A Convolutional Neural Network with Incremental Learning

Haruka Tomimori¹, Kui-Ting Chen² and Takaaki Baba³

¹,³Graduate School of Information, Production and Systems, Waseda University
²Research Center of Information, Production and Systems, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka 808-0135, Japan
E-mail: ¹tomiharu@fuji.waseda.jp, ²nore@aoni.waseda.jp, ³tbaba@waseda.jp

Abstract

Nowadays, a convolutional neural network (CNN) is considered as a deep learning method for image and voice recognition. A CNN can achieve higher recognition accuracy than other approaches since it can automatically extract features by its learning procedure. However, the training procedure of a CNN is time-consuming. Since the functions of a CNN are close to those of a human brain, when a CNN is applied to a complex application, it must be trained by a large amount of training data, resulting in the size of the CNN becoming huge. To train such a huge neural network by computers, a tremendous amount of training time is required. In this paper, an efficient approach is proposed that can markedly reduce the training time while only slightly sacrificing the recognition accuracy of the training procedure.

1. Introduction

A convolutional neural network (CNN) is considered as a deep learning method with similar accuracy to a human brain. Recently, many researchers have focused on the working principle of CNNs for complex image recognition, since a CNN can extract essential features by its learning procedure.

The concept of the CNN was introduced by Fukushima in 1980, which was named the neocognitron [1]. Fukushima proved that the neocognitron has the ability to recognize stimulus patterns without being affected by a shift in position or by a small distortion in the shape of the stimulus patterns. In 1982, Fukushima and Miyake successfully overcame the drawbacks of the neocognitron. The improved neocognitron could recognize not only stimulus patterns, but also stimulus patterns of Arabic numbers with position shifting and shape distortion [2]. Fukushima completed a set of theories for the neocognitron in 1988 [3]. However, the neocognitron could only be applied to image recognition with simple patterns. LeCun et al. [4] improved the neocognitron and proposed a prototype CNN which was named LeNet-5. Their results proved that, LeNet-5 could achieve higher recognition accuracy than other approaches for handwritten Arabic numbers. LeNet-5 not only could be applied to the recognition of handwritten Arabic numbers but also had the possibility of application to the recognition of more complicated images. The basic theories of the CNN were summarized by Behnke in 2003 [5]. Before a CNN is applied to a complex application, it must be trained by a large amount of training data since the functions of a CNN are close to those of a human brain. In addition, the size of a CNN becomes huge. To simulate such a huge neural network by computers, high performance computers are required.

For example, if a CNN is applied to face recognition for social networks such as Facebook and Twitter, the CNN may require tens of millions of images for training data. In such a situation, the amount of neurons will become several billion neurons. The calculation time in the training procedure may be a few months even if thousands of computers are used at the same time. Therefore, a drawback of CNNs is the extremely time-consuming training procedure. Moreover, a CNN needs to learn all the training data again when it accepts additional data, otherwise it loses its previous knowledge.

Therefore, the purpose of this study is to reduce the training time even if the recognition accuracy slightly decreases as a trade-off.

The rest of this paper is organized as follows. In Sect. 2, the concept of the CNN and previous studies on artificial neural networks with incremental learning are briefly introduced. In Sect. 3, the details of the proposed CNN architecture are presented. In Sect. 4, the simulation results of the proposed approach are provided. Finally, Sect. 5 gives a summary and the conclusions of this study.
2. Related Works

2.1 Convolutional neural network (CNN)

As shown in Fig. 1, the CNN in this study consists of convolution layers (C1 and C2), pooling layers (P1 and P2) and a fully connected neural network (F1 and F2). An outstanding feature of the CNN is that the convolution layers and pooling layers are used to obtain the pattern features during the training procedure.

The convolution layer C1 is used to detect the features in images such as edges, which are used to obtain the relationship between pixels of the input images. The convolution utilizes an $N \times N$ feature matrix $W$ (or kernel) to scan the input image $I$ to execute the sum of products. The definition of the convolution is given by

$$C_{ij} = f \left( \sum_{p=1}^{N} \sum_{q=1}^{N} (I_{i+p-N/2, j+q-N/2} \cdot W_{p,q}) + b \right)$$  \hspace{1cm} (1)

One neuron can execute one calculation of the convolution using a feature matrix. The elements of the feature matrix correspond to the weights of the neurons. By adopting this approach, the elements of the feature matrix can be adjusted through the training procedure. This procedure can be automatically regulated by the CNN to meet the various requirements of different purposes. In addition, the performance of the feature extraction of the image can be much more accurate than manually setting, like Canny method.

As shown in Fig. 2, if the sizes of the input image and feature matrix are respectively set as 8 by 8 and 3 by 3, a 6 by 6 convolved image, which is defined as a feature map, is produced. Therefore, the feature map contains the object feature information in the input image via the feature matrix. In order to extract different object features from the input image, a convolution layer is usually composed of several feature maps.

The pooling layer P2 is used to reduce the size of the feature maps, which can reduce the calculation time of the CNN. Furthermore, some indistinctive features of the feature maps can be eliminated as a noise component.

One of the most effective pooling methods is named max-pooling, which leaves the largest value in each small area of the feature map. An output feature map $P$ of the max-pooling layer is given by the equation below

$$P_{ij} = \max \left( C_{p+N(i-1), q+N(j-1)} \right)$$  \hspace{1cm} (2)

where $N$ is the size of the small area considered in the max-pooling function.

