

Research Article

Addressing Overfitting Problem in Deep Learning-Based Solutions for Next Generation Data-Driven Networks

Mansheng Xiao,^{1,2} Yuezhong Wu ,¹ Guocai Zuo ,³ Shuangnan Fan,² Huijun Yu,¹ Zeeshan Azmat Shaikh,⁴ and Zhiqiang Wen¹

¹School of Computer Science, Hunan University of Technology, Zhuzhou 412007, China

²College of Electrical and Information Engineering, Hunan Institute of Traffic Engineering, Hengyang 421009, China

³School of Software and Information Engineering, Hunan Software Vocational and Technical University, Xiangtan 411100, China

⁴Department of Electrical Engineering, University of the Punjab, Lahore 54660, Pakistan

Correspondence should be addressed to Yuezhong Wu; wuyuezhong@hut.edu.cn and Guocai Zuo; 474025986@qq.com

Received 27 May 2021; Revised 29 June 2021; Accepted 20 July 2021; Published 15 August 2021

Academic Editor: Dr. Muhammad Shafiq

Copyright © 2021 Mansheng Xiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Next-generation networks are data-driven by design but face uncertainty due to various changing user group patterns and the hybrid nature of infrastructures running these systems. Meanwhile, the amount of data gathered in the computer system is increasing. How to classify and process the massive data to reduce the amount of data transmission in the network is a very worthy problem. Recent research uses deep learning to propose solutions for these and related issues. However, deep learning faces problems like overfitting that may undermine the effectiveness of its applications in solving different network problems. This paper considers the overfitting problem of convolutional neural network (CNN) models in practical applications. An algorithm for maximum pooling dropout and weight attenuation is proposed to avoid overfitting. First, design the maximum value pooling dropout in the pooling layer of the model to sparse the neurons and then introduce the regularization based on weight attenuation to reduce the complexity of the model when the gradient of the loss function is calculated by backpropagation. Theoretical analysis and experiments show that the proposed method can effectively avoid overfitting and can reduce the error rate of data set classification by more than 10% on average than other methods. The proposed method can improve the quality of different deep learning-based solutions designed for data management and processing in next-generation networks.

1. Introduction

At present, the direction supported by the internet is changing from consumption to production, but the network architecture based on TCP/IP cannot adapt to this change in scalability, security, and other aspects. On the other hand, big data technology is also emerging in various industries. These emerging technologies are in the early stage of development, and there are still many problems to be solved [1–3]. In recent years, with the development of 5g technology, the amount of data stored in the computer system is increasing. Due to the diversity and uncertainty of massive data classification, there are a series of problems in data-driven network communication, such as the existence of these massive data not only occupies a lot of network space, but also

causes network congestion [4, 5]. Therefore, how to classify and process the massive data before data transmission to reduce the amount of data transmission in the network is a very worthy problem [6–9]. The development of deep academic technology provides a good solution to this problem. At present, artificial intelligence technology based on deep learning has become more and more widely used in various fields. Convolutional neural network (CNN) is one of the core technologies of deep learning [4]. Compared with fully connected neural networks, CNN has features such as local connection, weight sharing, and downsampling can greatly reduce the complexity of the model and improve the training efficiency and data processing accuracy of the model. Therefore, it has been highly recognized by scholars in the field, and its application is gradually being promoted [10–14].

However, during the training process of the CNN model, due to the characteristics of the training data set itself, such as the data set is too small and contains a lot of noise, or the structure of the model itself, such as the model is too complex, the training parameters are too much, and the training is too much, the model is very likely to fall into overfitting [15–17]. For example, because the number of training samples is too small, the network parameters obtained after training cannot accurately simulate the distribution of all samples, or a large number of noise samples are fitted during the training process, ignoring part of the correct sample data, so the model fits the training set data very well. Good, but the poor fitting effect to the data outside the training set is reflected in the small loss function value (deviation) during training, while the loss function value is very large in verification or testing [18].

Since the development of machine learning technology, many experts and scholars have been conducting research on overfitting problems in the regression process and have achieved a series of research results. For example, Li et al. [19] proposed earlier to solve the overfitting problem in the algorithm, and Hinton et al.'s literatures [20, 21] first proposed the dropout method in machine learning to solve the overfitting problem. Chen et al. [22] adjusted algorithm parameters and iterations through singular value decomposition (SVD) to avoid the occurrence of overfitting phenomenon, and other methods such as literatures [23–25] have also carried out related research. In view of the overfitting problem existing in the current deep learning CNN model, many scholars and engineering technicians are currently working on the research of this problem, and a series of solutions have been proposed [10, 11, 13, 16, 18, 26, 27], including data expansion enhancement, regularization, discarding, and early stop method. However, affected by the research environment, conditions, and scope of application, the related theories for the development of deep learning technology are not mature enough; these research results can be described as different, each has its own advantages, and there are certain limitations. For example, in response to problems such as long training time and overfitting of the CNN model, Gong et al. [16] proposed to improve the CNN based on the immune system, but the accuracy of this improved network model on the test set is not high (only 81.6%). In terms of processing training data sets, Yang et al. [10] proposed an attribute reduction algorithm based on visual ranging, which solved the network overfitting problem by introducing Bayesian weight factor distribution instead of CNN fixed weights, but its application is very limited. In terms of data enhancement, literature [11] is aimed at the overfitting problem of the deep learning breast mass detection algorithm for synthetic images, using synthetic mammograms for training data enhancement, which is a common method. In the training process of the model, literature [26] conducted a systematic study on the standard reinforcement learning agent (reinforcement learning agent), conducted a general discussion on reinforcement learning overfitting, and generalized it from the perspective of inductive bias. Research on the behavior of deep learning did not propose a unique solution to the problem of deep learning overfitting. Literature [18]

proposed a model prediction averaging method based on dropout double probability weighted pooling, which effectively reduces the error rate and inhibits overfitting, but the convergence speed of the algorithm becomes slow after the introduction of double probability. In the CNN model design, literature [12] introduced the particle swarm optimization (PSO) algorithm to reduce the back propagation of the error, avoid the lag error and the image overcombination, and improve the convergence speed, but the PSO is introduced to the CNN weight update lack of theoretical basis. Other methods have more or less “side effects” while improving the efficiency of CNN’s data processing [6].

