Quantized Neural Networks for Low-cost Computers

Trang Dang Phuoc Hai, Hien Huynh Thi Thu and Son Ngoc Truong*

Faculty of Electrical and Electronics Engineering,
HCMC University of Technology and Education, HCM City, Vietnam

IJASR 2022
VOLUME 5
ISSUE 1 JANUARY – FEBRUARY **ISSN: 2581-7876**

Abstract: In this work, we presented the performance of quantized neural networks with compressed weight matrices for low-cost edge devices. The ternary neural network and the binary neural network were deployed on a low-cost Raspberry Pi for the application of MNIST classification. The signed weight matrix of ternary and binary neural networks was represented by the complementary binary matrices to reduce memory usage and speed up the inference process. The sparse binary matrices were compressed by Compressed Sparse Row format to reduce the memory usage and mainly reduce the inference time. For MNIST classification, ternary and binary neural networks produce the same accuracy of 94%.The ternary neural network involves weight matrices; these are sparser than the weight matrices of binary neural network. The ternary neural network and binary neural network with compressed weight matrices produced faster inference time by 9.93× and 6.78×, compared to the ternary and binary neural network without compressed weight matrices. Ternary and binary neural networks with compressed weight matrices are promising for low-cost edge devices, which have limited memory and speed.

Keywords: quantized neural network, binary neural network, ternary neural network

1. Introduction

Deep neural networks (DNN) have created a breakthrough for the application of artificial intelligence (AI) in the field of computer vision [1], [2]. This is because most of the best performance of DNNs is not just the result of larger datasets, bigger models, more improved network architectures, but mainly the development of more powerful hardware. DNNs are computationally expensive since they are composed of a large number of computational tasks, such as addition and multiplication. DNNs are effectively implemented on servers with powerful CPUs (Central Processing Units) and GPUs (Graphics Processing Units), but it seems difficult to implement on edge devices with low-cost computers [3]. Smaller DNNs are more feasible to deploy on edge devices with limited memory and speed. Quantization is one of the techniques that attempt to scale down the deep network models for low-cost edge devices. Quantized neural networks utilize a lower number of bits to represent weights, such as 16-bit floating-point, 8-bit floating-point, 8-bit integer, ternary values, and binary values [4]-[8]. Ternary and binary neural networks are promising for low-cost embedded systems. Ternary neural networks use the ternary values of -1, 0, and 1 to represent the synaptic weights [5], [6]. In binary neural networks, the synaptic weights have either -1 or 1[7], [8]. Both ternary and binary neural networks have weight values of negative (-1) and positive (+1) since the synapses are either excitatory or inhibitory [9]. Ternary and binary neural networks produce lower accuracy than full-precision neural networks; however, they consume less memory and run much faster than full-precision neural networks. The signed weight matrix can be represented by using the complementary binary arrays to save the required memory for storing weights and speed up the inference process, as demonstrated in the previous work [10]. Ternary and binary neural networks produce the same performance in the merit of accuracy and inference time. In this work, we compress the binary arrays using the Compressed Sparse Row technique to speed up the pass propagate. By doing this, it turns out an interesting result that ternary neural networks run in real-time faster than binary neural networks because the weight matrices of ternary neural networks are sparser than those of binary neural networks.

2. Method

Figure 1 shows a concept of a three-layer ternary neural network and a three-layer binary neural network.In the ternary neural network presented in Figure 1(a), the negative synaptic weights (-1) are represented by the red lines, the positive synaptic weights are represented by the green lines, and the unconnected synapses (weights of 0) are represented by the dashed lines. In the binary neural network, the synaptic weights are either negative (-1) or positive (1), as depicted in Figure 1(b).

