Emerging AI Applications: Moving Beyond ResNet50 on ImageNet

Amir Gholami, Michael Mahoney, Kurt Keutzer

and Pallas Group: Suresh Krishna, Ravi Krishna, Zhewei Yao, Zhen Dong, Sehoon Kim, Tianren Gao, Bohan Zhai, Ani Nrusimha

Opening Keynote, Intel System Architecture Summit (ISAS), Feb. 2021
Our Group’s Focus:
All Deep Learning All The Time

Computer Vision and Core ML

Image Classification
Object Detection
Image Segmentation

Natural Language Processing
Translation
Question answering
Document Understanding

Audio Analysis
Audio Enhancement
Call-center Sentiment Analysis
Speech Recognition

Rec Systems
Video Recommendation
Music Recommendation
Ad Recommendation
State-of-the-Art Solutions
Typically Rely on one DNN (or a few)

<table>
<thead>
<tr>
<th>Application</th>
<th>Network Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Classification</td>
<td>Convolutional NN</td>
</tr>
<tr>
<td>Object Detection</td>
<td>Transformer</td>
</tr>
<tr>
<td>Image Segmentation</td>
<td>Recurrent NN (CTC/RNN-T)</td>
</tr>
<tr>
<td>Translation</td>
<td>DLRM</td>
</tr>
<tr>
<td>Question answering</td>
<td></td>
</tr>
<tr>
<td>Document Understanding</td>
<td></td>
</tr>
<tr>
<td>Audio Enhancement</td>
<td></td>
</tr>
<tr>
<td>Call-center Sentiment Analysis</td>
<td></td>
</tr>
<tr>
<td>Speech Recognition</td>
<td></td>
</tr>
<tr>
<td>Video Recommendation</td>
<td></td>
</tr>
<tr>
<td>Music Recommendation</td>
<td></td>
</tr>
<tr>
<td>Ad Recommendation</td>
<td></td>
</tr>
</tbody>
</table>
Most Focus has been on CV

Bandwidth Bound Applications

- Image Classification
- Object Detection
- Image Segmentation
- Convolutional NN
- Translation
- Question answering
- Document Understanding
- Transformer
- Audio Enhancement
- Call-center Sentiment Analysis
- Speech Recognition
- Recurrent NN (CTC/RNN-T)
- Video Recommendation
- Music Recommendation
- Ad Recommendation
- DLRM

CV workloads are not representative of other emerging AI applications
Executive Summary: New Opportunities for DSA

- Emerging AI applications with low arithmetic intensity
  - Recommendation Systems, that need DSA with
    - Large Memory Systems
    - Fast Interconnect
    - Efficient Prefetching and Cache Hierarchy

- AI at the Edge:
  - All AI domains (CV, NLP, RecSys, ASR, Robotics/RL, etc.)
    - Low-precision Inference
    - Unified software interface for better programmability

- Emerging AI Optimization Algorithms:
  - Moving beyond SGD based training to second-order methods
  - Need for DSA that supports fast Randomized algorithms
  - Important applications for Scientific ML/Computing
Executive Summary:
New Opportunities for DSA

- **Emerging AI applications with low arithmetic intensity**
  - Recommendation Systems, that need DSA with
    - Large Memory Systems
    - Fast Interconnect
    - Prefetching/Caching Hierarchy

- **AI at the Edge:**
  - All AI domains (CV, NLP, RecSys, ASR, Robotics/RL, etc.)
    - Low-precision Inference
    - Unified software interface for programmability
    - HW/NN Co-design

- **Emerging AI Optimization Algorithms:**
  - Moving beyond SGD based training to second-order methods
  - Need for DSA that supports fast Randomized algorithms
  - Important applications for Scientific ML/Computing
• The wrong way to interpret this trend is to only focus on increasing peak FLOPs of AI accelerators => Not optimal for emerging AI applications

AI and Memory Wall

Transformer Size: 240x / 2 yrs
AI HW Memory: 2x / 2 yrs

AI and Memory Wall

Scaling of Peak hardware FLOPS, and Memory/Interconnect Bandwidth

- **HW FLOPS:** 90000x / 20 yrs (3.1x/2yrs)
- **DRAM BW:** 30x / 20 yrs (1.4x/2yrs)
- **Inteconnect BW:** 30x / 20 yrs (1.4x/2yrs)

---

Majority of AI Workloads in Datacenter is RecSys (not NLP or CV)

- Recommendation systems, account for more than 80% of AI Cycles in FB.

Majority of AI Workloads in Datacenter is RecSys (not NLP or CV)

- Most academic papers have been focusing on optimizing:
  - CV: 81.6% papers
  - NLP: 6.5% papers

- Only 2% of papers in top conferences are considering recommendation systems

But this is rapidly changing. RecSys is now part of MLCommons/MLPerf benchmark.