Based on the advantages of these new technologies, this paper proposes a data-driven network architecture, which is aimed at solving the problem of massive data filtering and classification in the development of the emerging future network, and the specific contents are as follows: based on the full analysis of the characteristics of CNN and the current research on CNN model overfitting by scholars at home and abroad, this paper proposes an algorithm for the maximum pooling dropout and weight attenuation overfitting problem, and through the theoretical derivation, as well as the image data collected in the network for classification experiment comparison, it proves that this method can effectively avoid overfitting in the training process of CNN model and improve the generalization ability of the model. The convolutional neural network overfitting prediction method proposed in this paper can classify the massive data in the network, reduce the amount of data transmission, and improve the communication efficiency [28–31].

The main work of this paper is as follows: First, it analyzes the problems existing in mass data transmission based on data-driven network architecture and proposes mass data classification method of CNN in deep learning technology. The second is to design a CNN overfitting prediction model for maximum pooling dropout and weight attenuation, so as to reduce the overfitting in model training and provide the classification accuracy of the model. Thirdly, three experiments are designed to verify the effectiveness of the model.

The structure of this article is as follows. The second part introduces convolutional neural network and related technologies. The third part introduces maximum pooling dropout and the CNN model of weight attenuation. After proposing the overfitting prediction method, the fourth part is about analysis of experimental results. The last part includes conclusion.

2. Convolutional Neural Network and Related Technologies

A CNN structure generally consists of a convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer and the pooling layer are alternately connected. After the convolutional layer is calculated, the pooling layer starts to execute, and then, convolution and pooling are performed. In this way, the convolution operation and the pooling operation alternately perform the extraction of sample features, and then, the fully connected layer is used to classify the extracted features. Because CNN has the characteristics of

local connection, weight sharing, and downsampling, it is compared with the general neural network, and the extraction process has fewer connection parameters (weights), fewer feature dimensions, and stronger representation capabilities, so it has better generalization [8]. However, in CNN model training, due to various reasons, overfitting often occurs. Overfitting means that the gap between training error and test error is too large. That is to say, the complexity of the model is higher than that of the actual problem, and the model performs well in the training set, but poorly in the test set.

Let x_i^l be the i -th feature map of the l -th layer of the input sample, $i = 1, 2, \dots, f^l$, f^l is the total number of feature maps of the l -th layer, $w^{l,k}$ is the k -th convolution kernel of the l -th layer, and $b^{l,k}$ is the l -th layer corresponding to the k -th convolution kernel. $z^{l,k}$ is the output of the k -th convolution operation of the l -th layer, and $f(x^l)$ is the activation function of the l -th layer. The operation process of the convolutional layer is expressed as follows:

$$z^{l+1,k} = \sum_{i=1}^{f^l} \text{conv}(w^{l,k}, x_i^l) + b_i^{l,k}, \quad (1)$$

$$x^{l+1,k} = f(z^{l+1,k}). \quad (2)$$

Among them, the formula (1) indicates that the convolution matrix obtained by convolution between x_i^l and the corresponding convolution kernel $w^{l,k}$ is summed and the corresponding offset is added to obtain the input $z^{l+1,k}$ of the next layer ($l+1$). After the activation function is processed, you can get the k -th output matrix $x^{l+1,k}$ of the $l+1$ -th layer.

The next layer of the convolutional layer is the pooling layer. The pooling layer interprets the features of the convolutional layer, thereby reducing the feature dimension and network complexity. The pooling operation is as follows:

$$s_m^{l+1} = \text{pool}(s_{m,1}^l, s_{m,2}^l, \dots, s_{m,j}^l, \dots, s_{m,n}^l), \quad (3)$$

where $s_{m,j}^l$ represents the value of the j -th pooling unit in the m -th pooling area in the feature map obtained by the convolution operation (i.e., matrix $x^{l+1,k}$), l represents the pooling layer where it is located, and n is the number of pooling units in the pooling area. If $2 * 2$ pooling is used, then $n = 4$, $\text{pool}(\cdot)$ means pooling operation, and average pooling or maximum pooling is often used. Maximum pooling is widely used because it can retain its representative characteristics. This article uses the largest value pooling method.

After the pooling is completed, perform the full connection operation, calculate the loss function value, and determine whether to back propagate. The loss function value calculation is one of the important links in CNN. Its value is directly related to the efficiency and quality of the CNN.

The cross-entropy function is commonly used, expressed as follows:

$$L = -\frac{1}{N} \sum_{i=1}^N (y_i \ln(o_i)). \quad (4)$$

Among them, N is the number of training samples, y_i is the actual label value of sample i , and o_i is the network output value of sample i .

3. Maximum Pooling Dropout and Weight Attenuation CNN Model

The previous section introduced the basic structure of the convolutional neural network and the calculation methods of each network layer. On the basis of the above, this section designs a maximum pooling dropout and weight attenuation CNN model in order to avoid excessive in the model training process and the occurrence of the fitting situation.

3.1. Maximum Pooling Dropout. As explained in Section 2, maximum pooling is a commonly used method in the CNN pooling layer. In order to avoid overfitting during model training, this paper introduces the maximum pooling dropout in the CNN pooling layer.

