ISSN: 2664-9640

A Survey of Optimization Techniques of Convolutional Neural Networks Based on Stochastic Computing

Ya Dong[1], Xingzhong Xiong[1,2]

[1] School of Automation and Information Engineering, Sichuan University of Science & Engineering, Zigong 643000, China;

[2] Artificial Intelligence Key Laboratory of Sichuan Province, Zigong 643000, China.

Abstract

At present, the number of parameters in Convolutional Neural Network (CNN) shows a trend of dramatic increase with the rapid development of CNN, and the corresponding operations are becoming more and more complex. Stochastic Computing (SC) has the advantages of low power consumption, high fault tolerance and simplified circuits, and can efficiently handle the reasoning tasks of neural networks, which makes stochastic computing a hot direction for solving the computational complexity of CNNs. In addition, the problems of low precision, slow processing speed and complex design requirements in stochastic computing also hinder the application of this method. This paper discusses the application of stochastic computing in Convolutional Neural Networks. Firstly, the representative computing unit and convolutional neural network structure and computational complexity in stochastic computing are introduced. Secondly, stochastic computation-based designs in CNNs are described and their advantages and disadvantages are evaluated. Finally, the future development direction of stochastic computation-based convolutional neural networks is discussed for the implementation methods of these designs.

Keywords

Stochastic computing, convolutional neural networks, low-power consumption, Precision.

1. Introduction

Convolutional Neural Network (CNN), which is one of the commonly used networks in neural networks, has a large number of multiply-accumulate operations. As the number of network layers increases, the number of corresponding parameters also increases sharply. In resource-constrained mobile devices, such as embedded devices, the binary computing method is still used to perform operations, which will face the problem of computational complexity caused by data bandwidth and storage problems. Stochastic Computing (SC) method has attracted the attention of researchers. Stochastic computing is a method proposed from the perspective of data encoding, which mainly uses the probability of "1" appearing in a random sequence composed of "0" and "1" to encode numerical values. Compared with traditional binary computing methods, stochastic computing has the advantages of simple circuit structure, low power consumption and high fault tolerance, and can be applied to many applications, especially the reasoning tasks of convolutional neural networks[36]. However, due to the inherent random characteristics of stochastic computing, the accuracy of the computing results is lost when compared with binary computing. On the one hand, in order to solve this problem, researchers have proposed many improvement schemes for the basic unit of stochastic computing in convolutional neural networks. On the other hand, in order to expand the application scope of stochastic computing in convolutional neural networks, the computational

ISSN: 2664-9640

complexity of convolutional neural networks has also attracted extensive attention of researchers, and they have turned their attention to low-energy, high-precision convolutional neural networks. Optimal design of the network.

2. Convolutional neural network and Stochastic computing computing principles

2.1. Convolutional neural network

A convolutional neural network is a feed-forward multilayer network in which information flows from input to output. A typical CNN model consists of convolutional layers, pooling layers, activation functions and fully connected layers. Referring to the CNN model, a new network model is generated by combining convolutional layers and pooling layers, which can achieve the purpose of improving the accuracy of the network structure. The classic CNN models mainly include GoogLeNet, AlexNet, VGGNet, etc. In the CNN model, most of the operations are concentrated in the convolution operation in the convolution layer, especially the inner product operation between the input data and the corresponding weight value.

2.1.1. Convolutional layer

The function of the convolutional layer in CNN is to extract features from the input data through convolution operations. Convolve the input feature map x_i with the convolution kernel composed of weights $w_{i,j}$, add the bias b_i to the result of the operation, and output the result of the convolution layer operation y. The convolutional layer operation is shown in formula (1).

$$y = \sum_{j=1}^{n} x_j * w_{i,j} + b_i, 1 \le i \le n . \tag{1}$$

2.1.2. Pooling layer and activation function layer

The purpose of the pooling layer is to downsample the image without losing the image features as much as possible. Currently, the commonly used pooling types in CNN networks are Max pooling and Average pooling. Both of them downsample the data, but the former focuses on feature selection to select features with better classification recognition, and the latter mainly downsamples the overall feature information to better reduce the amount of parameters. Taking the size of the pooling operation as an example, the operation process of the maximum pooling method is shown in Equation (2), and the operation process of the average pooling method is shown in Equation (3).

$$y_{\max} = \max_{p \times p} \left(y_1, y_2, y_3, y_4 \right) . \tag{2}$$

$$y_{aver} = \frac{1}{4} \sum_{i}^{4} y_i . \tag{3}$$

After the pooling layer, an activation function layer is generally followed, in order to introduce nonlinear factors in order to improve the feature expression ability of the model. The commonly used activation functions mainly include functions such as Sigmod, Tanh, ReLU, etc. The operation process is as follows

$$Sig \bmod : f(x) = 1/(1+e^{-x}) . \tag{4}$$

$$Tanh : f(x) = \tanh(x) . \tag{5}$$

$$\mathrm{Re}\, LU : f(x) = \max(0, x) . \tag{6}$$

ISSN: 2664-9640

The ReLU function has a faster convergence speed than the Sigmod and Tanh functions, making it the most popular function at present.

