Multi-scale Hybrid Pooling Convolutional Neural Network Algorithm

Nan ZHAO\textsuperscript{1,2,*}, Xin WANG\textsuperscript{2}, Ying-na LI\textsuperscript{1} and Sheng WU\textsuperscript{1}

\textsuperscript{1}Faculty information Engineering and Automation, University of Science and Technology, Kunming 650500, China
\textsuperscript{2}Yunnan Power Grid Company Electric Power Research Institute, Kunming 650217, China
\textsuperscript{*}Corresponding author

Keywords: MSHP-CNN, Image processing, Data processing.

Abstract. With the popularization of the modern Internet and the rise of the mobile Internet, the data content of computer processing is more and more diverse. In this paper, we propose a multi-scale hybrid pooling algorithm based on convolutional neural network model, through convolution. The layer convolution method and the pooling mechanism of the pooling layer are improved to improve the generalization ability of the overall model.

Learning Algorithm Principle and Description

The Multi-scale Hybrid Pooling Convolutional Neural Network (MSHP-CNN) model proposed in this chapter is an improved model based on the CNN model. The model is also the same in the structure of the network model. It is composed of input layer I, convolution layer C, pooling layer P, full connection layer F and output layer O, as shown in Figure 1-1:

![Figure 1-1. Schematic diagram of multi-scale mixed pooled convolutional neural network model.](image)

In the pooling layer, unlike the traditional network structure, the multi-scale hybrid pooled convolutional neural network model adopts a mechanism of dynamic adjustment based on image information distribution.

Pooling algorithm and average pooling algorithm are as follows:

\[ X_j^l = f(u_j^l) \]  
\[ u_j^l = \beta_j^l \max(X_{i-1}^l) + b_j^l \]  
\[ u_j^l = \beta_j^l \text{avg}(X_{i-1}^l) + b_j^l \]  

Information entropy can be calculated using Equation 3-4 and Equation 3-5:

\[ H = \sum_{ij}^{255} p_{ij} \log p_{ij} \]
\[ p_j = \frac{f(i, j)}{M \times N} \]  

(3-5)

In Equation 3-4, H represents the obtained information entropy value, in Equation 3-5, the frequency at which the pixel gray value of position (i, j) appears, and M*N is the image size to be calculated.

**Experimental Result**

**Analysis of Results**

Experiment of MSHP-CNN algorithm in MNIST data set

Through multiple tests, the learning rate is set to 0.001, and the convolution kernel scale is 3*3, 5*5, 7*7 convolution kernel extraction features, which can achieve the optimal experimental results. The MNIST standard data set contains 70,000 handwritten digital images, each of which is a 28*28 pixel matrix consisting of 0 to 9 ten-digit samples. In this chapter, the experimental training set and test set are randomly selected from the 70,000 handwritten digital images to train the model, and another part of the test model is used. The specific allocation is shown in Table 2-1:

| Training set | Test set        |
|--------------|-----------------|
| 30000        | 5000/10000      |
| 50000        | 10000/15000     |
| 70000        | 20000/25000     |

The MSHP-CNN algorithm model extracts the images from the convolution kernels of different scales in the convolution layer in the process of training the model using MNIST dataset, as shown in Figure 2-3, which is 3*3, 5*5, 7*7 convolution kernels:

![Figure 2-3. Display of convolution kernels of different scales.](image)

As shown in Figure 2-4 below, for the original image input using the MNIST data set, the handwritten number “7” is taken as an example to illustrate the different feature mapping grading process in the pooling layer of the model training process.

![Figure 2-4. Original image.](image)

The number of training data samples in the training process is visualized in the MSHP-CNN algorithm in the pooled layer feature map as shown in Figure 2-5. From left to right, the feature maps are displayed in the order of 30000, 50000, and 70,000 training samples. It can be seen that when the number of training samples increases, the feature map of the pooling layer is more similar to the original image. Obviously, the more accurate the training samples, the higher the classification accuracy of the algorithm model.
After the CNN model learns by MSHP-CNN algorithm, the error rate increases with the number of iterations, as shown in Figure 2-6. After the number of iterations reaches 400, the model tends to be stable, and the error rate approaches 0.15%, which is the classification accuracy. Up to 99.85%. Compared with the SVM learning algorithm, the average classification accuracy of 91.7% is greatly improved. Compared with the standard CNN algorithm, the average classification accuracy is 97% and the improved MS-CNN (Multi-Scale CNN) algorithm averages 97.5%. Accuracy, the average classification accuracy of MSHP-CNN proposed in this chapter can reach 99.3%, and in the best case, it can achieve 99.85% classification accuracy.

**MSHP-CNN Algorithm in the Cifar-10 Data Set Experiment**

The Cifar-10 dataset includes a total of 60,000 physical images of a colored background, each of which is a 32*32 pixel matrix consisting of ten samples of airplanes, birds, and deer, each of which consists of 6000 sheets. Image composition. In this experiment, the Cifar-10 data set is divided into 45,000 training data sets and 15000 test data sets. The test data set is composed of 1500 images extracted from 10 types of image samples. The second verification experiment in this chapter is based on the MNIST dataset, and experiments are performed on a more complex dataset such as Cifar-10. The purpose of this is to more fully verify the algorithm proposed in this paper. Scalability and effectiveness. The same method was used to model and test on the Cifar-10 dataset. The following figure 2-7 shows the results of a raw image visualization randomly extracted from the Cifar-10 dataset, as shown in Figure 2-8 below. A feature map extracted by the convolution kernel for the original image.

As shown in Figure 2-9 below, the neural network model trained by MSHP-CNN algorithm has an increasing classification accuracy with the number of iterations. When the number of iterations reaches 1500, the accuracy of the model reaches a basically stable state. The classification accuracy rate no longer increases significantly with the number of iterations to reach 94.8%, which is different from the standard CNN algorithm and CNN's improved algorithm MS-CNN (Multi-Scale CNN) algorithm. The MSHP-CNN algorithm extracts the original image information from different dimensions by using the multi-scale convolution kernel in the feature extraction stage, and preserves the feature mapping effective information by selecting a more suitable pooling method in the pooling layer. Differentiate image information to improve classification accuracy.
Conclusion

The above experimental process was repeated a lot, and the trained neural network model was tested. By comparison, it was found that the new method improved the model classification accuracy by about 2.7% compared with the previous classical CNN method. This result is compared with the CNN algorithm. The improved algorithm MS-CNN (Multi-Scale CNN) algorithm also improved the classification accuracy of the model by 1.5%. The experimental results show that the multi-scale mixed pooled convolutional neural network classification algorithm proposed in this chapter is effective. However, compared with the results obtained, the shortcoming of this chapter is that the experimental part of the proposed MSHP-CNN algorithm only uses the MNIST data set and the Cifar-10 data set. The algorithm is in other common data sets such as Cifar-100 and STL-10. Whether Tiny-Image, etc. can also have better performance needs further verification.

References

[1] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553):436-444.
[2] Schmidhuber J. Deep learning in neural networks: An overview[J]. Neural Networks, 2015, 61
[3] Hameed A A, Karlik B, Salman M S. Back-propagation algorithm with variable adaptive momentum[J]. Knowledge-Based Systems, 2016, 114:79-87.
[4] Bengio Y. Learning Deep Architectures for AI[J]. Foundations & Trends® in Machine Learning, 2009, 2(1):1-127.