A Low-cost Artificial Neural Network Model for Raspberry Pi

Son Ngoc Truong
Faculty of Electrical and Electronics Engineering
HCMC University of Technology and Education
HCM City, Vietnam
sontn@hcmute.edu.vn

Abstract—**In this paper, a ternary neural network with complementary binary arrays is proposed for representing the signed synaptic weights. The proposed ternary neural network is deployed on a low-cost Raspberry Pi board embedded system for the application of speech and image recognition. In conventional neural networks, the signed synaptic weights of –1, 0, and 1 are represented by 8-bit integers. To reduce the amount of required memory for signed synaptic weights, the signed values were represented by a complementary binary array. For the binary inputs, the multiplication of two binary numbers is replaced by the bit-wise AND operation to speed up the performance of the neural network. Regarding image recognition, the MINST dataset was used for training and testing of the proposed neural network. The recognition rate was as high as 94%. The proposed ternary neural network was applied to real-time object recognition. The recognition rate for recognizing 10 simple objects captured from the camera was 89%. The proposed ternary neural network with the complementary binary array for representing the signed synaptic weights can reduce the required memory for storing the model's parameters and internal parameters by 75%. The proposed ternary neural network is 4.2, 2.7, and 2.4 times faster than the conventional ternary neural network for MNIST image recognition, speech commands recognition, and real-time object recognition respectively.**

Keywords-artificial neural network; deep learning; speech recognition; image recognition; ternary neural networks

I. INTRODUCTION

Artificial Neural Networks (ANNs) and deep learning have achieved impressive successes in fields such as image recognition, speech recognition, and prediction [1-6]. ANNs are computationally expensive because they are composed of a huge number of computational tasks and internal parameters. ANNs are often implemented on high-performance CPUs (Central Processing Units) and GPUs (Graphics Processing Units) rather than low-cost embedded systems [7]. Various models of ANNs have been proposed for low-cost embedded systems such as binary neural networks and ternary neural networks [8-14]. Binary neural networks are the optimized models of neural networks that constraint the synaptic weights to the binary space {–1, 1} [8-11]. In a binary neural network, the conventional 32-bit floating-point multipliers are replaced by the logical XNOR operation to speed up performance. However, their accuracy is lower than full-precision neural networks because only one bit is used to represent the synaptic weight and activation function. To increase speed and accuracy, ternary neural networks that constraint the synaptic weights to the ternary space {–1, 0, 1} have been proposed [12-15]. The accuracy of the ternary and binary neural networks is slightly lower than the one of full-precision neural networks. However, the required memory is reduced significantly. Ternary neural networks are suitable to be implemented on low-cost embedded systems [14]. For a ternary neural network, the signed values of –1, 0, and 1 require 8-bit width memories and the 8-bit multipliers. In this paper, a method for representing the signed ternary synaptic weights by using the binary numbers 0 and 1 is proposed. The multipliers are replaced by the logical AND operations to reduce the required storage space and speed up the performance of the network. The proposed ternary neural network is deployed on a Raspberry Pi board for a mobile robot for object recognition and speech command recognition.

Corresponding author: Son Ngoc Truong

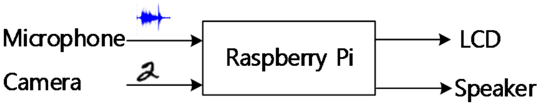

Fig. 1. Block diagram of the control unit for a mobile robot using the Raspberry Pi board

Figure 1 shows a conceptual diagram of a control unit for a mobile robot. The control unit is implemented on a low-cost Raspberry Pi board. The system performs the tasks of pattern recognition including image recognition, speech command recognition, and real-time object recognition.

II. THE PROPOSED METHOD FOR REPRESENTING THE SIGNED TERNARY SYNAPTIC WEIGHTS

A. Ternary Neural Networks

A ternary neural network is an optimized model of ANNs with the weights constrained to –1, 0, and +1 to reduce the amount of required memory for storing the model's parameters [12-15]. The synaptic weights are quantized by 2 bits. It should be noted that the negative synaptic weights are necessary because the synapses are either excitatory or inhibitory [16-18].