As shown in Fig. 3, if the sizes of the input feature map and the small area are respectively set as 8 by 8 and 2 by 2, a 4 by 4 feature map is produced.
2.2 NN with incremental learning

Tanaka et al. [6] and Polikar et al. [7] proposed an ANN with incremental learning to recognize irregular images, and the trained ANN can be remained with additional training data only. As shown in Fig. 5, this approach adds several neurons to both the hidden layer and the output layer in order to recognize irregular images. This approach can prevent the trained ANN from requiring retraining when a new training data set is required. As a result, they achieved higher accuracy than conventional approaches. Their simulation result proved that incremental learning can be applied to improve accuracy without retraining the whole ANN. Their training strategy can be used to reduce the execution time of the CNN training procedure because there is a small amount of retraining data and few retrained parameters in their retraining procedure.

3. Proposed CNN

As shown in Fig. 6, the training procedure of the CNN is divided into two stages, which are initial learning and incremental learning. In the initial learning stage, all the parameters of the ANN are regulated by a training procedure, as shown in Figure 6. Once the initial learning is finished, the parameters of the feature detection part (C1, P1, C2 and P2) are fixed to a constant value that is recognized as acquired knowledge. In the incremental learning stage, only a few parameters of the recognition part (F1 and F2) are regulated. Therefore, this approach can markedly reduce the execution time of the training procedure.

The proposed architecture consists of seven layers, which are the input layer, convolution layer C1, pooling layer P1, convolution layer C2, pooling layer P2, hidden layer F1 and output layer F2, as shown in Figure 4. Layer C1 adopts six convolution kernels, each with a size of 5 by 5, which can produce six feature maps. Layer C2 adopts 6×12 convolution kernels, which can produce 12 feature maps. The pooling layer adopts the max-pooling function whose matrix size is 2 by 2. Layer F2 adopts the softmax function as the activation function, which can normalize the total of the output values to 1. Therefore, each output value is a recognition probability, which indicates whether the output result is correct or not. For the proposed CNN, the training time is defined as the sum of the initial learning time and the incremental learning time. The proposed output is defined as the output with the highest probability in both the initial learning and the incremental learning, which means that the proposed CNN acquires two pieces of knowledge for image recognition and can achieve higher recognition accuracy than that in the initial learning and incremental learning.
4. Simulation Results

To investigate and compare the performances of a conventional CNN and the proposed CNN, the handwritten number recognition system for MNIST (Mixed National Institute of Standard and Technology) data is employed. The total number of MNIST images is 60,000, of which 50,000 images are used for the training procedure and the others are used to test the system and calculate the recognition accuracy. The size of each MNIST image is 28 by 28. In the proposed CNN, 25,000 images in the training data are used for training in the initial learning and the others are used for training in the incremental learning. Both the conventional CNN and the proposed CNN are implemented by Python 2.7 on a PC with an Intel Core i7 3.6 GHz processor and 32 GB RAM.

Table 1 shows the performances of the proposed CNN. The simulation results demonstrate that the incremental learning can reduce the calculation time of the training procedure to 5.94 min, which is half the calculation time of the initial learning. In addition, they demonstrate that the proposed approach can achieve recognition accuracy of up to 98.10%, which is better than those of the initial learning and incremental learning.

Table 2 shows a comparison between the proposed and conventional methods for the calculation time and recognition accuracy. The proposed CNN can reduce the calculation time of the training procedure from 28.61 min to 17.70 min. As a trade-off, the recognition accuracy slightly decreases from 98.72% to 98.1%. This decrease in the calculation time of 38.1% is an acceptable trade-off with the increase of 0.63% for the recognition error.

5. Conclusions

In this paper, the architecture of a CNN with incremental learning and its simulation were presented. The training procedure of the CNN is divided into two stages, which are initial learning and incremental learning. Compared with the conventional CNN approach, the proposed approach can markedly reduce the training time by 38.1% while sacrificing 0.63% recognition accuracy.

Acknowledgment

This work was supported by Japan Society for the Promotion of Science (JSPS) through a Grant-in-Aid for Scientific Research (No. 26280017) and a Grant-in-Aid for Young Scientists (No. 15K21435).

Table 1: Performances of the proposed CNN

|                      | Initial learning | Incremental learning | Proposed |
|----------------------|------------------|----------------------|----------|
| Time (min)           | 11.76            | 5.94                 | 17.70    |
| Accuracy (%)         | 97.74            | 97.85                | 98.10    |

Table 2: Comparison between calculation time and accuracy

|                      | Conventional                  | Proposed          | Improvement   |
|----------------------|-------------------------------|-------------------|---------------|
| Time (min)           | 28.61                         | 17.70             | 38.1%         |
| Accuracy (%)         | 98.72                         | 98.10             | -0.63%        |

References

[1] K. Fukushima: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biological Cybernetics, Vol. 36, No. 4, pp. 193-202, 1980.
[2] K. Fukushima and S. Miyake: Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position, Pattern Recognition, Vol. 15, No. 6, pp. 455-469, 1982.
[3] K. Fukushima: Neocognitron, A hierarchical neural network capable of visual pattern recognition, Neural Networks, Vol. 1, No. 2, pp. 119-130, 1988.
[4] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-based learning applied to document recognition, Proc. IEEE, Vol. 86, No. 11, pp. 2278-2324, 1998.
[5] V. I. Arnold: Hierarchical neural networks for image interpretation, Lecture Notes in Computer Science, Vol. 2766, Springer, 2003.
[6] N. Tanaka, H. Abe, and K. Kajitani: A method of additional learning for BP network by adding extra neural network, IEICE Trans. Systems, J82-D-2, Vol. 4, pp. 669-676, 1999.
[7] R. Polikar, J. Byorick, S. Krause, A. Marino and M. Moreton: Learn++: A classifier independent incremental learning algorithm for supervised neural networks, Proc. Int. Joint Conf. on Neural Networks, Vol. 2, pp. 1742-1747, 2002.