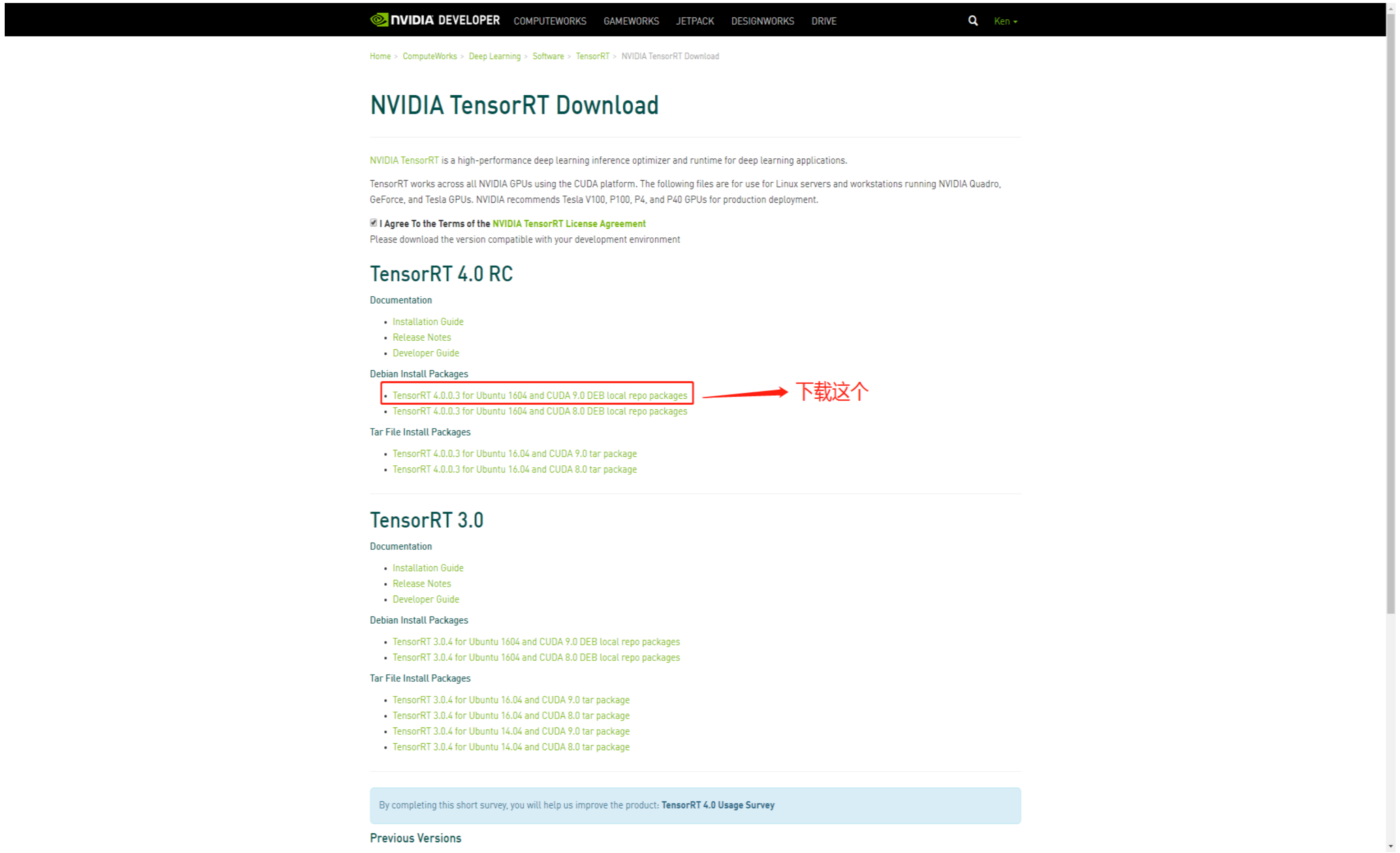
```
$ sudo dpkg -i libcudnn7_7.1.3.16-1%2Bcuda9.0_amd64.deb
$ sudo dpkg -i libcudnn7-dev_7.1.3.16-1%2Bcuda9.0_amd64.deb
$ sudo dpkg -i libcudnn7-doc_7.1.3.16-1+cuda9.0_amd64.deb
```

然后我们测试一下安装的结果：

```
$ cp -r /usr/src/cudnn_samples_v7/ $HOME
$ cd  ~/cudnn_samples_v7/mnistCUDNN
$ make
$ ./mnistCUDNN
```

## TensorRT环境配置与安装

3.TensorRT的下载与安装
首先，我们下载TensorRT：
https://developer.nvidia.com/nvidia-tensorrt-download