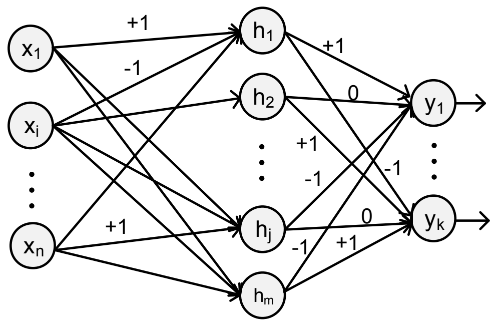

Fig. 2. The conceptual diagram of a ternary neural network, where the synaptic weights are -1, 0, or +1

Figure 2 shows a conceptual ternary neural network, where the synaptic weights are –1, 0, or +1. To represent the signed values of –1, 0, or +1, an 8-bit width memory can be used instead of the 32-bit width memory used in full-precision neural networks, so the amount of required memory for the model parameters of ternary neural networks is less than the full-precision neural networks. Though the amount of required memory is reduced, the ternary neural network is still far from the capability of a low-cost embedded system such as the Raspberry Pi board. The huge number of computation tasks (additions and multiplications) makes ternary neural networks too much complicated to be deployed on a low-cost embedded system. In this work, a complementary binary array is proposed to represent the signed synaptic weights. The multiplication is replaced by the bitwise AND operation to speed up the performance of the network.

B. Proposed Method

A complementary binary array to represent the signed synaptic weights is proposed. For the binary inputs, the output of the j^{th} neuron in the hidden layer can be calculated by (1) [19]:

$$\begin{aligned} h_j &= f(a_j) \\ a_j &= \sum_{i=1}^{n} x_i w_{i,j} \end{aligned} \qquad (1)$$

where f is an activation function, x_i is the i^{th} input, $w_{i,j}$ is the synaptic weight representing the connection strength between the i^{th} input neuron and the j^{th} neuron in the hidden layer. Equation (1) can be rewritten as:

$$\begin{aligned} h_j &= f(a_j) \\ a_j &= \sum_{i=1}^{n} x_i w_{i,j} = \sum_{i=1}^{n} x_i \left(w_{i,j}^{+} - w_{i,j}^{-} \right) \\ a_j &= \sum_{i=1}^{n} x_i w_{i,j}^{+} - \sum_{i=1}^{n} x_i w_{i,j}^{-} \\ &\text{where } w_{i,j} = w_{i,j}^{+} - w_{i,j}^{-} \end{aligned} \qquad (2)$$

In (2), the signed values of –1, 0, and +1 can be represented by the binary numbers 0 and 1. For example, $w_{i,j}$=–1 can be represented by $w^{+}_{i,j}$=0 and $w^{-}_{i,j}$=1. Similarly, $w_{i,j}$=+1 can be represented by $w^{+}_{i,j}$=1 and $w^{-}_{i,j}$=0. By doing this, two binary numbers are used to store a signed value instead of using an 8-bit number. As a result, the amount of required memory is reduced dramatically. Furthermore, for the binary input, the multiplication of two binary numbers can be performed by the bit-wise AND operation to reduce the computational task. The concept of the proposed complementary binary array for representing the signed ternary weights is shown in Figure 3.

