Research on Optimization Method of Convolutional Nerual Network

Xubin Feng
Photoelectric Tracking
Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences[1]
Xi'an, China
University of Chinese Academy of Sciences[2]
China
e-mail: 227927192@qq.com

Xiuqin Su
Photoelectric Tracking
Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences
Xi'an, China
e-mail: suxiuqin@opt.ac.cn

Minqi Yan
Photoelectric Tracking
Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences
Xi'an, China
e-mail: yanminqi@opt.ac.cn

Meilin Xie
Photoelectric Tracking
Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences
Xi'an, China
e-mail: xiemeilin@opt.ac.cn

Peng Liu
Photoelectric Tracking
Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences
Xi'an, China
e-mail: liupeng@opt.ac.cn

Xuezheng Lian
Photoelectric Tracking
Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences
Xi'an, China
e-mail: lianxuezheng@opt.ac.cn

Feng Jing
Photoelectric Tracking
Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences
Xi'an, China
e-mail: jingfeng@opt.ac.cn

***Abstract*—With the improvement of computers' computation and storage performance, the deep learning technology, especially the convolutional neural network (CNN) has been widely used in many fields such as Computer Vision (CV), Natural Language Processing (NLP) and Automatic Speech Recognition (ASR). CNNs have become the state-of-the-art technique in many vision tasks, such as image classification, object detection, etc. But the deep CNNs may make part of the kernels too thin by using parameterized convolution kernel to extract features. Therefore, this paper proposes a method to optimize CNNs by calculating the similarity coefficient between the feature maps. Experimental results showed that this method improved the training speed and the detecting speed with the accuracy been ensured.**

Keywords-component; convolutional neural network; similarity coefficient

I. INTRODUCTION

In recent years, CNNs have been showing outstanding effect and potential in many fields, especially in image processing, CNNs have already broken through human's own recognition ability [3], [4]. In the field of NLP, the models such as CNNs, recurrent neural network(RNN) ,gated recurrent neural network(GRU) and long-short term memory(LSTM) have achieved good results in the text classification, language model [5], machine translation [6], [7], text generation [8] and other application scenarios.

CNNs can automatically learn and distinguish the importance of features in training. For example, CNNs can extract relatively important features, such as color and texture from the image with ignoring the background automatically. Feature extraction plays an important role in deep learning, however, the more features extracted, the model performance may not be able to improve. This phenomenon is called Hughes effect. It is usually because the increase of the number of extraction features can significantly increase the sample size required for model training while adequate sample size is often difficult to

978-1-5386-5752-2/18/$31.00 ©2018 IEEE

obtain. Especially in some fields, the sample size is often inadequate.

Due to the CNNs' features selection process cannot be observed explicitly, we have to use a certain method to calculate the selected features and prune the features which are redundant to the whole training process. In this paper, Pearson correlation coefficient is used to judge whether any two features matrixes are similar or not. This method can determine whether the redundant features exist or not. This paper is structured as follows: section II introduces CNN briefly, section III introduces the Pearson correlation coefficient briefly, section IV introduces the method of calculating the similarity coefficient between feature maps to determine whether feature selection is reasonable or not, section V introduce the experiment, finally, the conclusion and outlook are given.

II. Brief Introduction of Convolutional Neutral Network (CNN)

As shown in Fig. 1, a typical CNN consists of five parts: input layer, convolution layer, pooling layer, full connection layer and output layer. Different types of CNN just are different arrangement of the five parts mentioned above.

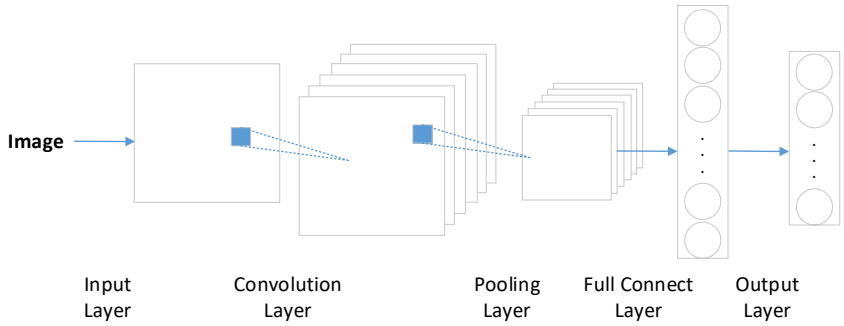

Figure 1. CNN structure.