The breakdown of application spaces of ML related papers published in HPCA, ASPLOS, ISCA, and MICRO over the last five years. Source: C. Wu et al.
How does this impact AI Hardware?
Rec Systems have extremely low arithmetic intensity

This trend is changing fast because:

• RecSys models are no longer using old NCF type models
• New models are using DNN based approaches similar to NLP but with much large model sizes
  – Importantly they have orders of magnitude smaller Arithmetic Intensity

Arithmetic Intensity = \frac{\#FLOPs}{\#Memory \ \text{OPS}}
Structure of a Recommendation System (DLRM)

- Find the most appropriate ad (Y), for a user (u) with a past history (X_i)

\[
\arg\max_Y P(Y|u, X_i)
\]

- This is done in two phases: Retrieval and Ranking

Retrieval Phase:
Use Lightweight Models

Ranking Phase:
DLRM

[Diagram of retrieval and ranking phases]
argmax \ P(Y|u, X_i)
RecSys is orders of magnitude Bigger in size than CV (and NLP)

- Recommendation Systems have very low arithmetic intensity

<table>
<thead>
<tr>
<th></th>
<th>#Non-zeros</th>
<th>#Sparse</th>
<th>#Dense</th>
<th>Size (GB)</th>
<th>MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>100</td>
<td>$8 \times 10^9$</td>
<td>$7 \times 10^5$</td>
<td>300</td>
<td>100</td>
</tr>
<tr>
<td>B</td>
<td>100</td>
<td>$2 \times 10^{10}$</td>
<td>$2 \times 10^4$</td>
<td>600</td>
<td>80</td>
</tr>
<tr>
<td>C</td>
<td>500</td>
<td>$6 \times 10^{10}$</td>
<td>$2 \times 10^6$</td>
<td>2,000</td>
<td>75</td>
</tr>
<tr>
<td>D</td>
<td>500</td>
<td>$1 \times 10^{11}$</td>
<td>$4 \times 10^6$</td>
<td>6,000</td>
<td>150</td>
</tr>
<tr>
<td>E</td>
<td>500</td>
<td>$2 \times 10^{11}$</td>
<td>$7 \times 10^6$</td>
<td>10,000</td>
<td>128</td>
</tr>
</tbody>
</table>

Existing model sizes are as large as 10TB!
### Comparison with Other Tasks

<table>
<thead>
<tr>
<th>Category</th>
<th>Model Type</th>
<th>Model Size (#Params)</th>
<th>Arithmetic Intensity</th>
<th>Maximum Live Activations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Vision</td>
<td>ResNeXt101-32x4-48</td>
<td>43-829M</td>
<td>300 (Avg)</td>
<td>2-29M</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100 (Min)</td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td>GRU/LSTM/Transformer</td>
<td>10M-1B</td>
<td>2-60</td>
<td>&gt; 100K</td>
</tr>
<tr>
<td>Recommendation</td>
<td>Fully Connected</td>
<td>1-10M</td>
<td>20-200</td>
<td>&gt; 10 K</td>
</tr>
<tr>
<td></td>
<td>Embeddings</td>
<td>&gt; 10 Billion</td>
<td>1-2</td>
<td>&gt; 10 K</td>
</tr>
</tbody>
</table>

Mikhail Smelyanskiy, AI at Facebook Datacenter Scale, Invited Lecture in UC Berkeley EE 290, 2020.
Training Recommendation Systems

For model parallel training we will need **allreduce** and **alltoall** communication:

- Optimized HW needs to have:
  - Fast interconnect to reduce alltoall communication overhead
  - Intelligent Caching and Prefetching
  - Large Capacity High Bandwidth Memory

- We also need optimized ML algorithms to enable:
  - Ultra low precision quantization and fast Structured/Unstructured pruning to speed up inference
  - Robust optimization algorithms that are require less tuning and have lower overhead

---

- Z. Dong*, Z. Yao*, A. Gholami*, M. Mahoney, K. Keutzer, HAWQ: Hessian Aware Quantization of Neural Networks With Mixed Precision, ICCV ’19.
Executive Summary: New Opportunities for DSA

- Emerging AI applications with low arithmetic intensity
  - Recommendation Systems, that need DSA with
    - Large Memory Systems
    - Fast Interconnect
    - Prefetching/Caching Hierarchy

- AI at the Edge:
  - All AI domains (CV, NLP, RecSys, ASR, Robotics/RL, etc.)
    - Low-precision Inference
    - Unified software interface for programmability
    - HW/NN Co-design

- Emerging AI Optimization Algorithms:
  - Moving beyond SGD based training to second-order methods
  - Need for HW that supports fast Randomized algorithms
  - Important applications for Scientific ML/Computing
We observed a big migration to cloud in the past decade.

