Memristor-based vector neural network architecture*

Hai-Jun Liu(刘海军), Chang-Lin Chen(陈长林), Xi Zhu(朱熙),
Sheng-Yang Sun(孙盛阳), Qing-Jiang Li(李清江), and Zhi-Wei Li(李智炜)†

College of Electronic Science, National University of Defense Technology, Changsha 410073, *China*

(Received 25 September 2019; revised manuscript received 11 November 2019; accepted manuscript online 27 December 2019)

Vector neural network (VNN) is one of the most important methods to process interval data. However, the VNN, which contains a great number of multiply-accumulate (MAC) operations, often adopts pure numerical calculation method, and thus is difficult to be miniaturized for the embedded applications. In this paper, we propose a memristor based vector-type backpropagation (MVTBP) architecture which utilizes memristive arrays to accelerate the MAC operations of interval data. Owing to the unique brain-like synaptic characteristics of memristive devices, *e.g.*, small size, low power consumption, and high integration density, the proposed architecture can be implemented with low area and power consumption cost and easily applied to embedded systems. The simulation results indicate that the proposed architecture has better identification performance and noise tolerance. When the device precision is 6 bits and the error deviation level (EDL) is 20%, the proposed architecture can achieve an identification rate, which is about 92% higher than that for interval-value testing sample and 81% higher than that for scalar-value testing sample.

Keywords: memristor, memristive devices, vector neural network, interval

PACS: 85.35.–p, 87.19.lv, 07.05.Mh **DOI:** 10.1088/1674-1056/ab65b5

1. Introduction

Interval data processing, an important technology, possesses broad application prospects in signal classification, speech recognition, image processing, *etc*. In many cases, the data to be processed are uncertain, and this uncertainty could be described by an interval or a vector. For example, for an electronic support measure (ESM) system, due to the diversity of the radar emitter parameters, the complexity of electromagnetic environment, and the error of measurement device itself, the measured radar emitter parameters, such as radio frequency (RF), pulse width (PW), pulse repetition interval (PRI), *etc*., would be expressed in the form of intervals.[1–3] Various approaches, including interval fuzzy logic system,[4] neural network,[5] are proposed to solve the classification problem of interval data. Among those methods, the vector neural network (VNN) is the most effective.[1,2] To the best of our knowledge, the current VNN implemented by software algorithm is mainly based on the general-purpose computing platform, which is a purely numerical calculation method without considering the problem of hardware implementation. As a result, the computation cost makes it difficult to realize the miniaturization and embedded system.

Memristive device possesses advantages of small size, low power consumption, and high integration density,[6–10] which are very suitable for building a brain-inspired computing system.[11–14] Therefore, various network architectures based on memristors, such as SNN, MLP, and CNN, have been proposed and achieved good results in many application fields.[15–19] However, the above networks can only handle scalar type data. When the input data is interval-value, those methods will not be workable.

To solve the above-mentioned problems of interval data classification, in this paper we propose a memristor-based vector-type backpropagation (MVTBP) architecture. Through integrating together the advantages of memristor and vector neural network, the multiplication and accumulation of interval type data can speed up effectively.

The rest of this paper is organized as follows. In Section 2, the memristor based VNN architecture is described in detail. In Section 3, two simulation experiments are carried out to verify the identification performance and analyze the influence of different device precision. Finally, some conclusions are drawn from the present study in Section 4.

2. Memristor based vector neural network architecture

The MVTBP architecture is shown in Fig. 1. In the MVTBP architecture, the input of each neuron is a vector of the interval type, and the output is also a vector of the interval type. Figure 1(a) shows a schematic diagram of the MVTBP network, where $x_{\mathrm{pi}} = \left[x_{\mathrm{pi}}^{\mathrm{L}}, x_{\mathrm{pi}}^{\mathrm{U}}\right]$ represents the i-th interval input value of each input node, $i = 1, 2, \ldots, n$, with n denoting the node number of input layer, $z_{\mathrm{pj}} = \left[z_{\mathrm{pj}}^{\mathrm{L}}, z_{\mathrm{pj}}^{\mathrm{U}}\right]$ represents the j-th interval output value of each hidden node, $j = 1, 2, \ldots, k$, with k being the node number of hidden layer, $y_{\mathrm{pq}} = \left[y_{\mathrm{pq}}^{\mathrm{L}}, y_{\mathrm{pq}}^{\mathrm{U}}\right]$ represents the q-th interval output value of each output node,

