Pooling Method On PCNN in Convolutional Neural Network

YAOLi-sha*, XUGuo-ming, ZHAOFeng

Institute of Information and Software, Institute of Information, Anhui Xinhua University, Hefei, Anhui230088, China

*Corresponding author’s e-mail: jsjyaolisha@163.com

Abstract. The pooling method aggregates the points in the neighborhood in Convolutional Neural Network (CNN). It can not only reduce the dimension, but also improve the results, so that the results are not easy to over-fit. However, the common pooling methods have the problems of single feature and lack of self-adaptability. In order to solve this problem, the Pulse Coupled Neural Network (PCNN) is introduced and a pooling method based on PCNN is proposed. The algorithm learns the weights of each eigenvalue from the convoluted neighborhood sub-region by PCNN and fuses them to get the final pooling result. The experimental results on image recognition datasets MNIST, CIFAR-100 show that the proposed PCNN-based pooling method has better recognition effect and improves the performance of CNN compared with the existing pooling methods.

1. Introduction

Deep learning has become a hot topic in the field of machine learning [11]-[12]. Through the construction of multiple hidden layers and the unsupervised learning of a large number of samples, deep learning can combine low-level features to form high-level features to express. Different from the traditional shallow learning theory, it makes classification or prediction simple [3]. Convolutional Neural Network (CNN) is an important model in deep learning. Convolutional neural networks were proposed by Lecun [4] as early as 1998, and the model is robust to light, translation and rotation. CNN has a unique advantage in image processing and voice recognition, benefiting from its weight sharing characteristics. Its layout is closer to the actual biological neural network, reducing the complexity of the network [5]-[7], directly carrying out end-to-end recognition, and avoiding the complexity of data reconstruction in the process of feature extraction and classification [8]-[9]. CNN mainly includes input layer, convolution layer, pooled layer, full connection layer, and output layer [10]-[11]. The pooling layer is mainly used to aggregate the eigenvalues in the neighborhood eigenmap into an eigenvalue. The pooling method reduces the dimension through aggregation operation and has translation invariance, which is beneficial to improve the performance of CNN. At the same time, the pooling operation can also effectively avoid the over-fitting phenomenon caused by the complex model and too many parameters in the training process to improve the results.

Common pooling methods are maximum pooling and average pooling. Each of the above two pools has its own characteristics. The maximum pooling method retains significant features, but loses some features and does not retain diversity information. The maximum pooling method retains the diversity feature and considers the pixel relevance, but weakens the significant feature and lacks the self-adaptability. A new stochastic pooling method is proposed by Zeiler [12]. Although the stochasticization method considers the correlation, the characterization ability is insufficient, and only the ReLU activation function can be used [13]-[15].
Therefore, this paper proposes a pooling method of convolution Neural Network based on Pulse Coupled Neural Network (PCNN). The pooling value of convolution feature graph is obtained by feeding the pooling sub-region divided by the pooling window into PCNN autonomous learning weights, so as to improve the performance of the convolution Neural Network.

2. Pooling method based on PCNN

Theoretical and empirical studies show that different pooling methods directly affect the results of CNN network model. The maximum pooling method retains significant features, but loses some features, and does not retain diversity information. The average pooling method retains the diversity feature and considers the pixel correlation, but weakens the significant feature and lacks the self-adaptability. The correlation is considered in stochastic pooling, but the problem of feature loss is not solved completely. Therefore, this paper proposes a pooling method of convolution neural network based on pulse-coupled neural network (PCNN). It sends the sub-convolution feature graph divided by pooling window into the CNN based on PCNN. According to the importance of different features, weight coefficients are allocated through PCNN autonomous learning, and the final pooled value is obtained by integrating weights. Reasonable allocation of adaptive weight coefficient will be beneficial to improving the overall performance of CNN.

As an artificial neural network, PCNN can simulate the ability of human brain to learn autonomously. By training the appropriate parameters of autonomic learning for different sample sets, PCNN can flexibly and adaptively adjust the weight coefficients of each characteristic value.

2.1 PCNN model

The PCNN model includes three parts: receiving part, modulator, and pulse generator [16]. The single neuron model composed of PCNN is shown in figure 1. Its work satisfies equation (1)-(5):

$$ F_j(n) = I_j $$

$$ L_j(n) = e^{-\alpha_L} L_j(n-1) + V_L \sum_{k,l} W_{kl} Y_{kl}(n-1) $$

$$ U_j(n) = F_j(n) (1 + \beta) L_j(n) $$

$$ \theta_j(n) = e^{-\alpha_\theta} \theta_j(n-1) $$

$$ Y_j(n) = \begin{cases} 1 & U_j(n) > \theta_j(n) \\ 0 & U_j(n) \leq \theta_j(n) \end{cases} $$

The receiving part mainly receives feedback input and connection input, the regulator mainly forms internal activities, and the pulse generator mainly produces dynamic threshold and neuron output.

