The training of convolution neural network for advanced driver assistant system

Kihun Nam¹, Heekyeong Jeon¹*

Department of computer engineering, Seokyeong University
{namkh, iq8580}@skuniv.ac.kr

Abstract

In this paper, the learning technique for CNN processor on vehicle is proposed. In the case of conventional CNN processors, weighted values learned through training are stored for use, but when there is distortion in the image due to the weather conditions, the accuracy is decreased. Therefore, the method of enhancing the input image for classification is general, but it has the weakness of increasing the processor size. To solve this problem, the CNN performance was improved in this paper through the learning method of the distorted image. As a result, the proposed method showed improvement of approximately 38% better accuracy than the conventional method.

Keywords: Convolution neural network, Deep Learning, Image enhancement, Learning method

1. Introduction

Recently, as the image recognition field reached the limit in the development of the conventional recognition algorithm of hand-writing based recognition algorithm, the interests on the artificial neural network are increasing rapidly. Especially, the image-based ADAS (Advanced Driver Assistant System) had the problem of difficulty in responding to the conventional hand-writing image recognition technology due to various environmental variables. Therefore, the neural network for recognition was developed, and the neural network that is based on the learning data to respond to various variables was raised as the new solution for ADAS. For example, NVIDIA developed the NVIDIA DRIVE™ PX to enable diverse classification of people, signs and vehicles, etc. than in the conventional system [1]. In addition, the MPPA® many core processor developed by KALRAY is an image recognition processor like ADAS, and it is mounted with the convolution neural network [2].

As the purpose of ADAS is to assist the driver, it must maintain consistent performance even in various weather changes. However, the conventional neural network processors require great memory calculation & size in the learning process of neural network, so instead of the real-time learning function, the weighted values that were learned in advance are stored in the memory for use.
Due to this reason, there is a weakness that the performance is proportional to the state of the inputted image. So in the conventional neural network, there is a preprocessor for solving this problem that increases the state of the inputted image before classifying them into the neural network [3]. The neural network is a technology that has high consumption amount of hardware resources, so in the field such as ADAS that has limited hardware resources, additional module becomes a high risk compared to other technologies.

In this paper, the distorted image learning was performed to solve this problem, because it enables high accuracy rate even in rainfall without the use of the preprocessor. In addition, in the image that is not distorted, the same accuracy rate is shown as in the conventional neural network to have better environmental responsiveness than the conventional neural network that learned the revised image. So in this paper, to prove this, the comparative test was conducted on the conventional CNN that learned preprocessed images and the proposed CNN that learned distorted images.

2. Related researches

2.1 Convolutional neural network

CNN (Convolution neural network) is a type of neural network that classifies the image characteristics to classify the images with big data [4][5]. Generally, CNN is composed of the input layer receiving the image, the convolution layer and the pooling layer that detect specific values, and the fully connected layer that classifies detected characteristic points.

The convolution layer is a layer that enables convolution on the image inputted through the input layer by various filters. The filters used in the convolution layer are initially designated in random, and as these are learned, it is improved according to the classification purpose of data. The output result of the convolution layer per one filter is called the feature map, and this increases according to the number of filters. Here, to enable filter learning, the convolution result is inputted as the activation function, and the output value of the activation function is stored in the feature map. Generally, as shown in Figure 1, sigmoid function, tanh function and the ReLU (Rectified Linear Unit) are mostly used for the types of activation function [6][7]. However, the tanh function and the sigmoid function are occurred with error vanishing problem due to the limitation in the expression range, therefore, the ReLU function with the wide expression range is used recently. The pooling layer is a process of integrating the feature map of the convolution layer output, and the integration process of its characteristics is differed according to the pooling method. Main pooling processes are max pooling and average pooling.

![Figure 1 The activation functions](image)

(a) $f(x) = \frac{1}{1+e^{-x}}$

(b) $f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

(c) $f(x) = \max(x, \epsilon) = \begin{cases} 1 & x > \epsilon \\ x & x \leq \epsilon \end{cases}$

Figure. 1 The activation functions (a) sigmoid (b) tanh (c) ReLU
The training of convolution neural network for advanced driver assistant system

The max pooling is a method that selects the largest value within the patch field, and the average pooling selects the average value of the patch field as the integrated value. In the main convolution neural network of AlexNet, the max pooling method was used as shown in Figure 2, and in the GoogleNet that was developed later, the average pooling method was used. The pooling layer is the layer that is not enabled with learning [8].