*Project supported by the National Natural Science Foundation of China (Grant Nos. 61471377, 61804181, 61604177, and 61704191).
†Corresponding author. E-mail: lizhiwei@nudt.edu.cn

and $q = 1,2,\ldots,m$, with m referring to the node number of output layer. The key of the MVTBP architecture is to use memristive arrays to realize the multiplication and accumulation operations of interval type data as shown in Fig. 1(a), the pink virtual box represents the interval process of non-linear mapping from the space of input feature vector to the space of the hidden layer, while the green solid box represents interval process of non-linear mapping from the space of the hidden layer to the space of identification or classification type.

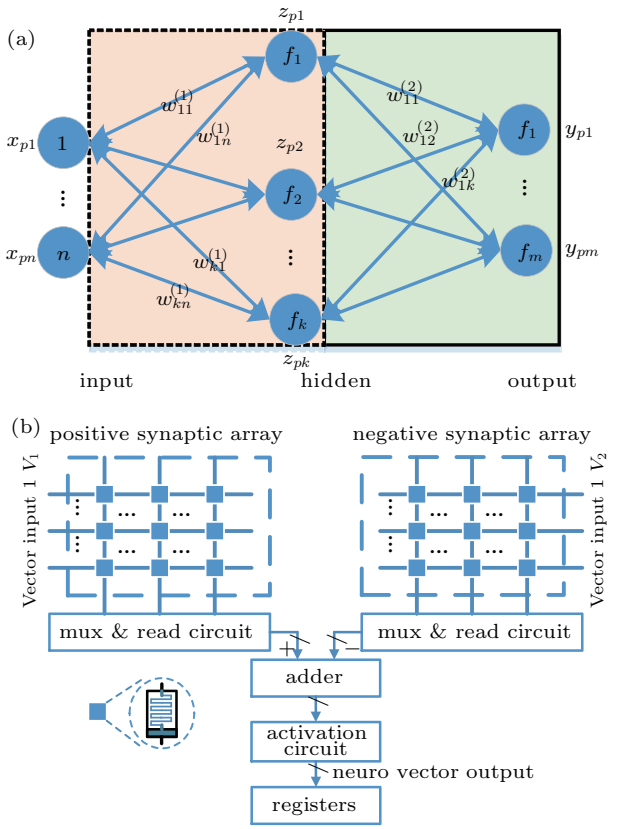

Fig. 1. MVTBP architecture: (a) schematic diagram of MVTBP network and (b) hardware implementation for realizing interval type data multiplicative and accumulative operations by using memristive synaptic arrays.

Figure 1(b) describes the hardware implementation for realizing multiplicative and accumulative operation of interval type data by using memristive synaptic arrays. It can be seen from Fig. 1(b) that the hardware architecture consists of five main components, namely, memristive synaptic array, multiplexer and read circuit, adder, activation circuit, and register. The memristive synaptic array is further divided into a positive synaptic array and a negative synaptic array, which are respectively used for realizing the multiplication and accumulation operations of the upper and lower limit of the interval data, and each intersection of the synaptic array is realized by a memristive device. The multiplexer and read circuit are used to select the column to be calculated and obtain the voltage value. The adder is devoted to computing the voltage sum or difference of the positive and negative synaptic array. The activation circuit is used for calculating the activation function, such as *sigmoid* function, *etc*., and the register is used to save the calculation results for the next layer.

