A Survey on Pruning Algorithm Based on Optimized Depth Neural Network

Qidi Song[1*], Xuanze Xia[2]

[1] International Education College, Changchun University of Technology, Changchun, China.
[2] SHU-UTS SILC Business School, Shanghai University, Shanghai, China.

* Corresponding author. Email: 20181033@stu.ccut.edu.cn, simonxxz1010@163.com
Manuscript submitted January 1, 2022; accepted March 8, 2022.
doi: 10.17706/ijcce.2022.11.2.10-23

Abstract: In recent years, deep neural network has continuously renewed its best performance in tasks such as computer vision and natural language processing, and has become the most concerned research direction. Although the performance of deep network model is remarkable, it is still difficult to deploy to the embedded or mobile devices with a limited hardware due to the large number of parameters, high storage and computing costs. It has been found by relevant studies that the depth model based on convolutional neural network has parameter redundancy, and there are parameters that are useless to the final result in the model, which provides theoretical support for the pruning of depth network model. Therefore, how to reduce the model size under the condition of ensuring the model accuracy has become a hot issue. This paper classifies and summarizes the achievements of domestic and foreign scholars in model pruning in recent years, selects several new pruning algorithm methods in different directions, analyzes their functionality through experiments and discusses the current problems of different models and the development direction of pruning model optimization in the future.

Key words: Artificial intelligence, deep learning, deep neural network, network pruning, optimization.

1. Introduction

Deep learning has become one of the most important parts of machine learning. The main task of deep learning is to build a "deep neural network" (DNN) and input a large number of sample data, and finally get a model with strong analysis ability and recognition. After decades of development, deep convolutional neural networks have achieved remarkable performance in many applications, especially in computer vision tasks [1]-[5]. Deep Neural Networks (DNNs) have shown extraordinary abilities in complicated applications such as image classification, object detection, voice synthesis, and semantic segmentation [6]. In recent years, with the rapid development of GPU computing power, the network scale of neural network is becoming increasingly large, and the data processing ability is improving. Many deep neural network models and convolutional neural network models have appeared, such as VGGNet, GoogleNet [7], AlexNet [8]. The development of artificial intelligence makes computers surpass human accuracy in many tasks. However, when recognizing and processing data, deep neural network exposes its disadvantage computation. This directly leads to the problems of high storage and high energy consumption in practical applications, (such as deeprior and deepprior + +). It also largely limits the productization of deep learning methods, especially on some edge devices (or embedded devices), such as our mobile phones. Edge devices are not specially designed for computing intensive tasks. If mobile phones are directly used for intensive

computing, power consumption and delay will become serious problems. Even on the server side, large-scale computing will directly lead to the increase of time cost.

Therefore, it is necessary to simplify the model so as to reduce the amount of calculation and storage. Model compression is regarded as a possible solution. Model compression methods include pruning and quantization. Network pruning in the early stage, refers to removing redundant parameters or neurons that do not significantly contribute to the accuracy of results [9]. Network pruning involves removing parameters that don't impact network accuracy [10], so that the neural network can match faster, speed up the calculation speed of the model and compress the storage space of the model. Network pruning involves removing parameters that don't impact network accuracy [11]. Based on the tailoring criterion of Taylor expansion, the tailoring convolutional neural network has superior performance in fine-grained classification tasks. Low rank decomposition of convolution kernel parameters can increase sparsity and reduce operation consumption [12]. The network pruning also adopts a certain measurement standard in removing part of the weights in the network or the connection between part of the weights.

Ordinary pruning only takes the absolute value of weight as the only evaluation standard of pruning, ignoring the influence of other factors. Although the model can be compressed effectively, the generated sparse model depends on a dedicated database and hardware. At present, with the increasing function and size of modern neural network, its calculation and storage requirements are also improved accordingly, and increasingly more complex pruning methods are emerging in endlessly. Based on convolution kernel dynamics, weight correlation and weak layer penalty, this paper analyzes the functional strength of various optimized pruning methods, and puts forward the prediction and suggestions on the future development trend of pruning.

2. Classification Algorithms

2.1. Combined Dynamic Pruning

Combined Dynamic Pruning (CDP) is divided into two parts: Kernel Dynamic Compression and Channel Dynamic Compression. Although they complement each other instead of separating each other to complete network pruning. According to the characteristics of the convolution kernel, the dynamic pruning algorithm of the convolution kernel is designed to permanently cut off part of the convolution kernel in order to improve the compression ratio. The pruning standard is L1 norm, however, these convolution kernels are not directly deleted from the network. Instead, the corresponding channel is zeroed to continue training and learning, allowing the zeroed convolution kernels to be updated in back propagation until convergence. In view of the characteristics of the input image, an algorithm of channel dynamic compression is designed. After the input image is sampled, the importance of the channel is predicted through linear change. The result of the corresponding channel is zeroed, which is equivalent to skipping part of the convolution operation without changing the network structure. Fig. 3 is taken as an example to illustrate the process of the joint dynamic pruning algorithm. The dynamic pruning ratio of convolution kernel is defined as β and the dynamic compression ratio of channel is defined as α. In order to balance accuracy and model complexity, $2\beta = \alpha + 1$ is defined, the complete network compression score is divided into two steps. Firstly, the network is compressed to β using the dynamic pruning algorithm of convolution kernel. Secondly, the network is compressed to α using the dynamic compression algorithm of channel. Thirdly, all channels that need to be zeroed are obtained.

