MAPPING CONVOLUTIONAL NEURAL NETWORKS ONTO NEUROMORPHIC CHIP FOR SPIKE-BASED COMPUTATION

Chenglong Zou[1,2], Xiaoxin Cui[1], Yisong Kuang[1], Xinan Wang[2]*
[1]Institute of Microelectronics, Peking University, Beijing 100871, China
[2]School of ECE, Peking University Shenzhen Graduate School, Shenzhen 518055, China
*Corresponding Author's Email: cuixx@pku.edu.cn

ABSTRACT

Recent years, spike-based neuron computing on scalable and event-based neuromorphic hardware has demonstrated impressive energy efficiency. In this paper, we propose a novel spiking scheme for 1-bit and 8-bit convolutional neural networks and a systematic mapping algorithm for their deployments on a digital neuromorphic ASIC, with which we can automatically partition input and output feature maps for a 1152*1024 crossbar computing element for a excellent resource efficiency. Experimental results on MNIST dataset show that we can achieve about 98.5% and 99.4% test accuracy for these two kinds of bit-width networks respectively, while the chip can achieve nearly 863 and 174 images/sec real-time inference speed at 0.9 V, 252 MHz.

INTRODUCTION

Convolutional neural networks (CNNs) have been widely used in computer vision such as object recognition and detection task. However, conventional applications on CPU or GPU adopt high precision (32/64 bits) fixed point or floating point computation paradigm, which usually require powerful processing ability with massive memory and energy consumption budget. As a result, this must be a big challenge for deployment of these deep learning models on mobile and embedded hardware with limited resource and memory. In the meantime, spiking neural networks (SNNs) designed as a brain-like system usually use a discretized spike-based representation for signal communication and computing without any multiplication. This kind of neural network can be mapped to specific neuromorphic chip such as IBM' TrueNorth [1] and achieve a quite high energy efficiency using low-bit spike-based computation. For example, a single TrueNorth chip comprised of 1 million neurons and 256 million synapses, can run image classification task for just 70 mW at real-time operation. Intel's Loihi [2] also supports a parallel synapse array, designed in units of spiking neurons, and could achieve several orders of magnitude of energy efficiency compared with common platforms like CPUs or GPUs. However, most of the contemporary neuromorphic hardwares are designed with 2D mesh crossbar structure and usually has a typical block-wise constraint for neuron connectivities and finite combination of synapse weights. According to address protocol of router, spiking neurons in a TrueNorth synaptic core can only communicate with neurons in another core with one axon. Therefore, if we want to achieve multiple fan-in and fan-out for overlapping feature map reuse and high-precision weights, numerous copy neurons are needed. Besides, because of block-wise 256*256 synapse array, TrueNorth chip can't even support a fully connected neural network with three layers (784-500-10) for MNIST dataset. For convolutional neural networks, they have to use group convolution to compress networks, which may cause a degraded accuracy and additional complexity of mapping algorithm. For facility of network deployment, they develop a specific hardware description functions called corelets [3] which can automatically compile the learned network parameters to program TrueNorth chip.

In this paper, we firstly describe a reconfigurable neuromorphic chip architecture alleviated above design drawback and then present a novel spiking scheme for 1-bit and 8-bit convolutional neural networks. Finally, we propose a systematic mapping algorithm for deployment on this chip. Experimental results using LeNet [4] on MNIST dataset show that we can achieve about 98.5% and 99.4% test accuracy for these two networks with kinds of bit-width respectively, while the chip can achieve nearly 863 and 174 images/sec real-time inference speed at 0.9 V, 252 MHz.

METHOD

Chip architecture

In this work, we choose a neuromorphic chip with reconfigurable bit width as our deployment platform. This chip consists of 1152 input axons, 1024 processing spiking neurons and a 1152*1024 neuro-synaptic crossbar (seeing Figure 1). Each neuron has only one programmable weight parameter for strength of synapse connectivity and 1152 on/off synapse switches. With the combination of multiple axons (1,2,4,8), we can achieve a multi-bit (1,2,4,8) input spike packet. In the similar way, we can achieve a multi-bit (1,2,4,8) weight representation with the combination of multiple (1,2,4,8) neurons. Both of them can be configured in advance for different neural network with different bit widths. Moreover, this crossbar can support a temporal axon reuse at most 64 (1,2,4,8,16,32,64) times during a complete computing time step, to achieve a larger feature map input at the cost of decreasing the number of output neurons. For example, we can support a mapping of a largest convolutional kernel of 3*3*2048, and output only

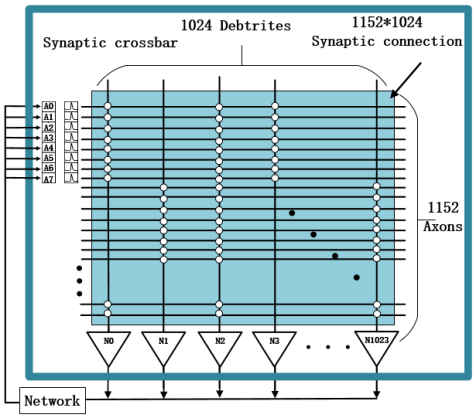

Figure 1. Chip structure

one feature point. Furtherly, we can further reuse this chip to compute for a complete convolutional layer.

