Toward More Efficient Acceleration of Deep Neural Networks Using Stochastic Computing

Jongeun Lee, UNIST, Korea
Motivation

[Hasler 2013]
Deep Learning

Map successful deep-neural-nets onto efficient digital hardware with least overall accuracy loss

Analog NN, BNN, SC-DNN, ...

Neuromorphic (SNN, etc)

Mimic bio-neural-nets for their efficiency and graceful learning
### Stochastic Computing

#### Number representation

- **0.75**
  - 10101111 (6/8)
  - 01111011 (6/8)
  - 11001111 (6/8)

- **0.11000000**

#### Arithmetic operations in SC

- **Multiplication example:**
  - $P(X \land Y) = P(X) \cdot P(Y) = x \cdot y$
  - Assumption: $X$ and $Y$ are uncorrelated

Conventional binary

$$x = \sum_i 2^i \cdot X_i$$

Stochastic computing

$$x = \text{Prob}(X = 1) = E[X]$$
Stochastic Computing

\[ P = A \times B \]

**Binary Multiplier**

- \( A \) \( B \)
- \( P = A \times B \)

**SC Multiplier**

- \( 011010 \) \( A \) \( 001010 \) \( P = A \times B \)
- \( 101110 \) \( B \) \( 001010 \)

- **bit-serial version**

- **bit-parallel version**

- \( a_0 \) \( b_0 \) \( p_0 \)
- \( a_1 \) \( b_1 \) \( p_1 \)
- \( \ldots \)
- \( a_k \) \( b_k \) \( p_k \)
Why SC?

- **Advantages**
  - Fault-tolerance
  - Very low-cost implementation
  - “Anytime” computing

- **Disadvantages**
  - Inaccuracy due to randomness
  - Exponential latency
  - Inefficient storage

- **SC applications**
  - Insensitive to small random errors
  - Low-power or high number of computation
  - e.g., low-power image processing, neural networks
SC-based DNN

- **Design challenges for SC-based DNN**
  - Random fluctuation of SC neuron is highest around zero
  - We remove near-zero weights, and rescale weight parameters
**Proposed SC neuron**

- Single neuron consists of multipliers, parallel counter, and up/down counter
- Up/down counter integrated with parallel counter continuously generates SC bitstream
- Enables unbuffered, cascaded design through all layers
Early Decision Termination

- **MNIST evaluation results**
  - High accuracy: achieves near floating-point accuracy
  - Early decision termination: can both reduce energy and improve decision speed

Recognition error comparison

- Recognition error vs. SC precision

“Typical” SC-based DNN

- End-to-end SC design
  - Low conversion overhead
  - Early decision termination

- 

<table>
<thead>
<tr>
<th>SC Multiplication</th>
<th>SC Addition</th>
<th>SC Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(AND/XNOR)</td>
<td>(MUX/Adder)</td>
<td>(FSM/Counter)</td>
</tr>
</tbody>
</table>

- Fully-parallel design

SC IN

SC OUT

AND

MUX

AND

SC-Neuron

FSM
More Scalable Approach

- **Departure from Fully-Parallel Design**
  - Externally Digital, Internally SC
  - Computation in stochastic computing, but storage in conventional binary
  - 100% compatible memory interface with conventional neural network accelerators
  - High scalability, ease of validation

- **Challenge**
  - High conversion overhead
  - Loss of EDT (Early Decision Termination)
Scalable & Reconfigurable SC-DNN

- Scalable & reconfigurable SC-DNN architecture
  - Scalability: How to support very large DNN models
  - Reconfigurability: How to support different DNNs
  - Scalability & reconfigurability \(\rightarrow\) binary-interfaced SC
  - Key features: Hybrid-across-layers, Hybrid-within-layer, approximate adder
    \(\rightarrow\) Best accuracy for SC-DNN & More energy-efficient than conventional binary
Accuracy and Energy Improvement

MNIST rec. rate

Normalized area-delay product

\( T_R \times T_C \times T_M : 8x8x8 \)
Taxonomy of SC-NN

- **SC-NN**
  - End-to-end SC
    - Potentially higher efficiency
  - Partially SC (Externally Digital, Internally SC)
    - Scalability & reconfigurability
  - Inference only
    - (eg., [Kim et al, DAC 2016])
  - Training HW
    - (eg., [Brown & Card, ToC 2001])
  - Inference only
    - (eg., [Sim et al, ASP-DAC 2017])
  - Training HW
    - (eg., [Liu et al., CASES 2018])
Need for Improving SC Multiply

High variance of SC multiplication result

More Efficient
More Accurate
Repeatability
New SC Multiply – More Efficient

Conventional binary number

Conventional binary number

6/16 → 0110₂
10/16 → 1010₂

x

SNG

w

SNG

AND

Counter

6/16 → 0110₂
10/16 → 1010₂

x

SNG

w

SNG’