Suppose the retention probability of each pooling area in the feature map to be pooled is p , and the size of p can be manually adjusted according to the actual situation. Generally, it is set to $p = 0.5$. Then, the inhibition probability of each unit in the pooling area is $q = 1 - p$. At the same time, it is assumed that the unit values ($s_{m,1}^l, s_{m,2}^l, \dots, s_{m,n}^l$) in each pooling area m of the l layer are rearranged in ascending order, that is, the unit value after the arrangement is as follows: $0 < d_{m,1}^l < d_{m,2}^l < \dots < d_{m,n}^l$ (note: the semilinear activation function ReLU is used here to make the activation value of all units nonnegative, so the smallest $d_{m,1}^l > 0$); then, $d_{m,j}^l$ is selected as the maximum value of the entire pooling area. The output after pooling is as follows: all unit values ($d_{m,j+1}^l, d_{m,j+1}^l, \dots, d_{m,n}^l$) greater than $d_{m,j}^l$ are suppressed, and only values ($0, d_{m,1}^l, d_{m,2}^l, \dots, d_{m,j}^l$) less than or equal to $d_{m,j}^l$ are suppressed retained, because of the maximum pooling (take the maximum value) among these retained values, the final output value of pooling is $d_{m,j}^l$, and the probability of its occurrence is $d_{m,j}^l$, that is,

$$p_j = \text{poss}(s_m^{l+1} = d_{m,j}^l) = pq^{n-j}, j = 1, 2, \dots, n. \quad (5)$$

In the above formula, p_j is the probability of the reserved output of the m -th pooled cell in the j -th pooled region, which is the product of the reserved probability of the entire pooled region and the suppressed probability of the suppressed cell, and n is the number of cells in the pooled region. If the pooled region is $3 * 3$, then the number of pooled units $n = 9$. Analyzing formula (5), it can be seen that when the maximum value pooling dropout is

performed in the pooling area, the j -th activation value $d_{m,j}^l$ in the pooling area is selected as the output value of the pooling area through polynomial arrangement, which is

$$s_m^{l+1} = d_{m,j}^l, p_j \in (p_1, p_2, p_3, \dots, p_n). \quad (6)$$

Suppose there are r feature maps in layer l , each feature map has a size of s , and a pooling area size of n , if overlapping pooling is not considered, there are rs/n pooling areas, then the model parameter that the l layer may need to be trained is $f(j) = (j+1)^{rs/n}$ (plus a bias), that is, the number of model parameter that the maximum pooling may need to train is exponentially related to the number of pooling area units input to the pooling layer. Because the function $f(j) (j > 1)$ is an increasing function, so after introducing the maximum pooling dropout, the pooling unit is randomly suppressed, that is, j is reduced, and the parameter value of the model to be trained is reduced exponentially, which effectively reduces the complexity and thus can more effectively suppress overfitting.

3.2. Weight Attenuation Regularization Method. When training a large CNN, in addition to using the abovementioned maximum pooling dropout suppression unit to avoid overfitting and if the function value changes drastically in some areas of the model, it means that the parameter value (weight) of the function is too large, making the area. The absolute value of the derivative value is large, and the model becomes complicated. The weight attenuation restricts the norm of the parameter so that it cannot be too large, so as to reduce the complexity of the model and reduce the influence of noise input, thereby reducing the occurrence of excessive fitting, and this method is also called regularization method.

Suppose that the loss function L_0 of the model is shown in formula (4). When calculating the weight attenuation, a penalty term is added to the original loss function L_0 , namely,

$$L = L_0 + \frac{\lambda}{2} \sum_w \|w\|^2. \quad (7)$$

In formula (7), L_0 is the original loss function (see formula (4)), w is the network weight, that is, the connection coefficient of neurons, and $\lambda (\lambda > 0)$ is the penalty coefficient, which is used to measure the ratio of the penalty to L_0 relationship, and $1/2$ is designed for the convenience of derivation. The above penalty term is the sum of the squares of the network weight w . Taking the derivative of (7), we get the following:

$$\begin{cases} \frac{\partial L}{\partial w} = \frac{\partial L_0}{\partial w} + \lambda w, \\ \frac{\partial L}{\partial b} = \frac{\partial L_0}{\partial b}. \end{cases} \quad (8)$$

In formula (8), b is the bias of the network neural unit (such as the convolution kernel), which is included in the o_n of L_0 , that is, $o_n = w \cdot x_{n-1} + b$. From the above formula, it can be found that after the penalty term is added, it has no effect on the update of the bias b , but for the weight value w ,

$$w' = w - \eta \left(\frac{\partial L_0}{\partial w} + \lambda w \right) = w - \eta \frac{\partial L_0}{\partial w} - \eta \lambda w = (1 - \eta \lambda) w - \eta \frac{\partial L_0}{\partial w}. \quad (9)$$

Among them, η is the learning rate. By analyzing formula (9), it can be seen that the coefficient of the weight value w is 1 before the penalty term is introduced, that is, $w' = w - \eta (\partial L_0 / \partial w)$. After the penalty term is added, the coefficient before w becomes $1 - \eta \lambda$, because both η and λ are positive numbers less than 1, so $1 - \eta \lambda < 1$, that is, the effect of formula (9) is to reduce the value of w , which is the theoretical meaning of weight attenuation. Note that the $\eta (\partial L_0 / \partial w)$ term in formula (9) is the gradient of the weight change of backpropagation. The expression is the same regardless of whether the penalty term is added or not. Therefore, the weight attenuation term mentioned here does not include this term. For further analysis of formula (9), when w is positive, the updated w' becomes smaller, and when w is negative, the updated w' becomes larger. Because of $|w| < 1$ (the weight after the network is normalized) in formula (9), the effect is to make w close to 0, that is, $|w| \rightarrow 0$, to make the value of w as small as possible, which is equivalent to reducing the weight of the network, reducing the complexity of the network, and avoiding overfitting. It should be pointed out that the setting of the value of parameter λ in formula (9) is very important. If λ is too large, the weight w decreases too fast, underfitting may occur, or even training may not be possible, and if λ is too small, overfitting may occur. Together, the λ size setting can be adjusted based on the Bayes decision rule. This method assumes that the weights and biases of the network are random variables with specific distributions and are automatically calculated by statistical methods. For details, please refer to literature [32] (due to space limitations, there are no more details).