We are observing a significant migration back to the device driven by privacy concerns, and need for real-time AI.
Inference at the server is **unreliable** and challenging for **latency/energy constrained applications**

- Example: Autonomous cars produce O(10) Gbps data and need latency <100ms under <100 Watts

**User privacy** can be comprised when data is sent to the cloud

- Example: More than 25% of smart speaker users do not want their data to be sent to the cloud [1]

---

We Want to Operate Across a Broad Range of Hosts at the Edge with Limited Resources

Average Power Dissipation

1-1000 mW
8-240 Hours

Webo (13 inch MacBook Air) | 54Wh = 194.4kJ
Apple: 12h = 4.5 W

Eee PC 1000HE | 49Wh = 176kJ
Asus: 9.5h = 5.2 W

15 inch MacBook Pro | 76Wh = 273.6kJ
Apple: 10h = 7.6 W

13 inch MacBook Air | 54Wh = 194.4kJ
Apple: 12h = 4.5 W

iMac Pro | 76Wh = 273.6kJ
Apple: 10h = 7.6 W

Typical handset
32g, 13cc, 5.5Wh = 19.8 kJ

Typical usage
5kJ active + 12kJ standby = 1 battery charge
Per Ljung – Nokia, 2012

典型手持设备
32g, 13cc, 5.5Wh = 19.8 kJ

典型使用
5kJ active + 12kJ standby = 1 battery charge

Per Ljung – Nokia, 2012

1 Wh = 3.6 kJ
We cannot naively use general DNN models
Challenges on the Machine Learning side:

- **We need systematic methods for efficient deployment**
  - Existing quantization and pruning often involve ad-hoc and do not work on new tasks.

- **We need efficient on-device training**, for personalization or online learning scenarios. However, current training methods are not suitable for on-device training
  - Brute force hyperparameter tuning
  - Prohibitive memory footprint

- **We need to rethink and co-design** the NN architecture for efficient edge deployment
Summary: Three Elements of Efficiency at the Edge

- We need to **co-design** Edge HW with the NN applications.
- A big opportunity is to take advantage of **low precision**, and **sparsity**.
**Restricted Energy and Compute at the Edge:**

*Use Low Precision*

- Memory accesses are the principal cost in both latency and energy
- Lower precision weights in DNN mean each memory access brings more data values
- More data values fewer accesses overall

![Image Credit: Sdxcentral, nvidia](image.png)

![Table credit: Mark Horowitz](table.png)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Energy (pJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8b Add</td>
<td>0.03</td>
</tr>
<tr>
<td>16b Add</td>
<td>0.05</td>
</tr>
<tr>
<td>32b Add</td>
<td>0.1</td>
</tr>
<tr>
<td>16b FP Add</td>
<td>0.4</td>
</tr>
<tr>
<td>32b FP Add</td>
<td>0.9</td>
</tr>
<tr>
<td>8b Multiply</td>
<td>0.2</td>
</tr>
<tr>
<td>32b Multiply</td>
<td>3.1</td>
</tr>
<tr>
<td>16b FP Multiply</td>
<td>1.1</td>
</tr>
<tr>
<td>32b FP Multiply</td>
<td>3.7</td>
</tr>
<tr>
<td>32b SRAM Read (8KB)</td>
<td>5</td>
</tr>
<tr>
<td>32b DRAM Read</td>
<td>640</td>
</tr>
</tbody>
</table>

[Horowitz, ISSCC 2014]
Uniform quantization is a linear mapping from floating point values to quantized integer values.

**Floating Point Values**

**8-bit Quantized Values**
Lower precision Multiply-Acc Reduces Energy

- Lower precision weights mean less energy per Multiply-Accumulate
- Also enables putting more MAC units per unit of silicon

<table>
<thead>
<tr>
<th>Bits in integer MAC</th>
<th>TOPS /Watt ~45nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.5 TOPS/Watt</td>
</tr>
<tr>
<td>8</td>
<td>1 (2x)</td>
</tr>
<tr>
<td>4</td>
<td>5 (10x)</td>
</tr>
<tr>
<td>2</td>
<td>10 (20x)</td>
</tr>
</tbody>
</table>

Data from Marian Verhelst, KU Leuven

Big opportunity to enable lower bit precision inference!
Mixed Precision INT4/8 Quantization Works!

