Advanced Science and Technology Letters
Vol.113 (Art, Culture, Game, Graphics, Broadcasting and Digital Contents 2015), pp.50-55
http://dx.doi.org/10.14257/astl.2015.113.11

Image Classification using Fast Learning Convolutional Neural Networks

Keonhee Lee[1] and Dong-Chul Park[2]

[1]Software Device Research Center
Korea Electronics Technology Institute
Seongnam-si, Gyeonggi-do, Rep. Of Korea
khlee@keti.re.kr
[2]Dept. of Electronics Engineering,
Myoung Ji University, YongIn, Rep. of Korea
parkd@mju.ac.kr

Abstract. In this paper, we propose an image classification method for improving the learning speed of convolutional neural networks (CNN). Although CNN is widely used in multiclass image classification datasets, the learning speed remains slow for large amounts of data. Therefore, we attempted to improve the learning speed by applying an extreme learning machine (ELM). We propose a learning method combining a network of CNN with this ELM. First, we use an orthogonal bipolar vector (OBV) for studying different types of neural networks. After learning a limited epoch used in the CNN, we determine the output of the neural network using the ELM. To evaluate the performance, we compared the learning speed and classification rate of conventional CNN against the proposed method. Experimental results show that the proposed algorithm has a faster learning curve than the existing algorithm.

Keywords: Image Classification, Orthogonal Bipolar Vector, Convolutional Neural Network

1 Introduction

Image classification is still attracting a considerable amount of interest in the fields of pattern recognition and computer vision. Currently, researchers are focusing on deep learning in many industrial areas. In particular, the concept of deep learning has been applied to the field of recognition and classification. The convolutional neural networks (CNN) used in deep learning have been applied to handwritten character recognition, pedestrian detection, and traffic sign recognition [1], [2]. CNN consist of pairs of convolution and subsampling layers. Further, the last layer of the CNN structure consists of multilayer perceptron neural networks (MLPNNs) [3]. Because of this structure, CNN is an integrated model used for feature extraction and classification. In recent years, many researchers have been studying the applications of parallel computing for increasing the training speed [4].

To address the need for fast learning, an extreme learning machine (ELM) was developed. The generalization performance of the ELM is better than that of

ISSN: 2287-1233 ASTL

conventional methods. However, the ELM has a disadvantage when big data and a large network are used because the hidden layer output matrix and the system memory required for calculating the inverse matrix are proportional. To overcome this disadvantage, an online sequential learning method (OS-ELM) was proposed [5].

In this paper, we propose a learning method combining a network of CNN with an ELM for fast-learning neural networks. To combine different algorithms, we used the orthogonal bipolar vector (OBV) [6] method for training the CNN without a data target. After training the CNN using an OBV, the ELM trains their last classification layer.

The rest of this paper is organized as follows: In Section 2, we provide a brief review of the related CNN and ELM algorithms. Section 3 introduces the OBV target method for training the hidden layer weight of the CNN. The results of the performance evaluation of the proposed networks are described in Section 4. Finally, some concluding remarks regarding this study are presented in Section 5.

2 Overview of Learning Algorithm

2.1 Convolutional Neural Networks

We can divide the structure of a CNN into a feature extraction part and a classification part. The feature extraction part contains convolution layers alternating with the subsampling layers. A convolution layer has many randomly generated two-dimensional shared weights [-0.5, 0.5]. This layer convolution operation using the shared weight of the input data can be expressed as follows:

$$h_j = \tanh(\sum x_i * k_{ji} + bias) \tag{1}$$

where x denotes the data on the previous layer, and k represents the weight matrix. After the convolution operation, input data are reduced according to the size of the shared weight matrix. Further, the data are calculated using an activation function called a hyperbolic tangent. Therefore, every output of the convolution layer is normalized from -1 to 1.

