VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Model Optimization and Deployment using Nvidia TensorRT

Georg Zitzlsberger georg.zitzlsberger@vsb.cz

12-11-2019

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS

Agenda

TensorRT Overview

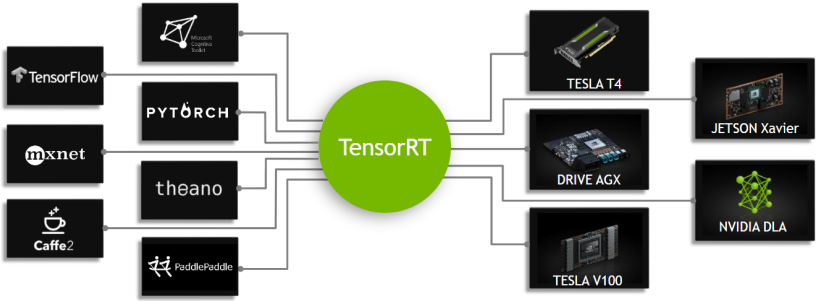
Optimizaitons and Performance

TensorRT Inference Server

TensorRT Workflow

Questions

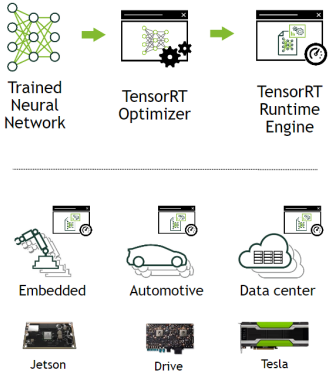
TensorRT Overview



(Image: Nvidia)

- Optimize for different target architectures
- Support all major Deep Learning frameworks and standards
- Run deep neural networks independently
  (i.e. outside the training framework)

TensorRT Overview

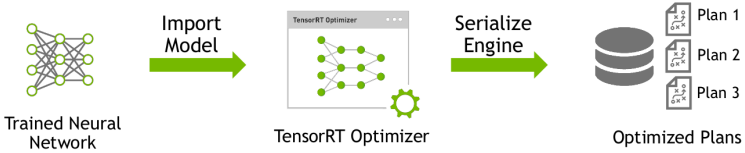


(Image: Nvidia)

- Two phases:
  1. TensorRT Optimizer: Optimize for target architecture/GPU
  2. TensorRT Runtime Engine: Execute on target GPU
- C++ and Python APIs
- Optimize execution and memory usage
- Quantize the neurons

TensorRT Overview

Step 1: Optimize trained model



Step 2: Deploy optimized plans with runtime

(Image: Nvidia)

TensorRT Overview

- Supports all important deep learning frameworks

(Image: Nvidia)

- Support of Open Neural Network Exchange (ONNX) format
- TensorFlow
  Tensorflow integration for more efficient inference
- MathWorks
  Also MATLAB is supported via GPU Coder

Optimizaitons and Performance

- **Weight & Activation Precision Calibration:**
  Maximizes throughput by quantizing models to INT8/FP8 while preserving accuracy

- **Layer & Tensor Fusion:**
  Optimizes use of GPU memory and bandwidth by fusing nodes in a kernel

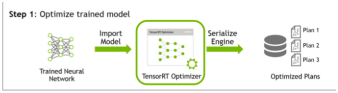
- **Kernel Auto-Tuning:**
  Selects best data layers and algorithms based on target GPU platform

Optimizaitons and Performance - cont'd

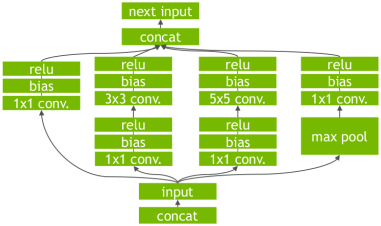
- **Dynamic Tensor Memory:**
  Minimizes memory footprint and re-uses memory for tensors efficiently

- **Multi-Stream Execution:**
  Scalable design to process multiple input streams in parallel

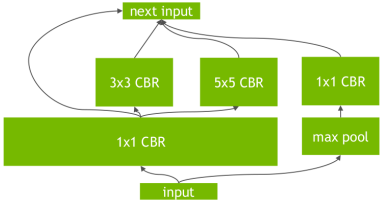
LAYER & TENSOR FUSION



Un-Optimized Network



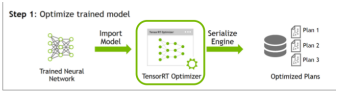
TensorRT Optimized Network



(Image: Nvidia)

