A Survey of Stochastic Computing in Energy-Efficient DNNs On-Edge

Danghui Wang, Zhaqing Wang, Linfan Yu, Ying Wu, Jiaqi Yang, Kuizhi Mei, Jihe Wang

1. Northwestern Polytechnical University, National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, Xi’an, China.
2. Northwestern Polytechnical University, Engineering Research Center of Embedded System Integration, Ministry of Education, Xi’an, China.
3. Faculty of Electronic and Information Engineering, Xi’an Jiaotong University Xi’an, 10049, China.

Abstract—In recent years, “edge intelligent computing power deployment” has become a key and difficult problem in key areas such as system structure and artificial intelligence. Some independent decision-making scenario requires the edge devices to have the ability to handle high energy efficient and high throughput tasks. However, restricted by the traditional accurate computing mode and memory-computation separation structure, the edge devices often fail to provide the computing power guarantee of high throughput and high energy efficiency, which seriously hinders the edge intelligent deployment. This paper summarizes the new computing model integrating memory and computation of deploying stochastic circuits around memory, and constructs an edge intelligent system taking stochastic computing as a core from three levels of microstructure of circuit, architecture of near-memory stochastic computing and system-level software/hardware co-optimization. It strives to break through the power consumption and performant bottleneck of traditional computing model, and greatly improves the deploying ability of complex intelligent computing on edge equipment.

Keywords—edge intelligent computing power, stochastic computing, memory-computation separation

I. INTRODUCTION

Now the edge intelligent computing power deployment has been paid wide attention from all walks of life, but there are still many problems in its practical application to be solved. For example, when a drone group launches a stealth approach attack on enemy positions, units have to remain radio silent, independently perceive and adjust the position in the formation to avoid being detected by an enemy antenna before arrival. This independent decision-making scenario requires the edge device to have the ability to independently complete complex real-time tasks such as target recognition, scene inversion, path planning, and flight control with low resource overhead. However, the traditional computing architecture based on the von Neumann structure essentially performs binary bitwise accurate computing with the memory-computation separation structure. Restricted by this traditional computing architecture, edge intelligent computing acceleration units such as embedded GPU, NPU and FPGA can only exchange large overhead of computation and memory for limited performance improvement, which is difficult to support the deployment of complex applications on the edge.

For edge devices, the demand for high energy efficiency and high throughput conflicts with the supply of limited performant improvement. Resolving the contradiction between supply and demand in deployment of edge computing power is the key to improve the intelligence of edge devices. And it is mainly caused by the limitations of accurate digital computing mode and memory-computation separation architecture. These two limitations become the main obstacles to edge intelligent computing power deployment [1]. The following explains the developmental dilemma of the current edge intelligent computing power from the accurate computing mode and memory-computation separation structure respectively:

Accurate computing is not a necessity of artificial intelligence, but will inevitably become the main factor restricting edge intelligent computing of high energy efficiency. Recent extensive research has shown that artificial intelligence computing represented by DNN is well robust to approximate computing modes such as low precision and hybrid precision[2]. These variable precision methods can reduce the delay, power consumption, area and other overhead caused by the accurate computing to some extent, which are effective ways to improve the edge computing power. However, under large-scale parallel computing conditions such as DNN, MAC units based on bitwise accurate computing are difficult to achieve the deployment of lightweight computing power. Take 28nm process as an example[3][4]: the single structure power consumption of FIX16-MAC standard algorithm unit with 16-bit bit-width reaches 1.98 mW, and power consumption of FIX32-MAC with higher precision is even up to 5.65mW. At the same time, the computing scale of the AI model is growing exponentially every year. Since 2012, the global computing volume of AI has increased by more than 300, 000 times[5], and the scale of the general AI model has also exceeded the billions of neurons[6]. The deployment of AI models on edge devices is often constrained by the energy efficiency of the system, and has to compromise in terms of accuracy of inference and real-time. The traditional accurate computing mode can no longer support the large-scale edge intelligent computing deployment. Therefore, in order to achieve a deployment of high energy efficient edge computing power, the problem of how to provide edge large-scale computing power with extremely low power consumption must be solved.

