MATLAB EXPO 2017
Deep Learning: Transforming Engineering and Science
DEEP LEARNING: TRANSFORMING ENGINEERING AND SCIENCE
THE RISE OF GPU COMPUTING

40 Years of Microprocessor Trend Data

GPU-Computing perf
1.5X per year

Single-threaded perf
1.5X per year

1.1X per year

1000X by 2025

The Big Bang of Deep Learning
NVIDIA IS THE WORLD’S LEADING AI PLATFORM

ONE ARCHITECTURE — CUDA
AMAZING ACHIEVEMENTS IN AI

- NVIDIA
  Interactive Ray Tracing
- NVIDIA / Remedy
  Audio-driven Facial Animation
- WRNCH
  Pose Estimation
- University of Edinburgh
  Character Animation
- UC Berkeley / OpenAI
  One-shot Imitation Learning
A WORLD OF INTELLIGENT MACHINES

10% of Manufacturing Tasks Are Automated
1M Pizzas Delivered Per Day by Domino’s
100M People 80+ Years Old
Ag Tech: 70% Increase in Farm Yields by 2050
600K Bridges to Inspect in the U.S.
300M Operations per Year WW
JETSON TX2
SUPERCOMPUTER FOR AI AT THE EDGE

2 Core i7 PCs in <10W
256 CUDA cores
>1 TFLOPS
<table>
<thead>
<tr>
<th></th>
<th>JETSON TX1</th>
<th>JETSON TX2</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU</td>
<td>Maxwell</td>
<td>Pascal</td>
</tr>
<tr>
<td>CPU</td>
<td>64-bit A57 CPUs</td>
<td>64-bit Denver 2 and A57 CPUs</td>
</tr>
<tr>
<td>Memory</td>
<td>4 GB 64 bit LPDDR4 25.6 GB/s</td>
<td>8 GB 128 bit LPDDR4 58.4 GB/s</td>
</tr>
<tr>
<td>Storage</td>
<td>16 GB eMMC</td>
<td>32 GB eMMC</td>
</tr>
<tr>
<td>Wi-Fi/BT</td>
<td>802.11 2x2 ac/BT Ready</td>
<td>802.11 2x2 ac/BT Ready</td>
</tr>
<tr>
<td>Video Encode</td>
<td>2160p @ 30</td>
<td>2160p @ 60</td>
</tr>
<tr>
<td>Video Decode</td>
<td>2160p @ 60</td>
<td>2160p @ 60</td>
</tr>
<tr>
<td>Camera</td>
<td>1.4Gpix/s</td>
<td>1.4Gpix/s</td>
</tr>
<tr>
<td></td>
<td>Up to 1.5Gbps per lane</td>
<td>Up to 2.5Gbps per lane</td>
</tr>
<tr>
<td>Mechanical</td>
<td>50mm x 87mm</td>
<td>400-pin Compatible Board to Board Connector</td>
</tr>
</tbody>
</table>
INDUSTRY ADOPTION

- Manufacturing
- Agriculture
- Construction
- Inventory Management
- Logistics/Retail
- Security
- Delivery
- Inspection
- Autonomous UAV
- Social
JETPACK SDK FOR AI @ THE EDGE

Sample Code

(nsight developer tools)

Multimedia API

(deep learning)

TensorRT

cuDNN

(VisionWorks)

OpenCV

(Vulkan)

OpenGL

(libargus)

Video API

Computer Vision

Graphics

Media

CUDA, Linux4Tegra, ROS

Jetson Embedded Supercomputer: Advanced GPU, 64-bit CPU, Video CODEC, VIC, ISP
How do we target the Jetson TX2 from MATLAB?
Introducing **GPU Coder**

- Generates **CUDA** code, which can be used only on NVIDIA GPUs*

- CUDA extends C/C++ code with constructs for parallel computing

* Any modern CUDA-enabled GPU with **compute capability 3.2** or higher
Why Use GPU Coder?

**Neural Networks**
Deep Learning, machine learning

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

**Signal Processing and Communications**
FFT, filtering, cross correlation,

**Performance**

- **Up to 7x faster**
  than state-of-art

- **Up to 700x faster**
  than CPUs for feature extraction

- **Up to 20x faster**
  than CPUs for FFTs
How fast is GPU Coder?

Fog removal
5x speedup

Distance transform
8x speedup

Orders magnitude speedup over optimized C code.

Stereo disparity
50x speedup

Ray tracing
18x speedup

SURF feature extraction
700x speedup
How to Use GPU Coder? Workflow to Embedded Jetson GPU

MATLAB algorithm (functional reference)

GPU Coder

Build type

Call CUDA from MATLAB directly

.mex

Call CUDA from (C++) hand-coded main() or via SIL

.lib

Cross-compiled .lib

Tesla GPU

Tesla GPU

Tegra GPU

Functional test

Deployment unit-test

Deployment integration-test

Real-time test

(Test in MATLAB on host)

(Test generated code in MATLAB on host + GPU)

(Test generated code within C/C++ app on host + GPU)

(Test generated code within C/C++ app on Jetson target)
Demo: Generate CUDA Code for AlexNet Prediction
“Hello World” for Deep Learning

```matlab
while true
    % Grab a frame from the camera
    ipicture = camera.snapshot;

    % Resize to Alexnet size
    picture = imresize(ipicture,[227,227]);

    % Call MEX function for alexnet prediction
    tic;
    pout = alexnet_predict(single(picture));
    newt = toc;

    % Compute Frames per second (fps)
    fps = .9*fps + .1*(1/newt);

    % top 5 scores
```
Deployment to NVIDIA Jetson: Cross-Compiled ‘lib’

Two small changes
1. Change build-type to ‘lib’
2. Select cross-compile toolchain
Alexnet Inference on NVIDIA Titan XP

<table>
<thead>
<tr>
<th>Testing platform</th>
<th>CPU</th>
<th>GPU</th>
<th>cuDNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50GHz</td>
<td>Pascal Titan Xp</td>
<td>v5</td>
</tr>
</tbody>
</table>

- MATLAB (R2017b)
- TensorFlow (1.2.0)
- Caffe2 (0.8.1)
- mxNet (0.10)
- MATLAB GPU Coder (R2017b)
Alexnet Inference on **Jetson TX2**: Frame-Rate Performance

![Graph showing frame-rate performance for different batch sizes for MATLAB GPU Coder and C++ Caffe](image)

- **MATLAB GPU Coder** (R2017b)
- **C++ Caffe** (1.0.0-rc5)

Frames per second vs. Batch Size
Why is GPU Coder Faster than OSS Deep Learning Frameworks?

- OSS frameworks are designed to do many things, including:
  - Training
  - Inference
  - Support various data types (singles, FP16, int8, etc)

- Tensorflow has the Python overhead

- GPU Coder generates code for the **specific** DNN with **specific** data types
  - Much less overhead
Additional Features: Optimizations for CUDA Code

- NVIDIA accelerated library support:
  - cuSolver: Dense and sparse direct solvers to accelerate computer vision and linear optimization applications
  - cuFFT: High-performance computation of FFTs
  - cuBLAS: GPU-accelerated implementation of the standard BLAS
  - cuDNN: GPU-accelerated library of primitives for deep neural networks
Lots of Examples to Get Started
▪ Easily target Jetson TX 2 from MATLAB

▪ Best in class performance for deep learning

Come See the Demo Live!
Sign Up for 50% Discount on Jetson TX2