Lecture 7:

Hardware Acceleration of DNNs

Visual Computing Systems
Stanford CS348K, Spring 2024
Hardware acceleration of DNN inference/training

- Google TPU3
- GraphCore IPU
- Huawei Kirin NPU
- Apple Neural Engine
- Intel Deep Learning Inference Accelerator
- SambaNova Cardinal SN10
- Cerebras Wafer Scale Engine
- Ampere GPU with Tensor Cores
Investment in AI hardware

**SambaNova Systems Raises $676M in Series D, Surpasses $2B Valuation and Becomes World’s Best-Funded AI Startup**

SoftBank Vision Fund 2 leads round backing breakthrough platform that delivers unprecedented AI capability and accessibility to customers worldwide.

April 16, 2023 | 08:30 AM Eastern Time Zone

Palo Alto, Calif. - (BUSINESS WIRE) - SambaNova Systems, the company building the industry’s most flexible and scalable to run AI applications, today announced a $676 million Series D funding round ("Fund 2"). The round includes additional new investors, including RMR and SIG, plus existing backers including BlackRock, Intel Capital, GV (formerly Google Ventures), Western International and WBA.

"We’re thrilled to re-invest in the AI market, and this round greatly accelerates that mission," said SambaNova’s co-founder and CEO, Brad Mollow. "Our Series D brings SambaNova’s total funding and raises its valuation to more than $6 billion. Now, the best-funded AI systems and services provider, SambaNova will use its latest iteration in Legacy Computers, an industry leader in AI hardware and software, to bring AI to the mainstream for private and public sectors around the world. We’re here to revolutionize the AI market and the round greatly accelerates that mission," said Founder and CEO, Brad Mollow. "Traditional CPUs and GPUs have reached their computational limits. To solve humanity’s greatest technology challenges, a new approach is needed. We’re excited to see our wealth of investors validate that..." 

SambaNova’s flagship offering is Dataflow-as-a-Service (DaaS), a subscription-based, extensible AI services platform designed to provide enterprises with AI capabilities and accelerating the work of existing data centers, allowing the enterprise to focus on its business without an investment in infrastructure.

---

**Artificial intelligence chip startup Cerebras Systems claims it has the "world's fastest AI supercomputer," thanks to its large Wafer Scale Engine processor that comes with 400,000 compute cores.**

The Los Altos, Calif.-based startup introduced its CS-1 system at the Supercomputing conference in Denver last week after raising more than $200 million in funding from investors, most recently with an $88 million Series D round that was raised in November 2018, according to Andrew Feldman, the founder and CEO of Cerebras who was previously an executive at AMD.

---

**NVIDIA Market Cap 2014 - 2021**

![NVIDIA Market Cap Graph](Image)

Applications based on artificial intelligence — whether they are systems running on autonomous services, platforms being used in drug development or to predict the spread of a virus, traffic management for TCO optimisation or something else altogether — require an unprecedented amount of computing power to run. And today, one of the big names in the world of designing and manufacturing AI hardware is NVIDIA.

---

**Intel Acquires Artificial Intelligence Chipmaker Habana Labs**

**Combination Advances Intel’s AI Strategy, Strengthens Portfolio of AI Accelerators for the Data Center**

SANTA CLARA, Calif., Dec. 16, 2019 — Intel Corporation today announced that it has acquired Habana Labs, an Israeli-based developer of programmable deep learning accelerators for the data center for approximately $2 billion. The combination strengthens Intel’s artificial intelligence (AI) portfolio and accelerates its efforts in the nascent, fast-growing AI silicon market, which Intel expects to be greater than $25 billion by 2024.

"This acquisition advances our AI strategy, which is to provide customers with solutions to fit every performance need – from the intelligent edge to the data center," said Navin Shenoy, executive vice president and general manager of the Data Platforms Group at Intel. "More specifically, Habana turbo-charges our AI offerings for the data center with a high-performance training processor family and a standards-based programming environment to address evolving AI workloads."
Review (again) two computer architecture facts
Compute specialization = energy efficiency

- Rules of thumb: compared to high-quality C code on CPU...

