Brain Inspired Computing:
The Extraordinary Voyages in Known and Unknown Worlds

Yiran Chen
EI-Lab (www.ei-lab.org)
Electrical and Computer Engineering
University of Pittsburgh
Evolution of Computing Power

The Age of Intelligent Machines
MIT Press, 1990
ISBN 0-262-11121-7

Dr. Raymond Kurzweil

Million instructions per second (MIPS)

1900 1920 1940 1960 1980 2000 2020

Human
100G Neurons

Monkey
3G Neurons

Lizard
2M Neurons

Worm
300 Neurons

Bacteria
1 “Neuron”

“A sufficiently advanced computer program could exhibit human-level intelligence”
**Artificial Brain – Intelligent, Creative, Self-aware**

**Google Brain Simulator (2012)**
- Unsupervised training
- Deep learning
- 16,000 processors
- 1B connections
- 10M YouTube videos

**IBM Watson (2011)**
- Defeated Humanity in “Jeopardy”
- 9032 core IBM servers
- 16TB memory

**In Developing (2014)**
- Unsupervised learning
- Largest cluster for deep learning
- 100B neural connections
- Heavy cluster of GPUs

**K Computer (2014)**
- Human brain activity
- The 4th most powerful computer in the world
- 40 minutes of simulation → 1-second of brain activity
- 700,000 processor cores and 1.4M GB RAM
What Are The Major Limitations?

- Stalled single-thread performance
- Limited data throughput
- Constrained power efficiency

“Taming the Power Hungry Data Center” by Fusion-IO.

D. Hammerstrom, Neucomp, 2013
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

A group of Pre-neurons

\[ (G_{N,M})^T \]

A group of Post-neurons

Synaptic network
Summary: Today, the White House is announcing a grand challenge to develop transformational computing capabilities by combining innovations in multiple scientific disciplines.
Neuromorphic Cognitive Computation

IBM
TrueNorth
SRAM synapse
Digital spike
1M neurons/chip
256M synapse/chip
J. Hsu, IEEE Spectrum, 2014

Stanford
Brain in Silicon
Mixed-signal VLSI
1M neurons/16 chips
1B synapse/16 chips
B. Benjamin, Neurogrid, 2014

Qualcomm
Zeroth
Custom hybrid
Spike neurons on chip
Synapse off chip
J. Gehlhaar, ASPLOS, 2014

Micron
Automata
Massively parallel
Memory driven
Non-von Neumann
XML-based language
F. Samarrai, UVAToday, 2014

A Brain Inspired Solution

HBP
BrainScaleS
Analog VLSI
64 neurons/chip
1024 synapses/chip
S. Miller, ESANN, 2012

J. Hsu, IEEE Spectrum, 2014
B. Benjamin, Neurogrid, 2014
J. Gehlhaar, ASPLOS, 2014
F. Samarrai, UVAToday, 2014
From Known to Unknown Worlds

- Voyages Extraordinaires — Voyages dans les mondes connus et inconnus
- *The Extraordinary Voyages in Known and Unknown Worlds* — *Jules Verne*
- 1863: Five Weeks in a Balloon (in Africa)
- 1863: *Paris in the 20th Century*
- 1864: Journey to the Center of the Earth
- 1865: From the Earth to the Moon

Known worlds

Unknown worlds
Outline

- Motivation for Brain Inspired Computing
- Research Spotlights
  - Recurrent Neural Network Based Language Model @ FPGA
  - An Efficient Learning Method of TrueNorth Chip
  - Memristor-based Neuromorphic Computing Engine
- Frequent Q&A about Neuromorphic Computing
- Conclusion
Outline

• Motivation for Brain Inspired Computing
• Research Spotlights
  – Recurrent Neural Network Based Language Model @ FPGA
  – An Efficient Learning Method of TrueNorth Chip
  – Memristor-based Neuromorphic Computing Engine
• Frequent Q&A about Neuromorphic Computing
• Conclusion
Context Aware Intelligent Text Recognition

...but beginning to perceive that the handcuffs were not for me and that the military had so far got....

Perception based on neural network models

BSB Recognition

Prediction

Word Level Confabulation

Knowledge Base (KB)

but beginning to perceive that the handcuffs were not for me and that the military had so far got....

Prediction

Sentence Level Confabulation

Knowledge Base (KB)
Sentence Confabulation on FPGA

I saw __ dog.