The fully connected layer is the most basic neural network structure, and all nodes are connected as weighted values. In addition, the fully connected layer is a layer that is possible for learning, and the final output passes through the activation function. As the hidden layer, the fully connected layer uses the ReLU function like the convolution layer. However, the last fully connected layer uses softmax for class classification [9]. As shown in Formula (1), the Softmax function outputs each node in the probability of inputted image to be relevant to each class.

\[
\sigma(z)_j = \frac{e^z_j}{\sum_{k=1}^{K} e^{z_k}} \text{ for } j = 1, \ldots, K
\]

In a common convolution neural network, there are 2 stages of the detecting element that connects the convolution layer and the pooling layer, and the fully connected layer is composed of 2~3 layers. Therefore, each layer requires great memory calculation, and also many memory resources. In addition, when the sigmoid or tanh function is used, floating calculation is required, so many resources are consumed in configuration into a hardware.

### 2.2 Conventional training method

Learning in the convolution neural network uses Backpropagation like in the general neural network learning [10][11]. Backpropagation is a type of supervised learning method that propagates the neural network result and the target value error to the neuron of each layer and then improves the weighted value in proportion. The representative method of this Backpropagation is the gradient descent algorithm. The gradient descent algorithm propagates the error, but it propagates its gradient to improve the weighted value and to gradually minimize the error. The process of propagating the error is shown in Formula (2). First, the weighted value to improve in the total error is differentiated, and the differentiation value of multiplying the learning rate to the existing weighted value is deleted. Generally, 0.1 is set for the learning rate.
However, to calculate each error in the large-scale image dataset for learning, great amount of learning time will be required. In addition, it will require great amount of memory space, and the memory calculation will be performed to give load to the hardware.

Therefore, two gradient descent algorithms that improved this problem are used recently.

The first method is the stochastic gradient descent algorithm [12]. The stochastic gradient descent algorithm combines the costs on all learning data, and mixes the learning data randomly to perform repetitive partial differentiation on the relevant cost. Therefore, the errors on the total dataset are not updated individually and calculated in order to be updated, which reduces the learning time and the memory calculation.

Secondly, the mini-batch gradient descent algorithm is used [13].

This is same and the intermediate processing of the stochastic gradient descent. The stochastic gradient descent algorithm processes on all learning data, but the mini-batch gradient descent algorithm calculates m-number of data for certain number of batches. m-number is calculated for one calculation, so the speed is faster than the conventional gradient descent algorithm, and sometimes, it is faster than the stochastic gradient descent algorithm. Figure 3 shows the stochastic gradient descent algorithm and the mini-batch gradient descent algorithm in pseudo code.

![Figure 3 The pseudo code of stochastic gradient descent algorithm and mini-batch gradient descent algorithm](image)

However, all two algorithms improved the speed, but require great amount of memory calculation, so the convolution neural network processor performs this process in advance through the CPU and stores the learned weighted values in the memory for use. Therefore, it has the weakness of difficulty in responding to the environmental variables that change in real-time.
3. Proposed training method for CNN processor

When the hardware is designed to which the weighted values in the convolution layer and the fully connected layer are learned and the accuracy rate of CNN is increased, each weighted value stores the weighted value learned in advance in the memory for use. However, this case is reduced in accuracy when the image is distorted.

The proposed convolution neural network uses the blurred image as the learning data.

Figure 4 shows the result value of the one of learning data of original pedestrian image and the distorted image passing through the convolution layer by using one filter. As we can see in Figure 4, the distorted image does not lose the characteristics of the existing image too much. The image characteristic of the convolution neural network is that the filter used in the convolution calculation enables learning as the filter appropriate for classifying the input image. Therefore, when the filter for learning the blur image and the filter for learning the original image were compared, the constant filter was verified to be all equal filters, and the filter modification that was occurred from partial image distortion was verified to have no great modification as the distortion strength was intensified. Table 1 shows the filter change rate according to the learning data for each sigma.

![Figure 4 The convolution result of original pedestrian image and blur pedestrian image](image)

| Sigma 0 to Sigma 1 | Sigma 1 to Sigma 2 | Sigma 2 to Sigma 3 |
|--------------------|--------------------|--------------------|
| Distortion rate    | 11.1529            | 8.756382           | 4.8846             |

4. Experiments

The test was performed with the pedestrian classification test in the Matlab. Regarding the dataset used for the learning and the test, the Daimler Pedestrian Benchmark Data Set was used, and the size was set to 16x32 [14]. The number of images for learning on each epoch was 6,000, and 2,000 images for testing were used for the accuracy test on each epoch. The Epoch was performed 100 times for gathering the accuracy. In the test, the comparative test was performed on the conventional CNN learning undistorted images and the proposed CNN learning distorted images, and Gaussian blur was used for the distortion.
The distortion intensity was increased for each sigma, and for the conventional CNN, the Weiner filter was used on the distorted image in the test to input the image that was restored from the distorted image. The test result is shown in Table 2 & 3. As shown in Table 2, the difference in the change rate of the filter decreased rapidly according to the level of distortion in the image. Table 2 shows the result of the filter reduction value according to each sigma intensity. Also, Table 3 shows the comparative results of the CNN performed with learning in the proposed method with the conventional CNN. As shown in Table 3, the image that was not distorted from the original image was verified to decrease in performance, but the image maintained similar accuracy in proportion to the distorted intensity.