- Throughput-maximized processor architectures: e.g., GPU cores
  - Approximately 10x improvement in perf/watt
  - Assuming code maps well to wide data-parallel execution and is compute bound

- Fixed-function ASIC (“application-specific integrated circuit”)
  - Can approach 100-1000x or greater improvement in perf/watt
  - Assuming code is compute bound and
    and is not floating-point math
Data movement has high energy cost

- Rule of thumb in modern system design: always seek to reduce amount of data movement in a computer

- “Ballpark” numbers
  - Integer op: ~ 1 pJ *
  - Floating point op: ~20 pJ *
  - Reading 64 bits from small local SRAM (1mm away on chip): ~ 26 pJ
  - Reading 64 bits from low power mobile DRAM (LPDDR): ~1200 pJ

- Implications
  - Reading 10 GB/sec from memory: ~1.6 watts
  - Entire power budget for mobile GPU: ~1 watt
    (remember phone is also running CPU, display, radios, etc.)
  - iPhone 6 battery: ~7 watt-hours (note: my Macbook Pro laptop: 99 watt-hour battery)
  - Exploiting locality matters!!!

[Sources: Bill Dally (NVIDIA), Tom Olson (ARM)]

* Cost to just perform the logical operation, not counting overhead of instruction decode, load data from registers, etc.
Improving hardware efficiency for DNN operations
Efficiency estimates *

- Estimated overhead of programmability (instruction stream, control, etc.)
  - Half-precision FMA (fused multiply-add) 2000%
  - Half-precision DP4 (vec4 dot product) 500%
  - Half-precision 4x4 MMA (matrix-matrix multiply + accumulate) 27%

NVIDIA Xavier (SoC for automotive domain)

Features a Computer Vision Accelerator (CVA), a custom module for deep learning acceleration (large matrix multiply unit)

~ 2x more efficient than NVIDIA V100 MMA instruction despite being highly specialized component. (includes optimization of gating multipliers if either operand is zero)

* Estimates by Bill Dally using academic numbers, SysML talk, Feb 2018
Ampere GPU SM (A100)

Each SM core has:
64 fp32 ALUs (mul-add)
32 int32 ALUs
4 “tensor cores”

Execute 8x4 x 4x8 matrix mul-add instr
A x B + C for matrices A,B,C
A, B stored as fp16, accumulation with fp32 C

There are 108 SM cores in the GA100 GPU:
6,912 fp32 mul-add ALUs
432 tensor cores
1.4 GHz max clock
= 19.5 TFLOPs fp32
+ 312 TFLOPs (fp16/32 mixed) in tensor cores

The NVIDIA tensor core approach is an evolutionary design: add DNN-specific instructions to a traditional programmable processor (“evolve, don’t replace”)
Google TPU
(version 1)
**TPU area proportionality**

Arithmetic units ~ 30% of chip
Note low area footprint of control

Key instructions:
- read host memory
- write host memory
- read weights
- matrix_multiply / convolve
- activate

---

Figure credit: Jouppi et al. 2017
Systolic array

(matrix vector multiplication example: $y=Wx$)
Systolic array

(matrix vector multiplication example: \( y = Wx \))
Systolic array

(matrix vector multiplication example: \( y = Wx \))
Systolic array
(matrix vector multiplication example: $y=Wx$)
Systolic array

(matrix vector multiplication example: \( y=WX \))
Systolic array

(matrix vector multiplication example: $y = Wx$)
Systolic array

(matrix matrix multiplication example: $Y = WX$)

Notice: need multiple 4x32bit accumulators to hold output columns
Building larger matrix-matrix multiplies

Example: $A = 8 \times 8$, $B = 8 \times 4096$, $C = 8 \times 4096$

Assume $4096$ accumulators
Building larger matrix-matrix multiplies

Example: $A = 8 \times 8$, $B = 8 \times 4096$, $C = 8 \times 4096$

Assume 4096 accumulators
Building larger matrix-matrix multiplies

Example: $A = 8 \times 8$, $B = 8 \times 4096$, $C = 8 \times 4096$

Assume 4096 accumulators
Building larger matrix-matrix multiplies

Example: $A = 8 \times 8$, $B = 8 \times 4096$, $C = 8 \times 4096$

Assume 4096 accumulators
TPU Performance/Watt

GM = geometric mean over all apps
WM = weighted mean over all apps

total = cost of host machine + CPU
incremental = only cost of TPU

Figure credit: Jouppi et al. 2017
Alternative scheduling strategies

TPU (v1) was “weight stationary”: weights kept in register at PE each PE gets different input pixel partial sum pushed through array (array has one output)