A. a    B. an    or

Statistics based language model

\[
\begin{align*}
P(I \text{ saw a dog}) &= P(I) \times P(\text{saw}| I) \times P(a | I \text{ saw}) \times P(\text{dog}| I \text{ saw a}) \\
P(I \text{ saw a dog.}) &> P(I \text{ saw an dog.})
\end{align*}
\]

Complexity: \(V^n\)

- Vocabulary size (10~60K)
- Sentence length

- Knowledge base is large and sparse
- Only the short-term perspective of a sequence.

Neural Network based language model

- Record the long-term historical information
- Stronger learning ability

RNNLM – Training

Feed forward

\[ h(t) = f(W_{ih} \cdot x(t) + W_{hh} \cdot h(t - 1) + b_h) \]
\[ y(t) = g(W_{ho} \cdot h(t) + b_o) \]

Activation functions

\[ f(x^i) = \frac{1}{1+e^{-x_i}} \]
\[ g(x^i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \]

Back propagation through time (BPTT)

\[ \delta_p(t) = t_p(t) - o_p(t) \]

Weigh update

\[ W_{hi} \leftarrow W_{hi} + \eta \sum_{t=1}^{B} \delta_j(t) \cdot x_i(t) \]

- High computational cost
  \(~10^7\) parameters between hidden/output layer
  10~20 epochs for convergence

- Computation resource utilization

- Memory efficiency

T. Mikolov, et al, ASRU, 2011
Extend Inherent Parallelism of RNNLM

Pipeline stages are not balanced

- Hidden layer: $O(H \times H)$
- Output layer: $O(H \times V)$

Output layer is more critical

- Vocabulary size: $V = 10K$
- Hidden layer size: $H = 0.1K$
- BPTT time step: $B = 4$

Speed-up (pipeline) = 1.07×
Speed-up (parallel) = 3.86×
System Overview

- **Inherent parallelism**
- **Data format conversion**
- **Approximation of activation functions**

**Memory Efficiency**
- Multi-thread management unit (TMU)
- Extensive data reuse

**Resource Utilization**
- Inherent parallelism
- Data format conversion
- Approximation of activation functions

**Design Scalability**
- Customized processing element (PE)
Microsoft Research Sentence Completion (MRSC)

1. I have seen it on him, and could _____ to it.
   (a) write       (b) migrate         (c) climb
   (d) swear      (e) contribute

2. …

Training corpus:  
19th and 20th Century novels (38M)

Vocabulary size: 10,583

Hidden layer size: 1024

BPTT: 4
## Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>20%</td>
</tr>
<tr>
<td>Smoothed 3-gram *</td>
<td>37%</td>
</tr>
<tr>
<td>RNN-100 with 100 classes</td>
<td>40%</td>
</tr>
<tr>
<td>RNNLM (this work)</td>
<td>46.2%</td>
</tr>
<tr>
<td>vLBL+NCE5 *</td>
<td>60.8%</td>
</tr>
<tr>
<td>Human *</td>
<td>91%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Intel Xeon E5-2630</th>
<th>Nvidia GeForce GTX580</th>
<th>Convey HC-2ex (CPU+FPGA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core (#)</td>
<td>12</td>
<td>512</td>
<td>8*4</td>
</tr>
<tr>
<td>Clock</td>
<td>2.3 GHz</td>
<td>772 MHz</td>
<td>150 MHz</td>
</tr>
<tr>
<td>Memory BW (GB/s)</td>
<td>42.6 GB/s</td>
<td>192.4 GB/s</td>
<td>76.8 GB/s</td>
</tr>
<tr>
<td>Runtime (s)</td>
<td>2,566.80 (1 ×)</td>
<td>130.01 (19.7 ×)</td>
<td>160.72 (16.0 ×)</td>
</tr>
<tr>
<td>Power-TDP (W)</td>
<td>95</td>
<td>244</td>
<td>25</td>
</tr>
<tr>
<td>Energy (J)</td>
<td>243,846 (1 ×)</td>
<td>31,722 (7.7 ×)</td>
<td>4,018 (60.7 ×)</td>
</tr>
</tbody>
</table>
Outline

• Motivation for Brain Inspired Computing
• Research Spotlights
  – Recurrent Neural Network Based Language Model @ FPGA
  – An Efficient Learning Method of TrueNorth Chip
  – Memristor-based Neuromorphic Computing Engine
• Frequent Q&A about Neuromorphic Computing
• Conclusion
A New Learning Method on TrueNorth Chip