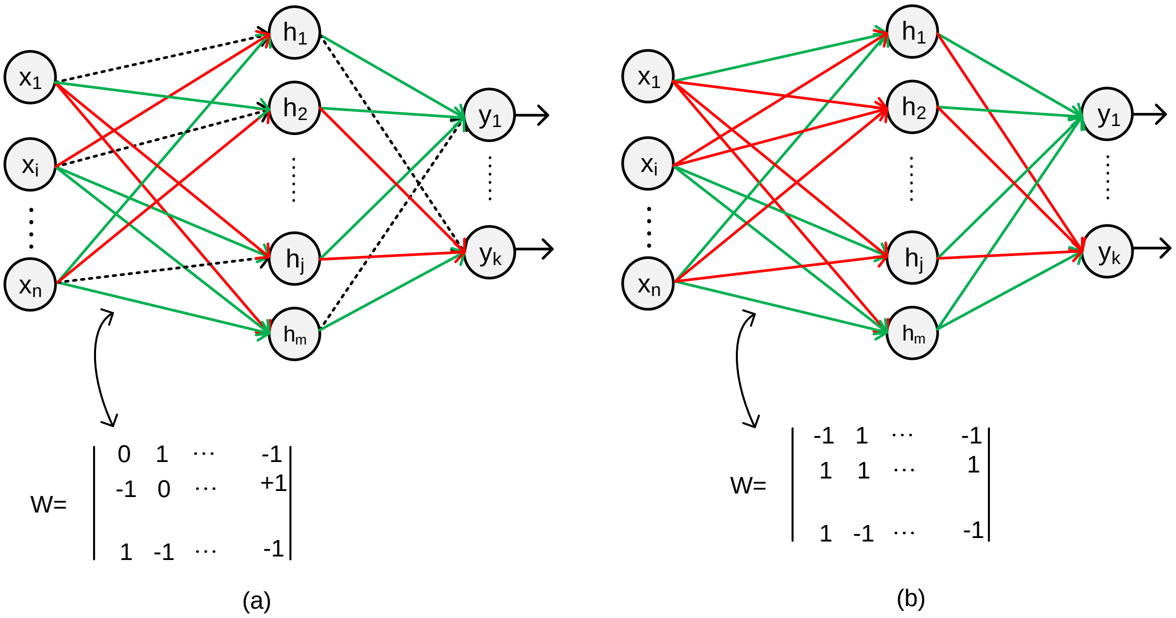


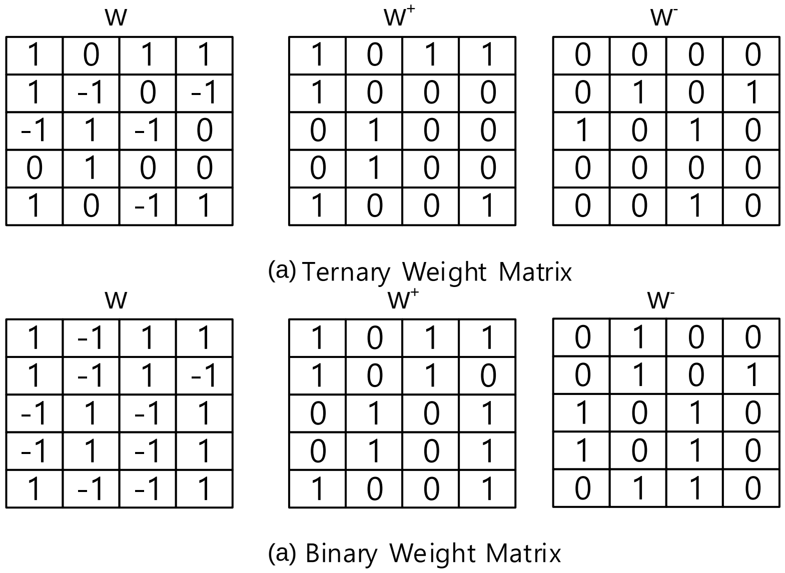
Figure 1. The conceptual diagram of (a) a ternary neural network and (b) a binary neural network

The signed weight matrix can be represented by two complementary binary matrices [10]. This technique is to replace the signed integer values with binary values of 0 and 1 to reduce the memory required to store weights and simplify the computational tasks for speeding up the neural network. Figure 2 shows a concept of the complementary binary matrices for representing the signed ternary and binary weight matrices proposed in the previous work [10]. Here W is a weight matrix that is composed of -1, 0, and 1 for the ternary weights and only -1 and 1 for the binary weights.The signed weight matrix can be composed of two complementary binary matrices of W+ and W-, as shown in Figure 2. The signed value of $W_{i,j}$ can be obtained by using the following equation [10]

$$W_{i,j} = W^{+}_{i,j} - W^{-}_{i,j} \quad (1)$$

To represent the value of -1, $W^{+}_{i,j}$ and $W^{-}_{i,j}$ are respectively 0 and 1. For the value of 1, $W^{+}_{i,j}$ and $W^{-}_{i,j}$ are respectively 1 and 0. For the value of 0, $W^{+}_{i,j}$is selected to be 0 and $W^{-}_{i,j}$ is selected to be 0.