The Von-Neumann structure based on memory-computation separation is difficult to support intelligent computing needs of high throughput, and the frequent handling of data between memory and computation components has become the main performant bottleneck. To improve the data feeding capability of memory, universal computing architecture widely uses multilevel on-chip cache. Although expanding the cache capacity can directly improve
the computing throughput, the SRAM-based cache power on the chip has consumed hundreds of mW, and occupies more than half of the overall energy consumption of the chip[7][8], seriously hindering the increase of intelligent computing power. In academia, the architecture of processing dataflow represented by MIT Eyeriss[9] relies on the interconnection between Process Elements (PEs) to establish distributed data migration networks. However, recent research shows that[10][11], such large-scale caching networks are limited by power consumption of cache[57] and the working frequency is difficult to break through 200MHz, resulting in PEs being “hungry” for a long time. In industry, Google’s TPU [12] can only equip its systolic array with 24MB on-chip SRAM cache, difficult to match the 64GB 3D-stacking High Bandwidth Memory (HBM) capacity[13][56], becoming the main bottleneck for expanding the computing throughput. The Ampere architecture[14] released by Nvidia in 2020 also expanded the global graphics capacity to 40-80 GB, but data migration across multilevel caches caused TDP power consumption to be 400W that makes TDP almost impossible to run at full capacity in an edge computing environment. Therefore, in order to support the deployment of high throughput edge computing power, the problem of how to design a simple system of both memory and computation to reduce the data cache overhead between memory and computation components must be solved.

For the current development dilemma of edge AI computing power, Stochastic Computing (SC) theory with ultra-low power consumption and high concurrency brings dawn to the construction and deployment of the next generation of edge AI systems. Although the concept of stochastic computing has been proposed by Gaines et al [15], as early as the 1960 s, it has not been widely used. The reason is computing power and energy efficiency of the chip in the past have developed steadily in accordance with Moore's Law and can continuously meet the needs of application, and there is no need to introduce new imprecise computing mode. However, in the current era of ubiquitous intelligence[16], the edge intelligent computing system also faces the challenges of the end of Moore's Law and the explosion of computing demand, which provides a strong demand traction for the use of stochastic computing to resolve the current computing power crisis.

However, since previous research has not gotten rid of the shackles of von Neumann structure, stochastic computing can’t be given the core status of computing power of the system. It can only rely on complex Stochastic Number Generator (SNG) for frequent data domain switching and lengthy data path for slow data migration, restricting the edge intelligent computing deployment of both high energy efficiency and high throughput.

In order to give full play to the performant advantages of stochastic computing, the paper plans to establish a hierarchical software/hardware systematic system for stochastic computing, for the two design objectives of high energy efficiency computing and the high throughput computing, from the three aspects of microstructure, architecture and software/hardware co-design. The paper is structured as follows. Section 2 introduces the advantages of edge intelligent stochastic computing and the research overview. Section 3 introduces the optimization of the stochastic circuit microstructure. Section 4 describes the design of stochastic computing architecture integrating memory and computation. Section 5 describes the software/hardware co-optimization of the resource-limited SC-DNN. Section 6 briefly summarizes edge intelligent stochastic computing.

II. EDGE INTELLIGENT STOCHASTIC COMPUTING

This section introduces the advantages and existing problems of edge intelligent stochastic computing system, and briefly analyzes the design of edge intelligent system based on stochastic computing from three aspects of microstructure optimization of stochastic circuit, stochastic computing architecture with integrating memory and computation design and software/hardware co-optimization of resource-limited SC-DNN.