TrueNorth Configuration:
- A network of 4000 neuro-synaptic cores each of which includes a $256 \times 256$ configurable synaptic crossbar
- Inter-core communication by spike events
- 4 possible integer weights at each synapse

$$y = \mathbf{w} \cdot \mathbf{x} + b$$
$$z = h(y)$$

McCulloch-Pitts neuron model

$$y' = \mathbf{w}' \cdot \mathbf{x}' - \lambda$$
$$z' = \begin{cases} 
1, & \text{reset } y' = 0; \text{ if } y' \geq 0 \\
0, & \text{reset } y' = 0; \text{ if } y' < 0
\end{cases}$$

Figure 1. Mapping neural networks in IBM TrueNorth.
A New Learning Method on TrueNorth Chip

Learn in traditional floating-point precision
Deploy in binary/low integer precision sampled by float probability

connectivity probabilities $p$
connectivity samples

(a) Tea learning
(b) Tea deploying

$P(x_i = 1) = x_i$
$P(x_i = 0) = 1 - x_i$

$y' = \sum_{i=0}^{n-1} w_i x_i$

$E\{y'\} = E\{ \sum_{i=0}^{n-1} w_i x_i \} = \sum_{i=0}^{n-1} E\{ w_i \} E\{ x_i \}$

$E\{z'\} = P(y' \geq 0) = \frac{1}{2} \left[ 1 + \text{erf} \left( \frac{-y}{\sqrt{2}\sigma_y} \right) \right]$

Works as a nonlinear activation function of input spikes parameterized by trainable probabilities/w

95.27% in Caffe
90.04% @1 copy in TN
94.63% @16 copies in TN

Figure 2. An overview of the learning and deploying of a neural network on TrueNorth.
A New Learning Method on TrueNorth Chip

Analysis on accuracy loss because of low quantitation precision

\[ \Delta y = y' - y = \sum_{i=0}^{n-1} w_i x_i' - \sum_{i=0}^{n-1} w_i x_i \]

\[ E \{ \Delta y \} = 0 \]

\[ var \{ \Delta y \} = \sum_{i=0}^{n-1} var \{ w_i x_i' \} \]

unbiased approximation with variance affected by both synaptic randomness and spiking randomness.

Zero synaptic deviation at two poles

\[ var \{ w_i' \} = E \{ (w_i')^2 \} - E \{ w_i' \}^2 = c_i^2 p_i (1 - p_i) \]
A New Learning Method on TrueNorth Chip

Probability-biased learning to bias synaptic connectivity to deterministic poles:

Minimization target: \( \hat{E}(w) = E_D(w) + \lambda \cdot E_W(w) \)

Log loss

L1 norm:

\[
E_W(w) = \|w\| = \sum_{k=1}^{M} |w_k| \quad \text{Enforcing elements in } w \text{ to zeros}
\]

Proposed biasing method:

\[
E_b(w) = \| |w - a| - b \| = \sum_{k=1}^{M} \left| w_k - a \right| - b
\]

Probability (weight) distribution under different penalties.
A New Learning Method on TrueNorth Chip

Test benches

<table>
<thead>
<tr>
<th>Test bench</th>
<th>Dataset</th>
<th>Block stride</th>
<th>Hidden layer #</th>
<th>Cores per layer</th>
<th>Accuracy in Caffe</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MNIST</td>
<td>12</td>
<td>1</td>
<td>4</td>
<td>95.27%</td>
</tr>
<tr>
<td>2</td>
<td>MNIST</td>
<td>4</td>
<td>1</td>
<td>16</td>
<td>96.71%</td>
</tr>
<tr>
<td>3</td>
<td>MNIST</td>
<td>2</td>
<td>3</td>
<td>49<del>9</del>4</td>
<td>97.05%</td>
</tr>
<tr>
<td>4</td>
<td>RS130</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>69.09%</td>
</tr>
<tr>
<td>5</td>
<td>RS130</td>
<td>1</td>
<td>2</td>
<td>16~9*</td>
<td>69.65%</td>
</tr>
</tbody>
</table>

* 16 and 9 correspond to the cores utilized by 1st and 2nd hidden layer.

Table 2. Core occupation and performance efficiency of probability-biased learning method

(a) Core occupation efficiency (1 spf)

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>0.904</th>
<th>0.924</th>
<th>0.929</th>
<th>0.935</th>
<th>0.938</th>
<th>0.939</th>
<th>0.942</th>
<th>0.942</th>
<th>0.943</th>
<th>0.944</th>
<th>0.945</th>
<th>0.946</th>
<th>0.947</th>
<th>0.947</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Copies</td>
<td>N1^1</td>
<td>N2</td>
<td>B1</td>
<td>N3</td>
<td>B2</td>
<td>N4</td>
<td>N5</td>
<td>B3</td>
<td>N7</td>
<td>N9</td>
<td>B4</td>
<td>N10</td>
<td>N16</td>
<td>B5</td>
</tr>
<tr>
<td>Saved Core</td>
<td>-</td>
<td>4 (50.0%)</td>
<td>4 (33.3%)</td>
<td>-</td>
<td>8 (40.0%)</td>
<td>-</td>
<td>20 (55.6%)</td>
<td>-</td>
<td>44 (68.8%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) Performance efficiency (1 network copy)

