SW/HW Innovations in Emerging DL Training Systems

Urmish Thakker
Principal Engineer
Too Hot | Goldilocks Zone | Too Cold

©2021 SambaNova Systems
Yesterday’s Goldilocks Zone is Constraining Progress

Highly Detailed

- Transformer (Distilled)
- EfficientNet

Bigger Models

- Transformer (Standard)
- ResNet-50
- GPT, DLRM
- YOLO, Mask R-CNN

HW Performance

Inefficient execution model

Insufficient memory

©2021 SambaNova Systems
Trend of SOTA Models

Highly detailed

Bigger Models

TinyBERT: Distilling BERT for Natural Language Understanding

Xiaoqi Jiao¹, Yichun Yin¹, Lifeng Shang², Xin Jiang²
Xiao Chen³, Linlin Li⁴, Fang Wang⁵ and Qun Liu⁶

¹Key Laboratory of Information Storage System, Huazhong University of Science and Technology, Wuhan National Laboratory for Optoelectronics
²Huawei Noah’s Ark Lab

DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF
Hugging Face
{victor, lysandre, julien, thomas}@huggingface.co

©2021 SambaNova Systems
Our Mission

Shaping the next-generation ML / DL computing system to accelerate the full model spectrum
How do we break out of the Godilocks Zone?

Fundamental advances required at all layers of the SW/HW stack.
The SambaNova Systems Advantage

Application innovations

- High model accuracy
- High compute efficiency

Models

Algorithms

Compiler

Architecture

VLSI

Flexibility and Efficiency

Optimization Within & Between Layers

©2021 SambaNova Systems
Part 1.

Enabling higher compute efficiency
Architecture: Reconfigurable Dataflow Unit (RDU)

Parallel Patterns

Array of reconfigurable compute, memory and communication

- map
- filter
- reduce
- groupBy

Key1
Key2
Key3
Spatial Dataflow Within an RDU

The old way: kernel-by-kernel

The Dataflow way: spatial

SambaFlow eliminates overhead and maximizes utilization

©2021 SambaNova Systems
Rapid Dataflow Compilation to RDU
SambaFlow Produces Highly Optimized Spatial Mappings
Uncompromised Programmability and Efficiency
Breaking out of the programmability vs. efficiency tradeoff curve
Achieve low time-to-accuracy
Part 2. High model accuracy:

+ Pure 16-bit FPU training
+ Asynchronous pipeline parallelization
Low Precision (< 32-bit) Training

Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to $+1$ or $-1$

Matthieu Courbariaux$^{1,2}$
Hay Huhara$^{2}$
Daniel Soudry$^{3}$
Ron El-Yaniv$^{4}$
Yoshua Bengio$^{1,4}$

$^{1}$Université de Montréal
$^{2}$Technion - Israel Institute of Technology
$^{3}$Columbia University
$^{4}$CIFAR Senior Fellow

*Indicates equal contribution. Ordering determined by coin flip.

Training Deep Neural Networks with 8-bit Floating Point Numbers

Naigang Wang, Jungwook Choi, Daniel Brand, Chia-Yu Chen and Kaushik Gopalakrishnan
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
{nwang, chotj, danbrand, cchen, kailash}@us.ibm.com

Recurrent Neural Networks With Limited Numerical Precision

Joachim Ott$^{*}$, Zhouhan Lin$^{1}$, Ying Zhang$^{1}$, Shih-Chii Liu$^{*}$, Yoshua Bengio$^{1,4}$

$^{1}$Institute of Neuroinformatics, University of Zurich and ETH Zurich
ottj@ethz.ch, shih@in.ti.t.hz.ch
$^{2}$DÉpartement d’informatique et de recherche opérationnelle, Université de Montréal
$^{3}$CIFAR Senior Fellow
{zhouhan.lin, ying.zhang}@umontreal.ca

Higher system efficiency, minimal impact on acc. for specific models
Efficiency of Low Precision Floating-point-units (16 vs. 32-bit)

1.5X lower chip area

3X higher energy efficiency

1.5X higher throughput

1. Horowitz. ISSCC 2014
2. Galal et al. ISCA 2013

©2021 SambaNova Systems
Mixed Precision for **Generic** DL Training (16 + 32 bits FPU)