Spiking scheme

In this work, we choose a classical convolutional neural network called LeNet (Fig. 3) with two kinds of bit width (1 bit and 8 bit) for demonstrate the mapping process. As mentioned above, our chip actually supports more flexible bit-width choices. For 1-bit LeNet, we adopted a simple binary quantization method in [5] and modified output activation as {0,1} rather than {-1,+1} to satisfy spike signal representation [6], seeing in (1).

$$Spiking(1\ bit) = \begin{cases} 0, \sum_i x_i(t) * w_i + L \le \theta \\ 1, \sum_i x_i(t) * w_i + L > \theta \end{cases} \quad (1)$$

where $x_i(t)$ is the spiking input (either 0 or 1) at time t and w_i is corresponding weight, leakage L and threshold θ was computed from batch normalization layer, one spike will be determined to emit or not, according to the behavior of leaky integrate-and-fire (LIF) spiking neuron model.

For 8-bit LeNet, we firstly trained a full-precision counterpart with ReLU activation by back-propagation algorithm, then we linearly quantize weight parameters to signed 8-bit integer and quantize activations to unsigned 8-bit integer to match multi-bit nonnegative spike signal representation. For its spiking scheme, we configure an 8-bit truncation range to keep each neuron emit at most 8 spikes according to truncated 8 bits as in Figure 2.

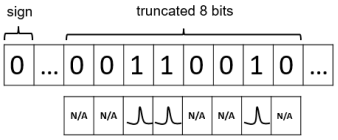

Figure 2. Spiking scheme using 8-bit truncation

Mapping algorithm

For the sake of description, we adopt a series of definition in Table 1 for each configuration parameter. For a standard 2-D crossbar unit with finite inputs and outputs (1152*1024 for our chip), we firstly need to partition input

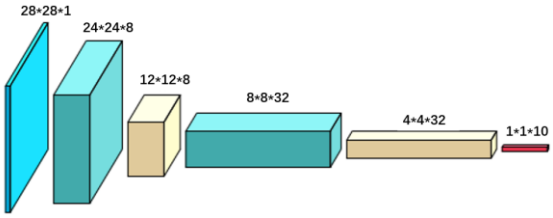

Figure 3. LeNet architecture

feature map into a number of $m*n$ patches to ensure that each patch is not exceeding the number of input axons. Certainly, we can extend for a larger input patch with larger width or height by reusing input axon for f times. Meanwhile, the size of calculated output patch should be also less than the number of output of axons. Hence there is a tradeoff between the size of input patch and output patch, larger input patch using axon extension may cause larger output patch, which might not match the reduced output neurons. Finally, we aim to map different patches of a convolutional layer or pooling layer on such a synaptic crossbar unit during multiple complete computing time steps with chip-level resource reuse, while our chip can show an excellent resource utilization efficiency in every complete computing time step.

For a systematic and modular mapping mechanism and relieving burden of spike signal multicast, we propose three basic criteria on mapping algorithm summarized as below:

1) Each input patch should be symmetrical and involve all channels of input feature maps, and satisfy a maximum axon occupation and input load balancing.

2) Each output patch should be also symmetrical and involve all channels of output feature maps, and satisfy a maximum neuron occupation and output load balancing.

3) Each of activations from input feature maps should contribute to a neuron output with a complete computation of convolution or pooling operation.

For a convolutional or pooling layer, we can conclude following constraints seeing (2) (3).

$$w_l * h_l * d_l \le \frac{1152 * f}{k}, \quad w_{l+1} * h_{l+1} * d_{l+1} \le \frac{1024}{f * k} \quad (2)$$

$$w_{l+1} = \frac{w_l - c_{l+1}}{s_{l+1}} + 1, \quad h_{l+1} = \frac{h_l - c_{l+1}}{s_{l+1}} + 1 \quad (3)$$

If we keep each patch equal, horizontal and vertical patches can be calculated as (4) and (5).

$$W_l = w_l * m_l - (m_l - 1) * (c_{l+1} - s_{l+1}) \quad (4)$$

$$H_l = h_l * n_l - (n_l - 1) * (c_{l+1} - s_{l+1}) \quad (5)$$

Therefore, the total number of patches or complete time steps will be $m*n$. By the way, we use a convolution with stride 2 to replace a pooling, which was proved to be feasible. Refer to above inequality constraints, we can obtain a proper w, h and f for each layer using a simple grid search algorithm, and furtherly figure out hardware resource overhead and computing time of each patch.