2.1.1. Kernel dynamic compression

The convolution kernel dynamic pruning algorithm is designed to permanently remove some of the less important convolution kernels from the model while maximizing the capacity of the model. Fig. 1. shows the structure of combined dynamic pruning [13]. For each convolution kernel of the same convolution layer, a

specific standard *SJ* is designed to measure its importance. This paper chose the L1 norm, namely convolution kernels tensor in the sum of the absolute value of the weights of $\sum | Fi, j |$. From an intuitive perspective, the convolution kernel with a smaller weight also has a smaller value of the corresponding output feature graph, which is less influential than other feature graphs at the same layer. The experimental results also show that the effect of convolution kernels with smaller absolute value of pruning is better than that of random pruning and convolution kernels with larger absolute value of pruning. Convolution of the importance of nuclear standards also have different choices, such as the L2 norm, namely convolution kernels tensor in the square sum of the weights to open square $\sum | Fi, j | 2$. However, the core of dynamic judgment of this method lies in the process of iteration and update, rather than the complex measurement standards in a single iteration, and the complex measurement standards did not bring improvement in performance, so L1 norm was still selected as the evaluation standard in the end.

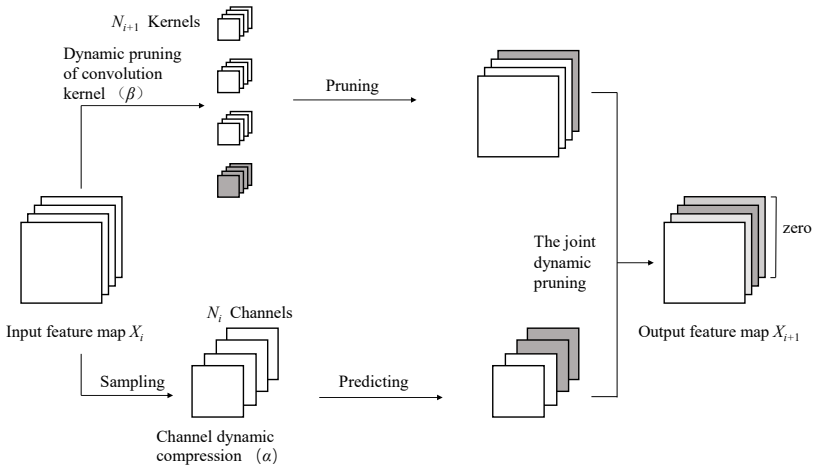

Fig. 1. Framework of combined dynamic pruning algorithm [13].

Algorithm 1: Kernel dynamic compression

Input ： Training data $\boldsymbol{X}$, convolution kernel dynamic pruning rate β, model $\boldsymbol{W}^{(i)} \in \mathbf{R}^{Ni+1 \times Ni \times K \times K}$, $1 \le i \le L$;

Output ： Model $\boldsymbol{W}^*$ after dynamic pruning of the convolution kernel.

Initialize model parameters

for $epoch = 1$; $epoch \le epoch_{max}$; $epoch++$ do

Update the parameters of model $\boldsymbol{W}$ based on input $\boldsymbol{X}$

for $i = 1$; $i \le L$; $i++$ do

Calculate the L1 norm of each convolution kernel $\boldsymbol{F}_{i,j}$:

$s_j = \sum | \boldsymbol{F}_{i,j} |$, $1 \le j \le N_{i+1}$

The convolution kernels are sorted according to s_j, and zero $N_{i+1}(1 - \beta)$ smaller convolution kernels are collated

end for

end for

Permanently cut off the zero convolution kernel

2.1.2. Channel dynamic compression

Channel dynamic compression speeds up the process of training and reasoning by dynamically selecting part of the model to participate in operations without permanently removing any parameters from the model. The specific method is to build a small linear change neural network, called predictive network,

which is used to establish the relationship between the input feature graph and the convolution kernel. It also can predict and select some channels to participate in the convolution operation, to maintain the capacity of the model. The first step is to extract the features of the input image. If the input image of the classification network is completely used as the input of the prediction network, the complexity of the whole model will be greatly increased, and even offset the benefits brought by pruning. Therefore, the input image must be compressed to an acceptable size as much as possible, while preserving its features for predictive network input. Here the down-sampling method is chosen, that is, reducing the size of the image. Through experiments, it can be found that the effect of Global Average Pooling (GAP) is the best. GAP is a simple and practical regularization method, that is, to take the average value of all elements in two-dimensional images. In recent years, there are also studies that it has a good ability to extract image features. Apply it to the three-dimensional feature graph, and there is the following sampling function:

$$ss(X_i) = \frac{1}{H_i W_i} \prod_{j=1}^{N_{i+1}} s(X_i^{[j]}) \tag{1}$$

Among them, $s(X_i^{[j]})$is the global average pooling operation, which compresses the J channel of the input feature graph into a single element, which is equivalent to reducing the $H_i * W_i$ dimensions and reducing the complexity of the input feature graph. Then, the relationship between the input image sampled in the following 2 and the convolution kernel needs to be established. In contrast to the dynamic pruning of the convolution kernel, this step does not need to explicitly consider the parameters of the convolution kernel itself. Instead, it builds a linearly varying neural network, called the predictive network, which is used to predict the relationship between the input image and the convolution kernel. The values of the prediction network are appended to the output of the classification network and are automatically updated as the back propagation occurs. The function of the linear change is as follows:

$$g_i(X_i) = (ss(X_i)\phi_i + \rho_i)_+ \tag{2}$$

The linear change prediction network, which is an additional full connection layer, has two learnable parameters, the weight ϕ_i and the bias ρ_i. The input of the prediction network is $ss(X_i)$, that is, the vector of Ni elements, and each element represents the feature of a channel of the input image of the current convolution layer. After linear variation, its output is a vector of N_{i+1} elements, and each element represents the importance of each convolution kernel in the current convolution layer for the input image. The larger the element is, the more the channel is activated by the input image and the more important it is. After obtaining the importance of each convolution kernel, the output of a part of channels should be eliminated from the original complete output feature graph based on the channel dynamic compression ratio α. Based on a simple *K-winner-take-all* operation, which selects the *k* largest elements of the vector and sets the rest to zero. In this way, the less important $N_{i+1}(1 - \alpha)$ channels will not participate in the convolution operation, thus completing the pruning operation. Since the prediction network needs to be updated with the classification network, the results of the prediction network also need to be added to the output of the classification network. Based on the previous complete definition of convolution layer including BN and ReLU operations, replace the γ parameter of BN layer with the output of the prediction network, namely $g_i(X_i)$:

$$\hat{f}_i(X_i) = (g_i(X_i) \cdot norm(conv_i(X_i, W^{(i)})) + \beta_i) \tag{3}$$

Algorithm 2: Channel dynamic compression

Input : Training data X, channel dynamic compression ratioα, model $W^{(i)} \in R^{Ni+1 \times Ni \times K \times K}$, $1 \le i \le L$;

Output : Dynamic compressed model of channels W^*.

Initialize model parameters

for $epoch = 1$; $epoch \le epoch_{max}$; $epoch++$ do

for $i = 1$; $i \le L$; $i++$ do

Calculate the undersampling $ss(X_i)$ of the input feature graph Xi

Calculation of channel importance Function $g_i(X_i)$ using predictive network

The smaller elements of $N_{i+1}(1 - \alpha)$ in $g_i(X_i)$ are set to zero

Replace gamma of BN layer in classification network with $g_i(X_i)$

end for

end for

2.2. Pruning Method of Convolutional Neural Network Model with Weight Dependence

Magnitude based pruning has been proposed and is widely accepted that trained weights with large values are more important than trained weights with smaller values [11]. The model pruning method of convolutional neural network based on weight relevance can be summarized include three steps. Firstly, the importance of all filters of the model is calculated by the proposed algorithm. Secondly, the importance was sorted, and the filter weights with relatively low importance are removed by specifying the pruning ratio of the model. Thirdly, it can fine-tune the pruned model to restore the precision of the model.

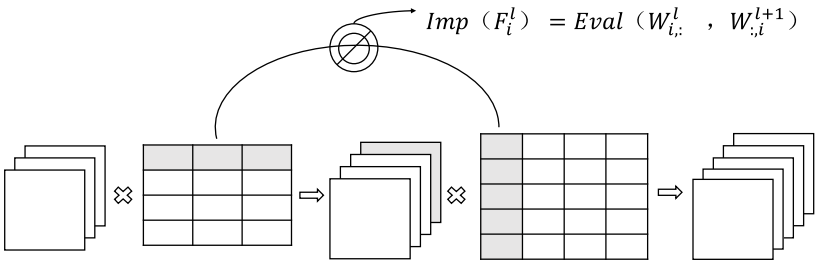

Fig. 2. Evaluation based on weight dependency for a filter [14].