Stochastic computing does not pursue the most accurate result. It uses “0/1 sequence” to quantify the value, a single and gate circuit can complete the multiplication operation٢ of random domain in Fig.1a, and a single multiplexer can achieve addition in Fig.1b. Fig.1c is a stochastic number generator that lets the n-bit Linear Feedback Shift Register (LFSR) and Base-2 number B pass through a comparator to generate a stochastic bit-stream X.

Fig.1. Stochastic computing elements: (a) multiplier circuit (b) adder circuit based on a multiplexer (c) Stochastic Number Generator.

A. Building an Edge Intelligent System with Stochastic Computing

Stochastic computing is an effective way to solve the edge computing power problem in the future, but its serialization processing mode poses a brand-new challenge for the system design. The following explains the advantages and urgent problems of building an edge intelligent system using stochastic computing from the perspectives of computing energy efficiency and computing throughput.

First, the logical gate-level stochastic computing can greatly optimize the energy efficiency of intelligent computing[58]. Current mainstream intelligent computing uses MAC units, in which the shift operation alternates with the addition operation[17] on the binary domain. And the data are constantly driven to repeatedly iterate between the “shift register” and the “carry-lookahead add circuit”. Compared to traditional MAC unit structures, stochastic computing improves energy efficiency by more than two orders of magnitude[18], with huge energy efficiency potential in large-scale deployment. However, in the traditional computing system, it is difficult to get the energy efficiency advantage of stochastic computing to give full play, and there are two main problems: (1) Digital compatibility problem: the data needs to frequently switch the data form between the “binary domain” and the “random domain”, which offsets the

٢Random computation can also use xor logic to implement multiplication in (-1,1) domain.
Real-time problem of operation: although the Operation per Joule (OpJ) index is significantly improved, the “bitwise operation” of the gate level will lead to the exponential increase in computing delay [19]. And it is difficult to balance the “low power consumption” and “real-time”. Therefore, it is necessary to improve the traditional computing architecture based on the inherent defects of stochastic computing, in order to develop a “high energy efficiency” edge intelligent system.

Secondly, the tight coupling design of the stochastic circuit and the memory can significantly improve the intelligent computing throughput. The performant bottleneck of a type of intelligent computing system mainly stems from the inability to establish an efficient dataflow path between memory and processing units”, which makes the computing throughput difficult to get close to the theoretical peak[20]. The scheme of inserting multilevel caching between the two essentially exchanges a very high area/power for limited performant improvements, and it is difficult to exceed 50% in real systems[21]. Stochastic computing provides the possibility to crack the limitations of the memory-computing binary structure; due to its great advantages in the circuit area, the stochastic computing units can be tightly coupled with the memory[22] and deploy the intelligent computing power around the memory. On the one hand, it can significantly reduce the migration overhead of the dataflow. On the other hand, the computing throughput matches the theoretical upper limit of the memory bandwidth to reach the “Roofline model optimum”[23]. However, once this “near-memory deployment” is fixed in structure, it will cause the problem of adaptability of the algorithm, mainly manifested in two aspects: (1) When the computation is solidified into a near-memory circuit, it can only support the inference of fixed accuracy and bit-width, which is difficult to be comparable with the “mixed precision method”[24][25] commonly used in edge intelligent computing; (2) near-memory stochastic computing forms a complex “pipelining” dependence between the memory and its surrounding operation units, which is easily affected by software/hardware factors and causes “unbalanced pipelining”. Therefore, the software/hardware system of stochastic computing must be reconstructed using “software/hardware co-optimization” methods to develop “high throughput” edge intelligent systems.

B. The Design of Edge Intelligent Stochastic Computing System

Building edge intelligent system using stochastic computing can significantly improve computing energy efficiency and throughput, but the traditional computing architecture needs to be improved and the software/hardware system needs to be reconstructed with the “software/hardware co-optimization” method to solve the problems in these two aspects. The following is a simple analysis from bottom to up layer by layer from “microstructure”, “architecture” and “software/hardware co-design”, as shown in Table I.