2.1.3. fully connected layer

The fully connected layer (FC layer), as the layer at the end of the neural network model, mainly performs linear space transformation on the input to obtain the output, and the corresponding operation process is shown in formula (7). The fully connected layer essentially implements the weighted sum operation of the input feature maps, that is, the inner product operation of different input feature maps and the corresponding weights. The calculation method in the fully connected layer is similar to that of the convolution layer. In a sense, the operation in the fully connected layer can be regarded as the convolution operation of the size of the convolution kernel.

$$y = f\left(\sum W_i x_i + b\right). \tag{7}$$

2.2. Stochastic computing

Stochastic computing is to convert the traditional binary numerical representation into a random sequence composed of "0" and "1", so that some complex operations that are represented by binary numerical values can be represented by simple gate circuits represented by the probability domain. Stochastic computing has the advantages of high fault tolerance, high flexibility and simple computing units involved in the operation. However, this method has the defects of low computing accuracy, long delay and low conversion efficiency.

In order to apply stochastic computing to neural network, a computing system based on stochastic computing is given, as shown in Figure 1. The system consists of three parts, namely sequence generation unit, stochastic computation unit and backward transformation unit**错误！未找到引用源。**. Among them, the sequence generation unit is to convert the binary numerical representation into a random sequence composed of "0" and "1"; the stochastic computingunit is to perform the correlation operation of the random sequence in the probability domain, such as multiplication and addition, etc; and the backward conversion converts the output random sequence of the stochastic computing unit into a binary numerical representation, and outputs the final result.

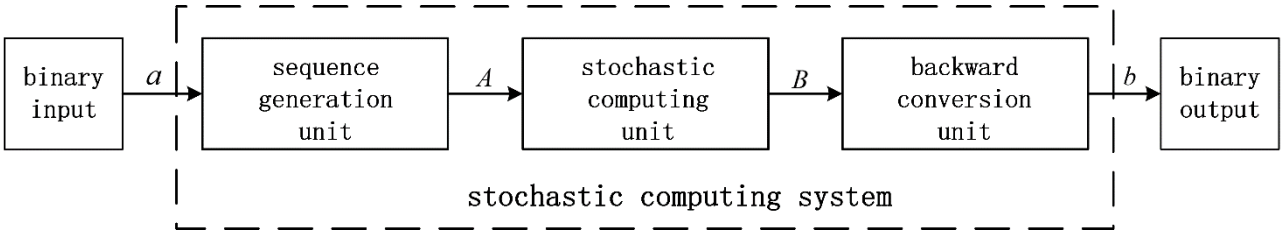

Figure 1: Components of a stochastic computing system

2.2.1. Sequence generation unit

The function of the sequence generation unit is to convert the binary numerical representation into a random sequence so that multiplication in the probability domain can be performed. At present, the encoding methods for this conversion process mainly include random distribution sequence stochastic computing (Random Stochastic Computing, RSC) and deterministic distribution sequence stochastic computing (Deterministic Stochastic Computing, DSC. The difference between the two lies in how to determine the distribution of "1" in the random sequence. The RSC method is implemented by a Stochastic Number Generator (SNG). The random sequence generator consists of a random number generator (RNG) and a comparator. The DSC method is implemented using a counter and a comparator. The advantage of using this method is that, this method ensures that the number of "1" occurrences in the sequence is equal to the size of the binary value, making the result of multiplication more accurate than the RSC method.

ISSN: 2664-9640

2.2.2. Stochastic computing unit

The stochastic computing unit mainly performs operations like multiplication, addition, and so on. Multiplication operations are commonly implemented with multipliers. For unipolar stochastic computation, the multiplier can be implemented with an AND gate, while for a bipolar stochastic computation, the multiplier is implemented with an XNOR gate. There are three main factors that affect the accuracy of the multiplier. One is the encoding method of the input sequence. The calculation accuracy of the unipolar stochastic computingmultiplier is higher than that of the bipolar stochastic computingmultiplier. The second is the difference between the input sequences. When the correlation is lower, the precision of the multiplier is higher. The third is the representation range of the input sequence, especially the representation range of the input multiplier. When the representation range of the input multiplier is small, the multiplication have the lower the accuracy[12].

Similarly, addition operations are implemented with adders. Due to the limited representation range of stochastic computing, the addition of two numbers in the range of [0,1] may result in a result that exceeds 1, making the result unrepresentable by stochastic computing. Therefore, when implementing stochastic computing addition to consider the problem of numerical overflow. To this end, the existing adders can be divided into two types: adders with scaling and adders without scaling, as shown in figure 1. The multi-selector (MUX)-based adder is a typical adder with scaling, which can solve the problem that the calculation result exceeds the range of representation, but it is not suitable for the case of multiple input sequences. There are two implementations of the adder without scaling , which are the structure based on the parallel accumulator (APC) and the structure based on the OR gate. The former has no precision loss, but the circuit structure is more complicated, and the latter has s ome precision loss and the structure is relatively simple.