NVIDIA DEVELOPER COMPUTEWORKS GAMEWORKS JETPACK DESIGNWORKS DRIVE Ken

Home > ComputeWorks > Deep Learning > Software > TensorRT > 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 workstations running NVIDIA Quadro, GeForce, and Tesla GPUs. NVIDIA recommends Tesla V100, P100, P4, and P40 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 4.0 RC

Documentation

- Installation Guide
- Release Notes
- Developer Guide

Debian Install Packages

- TensorRT 4.0.0.3 for Ubuntu 1604 and CUDA 9.0 DEB local repo packages
- TensorRT 4.0.0.3 for Ubuntu 1604 and CUDA 8.0 DEB local repo packages

Tar File Install Packages

- TensorRT 4.0.0.3 for Ubuntu 16.04 and CUDA 9.0 tar package
- TensorRT 4.0.0.3 for Ubuntu 16.04 and CUDA 8.0 tar package

TensorRT 3.0

Documentation

- Installation Guide
- Release Notes
- Developer Guide

Debian Install Packages

- TensorRT 3.0.4 for Ubuntu 1604 and CUDA 9.0 DEB local repo packages
- TensorRT 3.0.4 for Ubuntu 1604 and CUDA 8.0 DEB local repo packages

Tar File Install Packages

- TensorRT 3.0.4 for Ubuntu 16.04 and CUDA 9.0 tar package
- TensorRT 3.0.4 for Ubuntu 16.04 and CUDA 8.0 tar package
- TensorRT 3.0.4 for Ubuntu 14.04 and CUDA 9.0 tar package
- TensorRT 3.0.4 for Ubuntu 14.04 and CUDA 8.0 tar package

By completing this short survey, you will help us improve the product: **TensorRT 4.0 Usage Survey**

Previous Versions

## TensorRT环境配置与安装

```
$ sudo dpkg -i nv-tensorrt-repo-ubuntu1604-cuda9.0-rc-trt4.0.0.3-20180329_1-1_amd64.deb
$ sudo apt-get update
$ sudo apt-get install tensorrt
```

如果你是python 2.7

```
$ sudo apt-get install python-libnvinfer-doc swig
```

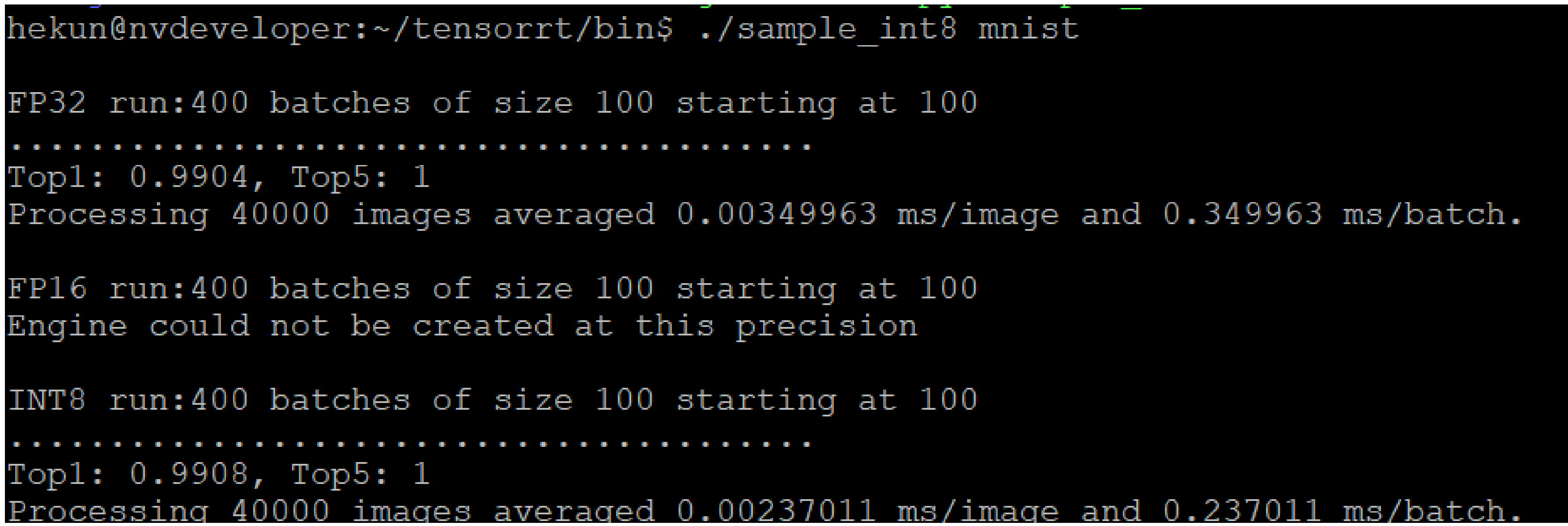
如果是3.5

```
sudo apt-get install python3-libnvinfer-doc
```