It should be noted that the input vectors V_1 and V_2 in Fig. 1(b), applied to the positive and negative synaptic arrays respectively, are complementary to each other. That is, when the input vector V_1 denotes the upper limit of the interval-value x_{pi}, *i.e*., $x_{\mathrm{pi}}^{\mathrm{U}}$, the input vector V_2 should be the lower limit of the interval-value x_{pi}, *i.e*., $x_{\mathrm{pi}}^{\mathrm{L}}$, and the neuron vector output is the upper limit vector $z_{\mathrm{pj}}^{\mathrm{U}}$; when the input vector V_1 represents the lower bound of the interval-value x_{pi}, *i.e*., $x_{\mathrm{pi}}^{\mathrm{L}}$, the input vector V_2 should be the upper bound of the interval-value x_{pi}, *i.e*., $x_{\mathrm{pi}}^{\mathrm{U}}$, then the neuron vector output is the lower bound vector $z_{\mathrm{pj}}^{\mathrm{L}}$. The positive memristive synaptic array stores all the positive weights, while the negative memristive synaptic array stores the absolute values of all negative weights, and the sign of the negative weights is reflected in the subtraction portion of the adder. The positive and negative weight in the same row together constitute the weight value of the MVTBP network, and the negative weight is subtracted from the positive weight.

The above hardware architecture mainly realizes the multiplicative and accumulative operations of interval data. Taking the hidden layer of the 3-layer MVTBP network for example, and the neuron output of the hidden layer can be computed as follows:

$$z_{\mathrm{pj}} = \left[z_{\mathrm{pj}}^{\mathrm{L}}, z_{\mathrm{pj}}^{\mathrm{U}}\right] = \left[f\left(net_{\mathrm{pj}}^{\mathrm{L}}\right), f\left(net_{\mathrm{pj}}^{\mathrm{U}}\right)\right], \tag{1}$$

where z_{pj} represents the j-th interval output value of each hidden node, $j = 1,2,\ldots,k$, k denotes the node number of the hidden layer, f is the activation function of the neurons, such as the *sigmoid* function, *etc*., $net_{\mathrm{pj}}^{\mathrm{L}}$ and $net_{\mathrm{pj}}^{\mathrm{U}}$ refer to the lower and upper limit of the input interval-value of the hidden neuron respectively, and their calculation process is shown below:

$$net_{\mathrm{pj}}^{\mathrm{L}} = \sum_{\substack{i=1 \\ w_{ji}^{(1)} \geqslant 0}}^{n} w_{ji}^{(1)} x_{\mathrm{pi}}^{\mathrm{L}} + \sum_{\substack{i=1 \\ w_{ji}^{(1)} < 0}}^{n} w_{ji}^{(1)} x_{\mathrm{pi}}^{\mathrm{U}}, \tag{2}$$

$$net_{\mathrm{pj}}^{\mathrm{U}} = \sum_{\substack{i=1 \\ w_{ji}^{(1)} \geqslant 0}}^{n} w_{ji}^{(1)} x_{\mathrm{pi}}^{\mathrm{U}} + \sum_{\substack{i=1 \\ w_{ji}^{(1)} < 0}}^{n} w_{ji}^{(1)} x_{\mathrm{pi}}^{\mathrm{L}}, \tag{3}$$

where $w_{ji}^{(1)}$ denotes the weight between the i-th input node and the j-th hidden node, and superscript 1 represents the first layer of the network. It is noticed that the calculation process for the hidden layer and that for the output layer are similar to each other.

The main purpose of backward propagation of the MVTBP architecture is also to compute the weight (such as $w_{ji}^{(1)}$, *etc*.) correction according to the output error, and we can use a similar method to that in Refs. [1,2] to update the weight of the memristive synaptic array.

3. Results and discussion

In this section, the five-emitter-identification (EID) problem in Ref. [1] is taken for example to verify the performance of the proposed MVTBP architecture. For the five-EID problem, there are 50 interval-value and scalar-value training samples in the training step (each emitter type has 10 training samples) respectively, and there are also 150 interval-value and scalar-value test samples in the testing step (each emitter type has 30 testing samples). It should be noted that all the simulated emitter samples are from the radar manual[2] or the emitter samples repository,[1] which stores the measured emitters of five types. The parameters of the emitter samples are RF, PRI and PW, and can be seen in detail in Ref. [1]. So, these samples can represent the actual application situation to a certain extent.