Table 2. The distortion between filter per sigma

| Heading level | Example | Font size and style |
|---------------|---------|---------------------|
| 0.061 | 0.015 | 0.024 | 0.024 | 0.035 | 0.025 | 0.039 | 0 | 0.007 | 0 | 0.074 | 0.074 |
| 0.066 | 0.017 | 0.025 | 0.012 | 0.016 | 0.005 | 0.028 | 0 | 0.004 | 0 | 0.001 | 0.001 |
| 0.083 | 0.046 | 0.013 | 0.014 | 0.013 | 0.025 | 0.003 | 0.005 | 0.003 | 0.009 | 0.006 | 0.006 |
| 0.066 | 0.051 | 0.031 | 0.020 | 0.014 | 0.027 | 0.011 | 0 | 0.002 | 0.003 | 0.002 | 0.002 |

Table 3. The error per sigma (%)

| Sigma 0 | Sigma 1 | Sigma 2 | Sigma 3 |
|---------|---------|---------|---------|
| Conventional CNN | 7.0965 | 11.4443 | 14.9425 | 20.2399 |
| Proposed CNN | 9.7451 | 8.4958 | 9.145 | 9.945 |

5. Conclusion

In this paper, the CNN that enables learning on the distorted image was proposed. The convolution neural network processor for ADAS has the characteristic that the weighted value learned in advance is used due to the limitation in hardware resources. However, this is no appropriate for the purpose of ADAS to maintain the consistent accuracy level according to the environmental variables that change rapidly in real-time. To solve this problem, this paper proposes the learning technique that can respond to the distorted image even when it is a weighted value learned in advance. Unlike the conventional CNN, the proposed CNN learns the distorted image, but not the general image when learning the weighted value for the CNN processor using fixed weighted values. This can maintain the accuracy level without the preprocessor that additionally restores the image quality. In addition, it is appropriate for reducing the complexity of the processor, and in the environment with limited resources such as the embedded environment. Moreover, the most critical problem in the CNN processor of the size can be improved, so it is expected to contribute to the application fields that require small-size processors such as in vehicles.
References

[1] http://www.nvidia.com/object/advanced-driver-assistance-systems.html

[2] https://indico.cern.ch/event/395242/attachments/791742/1085286/DEEP_LEARNING_ON_MPPA_MANYCORE_PROCESSOR_V2.2.pdf

[3] Clément Farabet, Berin Martini, Polina Akselrod, Selçuk Talay and Yann LeCun, “Hardware accel-erated convolutional neural networks for synthetic vision systems,” Proceedings of 2010 IEEE International Symposium on Circuits and Systems, (2010) May 30 – June 2; Paris, France

[4] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton, “ImageNet Classification with Deep Con-volutional Neural Networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence. (2016), Vol.38, Issue.10, pp.1943-1955

[5] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama and Trevor Darrell, “Caffe: Convolutional Architecture for Fast Feature Embedding,” Proceedings of the 22nd ACM international conference on Multimedia, (2014), November 3-7; Florida, USA

[6] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. “Deep Sparse Rectifier Neural Net-works,” Aistats, (2011), Vol. 15, No. 106.

[7] Hee-kyeong Jeon, Kwang-yeob Lee, Chi-yong Kime Yong-beom Cho, “An implementation of a convolutional accelerator based on a GPGPU for a Deep Learning,” Journal of IEEE, (2016), Vol.20, No.3, pp.303-306

[8] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going deeper with convolutions”

[9] Gold, Steven, and Anand Rangarajan. “Softmax to softassign: Neural network algorithms for combinatorial optimization,” Journal of Artificial Neural Networks2.4, (1996), 381-399.

[10] Hecht-Nielsen, Robert. “Theory of the backpropagation neural network,” International Joint Conference on. IEEE, (1989).

[11] Mi-young Kim, Yong-beom Cho, “Comparisons of Recognition Rates for the Off-line Handwritten Hangul using Learning Codes based on Neural Network,” Journal of IEEE, (1998), Vol.2, No.1, pp.150-159

[12] Bottou, Léon. "Large-scale machine learning with stochastic gradient descent," Proceedings of COMPSTAT'2010. Physica-Verlag HD, (2010), pp.177-186.

[13] Konečný, Jakub, et al. "mS2GD: Mini-batch semi-stochastic gradient descent in the proximal setting," arXiv preprint arXiv:1410.4744 (2014).

[14] http://www.gavrila.net/Datasets/Daimler_Pedestrian_Benchmark_D/daimler_pedestrian_benchmark_d.html