Perception Systems for Autonomous Vehicles using Energy-Efficient Deep Neural Networks

Forrest Iandola, Ben Landen, Kyle Bertin, Kurt Keutzer
and the DeepScale Team
IMPLEMENTING AUTONOMOUS DRIVING

THE FLOW

IMPLEMENTING AUTONOMOUS DRIVING

SENSORS

ULTRASONIC

RADAR

LIDAR

OFFLINE MAPS

REAL-TIME PERCEPTION

PATH PLANNING & ACTUATION
What does a car need to see?
What does a car need to see?

Object Detection
What does a car need to see?

Distance

Note: above visuals are an artist's rendering created to help convey concepts. They should not be judged for accuracy.
What does a car need to see?

Object Tracking
What does a car need to see?

Free Space & Driveable Area
What does a car need to see?

Lane Recognition
Today's autonomous cars require a lot of computing hardware!

...and perception is the most computationally-intensive part of the software stack

Audi
https://www.slashgear.com/man-vs-machine-my-rematch-against-audis-new-self-driving-rs-7-21415540/

BMW + Intel

Ford
http://cwc.ucsd.edu/content/connected-cars-long-road-autonomous-vehicles
Big computers =
expensive cars
As a workaround, companies want people to share autonomous vehicles to amortize hardware costs.
As a workaround, companies want people to share autonomous vehicles to amortize hardware costs.
Shared autonomous vehicles will likely have some of the downsides as public transportation.
Will better computer chips make autonomous cars affordable?
Will better computer chips make autonomous cars affordable?
Deep Learning Processors have arrived!

**THE SERVER SIDE**

<table>
<thead>
<tr>
<th>Platform</th>
<th>Computation (GFLOPS/s)</th>
<th>Memory Bandwidth (GB/s)</th>
<th>Computation-to-bandwidth ratio</th>
<th>Power (TDP Watts)</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA K20 [1]</td>
<td><strong>3500</strong> (32-bit float)</td>
<td><strong>208</strong> (GDDR5)</td>
<td><strong>17</strong></td>
<td><strong>225</strong></td>
<td>2012</td>
</tr>
<tr>
<td>NVIDIA V100 [2]</td>
<td><strong>112000</strong> (16-bit float)</td>
<td><strong>900</strong> (HBM2)</td>
<td><strong>124</strong> (yikes!)</td>
<td><strong>250</strong></td>
<td>2018</td>
</tr>
</tbody>
</table>

Uh-oh... Processors are improving much faster than Memory.

Deep Learning Processors have arrived!

**MOBILE PLATFORMS**

<table>
<thead>
<tr>
<th>Device</th>
<th>Cores</th>
<th>Computation (GFLOPS/s)</th>
<th>Memory Bandwidth (GB/s)</th>
<th>Computation-to-bandwidth ratio</th>
<th>System Power (TDP Watts)</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samsung Galaxy Note 3</td>
<td>Arm Mali T-628 GPU [1]</td>
<td>120 (32-bit float)</td>
<td>12.8 (LPDDR3)</td>
<td>9.3</td>
<td>~10</td>
<td>2013</td>
</tr>
<tr>
<td>Huawei P20</td>
<td>Kirin 970 NPU [2]</td>
<td>1920 (16-bit float)</td>
<td>30 (LPDDR4X)</td>
<td>64 (ouch!)</td>
<td>~10</td>
<td>2018</td>
</tr>
<tr>
<td>NVIDIA Jetson Xavier [3,4]</td>
<td>NVIDIA Tensor Cores</td>
<td>30000 (8→32 int)</td>
<td>137</td>
<td>218 (yikes!)</td>
<td>10 to 30 (multiple modes)</td>
<td>2018</td>
</tr>
</tbody>
</table>

What will the next generation Deep Learning servers look like?

Mythic is aiming for a mere 0.5 picojoules per multiply and accumulate, which would result in about 4 trillion operations per watt (TOPS/W). Syntiant is hoping to get to 20 TOPS/W. An Nvidia Volta V100 GPU can do 0.4 TOPS/W, according to Syntiant. However, real apples-to-apples comparisons in the machine learning world are difficult to determine, Fick points out.

Groq

Groq is founded by Ex-googlers, who designed Google TPU. Groq’s website claims that its first chip will run 400 trillion operations per second with 8TOP/s per Watt power efficiency.

https://medium.com/@shan.tang.g/a-list-of-chip-ip-for-deep-learning-48d05f1759ae

AI Startup Seeks its Voice

Syntiant to sample 20-TOPS/W chip this year

By Rick Merritt, 06.2018 0 0

SAN JOSE, Calif. — Battery-powered devices will get a new option for hardware-accelerated speech interfaces next year if Kurt Busch makes his targets this year. The chief executive of Syntiant aims in 2018 to sample a novel machine-learning chip and raise a Series B to make it in volume.