The subsampling layer has two operations, average pooling and max-pooling. Many researchers use the max-pooling method because it performs better than average pooling for certain datasets.

$$p_j = \tanh(\max_{k=1}^{r}(h_{jk}^{n\times m}) + bias) \tag{2}$$

where p_j denotes the output data of the subsampling layer. The max value is extracted from h, which is an image of the previous layers. The convolution- and subsampling-layer operations are repeated until the input data size is 1×1. The final data in the feature extraction part are the inputs of the classification layer. The

classification layer consists of multilayer perceptron neural networks. This layer is trained using gradient descent-based learning.

2.2 Extreme Learning Machine

Traditional training algorithms require complex calculations and a lengthy learning time. Because a hidden node parameter is randomly generated according to a continuous distribution probability, the ELM was proposed for use in SLFNs, and the hidden layer parameter was not trained. Therefore, the ELM is calculated as a one-time inverse operation of the output layer parameter. The calculation of the hidden layer output matrix of the ELM can be expressed as follows:

$$\boldsymbol{H} = sig(\boldsymbol{X} * \boldsymbol{W} + \boldsymbol{bias}) \tag{3}$$

where **X** denotes the input data matrix and **W** represents the weight between the input layer and hidden layer. The *sig*() function refers to the sigmoid function. Therefore, our input data must be generalized from zero to 1. Further, a matrix H used to train the ELM can be expressed as follows:

$$\begin{aligned} \boldsymbol{H\beta} &= \boldsymbol{T} \\ \boldsymbol{H}^{\dagger}\boldsymbol{T} &= \boldsymbol{\beta} \end{aligned} \tag{4}$$

where $\mathbf{H}^{\dagger}$ represents the *Moore-Penrose generalized inverse* of matrix **H**. Some methods can be used for calculating an inverse matrix **H**. Further, orthogonalization, orthogonal projection, singular value decomposition (SVD), and Gauss elimination are used for calculating $\mathbf{H}^{\dagger}$.

For a huge dataset, a large amount of system memory is required to calculate the inverse matrix of H. Therefore, we apply OS-LEM as follows:

Initial phase -

$$\begin{aligned} \boldsymbol{P}_0 &= (\boldsymbol{H}_0^T\boldsymbol{H}_0)^{-1} \\ \boldsymbol{\beta}^{(0)} &= \boldsymbol{P}_0\boldsymbol{H}_0^T\boldsymbol{T}_0 \end{aligned} \tag{6}$$

Sequential phase -

$$\begin{aligned} \boldsymbol{P}_{(k+1)} &= \boldsymbol{P}_0 - \boldsymbol{P}_k\boldsymbol{H}_{(k+1)}^T \times \\ &\qquad \left(\boldsymbol{I} + \boldsymbol{H}_{(k+1)}\boldsymbol{P}_k\boldsymbol{H}_{(k+1)}^T\right)^{-1}\boldsymbol{H}_{(k+1)}\boldsymbol{P}_k \\ \boldsymbol{\beta}^{(k+1)} &= \boldsymbol{\beta}^{(k)} + \\ &\qquad \boldsymbol{P}_{(k+1)}\boldsymbol{H}_{(k+1)}^T(\boldsymbol{T}_{(k+1)} - \boldsymbol{H}_{(k+1)}\boldsymbol{\beta}^{(k)}) \end{aligned} \tag{7}$$

3 Orthogonal Bipolar Vector target

An OBV-type target is characteristic in that the distance between it and the other target vector is constant. The component of an OBV target n is calculated using $n = 2^k m$, where m is the number of components of the seed vector.
Steps:

1) Initialize parameters m and k
2) Generate the seed vector $V_m(1) = (1, 1, 1, 1,...,1)$
3) Construct the first vector: $V_{2m}(1) = [V_m(1), V_m(1)]$ and $V_{2m}(1) = [V_m(1), -V_m(1)]$
4) Construct the next Vector: $V_{4m}(1) = [V_{2m}(1), V_{2m}(1)]$, $V_{4m}(2) = [V_{2m}(1), -V_{2m}(1)]$, $V_{4m}(3) = [V_{2m}(2), V_{2m}(1)]$ and $V_{4m}(4) = V_{2m}(2), -V_{2m}(2)]$
5) Repeat steps 4 k times to generate a vector with n components.