Filter level model pruning is to reduce the number of parameters and computation of the model by cutting some filters of low importance. After a part of filters are cut, the associated weights of the output feature graph of the filter at the next layer will also be cut, which shows the relevance of weights. In this paper, both the weight of the filter itself and the associated weight of the filter are considered to be evaluated, as shown in Fig. 2. The importance of defining a filter is shown in Formula (4):

$$Im\,p\,(F_i^l) = Eval(W_{i,:}^l, W_{:,i}^{l+1}) \tag{4}$$

Filter level model pruning is to reduce the number of parameters and computation of the model by cutting some filters of low importance. After a part of filters are cut, the associated weights of the output feature graph of the filter at the next layer will also be cut, which shows the relevance of weights. In this paper, both the weight of the filter itself and the associated weight of the filter are considered to be evaluated, as shown in Fig. 2. The importance of defining a filter is shown in Formula (5):

$$Im\,p\,(F_i^t) = Eval(W_{i,:}^l, W_{:,i}^{l+1}) \tag{5}$$

where, $F_i^l$is the i_{th} filter of the lth layer in the model, X^l is the input of the lth layer, $\text{Im}\,\text{p}\,(F_i^l)$ is the importance of F_i^l, and $W_{i,:}^l$ is the weight vector of the filter F_i^l, $W_{:,i}^{l+1}$ is the associated weight vector of the filter F_i^l, and $\text{Eval}(W_{i,:}^l, W_{:,i}^{l+1})$ is the evaluation value of the weight of the filter and its associated weight. While using the local pruning method, the importance of a filter is defined by L1 norm, as shown in Formula (6):

$$L_i^l = \sum_j |W_{i,j}^l| \tag{6}$$

where L_i^l is the L1 norm value of the i_{th} filter weight in l the model, and $W_{i,j}^l$ is the weight vector of the j_{th} convolution kernel that constitutes the filter. This method ignores the importance of the associated weights of filters, which may result in relatively important associated weights being removed along with the filters. Therefore, this paper evaluates the weight of the filter itself and its associated weight together, as shown in Formula (7):

$$L_i^{l,l+1} = \sum_j |W_{i,j}^l| + \sum_j |W_{i,j}^{l+1}| \tag{7}$$

where $L_i^{l,l+1}$ is the L1 norm value calculated by the filter weight and its associated weight, and $W_{i,j}^{l+1}$ is the convolution kernel vector that constitutes the associated weight. As the calculation result of norm value in "norm value hypothesis" depends on the weight value, however, the weight value distribution is different due to the different features extracted from each layer. Therefore, the evaluation value obtained by this hypothesis can only be compared locally within the layer. In order to achieve global comparison, the evaluation values obtained by Formula (7) are globally standardized in this paper. After analysis and experiment, this paper proposes a "log" standardized method to achieve global comparability. The evaluation value of the filter in this paper is defined as formula (8):

$$reL_i^{l,l+1} = \frac{log(L_i^{l,l+1})}{log(max(L_i^{l,l+1}))} \tag{8}$$

where $reL_i^{l,l+1}$ is the globally standardized evaluation value, $\max(L_i^{l,l+1})$ is the maximum value of the norm value in the first layer of the model, $q, i \in [1, N^l]$, N^l is the number of filters in the l_{th} layer. Finally, pruning and fine-tuning. Obtained from formula (8) the importance of the entire model filter set $M = \{\text{Im}\,p_1, \text{Im}\,p_2, \ldots, \text{Im}\,p_n\}$, according to the preset proportion P model and pruning get the pruning threshold, the filter screen for each layer: among them, the theta is the importance of the entire model filter thresholds, $sort_p(M)$ means to sort M in ascending order, and returns the position $n * P$ value as a threshold, N is the number of filters for the whole model and P is the percentage. At the same time, the algorithm can be selectively extended from the single pruning described above to the iterative pruning method, and the pruning process mentioned above can be repeated to compress the model. After the iterative pruning method is extended, the pruning proportion of each model can be lowered and the compressed model can be pruned repeatedly, making the pruning process smoother and the model more compact.

2.3. Pruning Method of Convolutional Neural Network Model with Weak Layer Penalty

Firstly, the Euclidean distance is used to calculate the information distance between convolutional kernels at each level. Secondly, the data distribution characteristics of the information distance of each

convolutional layer are used to identify the weak layer, and the proposed normalization function based on contribution degree is used to punish the weak layer and eliminate the differences between layers. Thirdly, the global importance of convolutional kernel is evaluated, and dynamic pruning is achieved by global mask technology.

2.3.1. Concepts of algorithms

Assuming that a convolutional neural network has L layer, C_i and C_{i+1} are used to represent the number of input and output channels of the i_{th} convolutional layer respectively, and $F_{i,j}$ represents the j_{th} convolutional kernel of the i_{th} layer. Where the dimension of $F_{i,j}$ is $R^{(C_i*H_i*W_i)}$, and K represents the size of the core. The input characteristic graph S and output characteristic graph of the layer are O and $C_i * H_i * W_i$ $C_i * H_i * W_i$ and $C_{i+1}*H_{i+1}*W_{i+1}$ respectively. The weight W_i of i_{th} layer can be expressed as $\{F_{i,j}, 1 \le j \le C_{i+1}\}$. i_{th} layer convolution operation, therefore, can be expressed as: $\{O = F_{i,j} * S, 1 \le j \le C_{i+1}\}$ convolution neural network can be parameterized shown: $\{W^{(i)} \in R^{C_i*K*K*C_{i+1}}, 1 \le i \le L\}$, also can be written as:$\{W^{(i)} \in R^{C_i*Z}, 1 \le i \le L, Z = K * K * C_{i+1}\}$, $F_{i,j}$ weights for $W_i^j \in R^Z$.