The startup is designing a 20 tera-operations/watt chip using 4- to 8-bit precision to speed up AI operations initially for voice recognition. It uses an array of hundreds of thousands of NOR cells, computing TensorFlow neural-network jobs in the analog domain.
What will the next generation Deep Learning servers look like?

## 20 TOP/W COMPUTATION

<table>
<thead>
<tr>
<th>Platform</th>
<th>Efficiency (TOP/s/W)</th>
<th>Computation (TOP/s)</th>
<th>Memory Bandwidth (TB/s)</th>
<th>Computation-to-bandwidth ratio</th>
<th>Power (TDP Watts)</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA K20 [1]</td>
<td>0.015</td>
<td>3.50 (32-bit float)</td>
<td>0.208 (GDDR5)</td>
<td>17</td>
<td>225</td>
<td>2012</td>
</tr>
<tr>
<td>NVIDIA V100 [2]</td>
<td>0.45</td>
<td>112 (16-bit float)</td>
<td>0.900 (HBM2)</td>
<td>124</td>
<td>250</td>
<td>2018</td>
</tr>
<tr>
<td>Next-gen: 20 TOP/W</td>
<td>20</td>
<td>2500*</td>
<td>1.800 (HBM3) [3]</td>
<td>1389 (oh no!)</td>
<td>250</td>
<td>2020 (est.)</td>
</tr>
</tbody>
</table>

* Assuming half the power is spent on computation, and the other half is spent on memory and other devices.

20 TOP/s/W * 20W * 0.5 = 2500 TOP/s

Small Neural Nets to the rescue
squeeze (verb): to make an AI system use less resources using whatever means necessary
**squeeze** *(verb)*: to make an AI system use less resources using whatever means necessary
**squeeze** *(verb)*: to make an AI system use less resources using whatever means necessary.
Most CV Applications Rely on Only a Few Core CV Capabilities

And the best accuracy for each of these capabilities is given by Convolutional Neural Nets
But We Need a Very Different Kind of DNN

VGG16\textsuperscript{[1]} model:
- Parameter size: 552 MB
- Memory: 93 MB/image
- Computation: 15.8 GFLOPs/image

Speed more Related to Memory Accesses than Operations

<table>
<thead>
<tr>
<th></th>
<th>L1 D-Cache (per core)</th>
<th>L2 Cache (shared)</th>
<th>Off-chip DRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>32 KB</td>
<td>2 MB</td>
<td>4 GB</td>
</tr>
<tr>
<td>Read Latency</td>
<td>4 cycles</td>
<td>22 cycles</td>
<td>~200 cycles</td>
</tr>
<tr>
<td>Read Bandwidth</td>
<td>20.8 GB/s</td>
<td>166.4 GB/s</td>
<td>28.7 GB/s</td>
</tr>
</tbody>
</table>
Energy More Related to Memory Accesses than operations (45nm 0.9V)

Mark Horowitz, “Computing’s Energy Problem (and what we can do about it),” ISSCC 2014
10,000 DNN Architectural Configurations
Later: SqueezeNet (2016)

AlexNet [1]

SqueezeNet [2]

<table>
<thead>
<tr>
<th>CNN</th>
<th>Top-5 Accuracy</th>
<th>Model Parameters</th>
<th>Model Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet[1]</td>
<td>80.3%</td>
<td>60M</td>
<td>243MB</td>
</tr>
<tr>
<td>SqueezeNet[2]</td>
<td>80.3%</td>
<td>1.2M</td>
<td>4.8MB</td>
</tr>
</tbody>
</table>

compresses to 500KB

SqueezeNet: Immediate Success in Embedded Vision

Enabled embedded processor vendors (ARM, NXP, Qualcomm) to demo CNNs
Quickly ported to all the major Deep Learning Frameworks
SqueezeDet for Object Detection (2017)


- ~2M model parameters
- 57 FPS
- 1.4 Joules Frame

Input image

feature map

Conv Det

Bounding boxes

Filter

Final detections
SqueezeSeg: Semantic Segmentation for LIDAR (2018)

LIDAR point cloud segmentation

SqueezeSegV2:
- Higher accuracy: v1[1]: 64.6% -> v2[2]: 73.2% (+8.6%)
- Better Sim2Real performance: v1[1]: 30% -> v2[2]: 57.4% (+27.4%)
  - Outperforms v1 trained on real data w/o intensity

Squeeze Family

- SqueezeNet
- SqueezeNext
- SqueezeDet
- SqueezeSeg-{v1, v2}
- DiracDeltaNet
- ShiftNet
- DNASNet