In image processing field, the convolution layer's main function is to extract the image features and reduce the impact of noise.

The pooling layer is also called the down-sampled layer. Because of the correlation between different parts of the image, the amount of data to be processed is large and easily result in over fitting. The pooling layer usually take the adjacent four pixels' maximum value or average value to reduce the data size.

The full connection layer mainly combines features proposed by the convolution layer to perform operations such as target recognition.

CNN's three core ideas are: weight sharing, local receptive field and down-sampling.

CNNs are usually trained by the Back Propagation (BP) algorithm which take the gradient descent method as its main idea. Its cost function generally uses a non-linear function, such as Relu function which is shown in Fig. 2.

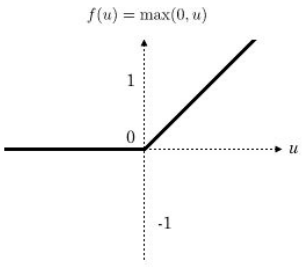

Figure 2. Relu function.

BP Algorithm is a common supervised learning method. The principle of this algorithm is to calculate the cost function between the ground truth and the actual output of the sample, then propagate backwards the cost through the layers, finally, adjust parameters (weights and bias) in the network along the negative direction of the cost function gradient. Training process will not stop until the cost function reaches the minimum.

III. Brief Introduction of Pearson Correlation Coefficient

Correlation coefficient is used to reflect the close correlation between variables statistical indicators. Correlation coefficient is calculated by the method of product difference which is based on the variance of the two variables and their respective means, then use the product of the two deviations to reflect the degree of correlation between the two variables.

According to the difference characteristics between the related phenomenon, the name of the statistic indicators are different. For example, the statistic indicator that reflecting the linear correlation between two variables is called the correlation coefficient(the square of the correlation coefficient is called the judgment coefficient), he statistic indicator that reflecting the curvilinear correlation between the two variables is called non-linear correlation coefficient, the statistic indicator that reflecting multiple linear correlations is called complex correlation coefficient, etc.

The Pearson correlation coefficient is defined as follows:

$$\rho_{x,y} = \frac{\mathrm{cov}(X,Y)}{\sigma_x \sigma_y} = \frac{E((X-\mu_x)(Y-\mu_y))}{\sigma_x \sigma_y} = \frac{E(XY)-E(X)E(Y)}{\sqrt{E(X^2)-E^2(X)}\sqrt{E(Y^2)-E^2(Y)}} \quad (1)$$

As the formula shows, the Pearson correlation coefficient between the two variable is the quotient of the covariance and standard deviation of the two variables.

The value of the correlation coefficient is between -1 and +1, that is -1<<r<<+1. Its natures are shown as follows:

- When r>0, it means the two variables are positive correlation, when r<0, the two variables are negative correlation.
- When |r|=1, it means the two variables are completely linearly related, that is the functional relationship.
- When r=0, it means there is no linear correlation between the two variables.
- When 0<|r|<1, it means there is a certain degree of linear correlation between the two variables. |r| is closer to 1, the closer the linear correlation between the two variables is; |r| is closer to 0, the weaker the linear correlation between the two variables is.
- Generally the correlation coefficient can be divided into three levels: |r|≤ 0.3 means low linear correlation; 0.3<|r|≤ 0.7 means significant correlation; 0.7≤ |r|<1 means high linear correlation.

IV. CNN Optimization Method via Feature Maps' Pearson Correlation Coefficient

CNNs usually use parametric convolution kernels to extract features. However, this approach may result in redundancy between extracted features. This not only increases the number of training parameters but also reduces the generalization ability and the network's accuracy. Conversely, too few convolution kernels can make the extracted features insufficient, thereby affecting the network's accuracy. Therefore, the number of convolution kernels can be optimized by calculating the difference between each feature map to further improve the ability of sample detection.