4 Implementation

4.1 Image dataset

The NORB dataset consists of five differently shaped toys. Each of the main categories has a different shape for each of ten data items, where all data are light in weight, and their angles and azimuth are slightly different.

The MNIST handwritten digit dataset has a training set of 60,000 items and a test dataset of 10,000 examples. This dataset consists of images of numbers ranging from zero to 9, and the size of each gray-scale image is 28 pixels $\times$ 28 pixels.

The CIFAR-10 dataset consists of 60,000 32×32 color images divided into ten classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images. The dataset is divided into five training batches and one test batch, each with 10,000 images.

Table 1. Performance of the CNN and the proposed method

	CNN		CNN + ELM	
	Training time	Recognition rate	Training time	Recognition rate
NORB	29234.s	87.54%	4537.4s	89.65%
MNIST	10475.1s	92.14%	3078.5s	93.54%
CIFAR-10	8020.5s	63.54%	1244.8s	64.36%

4.2 Experiment

In this study, we used an Intel Core i5-6600k CPU (3.50 GHz), with 16.00 GB of DDR4 RAM. Further, we used the Matlab 2014b version of the engine library written

in C language for testing the CNN. Note that the data chunk size was never less than the number of hidden nodes.

We compared the proposed algorithm with the existing algorithms experimentally using the NORB, CIFAR-10, and MNIST datasets. The results of the performance evaluation are shown in Table 1. These results were obtained by measuring the recognition rate using the learning time and test data. The results of a recognition test of the proposed algorithm using the NORB dataset showed that the learning speed was improved by 6.4 times and that the recognition rate was improved by 2.11%. Similarly, during experiments conducted using the MNIST dataset, the learning rate was improved by 3.4 times, and the recognition rate was improved by 1.4%. CIFAR was also classified by the proposed algorithm, which in this case showed a slight improvement over the conventional algorithms.

5 Conclusion

In this paper, we proposed a combination of CNN with an ELM for image classification. A learning method called an orthogonal bipolar vector (OBV) was also well applied by analyzing the neural networks learned by combining both algorithms. The experimental results demonstrate that the proposed method can conduct network learning at a faster rate than conventional CNN. In addition, it can be used to solve the local minima and overfitting problems. When applied to an image classification problem, the proposed method shows promising results in terms of the training speed. As a result of combining both algorithms mentioned above, the proposed method can reduce the learning time by 3 to 6 times compared to conventional CNN, and improves the recognition rate.

Acknowledgments. This work was supported by the IT R&D program of MKE/KEIT (1004019, the development of automotive synchronous Ethernet combined IVN/OVN and safety control system for 1 Gbps class).

References

1. Szarvas, M., Yoshizawa, A., Yamamoto, M., Ogata, J.,: Pedestrian detection with convolutional neural networks. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 886–893, IEEE, San Diego (2005)
2. Jin, J., Fu, K., Zhang, C.,: Traffic Sign Recognition With Hinge Loss Trained Convolutional Neural Networks. IEEE Tr. Intelligent Transportation System. 15(5). 1991–2000 (2014)
3. Huang, F. J., LeCun, Y.,: Large-scale Learning with SVM and Convolutional Nets for Generic Object Categorization. In: IEEE Computer Society conference on Computer Vision and Pattern Recognition, pp. 284–291. IEEE, New York (2006)
4. Scherer, D., Schulz, H., Behnke, S.,: Accelerating Large-Scale Convolutional Neural Networks with Parallel Graphics Multiprocessors In: Diamantaras, Duch, Iliadis. (eds.) ICANN 2010. LNCS, vol. 6354, pp. 82–91. Springer, Thessaloniki (2010)

5. Liang, N. Y., Huang, G. B., Saratchandran, P., Sundararajan, N.,: A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. IEEE Tr. Neural Networks, 17(6), 1411–1423 (2006)
6. Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., Shiose, T.,: Improved MLP Learning via Orthogonal Bipolar Target Vectors. JACIII 9(6), 580–589 (2005)