In order to facilitate the hardware implementation, we normalize the emitters' parameters, making them situated in a range between 0 and 1. In addition, to ensure the convergence of the network, we also increase the node number of hidden layers and employ the MVTBP architecture with a 3–30–5 network. For verifying the adaptability of the proposed MVTBP architecture to different measurement errors, it is also necessary to test interval emitter samples with additive noise at different levels. By introducing the error deviation level (EDL) proposed in Ref. [2], we can generate noisy testing samples by adding noise to the samples without noise. In this section, the noisy interval-value testing samples with different EDLs (from 0 to 20%) are given to verify the performance of the MVTBP architecture.

3.1. Identification performance analysis

In order to verify the identification performance under a similar neural network architecture, the NVTBP algorithm is selected[2] as the comparison objects in this subsection. The identification performance on interval-value and scalar-value testing samples are shown in Figs. 2 and 3, respectively.

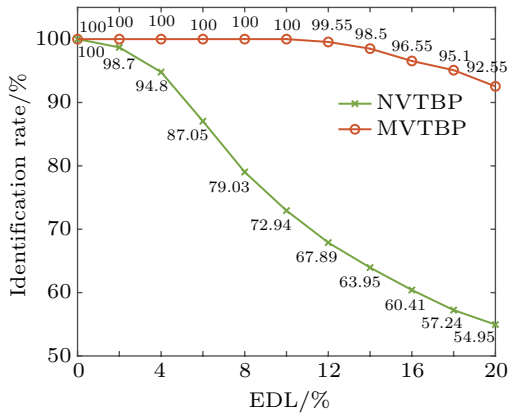

Fig. 2. Performance of NVTBP and MVTBP on interval-value testing sample.

Compared with the NVTBP algorithm, the MVTBP architecture has general advantages in both low EDL and high EDL situation as can be seen in Fig. 2. When the EDL arrives at 10%, the identification rate of the MVTBP architecture could still achieve 100% whereas the identification rate of the NVTBP algorithm is only about 72.94%. When the EDL is equal to 20%, the identification rate of the MVTBP architecture drops to 92.55%, which is about 37% higher than that of the NVTBP algorithm. To further verify the performance of the proposed architecture, we also introduce the CVNN algorithm[1] with different network structures for comparison. Compared with the CVNN algorithm, the MVTBP architecture has clear superiority. When the EDL is equal to 20%, the identification rate of the CVNN algorithm is only 77.67%, while the identification rate of the MVTBP architecture is about 15% higher.

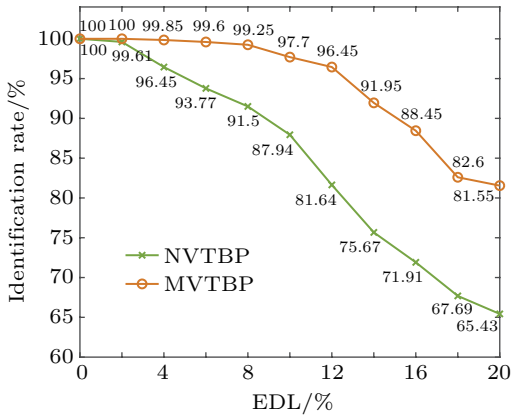

Fig. 3. Performance of NVTBP and MVTBP on scalar-value testing sample.

From Fig. 3, it can be seen that with the increase of EDL, the performances of both methods decrease, but the MVTBP architecture drops more slowly. When the EDL reaches 10% and 20% separately, the identification rate of the MVTBP architecture achieves 100% and 81.55% respectively, which is about 10% and 16% higher than that of the NVTBP algorithm. In this scenario, we also chose the CVNN algorithm[1] for comparison. Compared with the CVNN algorithm, the MVTBP architecture also has universal superiority, especially in the case of high EDL. As the EDL is 20%, the MVTBP architecture is about 15% higher than that of the CVNN algorithm.

From Figs. 2 and 3, it can be concluded that the MVTBP architecture has better noise adaptability, in the case of high EDL. The identification rate of MVTBP could be higher than 92% for interval-value testing sample and 81% for scalar-value testing sample.