4. Experimental Results

4.1. Experimental Setup. Network training adopts Stochastic Gradient Descent, SGD, batch size = 100, initial learning rate $\eta = 0.1$, and when the error tends to be flat, reduce η and use a normal distribution with a mean of 0 and a variance of 0.01 to initialize the weight w ; the bias is initialized to 0, and the probability $p = 0.5$ is retained by default. The method of parameter λ in the experimental penalty item of the method proposed in this paper is carried out according to the method described in Section 3.2, and experiment $\lambda = 0.2$.

Experimental environment: Win10, Intel Core i7 CPU@3.00, RAM 16GB, GPU NVIDIA GTX 1080Ti. *Experimental data set:* considering that CNN is very suitable for processing matrix data in digital images, and the image data in the network has a wide range of sources and a large



FIGURE 1: Partial samples of (a) MNIST, (b) CIFAR-10, and (c) Chinese herbal medicine data sets.

amount of data, this paper selects 3 image data sets for experiments [30]. It is handwritten digit recognition data set MNIST, CIFAR-10, and Chinese herbal medicine identification data set. The first two data sets are currently commonly used benchmark data sets for testing the performance of CNN in the field of computer vision and deep learning. Because these two data sets exist in many websites, they have been tested by overfitting, identification, and classification to verify the effectiveness of the scheme, which is widely representative. The third data set is based on the massive Chinese herbal medicine data set freely collected by network communication technology. Some samples of these 3 data sets are shown in Figure 1. Among them, the MNIST data set contains 60,000 training samples and 10,000 test samples. Each sample is a $28 * 28$ single-channel grayscale image, containing 0-9 handwritten digits, which are normalized to $[0, 1]$ before entering the CNN network. CIFAR-10 is a data set containing 60,000 natural images in 10 categories, including 50,000 training samples and 10,000 test samples. Each sample is a $32 * 32 * 3$ RGB image, which is also normalized before input to the interval $[0,1]$. A total of 108000 images

of Chinese herbal medicine data set were collected from the internet, including five categories of lily, *Codonopsis pilosula*, wolfberry, *Sophora japonica*, and honeysuckle, and these images were processed into the same size ($224 * 224$) images and then normalized to the $[0,1]$. At the same time, in order to verify the performance and effect of the method proposed in this paper, the CNN method is based on the improvement of the immune system (referred to as ICNN) proposed in literature [16], the dropout method proposed by Hinton et al. in literature [21] (referred to as H_Dropout), and the random pooling method (abbreviated as SP) proposed in [33] and the method in this paper for comparison experiments, in which ICNN proposed an improved CNN network method based on the immune system for problems such as overfitting, although this method is tested in the data set. The accuracy is not high (only 81.6%), but its performance is stable, and the theoretical basis is sufficient; Hinton et al.'s H_Dropout method is a dropout method that combines a fully connected layer. They are the first scholars to use dropout to avoid CNN overfitting, and the method is mature. It is easy to implement, while the random pooling

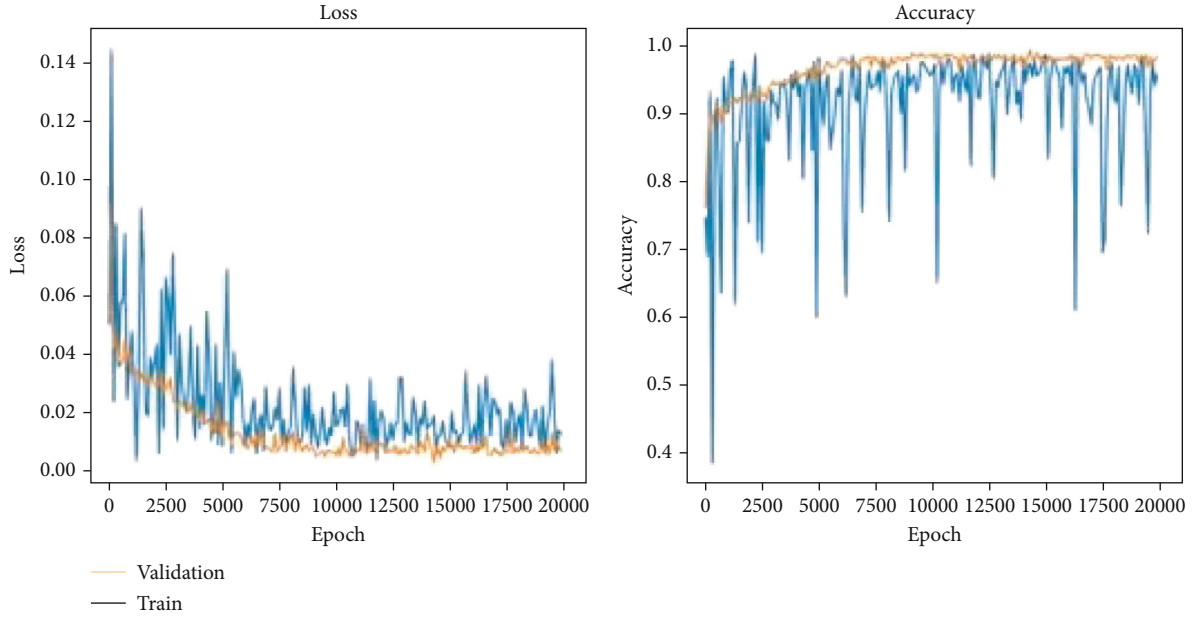


FIGURE 2: MNIST data set: the relationship between the number of iterations and the loss value and accuracy rate.