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>0.904</th>
<th>0.920</th>
<th>0.927</th>
<th>0.928</th>
<th>0.929</th>
<th>0.932</th>
<th>0.933</th>
<th>0.934</th>
<th>0.940</th>
<th>0.943</th>
<th>0.946</th>
<th>0.947</th>
<th>0.948</th>
<th>0.950</th>
</tr>
</thead>
<tbody>
<tr>
<td>spf</td>
<td>N1^1</td>
<td>N2</td>
<td>N3</td>
<td>N6</td>
<td>B1</td>
<td>N7</td>
<td>N11</td>
<td>N13</td>
<td>B2</td>
<td>B3</td>
<td>B4</td>
<td>B5</td>
<td>B9</td>
<td>B13</td>
</tr>
<tr>
<td>Speedup</td>
<td>6</td>
<td>6.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

^TrueNorth neural networks are denoted by N# and B#, where N/B indicates the model is learned without penalty (None)/Biasing penalty. # is the number of network copies in (a) and spf in (b) respectively.
Outline

• Motivation for Brain Inspired Computing
• Research Spotlights
  – Recurrent Neural Network Based Language Model @ FPGA
  – An Efficient Learning Method of TrueNorth Chip
  – Memristor-based Neuromorphic Computing Engine
• Frequent Q&A about Neuromorphic Computing
• Conclusion
Memristor – Rebirth of Neuromorphic Circuits

Memristor

Synapse Network

Memristor Crossbar

Programmable resistor w/ analog states

Natural matrix operation

\[ [x_1 \ x_2 \ \ldots \ x_m] \]

\[ y_1 = \Sigma x_i \cdot g_{il} \]

High density

DAC’12

APL’13
Brain-State-in-a-Box (BSB)

BSB Training Process

\[ \Delta A = lr \ast (X - AX) \otimes X \]

\[ A = A + \Delta A \]

- \( X \) is the normalized input training vector;
- \( lr \) is the “Learning Rate”;
- \( \otimes \) is the outer product of two vectors;

BSB Recall Process

\[ X(t + 1) = S(\alpha \cdot A \cdot X(t) + \lambda \cdot X(t)) \]

- \( X(t+1) \) and \( X(t) \) are \( N \) dimensional real vectors;
- \( X(0) \) is the input pattern (vector);
- \( A \) is the \( N \times N \) connection matrix (memory);
- \( S() \) is a linear output-limiting function

BSB recall convergence criteria

\[ X(t + 1) = X(t) \]
BSB Circuit: Recall Only

The recall function: \( \mathbf{x}(t + 1) = S(\alpha \cdot \mathbf{A} \times \mathbf{x}(t) + \lambda \cdot \mathbf{x}(t)) \)

Comparers detect the converge status.

Next iteration

Input vector \( V(0) \)

\( V_{A+}(t) = A^+ V(t) \)

\( V_{A-}(t) = A^- V(t) \)

Summing op-amps perform analog voltage signal addition/subtraction

We need two memristor arrays since memristor can only represent positive weights.
BSB Circuit: Training

- Original Delta rule: \[ \Delta w_{ij} = \alpha \cdot (t_j - y_j) \cdot x_i \]
- Modification for hardware implementation:
  \[ \Delta g_{ij} \propto V_t \cdot T_t \cdot \text{Sign}(V_{ref,j} - V_{out,j}) \cdot \text{Sign}(V_{in,i}) \]

Minimize the design complexity meanwhile ensuring the weight change in the same direction as that of the Delta rule

\( x_i \): the \( i \)th entry of input
\( t_j \): the \( j \)th entry of target output
\( y_j \): the \( j \)th entry of net output
\( V_t \): tuning pulse amplitude
\( T_t \): tuning pulse period

El-lab, DAC'12
Racing-BSB Model for Pattern Recognition

- **BSB Model**
  - Simple
  - Good noise resistibility
  - High correlation between convergence speed and pattern similarity

Each BSB model remembers one pattern (and its variations)

Recall against ALL BSB models

Multiple matching patterns for each image.