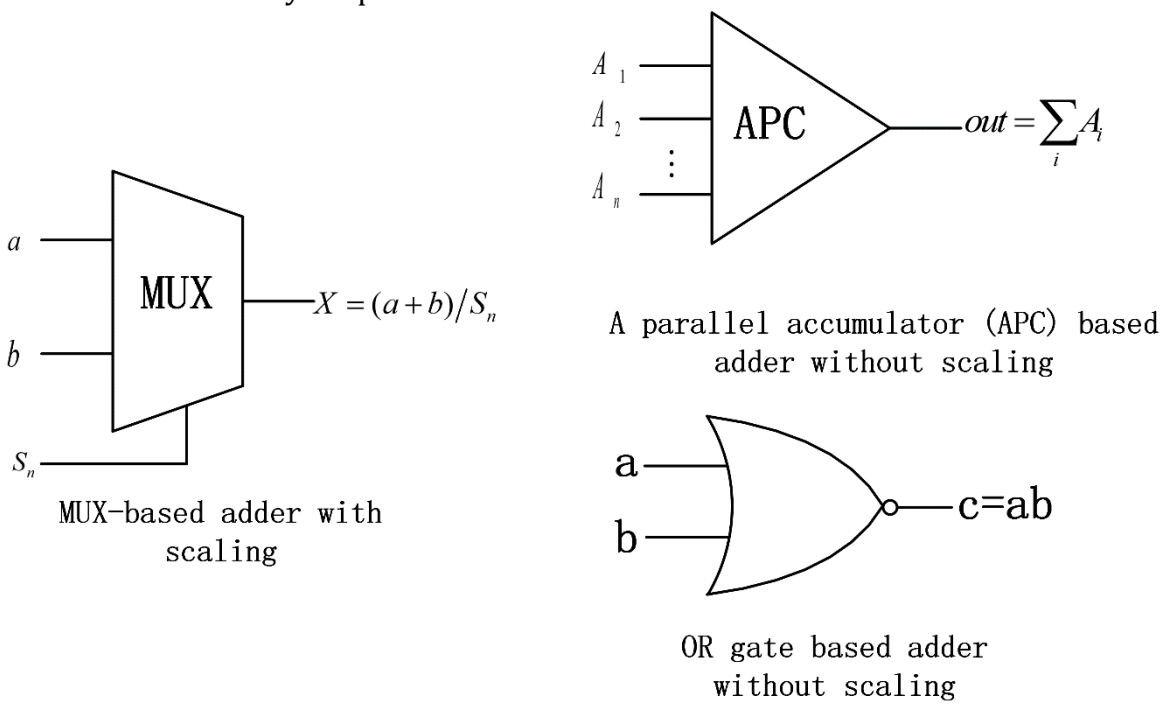

Figure 2: Adder with scaling and adder without scaling

In stochastic computing, in addition to simple multiplication and addition operations, more complex functions can also be implemented through Finite State Machine (FSM), such as the tanh function, the Stanh function[24], and the random sequence input method by Stanh The The Btanh algorithm proposed in [31] is generalized to an input of a sequence of integers. As shown in Figure 3, the structure of the most basic FSM-based stochastic computing unit is shown. The design of the stochastic computing unit based on FSM is simple, but the overall

ISSN: 2664-9640

accuracy of the method is not high due to the use of more approximate calculations in the algorithm, and it needs to be used according to the calculation accuracy requirements of the application scenario.

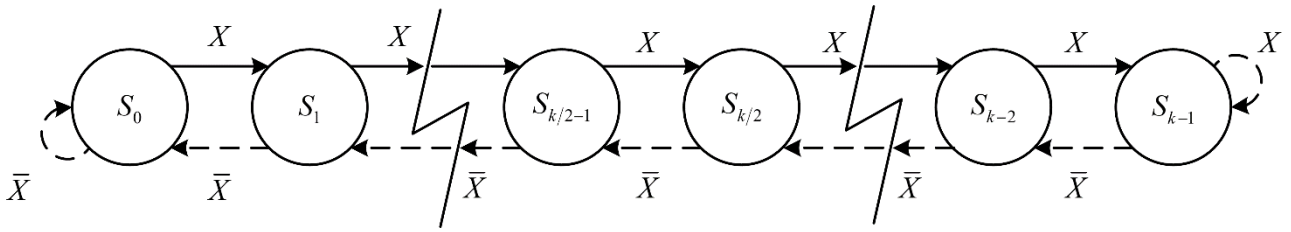

Figure 3: Stochastic computing unit based on FSM

2.2.3. Backward conversion unit

The backward conversion unit converts the random sequence into binary numerical representation, which can be realized by the accumulator.

3. Optimization Technology of Convolutional Neural Network Based on Stochastic Computation

In recent years, after researchers discovered the good value of stochastic computing in neural networks, the research on stochastic computing has gradually increased. According to the existing theoretical technology, the optimization technology of convolutional neural network based on stochastic computingis mostly carried out according to the design of stochastic computingunit. The stochastic computing unit can be roughly divided into three categories, namely, the coding method based on the generation of random sequences, the basic computing unit based and the neural network based processing layer type.

3.1. Encoding method based on generating random sequences

Introducing stochastic computing into a convolutional neural network means converting data from the binary domain to the probability domain for correlation operations. For this reason, the conversion process of converting binary data into a random sequence is based on the optimization of convolutional neural networks based on stochastic computing. An essential part of technology.

In general, in stochastic computing, binary data is converted into a random sequence by using a random sequence generator (SNG). In most previous SC works, linear shift registers (LFSRs) were used to implement RNG structures to generate random numbers. However, using the SNG-based method to generate random sequences will have two error sources: one is the initial conversion error caused by the introduction of SNG; the other is the calculation error caused by the correlation between different bitstream data[1].

Some researchers try to alleviate such problems by improving the design of SNG to achieve the purpose of improving computational accuracy and generating random sequences with high irrelevance[1]. Common improvement methods can be divided into improved implementation based on RNG generation [10][11][12][13], improved implementation based on SNG generation [14][15], and improved implementation between multiple SNGs [16][17].