TABLE 1: Comparison of error rate (%) and retention probability of 4 methods in MNIST data set.

Retention probability	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ICNN	6.36	6.36	6.36	6.36	6.36	6.36	6.36	6.36	6.36
H_Dropout	7.27	6.85	6.50	6.32	6.30	6.31	6.38	6.40	6.51
SP	2.23	2.17	2.10	2.08	2.13	2.26	2.31	2.37	2.49
MDWS	2.21	2.15	2.03	1.95	1.63	1.88	2.19	2.26	2.27

method proposed in [29] randomly selects the pooling activation value in training and uses the probability of each unit in the pooling area as the model average of the weighted probability during the test, which is a kind of efficiency. It is a high method to avoid overfitting. Therefore, the above three methods are feasible for comparison experiments. For convenience, the maximum pooling dropout and weight decay method proposed in this paper are referred to as MDWS (Maxpooling Dropout and Weight Scaled) for short. The experiment verifies the effectiveness of the method proposed in this paper by comparing the relationship between the loss function of the training set and the verification set, the iterative curve of the correct rate, and the error rate and retention probability in the test set.

4.2. Experiment 1: MNIST Data Set. In the experiment, the CNN model used in the method proposed in this paper is as follows: $1 * 28 * 28 \rightarrow 6C5 \rightarrow 2P2 \rightarrow 12C5 \rightarrow 2P2 \rightarrow 1000N \rightarrow 10N$: $1 * 28 * 28$ means the input is $28 * 28$ 1 channel image, C means Convolution, 6C5 means that the convolution kernel is $5 * 5$, and it contains a convolution layer with 6 convolution kernels (6 channels). P means pooling. The first “2” in the pooling layer 2P2 means the pooling

step size. The “2” at the back means that the pooling core size is $2 * 2$, and 1000N means that the fully connected layer contains 1000 neurons. After calculation, the first fully connected layer should be 1152 neurons. The CNN network structure of the other counterparts (ICNN/H_Dropout/SP) is given in the relevant literature and will not be described in detail here. Figure 2 shows the relationship between the loss function value, the correct rate, and the iteration rounds of the method proposed in this article when the iteration round epoch = 20000. It can be seen from the figure that as the number of iterations increases, the loss of the training set, the value of the function keeps decreasing, and the accuracy rate keeps increasing. When the number of iterations exceeds 10,000 times, the value change tends to be stable. Similarly, the loss function value of the verification set also decreases with the increase of iteration rounds, and the accuracy rate continues to increase, and it stabilizes after 10,000 iterations, indicating that the method proposed in this paper can avoid the occurrence of overfitting.

Table 1 is a comparison of the average error rates of the four methods tested in their respective network models under different retention probabilities. Since the ICNN method has nothing to do with the retention probability, the ICNN method is only used to compare the error rate of the test without considering the impact of the retention probability. Analyzing the data in Table 1, the three methods (H_Dropout/SP/MDWS) have the lowest error rate when the retention probability is about 0.5, and the error rate of the H_Dropout method changes most drastically with the value of p . In addition, looking at the four methods under the same p value, the MDWS method proposed in this paper has the smallest error rate, such as 1.63% when $p = 0.5$, SP method has a minimum error rate of 2.08% when $p = 0.4$, and H_Dropout method when $p = 0.5$. The error rate is 6.3%, which shows that the method in this paper can better reduce overfitting and have better pan-China.

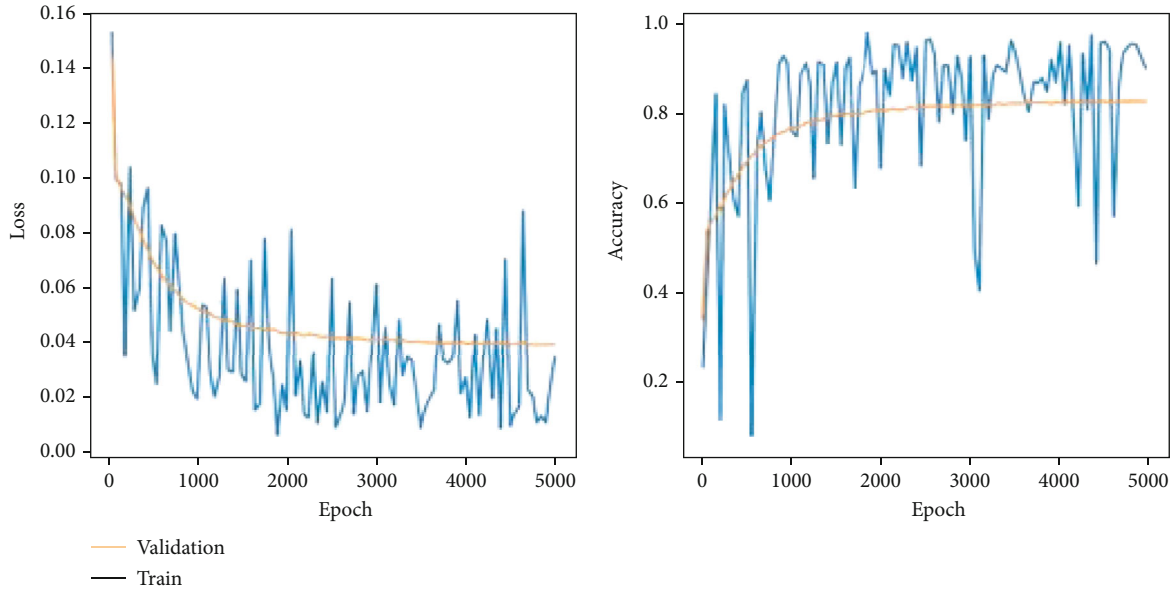


FIGURE 3: CIFAR-10 data set: the relationship between the number of iterations and the loss value and the accuracy rate.