Where, $F_j$ is the feedback input, $I_j$ is the external input, namely the eigenvalue in the image pooling domain. $L_j$ is the connection input, $\alpha_L$ is the attenuation coefficient, $V_L$ is the amplitude, $W$ is the connection matrix, $U_j$ is the internal activity item, $\beta$ is the connection strength, $\theta_j$ is the dynamic threshold. In the first iteration, the threshold starts to decay from the initial value, and $\alpha_\theta$ is the time decay coefficient of $\theta_j$. $Y_j$ is the output of the neuron. When the dynamic threshold $\theta_j$ attenuates to less than the internal activity item $U_j$, the neuron outputs $Y_j$ as 1, otherwise, the neuron outputs $Y_j$ as 0. When the neuron output $Y_j$ is 1, the function $T_j$ which records the number of fires plus 1. After ignition, the dynamic threshold suddenly increases and inhibits the second output of the neuron, so the dynamic threshold starts to decay again. When the dynamic threshold attenuates to less than the internal activity, the neuron output outputs 1 again. After such iteration for Nmax times, the ignition times $T_j$ corresponding to point $(i, j)$ in the characteristic graph is generated. In
the iterative process, the function $T_{ij}$ of ignition times is expressed as:

$$T_{ij}(n) = T_{ij}(n-1) + Y_{ij}(n)$$  \tag{6}$$

(1) After iterating $N_{max}$ times, PCNN obtained the ignition times of each eigenvalue $f_{ij}$ in the convolution layer characteristic graph pooling sub-window $m \times m$, which were respectively denoted as $t_{ij}$.

(2) The sigmoid threshold function is used to map the ignition times $t_{ij}$ corresponding to each eigenvalue in the pooled sub-window to between $[0,1]$ and take it as the fusion weight coefficient $w_{ij}$ of each eigenvalue in the child window, as shown in equation (7).

$$w_{ij} = \frac{1}{1 + e^{-t_{ij}}}$$  \tag{7}$$

(3) The final pooling result value $S$ of the pooled sub-window is determined according to the fusion weight coefficient, as shown in equation (8).

$$S = \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} \times f_{ij}$$  \tag{8}$$

3. Experimental results and analysis

3.1 CNN model structure design

The structure design of the CNN model designed in this paper is shown in figure 2. The CNN model structure has 8 layers (excluding the input layer). As shown in table 1, the parameter settings of each layer include 3 layers of convolution layer C1, C2 and C3, and 3 layers of pooling layer S1, S2, S3 and 1 layer of output layer. In this paper, ReLu activation function is adopted, Dropout layer is added before the full connection layer to improve the generalization ability of the model, and SoftmaxLoss loss function is adopted. The study found that the optimal pooling domain was $3 \times 3$, and too large pooling domain could easily lead to over-fitting or noise, so this paper chose the pooling domain of $3 \times 3$. 
3.2 Comprehensive analysis of different pooling methods

In order to verify the performance of the PCNN pooling method proposed in this paper, the experiment adopted different pooling methods for comparative analysis based on the same CNN model designed in section 3.1.

In order to reduce the randomness of the experiment, the data set MNIST, CIFAR-10, CIFAR-100 were respectively used in the experiment to calculate the average recognition rate of the image by 5 times of 50% cross validation.

Example 1: MNIST digital script data set is selected, which consists of 60,000 training sets and 10,000 test sets. It contains 10 types of grayscale images of 0-9, and the size of each image is 28×28. The grayscale value of pixel points is normalized to [0, 1]. A sample image of MNIST dataset is shown in figure 4(a). Table 2 shows the identification results of different pooling methods on the MNIST data set.

Table 2. The recognition rate of pooling methods on MNIST data set

| Pooling methods         | Recognition rate(%) |
|-------------------------|---------------------|
| Maximum pooling         | 99.28               |
| Mean pooling            | 99.51               |
| Stochastic pooling      | 99.56               |
| PCNN pooling            | 99.56               |

As can be seen from the results in table 2, PCNN pooling method has a good identification effect. However, due to the simplicity of MNIST data set, compared with other traditional pooling methods, the recognition rate did not improve much.

Example 2: Select the CIFAR-100 data set, which contains 100 categories of natural color images. Table 3 shows the identification results of different pooling methods on the CIFAR-100 data set.

Table 3. The recognition rate of pooling methods on CIFAR-100 data set

| Pooling methods   | Recognition rate(%) |
|-------------------|---------------------|
| Maximum pooling   | 80.14               |
| Mean pooling      | 80                  |
As can be seen from table 3, PCNN pooling method has the best identification effect. Among other traditional pooling methods, stochastic pooling method has the best effect and the highest recognition rate. Compared with the stochastic pooling method, the recognition rate of this method is improved by 1.52%.