At the microstructure level, “low computing accuracy” and “poor SNG energy efficiency” have become the main problems of current stochastic circuit design. Therefore, computing accuracy and SNG energy efficiency need to be optimized. The optimization of stochastic computing accuracy mainly includes optimization of multipliers, optimization of max pooling operations in neural networks, and the use of deterministic encoding. The optimization of SNG circuit energy efficiency mainly includes optimization of RNS, optimization of probabilistic convert circuit (PCC), and the utilization of the “Spin-Hall-effect (SHE)”.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Source</th>
<th>Target</th>
<th>Technology</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Optimization of memory mode for SC-DNN</td>
<td>Reconstruction of computing mode</td>
<td>[49-50]</td>
<td></td>
</tr>
<tr>
<td>SC on traditional memory</td>
<td>Binary storage</td>
<td>[19][51-52]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Compressed storage</td>
<td>[53-55]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC on new non-volatile memory</td>
<td>ReRAM-SC architecture</td>
<td>[40-43]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DWM-SC architecture</td>
<td>[44-45]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Technology</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization of stochastic computing accuracy</td>
<td>High-accuracy SC multiplication and addition</td>
<td>[26-27]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>High-accuracy SC pooling</td>
<td>[26]</td>
<td></td>
</tr>
<tr>
<td>Deterministic SC encoding</td>
<td>RNS circuit of high energy efficient</td>
<td>[32-33],[56]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCC circuit of high energy efficient</td>
<td>[31]</td>
<td></td>
</tr>
<tr>
<td>Spin-Hall-effect</td>
<td>[34-35]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

At the architecture level, the stochastic circuit is tightly coupled with the “traditional memory (DRAM)” and the “new non-volatile memory (NVM)”, respectively, forming the stochastic computing architecture of traditional DRAM integrating memory and computation and the stochastic computing architecture of new NVM integrating memory and computation. The traditional research customizes and optimizes the memory system from three aspects: “reconstruction of memory principle” and “logical unit embedding”[60][61][62]. Depending on the type of new non-volatile memory materials used, the architecture of SC on new non-volatile memory can be divided into ReRAM-SC architecture and DWM-SC architecture.

At the level of “software/hardware co-optimization”, the relevant research can be divided into optimization of both computing mode for SC-DNN and storage mode for SC-DNN. The main feature of SC-DNN computing is the “inability to provide accurate computational results with limited resources”, but SC-DNN generally has the accurate superiority in the AI model software/hardware co-optimization. It can be realized through three neural networks: RNN, TNN and CNN. The main characteristics of SC-DNN data storage are “low density of random sequence storage and high transmission cost”. Optimization of storage mode for SC-DNN can be divided into two ideas of binary storage and compressed storage.

III. MICRO-STRUCTURAL OPTIMIZATION OF STOCHASTIC CIRCUIT

Stochastic Computing is an example of approximate computing, which uses random bit-stream to represent data. Sophisticated operations can be transferred simple bit-computation. At present, there are two main problems in the stochastic circuit design: it requires a long bit-stream to achieve satisfactory accuracy due to the fact that the number is expressed as the probability of “1” in the series bit-stream, so the calculation accuracy is low. The conversion between random number and binary number requires a SNG circuit.
The area of these circuits will reduce the total gain of area, so the circuit energy efficiency is poor. Therefore, stochastic circuit optimization mainly focuses on these two aspects.