In terms of resource consumption, the MVTBP architecture requires only 480 ($3 \times 30 \times 2 + 30 \times 5 \times 2$) memristive devices, which is far less than those required by the traditional

numerical calculation method, so the proposed architecture is very suitable for miniaturization and embedded applications.

3.2. Influence of device precision

Since the resistance state of the actual memristive device is limited, it is of great practical significance to discuss the radar emitter identification performances of the memristors with different precisions (*i.e.*, the resistance state of the memristive device). Assuming that the adjustable precision of the memristive device is N bits, the adjustable state S of its resistance value is $S = 2^N$. For example, when the precision N of the memristive device is 6 bits, the corresponding adjustable resistance state number is $2^6 = 64$.

In this subsection, the identification performance of different N bits (*i.e.*, *Software,* 10 bits, 8 bits, 6 bits, 5 bits, 4 bits) is discussed. The item *Software* denotes adopting default arithmetic precision of Matlab, that is, the double-precision 16-bit effective number. The identification results on interval-value and scalar-value testing samples are discussed as follows. As can be seen from Table 1 and Table 2, with the increase of EDL, the identification rate of the MVTBP architecture decreases gradually.

Table 1. Performances on interval-value testing samples with different precisions of memristive device.

Error deviation level/%	*Software*	10 bits	8 bits	6 bits	5 bits	4 bits
20	92.55	92.99	92.91	92.94	83.13	52.53
18	95.10	94.78	94.57	93.59	83.56	53.61
16	96.55	96.34	95.98	94.98	84.09	54.74
14	98.50	98.52	98.06	96.86	84.38	55.79
12	99.55	99.74	99.63	98.96	84.56	57.11
10	100	99.95	99.97	99.82	84.48	58.26
8	100	100	100	100	84.78	59.47
6	100	100	100	100	84.53	59.95
4	100	100	100	100	84.03	60
2	100	100	100	100	83.95	60
0	100	100	100	100	83.50	60

As the precision N of the memristive device decreases, the identification rate of the MVTBP architecture does not change obviously at the beginning, and then suddenly drops at the ending. When $N = 6$ bits, that is, the number of adjustable resistance state is 64, it shows better identification performance, almost the same as the high precision case. When $N = 5$ bits, for interval-value testing samples, the identification rate is about 84%, and remains basically unchanged with the increase of EDL, while for scalar-value testing samples, the identification rate decreases with EDL increasing. When N arrives at 4 bits, the identification rate drops to about 50%–60%. It can be seen from the above analysis that in order to make the emitter identification have better performance, the condition that the precision N of the memristive device should be at least 6 bits is required to be satisfied.

Table 2. Performances on scalar-value testing samples with different precisions of memristive device.

Error deviation level/%	*Software*	10 bits	8 bits	6 bits	5 bits	4 bits
20	81.55	80.27	79.74	76.01	69.80	47.73
18	82.60	84.47	82.13	79.66	72.80	50.32
16	88.45	88.39	86.77	83.52	76.60	52.55
14	91.95	92.79	91.93	88.99	79.54	54.57
12	96.45	96.34	95.49	93.69	83.71	57.63
10	97.70	98.34	97.80	97.06	86.89	58.83
8	99.25	99.42	99.24	99.10	88.45	59.48
6	99.60	99.60	99.61	99.73	88.35	60
4	99.85	99.87	99.81	99.92	89.00	60
2	100	100	100	100	88.20	60
0	100	100	100	100	88	60

For the feasibility of the MVTBP architecture, we verify the identification performance and analyze the influence of different device precision. Besides, we have designed neuron circuits to generate the impulses needed for the network and built a quasi-analytical model of three-dimensional (3D) vertical-RRAM array architecture for MB-level design in the early stage,[20] in which discussed were the influences of array device parameters, such as resistance state, device nonlinearity, reading voltage, array scale, *etc*., on reading and writing performance. All of these can provide a benchmark for the practical application of this architecture. In addition, with the development of device fabrication technology, the performance of memristive array can be improved. Therefore, we believe that the proposed MVTBP architecture will be feasible in practice in the future.