**Image Classification**

- (a) Image classification

**Object Detection**

- (b) Object detection

**Semantic Segmentation**

- (c) Semantic segmentation
Andrew Howard's MobileNets: Efficient On-Device Computer Vision Models

Designed for efficiency on mobile phones.
Family of pareto optimal models to target needs of the user.
V1 based on Depthwise Separable Convolutions.
V2 introduces Inverted Residuals and Linear Bottlenecks.
Supports Classification, Detection, Segmentation and more.
Model Compression

10X

DNN Architecture Search

≥50X

Slide Credit: Prof. Warren Gross (McGill Univ.)
Anatomy of a convolution layer
Filters: Kernel Reduction

9x reduction in model parameters
Filters/Channel Reduction

9x reduction in model parameters
Model Distillation

Examples of what's on a DNN Architect's Palette

Spatial Convolution
e.g. 3x3

Pointwise Convolution
1x1

Depthwise Convolution

Channel Shuffle

Shift
The palette of an adept mobile/embedded DNN designer has grown very rich!

Overall architecture: economize on layers while retaining accuracy

Layer types

- Kernel reduction: $5 \times 5 \rightarrow 3 \times 3 \rightarrow 1 \times 1$
- Channel reduction: e.g. FireLayer
- Experiment with novel layer types that consume no FLOPS
  - Shuffle
  - Shift

Model distillation: let big models teach smaller ones

Apply pruning

Tailor bit precision (aka quantization) to target processor

Artistic/Engineering Process of Designing a Deep Neural Net

- Manual design:
  - Each iteration to evaluate a point in the design space is very expensive
  - Exploration limited by human imagination
Can we automate this?

• Manual design:
  • Each iteration to evaluate a point in the design space is very expensive
  • Exploration limited by human imagination
DNAS: Differentiable Neural Architecture Search

Differentiable Neural Architecture Search:
- Extremely fast: 8 GPUs, 24 hours
  - Can search for different conditions case-by-case
- Optimize for actual latency

Bichen Wu, Kurt Keutzer, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia
DNAS in context (FLOPs to normalize comparison)

- MobileNetV2: 
  - Acc: 71.8%
  - FLOPs: 300M

- More FLOPs - BAD

- ImageNet top-1 Accuracy -- Good

- PNAS: [2] Acc: 74.2%
  - FLOPs: 588M
  - Search cost*: 6,000 GPU-hrs

- DARTS: [3] Acc: 73.1%
  - FLOPs: 595M
  - Search cost: 288 GPU-hrs

- AMC: [5] Acc: 70.8%
  - FLOPs: 150M

- NAS: [1] Acc: 74.0%
  - FLOPs: 564M
  - Search cost: 48,000 GPU-hrs

- MnasNet: [6]
  - Acc: 74.0
  - FLOPs: 317M

- Search Cost*: 91,000 GPU-hrs

- DNASNet: (ours)
  - Acc: 74.2%
  - FLOPs: 295M
  - Search Cost: 216 GPU-hrs

- X-axis: FLOPs
- Y-axis: accuracy
- Mark size: search cost
- Circles: search cost unknown

* Estimated from the paper description

DNAS for device-aware search

- For different targeted devices, both DNASNets achieve similar accuracy.
- However, per target DNN optimization was required

<table>
<thead>
<tr>
<th>NET</th>
<th>Latency on iPhoneX</th>
<th>Latency on Samsung S8</th>
<th>Top-1 acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNAS-iPhoneX</td>
<td>19.84 ms</td>
<td>23.33 ms (20% slower)</td>
<td>73.20%</td>
</tr>
<tr>
<td>DNAS-S8</td>
<td>27.53 ms (25% slower)</td>
<td>22.12 ms</td>
<td>73.27%</td>
</tr>
</tbody>
</table>
The Future: Breaking down the wall between DNN Design & Hardware Design

DNN Designers
• Unaware of arithmetic intensity
• Floating point vs fixed point costs
• Memory hierarchy and latency

NN HW Accelerator architects
• Using outdated models:
  - AlexNet
  - VGG16
• Using irrelevant datasets:
  - MNIST
  - CIFAR
Key Takeaways

• Autonomous vehicles currently need thousands (or even hundreds of thousands) of dollars of computing hardware

• Processing is on a trajectory of rapid improvement (in operations-per-Watt)
  • but other aspects of the system (e.g. memory) are improving much more slowly
  • today's neural networks will be choked by slow memory on tomorrow's DNN accelerators (this is already happening and will get worse)

• Designing new (smaller) neural networks helps with all of the following
  • making full use of next-generation computing platforms
  • reducing the hardware costs in autonomous vehicles
  • enabling lower-cost, larger-scale rollouts of autonomous vehicles