A. Optimization of Stochastic Computing Accuracy

For the multiplication and addition operation of SC, it depends on the weak correlation of “Bernoulli assumption” and determines the overall calculation efficiency and reliability. To solve this problem, T.J. Baker[26] uses hypergeometric distribution instead of Bernoulli distribution to propose a new modeling method, and designs a high-precision tree adder, which reduces the error and overall circuit area, achieves an accuracy improvement. H. Zhou et al.[27] proposed a new Scaled Population (SP) based arithmetic computation approach that maps the multiplication of decimals (between [0,1]) to a wider range of numerical fields (between [0,2^M]), and specially designed a base-exponent separated random coding method to ensure O (1) gate delays. Different from the multiply-add operation, the max pooling in neural networks depends on the strong correlation among the operands, which leads to the same encoding method appearing the “precision vibration” at different stages of the DNN operation. M. Lunglmayr et al.[28] proposed a new SMAX / SMIN function based on shift registers, which improves the accuracy of the uncorrelated bit-stream and overcomes the low-precision pooling problem of the weak correlation among the data. Due to the inherent random fluctuation error and long latency of SC, the accuracy and energy efficiency when applied to convolutional neural networks decrease. H. Sim et al.[29] proposed a new SC multiply algorithm and its vector extension (SC-MVM), which makes CNN more accurate and more energy-efficient than traditional methods. Zhiyuan Chen et al.[30] proposed an optimized and scalable SC-MAC unit to solve the problem of long bit-stream latency in SC, which has improved performance, energy saving, and reduced area design.

In order to get rid of the influence of random distribution assumption on accuracy, a random sequence generation method (LD: Low-discrepancy) of deterministic coding uses Monte-Carlo sampling principle to replace global sequence distribution with local sequence distribution, which significantly improves the calculation accuracy and stability. Therefore, deterministic coding has gradually become a mainstream solution. However, these methods do not extend well and They lack the progressive accuracy of SC. M. H. Najafin et al.[31] proposed two fast-converging, extensible deterministic approaches to SC based on LD sequences. The first one is to use different Sobol sequences to generate bit-streams and process the streams with specific cycles, so that the best accuracy can be achieved at a fixed processing time. The second is to use LD sequence generators instead of pseudo-random source to generate deterministic accurate bit-streams, which has the minimum area × latency product[31]. Fig.2a shows the source structure of generating Sobol sequences based on the first method. Fig.2b shows the source structure of the Sobol sequence generated by the second method.

Table II. The Proportion of RNS, PCC Structure and Energy Consumption[32]

<table>
<thead>
<tr>
<th>Items</th>
<th>RNS</th>
<th>PCC</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion</td>
<td>70%</td>
<td>26.4%</td>
<td>More than 80%</td>
</tr>
</tbody>
</table>

The memory structure used in binary computing will increase the additional data costs, which will also reduce energy efficiency. Xinyue Zhang et al.[32] proposed a new memory system for SC based multiply-accumulate (MAC) engine applied in CNN, which is compatible with traditional memory systems and reduces energy consumption. Aiming at the problem that each SNG circuit contains LFSR and a PCC composed of multiplexer. This huge circuit seriously affects the energy efficiency advantages of SC. Table II shows the proportion of RNS, PCC structure and energy consumption in the whole circuit design.

The improvement of SC accuracy requires a longer bit-stream, which often increases greatly in computational latency and energy consumption, which is contrary to the original intention of SC. S. Asadi et al.[36] proposed a context-aware bit-stream generator to improve the performance of deterministic bit-stream processing systems, designed a control unit to determine the bit-width of the input data, and dynamically adjusted the bit-streams with the minimum length required for system generation. Fig.3 shows the structure of the bit-stream generator. With this structure, the processing time can be greatly reduced and the latency in calculation can be solved under reasonable hardware overhead.
The data conversion overhead of edge intelligent analog sensors and bit line (BL) logic operation components is a significant challenge. In the face of the demand for large-scale parallel computing of edge, the existing SC researches still follow the microstructure of “SNG + Gate”. The random circuit is still a subsidiary acceleration component of Von Neumann, and it does not fundamentally solve the problem of inefficient data conversion between the “binary domain” and the “random domain”. Therefore, the microstructure foundation of large-scale parallel needs to be improved.