TABLE 2: The relationship between the error rate and retention probability of the CIFAR-10 data set.

Retention probability	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ICNN	7.44	7.44	7.44	7.44	7.44	7.44	7.44	7.44	7.44
H_Dropout	8.87	7.95	7.76	7.12	6.15	7.21	7.38	8.12	8.69
SP	4.83	4.67	4.55	4.31	4.09	4.02	4.16	4.47	5.39
MDWS	3.21	3.08	2.97	2.93	2.86	2.88	3.08	3.16	3.35

4.3. Experiment 2: CIFAR-10 Data Set. This data set is a natural color image data set containing 3 channels. Compared with MNIST, the difference in each category is greater. For this reason, we designed the following CNN network structure: $3 \times 32 \times 32 \rightarrow 32C5 \rightarrow 3P2 \rightarrow 128C3 \rightarrow 3P2 \rightarrow 2000N \rightarrow 10N$; the interpretation of the model is the same as the MNIST data set. The other three methods are also carried out in accordance with the relevant requirements of the literature. The training is iterated for 5000 rounds in total, and the relationship between the loss function, the correct rate, and the iteration rounds of the training set and the validation set is shown in Figure 3. Analyzing Figure 3, it can be seen that as the number of iterations increases, the loss value of both the test set and the verification set continues to decrease, and the change gradually stabilizes. Similarly, as the accuracy value increases with the epoch, the training and verification curves are gradually rising and tending to be flat. It shows that the method proposed in this paper can effectively avoid overfitting when training on the CIFAR-10 data set.

Same as the experiment in Section 4.2, use the above 4 methods (ICNN, H_Dropout, SP, and MDWS in this article) to test in the respective trained networks with the test set and take different retention probability p values to obtain 4 methods under different retention probabilities. The error

rate is shown in Table 2, where ICNN has nothing to do with the retention probability, which has been explained in Section 4.2. Analyzing the data in Table 2, it can be seen that the three methods have a lower classification error rate between the retention probability of 0.4 and 0.6. Among them, the H_Dropout method has the lowest error rate of 7.12% when $p = 0.4$, the SP method has the lowest error rate of 4.02% when $p = 0.6$, and the method in this paper has the lowest error rate of 2.86% when $p = 0.5$. In addition, from the entire table under the same retention probability of the four methods, the MDWS method proposed in this article has the lowest error rate (although the ICNN method does not retain the concept of probability, its classification error rate is higher than the method proposed in this article). In addition, from the entire table under the same retention probability of the four methods, the MDWS method proposed in this article has the lowest error rate (although the ICNN method does not retain the concept of probability, its classification error rate is higher than the method proposed in this article). This shows that the method proposed in this paper also has good generalization in the CIFAR-10 data set. The method is also available in the CIFAR-10 data set and has achieved good generalization.

4.4. Experiment 3: Chinese Herbal Medicine Identification Data Set. The data set is a large-scale data set freely collected from the internet by using modern network technology, and the basic introduction has been stated before. There are many kinds of Chinese herbal medicine, even the same kind of Chinese herbal medicine will appear different forms due to the influence of growth environment and other factors, and some different kinds of Chinese herbal medicine are very similar. Compared with the previous two data sets, the image difference of each category in Chinese herbal medicine data set is small, so the classification processing is more difficult. Since each image in the

