MATLAB EXPO 2018

Deploying Deep Learning Networks to Embedded GPUs and CPUs

Pierre Nowodzienski
Deep Learning enablers

Increased GPU acceleration

Labeled public datasets

World-class models to be leveraged

- AlexNet: PRETRAINED MODEL
- VGG-16: PRETRAINED MODEL
- ResNet: PRETRAINED MODEL

Caffe
GoogLeNet
TensorFlow/Keras

Human Accuracy

Error (%)
Deep Learning Applications:
Image classification, speech recognition, autonomous driving, etc…

Detection of cars and road in autonomous driving systems

Rain Detection and Removal

1. Deep Joint Rain Detection and Removal from a Single Image

Traffic Sign Recognition
GPUs and CUDA programming

- CUDA
- OpenCL
- C/C++
- MATLAB
- Python

Ease of programming (expressivity, algorithmic, …)

faster
easier

GPUs are “hardware on steroids”, ...

but, programming them is hard
Deep learning workflow in MATLAB

- **Deep Neural Network Design + Training**
  - **Design in MATLAB**
    - Manage large data sets
    - Automate data labeling
    - Easy access to models
  - **Training in MATLAB**
    - Acceleration with GPU’s
    - Scale to clusters
Deep learning workflow in MATLAB

Deep Neural Network Design + Training

- Idea
- Caffe
- TensorFlow
- Keras
- Model importer
- Train in MATLAB
- Trained DNN

Application design

- Application logic
- Localization
- Planning
- Autonomous systems
- Deep learning Perception
- Controls
Deep learning workflow in MATLAB

Deep Neural Network Design + Training

Deep Neural Network Design + Training

Standalone Deployment

Application design

Application logic

Train in MATLAB

Trained DNN

Model importer

Caffe

Keras

TensorFlow

Model importer

GPU Coder

C++/CUDA
GPU Coder for Deployment

Accelerated implementation of parallel algorithms on GPUs & CPUs

Deep Neural Networks
Deep Learning, machine learning

Image Processing and Computer Vision
Image filtering, feature detection/extraction

Signal Processing and Communications
FFT, filtering, cross correlation,

**5x faster** than TensorFlow
**2x faster** than MXNet

**60x faster** than CPUs for stereo disparity

**20x faster** than CPUs for FFTs

Intel MKL-DNN Library
NVIDIA CUDA C/C++
ARM Compute Library

GPU Coder

Accelerated implementation of parallel algorithms on GPUs & CPUs
GPUs and CUDA

CUDA kernels

C/C++

GPU CUDA Cores

GPU Memory Space

ARM Cortex

CPU Memory Space

SECURITY ENGINES

4K60 VIDEO ENCODER

60 VIDEO DECODER

AUDIO ENGINE

2D ENGINE

DISPLAY ENGINES

128-bit LPDDR4

BOOT and PM PROC

GigE Ethernet MAC

IMAGE PROC (ISP)

Safety Engine

I/O
Challenges of Programming in CUDA for GPUs

- Learning to program in CUDA
  - Need to rewrite algorithms for parallel processing paradigm

- Creating CUDA kernels
  - Need to analyze algorithms to create CUDA kernels that maximize parallel processing

- Allocating memory
  - Need to deal with memory allocation on both CPU and GPU memory spaces

- Minimizing data transfers
  - Need to minimize while ensuring required data transfers are done at the appropriate parts of your algorithm
GPU Coder Helps You Deploy to GPUs Faster

- Library function mapping
- Loop optimizations
- Dependence analysis

- Data locality analysis
- GPU memory allocation

- Data-dependence analysis
- Dynamic memcpy reduction

GPU Coder

CUDA Kernel creation

Memory allocation

Data transfer minimization

NVIDIA CUDA C/C++
GPU Coder speeds up MATLAB for Image Processing and Computer Vision

- Fog removal: 5x speedup
- Distance transform: 8x speedup
- Ray tracing: 18x speedup
- Frangi filter: 3x speedup
- Stereo disparity: 50x speedup
- SURF feature extraction: 700x speedup
GPU Coder speeds up MATLAB at least 2x for inference

MATLAB 18a on TitanXP GPU - Linux

Images / Sec

- AlexNet: 700 Images / Sec
- ResNet-50: 150 Images / Sec
- VGG-16: 200 Images / Sec

Single image prediction using Intel® Xeon® CPU - 3.6 GHz, NVIDIA libraries: CUDA8 - cuDNN 7, TensorFlow 1.6.0, MXNet 1.1.0, MATLAB 18a
With GPU Coder, MATLAB is faster than other frameworks

Single Image Prediction (TitanXP GPU, Linux)

Images / Sec

- AlexNet
- ResNet-50
- VGG-16

TensorFlow
MXNet
GPU Coder

Single image prediction using Intel® Xeon® CPU - 3.6 GHz, NVIDIA libraries: CUDA8 - cuDNN 7, TensorFlow 1.6.0, MXNet 1.1.0, MATLAB 18a
Embedded GPU Benchmarking: Jetson TX2

Images / sec

- GPU Coder (TensorRT v3.0.4)
- GPU Coder (cuDNN v7)
- Caffe

Memory Usage (MB)

- GPU Coder (TensorRT v3.0.4)
- GPU Coder (cuDNN v7)
- Caffe
Algorithm Design to Embedded Deployment Workflow

MATLAB algorithm (functional reference)

Build type

Call CUDA from MATLAB directly

Call CUDA from (C++) hand-coded main()

Call CUDA from (C++) hand-coded main().

Cross-compiled .lib

Embeded .lib

Desktop GPU

C++

Real-time test

1. Functional test
2. Deployment unit-test
3. Deployment integration-test
Demo: Alexnet Deployment with ‘mex’ Code Generation
Algorithm Design to Embedded Deployment on Tegra GPU

MATLAB algorithm (functional reference) → GPU Coder → Build type

1. Functional test (Test in MATLAB on host)
2. Deployment unit-test (Test generated code in MATLAB on host + GPU)
3. Deployment integration-test (Test generated code within C/C++ app on host + GPU)
4. Real-time test (Test generated code within C/C++ app on Tegra target)

Build type:
- Call CUDA from MATLAB directly
- Call CUDA from (C++) hand-coded main()

Cross-compiled .lib

Call CUDA from (C++) hand-coded main(). Cross-compiled on host with Linaro toolchain
Alexnet Deployment to Tegra: Cross-Compiled with ‘lib’

Two small changes

1. Change build-type to ‘lib’

2. Select cross-compile toolchain
End-to-End Application: Lane Detection

Alexnet

Transfer Learning

Output of CNN is lane parabola coefficients according to: \( y = ax^2 + bx + c \)

Lane detection CNN

Left lane coefficients

Right lane coefficients

Post-processing (find left/right lane points)

Image with marked lanes

GPU coder generates code for whole application
Deep learning workflow in MATLAB

Deep Neural Network Design + Training

- Caffe
- IDEA
- TensorFlow

Train in MATLAB

Model importer

Trained DNN

Application design

GPU Coder

Standalone Deployment

- Intel MKL-DNN Library
- NVIDIA TensorRT cuDNN Libraries
- ARM Compute Library

Libraries

- NVIDIA cuDNN
- Intel MKL
- ARM Compute
- NVIDIA TensorRT