In CNN forward propagation, we defined a similarity coefficient R. By calculating R, We can determine the similarity of every two feature maps which are extracted by the convolution kernels in the same conclusion layer.

$$R = \rho(A, B) \tag{2}$$

As the formula shows, R is the similarity coefficient of two feature maps in the same convolution layer, P is the function of Pearson correlation coefficient, A and B respectively represent the transformed column vectors of the two n*n feature map matrices.

This method uses greedy rules to optimize layer by layer. Use the formula to calculate the similarity parameter between every two feature maps which are extracted by the convolution kernels in the first layer, then decrease the number of convolution kernels one by one until all the similarity coefficients' absolute value of first layer are less than 0.3. Then use the same method to optimize the other layers. Optimization will stop until all layers' similarity coefficients' absolute value are less than 0.3. It shows that all the extracted features can meet the requirements without redundancy at this moment.

V. Experiment

In order to verify the effectiveness of this method, we did the experiment by using the handwritten digital library—— Mnist. This dataset contains 70,000 pieces of 28*28 handwriting digital images. We used 60,000 images for training and 10,000 images for testing, the batch size was set to 50. We used Python-based Tensorflow open source software library to code. Our operating system was Ubuntu 16.04, CPU was i5 and memory was 4 gigabyte. The network's structure was based on the LeNet-5.

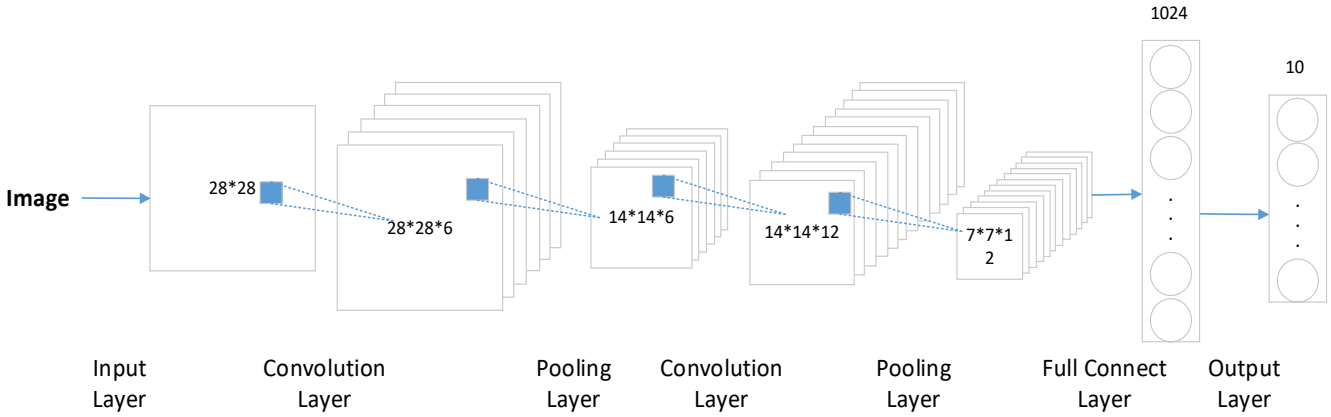

Figure 3. Experiment's network's structure.

As Fig. 3 shows, we used two convolution layers. At beginning, the first convolution layer had 6 kernels and the second convolution layer had 12 kernels. Every convolution kernel's size was 5*5 and the steps were set to 1. The learning rate and the dropout rate were set to a fixed value 0.5.

When training ended, we calculated the similarity coefficient R between every two feature maps in the first convolution layer. The results were shown in Table I.

TABLE I. SIMILARITY COEFFICIENTS BETWEEN EVERY TWO FEATURE MAPS IN FIRST CONVOLUTION LAYER

	A1	A2	A3	A4	A5	A6
A1	1	-0.0675	-0.1381	0.5814	-0.0665	-0.1050
A2		1	-0.1464	-0.2463	0.8384	0.0470
A3			1	-0.1924	-0.2160	0.3344
A4				1	-0.2041	-0.1515
A5					1	-0.0397
A6						1

It could be seen that there are several large similarity coefficients. This proved that the convolution kernels in first convolution layer was redundant. So the next step was to reduce the convolution kernel's number to five in first convolution layer. We calculated the similarity coefficient again after training. The results were shown in TABLE II.