2.3.2. Pruning process

Algorithm 3: Pruning algorithm based on weak layer punishment

Input : Training data **X**, global pruning rate **P**, contribution factor **V**

Output : The compressed model and parameter **W** initializes the model parameter **W** and global mask **M=1**

for epoch=1;epoch＜epoch $_{max}$, epoch++;

Update model parameter **W** with training set **X;**

Calculate **R** value of each kernel by $R(F_{i,j}) = R\sum_{j^* \in [1,C_{i+1}], j^* \ne j} \left\| W_i^j - W_i^{j^*} \right\|_2$;

$Z(F_{i,j^*}) = \frac{R(F_{i,j^*})}{Max(R_i) - Min(R_i) + \theta} \times IMP_i$ is used to calculate the normalized **Z** value;

Find a kernel corresponding to the lowest **Z**-value;

Update **M** by $Avg_{all} = \frac{1}{L}\sum_{i=1}^{L} Std_i$;

The selected convolution kernel weight Max is reset to zero by $Std_{all} = \sqrt{\frac{1}{L}\sum_{i=1}^{L}[Std_i - Avg_{all}]^2}$;

end for

3. Result and Further Analysis

3.1. Training Methods and Evaluation Standards

With reference to some previous related studies, this paper make comparison between three pruning methods including the joint dynamic pruning, pruning based on weight correlation and the pruning that fuses the weak layer penalty. In order to verify the effects of these three pruning algorithms, this paper quotes some data and Figure made by previous researchers which are based on unified standards.

In order to verify the effects of these three pruning algorithms, we list the data of three algorithms based on CIFAR-10 to complete the experiment. CIAFR-10 contains 60,000 32×32 color image data sets, including 50,000 training images and 10,000 test images.

This article compares the compression effect of three algorithms on VGGNet, and M-CifarNet convolutional neural network model. VGG is a deep-level convolutional neural network model proposed by Simonyan [15], the visual geometry group of Oxford University. It achieved outstanding results in the 2014 ImageNet image classification and target detection competition. M-CifarNet is an 8-layer convolutional neural network designed for the CIFAR data set. It only uses 1.3 × 106 parameters to achieve 91.37% and

99.67% of Top-1 and Top on the CIFAR-10 data set. All convolutional layers in M-CifarNet use a 3 × 3 convolution kernel, and pool is a global average pooling layer.

With reference to some previous related studies, this article sets the evaluation index of the algorithm to the accuracy of the pruned and fine-tuned model, as well as the reduction ratio of the parameter amount and the reduction ratio of the calculation amount relative to the original model. The model training in this article is optimized using the stochastic gradient descent (SGD) method, the batch size is 64, and the training is set to 160 epochs. The initial value of the learning rate is 0.1. When the training reaches 50% and 75%, the value of the learning rate will decay to 1/10 of the previous value. The value of the weight decay is set to 10-4, while the momentum coefficient is set to 0.9.

The most intuitive criterion is to compare the accuracy changes brought about by different algorithms under a specific pruning rate of 0.4. The compression ratio of the model can be defined as α, the pruning rate is γ, and $\alpha = 1-\gamma$. However, the implementation of the three algorithms is different, and the complexity reduction brought by the same pruning rate is also different. Therefore, in order to measure the complexity of the model more objectively, this paper further chooses the two standards of Floating-Point Operations Per second (FLOPs) and the compression ratio of the parameter scale, and compares the accuracy of the model in the three algorithms.

3.2. Result Analysis

3.2.1. M-CifarNet experimental results

Based on Table 1, we can see the network structure of M-CifarNet and the comparison of FLOPs operations brought about by different pruning algorithms. First, when the pruning rate is 0.4, the FLOPs compression ratio of WLP is 2.82 (Fang, Z.Y. *et al.*, 2021) [16]. And Zhang, M. M. *et al.* (2021) points out that the FLOPs compression ratio of CDP is 1.68 at the same pruning rate [13]. However, the FLOPs compression ratio of WDP is 2.82 (Yan, Y.C. *et al.*, 2021, pp.5) [14]. This result is based on following reasons. In the previous analysis, as a classic algorithm for dynamic pruning, WDP does not change the network structure, so the number of output channels remains unchanged. However, it only selects part of the convolution kernel for calculation, which is reflected in the calculation of FLOPs. The number of input channels remains unchanged, while the number of output channels decreases. Therefore, the compression ratio of FLOPs is also the smallest. WLP has the largest compression ratio because part of the convolution kernel is permanently removed. Although CDP also permanently removes part of the convolution kernel, its removal ratio is not as large as that of WLP. It achieves a FLOPs compression ratio between the two.