“Output stationary”: each PE computes one output push input pixel through array each PE gets different weight each PE accumulates locally into output

Takeaway: many DNN accelerators can be characterized by the data flow of input activations, weights, and outputs through the machine. (Just different “schedules”!)
Input stationary design (dense 1D conv example)

(matrix vector multiplication example: \( y = Wx \))

Assume:
1D input/output
3-wide filters
2 output channels (K=2)

Stream Order | Weight
---|---
6 | w(1,2)
5 | w(1,1)
4 | w(1,0)
3 | w(0,2)
2 | w(0,1)
1 | w(0,0)

Stream of weights
(2 1D filters of size 3)

Processing elements
(implement multiply)

Accumulators
(implement +=)
Scaling up (for training big models)

Example: GPT-3 language model

Very big models + More training = Better accuracy

Power law effect: exponentially more compute to take constant step in accuracy

\[ L = 2.57 \cdot C^{-0.048} \]

(Amount of training — note this is log scale)
TPU v3 supercomputer

TPU v3 board
4 TPU3 chips

One TPU v3 board

TPUs connected by 2D Torus interconnect

TPU supercomputer (1024 TPU v3 chips)
Additional examples of “AI chips”

Key ideas:

1. Huge numbers of compute units

2. Huge amounts of on-chip storage to maintain input weights and intermediate values
GraphCore MK2 GC200 IPU

- IPU-Tiles™: 1472 independent IPU-Tiles™ each with an IPU-Core™ and In-Processor-Memory™
- IPU-Core™: 1472 independent IPU-Core™
- In-Processor-Memory™: 400MB In-Processor-Memory™ per IPU
- 900 MB on-chip storage
- Access to off-chip DDR4
- (59B transistors similar size to A100 GPU)
Cerebras Wafer-Scale Engine (WSE)

- Tightly interconnected tile of chips (entire wafer)
- Many more transistors (1.2T) than largest single chips
  (Example: NVIDIA A100 GPU has 54B)

Compilation of DNN to platform involves “laying out” DNN layers in space on processing grid.

<table>
<thead>
<tr>
<th>Cerebras WSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chip size</td>
</tr>
<tr>
<td>Cores</td>
</tr>
<tr>
<td>On chip memory</td>
</tr>
<tr>
<td>Memory bandwidth</td>
</tr>
<tr>
<td>Fabric bandwidth</td>
</tr>
</tbody>
</table>
SambaNova reconfigurable dataflow unit

Again, notice tight integration of storage and compute
Another example of spatial layout

Notice: inter-layer communication occurs through on-chip interconnect, not through off-chip memory.
Low precision
Numerical data formats

<table>
<thead>
<tr>
<th>Format</th>
<th>Range</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP32</td>
<td>$10^{-38} - 10^{38}$</td>
<td>0.000006%</td>
</tr>
<tr>
<td>FP16</td>
<td>$6 \times 10^{-5} - 6 \times 10^{4}$</td>
<td>0.05%</td>
</tr>
<tr>
<td>Int32</td>
<td>$0 - 2 \times 10^{9}$</td>
<td>33%</td>
</tr>
<tr>
<td>Int16</td>
<td>$0 - 6 \times 10^{4}$</td>
<td>33%</td>
</tr>
<tr>
<td>Int8</td>
<td>$0 - 127$</td>
<td>33%</td>
</tr>
<tr>
<td>BF16</td>
<td>Same range as FP32, but lower accuracy</td>
<td></td>
</tr>
<tr>
<td>BF8 E4M3</td>
<td>$0 - 448$</td>
<td></td>
</tr>
<tr>
<td>BF8 E5M2</td>
<td>$0 - 57344$</td>
<td></td>
</tr>
</tbody>
</table>

Reminder:
$-1^S \times (1 + (M \times 2^{-23})) \times 2^{E-127}$

Slide credit: Bill Dally
Summary of hardware accelerators for efficient inference

- Specialized instructions for dense linear algebra computations
  - Reduce overhead of control (compared to CPUs/GPUs)
- Systolic / dataflow architectures for efficient on-chip communication
  - Different scheduling strategies: weight-stationary, input/output stationary, etc.
- Reduced precision operations (cheaper computation + reduce bandwidth requirements)
- Huge amounts of on-chip memory to avoid off-chip communication