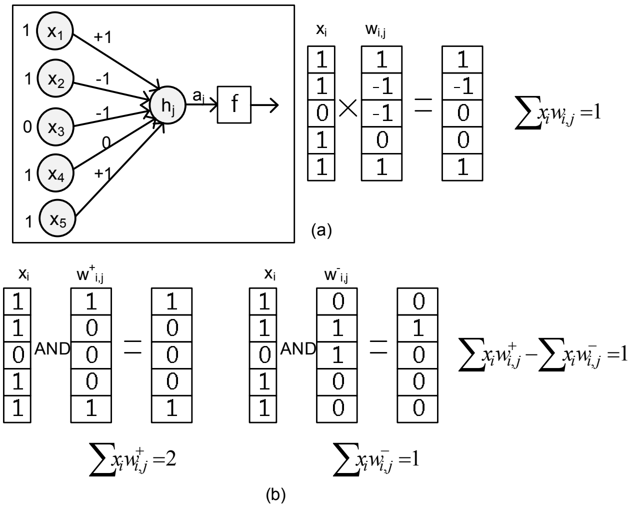

Fig. 3. (a) The output of a neuron with signed ternary synapses. Here multiplications are used to weight the inputs. (b) The concept of the proposed complementary binary array for computing the output of a ternary neuron, where the bitwise AND operation is used to calculate the neuron's output.

Figure 3(a) shows the performance of calculating the output of a conventional ternary neural network. The multiplications are used to perform the synaptic weighting. It requires a huge number of computational tasks. Figure 3(b) presents the proposed method for storing the signed parameters and perform synaptic weighting. The signed values of –1, 0, and 1 are stored in two complementary binary arrays. If the weight is 1, the value of w^{+} and w^{-} are respectively 1 and 0. If the weight is –1, the value of w^{+} and w^{-} are respectively 0 and 1. By doing this, we need only two 1-bit memory cells for representing the signed value of –1, 0, or +1. Furthermore, the multiplication of two binary numbers can be replaced by the bit-wise AND operation, as shown in Figure 3(b). The multiplications are omitted. The proposed ternary neural network can be deployed on low-cost embedded systems effectively. In Figure 3(b), the summation of weighted inputs is performed by counting the "1" bits (population counting) in the result of the bit-wise AND operation. Employing two 1-bit memory cells to represent the signed synaptic weights can reduce the amount of memory dramatically. The proposed ternary neural network is effective for low-cost embedded systems used for mobile robots.

III. EXPERIMENTAL RESULTS

The proposed ternary neural network with the complementary binary array representing the signed synaptic weights is deployed on the Raspberry Pi board for the applications of speech recognition, image recognition, and real-time object recognition. For image recognition, a three-layer neural network is deployed on the Raspberry Pi. The network is trained and tested on the MNIST dataset for recognizing images of handwritten digits [20]. The input layer has 784 nodes corresponding to 784 pixels of images. The inputs are binary. The hidden layer has 512 neurons and the output layer has 10 neurons for recognizing 10 digits. The recognition rate is as high as 94% for 10,000 testing images. For speech recognition, an optimized Convolutional Neural Network (CNN) is deployed on the Raspberry Pi board. The performance of the ternary CNN is evaluated using the Google Speech Commands dataset [21] which contains 65,000 samples of 30 words. To deploy it on the Raspberry Pi board, a convolution neural network is designed with 4 convolutional layers, a fully-connected layer with 1024 neurons, and a softmax layer of 30 neurons for the outputs. MFCC (Mel-Frequency Cepstral Coefficient) is used for feature extraction. The coefficients of MFCCs are quantized by 8 bits. The recognition rate for speech commands recognition is 91%. This is the first test of the proposed ternary neural network for the application of image recognition and speech recognition for a mobile robot. The ternary neural network with the proposed method for representing the ternary synaptic weights is also deployed for the application of real-time object recognition. In this experiment, 10 simple objects, shown in Figure 4, are used to evaluate the performance of the proposed neural network.

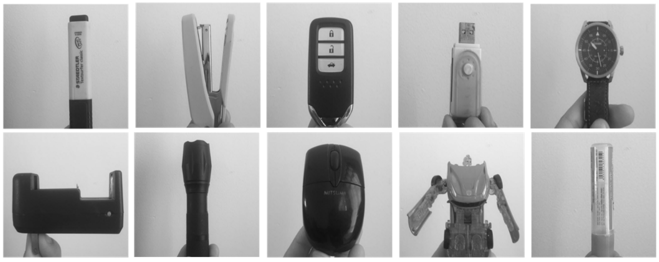

Fig. 4. Simple objects used to evaluate the performance of the proposed ternary neural network