It turns out that for the binary weights, the total number of “0” bits in W+ and W- is constant and is equal to the total number of “1” bits because the matrices are complementary. For the ternary weight matrix, the value of $W^{+}_{i,j}$ and $W^{-}_{i,j}$ that represent the value of -1 and 1 are complementary. However, the value of 0 can be represented by the value of 0 in both matrices, $W^{+}_{i,j}$ and $W^{-}_{i,j}$. As a result, the number of “0” bits in the two matrices are higher than the number of “1” bits, as illustrated in Figure 2. The matrices of W+ and W- for ternary weight matrix and binary weight matrix are sparse since they have a lot of zero entries.



$$W = \begin{array}{|c|c|c|c|}\hline 1 & 0 & 1 & 1 \\ \hline 1 & -1 & 0 & -1 \\ \hline -1 & 1 & -1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 1 & 0 & -1 & 1 \\ \hline \end{array} \quad W^+ = \begin{array}{|c|c|c|c|}\hline 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 \\ \hline \end{array} \quad W^- = \begin{array}{|c|c|c|c|}\hline 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline \end{array}$$

(a) Ternary Weight Matrix

$$W = \begin{array}{|c|c|c|c|}\hline 1 & -1 & 1 & 1 \\ \hline 1 & -1 & 1 & -1 \\ \hline -1 & 1 & -1 & 1 \\ \hline -1 & 1 & -1 & 1 \\ \hline 1 & -1 & -1 & 1 \\ \hline \end{array} \quad W^+ = \begin{array}{|c|c|c|c|}\hline 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 \\ \hline \end{array} \quad W^- = \begin{array}{|c|c|c|c|}\hline 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 1 & 0 \\ \hline \end{array}$$

(a) Binary Weight Matrix

Figure 2. Representation of signed weight matrix using complementary matrices

To reduce the number of bits representing the signed value in ternary and binary weight matrices, the complementary binary matrices are utilized, which is conceptually illustrated in Figure 2. The binary matrices for representing ternary and binary matrices are sparse. The matrices that represent ternary weight matrix is sparser than thoseones that represent binary weight matrix, leading to an interesting idea that if we can compress these sparse matrices, we can achieve more benefit interns of memory usage and computational time. The sparse matrix can be compressed using the Compressed Sparse Row (CSR) format [11]. The computation of vector-matrix multiplication and the matrix-matrix multiplication of CSR-compressed matrices is much faster than that one of uncompressed matrix [11]. An example of Compressed Sparse Row representation is illustrated in Figure 3. Figure 3(a) shows a sparse matrix and Figure 3(b) shows a CRS representation of the sparse matrix in Figure 3(a). The CSR representation of a sparse matrix involves three arrays: a row pointer array, a column indices array, and a data value array. The elements of the row pointer array indicate the pointer offset for rows. The column indices array stores the column indices of the non-zero data that is stored inthe data values array. In Figure 3(a), the value of -1 in the first cell constituted by the first row and the first column of the sparse matrixis represented by an offset of 0 in the row pointer array, the index of 0 in the column indices array, and the value -1 in the data values array. Similarly, the value 1 in the fourth row of the sparse matrix is represented by the offset of 5 in the row pointer array, the index of 1 in the column indices array, and the value of 1 in the data values array. If the matrix is large and sparse, compressing matrix using CSR archives small arrays and fast CSR matrix multiplication.

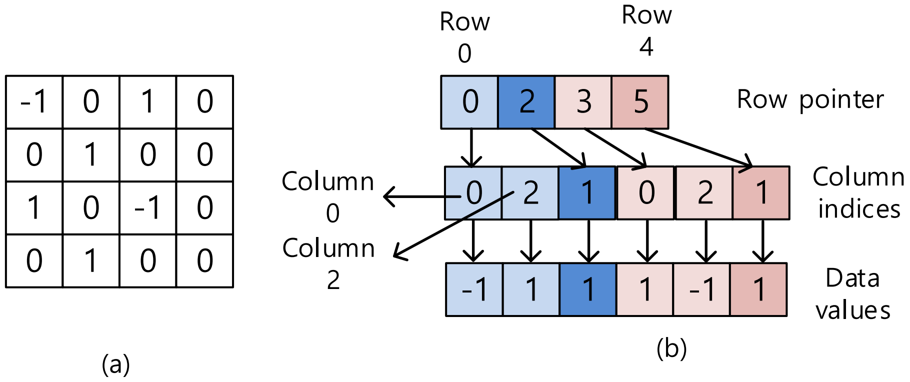


Figure 3. An example of Compressed Sparse Row (CSR). (a) Sparse matrix with high sparsity and (b) Compressed sparse Row representation of a sparse matrix

