Network Deployment

Steve Byun
Sr. Solution Architect
NVIDIA Corporation
Part 1: Inference using DIGITS
PART 1: INFERENC USING DIGITS
Neural network training and inference
CCTV 서비스

서비스 서버

딥러닝 학습 서버

Camera
All video
3. Run video analytics on real-time feeds
4. Control something (if needed)

Operator console/video wall
Thin client

HPE Edgeline EL4000
NVIDIA Tesla P4
4. Selectively upload video or attributes

HPE switches/access points

Only video of interest
5. Display events to operators

The edge

HPE storage
NVIDIA Tesla P100
Training data
1. Train model
6. Archive for post analysis
개발환경 설정

가상환경
서버 설정
Prepare System (H/W, OS)
Install NVIDIA driver
Install NVDOCKER

CUDA 개발 프로그래밍
docker pull nvidia/cuda
docker run -it nvidia/cuda:8.5

딥러닝 학습 환경

```bash
# nvidia/digits
docker pull nvidia/digits
docker pull tensorflow/tensorflow:latest-gpu

docker run 익션 nvidia/digits
```
PART 1: INFERENCE USING DIGITS

DIGITS: Web based interface of Caffe and Torch

Open Your Lab Page
Part 2: Inference using pycaffe
PART 2: INFERENCES USING PYCAFFE

Pycaffe APIs

- `caffe.Net` is the central interface for loading, configuring, and running models. `caffe.Classsifier` and `caffe.Detector` provide convenience interfaces for common tasks.

- `caffe.SGDSolver` exposes the solving interface.

- `caffe.io` handles input / output with preprocessing and protocol buffers.

- `caffe.draw` visualizes network architectures.

- Caffe blobs are exposed as numpy ndarrays for ease-of-use and efficiency.
PART 2: INFEERENCE USING PYCAFFE

Pycaffe APIs

Open Your Lab Page
PART 2: INFEERENCE USING PYCAFFE

caffe.io.Tranformer: Tips

- **set_transpose & set_channel_swap**: set_transpose is defined for changing the dimensions of the input image. set_transpose of an input of the size \((227, 227, 3)\) with parameters \((2, 0, 1)\) will be \((3, 227, 227)\). Applying set_channel_swap will preserve the order \((3, 227, 227)\) but change it for example, from RGB to BGR.

- **set_raw_scale**: Set the scale of raw features s.t. the input blob = input * scale. While Python represents images in \([0, 1]\), certain Caffe models like CaffeNet and AlexNet represent images in \([0, 255]\) so the raw_scale of these models must be 255.
Part 3: NVIDIA TensorRT
TENSORRT
Maximum Performance for Deep Learning Inference

- High-performance framework makes it easy to develop GPU-accelerated inference
  - Production deployment solution for deep learning inference
  - Optimized inference for a given trained neural network and target GPU
  - Solutions for Hyperscale, ADAS, Embedded
  - Supports deployment of 32-bit or 16-bit inference

developer.nvidia.com/gpu-inference-engine
TENSORRT
Maximum Performance for Deep Learning Inference

- High-performance framework makes it easy to develop GPU-accelerated inference
  - Production deployment solution for deep learning inference
  - Optimized inference for a given trained neural network and target GPU
  - Solutions for Hyperscale, ADAS, Embedded
  - Supports deployment of 32-bit or 16-bit inference

developer.nvidia.com/gpu-inference-engine
TENSORRT Optimizations

- Fuse network layers
- Eliminate concatenation layers
- Kernel specialization
- Auto-tuning for target platform
- Select optimal tensor layout
- Batch size tuning

TRAINED NEURAL NETWORK

OPTIMIZED INFERENCE RUNTIME

developer.nvidia.com/gpu-inference-engine
# TENSORRT Performance

<table>
<thead>
<tr>
<th>BATCH SIZE</th>
<th>PERFORMANCE</th>
<th>POWER EFFICIENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tesla M4</td>
<td>128</td>
<td>1153 images/s</td>
</tr>
<tr>
<td>Jetson TX1</td>
<td>2</td>
<td>133 images/s</td>
</tr>
</tbody>
</table>

developer.nvidia.com/gpu-inference-engine
PART 3: NVIDIA TENSORRT

Two Phases

- **Build**: optimizations on the network configuration and generates an optimized plan for computing the forward pass.

- **Deployment**: Forward and output the inference result.
PART 3: NVIDIA TENSORRT

Supported layers

- Convolution: 2D
- Activation: ReLU, tanh and sigmoid
- Pooling: max and average
- ElementWise: sum, product or max of two tensors
- LRN: cross-channel only
- Fully-connected: with or without bias
- SoftMax: cross-channel only
- Deconvolution

Scalability: Output/Input Layers can connect with other deep learning framework (e.g. caffe) directly
PART 3: NVIDIA TENSORRT

Open Your Lab Page
PART 3: NVIDIA TENSORRT

Optimizations

- Layers with unused output are eliminated to avoid unnecessary computation
- **Vertical layer fusion:** Convolution, bias, and ReLU layers are fused to form a single layer
- **Horizontal layer fusion:** combining layers that take the same source tensor and apply the same operations with similar parameters
PART 3: NVIDIA TENSORRT

Optimizations: Original Network
PART 3: NVIDIA TENSORRT
Optimizations: Vertical Layer Fusion
PART 3: NVIDIA TENSORRT

Optimizations: Horizontal layer fusion
PART 3: NVIDIA TENSORRT

Open Your Lab Page
WHAT’S NEXT

TAKE SURVEY
...for the chance to win an NVIDIA SHIELD TV.
Check your email for a link.

ACCESS ONLINE LABS
Check your email to access more DLI training online.

ATTEND WORKSHOP
Visit www.nvidia.com/dli for workshops in your area.

JOIN DEVELOPER PROGRAM