4. Conclusion
The pooling method is mainly to aggregate the eigenvalues in the neighborhood eigenmap into an eigenvalue. The pooling operation can reduce the dimension, have a certain degree of translation invariance, also can prevent over-fitting, improve the generalization ability. It is conducive to improving the performance of CNN. However, the maximum pooling method retains significant features, but loses some features and does not retain diversity information. The mean pooling method retains the diversity feature and considers the pixel correlation, but weakens the significant feature and lacks the self-adaptability. Stochastic pooling preserves the diversity characteristics and considers the pixel relevance, but it does not completely solve the problem of information loss. In view of the shortcomings of traditional pooling methods, this paper proposes a pooling method based on PCNN. This method introduces PCNN neural network, distributes weight coefficients through PCNN autonomous learning, and fuses the weights to get the final pooled value. In this paper, CNN-based neural network was constructed and different pooling methods were verified by MNIST and CIFAR-100 data sets. Experimental results show that PCNN pooling method has the best identification results. Compared with the existing pooling method, the recognition effect of this method has been improved and the performance of CNN has been improved.

Acknowledgments
Special thanks to the following funds for their support: Key Projects in Natural Science Research in Anhui University (No. KJ2018A0587); Anhui Xinhua university key research project (No. 2018zr006); Anhui Xinhua university key research project (No. 2018zr001); Provincial quality engineering in Anhui Province Grass-roots teaching and research office demonstration project (No. 2018 jysf111).

References
[1] A.K. Sampath, Dr. N. Gomathi. Decision tree and deep learning based probabilistic model for character recognition [J]. Journal of Central South University, 2017, 24(12): 2862-2876.
[2] Murat Abdughani, REN J, WU L, YANG J M, ZHAO J. Supervised Deep Learning in High Energy Phenomenology: a Mini Review [J]. Communications in Theoretical Physics, 2019, 71(08): 955-990.
[3] Polapat Khlongkhoi, Kittisak Chayantrakom, Wattana Kanbua. Application of a deep learning technique to the problem of oil spreading in the Gulf of Thailand [J]. Advances in Difference Equations, 2019, 2019 (1): 1-9.
[4] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
[5] Dipanjan Moitra, Rakesh Kr. Mandal. Automated AJCC (7th edition) staging of non-small cell lung cancer (NSCLC) using deep convolutional neural network (CNN) and recurrent neural network (RNN) [J]. Health Information Science and Systems, 2019, 7(1): 1-12.
[6] Eren L, Ince T, Kiranyaz S. A Generic Intelligent Bearing Fault Diagnosis System Using Compact Adaptive 1D CNN Classifier [J]. Journal of Signal Processing Systems, 2019, 91(2): 179-189.
[7] Xu M, Qi S, Yue Y, et al. Segmentation of lung parenchyma in CT images using CNN trained with the clustering algorithm generated dataset [J]. BioMedical Engineering OnLine, 2019, 18(1): 2.
[8] Tanzila Saba, Muhammad Attique Khan, Amjad Rehman, Souad Larabi Marie-Sainte. Region Extraction and Classification of Skin Cancer: A Heterogeneous framework of Deep CNN.
Features Fusion and Reduction [J]. Journal of Medical Systems, 2019, 43(9): 1-19.

[9] Pourya Shamsolmoali, Deepak Kumar Jain, Masoumeh Zareapoor, Jie Yang, M. Afshar Alam. High-dimensional multimedia classification using deep CNN and extended residual units [J]. Multimedia Tools and Applications, 2019, 78(17): 23867-23882.

[10] Zhang S, Xu X, Pang Y, et al. Multi-layer Attention Based CNN for Target-Dependent Sentiment Classification [J]. Neural Processing Letters, 2019(3): 1-15.

[11] Zhai Y, Cao H, Deng W, et al. BeautyNet: Joint Multiscale CNN and Transfer Learning Method for Unconstrained Facial Beauty Prediction [J]. Computational Intelligence and Neuroscience, 2019(4): 1-14.

[12] Zeiler M D, Fergus R. Stochastic pooling for regularization of deep convolutional neural networks [J]. Eprint Arxiv, 2013: 1301.3557.

[13] Javad Abbasi Aghamaleki, Vahid Ashkani Chenarlogh. Multi-stream CNN for facial expression recognition in limited training data [J]. Multimedia Tools and Applications, 2019, 78(16): 22861-22882.

[14] Long C, Hu X, Wei T, et al. Parallel Planning: A New Motion Planning Framework for Autonomous Driving [J]. IEEE/CAA Journal of Automatica Sinica, 2019, 6(1): 1-12.

[15] Yuan C, Xia Z, Jiang L, et al. Fingerprint Liveness Detection Using an Improved CNN With Image Scale Equalization [J]. IEEE Access, 2019, 7(99): 26953-26966.

[16] Ding S, Zhao X, Hui X, et al. NSCT-PCNN image fusion based on image gradient motivation [J]. Iet Computer Vision, 2018, 12(4): 377-383.

[17] Lecun Y, Haffner P, Bengio Y. Object recognition with gradient-based learning [C]. In: Proceedings of 1999 Shape, Contour and Grouping in Computer Vision. 1999: 751-760.

[18] Krizhevsky A. Learning multiple layers of features from tiny images [J]. Computer Science Department, University of Toronto, 2012: 1-7.