Z. Dong*, Z. Yao*, A. Gholami*, M. Mahoney, K. Keutzer, HAWQ: Hessian Aware Quantization of Neural Networks With Mixed Precision, ICCV'19.
Integer-only Quantization

• It is possible to perform integer-only quantization algorithm with only INT multiplication, addition, and bit shifting

• No accuracy degradation for INT8 (5% higher than prior art)

• Direct hardware implementation and verification
  – Up to \textbf{1.5x speed up compared to INT8 quantization}

Lower Precision Can Improve Edge HW

Slide: Courtesy of Prof. Keutzer

- **Peak AI TOPS (INT8)**
  - 100 TOPS
  - 10 TOPS
  - 1 TOPS

- **SmartWatch, Micro Drone**
  - 10μW-200mW
- **Smart Speakers & Cameras**
  - 0.1W-3W
- **Synaptics AS-371**
- **Kneron KL720**
- **NXP i.MX8M+**
- **Qualcomm Wear 4100+**
- **Lattice CrossLink NX-40**
- **FPGA**
- **GreenWaves GAP9**

- **Smartphone, AR/VR goggles**
  - 3W-10W
- **Edge Accelerator**
  - 5W-15W
- **Automotive**
  - 10W-36W

- **Apple A14**
- **Qualcomm XR2**
- **Snapdragon 888**
- **Mythic M1108**
- **FlexLogix InferX1**
- **MobileEye Q5**
- **Tesla FSD**

**Rapid Improvements in Edge Processors is Going to Help**
EdgeAI: Challenging to Deploy

- An important challenge at the edge, is the wide variety of the HW, and the lack of good software that could help accelerate deployment
  - Existing solutions such as TVM are still not fully developed
  - Great opportunity for Intel to provide a standard family of edge HW along with integrated software
Executive Summary:
New Opportunities for DSA

- Emerging AI applications with low arithmetic intensity
  - Recommendation Systems, that need DSA with
    - Large Memory Systems
    - Fast Interconnect
    - Prefetching/Caching Hierarchy

- AI at the Edge:
  - All AI domains (CV, NLP, RecSys, ASR, Robotics/RL, etc.)
    - Low-precision Inference
    - Unified software interface for programmability
    - HW/NN Co-design

- **Emerging AI Optimization Algorithms:**
  - Moving beyond SGD based training to second-order methods
  - Need for HW that supports fast Randomized algorithms
  - Important applications for Scientific ML/Computing
In every iteration of SGD we load a **random mini-batch of training** data, and compute the gradient.
Background: Stochastic Gradient Descent (SGD)

In every iteration of SGD we load a random mini-batch of training data, and compute the gradient.
Rapid Training of NNs

Illustration from Nvidia (B. Ginsburg)
ImageNet Scaling!

- Main benchmark for hardware performance in ML
Great Progress on Scaling ResNet50 on ImageNet

Training ResNet50 on ImageNet requires **720 hours** on a SINGLE Maxwell Titan X

- <1 minute
- >50,000x speedup

Despite the great progress the solutions **do not work** on problems other than ResNet50 on ImageNet
ResNet50 on ImageNet is too Simple!

- The loss landscape of ResNet50 is too simple, and not representative of new workloads

Image Source

Emerging Solutions:
Going Beyond Simple SGD

Loss Function

First-Order Methods
(SGD)

Second-Order Methods
(AdaHessian)

Benchmark Link: https://github.com/jettify/pytorch-optimizer
Emerging Solutions: Going Beyond Simple SGD

• New second-order algorithms are emerging for training NNs

How would this impact HW design?
• Need fast Randomized Linear Algebra
  – Randomized Matrix-Matrix Operations
  – RBLAS

Z. Yao*, A. Gholami*, Q. Lei, K. Keutzer, M. Mahoney, Hessian-based Analysis of Large Batch Training and Robustness to Adversaries, NeurIPS'18, 2018.

Code: https://github.com/amirgholami/PyHessian
Code: https://github.com/amirgholami/AdaHessian
DSA with Fast Randomized Matrix Operations

- The core of these algorithms is multiplying a matrix with a random matrix
  - Accelerating Randomized operations can have a huge impact

```plaintext
% inner product approach
for i = 1:I
  for j = 1:J
    for k = 1:K
      C(i,j) = C(i,j) + A(i,k)*B(k,j);
```


Matrix Image Source: https://patterns.eecs.berkeley.edu/?page_id=158

NSF Ballistic Project (UCB, UoT, ORNL, UOM, CU)

Fast RNG can significantly improve performance
Summary: New Opportunities for DSA

- **Emerging AI applications with low arithmetic intensity**
  - Recommendation Systems, that need DSA with
    - Large Memory Systems
    - Fast Interconnect
    - Efficient Prefetching and Cache Hierarchy

- **AI at the Edge:**
  - All AI domains (CV, NLP, RecSys, ASR, Robotics/RL, etc.)
    - Low-precision Inference
    - Unified software interface for better programmability

- **Emerging AI Optimization Algorithms:**
  - Moving beyond SGD based training to second-order methods
  - Need for DSA that supports fast Randomized algorithms
  - Important applications for Scientific ML/Computing
Thanks for Listening

For any feedback/questions please contact amirgh@berkeley.edu