Deep learning acceleration: A killer application for in-memory computing?

Abu Sebastian
Distinguished Research Staff Member
IBM Research - Zurich

MiM webinar, TU Vienna, June 21st 2021
Outline

• **Introduction**
  – Deep learning
  – In-memory computing

• **Deep learning based on computational phase-change memory**
  – Phase-change memory and synaptic emulation
  – DL inference and training
  – In-memory compute core
  – Device-level innovations

• **Applications beyond conventional DL**
  – DNN + “something”
  – Spiking deep neural networks
Outline

• **Introduction**
  – Deep learning
  – In-memory computing

• **Deep learning based on computational phase-change memory**
  – Phase-change memory and synaptic emulation
  – DL inference and training
  – In-memory compute core
  – Device-level innovations

• **Applications beyond conventional DL**
  – DNN + “something”
  – Spiking deep neural networks
Deep (artificial) neural networks

DNNs meet the IT revolution

Mainstay of the AI portfolio of almost all IT companies
- Translation
- Search ranking
- News feed
- Face recognition
- Content understanding
DNN’s Computational Efficiency Problem

Training Image recognition model
Dataset: ImageNet-22K
Network: ResNet-101

4 GPUs
16 days
~385 kWh

256 GPUs
7 hours
~450 kWh

For reference: 1 model training run is
Approx. 2 weeks of home energy consumption
https://arxiv.org/abs/1708.02188

- Deep learning is computationally intensive
  - Time consuming even with high-performance computing resources
  - Power consumption prohibitive for applicability in domains such as IoT

Abu Sebastian, IBM Research - Zurich
Key driver for innovations in computing systems

**General purpose computing systems**
- **STORAGE** (e.g. Flash, HDD) (nonvolatile, slow)
- **MEMORY** (e.g. DRAM) (volatile, fast)
- **NEAR MEMORY COMPUTING**

**Conventional digital accelerators**
- GPUs, TPUs, Graphcore, Inherentia, Kunlun, Hanguang etc.
- High-bandwidth memory

**Non-von Neumann accelerators with post-CMOS devices**

Abu Sebastian, IBM Research - Zurich
In-memory computing

- Perform “certain” computational tasks in place in memory
- Achieved by exploiting the physical attributes of memory devices
- Can we viewed as a sub-category of processing in memory (PIM) or compute in memory (CIM)
- At no point during computation, the memory content is read back and processed at the granularity of a single memory element

Memory devices and applications for in-memory computing

Abu Sebastian, IBM Research - Zurich
Outline

• **Introduction**
  – Deep learning
  – In-memory computing

• **Deep learning based on computational phase-change memory**
  – Phase-change memory and synaptic emulation
  – DL inference and training
  – In-memory compute core
  – Device-level innovations

• **Applications beyond conventional DL**
  – DNN + “something”
  – Spiking deep neural networks
Phase-change memory

Commonly used phase change materials

- A nanometric volume of phase change material between two electrodes
- “WRITE” Process
  - By applying a voltage pulse the material can be changed from crystalline phase (SET) to amorphous phase (RESET)
- “READ” process
  - Low-field electrical resistance

Burr et al., JETCAS (2016)

Analog storage and accumulation behavior

- Can achieve a continuum of conductance states
- A non-volatile integrator of pulses

Phase-change synapses

**Synaptic efficacy (Inference)**
- **Analog Storage**
  - (Ohm’s law and Kirchhoff’s circuit law)

- **Presynaptic neurons**
  - $u_1$
  - $u_2$

- **Postsynaptic neurons**
  - $I_1$
  - $I_2$

**Synaptic plasticity (Training)**
- **Accumulation behavior**
  - (PCM crystallization dynamics)

---


Abu Sebastian, IBM Research - Zurich
DNN inference with in-memory computing


The trained synaptic weights are mapped to an array of computational memory cores performing matrix vector multiply operations corresponding to each layer.
Key challenge is the imprecision arising from conductance fluctuation, drift etc.

A custom “additive noise training” procedure is essential to overcome this

Experimental demonstration using PCM devices fabricated in 90nm CMOS technology

DNN training with in-memory computing

![Diagram of neural network]

High Precision!
Non-ideal accumulative behavior
Update weights
Apply acc. pulses sporadically & blindly!

Exploits the accumulative behavior

Nandakumar et al., ArXiv, 2017
Sebastian et al., VLSI, 2019
Abu Sebastian, IBM Research - Zurich
DNN training with in-memory computing

- Each synaptic weight mapped to two PCM devices (~400,000 PCM devices)
- Comparable test accuracy as FP32 training
- Negligible accuracy drop during inference after training


Abu Sebastian, IBM Research - Zurich
Architecture of an IMC-based accelerator

- Mixed-signal in-memory compute cores and system integration


Hermes: IMC compute core in 14nm CMOS technology

- 256x256 array of 8T4R unit cells
- PCM devices integrated in the back-end of 14nm CMOS chip
- Compact current controlled oscillator-based ADCs
- Local digital processing
- MVM performance: 10.5 TOPS/W and 16.5 TOPS/W/sq.mm.

Towards higher precision: Projected memory

- Modified PCM device concept
- Exploits the I-V characteristic of phase change materials
- Substantially lower drift and conductance fluctuations arising from 1/f noise
- Precision equal to 8-bit fixed-point arithmetic

Towards higher speed: Photonic in-memory computing

Feldmann et al., Parallel convolution processing using an integrated photonic tensor core, Nature (2021)
Outline

• Introduction
  – Deep learning
  – In-memory computing

• Deep learning based on computational phase-change memory
  – Phase-change memory and synaptic emulation
  – DL inference and training
  – In-memory compute core
  – Device-level innovations

• Applications beyond conventional DL
  – DNN + “something”
  – Spiking deep neural networks
Hardware lottery

• Most of the algorithmic components for DNNs were in place decades earlier
  – Backprop (1963, 1976, 1988 (Rumelhart et al.)
  – CNNs (Fukushima & Miyake, 1982, LeCun et al., 1989)
• Several decades lost due to the lack of adequate hardware
• We need to ensure that good ideas are not lost or delayed this way
• What is next in deep learning?
• What role can IMC play going forward?

Hooker et al., The hardware lottery, ArXiv, 2020
The deep learning landscape

Computer vision

He et al., Deep residual learning for image recognition (2015)

Machine translation

Vaswani et al., Attention is all you need (2017)

DNNs + “something”

Graves et al., Neural Turing machines (2014)

Hassabis et al., Neuroscience-inspired AI (2017)

More biologically plausible DNNs

Pfeiffer and Pfeil, Deep learning with spiking neurons: Opportunities and challenges (2018)
Recent work on realizing explicit memory in terms of high dimensional vectors
- Powerful tool for few-shot learning
- High-dimensional algebraic operations for variable binding?

Efficient realization using IMC

- The high-dimensional explicit memory content (support vectors) stored in a PCM crossbar array
- The similarity between the input query vectors and support vectors computed through in-memory dot product operations

Spiking Neural Networks (SNNs)

Information transmitted as floating point numbers!

Conventional Neural Networks

Static non-linear functions

Scalar multiply units

Neuronal dynamics

\[ \frac{du}{dt} = F(u) + G(u)I \]

Information transmitted in terms of spikes (rate, timing etc.)

Asynchronous
- Local, event-based learning
- Employed by the brain

Synaptic dynamics

\[ I_{syn} = g_{syn}S(V - E_{syn}) \]
Why SNNs?

- Asynchronous processing (Energy efficiency)
- Temporal codes (Ultra-low latency)
- Local event-based learning (Energy efficiency)
- Synaptic dynamics (Computational superiority in specific AI tasks?)

_Pfeiffer and Pfeil, Front. Neuroscience (2018)_
_Rajendran, Sebastian et al., IEEE SP Magazine (2019)_
Inference in Dynamically Changing Environments

- Demonstration of an SNN surpassing ANNs in a specific task
- Purely neuromorphic and biologically modeled
- Cortical-like circuits can perform Bayesian inference on dynamic environments

Moraitis, Sebastian, Eleftheriou, Short-term synaptic plasticity optimally models continuous environments, ArXiv (2020)
IMC for SNNs

- IMC-based DNN acceleration can be easily extended to Spiking Neural Networks
- Synaptic efficacy and plasticity efficiently realized with physically instantiated synaptic arrays
- Potential to implement even more intricate synaptic dynamics and update rules


Moraitis et al., “Short-term synaptic plasticity optimally models continuous environments”, ArXiv, 2020

Abu Sebastian, IBM Research - Zurich
Summary

- Deep learning is a key driver for innovations in computing systems
- New forms of computing such as in-memory computing (IMC) are being explored
- Attributes such as synaptic efficacy and plasticity can be implemented in-memory by exploiting the physical attributes of memory devices such as phase-change memory
- Iso-accuracy DNN inference and training is possible with IMC
- Recently fabricated mixed-signal IMC cores demonstrate the promise of this technology
- Concepts such as projected memory and photonic in-memory computing could significantly improve the computational precision and performance
- The IMC approach could also impact applications that transcend conventional DL such as memory-augmented neural networks and spiking neural networks
Acknowledgements

- In-memory computing group, IBM Research – Europe
- Several other groups @ IBM Research – Europe
- New Jersey Institute of Technology, University of Patras, ETH Zürich, École polytechnique fédérale de Lausanne, RWTH Aachen, Oxford University, University of Münster
Analog AI Hardware Acceleration Toolkit

Current Capabilities Include:

- Simulate analog MVM operation including analog backward/update pass
- Simulate a wide range of analog AI devices and crossbar configurations by using abstract functional models of material characteristics with adjustable parameters
- Abstract device (update) models
- Analog friendly learning rule
- Hardware-aware training for inference capability
- Inference capability with drift and statistical (programming) noise models

Roadmap:

- Integration of more simulator features in the PyTorch interface
- Tools to improve inference accuracy by converting pre-trained models with hardware-aware training
- Algorithmic tools to improve training accuracy
- Additional analog neural network layers
- Additional analog optimizers
- Custom network architectures and dataset/model zoos
- Integration with the cloud
- Hardware demonstrators

https://analog-ai.mybluemix.net/