These improved methods have the characteristics of scalability, small circuit area, and high precision. However, due to the inherent random characteristics of stochastic computing, there are significant limitations in terms of the conversion between Binary-encoded (BE) and stochastic computing (SC) in terms of high area cost and delay of the circuit and computing precision[18]. On the other hand, according to the principle of determining the distribution of “1” in random sequences to generate random sequences, some researchers have proposed encoding generation methods for centrally distributed sequences and uniformly distributed sequences[19][20]. This way could achieve the purpose of small circuit area, however, like the

ISSN: 2664-9640

implementation based on SNG, there is still the problem of large calculation delay. In addition, literature [21] proposed a new sequence generation method by simulating the neural synapse structure, which simulated the neural synaptic plasticity structure from the perspective of circuit wire connection, and the multiplication between input data and weights in a product neural network is realized by the wire connection transformation method of neural synaptic plasticity calculation.

3.2. Based on the basic computing unit

multiplication and addition operations, which require a lot of silicon area and sufficient memory space. In order to solve the above problems, people have studied many ideas, which can be roughly divided into two categories, one is to speed up the calculation process[22], and the other is to optimize the convolutional neural network model[23]. In 2001, Brown et al.[24] first proposed to apply stochastic computing to the calculation of neural networks, and designed a variety of stochastic computing units to replace traditional binary operations such as addition and multiplication, which greatly simplifies the computing circuit, highlighting the advantage of low complexity of stochastic computing. Multiplication operations can be efficiently implemented using XNOR gates in bipolar representation and AND gates in unipolar representation in stochastic computing. The addition operation can be performed by multiple modules, such as OR gates, multiplexers (MUX), accumulating parallel counters (APC)[25] and approximate parallel counters (AxPC)[26]. In order to reduce hardware overhead and calculation delay, the combination of stochastic computingand adder can form an approximate parallel counter (AxPC), such as APC structure based on AND-OR[27], AxPC structure based on 4-1MUX[28], APC structure of 2-1MUX structure, AxPC structure based on mirror full adder[29].

Based on the above research, people combine multiplication and addition operations in convolutional neural networks into inner product operations for research. In previous work, the work on stochastic computing of inner product operations can be divided into two categories, one is the inner product operation module based on APC structure [29][30][31], and the other is the inner product operation based on weighted summation modules (MUX trees) [32][33][34], both of which are discussed under the unipolar representation of stochastic computation. In order to improve the calculation accuracy, in these traditional stochastic computinginner product operations, the stochastic computingmultiplication part must use two uncorrelated random sequences to participate in the calculation.

The inner product operation module based on APC structure is widely used in the realization of neural network based on stochastic calculation. However, this inner product operation method has the disadvantage of being sensitive to correlation, and needs to generate random sequences of appropriate length for improvement. As the sequence length increases, the computational accuracy of this method can be improved, but the computational delay time is also increased. This is because the addition part is calculated on a bit-by-bit basis. In order to obtain reasonable calculation accuracy, a long sequence length is required, resulting in a serious increase in calculation delay and high cost consumption. In addition, the inner product method needs to convert the random sequence obtained by completing the multiplication operation into the BE output, which is usually implemented by a counter. Chen et al.[35] proposed a method to reduce the length of random sequences using binary decomposition, which can greatly save space and reduce computational latency. From the perspective of shortening the sequence length, Xiong et al.[36] proposed two high-precision stochastic computing units to improve the accuracy of DNNs based on stochastic computing, namely the reassignment-based correlation irrelevant multiplier and the accumulator-based ReLU unit, respectively. And adopt a length-adaptive method that uses variable lengths for different images to reduce the average sequence length. Chang et al.[33] proposed a scaling method that randomly computes the inner product function, which incorporates sparseness into the

ISSN: 2664-9640

probability of a random sequence of MUX selectors. In addition, the inner product operation module based on the APC structure has other disadvantages, such as the need to add a conversion circuit to convert the random sequence into binary format, and the higher area cost than the traditional inner product operation module.

The inner product operation module based on weighted summation is mainly used in the realization of digital filter based on random calculation. However, this method suffers from a decrease in computational accuracy when the number of inputs is large[37]. This approach has the advantage of being insensitive to input data. Ichihara et al.[32] based on this proposed a cyclic transfer sharing scheme that allows sharing a single RNG among multiple SNGs to achieve significant savings in hardware the goal of.

In addition to the discussion under the unipolar representation, Yuan et al.[38] and Haselmayr et al.[38] designed a high-accuracy and unscaled inner product arithmetic module using the two-line SC bipolar representation. However, both methods are sensitive to dependencies, and the two-line representation generally has a higher hardware footprint than the single-line representation.

Sim and Lee [39] also proposed another way of SC domain inner product operation. They proposed a method to implement SC domain inner product operation using up and down counters. Compared with the traditional SC method, this counter-based method has higher precision and lower computational latency, but this method is only relevant for SC with binary interface.

Further, Abdellatef et al.[40] proposed a high-precision, low-area-cost, circuit-independent design scheme for SC-domain inner product operation. Compared with existing methods, this method has better accuracy performance and is suitable for designs with large numbers of inputs and low sequence lengths.