**Illusion:**

16-bit FPU alone is not enough to maximize model acc.
Can we support only 16-bit FPU on accelerators & achieve model acc. matching 32-bit training?
Pure 16-bit (BFloat16) FPU Training

Data Flows

- 16-bit Activation
- 16-bit Gradient
- 16-bit Weight

Forward

Weight update

Multiply-Accumulation Units

- 16-bit Multiplier
- 32-bit Accumulator
- 32-bit Nearest rounding
- 16-bit

Primary source of numerical error

©2021 SambaNova Systems
The Accuracy Challenge

Standard 16-bit FPU training degrades model accuracy

16% acc. gap
The Devil: Nearest Rounding (NR) for Model Weight Updates

\[ w_{t+1} = Q \left( w_t - \alpha \nabla f_{\sigma(t)}(w_t) \right) \]

Model weights halt when updates becomes small
The Devil: Nearest Rounding (NR) for Model Weight Updates

Theory sketch for least-squares regression

\[ \| w_t - w^* \| \geq O \left( \epsilon \cdot \min_j |w_j^*| \right) \]

- Machine precision
- Optimal solution
- j-th dim of the optimal solution

Inaccurate weight update fundamentally degrades convergence
Stochastic Rounding to the Rescue

Intuition
The expectation of unbiased estimates is as accurate as weights w/o rounding
Kahan Summation as Alternative Enhancement

Auxiliary 16-bit values to track and correct weight update errors from NR

Large weight + (Update - Error) = New weight

Nearest rounding

Rounded new weight - Error

Next iteration

Weight Weight - Update

Auxiliary 16-bit value
Experiment:

Pure 16-bit training can match 32-bit training in model acc.
Summary

With support for

Stochastic rounding &

Kahan summation

Accelerators with only 16-bit compute units can match acc. of 32-bit training
Model (Pipeline) Parallelism

Loss

Compute unit 3
F_3 → B_3

Compute unit 2
F_2 → B_2

Compute unit 1
F_1 → B_1

Compute unit 0
F_0 → B_0

Gradients

Microbatches
F_{0,0} F_{0,1} F_{0,2} F_{0,3}
F_{1,0} F_{1,1} F_{1,2} F_{1,3}
F_{2,0} F_{2,1} F_{2,2} F_{2,3}
F_{3,0} F_{3,1} F_{3,2} F_{3,3}

Synchronization barrier

Stage 3
Update
Stage 2
Update
Stage 1
Update
Stage 0
Update

Time

F_0

Update
Update
Update
Update

Stage 3
Stage 2
Stage 1
Stage 0

Bubble

B_0

B_{0,0} B_{0,1} B_{0,2} B_{0,3}
B_{1,0} B_{1,1} B_{1,2} B_{1,3}
B_{2,0} B_{2,1} B_{2,2} B_{2,3}
B_{3,0} B_{3,1} B_{3,2} B_{3,3}

1. Huang et. Neurips 2019

©2021 SambaNova Systems
Model (Pipeline) Parallelism: Are we there yet?

Conventional processor pipeline

# of pipeline stages \[\uparrow\]    Throughput \[\uparrow\]

Model training pipeline with synchronization barrier

# of pipeline stages \[\uparrow\]    Utilization \[\downarrow\]

How much utilization do we really need to sacrifice?
Async. Pipeline Parallelism Steady State

\[ W_{i,j} \text{ Stage } i \text{ weight after } j\text{-th update} \]

\[ M_i \text{ i-th minibatch} \]

Stage 1

\[ W_{1,0} \]

\[ M_6 \]

\[ M_1 \]

Stage 2

\[ W_{2,1} \]

\[ M_5 \]

\[ M_2 \]

Stage 3

\[ W_{3,2} \]

\[ M_4 \]

\[ M_3 \]

Goal: No hardware sacrifices!
Async. Pipeline Parallelism Steady State

$W_{i,j}$ Stage $i$ weight after $j$-th update

$M_i$ $i$-th minibatch

$W_{1,5}$

$W_{2,6}$

$W_{3,7}$

Stage 1

Stage 2

Stage 3

$M_6$ uses $W_{1,0}$ for forward and $W_{1,5}$ for backward: delay = 5

$M_6$ uses $W_{3,4}$ for forward and $W_{3,5}$ for backward: delay = 1

Panic: Introduces different asynchrony (delays) at different stages.
Houston, we have a problem.