Layer & Tensor Fusion

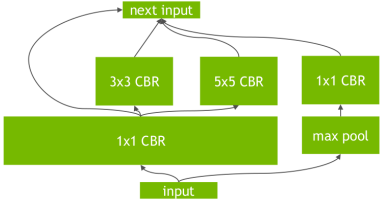
LAYER & TENSOR FUSION



- Vertical Fusion
- Horizonal Fusion
- Layer Elimination

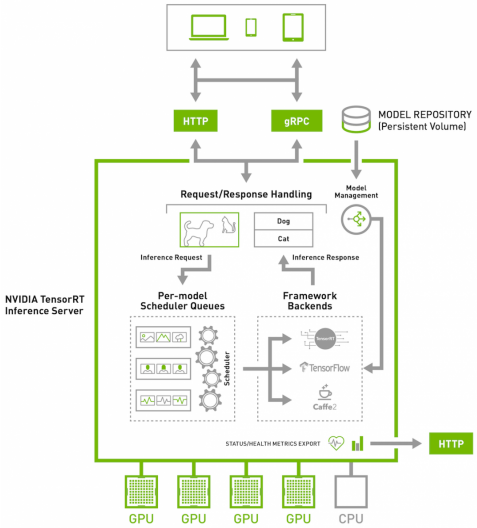
Network	Layers before	Layers after
VGG19	43	27
Inception V3	309	113
ResNet-152	670	159

TensorRT Optimized Network



(Image: Nvidia)

TensorRT Inference Server



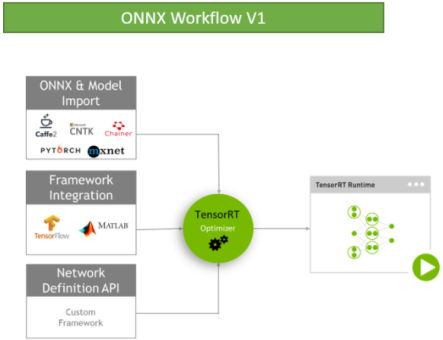
(Image: Nvidia)

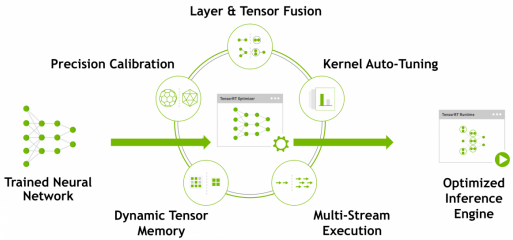
- Optimized inference as microservice
- As Docker container available on the Nvidia GPU Cloud ▸ NGC
- Or as open source on ▸ Github

TensorRT Inference Server - cont'd

- Maximizes utilization by enabling inference for multiple models on one or more GPUs
- Supports all popular AI frameworks
- Supports audio streaming inputs
- Dynamically batches requests to increase throughput
- Provides latency and health metrics for auto scaling and load balancing

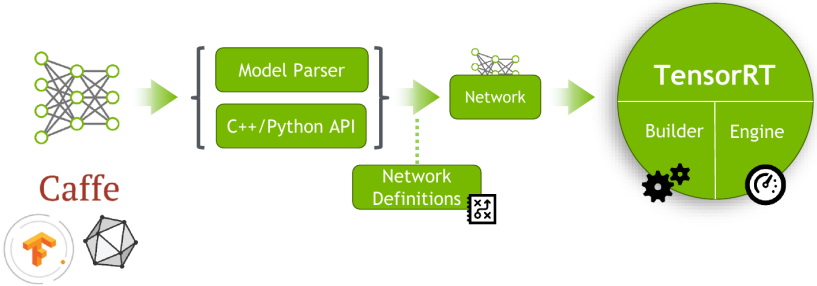
TensorRT Workflow





(Images: Nvidia)

TensorRT Workflow - cont'd



(Image: Nvidia)

- Import model (e.g. from ONNX)
- Parse model using C++/Python API
- Serialize network
- Execute in TensorRT engine

TensorRT Workflow - Example

Step 1: Convert trained model into TensorRT format

```
uff_model = uff.from_tensorflow_frozen_model("frozen_model_file.pb",
                                             OUTPUT_LAYERS)
```

Step 2: Create a model parser

```
parser = uffparser.create_uff_parser()
```

Step 3: Register inputs and outputs

```
parser.register_input(INPUT_LAYERS[0], (INPUT_C,INPUT_H,INPUT_W),0)
parser.register_output(OUTPUT_LAYERS[0])
```

Step 4: Optimize model and create a runtime engine

```
engine = trt.utils.uff_to_trt_engine(G_LOGGER,
                                     uff_model,
                                     parser,
                                     INFERENCE_BATCH_SIZE,
                                     1<<20,
                                     trt.infer.DataType.FLOAT)
```

Step 5: Serialize optimized engine

```
trt.utils.write_engine_to_file(save_path, engine.serialize())
```

Step 6: De-serialize engine

```
engine = Engine(PLAN=plan,
                postprocessors={"output_layer_name":post_processing_function})
```

Step 7: Perform inference

```
result = engine_single.infer(image)
```

(Image: Nvidia)

Questions

Q&A

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

IT4Innovations National Supercomputing Center

VŠB – Technical University of Ostrava
Studentská 6231/1B
708 00 Ostrava-Poruba, Czech Republic
www.it4i.cz

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS