Neuromorphic Computing based Processors

Hao Jiang

A collaborative research among
San Francisco State University, EL-Lab at University of Pittsburgh, HP Labs, and AFRL

Outline

- Why Neuromorphic Computing?
- Challenges and New Opportunity
- Spiking Neuromorphic Design
- A Framework of Heterogeneous Computing Systems
- Conclusion
Why Neuromorphic Computing?

- Von Neumann arch. is facing severe challenges
  - Von Neumann bottleneck
  - Inefficient in cognitive computations
- Human brain: high efficiency
  - 100 TFLOPS vs. 20 Watt
  - Highly connected: 50B neurons & $10^{14}$ synapses
  - Very light: 3 lbs

Neuromorphic Design by Leveraging Memristor Technology

Brain – The Most Efficient Computing Machine

Brain:
- 15-30B neurons
- Extremely complex
- 4km/mm³
- 35w

Neuron:
- Process signals from other neurons.

Synapse:
- Memory
- Weight signals

Neocortex
- 6 layers
- Signals travel within and between layers

Brain: T h e M o s t E f f i c i e n t C o m p u t i n g M a c h i n e
Brain-like Neuromorphic Circuits

- Slow progress in neuromorphic hardware implementation
  - Lack of efficient synapse design
  - Not supportive to mass connection

Outline

- Why Neuromorphic Computing?
- Challenges and New Opportunity
- Spiking Neuromorphic Design
- A Framework of Heterogeneous Computing Systems
- Conclusion
Challenges in Traditional Approach

Developing and implementing neural network models on large scale computer clusters or supercomputers.

**Performance (100M MIPS) Challenge**

- Energy Challenge

  - > Megawatts
  - 1000 U.S. households

---

Traditional Analog Approach

**Weight**

- Implementation: floating gates, capacitor, etc.

**Compute**

- op-amps, analog voltage multipliers and differentiators

**Successful in small scale systems**

**Difficulties**

- Volatile data, low precision, control signal

**Scaling**

- O(N^2) for weight carrier

- O(N^2) for voltage multiplier

**Intrinsic, hard to overcome**

- Design complexity, power, and area grow very fast
Memristor – Rebirth of Analog Approach

Memristor

\[ M = R_L \cdot \alpha + R_H \cdot (1-\alpha) \]

Natural weight carriers:
- Non-volatility, high density
- Analog resistance states
- Two terminal programming

Memristor Crossbar

\[ I = V_{M1}/M_1 + V_{M2}/M_2 + \ldots + V_{Mn}/M_n \]

- Natural weight summation
- MIMO: avoid sneak path
- Cost \( \sim O(N) \), not \( O(N^2) \)
Memristor – Rebirth of Neuromorphic Circuits

Memristor ↔ Synapse
- Two terminal, high density
- Non-volatility
- Analog/multi-level states

Crossbar ↔ Network
- Natural matrix function
- A MIMO system
- Good combination with memristor

Outline
- Why Neuromorphic Computing?
- Challenges and New Opportunity
- Spiking Neuromorphic Design
- A Framework of Heterogeneous Computing Systems
- Conclusion
Spiking-based Neuromorphic Computing

- **Why spiking?**
  - Inspired by human brains
  - Minimized transition electrical charge
  - Reduced data communication distance
  - High parallelism in processing

- **Approaches in hardware system**
  - Analog and digital circuit blocks, capacitors
  - Crossbar array basing on SRAM, PCM, Memristor cell
  - In this work: Memristor based crossbar array as synapse

---

Spiking Neuromorphic System

- **Matrix computation transformation**

  \[
  \begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_m
  \end{bmatrix} =
  \begin{bmatrix}
  g_{11} & g_{12} & \cdots & g_{1n} \\
  g_{21} & g_{22} & \cdots & g_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  g_{m1} & g_{m2} & \cdots & g_{mn}
  \end{bmatrix}
  \begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
  \end{bmatrix}
  \]

  Mathematical matrix computation

  Memristor-based crossbar array for matrix computation

- **Spiking-based computation system**

  - Closer to biological system
  - Power efficiency
  - High reliability
Computation Methodology

Working mechanism:
- $V_m < V_{th} \rightarrow C_m$ integrates
  $V_m \geq V_{th} \rightarrow$ A spike is fired out,
  then $C_m$ is rest to 0
- Spike occurs: Weighted current to integrator
  No spike: No current to & from integrator

✓ Traditional integrate and fire model
✓ Rate coding

Memristor crossbar array structure

- **1S1M memristor-based cell**
  - Alleviate the impact of sneak path leakage
  - A thin-film based selector after each memristor
  - Minimal unit cell area of $4 F^2$

- **Selector property and operation**
  - Spike occurs: Selector $ON$, and $R_{s, on} < < R_M$
  - No spike: Selector $OFF$, and $R_{s, off} > > R_M$

\[ g_{ij} \mapsto \tilde{g}_{ij} = \frac{g_{ij} \cdot g_s}{g_{ij} + g_s} \]
High-speed Integrate and Fire Circuit (H-IFC)

\[ \tau = R_{mem} \times C_S \]

\[ \Delta t = \frac{V_{REF}C_s}{i_{in}} \]

- Low resistance
- High resistance

\[ v_{CS} = v_{in}(1 - e^{-\frac{t}{\tau}}) \approx v_{in} \frac{t}{\tau} = \frac{v_{in}t}{R_{mem}C_S} = i_{in} \frac{t}{C_S} \]

Integrate and Fire Circuit

- Structure and property
  - Integrate capacitor, Reset transistor, Comparator with positive feedback
  - High speed
  - \( V_{th} \) is much smaller than \( V_{dd} \)

- Power and area
  - \( \sim 100 \mu W \)
  - \( 28 \mu m \times 12 \mu m \)

(180 \( \mu m \) Technology)
System Verification

Output Pulse Number

\[ n_j = \frac{t_{ON} V_{DD}}{C_m V_{REF}} \sum_{i=1}^{N} g_{ij} m_j \]

- Pulse Duration
- Input Voltage
- Memristor Conductance
- Sensing Capacitance
- Comparator Trigging Voltage
- Output Pulse Number
- Selected Row
Theoretical Computation vs. Simulation Results

- **Real: Nonlinearity**
- **Sums of weighted signal dependent**

**Reasons:**
- Reset time
- IFC delay

**Optimization:**
- Larger $C_m$
- Higher speed of IFC

**Be used in neural network**

---

**Adaptability in Neuron Network**

**Output Spike Number ($n_j$):**
- a: 19
- b: 20
- c: 19

**Good adaptability in neural network**
Outline

• Why Neuromorphic Computing?
• Challenges and New Opportunity
• Spiking Neuromorphic Design
• A Framework of Heterogeneous Computing Systems
• Conclusion

Our Approach

• A framework of heterogeneous computing systems enhanced with *neuromorphic computing accelerators* (NCAs).

• **Purpose:** To combine the *flexibility* of conventional architecture in logic and scientific computation and the *efficiency* of neuromorphic architecture for ANN applications.
Frontend: Prepare Data & Instructions

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>set p reg</td>
<td>Configuration</td>
<td>Place the routing information stored at register reg to central router</td>
</tr>
<tr>
<td>movd #(reg)</td>
<td>I/O</td>
<td>Load the data from memory to NCA</td>
</tr>
<tr>
<td>launch</td>
<td>Configuration</td>
<td>Notify the central router to start transmitting</td>
</tr>
<tr>
<td>deq reg</td>
<td>I/O</td>
<td>Dequeue the head data of Out-queue and write it to register reg</td>
</tr>
</tbody>
</table>

Backend: System Design

- Tightly coupled design
- Invoked by special inst.
NCA Architecture

- A hierarchical structure of MBC arrays

[Diagram of hierarchical structure]

- Mixed signal NoC
  - Analog computation
  - Digital control and routing signal transition
  - MBC arrays connected in a metamorphous centralized mesh (MCMesh) manner

System Level Evaluation

- Two implementations representing tradeoffs between computation performance and accuracy
  - Multi-layer perception (MLP)
  - Auto-associative memory (AAM)

- 7 classification benchmarks
- Classification rate is used as reliability metric

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer</td>
<td>breast cancer diagnose</td>
</tr>
<tr>
<td>connect-4</td>
<td>connect-4 game</td>
</tr>
<tr>
<td>gene</td>
<td>nucleotide sequences detection</td>
</tr>
<tr>
<td>lymphography</td>
<td>lymph diagnose</td>
</tr>
<tr>
<td>MNIST</td>
<td>digit recognition</td>
</tr>
<tr>
<td>mushroom</td>
<td>poisonous mushroom discrimination</td>
</tr>
<tr>
<td>thyroid</td>
<td>thyroid diagnose</td>
</tr>
</tbody>
</table>
Experimental Setup

The Design Parameters of NCA Components

<table>
<thead>
<tr>
<th>Component</th>
<th>R_L (Ω)</th>
<th>R_Ω (KΩ)</th>
<th>V_{IN} (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memristor</td>
<td>200 KΩ</td>
<td>160 MΩ</td>
<td>2 V</td>
</tr>
</tbody>
</table>

MBC Array & M-Net

<table>
<thead>
<tr>
<th>Component</th>
<th>Gain</th>
<th>Network</th>
<th>Sigmoid</th>
<th>MBC</th>
<th>DAC</th>
<th>ADC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>100 µW</td>
<td>0.72 mW</td>
<td>100 µW</td>
<td>0.09 µW</td>
<td>5.2 mW</td>
<td>3.8 mW</td>
</tr>
<tr>
<td>Speed</td>
<td>0.6 ms</td>
<td>4.2 ms</td>
<td>0.24 ns</td>
<td>3 ns</td>
<td>333 MHz</td>
<td>333 MHz</td>
</tr>
</tbody>
</table>

Area Estimation

<table>
<thead>
<tr>
<th>Component</th>
<th>NoC (mm²)</th>
<th>NoC (mm²)</th>
<th>MBC/ADC (mm²)</th>
<th>MBC (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCA area</td>
<td>0.943 mm²</td>
<td>0.598 mm²</td>
<td>0.014 mm²</td>
<td>0.252 mm²</td>
</tr>
<tr>
<td>D-Net</td>
<td>1.793 mm²</td>
<td>0.268 mm²</td>
<td>0.065 mm²</td>
<td>0.301 mm²</td>
</tr>
<tr>
<td>M-Net</td>
<td>0.943 mm²</td>
<td>0.598 mm²</td>
<td>0.014 mm²</td>
<td>0.252 mm²</td>
</tr>
</tbody>
</table>

The Benchmark Implementation Details

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Description</th>
<th>Training error</th>
<th>Topology</th>
<th>MBC array usage</th>
<th>Training error</th>
<th>MBC array usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer</td>
<td>breast cancer diagnosis</td>
<td>0.02%</td>
<td>16+16+16</td>
<td>2 arrays in 1 group</td>
<td>0.07%</td>
<td>2 arrays in 1 group</td>
</tr>
<tr>
<td>connect-4</td>
<td>connect-4 game</td>
<td>0.02%</td>
<td>42+30+15</td>
<td>2 arrays in 1 group</td>
<td>0.08%</td>
<td>3 arrays in 1 group</td>
</tr>
<tr>
<td>yeast</td>
<td>nucleosome sequences detection</td>
<td>0.00%</td>
<td>150+100+3</td>
<td>6 arrays in 3 groups</td>
<td>0.03%</td>
<td>12 arrays in 3 groups</td>
</tr>
<tr>
<td>leuography</td>
<td>leuography</td>
<td>0.05%</td>
<td>29+19+4</td>
<td>2 arrays in 1 group</td>
<td>0.02%</td>
<td>4 arrays in 1 group</td>
</tr>
<tr>
<td>MNIST</td>
<td>digit recognition</td>
<td>0.25%</td>
<td>64+128+32+16</td>
<td>3 arrays in 3 groups</td>
<td>0.02%</td>
<td>10 arrays in 3 groups</td>
</tr>
<tr>
<td>mushroom</td>
<td>poisonous mushroom discrimination</td>
<td>0.01%</td>
<td>125+32+2</td>
<td>3 arrays in 1 group</td>
<td>0.01%</td>
<td>8 arrays in 2 groups</td>
</tr>
<tr>
<td>thyroid</td>
<td>thyroid diagnosis</td>
<td>0.15%</td>
<td>21+32+3</td>
<td>2 arrays in 1 group</td>
<td>0.11%</td>
<td>3 arrays in 1 group</td>
</tr>
</tbody>
</table>

Impact of Deficient Hardware

- **Programming precision** due to limited device resolution
- **Device variations and signal fluctuations**
- **AAM is more robust than MLP**
MBC Size Exploration

- Larger array size is preferable from performance perspective
- However, as array size increases the classification rate degrades induced by the aggravated variations

Comparison w/ Other Designs

- **Baseline**: CPU as general purpose processor
- **D-NPU**: a popular digital neuromorphic accelerator (MICRO’12)
- **MBCs+D-Net**: MBC arrays w/ digital NoC in order to evaluate the efficiency of mixed-signal NoC
- **NCA**: our design w/ MBC arrays and mixed-signal NoC
Comparison w/ Other Designs

NCA Improvement

<table>
<thead>
<tr>
<th></th>
<th>MLP</th>
<th>AAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speedup</td>
<td>177.67</td>
<td>27.20</td>
</tr>
<tr>
<td>Energy Saving</td>
<td>184.71</td>
<td>25.18</td>
</tr>
</tbody>
</table>

**D-NPU** is limited by the computational bandwidth.

**MBCs+D-Net** is limited by the costly AD/DA conversions.

Outline

- Why Neuromorphic Computing?
- Challenges and New Opportunity
- Spiking Neuromorphic Design
- A Framework of Heterogeneous Computing Systems
- Conclusion
Conclusion & Perspective

- Invention of new devices inspires the study of the next generation neuromorphic computing systems.
- A spiking neuromorphic computing system by leveraging the memristor crossbar array is demonstrated.
- We propose a heterogeneous system that combines the flexibility of conventional architecture and the efficiency of neuromorphic architecture.
- In the future research, we plan to extend the investigation to larger scale ANN applications.
- The techniques to enhance the run-time robustness in training and testing procedures will be studied.

Thank You and Questions?