IV. STOCHASTIC COMPUTING ARCHITECTURE WITH INTEGRATED MEMORY AND COMPUTING DESIGN

While traditional computing unit is difficult to integrate with the memory, SC’s micro-structure has a small circuit area and satisfies the precision of simple computing. It can be directly deployed around the memory, forming a new integrated design of memory and computing and reducing the data migration overhead between them. Related research covers the tightly coupled design of random circuits with “traditional memory” and “new non-volatile memory (NVM)”.

A. SC on Traditional Memory

For the characteristic of the “random domain” data, traditional Processing-In-Memory (PIM) research customizes memory systems from the aspects of “reconstruction of memory principles” and “logic unit embedding” [59].

Conventional memory are designed for binary computing. When they are applied to SC, extra conversion unit between stochastic and binary numbers is necessary. As early as in 2015, J. Lu designed an analog memory (AM) [37] based on standard CMOS process, which was used to store the analog quantities in neural network reasoning. AM overcomes the limitations of digital signals, and achieves higher energy efficiency. Based on AM, SK Khattamifard proposed StochMem architecture [38], which converted random sequences into analog quantities and stored them in AM directly. It converts the analog data into random sequences, and uses a low-power method in computing. This SC architecture can reduce the data-conversion overhead of edge intelligent analog sensors by 52.8%.

However, analog memory needs further improvement in order to be compatible with conventional memory systems. SCOPE [22] accelerator architecture proposed by Xie Yuan’s team is a typical representative of the research on logic unit embedding. As shown in Fig.4, SNG and SC amplifier circuits are embedded in the existing DRAM structure. A complete data path inside the memory is designed by adding registers and bit line (BL) logic operation components.

This architecture couples computing and memory resources, and combines stochastic computing with DRAM-based in-situ accelerators. It is a classic PIM architecture and has achieved a 2.3 times energy efficiency improvement.

These new architectures cannot solve the low storage efficiency of random sequence. In the aspect of optimization, H. Sim proposed a “logarithmic” representation method [39], which efficiently performs “logarithmic domain storage” on neural network weights. Reducing the accuracy of weights will not affect the accuracy of large-scale DNNs computing. By applying this weight logarithm to the training process, the neural network model gradually adapts to the logarithmic error. In the final reasoning process, the power consumption has decreased by 24%.

B. SC on New Non-volatile Memory

The implement of SC based on CMOS involves the generation of complex circuit structure of “random bits”, which is restricted by area costs. New non-volatile storage materials represented by ReRAM and DWM are vulnerable to unstable states (Probabilistic - Switching), suppressing this unstable state is the main goal of general memory design. However, for SC, this unstable state can be used to generate random sequences, and realize an efficient PIM design [40].

On ReRAM devices, the probability of charge drift can be controlled by modifying the duration of the write current pulse, and then the ReRAM unit will generate corresponding random sequences. S. Gupta et al. designed a random sequence generator SCPIMP [41] on the basis of this feature, and combined with the logic computing capabilities of ReRAM, which realized a PIM architecture of SC.

In order to eliminate the signal conversion overhead between SNG based on ReRAM design and random logic based on CMOS circuit, W. Shen et al. introduced a configurable control structure for ReRAM [42], using stateful logic to replace conventional random logic circuits for SC. The parallelism of computing and memory can be adjusted flexibly, which significantly improves the throughput and speed, as shown in Table III.

<table>
<thead>
<tr>
<th>Material</th>
<th>Parallellism</th>
<th>Area x Speed (μm·ns)</th>
<th>Normalized Area x Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOS</td>
<td>1 2024</td>
<td>2078720</td>
<td>1</td>
</tr>
<tr>
<td>Mixed</td>
<td>1 1024</td>
<td>1781760</td>
<td>0.857</td>
</tr>
<tr>
<td>ReRAM</td>
<td>1024</td>
<td>97110</td>
<td>0.047</td>
</tr>
</tbody>
</table>

Because the existing in-memory SC’s results are approximate, M. R. Alam et al. designed an auxiliary operation logic based on ReRAM units [43], which converted data into deterministic codes to obtain accurate results.

