DEEP NEURAL NETWORKS
CHANGING THE AUTONOMOUS VEHICLE LANDSCAPE

Dennis Lui | August 2017
THE RISE OF GPU COMPUTING

GPU-Computing perf
1.5X per year

Single-threaded perf
1.5X per year

GPU-Computing perf
1.1X per year

AI BREAKTHROUGHS

Recent Breakthroughs

- **2015**
  - Atari Games

- **2016**
  - “Superhuman” Image Recognition
  - AlphaGo Rivals World Champion

- **2017**
  - Conversational Speech Recognition
  - Lip Reading
AI IMPROVING AT AMAZING RATES

**IMAGENET ACCURACY**

- GPU Deep Learning: 96%
- Hand-coded CV: 74%

**SPEECH RECOGNITION ACCURACY**

- Microsoft 9/13/16
- Conversational Speech
- Switchboard
AI IS THE SOLUTION TO SELF DRIVING

- Perception
- Reasoning
- Driving
- HD Map
- Mapping
- AI Computing
DEEP LEARNING FOR AUTONOMOUS DRIVING
DEEP NEURAL NETWORK
## MULTICLASS OBJECT DETECTION & CLASSIFICATION NETWORK

<table>
<thead>
<tr>
<th>Description</th>
<th>Demonstrates NVIDIA’s proprietary deep neural network (DNN) to perform object detection</th>
</tr>
</thead>
</table>

### Types Detected/Color Code
- Red: Cars
- Cyan: Trucks
- Green: Traffic Signs (Detection Only)
- Blue: Bicycles
- Yellow: Pedestrians
### LANE DETECTION NETWORK

| Description | Demonstrates NVIDIA’s proprietary deep neural network (DNN) to perform lane detection on the road
|             | Detects ego-lane by showing the boundaries of the left and right lane, and in some cases, is able to show the left and right boundaries of adjacent lanes as well |
| Color Code | Red: Ego-lane left  
|             | Green: Ego-lane right  
|             | Yellow: Left adjacent lane  
|             | Blue: Right adjacent lane |
# FREE SPACE DETECTION NETWORK

<table>
<thead>
<tr>
<th>Description</th>
<th>Demonstrates NVIDIA’s proprietary deep neural network (DNN) to detect free space in front of the vehicle.</th>
</tr>
</thead>
</table>
| Color Code        | Red: cars  
                      Green: Curbs  
                      Blue: Pedestrians  
                      Yellow: Others |

![Image of vehicle detection system](image-url)
END-TO-END AUTONOMOUS DRIVING NETWORK

Training data

Sensor data

Human actuator commands

CNN

CNN actuator commands

Error (training) signal
END-TO-END AUTONOMOUS DRIVING NETWORK
GPU DEEP LEARNING COMPUTING MODEL
A COMPLETE DEEP LEARNING PLATFORM

MANAGE

TRAIN

DEPLOY

DIGITS

TensorRT

A COMPLETE DEEP LEARNING PLATFORM

MANAGE / AUGMENT

TEST

TRAIN

PROTOTXT

EMBEDDED

DATA CENTER

AUTOMOTIVE

DIGITS

Caffe

torch
DEEP LEARNING PLATFORMS
From Training to Development and Production

**TRAINING**

Nvidia DGX-1 with Tesla V100 (DGX-1V)

**DEPLOY**

DRIVE PX 2
- 2 PARKER + 2 PASCAL GPU
  - 20 TOPS DL
  - 120 SPECINT
  - 80W

XAVIER
- 30 TOPS DL
- 160 SPECINT
- 30W
CUDA TensorOp instructions & data formats

4x4 matrix processing array

$$D_{[FP32]} = A_{[FP16]} \times B_{[FP16]} + C_{[FP32]}$$

Optimized for deep learning
DATASET CREATION
DATA ACQUISITION

- GMSL 12x Camera
- Gigabit Ethernet
  - Lidar / Radar / GPS
- CAN
  - Vehicle Information
  - Other Sensors
- DRIVE™ PX 2
  - 10 Gigabit Ethernet / USB 3
  - Network Drive / SSD
DATASET CREATION

DATA CURATION
Filter and keep data of interest

DATA ANNOTATION
Bounding boxes, per pixel labeling

START FROM TRAINED NETWORK
May reduce required data size
TRAINING DEEP NEURAL NETWORKS
NVIDIA DIGITS
Interactive Deep Learning GPU Training System

MANAGE DATA

CONFIGURE NEURAL NET

MONITOR TRAINING

VISUALIZE RESULTS
NVIDIA DIGITS
Monitor Training

WELL BEHAVED

OVERFIT

ILL BEHAVED

MANY OUTPUTS
DNN TRAINING
Iterate and Innovate Faster

DGX-1 with Tesla V100
- 7.4 hours

8-way GPU Server
- 18 hours

Dual Socket CPU
- 711 hours

1X 40X 96X

Workload: ResNet50, 90 epochs to solution | CPU Server: Dual Xeon E5-2699 v4, 2.6GHz
DNN INFERENCE OPTIMIZATIONS
DNN INFEERENCE OPTIMIZATIONS

**HARDWARE ACCELERATIONS**

Specialized instructions for deep learning operations

**PRUNING**

Prune down the network size (neurons + connections) to reduce inference time

**TensorRT**

Accelerated neural network inference engine
HARDWARE ACCELERATIONS
Specialized Instruction for Deep Learning Operations

DP4A

\[
\begin{align*}
A & \times \times \times \times \\
B & \times \times \times \times \\
\text{int32} & + \\
C & \text{int32} = \\
D & \text{int32}
\end{align*}
\]