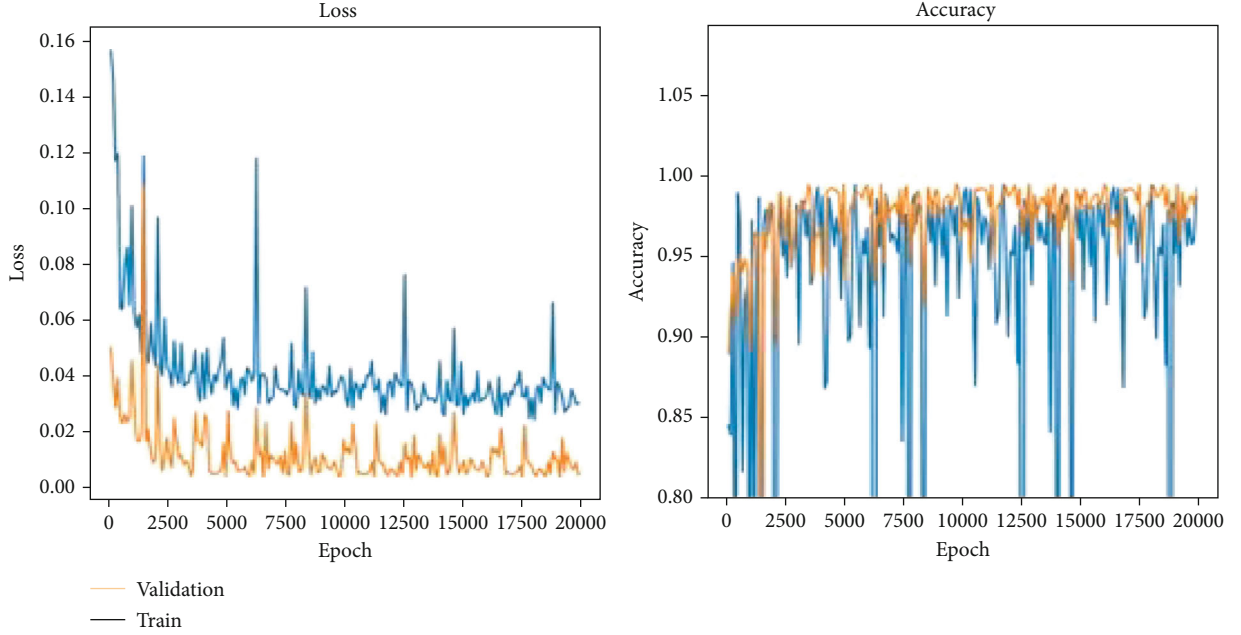


FIGURE 4: Chinese herbal medicine data set: the relationship between the number of iterations and the loss value and the accuracy rate.

TABLE 3: The relationship between the error rate and retention probability of the Chinese herbal medicine data set.

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ICNN	7.44	7.44	7.44	7.44	7.44	7.44	7.44	7.44	7.44
H_Dropout	8.87	7.95	7.76	7.12	6.15	7.21	7.38	8.12	8.69
SP	4.83	4.67	4.55	4.31	4.09	4.04	4.16	4.47	5.39
MDWS	3.21	3.08	2.97	2.93	2.76	2.88	3.08	3.16	3.35

data set is also composed of 3-channel natural color images, all the images are processed into 224×224 size as input, so we designed the following CNN network structure: $3 \times 224 \times 224 \rightarrow 64C3 \rightarrow 1P2 \rightarrow 128C3 \rightarrow 1P2 \rightarrow 256C3 \rightarrow 1P2 \rightarrow 512C3 \rightarrow 4096N \rightarrow 5N$. The other three methods were also carried out according to the relevant requirements of the literature. The total number of training iterations is 20000, and the loss function of training set and validation set and the relationship between accuracy and iteration epochs are obtained, as shown in Figure 4. It can be seen from the analysis of Figure 4 that with the increase of iteration rounds, the loss value of both test set and validation set is decreasing, and the change gradually tends to be stable after 3000 epochs. Similarly, with the increase of epoch, the training and validation curve tends to be flat at 3000 epochs. The results show that the proposed method can effectively avoid overfitting when training on Chinese herbal medicine data set.

As in the previous experiment, the above four methods (ICNN, H_Dropout, SP, and MDWS) are tested in the trained network with the test set, and different retention probabilities are taken to obtain the error rates of the 4 methods under different retention probabilities, as shown

in Table 3. ICNN has nothing to do with the retention probability, which has been explained previously. By analyzing the data in Table 3, it can be seen that the three methods have a lower classification error rate between the retention probability of 0.5~0.7. Among them, H_Dropout method has the lowest error rate of 6.15% when $p = 0.5$, SP method has the lowest error rate of 4.04% when $p = 0.6$, and the method proposed in this paper has the lowest error rate of 2.76% when $p = 0.5$. Comparing the minimum error rate of the three methods, the MDWS method proposed in this paper has the minimum error rate, and it shows that the method proposed in this paper also has good generalization in Chinese herbal medicine data set.

5. Conclusion

With the development of computer network technology, in the network communication based on data-driven, the amount of data in cyberspace is increasing. The existence of massive duplicate data brings great problems to network communication and security. Therefore, how to correctly identify the same or similar data in the network is the basis to solve this problem. The development of deep learning technology provides a new solution to this problem. But the overfitting problem in the deep learning network model is a common problem encountered in model training. In the design of the CNN model, this article designs a maximum pooling dropout method in the pooling layer and introduces weights in the backpropagation. The weight attenuation mechanism is used to reduce the complexity of the deep learning model, so as to avoid the overfitting of the deep learning model in training and improve the robustness of network data communication. There are two main innovations in this article. One

is the design of the maximum pooling dropout, which uses the unit value sorting of the pooled area to design the unit (neuron) discarding method. The second is to introduce a penalty term in the backpropagation to design the weight attenuation. The implementation process: theoretical analysis and experimental comparison verify that the method in this paper can effectively avoid overfitting and improve the generalization performance of the network. However, the method proposed in this paper still has the following problems. First, when the maximum pooling dropout is used, the semilinear activation function (ReLU) of the front convolutional layer may cause nonmaximum output unit value 0, so that the neuron corresponding to the unit will not be updated in the future (the gradient is 0), that is, dead nerve; the second is that the introduced penalty term parameter λ and the maximum pooled dropout probability p should theoretically have a certain connection, but in this article, the two are not connected for research, and the solution of these problems will be the next research goal.

Data Availability

The two data sets used in this experiment are MNIST and CIFAR-10, which are classic open source data sets for deep learning network testing, and can be downloaded from related websites. The two data sets used in this experiment are downloaded from Baidu paddle platform: <https://www.paddlepaddle.org.cn/>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by the National Key R&D Program of China under grant number 2019YFE0122600, in part by the key scientific research projects of Hunan Provincial Education Department under grant number 20A162, and in part by the Natural Science Foundation of Hunan Province under grant numbers 2021JJ50050, 2020JJ7007 and 2020JJ6088.