The main advantages of DWM devices are ultra-low static power consumption and ultra-high storage density. X. Ma et al. designed a neural network weights storage method based on DWM [44], which compressed the weights in the “random domain”, thus obtained a storable random sequence and obtained a high accuracy in the application level.
H. Zhang et al. of Bei Hang University designed SC logic with new magnetic skyrmion material, which triggered the logic operation behavior of electrons in the storage unit [45]. In this device, “AND-OR” logic can be used for stochastic multiplication and two types of multiplexers are used for addition. This work has established SC architecture on a quantum scale.

C. Existing Problems at the Present Stage

The PIM architecture, which combining memory and computing, mainly faces the problem of “inefficient storage of random sequence”. Past research often adopts a compromise solution combining “binary storage” and “stochastic computing”. The lengthy and slow process of feeding random sequences is difficult to drive the large-scale parallel SC unit. Therefore, efficient storage methods of “random sequences” must be researched specifically to support the design of high throughput near-memory SC architectures.

V. SOFTWARE/HARDWARE CO-OPTIMIZATION OF RESOURCE-LIMITED SC-DNN

The inability to provide accurate computational results with limited resources is the main feature of SC-DNN computation; The main characteristics of SC-DNN data storage are low density of random sequence storage and high transmission cost. Therefore, the optimization of both computing mode for SC-DNN and the storage mode for SC-DNN are the main aspects of the research.

A. Optimization of Computational Mode for SC-DNN

Y. Wang et al. theoretically analyzed and compared SCNNs (stochastic computation-based neural networks) and BNNs (binary neural networks) in terms of universal approximation properties, energy complexity, and applicability of hardware implementation. Theoretically, the algorithmic reliability of the universal approximation principle on SC-DNNs is demonstrated, which lays the theoretical foundation for the feasibility of SC-DNNs and reveals the accuracy advantages of SC-DNNs in artificial intelligence models in general [46].

XNOR networks aims to reduce the model size and computational cost of neural networks for deployment on specialized hardware that requires limited hardware resources for real-time processes. A. Ardakani et al. analyzed the compatibility of stochastic computation with XNOR-RNN, a typical binary network structure, and proposed a way to use stochastic computation to combine all of the weights in LSTMs (long short-term memories) by using stochastic computation to convert all multiplications in LSTMs in XNOR operations [47]. Experimental results show that the proposed XNOR LSTMs reduce the computational complexity of the quantized LSTMs by a factor of 86 without sacrificing latency, while obtaining better accuracy in different time tasks. This is also the first time that XNOR operations are used to perform gate and state multiplication operations. As a typical three-valued network, ternary neural network (TNN) can also be fully implemented by stochastic computation. Y. Zhang et al. of Peking University proposed a parallel SC-based NN accelerator purely using bitstream computation which apply a bitonic sorting network for simultaneously implementing the accumulation and activation function with parallel bitstreams [48]. The proposed design not only has high fault tolerance, but also achieves at least 2.8× energy efficiency improvement over the binary computing counterpart.

For the more general Convolutional Neural Networks (CNN) model, a new structure SkippyNN is proposed that reduces the computation time of SC-based multiplication in the CNN convolutional layer. each convolution in CNN consists of a number of multiplications where each input value is multiplied by a weight vector. The result of generating the first multiplication can be performed by multiplying the difference between the input and successive weights by the following multiplication. Using this property, a difference multiplication and accumulation unit called DMAC was developed to reduce the time consumed by convolution in SkippyNN [49]. The efficiency of SkippyNN is evaluated using four modern CNNs, and SkippyNN provides an average of 1.2x speedup ratio and 2.7x energy savings compared to the binary implementation of the CNN gas pedal.

In the field of quantum computing, R. Cai et al. used adiabatic quantum superconducting materials to establish the SC-DNN framework [50], which can obtain 96% inference accuracy with 6.8×104 times energy efficiency advantage under ideal environment.