Table 1. Comparison of FLOPs Operations brought about by Different Pruning Algorithms Based on M-CifarNet Network Structure [13]

level	Image Input	FLOPs			
		Original network	WLP(γ=40%)	WDP(γ=40%)	CDP(γ=40%)
Conv0	30*30	$3.2*10^6$	$1.9*10^6$	$1.9*10^6$	$1.9*10^6$
Conv1	30*30	$6.6*10^7$	$2.4*10^7$	$3.9*10^7$	$3.1*10^7$
Conv2	15*15	$3.3*10^7$	1.2*107	2.0*107	$1.6*10^7$
Conv3	15*15	$6.6*10^7$	$2.3*10^7$	$3.9*10^7$	$1.8*10^7$
Conv4	15*15	$6.6*10^7$	$2.3*10^7$	$3.9*10^7$	$3.1*10^7$
Conv5	8*8	$2.8*10^7$	$1.0*10^7$	$1.7*10^7$	$3.1*10^7$
Conv6	8*8	$4.2*10^7$	$1.5*10^7$	$2.5*10^7$	$1.3*10^7$
Conv7	8*8	$4.2*10^7$	$1.5*10^7$	$2.5*10^7$	$2.0*10^7$
Pool	8*8				
Fc	1*1	$3.8*10^3$	$1.2*10^3$	$2.1*10^3$	$1.7*10^3$
Total		$3.5*10^8$	$1.2*10^8$	$2.1*10^8$	$1.7*10^8$
FLOPs compression ratio		1.00	2.82	1.68	2.11

From the perspective of parameter scale, since WDP and CDP introduce additional predictive networks, there are additional parameters compared to the original network structure. However, compared with the parameters of the convolutional layer, the parameters of the fully connected layer account for only a small part, so it can be ignored. The focus is on the pruning algorithm to remove the number of parameters of the original model. The network compression ratio α is decreased from 1 to 0.2, and the result is shown in Fig. 3. Since both WDP and CDP introduce an additional fully connected layer for channel importance prediction, when the compression ratio is 1, the accuracy of both exceeds the accuracy of the original model. In general, the accuracy of CDP is significantly higher than the previous two algorithms. It is more obvious when the compression ratio α ≤ 0.6, and the accuracy is higher about 1%.

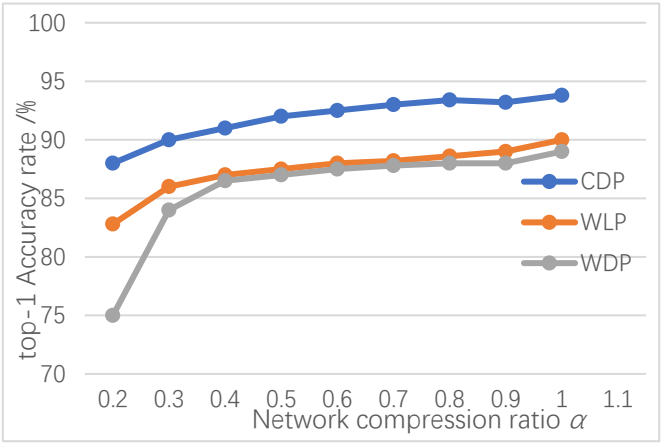

Fig. 3. Accuracy changes brought by different pruning algorithms (M-CifarNet).

3.2.2. VGG experiment results

Table 2 shows the network structure of VGG16 and the comparison of changes in FLOPs operations brought about by different pruning algorithms. When the pruning rate is 0.4, the FLOPs compression ratio of WLP is 2.57 (Fang, Z.Y. *et al.*, 2021) [16]. And Zhang, M.M *et al.* (2021) points out that the FLOPs compression ratio of CDP is 1.99 at the same pruning rate [13]. However, the FLOPs compression ratio of WDP is 1.62 (Yan, Y.C. *et al.*, 2021, pp.5) [14]. The WLP algorithm permanently removes the most parameters, thus obtaining the highest FLOPs compression ratio. The CDP algorithm proposed in this paper achieves an intermediate FLOPs compression ratio.

Table 2. Comparison of FLOPs Operations brought about by Different Pruning Algorithms Based on VGG16 Network Structure [13]

level	Image input	FLOPs			
		Original network	WLP(γ=40%)	WDP(γ=40%)	CDP(γ=40%)
Conv0	32*32	$3.7*10^3$	$2.2*10^3$	$2.2*10^3$	$2.2*10^3$
Conv1	32*32	$7.6*10^3$	$2.7*10^7$	$4.5*10^7$	$3.6*10^7$
Pool1					
Conv2	16*16	$3.8*10^7$	$1.3*10^7$	$2.2*10^7$	$1.8*10^7$
Conv3	16*16	$7.6*10^7$	$2.7*10^7$	$4.5*10^7$	$3.6*10^7$
Pool2					
Conv4	8*8	$3.8*10^7$	$1.3*10^7$	$2.2*10^7$	$1.8*10^7$
Conv5	8*8	$7.6*10^7$	$2.7*10^7$	$4.5*10^7$	$3.6*10^7$
Conv6	8*8	$7.6*10^7$	$2.7*10^7$	$4.5*10^7$	$3.6*10^7$
Pool3					
Conv7	4*4	$3.8*10^7$	$1.3*10^7$	$2.2*10^7$	$1.8*10^7$
Conv8	4*4	$7.6*10^7$	$2.7*10^7$	$4.5*10^7$	$3.6*10^7$
Conv9	4*4	$7.6*10^7$	$2.7*10^7$	$4.5*10^7$	$3.6*10^7$