References

- [1] X. Chen, C. Li, D. Wang et al., "Android HIV: a study of repackaging malware for evading machine-learning detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 987–1001, 2020.
- [2] G. Lin, S. Wen, Q. Han, J. Zhang, and Y. Xiang, "Software vulnerability detection using deep neural networks: a survey," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1825–1848, 2020.
- [3] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, "A blockchain-enabled deduplicatable data auditing mechanism for network storage services," *IEEE Transactions on Emerging Topics in Computing*, 2020.
- [4] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, "Deep learning enhanced multi-target detection for end-edge-cloud surveillance in smart IoT," *IEEE Internet of Things Journal*, 2021.
- [5] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 289–300, 2020.
- [6] L. Qi, C. Hu, X. Zhang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4159–4167, 2021.
- [7] Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, "An efficient privacy-enhanced attribute-based access control mechanism," *Concurrency & Computation Practice & Experience*, vol. 32, no. 5, pp. 1–10, 2020.
- [8] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, "Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5790–5798, 2021.
- [9] Y. Xu, X. Yan, Y. Wu, Y. Hu, W. Liang, and J. Zhang, "Hierarchical bidirectional RNN for safety-enhanced 5G heterogeneous networks," *IEEE Transactions on Network Science and Engineering*, 2021.
- [10] X. Yang, X. Li, Y. Guan, J. Song, and R. Wang, "Overfitting reduction of pose estimation for deep learning visual odometry," *China Communications*, vol. 17, no. 6, pp. 196–210, 2020.
- [11] K. H. Cha, N. Petrick, A. Pezeshk, C. G. Graff, and B. Sahiner, "Reducing overfitting of a deep learning breast mass detection algorithm in mammography using synthetic images," in *Progress in Biomedical Optics and Imaging-Proceedings of SPIE*, pp. 188–194, San Diego, CA, USA, March 2019.
- [12] G. González, S. Y. Ash, R. San José Estépar, and G. R. Washko, "Reply to Mummadiet al.: overfitting and use of mismatched cohorts in deep learning models: preventable design limitations," *American Journal of Respiratory and Critical Care Medicine*, vol. 198, no. 4, pp. 545–555, 2018.
- [13] A. Ashiquzzaman, A. K. Tushar, M. R. Islam et al., "Reduction of overfitting in diabetes prediction using deep learning neural network," in *IT Convergence and Security 2017. Lecture Notes in Electrical Engineering*, vol. 449, pp. 35–43, Springer, Singapore, 2018.
- [14] S. Zhang, Y. Gong, and J. Wang, "The development of deep convolutional neural networks and their applications in the field of computer vision," *Chinese Journal of Computers*, vol. 42, no. 3, pp. 453–482, 2019.
- [15] Y. Xu, C. Zhang, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, "Blockchain-enabled accountability mechanism against information leakage in vertical industry services," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1202–1213, 2021.
- [16] T. Gong, T. Fan, J. Guo, and Z. Cai, "GPU-based parallel optimization of immune convolutional neural network and embedded system," *Engineering Applications of Artificial Intelligence*, vol. 62, no. 25, pp. 384–395, 2017.
- [17] D. Liu and J. Liu, "Neural network model for deep learning overfitting problem," *Journal of Natural Science of Xiangtan University*, vol. 40, no. 2, pp. 96–99, 2018.
- [18] J. Cheng, G. Zeng, D. Lu, and B. Huang, "Dropout-based improved convolutional neural network model averaging method," *Journal of Computer Applications*, vol. 39, no. 6, pp. 1601–1606, 2019.
- [19] J. Li, G. Qin, X. Wen, and F. Hu, "Over-fitting in neural network learning algorithms and its solving strategies," *Journal*

- of Vibration, Measurement & Diagnosis*, vol. 22, no. 4, pp. 260–264+320, 2002.
- [20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detector,” *Computer Science*, vol. 3, no. 4, pp. 212–223, 2012.
- [22] D. Chen, Z. Yan, and H. Liu, “Overfitting phenomenon of SVD series algorithms in scoring prediction,” *Journal of Shandong University (Engineering Science Edition)*, vol. 44, no. 3, pp. 15–21, 2014.
- [23] W. Shen, Y. Li, Z. Yang, X. Wang, and X. Ye, “Attribute reduction to prevent overfitting,” *Application Research of Computers*, vol. 37, no. 9, pp. 2665–2668, 2020.
- [24] G. Qin and Z. Li, “Over-fitting of BP NN research and its application problem,” *Engineering Journal of Wuhan University*, vol. 39, no. 6, pp. 55–58, 2006.
- [25] Z. Yang, F. Gao, S. Fu, M. Tang, and D. Liu, “Overfitting effect of artificial neural network based nonlinear equalizer: from mathematical origin to transmission evolution,” *Science China Information Sciences*, vol. 63, no. 6, pp. 82–97, 2020.
- [26] C. Zhang, O. Vinyals, R. Munos, and S. Bengio, “A study on overfitting in deep reinforcement learning,” *Statistics*, no. 2, pp. 1–25, 2018.
- [27] J. Wang, Z. Wang, and H. Wang, “Improved CNN algorithm based on PSO algorithm and dropout,” *Journal of Changchun University of Technology*, vol. 40, no. 1, pp. 26–30, 2019.
- [28] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, “A blockchain-based multi-cloud storage data auditing scheme to locate faults,” *IEEE Transactions on Cloud Computing*, 2021.
- [29] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo, and T. Guo, “Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6182–6192, 2020.
- [30] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, “Variational LSTM enhanced anomaly detection for industrial big data,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3469–3477, 2021.
- [31] X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo, and C. Li, “Learning URL embedding for malicious website detection,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6673–6681, 2020.
- [32] D. J. C. Mac Kay, “Bayesian interpolation,” *Maximum Entropy and Bayesian Methods*, vol. 50, no. 3, pp. 39–66, 1992.
- [33] M. D. Zeiler and R. Fergus, “Stochastic pooling for regularization of deep convolutional neural networks,” 2013, <http://arxiv.org/abs/1301.3557>.