AND

Counter

6/16 → 0110₂
10/16 → 1010₂

x

SNG

if (counter==0) done=1

w

Down-counter

 initialized to w·16

sort by W-bits

x·w

0100₂ → 4/16
≈ (6/16)·(10/16) = 3.75/16

x·w

0100₂ → 4/16
≈ (6/16)·(10/16) = 3.75/16

x·w

0100₂ → 4/16
≈ (6/16)·(10/16) = 3.75/16
New SC Multiply – More Accurate

Discrepancy
- Low discrepancy: 0010100101001010 (GOOD)
- High discrepancy: 0000000000111111 (WORST)
- High discrepancy: 1111110000000000 (WORST)

Optimal sequence
\[
\begin{array}{cccccccccccc}
  & x_3 & x_2 & x_3 & x_1 & x_3 & x_2 & x_3 & x_0 & x_3 & x_2 & x_1 & x_3 & x_2 & x_3 \\
\end{array}
\]
New SC Multiply – More Accurate

- Low-discrepancy SC bitstream
- Very low-cost
  - Requires a MUX and an FSM only
New SC Multiply: Efficient & Accurate

Improved Accuracy of SC-Mult

- LFSR
- Halton [1]
- ED [8]
- Proposed
- Proposed (MAX)

(a) Std. dev. (5-bit)
(b) Mean (5-bit)
(c) Std. dev. (10-bit)
(d) Mean (10-bit)

x-axis: \( N \) binary bits or \( 2^N \) stochastic bits

Improved Accuracy of SC-DNN

- Floating-point
- Fixed-point
- Conv. SC
- Proposed

(a) MNIST, inference only
(b) MNIST, after fine-tuning
(c) CIFAR-10, inference only
(d) CIFAR-10, after fine-tuning
SC Opportunity: Dynamic Precision Scaling

DVFS (Dynamic Volt-Freq Scaling)

DPS (Dynamic Precision Scaling)

Intel Multi-core revolution (c. 2004)

E = \frac{1}{2} CV^2
DPS SC-DNN: Scalable & Efficient

Scalable to Large DNNs

<table>
<thead>
<tr>
<th>Model</th>
<th>CNN precision</th>
<th>DPS precision</th>
<th>DPS precisions found</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>0.9904</td>
<td>0.9826</td>
<td>5 (uniform)</td>
</tr>
<tr>
<td>AlexNet (top-5)</td>
<td>0.807</td>
<td>0.7999</td>
<td>10-9-8-9-9</td>
</tr>
<tr>
<td>GoogLeNet (top-5)</td>
<td>0.8926</td>
<td>0.8844</td>
<td>13 (uniform, w/o fine-tune)</td>
</tr>
<tr>
<td>VGG_S (top-5)</td>
<td>0.8341</td>
<td>0.8247</td>
<td>9-9-10-9-10</td>
</tr>
</tbody>
</table>

Feature Comparison Table

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Large (&gt; 5 Conv. layers) DNNs</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>Tile based</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Per-DNN precision</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>Per-layer precision</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Per-bit precision</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Multi-bit acceleration</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Variable latency operation</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Half-range specialization</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
</tbody>
</table>
FPGA Prototyping of SC-DNN

Prototyping System Demo

System Block Diagram

FPGA Synthesis Result

<table>
<thead>
<tr>
<th>Case</th>
<th>P</th>
<th>D</th>
<th>LUT</th>
<th>FF</th>
<th>BRAMs</th>
<th>DSP</th>
<th>Freq. (target)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>4</td>
<td>6</td>
<td>1664</td>
<td>2005</td>
<td>4</td>
<td>0</td>
<td>100 MHz</td>
</tr>
<tr>
<td>SC-1</td>
<td>4</td>
<td>6</td>
<td>1281</td>
<td>1896</td>
<td>4</td>
<td>0</td>
<td>100 MHz</td>
</tr>
<tr>
<td>SC-2</td>
<td>16</td>
<td>24</td>
<td>3055</td>
<td>8803</td>
<td>4</td>
<td>0</td>
<td>100 MHz</td>
</tr>
<tr>
<td>SC-3</td>
<td>64</td>
<td>24</td>
<td>8808</td>
<td>30602</td>
<td>4</td>
<td>0</td>
<td>100 MHz</td>
</tr>
</tbody>
</table>
Log-SC: Motivation

- Previous SCs are limited with linearly quantized fixed-point data flow.
- With bit-parallelism of SC, 1s counter becomes the largest overhead, even for state-of-the-art (SotA) SC-MAC.
Log-SC

- Storing weights as power of 2 greatly helps simplify 1s counter
- New overhead of log-to-linear converter can be shared by multiple MAC units
Toward More Efficient Acceleration of DNNs Using SC

(a) 10-class image classification

(b) 1,000-class image classification

Area (um²)

Power (mW)

Avg. MAC cycles

Area (mm²)
Conclusion

- **Two regimes for SC-based DNN acceleration**
  - Fully-parallel architecture: potentially higher energy efficiency
  - Tile-based architecture (BISC): scalability and reconfigurability

- **New SC multiplier for BISC**
  - More efficient, accurate, and reproducible
  - Variable-latency multiplication
  - Seamlessly support “vector” operations

- **Applications to Deep CNNs**
  - Great improvement in area-delay product over conventional SC
  - Better efficiency than optimized conventional binary designs