In addition, Xia et al[41] proposed an inner product operation scheme based on stochastic computingto simulate the neural synapse structure based on the literature [21]. The scheme divides the inner product operation into forward transformation, probability and backward transformation, and the wire fan-out unit, wire selection unit and backward transformation unit are designed respectively to solve the problem of high complexity and high delay. In addition, Xia et al[42] also took advantage of the parallelism of stochastic computing, and proposed a method based on reconfigurable synaptic plasticity computing, which can perform inner product operations through bit multiplication and some complete adders, thus building a high-speed Parallelism, low hardware overhead architecture.

3.3. Types of processing layers based on neural networks

Since 2016, many researchers have studied the implementation of stochastic computation in convolutional neural networks. They achieve substantial reductions in area while maintaining an acceptable loss of precision. The focus of research on stochastic computing-based convolutional neural networks lies in the optimal design of a single function and the entire network, including convolutional layers, pooling layers, and activation functions. For the reasoning process of these layers, different tiny circuits are designed to perform the operations in the convolutional neural network in the SC domain. Furthermore, the currently proposed SC-CNNs are all created using the above SC circuits with different configurations, and are optimally designed to obtain acceptable computational accuracy. According to the reviewed literature, two types of SC-CNN have been proposed in the existing work, one is SC-CNN that realizes most CNNs in SC as the goal, and the other is hybrid SC that realizes only specific layers in SC -CNN. However, both only focus on the inference process in CNN [18][43].

Ren et al.[44] and Li et al.[45] used traditional SC elements to encode BP and adopted SC to implement inner product operation, average pooling, and easy-to-implement Stanh function in CNNs[26]. These work suffer from high error rates or very long bitstreams up to 1024 bits.

ISSN: 2664-9640

Since the inner product operation module based on the APC structure has better performance than the traditional XNOR-MUX structure [27], most existing SC-CNNs use the inner product operation module based on the APC structure or the AxPC structure perform convolution operations in convolutional layers. Sim et al.[46] used an APC-based structure to create convolutional layers. Ren et al.[47] used an AxPC-based structure with Stanh and Btanh functions used as activation functions, and a pooling approach to find feature extraction blocks suitable for SC-CNN. The previously mentioned literatures [41][42] also apply the designed SC computing unit to the convolution operation to achieve high memory bandwidth efficiency.

In addition, there is more work on SC-based pooling or activation units. Li et al. [48][34] designed a SC-based ReLU and applied ensemble and module-level optimization to their SC-DNN. Yu et al.[49] proposed SReLU and Smax for max pooling. Li et al.[50] replaced the existing sigmiod function with an approximate format sigmiod function to reduce hardware overhead. Nguyen et al.[51] proposed a new nonlinear activation function approximation method including tanh and sigmiod functions using stochastic computational logic based on piecewise linear approximation (PWL) in the range [-1,1]. Reference [52] introduces normalization and dropout into existing SC-CNN and proposes a novel normalization design including square summation unit, activation unit and division unit.

Many researchers have also conducted research on the hybrid SC-CNN model. Lee et al.[53] proposed a hybrid BE-SC CNN model that can be used for near-sensors, which implements only the first convolutional layer in the sensor with the SC-based approach, while the rest of the layers use the BE-based approach. Faraji et al.[54] proposed a low-discrepancy deterministic bitstream-based CNN implementation, in which the product operation in the first convolutional layer is implemented with SC and an APC -based structure for inner product operation. Zhakatayev et al.[55] tried to use both stochastic computing and binary computation in the network, so that the computational accuracy was improved compared to using only stochastic computing. Ardakani et al.[56] proposed a method for stochastic computingof integers, which can realize multiplication and addition of integer sequences and random sequences. Sim and Lee et al.[57][58] proposed a new SC multiplication operation for BISC (Binary Interface Stochas tic Computing), this inner product operation method is limited to BISC use, therefore, in SC only multiply-accumulate operation , all other operations are implemented in BE. The algorithm can achieve almost the same calculation accuracy as traditional binary calculation at lower delay, and this algorithm has a simpler circuit structure than traditional SC multiplication, because it eliminates SNG and AND gates in exchange for ratio SNG has a smaller down counter.

In addition to improving the operation modules in CNN, some researchers have proposed several new optimization methods from other perspectives. Lammie et al.[59] first attempted to approximate the multiplication of fixed-point weights and bias values using stochastic computation techniques during CNN training, and demonstrated the method's state-of-the-art learning performance on multiple datasets. Kim et al.[60] proposed a conditional computation scheme that combines precision cascades and synergistically with zero jumps to improve the computational speed of DCNNs. Zhang et al.[61] addressed the high latency, random fluctuations, and high hardware cost of pseudorandom number generators (PRNGs) with a new technique for generating parallel bitstreams. Zhang et al[61] also proposed a novel storage system applied to the SC-MAC (Stochastic Computation-based Multiply Accumulate Operation) engine of CNN to reduce energy consumption. Similarly, Wang et al.[62] also designed a SC-CNN s ynthesis scheme based on MLC PCM (multi-level cell phase-change memory) from the perspective of alleviating storage pressure, and reduced the neural network size without sacrificing CNN accuracy. This is determined by the inherently random nature of stochastic computing itself.

These implementation methods based on stochastic computing all have a problem, that is, they are not applied to the large-scale design optimization and extensive application of SC-based

ISSN: 2664-9640

CNNs, and most of them are for inner product operations such as multiplication and addition, activation functions or input sequences length and other parts have been improved, and a more general implementation method has not been explored.