For each object, 200 images were captured and converted to grayscale images for the training process. All images were captured with white background. Then, a ternary convolutional neural network was deployed on the Raspberry Pi board. There were 64 3×3 kernels in each convolutional layer, followed by a Max pooling layer. The fully-connected layer was composed of two hidden layers of 1024 hidden nodes. The output layer had 10 neurons for recognizing 10 objects. The convolutional neural network constraints the synaptic weights to the ternary space $\{-1, 0, 1\}$. The evaluated recognition rate for recognizing the 10 simple objects shown in Figure 4 is 89%. To compare the required memory and speed of a ternary neural network with the conventional method and the proposed technique for representing the signed synaptic weights, we deploy the two models on a Raspberry Pi board. Table I shows the comparison of the required memory of the conventional ternary neural network, and the proposed ternary neural network with a complementary binary array representing the signed synaptic weights. In Table I, we evaluate the memory that is required for storing the model and internal parameters. For a multilayer neural network, the conventional ternary neural network requires a memory of 398.2754KB, whereas the proposed ternary neural network with a complementary binary array representing the signed synaptic weights requires a memory of 99.5688KB, i.e. the proposed ternary neural network requires 75% less memory than the conventional ternary neural network. By using the bitwise AND operation and population counting instead of multiplication, the proposed ternary neural network is 4.2 times faster than the conventional ternary neural network for MNIST image recognition. For the speech recognition and real-time object recognition, the proposed ternary neural network can reduce the required memory by 75%, compared to the conventional ternary neural network. For speech recognition, the proposed ternary neural network is 2.7 times faster than the conventional neural network. For real-time object recognition, the ternary convolutional neural network with the proposed technique for ternary synaptic weight representation is 2.4×times faster than the ternary convolutional neural network using 8-bit ternary synaptic weights.

TABLE I. REQUIRED MEMORY (KB) AND INTERNAL PARAMATERS

Applications (model)	Conventional	Proposed
Image recognition (multilayer NN)	398.2754	99.5688
Speech recognition (CNN)	396.1543	99.0386
Real-time object recognition (CNN)	2,573.5	643.38

IV. CONCLUSION

Ternary neural networks have been proposed for reducing the required storage capacity and enhancing the speed of ANNs. However, the implementation of signed ternary weights still consumes high power and requires large computational resources. Many ternary neural network models have been deployed on high-performance processors such as GPUs for the application of image recognition [12-15]. In this work, the signed ternary synaptic weights are represented by two complementary binary synaptic weights. By doing this, the ternary neural networks are treated as binary neural networks. Power-hungry computational tasks such as multiplications are replaced by the bitwise AND operations to enhance the speed of the ANNs. The proposed technique is useful for deploying ANNs on low-cost embedded systems for mobile robots.

The proposed ternary neural network is deployed on a Raspberry Pi board suitable for a mobile robot. For reducing the amount of required memory, the signed values of ternary synaptic weights are represented by complementary binary arrays. Regarding image recognition, the proposed ternary neural network is tested in the MNIST dataset and achieves a recognition rate as high as 94%. For speech recognition, the proposed ternary neural network is evaluated using the Google Speech Commands and its accuracy is 91%. The proposed ternary neural network was also applied to real-time object recognition for a mobile robot. The proposed technique of representing singed synaptic weights reduces the required memory by 75% when compared to the conventional method. Overall, the proposed ternary neural network is 4.2, 2.7, and 2.4 times faster than the conventional ternary neural networks for image recognition, speech commands recognition, and real-time object recognition respectively.

ACKNOWLEDGMENT

This work is financially supported by the Ministry of Education and Training (Grant number B2019-SPK-05, 2019).