Table 1. Parameter definitions

w, h	Patch width and height		***m***	Horizontal patchs
W, H	Feature width and height		***n***	Vertical patch
d	Feature depth		***c***	Convolution/pooling width/height
f	Axon extension		***s***	Stride for convolution or pooling
l	l-th layer		***patch***	Partial feature map
k	Quantization bit width		***overlap***	Overlapping between patches

EXPERIMENTAL RESULTS

Network configuration

For 1-bit convolutional neural networks, we can employ only one axon for one input activation point and two neurons with respective weights of {-1,+1} for one output activation. It should be noted that feature activation point is represented as a spike signal in our chip. For 8-bit convolutional neural networks, we use eight axons for one input activation point and eight weighted neurons for one output activation point, each axon or neuron represent each bit in 8-bit spike-based feature activation or kernel weight. By searching optimized parameters include *w*, *h* and *f*, we summarize final configuration information for both two kinds of spiking LeNet as in Table 2 and Table 3:

Table 2. 1-bit LeNet configuration

Feature map	Patch size	*f*	Time steps
28*28*1	12*12*1	1	9
24*24*8	12*12*8	1	4
12*12*8	8*8*8	1	4
8*8*32	4*8*32	1	2

Table 3. 8-bit LeNet configuration

Feature map	Patch size	*f*	Time steps
28*28*1	8*8*1	1	36
24*24*8	4*8*8	2	18
12*12*8	5*6*8	2	32
8*8*32	2*4*32	2	8

Accuracy and speed

We adopt an original STE training and quantization algorithm [7] for a 1-bit LeNet, and a simple post-training quantization algorithm for the 8-bit one. Their accuracy performances on MNIST test set can be found in Table 4. According to above mapping workflow, we can achieve a final 98.5% and 99.4% test accuracy for these two kinds of bit-width networks respectively, while our chip can achieve nearly 863 images/sec and 174 images/sec real-time classification speed at 0.9 V, 252 MHz.

Table 4. Real-time classification accuracy and speed for 1/8-bit LeNet

	accuracy	speed
1-bit LeNet	98.5%	863 images/sec
8-bit LeNet	99.4%	174 images/sec

CONCLUSION

In this paper, we present a reconfigurable neuromorphic chip with a delicate architecture which alleviated block-wise connection drawbacks and propose a novel spiking scheme for 1-bit and 8-bit convolutional neural networks. More importantly, we develop a systematic mapping algorithm to automatically partition feature patch on our chip. Experimental results show that the spike-based 1/8-bit LeNet on MNIST dataset achieved about 98.5% and 99.4% test accuracy respectively, while the chip can achieve nearly 963 and 174 images/sec real-time inference speed at 0.9 V, 252 MHz.

ACKNOWLEDGEMENTS

This work is supported by National Key Research and Development Project (No.2018YFB2202605), R&D Project of Shenzhen Science and Technology Innovation Committee (No. JCYJ20180503182125190 and JCYJ20200109120404043) and the 111 Project (B18001).

REFERENCES

[1] F. Akopyan et al., "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 10, pp. 1537-1557, Oct. 2015.

[2] M. Davies et al., "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," in IEEE Micro, vol. 38, no. 1, pp. 82-99, January/February 2018.

[3] Lecun Y , Bottou L . Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[4] A. Amir et al., "Cognitive computing programming paradigm: A Corelet Language for composing networks of neurosynaptic cores," The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1-10 (2013).

[5] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized neural networks. In Advances in neural information processing systems, pages 4107–4115 (2016).

[6] Esser S K , Merolla P A , Arthur J V , et al. Convolutional Networks for Fast, Energy-Efficient Neuromorphic Computing[J]. Proc Natl Acad Sci U S A, 2016, 113(41):11441-11446.

[7] C. Zou, X. Wang, B. Xu, Y. Kuang and X. Cui, "Deep Spiking Convolutional Neural Networks for Programmable Neuro-synaptic System," 2019 IEEE 13th International Conference on ASIC (ASICON), Chongqing, China, 2019, pp. 1-4.