In summary, this proposed MVTBP architecture, can successfully process the interval-value and scalar-value data with noise, and is very suitable for building a specialized embedded system, such as electronic reconnaissance system, speech recognition system, image processing system, *etc*., because in many cases, these systems have relatively high requirements for size, power consumption, *etc*., and the measured input data are generally uncertain.

4. Conclusions

Aiming at the identification problem of interval type data and the requirements of embedded application, in this paper we design an MVTBP architecture, which can realize multiplicative and accumulative operations of interval type data. The results of simulation experiments verify the performance of the proposed MVTBP architecture, and show that the proposed architecture has better identification performance and noise tolerance, and the device precision of 6 bits can meet the requirement of emitter identification. The research in this

paper can also provide support for extending memristor-based network applications.

Of course, there is also some room for further improvement of the MVTBP architecture, such as the resistance fluctuation, yield, etc. In addition, considering other cases such as image classification, the designing of different network architectures is also an important research hotspot. These problems can be realized by improving the device preparation process, optimizing the network architecture design and the like.

References

[1] Liu H J, Liu Z, Jiang W L and Zhou Y Y 2010 *IET Signal Process.* **4** 137
[2] Shieh C and Lin C 2002 *IEEE T. Anten. Propag.* **50** 1120
[3] Chen X, Li D, Yang X and Li H 2018 *Int. J. Aeronaut. Space* **19** 685
[4] Chen X and Hu W D 2012 *Electron. Lett.* **48** 1156
[5] Sun J, Xu G, Ren W and Yan Z 2018 *IET Radar Sonar. Nav.* **12** 862
[6] Chua L 1971 *IEEE Trans. Circuit Theory* **18** 507
[7] Strukov D B, Snider G S, Stewart D R and Williams R S 2008 *Nature* **453** 80
[8] Sun Y, Xu H, Liu S, Song B, Liu H, Liu Q and Li Q 2018 *IEEE Electron. Dev. Lett.* **39** 492
[9] Hua-Gan W, Bo-Cheng B and Mo C 2014 *Chin. Phys. B* **23** 118401
[10] Wang S, He C, Tang J, Yang R, Shi D and Zhang G 2019 *Chin. Phys. B* **28** 017304
[11] Upadhyay N K, Jiang H, Wang Z, Asapu S, Xia Q and Yang J J 2019 *Adv. Mater. Technol.* **4** 1800589
[12] Burr G W, Shelby R M, Sebastian A, Kim S, Kim S, Sidler S, Virwani K, Ishii M, Narayanan P, Fumarola A, Sanches L L, Boybat I, Le Gallo M, Moon K, Woo J, Hwang H and Leblebici Y 2017 *Adv. Phys. X* **2** 89
[13] Li C, Wang Z, Rao M, Belkin D, Song W, Jiang H, Yan P, Li Y, Lin P, Hu M, Ge N, Strachan J P, Barnell M, Wu Q, Williams R S, Yang J J and Xia Q 2019 *Nat. Mach. Intell.* **1** 49
[14] Zhou E, Fang L, Liu R and Tang Z 2017 *Chin. Phys. B* **26** 118502
[15] Li Z, Chen P, Xu H and Yu S 2017 *IEEE T. Electron. Dev.* **64** 2721
[16] Zhang X, Wang W, Liu Q, Zhao X, Wei J, Cao R, Yao Z, Zhu X, Zhang F, Lv H, Long S and Liu M 2018 *IEEE Electron. Dev. Lett.* **39** 308
[17] Li Y, Zhong Y, Zhang J, Xu L, Wang Q, Sun H, Tong H, Cheng X and Miao X 2015 *Sci. Rep.* **4** 4096
[18] Cai F, Correll J M, Lee S H, Lim Y, Bothra V, Zhang Z, Flynn M P and Lu W D 2019 *Nature Electron.* **2** 290
[19] Sun S, Xu H, Li J, Li Q and Liu H 2019 *IEEE Access.* **7** 61679
[20] Li Z, Chen P, Liu H, Li Q, Xu H and Yu S 2017 *IEEE T. Electron. Dev.* **64** 1568