2. Result and discussion

In this work, we deployed a three-layer ternary neural network and a three-layer binary neural network for the MNIST classification on a low-cost computer, Raspberry Pi. The input layer has 784 nodes for input 28×28 images of hand-written characters. The hidden layer is composed of 512 neurons and the output layer contains 10 neurons for performing 10 ways classification.The ternary neural network and binary neural network were trained using the dynamic threshold quantization method [12]. Ternary and Binary neural networks produce the same accuracy of 94% for MNIST classification. The synaptic weight matrices in the ternary neural network and binary neural network are compressed with CSR format. We employed the Scientific Python Library for performing the vector-matrix multiplication of vector and CSR-compressed matrix and evaluated the inference time of a neural network without compressing weight matrix, a ternary neural network with compressed weight matrices, and a binary neural network with compressed weight matrices. The ternary and binary neural network without compressed weight matrices takes 90.835ms to propagate an input image from the input layer to the output layer. In other words, the inference time of ternary neural network and binary neural network without compressing weight matrices is 90.835ms. The binary neural network with compressed weight matrices consumes 15.78ms to propagate an input image to the output layer, and the ternary neural network with the compressed weight matrices takes only 9.335ms to propagate an input image to the output layer.

Ternary neural network and binary neural network produce the same accuracy for MNIST classification. Compressing weight matrices can speed up the network significantly. The binary neural network with compressed weight matrices is 6.78 times faster than the binary neural network without compressed weight matrix. The ternary neural network produces parser weight matrices. As a result, the ternary neural network with compressed weight matrices is 9.73 times faster than ternary neural without compressed weight matrices. The ternary neural network with compressed weight matrices is faster by 1.69x, compared to the binary neural network with compressed weight matrices. Ternary and binary neural networks produce the same accuracy, however, a ternary neural network with compressed weight matrices is faster than a binary neural network with compressed weight matrices.

3. Conclusion

A ternary neural network and binary neural network for MNIST classification have been presented in this work. The signed weight matrices were represented by the complementary binary matrices to reduce memory usage and speed up the network. The sparse binary matrices were compressed by Compressed Sparse Row format. The ternary neural network and binary neural network with compressed weight matrices produced faster inference time by 9.93× and 6.78×, compared to the ternary and binary neural networks without compressed weight matrices. Ternary and binary neural networks with compressed weight matrices are promising for low-cost edge devices, which has limited memory and speed.

Acknowledgements

This work belongs to the project grant No: T2021-15 funded by Ho Chi Minh City University of Technology and Education, Vietnam.

References

1. Shah U. and Harpale A. 2018, A Review of Deep Learning Models for Computer Vision, 2018 IEEE Punecon, 1-6.
2. Wu Q., Liu Y., Li Q., Jin S. and Li F., 2017, The application of deep learning in computer vision, 2017 Chinese Automation Congress (CAC), 6522-6527.
3. Jang H., Park A., Jung K., 2008, Neural network implementation using CUDA and Open MP, Proceedings - Digital Image Computing: Techniques and Applications, DICTA, 155-161.
4. Lee J, Lee J,Han D., Lee J., Park G. and Yoo H-J, 2019, An Energy-Efficient Sparse Deep-Neural-Network Learning Accelerator With Fine-Grained Mixed Precision of FP8–FP16, IEEE Solid-State Circuits Letters, vol. 2, no. 11, 232-235.
5. Hwang K.,Sung W., 2014, Fixed-point feedforward deep neural network design using weights +1, 0, and −1, Proceedings of the 2014 IEEE Workshop on Signal Processing Systems (SiPS), Belfast, UK, 1–6

6. Yonekawa H., Sato S., Nakahara H., 2018, A Ternary Weight Binary Input Convolutional Neural Network: Realization on the Embedded Processor, International Symposium on Multiple-Valued Logic, Linz, Austria.
7. Wang Y., Lin J., Wang Z., 2017, An Energy-Efficient Architecture for Binary Weights Convolution Neural Networks, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 26, 280-293
8. Baldassi C., Braunstein A., Brunel N., Zecchina R., 2007, Efficient supervised learning in networks with binary synapses, Proc. Natl. Acad. Sci. 11079–11084.
9. Abbott L. F., Regehr W. G., 2004, Synaptic computation, Nature, 431, 796-803.
10. S.Truong N. S., 2020, A Low-cost Artificial Neural Network Model for Raspberry Pi, Engineering, Technology & Applied Science Research, 10, 5466-5469
11. J. Greathouse L. and Daga M., 2014, Efficient Sparse Matrix-Vector Multiplication on GPUs Using the CSR Storage Format, SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2014, pp. 769-780.
12. Truong N. S,2020, A Dynamic Threshold Quantization Method for Ternary Neural Networks for Low-cost Mobile Robots, International Journal of Computer Science and Network Security, 20, 16-20.