EI-lab, DAC'12
Racing-BSB Model for Pattern Recognition

- **BSB Model**
  - Simple
  - Good noise resistibility
  - High correlation between convergence speed and pattern similarity

Successfully adopted in various applications

EI-Lab, DAC'13

EI-lab, DAC'12
Two Design Approaches

- **Level-based Design**
  - Compatible to existing signal processing
  - High speed computation
  - High design complexity

- **Spike-based Design**
  - Closer to biological system
  - Extremely high power efficiency
  - Slow operation

EI-lab, DAC’15
Single-layer Design: Block Diagram

**Input signal generator**

- **IPNT<31:0>**

**Driver**

- **WL0**, **WL1**, **WLj**, **WL31**

**Block Diagram**

- **g_{i,j}**

**Timing & Control**

- **EN**, **GCLK**, **GRSTB**

- **I4PN<2:0>**

**Counter**

- **Cnt<11:0>**

**Subtracter**

- **SBT**

**Comparator**

- **CMP**

**ICLK**

**IFC**

- **Vm<11:0>**

**CLR4CLK1**

**RSTB4CNT**

- **OCMP<5:0>**

**El-lab, DAC’15**
Implementation

- **Process:** GlobalFoundries 130nm
- **Core:** 2.5×2.5mm²
- **After Packaging:** 1.2×1.2cm²
- **Clock Frequency:** 25MHz
- **Recognition Frequency:** 2MHz
- **Crossbar Size:**
  - FW: 32×12
  - ML: 64×40, 20×20
- **Input Resolutions:**
  - FW: 32bits (8×4 Image)
  - ML: 64bits (8×8 Image)
- **Capacity:**
  - FW: “0” ~ “5”, “EI-LAB”
  - ML: “0” ~ “9”
Neuromorphic Computing Acceleration (NCA)

NCA Hardware

- Digital
- Analog

From CPU

Input data

Control

Controller

Buffer

MBC groups

DAC

ADC

Out Queue

To CPU

Output data

Memristor-based crossbar (MBC)

NCA Software

bool Recall(float *vec, float *wm)
{
    /* simulate the synapse network */
    for(i=0;i<BsbSize;++i) wx[i] +=
        wm[i*BsbSize+j] * vec[j];
        ......

Find the candidate codes

Source-to-source translation

bool Recall(float *vec)
{
    Send(NCA.id, vec);
    return Receive(NCA.id);
    ......

The neural topology

NCA-aware compilation

MOV NCA.id, R1
......
SET NCA.id, #VAL
LAUNCH
DEQ R1, NCA.id

The NCA-aware executable

EI-lab, DAC’15

EI-lab, DAC’15

EI-lab, DAC’15

EI-lab, DAC’15
Compare to Other Designs

Example: Multilayer Perception (MLP)

Seven representative learning benchmarks.
All the results are normalized to the baseline CPU.

Digital NPU + Digital NoC [1]
MBC + Digital NoC
NCA (MBC + Mixed-signal NoC)

[1] H. Esmaeilzadeh et al., MICRO’12
Outline

• Motivation for Brain Inspired Computing
• Research Spotlights
• Frequent Q&A about Neuromorphic Computing
• Conclusion
Challenges in Brain Inspired Computing

- Do we really need to fully understand the human brain before designing a useful Neuromorphic Computing system?
  - Unfortunately we still don’t know much about human brains.
  - The answer is “NO”, which has been proven.

- What will be the most efficient, robust and scalable platform & computation format?
  - Depends.

- Are conventional CMOS Technologies and Design Methodologies capable of supporting long-term research and development of Neuromorphic Systems?
  - Absolutely NO.
Outline

• Motivation for Brain Inspired Computing
• Research Spotlights
• Frequent Q&A about Neuromorphic Computing
• Conclusion
Conclusion

- Non-conventional hardware architectures become critical for cognitive applications.
- A holistic scheme integrating the efforts on device, circuit, architecture, System, algorithm, etc. is necessary.
- There are many challenges and opportunities in device-circuit-architecture-system co-designs.

“I imagine a world where the difference between man and machine blurs, where the difference between humanity and technology fades, where the soul and silicon chip unite.”

Raymond Kurzweil

*The Age of Intelligent Machines*
Evolutionary Intelligence Lab (EI-Lab)

Our objectives are to enhance conventional systems and to explore new computing diagram by leveraging emerging technologies.

http://www.ei-lab.org
Sponsors
Thanks to EI-LAB members
(Xiaoxiao Liu, Chenchen Liu, Sicheng Li, Mengjie Mao, Xue Wang, Chunpeng Wu, Chaofei Yang, et al.)