TABLE II. SIMILARITY COEFFICIENTS BETWEEN EVERY TWO FEATURE MAPS IN FIRST CONVOLUTION LAYER

	A1	A2	A3	A4	A5
A1	1	-0.0589	0.00993	-0.2207	-0.0352
A2		1	-0.0420	-0.0457	-0.1181
A3			1	-0.0373	-0.0951
A4				1	0.0363
A5					1

It could be seen that the absolute value of similarity coefficient between every two feature maps was less than 0.3, the optimization of the first convolution layer was stopped at this moment. To verify whether this method was accurate and effective or not, we compared the network training time, the detection time and the accuracy under the two kinds of convolution kernels above. The results were shown in Table III.

TABLE III. COMPARE RESULT OF TWO KINDS OF CONVOLUTION KERNELS

First convolution layer's kernel numbers	Second convolution layer's kernel numbers	Training time(s)	Detection time(s)	Accuracy
6	12	74.9709	0.0119	94.96%
5	12	69.0132	0.01113	95.39%

TABLE IV. COMPARE RESULT OF ALL KINDS OF CONVOLUTION KERNELS IN SECOND CONVOLUTION LAYER

First convolution layer's kernel numbers	Second convolution layer's kernel numbers	Training time(s)	Detection time(s)	Accuracy
5	12	69.0132	0.01113	95.39%
5	11	72.6906	0.01336	93.52%
5	10	68.7076	0.01271	94.67%
5	9	67.3078	0.01271	94.45%
5	8	64.5753	0.01222	94.67%

Then we used the same method to optimize the second convolution layer until the difference between every two extracted feature maps in second convolution was not big. The results were shown in Table IV. We kept optimizing the second convolution layer until the convolution kernels in this layer number was 8. At this moment, the absolute value of similarity coefficients between every two feature maps in this layer were all less than 0.3.

It could be seen from Table IV that the number of convolution kernels and the accuracy was not a simple proportional relationship. Too many convolution kernels might result in redundancy between extracted features and affect network's performance. This method could provide an evaluation criterion for determining the number of convolution kernels in the network. This method also could improve the training speed and the detecting speed with the accuracy been ensured.

VI. CONCLUSION

This paper proposes a method to optimize the number of convolution kernels by calculating the similarity coefficient between feature maps. If there are one or more large similarity coefficients, that means the number of convolution kernels is redundancy. So we can optimize the network's structure and improve the training and detecting speed by reducing the number of convolution kernels. Experimental results showed that this method improved the training speed and the detecting speed with the accuracy been ensured in handwritten digits detection. It shows that this method can evaluate whether the number of convolution kernels is redundant or not and then optimize CNNs.

REFERENCES

[1] Lecun. L, Bottou. L, Bencio. Y, et al. "Gradient-based learning applied to document recognition"[J]. Proceedings of IEEE, 1988, 86(11):2278-2324.

[2] L. Chen, Q. Changwen, Z. Qiang and L. Zhi. "An Optimization Method of Convolution Neural Network:[J]. Ship Electronic Engineering, 2017(5):36-40.

[3] Szegedy C, Liu W, Jia YQ, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich, A. "Going deeper with convolutions"[C]. Computer Vision and Pattern Recognition. IEEE Computer Society, 2014. 1−9.

[4] He KM, Zhang XY, Ren SQ, Sun J. "Deep residual learning for image recognition"[C]. Computer Visionand Pattern Recognition. 2016. 770−778.

[5] Mikolov T, Karafiát M, Burget L, et al. "Recurrent neural network based language model"[C]. 11th Annual Conf. of the Int'l Speech Communication Association (INTERSPEECH 2010). 2010.

[6] Liu SJ, Yang N, Li M, Zhou M. "A recursive recurrent neural network for statistical machine translation"[C]. Association for Computational Linguistics. 2014. 1491-1500.

[7] Cho K, Merrienboer BV, Gulcehre C, et al. "Learning phrase representations using RNN, encoder-decoder for statistical machine translation". arXiv preprint arXiv:1406.1078, 2014.

[8] Sutskever I, Martens J, Hinton GE. "Generating text with recurrent neural networks"[C]. Machine Learning, (ICML 2011). 2011. 1017-1024.

ICET 2018

Electronic and Communication Engineering