这样就安装好了，我们测试一下：

```
$ cp -r /usr/src/tensorrt/ ~/
$ cd ~/tensorrt/samples
$ make
$ cd ../bin
$ ./sample_int8 mnist
```

## TensorRT环境配置与安装



```
hekun@nvdeveloper:~/tensorrt/bin$ ./sample_int8 mnist

FP32 run:400 batches of size 100 starting at 100
........................................
Top1: 0.9904, Top5: 1
Processing 40000 images averaged 0.00349963 ms/image and 0.349963 ms/batch.

FP16 run:400 batches of size 100 starting at 100
Engine could not be created at this precision

INT8 run:400 batches of size 100 starting at 100
........................................
Top1: 0.9908, Top5: 1
Processing 40000 images averaged 0.00237011 ms/image and 0.237011 ms/batch.
```

## TensorRT编程模型

### 1. caffeToGIEModel – 将 caffe model 转换到 TensorRT 格式

```
// 1.创建builder
IBuilder* builder = createInferBuilder(gLogger);

// 2.解析caffe模型，保存到 Network
INetworkDefinition* network = builder->createNetwork();
ICaffeParser* parser = createCaffeParser();
const IBlobNameToTensor* blobNameToTensor = parser->parse(locateFile(deployFile,          directories).c_s
tr(), locateFile(modelFile, directories).c_str(),*network, DataType::kFLOAT);

// 3.指定输出Tensor
for (auto& s : outputs)
    network->markOutput(*blobNameToTensor->find(s.c_str()));

// 4.构建engine
builder->setMaxBatchSize(maxBatchSize);
builder->setMaxWorkspaceSize(1 << 20);

ICudaEngine* engine = builder->buildCudaEngine(*network);
assert(engine);

// 5.销毁parser
network->destroy();
parser->destroy();

// 6.将engine序列化到GIE，退出
gieModelStream = engine->serialize();
engine->destroy();
builder->destroy();
```

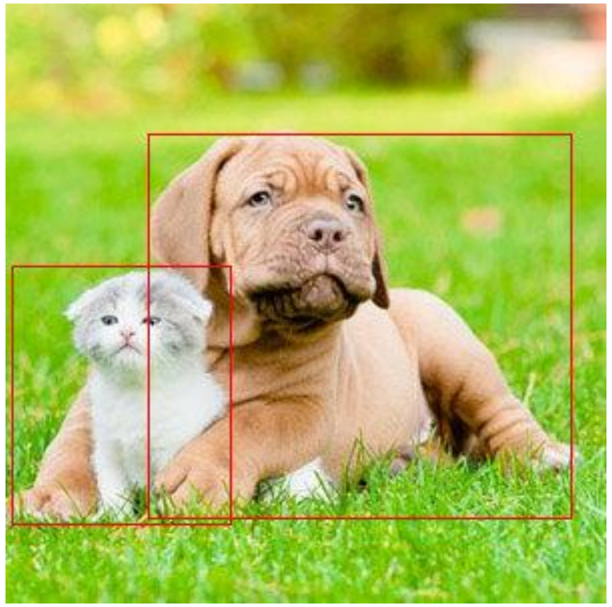
# TensorRT编程模型

## 2. 执行过程 main

```
// 1.从caffe模型创建GIE模型，序列化到流
IHostMemory *gieModelStream{nullptr};
caffeToGIEModel("mnist.prototxt", "mnist.caffemodel", std::vector < std::string > { OUTPUT_BLO
B_NAME }, 1, gieModelStream);

// x.数据获取（略）
// x.解析mean文件(略)

// 2.反序列化，得到Runtime engine
IRuntime* runtime = createInferRuntime(gLogger);
ICudaEngine* engine = runtime->deserializeCudaEngine(gieModelStream->data(), gieModelStr
eam->size(), nullptr);
if (gieModelStream) gieModelStream->destroy();

// 3.创建上下文
IExecutionContext *context = engine->createExecutionContext();

// 4.运行inference
float prob[OUTPUT_SIZE];
doInference(*context, data, prob, 1);

// 5.销毁engine
context->destroy();
engine->destroy();
runtime->destroy();
```

TensorRT编程模型

## SampleUffSSD



TensorRT编程模型

## Main Components

Preprocessor

FeatureExtractor

BoxPredictor

GridAnchorGenerator

PostProcessor

## TensoRT实例展示