4. Future direction of development

According to the previous description, stochastic computing has excellent application prospects in convolutional neural networks. Analyzing these optimization methods, the application and development directions of convolutional neural networks based on stochastic computing can be divided into three types. The first is the inner product operation method; the second is how to make full use of the parallelism of CNN; the third is the mixed operation method based on stochastic computing and binary computing.

4.1. Inner product operation method

In the convolution operation, the existing implementation method of multiply-accumulate operation (MAC) based on stochastic computingis still composed of the structure of multiplier and adder realized by AND gate. From the perspective of computational accuracy, an important improvement point to reduce the loss error lies in the method of generating random sequences. In addition to using LFSR, combining the coding method of uniformly distributed random sequences with LFSR is also helpful to improve the accuracy loss. Another improvement is in the way the adder is implemented. At present, there are mainly OR gate-based, MUX-based and APC-based structures. The structure based on OR gate has the advantages of low calculation accuracy but simple circuit implementation. Combining with the structure based on APC can achieve the goals of high precision and simple circuit structure at the same time. The MUX-based structure has a small loss of accuracy, and the combination with the APC-based structure also has a certain improvement in accuracy and hardware overhead. Therefore, the inner product operation method is also one of the future development directions.

4.2. Leveraging the Parallelism of Convolutional Neural Networks

In stochastic computing, the high precision is proportional to the length of the input sequence, and the increase of the sequence length will generate more energy consumption, but in practical applications, so much energy consumption is not desired. Large circuit area, reducing the input sequence and other angles to improve. In addition, the increase of the sequence length will also increase the computational delay, because it takes at least 2^k clock cycles to obtain a random sequence of length is 2^k. The convolutional neural network has parallelism and can input multiple data in parallel, greatly reducing the delay time and improving the calculation speed. How to take advantage of the parallelism in convolutional neural networks and play a great role is the direction to be explored next.

4.3. Mixed Operation Method Based on Stochastic Computing and Binary Calculation

The implementation method of the convolutional neural network based on the binary computing method faces the defect of large hardware resources. In order to solve this problem, researchers have introduced stochastic computing. However, the implementation of convolutional neural networks using only stochastic computing suffers from low accuracy, despite its advantages of low complexity and high fault tolerance. Therefore, the researchers proposed a hybrid operation method based on stochastic computing and binary computing. The primary problem to be solved in this computing method is the mutual conversion process between binary representation and random sequence, because this conversion process accounts for a large part of the resources of the entire method. consume. In addition, it is also necessary to explore how much the operation part based on stochastic computingin the entire

ISSN: 2664-9640

CNN occupies to make the overall performance better. However, at present, only the application of the operation module based on stochastic computingin CNN has been realized, and more in-depth exploration is needed.

5. Conclusion

Starting from the application and development of convolutional neural network based on stochastic computing, this paper introduces the composition of computing system based on stochastic computing, the structure of convolutional neural network and the computational complexity. The design unit of stochastic computing in convolutional neural network is briefly explained, based on the recent application of SC-CNN, it can be seen that the research objects of convolutional neural networks based on stochastic computing tend to have the application effect of stochastic computing under large-scale convolutional neural networks. This trend will promote the further application and development of stochastic computing in convolutional neural networks.

Acknowledgements

This work was supported in part by the key research and development project 2021YFG0127 of the Sichuan Provincial Department of Science and Technology, in part by the Graduate Innovation Foundation of Sichuan University of Science and Engineering under Grant y2020016.

References

[1] Alaghi A, Hayes J P. Survey of stochastic computing[J]. ACM Transactions on Embedded Computing Systems (TECS), Vol. 12 (2013), No. 2s, p. 1-19.

[2] Gai Rongli, Cai Jianrong, Wang Shiyu, et al. A review of the application of convolutional neural network in image recognition [J]. Small Microcomputer System, 2021, 42(09): 1980-1984.

[3] Zheng Y, Li G, Li Y. Survey of application of deep learning in image recognition[J]. Comput. Eng. Appl, Vol. 55 (2019), No. 12, p. 20-36.

[4] Naderi A, Mannor S, Sawan M, et al. Delayed stochastic decoding of LDPC codes[J]. IEEE Transactions on Signal Processing, Vol. 59 (2011), No. 11, p. 5617-5626.

[5] Chen Jienan. Research on efficient VLSI implementation technology of wireless communication DSP system based on probability calculation [D]. University of Electronic Science and Technology, 2014.

[6] Chen J, Hu J, Zhou J. Hardware and energy-efficient stochastic LU decomposition scheme for MIMO receivers[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 24 (2015), No. 4 , p. 1391-1401.

[7] LEE V T, ALAGHI A, CEZE L. Correlation manipulating circuits for stochastic computing[C]//2018 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2018: 1417-1422.

[8] Hojabr R, Givaki K, Tayaranian S M R, et al. Skippynn: An embedded stochastic-computing accelerator for convolutional neural networks[C]//2019 56th ACM/IEEE Design Automation Conference (DAC). IEEE, 2019: 1- 6.

[9] Liang Zhewei. Hardware optimization of two nonlinear systems based on stochastic computation [D]. Guangxi Normal University, 2020.Miao L, Chakrabarti C. A parallel stochastic computing system with improved accuracy[C]//SiPS 2013 Proceedings. IEEE, 2013: 195-200.