INT8 dot product
PRUNING

Train Connectivity

Prune Connections

Train Weights

Before Pruning

After Pruning

Pruning Synapses

Pruning Neurons
## PRUNING

<table>
<thead>
<tr>
<th>NETWORK</th>
<th>TOP-1 ERROR</th>
<th>TOP-5 ERROR</th>
<th>PARAMETERS</th>
<th>COMPRESSION RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet-300-100 Ref</td>
<td>1.64%</td>
<td>-</td>
<td>262K</td>
<td>12x</td>
</tr>
<tr>
<td>LeNet-300-100 Pruned</td>
<td>1.59%</td>
<td>-</td>
<td>22K</td>
<td></td>
</tr>
<tr>
<td>LeNet-5 Ref</td>
<td>0.80%</td>
<td>-</td>
<td>431K</td>
<td>12x</td>
</tr>
<tr>
<td>LeNet-5 Pruned</td>
<td>0.77%</td>
<td>-</td>
<td>36K</td>
<td></td>
</tr>
<tr>
<td>AlexNet Ref</td>
<td>42.78%</td>
<td>19.73%</td>
<td>61M</td>
<td>9x</td>
</tr>
<tr>
<td>AlexNet Pruned</td>
<td>42.77%</td>
<td>19.67%</td>
<td>6.7M</td>
<td></td>
</tr>
<tr>
<td>VGG-16 Ref</td>
<td>31.50%</td>
<td>11.32%</td>
<td>138M</td>
<td>13x</td>
</tr>
<tr>
<td>VGG-16 Pruned</td>
<td>31.34%</td>
<td>10.88%</td>
<td>10.3M</td>
<td></td>
</tr>
</tbody>
</table>

TensorRT

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 fp32, fp16, int8* inference

* int8 support will be available from v2
TensorRT
Optimizations

- Fuse network layers
- Eliminate concatenation layers
- Kernel specialization
- Auto-tuning for target platform
- Tuned for given batch size

TRAINED NEURAL NETWORK

OPTIMIZED INFERENCERUNTIME
TensorRT – iGPU (FP16)

GoogleNet, AlexNet, VGG19

<table>
<thead>
<tr>
<th>Batch size</th>
<th>GoogleNet</th>
<th>AlexNet</th>
<th>VGG19</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>94</td>
<td>112</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>189</td>
<td>232</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>226</td>
<td>353</td>
<td>39</td>
</tr>
<tr>
<td>8</td>
<td>258</td>
<td>366</td>
<td>39</td>
</tr>
<tr>
<td>16</td>
<td>272</td>
<td>450</td>
<td>40</td>
</tr>
</tbody>
</table>

Images / second

Batch size

Configuration:
HW (DRIVE PX 2 iGPU@1275 MHz), SW (PDK ALPHA 2.0, TensorRT 1.0RC), GoogleNet & VGG19 Input Image Resolution (224x224)
AlexNet Input Image Resolution (227x227)
TensorRT — dGPU (INT8)

GoogleNet, AlexNet, VGG19

Images / second

GoogleNet, AlexNet, VGG19

Configuration:
HW (DRIVE PX 2 dGPU@1290 MHz), SW (PDK ALPHA 2.0, TensorRT 2.0EA), GoogleNet & VGG19 Input Image Resolution (224x224) 
AlexNet Input Image Resolution (227x227)
TensorRT

INT8 Workflow

- Calibration Dataset
- Validation Dataset
- Calibration Parameters
- Scoring Function

FP32 Training Framework

FP32 NEURAL NETWORK

INT8 OPTIMIZATION USING TensorRT

INT8 PLAN

INT8 RUNTIME USING TensorRT
## TensorRT

### 8-bit Inference: Top-1 Accuracy

<table>
<thead>
<tr>
<th>NETWORK</th>
<th>FP32 TOP1</th>
<th>INT8 TOP1</th>
<th>DIFFERENCE</th>
<th>PERF GAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>57.22%</td>
<td>56.96%</td>
<td>0.26%</td>
<td>3.70x</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>68.87%</td>
<td>68.49%</td>
<td>0.38%</td>
<td>3.01x</td>
</tr>
<tr>
<td>VGG</td>
<td>68.56%</td>
<td>68.45%</td>
<td>0.11%</td>
<td>3.23x</td>
</tr>
<tr>
<td>Resnet-152</td>
<td>75.18%</td>
<td>74.56%</td>
<td>0.61%</td>
<td>3.42%</td>
</tr>
</tbody>
</table>
TensorRT — GRAPH OPTIMIZATION

Unoptimized Network
TensorRT — GRAPH OPTIMIZATION

Vertical Fusion

Diagram showing a network with operations such as `concat`, `1x1 CBR`, `3x3 CBR`, `5x5 CBR`, `1x1 CBR`, `max pool`, and `input`.
TensorRT — GRAPH OPTIMIZATION

Horizontal Fusion

next input
concat

3x3 CBR
5x5 CBR
1x1 CBR

1x1 CBR
max pool
input
concat
TensorRT — GRAPH OPTIMIZATION

Concat Elision

next input

3x3 CBR  5x5 CBR  1x1 CBR

1x1 CBR

max pool

input
AUTONOMOUS DRIVING CHALLENGES
AUTONOMOUS DRIVING

Challenges
AUTONOMOUS DRIVING

Challenges
PUTTING IT ALL TOGETHER
<table>
<thead>
<tr>
<th>RESOURCES</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA DRIVE Platform</td>
<td><a href="https://developer.nvidia.com/drive">https://developer.nvidia.com/drive</a></td>
</tr>
<tr>
<td>TensorRT</td>
<td><a href="https://developer.nvidia.com/tensorrt">https://developer.nvidia.com/tensorrt</a></td>
</tr>
</tbody>
</table>
QUESTIONS?