Pool4					
Conv10	2*2	$1.9*10^7$	$6.7*10^7$	$1.1*10^7$	$8.9*10^6$
Conv11	2*2	$1.9*10^7$	$6.7*10^7$	$1.1*10^7$	$8.9*10^6$
Conv12	2*2	$1.9*10^7$	$6.7*10^7$	$1.1*10^7$	$8.9*10^6$
Pool5					
Fc1	1*1	$4.2*10^6$	$2.5*10^6$	$4.2*10^6$	$3.3*10^6$
Fc2	1*1	$3.4*10^7$	$3.4*10^7$	$3.4*10^7$	$3.4*10^7$
Fc3	1*1	$8.0*10^5$	$8.0*10^5$	$8.0*10^5$	$8.0*10^5$
total		$6.7*10^8$	$2.6*10^8$	$4.1*10^8$	$3.3*10^8$
FLOPs compression ratio		1.00	2.57	1.62	1.99

Shown in Fig. 4, the network compression ratio α is gradually reduced from 1 to 0.2. And the influence of different pruning algorithms on the accuracy of the model is observed, as shown in Table 2. The results show that the accuracy of CDP is still higher than the previous two algorithms. When $\alpha \leqslant 0.6$, the accuracy is improved by 0.3% to 1.0% or even higher. The WLP algorithm that permanently removes the most convolution kernels has the lowest accuracy. It proves that permanently removing the convolution kernel will cause an irreversible reduction in model capacity, thereby affecting the accuracy. The result is similar to the that based on M-CifarNet.

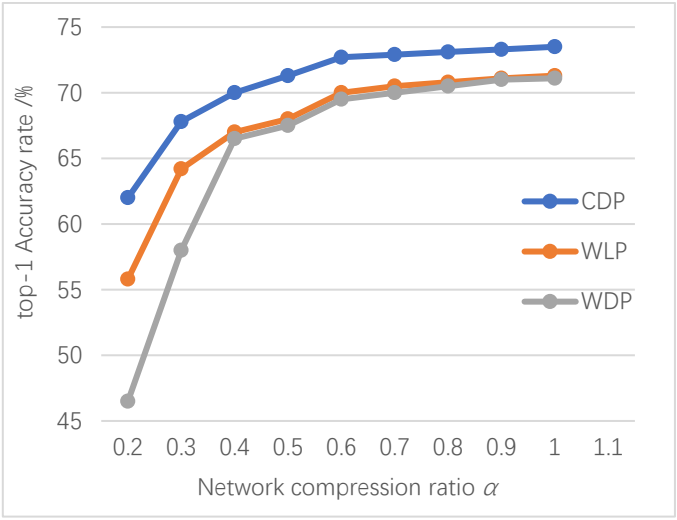

Fig. 4. Accuracy changes brought by different pruning algorithms (VGG16).

3.3. Comprehensive Comparison and Analysis

The experimental results based on M-CifarNet and VGG16 show that the joint dynamic pruning achieves the floating point compression ratios of 2. 11 and 1. 99, respectively, while the accuracy rate only decreases. Compared with the benchmark model of M-CifarNet and VGG16, each result is less than 0.8 percentage points and 1.2 percentage points.

Table 3 shows the changes of the accuracy, FLOPs, parameter scale and other indicators of different pruning algorithms on M-CifarNet. The bold items in the table indicate that the accuracy of the three algorithms is the best on different indicators. Moreover, the special cases of the three algorithms, including CDP (γ = 40%), WDP (γ = 30%), and WLP (γ = 20%), are selected for comparison instead of hose at the same pruning rate γ. This is because when the pruning rate γ is the same, the accuracies of the WDP and WLP algorithms are is lower than that of the CDP algorithm. Therefore, it is necessary to appropriately reduce the pruning rate and increase the accuracy to certain level comparable to that of CDP (γ = 40%). After experiments, WLP (γ = 20%) and WDP (γ = 30%) have been selected as comparison references.