B. Storage Mode Optimization for SC-DNN

The optimization for storage is mainly divided into two ideas: “binary storage” and “compressed storage”.

a) Binary Storage

P. K. Muthappa et al. designed a “binary register” → SNGs → SCNeurons → “binary cache” storage path [19] for random sequences under the FPGA on-chip resource constraint. Y. Liu et al. proposed for the first time to have binary numbers and random sequences jointly participate in the computation and store only the binary numbers of them [51] to improve the ability of RNNs (Recurrent neural networks) against random noise. The proposed design has a higher noise tolerance compared to the binary implementation. When the noise level is high during the computation, its inference accuracy is 10% ~ 13% higher than the FP (floating-point) design. To further improve the computational flux of neurons, an efficient stochastic computation-based inference framework for large-scale DCNNs. HEIF framework [52], is proposed to extend the whole storage path into a pipeline structure by adding caches between operations. Experimental results show that HEIF improves throughput by 4.1×, area efficiency by 6.5×, and energy by 5.6× over previous SC-DCNNs in large-scale applications.

| Table IV. Performance of the Proposed XNOR-LSTM Models vs Their Quantized Counterparts. |
|---|---|---|---|---|---|---|
| # | LSTMs (GQE11/10) | ASIQ (GQE11/10) | HiNet (VQ-15/16) | XNOR (VQ-15/16) |
| LSTM Accuracy (%) | CELM | 67.5 | 69.3 | 69.9 | 69.2 |
| XNOR Accuracy (%) | CE | 68.0 | 69.3 | 69.9 | 69.2 |
| Conv(XOR) Accuracy (%) | 99.9999 | 99.9999 | 99.9999 | 99.9999 |
| Conv(XOR) Energy (W) | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Conv(XOR) Delay (ns) | 1000 | 1000 | 1000 | 1000 |

1559
Fig. 5. Structure of a multicell memory block in the SC RNN [51]

b) Compressed Storage

To reduce BRAM consumption on FPGAs (Field Programmable Gate Array), G. Maor et al. introduces SC into LSTM and creatively proposes an SC-based LSTM architecture design to save the hardware cost [53]. The evaluation results show that the SC-LSTM design works smoothly and can significantly reduce power consumption by 73.24% compared to the baseline binary LSTM implementation without much accuracy loss. So far, SC-based CNN (convolutional neural network) accelerators have been kept to relatively small CNNs only, primarily due to the inherent precision disadvantage of SC. H. Sim et al. present a DPS[54] (dynamic precision scaling) SC-CNN that is able to exploit dynamic precision with very low overhead, along with the design methodology for it. Our experimental results demonstrate that our DPS SC-CNN is highly efficient and accurate up to ImageNet-targeting CNNs, and show efficiency improvements over conventional digital designs ranging in 50–100% in operations-per-area depending on the DNN and the application scenario, while losing less than 1% in recognition accuracy.

H. Xiong et al. reduce the storage overhead by transforming longer random sequences in DNNs into a shorter compressed format through look-up tables (LUTs) based on deterministic compressive coding[55].

C. Existing Problems at the Present Stage

Previous software/hardware co-design generally ignored the practical problems in the deployment of edge intelligence. The computation and storage requirements of DNNs vary greatly from layer to layer, making it difficult to perform end-to-end processing in a pipeline manner. Therefore, an edge-oriented software/hardware co-optimization approach needs to further consider the relationship between accuracy, resources, and latency to form an integrated optimization approach.

VI. CONCLUSIONS

Building an edge intelligent system using stochastic computing is effective for the targets of both high energy efficiency computing and high throughput computing. However, there are still problems in micro-architecture, architecture and software/hardware co-design.

ACKNOWLEDGMENT

This work is supported by National NSF of China (61802312 and 61472322), Natural Science Basic Research Plan in Shaanxi Province of China (2019JQ-618), and open fund of Integrated Aero-Space-Ground-Ocean Big Data Application Technology (20200105).

REFERENCES