[10] Alspector J, Gannett J W. A VLSI-efficient technique for generating multiple uncorrelated noise sources and its application to stochastic neural networks[J]. IEEE Transactions on Circuits and Systems, Vol. 38 (1991), No. 1, p. 109-123.

[11] Ichihara H, Ishii S, Sunamori D, et al. Compact and accurate stochastic circuits with shared random number sources[C]//2014 IEEE 32nd International Conference on Computer Design (ICCD). IEEE, 2014: 361-366.

[12] Xie Y, Liao S, Yuan B, et al. Fully-parallel area-efficient deep neural network design using stochastic computing[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 64 (2017), No. 12, p. 1382-1386.

[13] Neugebauer F, Polian I, Hayes J P. S-box-based random number generation for stochastic computing[J]. Microprocessors and Microsystems, Vol. 61 (2018), p. 316-326.

[14] S. M. Shivanandamurthy, I. G. Thakkar and S. A. Salehi, "Work-in-Progress: A Scalable Stochastic Number Generator for Phase Change Memory Based In-Memory Stochastic Processing,"*2019 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2019, pp. 1-2.

[15] Hu J, Li B, Ma C, et al. Spin-hall-effect-based stochastic number generator for parallel stochastic computing[J]. IEEE Transactions on Electron Devices, Vol. 66, No. 8, p. 3620-3627.

[16] Hoe D H K, Pajardo C. Implementing stochastic bayesian inference: Design of the stochastic number generators[C]//2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2019: 1105-1109.

[17] Salehi S A. Low-cost stochastic number generators for stochastic computing[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 28 (2020), No. 4, p. 992-1001.

[18] Abdellatef H, Khalil-Hani M, Shaikh-Husin N, et al. Accurate and compact convolutional neural network based on stochastic computing[J]. Neurocomputing, Vol. 471 (2022), p. 31-47.

[19] Chen J, Hu J, Zhou J. Hardware and energy-efficient stochastic LU decomposition scheme for MIMO receivers[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 24 (2015), No. 4, p. 1391-1401.

[20] Zhakatayev A, Kim K, Choi K, et al. An efficient and accurate stochastic number generator using even-distribution coding[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 37 (2018), No. 12, p. 3056-3066.

[21] Xia Z, Chen J, He S, et al. Neural Synaptic Plasticity-Like Computing: An Ultra-Low Cost Approach for Artificial Neural Networks Implementation[C]//2020 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2020: 1-5.

[22] Wang J, Lin J, Wang Z. Efficient hardware architectures for deep convolutional neural network[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 65 (2017), No. 6, p.1941-1953.

[23] Zhou A, Yao A, Guo Y, et al. Incremental network quantization: Towards lossless cnns with low-precision weights[J]. arXiv preprint arXiv:1702.03044, 2017.

[24] Brown B. D., Card H. C., Stochastic neural computation. I. Computational elements[J], IEEE Transactions on Computers, Vol. 50 (2001), No. 9, p. 891-905.

[25] Parhami B, Yeh C H. Accumulative parallel counters[C]//Conference Record of The Twenty-Ninth Asilomar Conference on Signals, Systems and Computers. IEEE, 1995, 2: 966-970.

[26] Li J, Ren A, Li Z, et al. Towards acceleration of deep convolutional neural networks using stochastic computing[C]//2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2017: 115-120.

[27] Kim K, Lee J, Choi K. Approximate de-randomizer for stochastic circuits[C]//2015 International SoC Design Conference (ISOCC). IEEE, 2015: 123-124.

[28] Sadi M H, Mahani A. Accelerating deep convolutional neural network base on stochastic computing[J]. Integration, Vol. 76 (2021), p. 113-121.

[29] Zhou You, Li Jie, He Guanghui. Design of Addition Operation Circuit and MAX Operation Circuit for High-precision Stochastic Computing Units [J]. Microelectronics and Computers, 2021, 38(7): 1-6.Li Z, Li J, Ren A, et al. HEIF: Highly efficient stochastic computing-based inference framework for deep neural networks[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 38 (2018), No. 8, p.1543-1556.

[30] Ting P S, Hayes J P. Stochastic logic realization of matrix operations[C]//2014 17th Euromicro Conference on Digital System Design. IEEE, 2014: 356-364.

[31] Kim K, Kim J, Yu J, et al. Dynamic energy-accuracy trade-off using stochastic computing in deep neural networks[C]//Proceedings of the 53rd Annual Design Automation Conference. 2016: 1-6.

[32] Ichihara H, Sugino T, Ishii S, et al. Compact and accurate digital filters based on stochastic computing[J]. IEEE Transactions on Emerging Topics in Computing, Vol. 7 (2016), No. 1, p. 31-43.

[33] Chang Y N, Parhi K K. Architectures for digital filters using stochastic computing[C]//2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013: 2697-2701.

[34] Liu Y, Parhi K K. Linear-phase lattice FIR digital filter architectures using stochastic logic[J]. Journal of Signal Processing Systems, Vol. 90 (2018), No. 5, p. 791-803.

[35] Chen J, Hu J, Zhou J. Hardware and energy-efficient stochastic LU decomposition scheme for MIMO receivers[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 24 (2015), No. 4, p. 1391-1401.

[36] Xiong H, He G. Hardware implementation of an improved stochastic computing based deep neural network using short sequence length[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 67 (2020), No. 11, p. 2667-2671.