**Key Insight:** Scale your learning rate proportional to the delay.

\[ \alpha = \min \left( \alpha_{\text{sync}}, \frac{C}{T_i} \right) \]

Chris De Sa

©2021 SambaNova Systems
Maximize efficiency with no accuracy compromise

ResNet 50: Cifar 10

Transformer: IWSLT

©2021 SambaNova Systems
The SambaNova Systems Advantage

Application innovations

- Models
- Algorithms
- Compiler
- Architecture
- VLSI

Flexibility and Efficiency

Optimization Within & Between Layers

©2021 SambaNova Systems
Part 3. Model Innovations:

Powered by our architecture and algorithm
Computer Vison
Evolution of high-resolution Deep Learning

Low-resolution
(e.g. cats)

4k images
(e.g. Autonomous driving)

50k x 50k
(e.g. astronomy, medical imaging, virus, …)
No Compromise High-Res Segmentation

Classic: chop image into sub-images

Loses information in output!

<table>
<thead>
<tr>
<th>Tiled input</th>
<th>conv</th>
<th>conv</th>
<th>Tiled output</th>
</tr>
</thead>
<tbody>
<tr>
<td>SambaNova: Full image processing</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tiled input</th>
<th>conv</th>
<th>conv</th>
<th>Full output</th>
</tr>
</thead>
</table>

Training w/o information loss from full-image processing

83.1% SOTA IOU
89.6%
High-Res Pathology with Slide-level Label (TCGA)

Train with Patch label = slide label

Noisy patches limit model accuracy
High-Res Pathology with Slide-level label (TCGA)

16X larger patches \(\rightarrow\) 6 Pt higher AUC
Recommender Models
The backbone of many internet services

Entertainment: Apple, hulu, Netflix
Social Media: Facebook, LinkedIn, Pinterest
E-Commerce: Alibaba, Amazon, eBay
Consumer Services: Lyft, Mastercard, Yelp
Recommender systems

Key common component: Sparse embedding feature
Recommender systems

More embedding features, more accuracy
State-of-the-art accuracy on DLRM

80.46% SOTA Accuracy

33% Faster Step-to-accuracy
Bigger isn’t always better…but it is sometimes.

**Training Performance**

<table>
<thead>
<tr>
<th>Embed Dim 128</th>
<th>Embed Dim 256</th>
<th>Embed Dim 512</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput</td>
<td>Throughput</td>
<td>Throughput</td>
</tr>
<tr>
<td>RDU</td>
<td>RDU</td>
<td>RDU</td>
</tr>
<tr>
<td>4GB</td>
<td>8GB</td>
<td>16 GB</td>
</tr>
<tr>
<td>163GB</td>
<td>327 GB</td>
<td>655 GB</td>
</tr>
</tbody>
</table>

SambaNova scales to training massive recommender models

r5d.metal (CPU, FP32)
Natural Language Processing

Breakthrough efficiency in NLP model online deployment

Distilled tiny Bert model + Short sequence input

Siri, what is the weather in SF?
Breakthrough Efficiency in NLP Model Online Deployment

Enable up to 11X speedup for online training and inference
Pretraining and Finetuning

Pretraining

StackExchange

LM

Specialize

Free Law Project

Finetuning

Deploy Application

©2021 SambaNova Systems
Domain Adaptation

- BERT
- Legal
- Medical
- Finance
Scaling Laws for Neural Language Models

Application accuracy improves as the size of the language model increases.
Language Models are Few-Shot Learners
GPT Family

- BERT: 0.3 Billion
- GPT2-XL: 1.2 Billion
- GPT3-13b: 13 Billion
- GPT3: Will start training next month

©2021 SambaNova Systems