REFERENCES

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks", Advances in Neural Information Processing Systems, Lake Tahoe, USA, December 3-8, 2012

[2] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition", in: The IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, IEEE, 2016

[3] A. Graves, N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks", International Conference on Machine Learning, Beijing, China, June 21-26, 2014

[4] P. B. Patil, "Multilayered network for LPC based speech recognition", IEEE Transactions on Consumer Electronic, Vol. 44, No. 2, pp. 435-438, 1998

[5] B. M .Zahran, "Using neural networks to predict the hardness of aluminum alloys", Engineering, Technology & Applied Science Research, Vol. 5, No. 1, pp. 757-759, 2015

[6] G. S. Fesghandis, A. Pooya, M. Kazemi, Z. N. Azimi, "Comparison of multilayer perceptron and radial basis function neural networks in predicting the success of new product development", Engineering, Technology & Applied Science Research, Vol. 7, No. 1, pp. 1425-1428, 2015

[7] H. Jang, A. Park, K. Jung, "Neural network implementation using CUDA and Open MP", in: Proceedings - Digital Image Computing: Techniques and Applications, pp. 155-161, IEEE, 2008

[8] Y. Wang, J. Lin, Z. Wang, "An energy-efficient architecture for binary weights convolution neural networks", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 26, No. 2, pp. 280-293, 2017

[9] T. Simons, D. J. Lee, "A review of binarized neural networks", Electronics, Vol. 8, No. 6, pp. 1-25, 2019

[10] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio, "BinaryNet: Training deep neural networks with weights and activations constrained to +1 or −1", available at: https://arxiv.org/abs/1602.02830, 2016

[11] C. Baldassi, A. Braunstein, N. Brunel, R. Zecchina, "Efficient supervised learning in networks with binary synapses", Proceedings of the National Academy of Science of the USA, Vol. 104, No. 26, pp. 11079-11084, 2007

[12] K. Hwang, W. Sung, "Fixed-point feedforward deep neural network design using weights +1, 0, and −1", 2014 IEEE Workshop on Signal Processing Systems, Belfast, UK, October 20–22, 2014

[13] H. Yonekawa, S. Sato, H. Nakahara, "A ternary weight binary input convolutional neural network: Realization on the embedded processor", IEEE 48th International Symposium on Multiple-Valued Logic, Linz, Austria, May 16-18, 2018

[14] S. Yin, P. Ouyang, J. Yang, T. Lu, X. Li, L. Liu, S. Wei, "An energy-efficient reconfigurable processor for binary-and ternary-weight neural networks with flexible data bit width", IEEE Journal of Solid-State Circuits, Vol. 54, No. 4, pp. 1120-1136, 2018

[15] L. Deng, P. Jiao, J. Pei, Z. Wu, G. Li "GXNOR-Net: Training deep neural networks with ternary weights and activations without full-precision memory under a unified discretization framework", Neural Networks, Vol. 100, pp. 49-58, 2018

[16] L. F. Abbott, W. G. Regehr, "Synaptic computation", Nature, Vol. 431, pp. 796-803, 2004

[17] R. S. Zucker, W. G. Regehr, "Short-term synaptic plasticity", Annual Review of Physiology, Vol. 64, pp. 355–405, 2002

[18] R. Lamprecht, J. LeDoux, "Structural plasticity and memory", Nature Reviews, Neuroscience, Vol. 5, No. 1, pp. 45-54, 2004

[19] T. Mitchell, Machine learning, McGraw-Hill, 1997

[20] L. Deng, "The MNIST database of handwritten digit images for machine learning research [Best of the Web]", IEEE Signal Processing Magazine, Vol. 29, No. 6, pp. 141-142, 2012

[21] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition", available at: https://arxiv.org/abs/1804.03209, 2018

AUTHORS PROFILE

Son Ngoc Truong received his BSc and MSc Degrees in Electronic Engineering from Ho Chi Minh City University of Technology and Education, Vietnam, in 2006 and 2011 respectively, and a PhD Degree in Electronic Engineering from Kookmin University, Seoul, Korea, in 2016. He was a postdoctoral researcher fellow at Kookmin University, Seoul, Korea in 2016-2017. He is currently a lecturer at Ho Chi Minh City University of Technology and Education, Vietnam. His research interests include circuits and systems for neuromorphic computing systems, artificial intelligence, and deep learning.