[37] Yuan B, Wang Y, Wang Z. Area-efficient scaling-free DFT/FFT design using stochastic computing[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 63 (2016), No. 12, p. 1131-1135.

[38] Haselmayr W, Wiesinger D, Lunglmayr M. High-accuracy and fault tolerant stochastic inner product design[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 67 (2019), No. 3, p. 541-545.

[39] Sim H, Lee J. Cost-effective stochastic MAC circuits for deep neural networks[J]. Neural Networks, Vol. 117 (2019), p. 152-162.

[40] Abdellatef H, Khalil-Hani M, Shaikh-Husin N, et al. Low-area and accurate inner product and digital filters based on stochastic computing[J]. Signal Processing, 2021, 183: 108040.

[41] Xia Z, Chen J, Huang Q, et al. Neural Synaptic Plasticity-Inspired Computing: A High Computing Efficient Deep Convolutional Neural Network Accelerator[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 68 (2020), No. 2, p. 728-740.

[42] Xia Z, Dong Y, Chen J, et al. Reconfigurable Neural Synaptic Plasticity-Based Stochastic Deep Neural Network Computing[C]//2021 IEEE Workshop on Signal Processing Systems (SiPS). IEEE, 2021: 229-234.

[43] Liu Y, Liu S, Wang Y, et al. A survey of stochastic computing neural networks for machine learning applications[J]. IEEE Transactions on Neural Networks and Learning Systems, Vol. 32 (2020), No. 7, p. 2809-2824.

[44] Ren A, Li Z, Wang Y, et al. Designing reconfigurable large-scale deep learning systems using stochastic computing[C]//2016 IEEE International Conference on Rebooting Computing (ICRC). IEEE, 2016: 1-7.

[45] Li Z, Ren A, Li J, et al. Dscnn: Hardware-oriented optimization for stochastic computing based deep convolutional neural networks[C]//2016 IEEE 34th International Conference on Computer Design (ICCD). IEEE, 2016: 678-681.

[46] Sim H, Nguyen D, Lee J, et al. Scalable stochastic-computing accelerator for convolutional neural networks[C]//2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2017: 696-701.

[47] Ren A, Li Z, Ding C, et al. Sc-dcnn: Highly-scalable deep convolutional neural network using stochastic computing[J]. ACM SIGPLAN Notices, Vol. 52 (2017), No. 4, p. 405-418.

[48] Li J, Yuan Z, Li Z, et al. Hardware-driven nonlinear activation for stochastic computing based deep convolutional neural networks[C]//2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017: 1230-1236.

[49] Yu J, Kim K, Lee J, et al. Accurate and efficient stochastic computing hardware for convolutional neural networks[C]//2017 IEEE International Conference on Computer Design (ICCD). IEEE, 2017: 105-112.

[50] Li B, Qin Y, Yuan B, et al. Neural network classifiers using a hardware-based approximate activation function with a hybrid stochastic multiplier[J]. ACM Journal on Emerging Technologies in Computing Systems (JETC), Vol. 15 (2019), No. 1, p. 1-21.

[51] Nguyen V T, Luong T K, Le Duc H, et al. An efficient hardware implementation of activation functions using stochastic computing for deep neural networks[C]//2018 IEEE 12th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC). IEEE, 2018: 233-236.

[52] Li J, Yuan Z, Li Z, et al. Normalization and dropout for stochastic computing-based deep convolutional neural networks[J]. Integration, Vol. 65 (2019), p. 395-403.

[53] Lee V T, Alaghi A, Hayes J P, et al. Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing[C]//Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017. IEEE, 2017: 13-18.

[54] Faraji S R, Najafi M H, Li B, et al. Energy-efficient convolutional neural networks with deterministic bit-stream processing[C]//2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2019: 1757-1762.

[55] Zhakatayev A, Lee S, Sim H, et al. Sign-magnitude SC: Getting 10X accuracy for free in stochastic computing for deep neural networks[C]//2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC). IEEE, 2018: 1-6.

[56] Ardakani A, Leduc-Primeau F, Onizawa N, et al. VLSI implementation of deep neural network using integral stochastic computing[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 25 (2017), No. 10, p. 2688-2699.

[57] Neugebauer F, Polian I, Hayes J P. On the maximum function in stochastic computing[C]//Proceedings of the 16th ACM International Conference on Computing Frontiers. 2019: 59-66.

[58] Sim H, Lee J. A new stochastic computing multiplier with application to deep convolutional neural networks[C]//Proceedings of the 54th Annual Design Automation Conference 2017. 2017: 1-6.

[59] Lammie C, Azghadi M R. Stochastic computing for low-power and high-speed deep learning on FPGA[C]//2019 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2019: 1-5.

[60] Kim M, Seo J S. Deep convolutional neural network accelerator featuring conditional computing and low external memory access[C]//2020 IEEE Custom Integrated Circuits Conference (CICC). IEEE, 2020: 1-4.

[61] Zhang Y, Zhang X, Song J, et al. Parallel convolutional neural network (CNN) accelerators based on stochastic computing[C]//2019 IEEE International Workshop on Signal Processing Systems (SiPS). IEEE, 2019: 19-24.

[62] Wang Z, Jia Z, Shen Z, et al. Optimization for Deep Convolutional Neural Network of Stochastic Computing on MLC-PCM-Based System[J]. Microprocessors and Microsystems, 2022: 104505.