It can be seen that the accuracy of the CDP algorithm is basically the same as that of the WLP and WDP algorithms, and it is slightly higher than the latter two. In the case of consistent accuracy, the results of FLOPs compression ratio and parameter compression ratio can be further discussed. The CDP algorithm provides the highest FLOPs compression ratio, which is 2.11. However, WLP and WDP only provide FLOPs compression ratios of 1.57 and 1.42, respectively. It can be seen from the parameter compression ratio. The values for WLP and CDP are both roughly equal to 1.57. Since the FBS algorithm does not permanently remove the convolution kernel, it has no effect on the parameters of the model. That is to say, the condition of consistent accuracy, CDP algorithm provides the highest FLOPs compression ratio and parameter size compression ratio.

Table 3. M-CifarNet Network Structure Based CIFAR-10 Dataset and Comprehensive Comparison of Different Pruning Algorithms [13]

model	Top-1/%	Top-5/%	FLOPs /%	FLOPs compression ratio	Parameter scale	Parameter compression ratio
M-CifarNet	93. 75	99. 80	3. 49×108	1. 00	1. 29×106	1.00
WLP(r=20%)	92.88	99. 74	2. 22×108	1. 57	8. 23×105	1.57
WDP(r=30%)	92.90	99. 76	2. 45×108	1. 42	1. 29×106	1.00
CDP(r=40%)	92. 95	99. 78	1. 66×108	2. 11	8. 23×105	1.57

Based on the above analysis, it can be concluded that CDP has the highest accuracy rate when the network compression ratio is the same. Moreover, when the accuracy rate is the same, CDP has the highest FLOPs compression ratio and close to the highest parameter compression. Generally speaking, WLP sacrifices accuracy in exchange for lower model complexity, while WDP as a whole shows a disadvantage. However, WDP introduces an additional fully connected layer, which results in a larger oscillation amplitude during training and a slightly slower convergence rate than WLP.

4. Future Research Trends

The success of joint dynamic pruning lies in the combination of channel dynamic compression and convolution kernel dynamic pruning. Channel dynamic compression uses structural pruning to replace unstructured pruning, and prunes in the unit of channel to speed up the pruning rate. Convolution kernel dynamic pruning is to permanently remove some convolution kernels of low importance from the model, maximize the capacity of the model, and improve the compression rate as much as possible while ensuring the accuracy.

Although the convolution neural network model pruning method based on weight correlation can also efficiently compress and accelerate the model, it lacks the combination with other model compression methods, resulting in disadvantages such as knowledge distillation and quantization. Thus, it is necessary to further lighten the model.

The model pruning method integrating weak layer punishment uses the theory of FPGM to evaluate the redundancy of convolution cores, uses the Euclidean distance to calculate the information distance of all convolution cores in each convolution layer, and eliminates the difference between layers through the normalization function based on contribution degree. The advantage of this model is that it successfully quantifies the information distance between convolution layers, so as to eliminate the differences between layers, and has achieved the purpose of evaluating the importance of convolution kernel from the global level to complete the screening task. However, this method also has limitations. It still adopts the traditional static pruning algorithm, which cannot deal with the more complex dynamic neural network.

In conclusion, compared with the existing dynamic pruning algorithms, the joint dynamic pruning algorithm has better accuracy and floating-point operation compression ratio; However, compared with the

traditional static pruning algorithm, the parameter compression ratio is still not high enough and the convergence is slow. In order to achieve better pruning effect, the joint dynamic pruning algorithm still needs to gradually reduce the pruning rate until it reaches the required compression ratio. In the future, some advantages of static pruning, including convolution layer information, distance pruning and weight association, will be widely integrated and added to the dynamic pruning model to construct a new pruning method. Diversification and lightweight will be the future development trend of pruning algorithm model.

5. Conclusion

In this paper, by quoting the experimental results of three different pruning methods of joint dynamic pruning, convolutional neural network model pruning based on weight correlation, and convolutional neural network model pruning method fused with weak layer penalty, the experiment results of different researchers, Compare and analyze the effects of the pruning algorithms of these three models, and predict the trend of pruning optimization in the future. We found a conclusion: when the pruning rate is the same, joint dynamic pruning has the best Flops compression ratio and parameter accuracy of the three, and the comprehensive level is better than the model pruning method that fuses weak layer penalties, and the weights are correlated. The pruning method of the convolutional neural network model has the worst effect. The success of joint dynamic pruning lies in the fusion of the two algorithms of channel dynamic compression and convolution kernel dynamic pruning. The former uses structured pruning instead of unstructured pruning, and pruning takes the channel as the unit to speed up the pruning rate. The latter is to permanently remove some less important convolution kernels from the model, while maximizing the capacity of the model, and increasing the compression rate as much as possible while ensuring the accuracy. However, the pruning method of convolutional neural network model based on weight correlation lacks the combination with other model compression methods, resulting in weak areas such as knowledge distillation and quantification, and the model needs to be further made lighter. The model pruning method that incorporates the weak layer penalty has a strong limitation. It still uses a similar traditional static pruning algorithm, so it cannot cope with more complex dynamic neural networks, and the pruning effect is also very poor.

Therefore, we believe that in the future, some of the advantages of static pruning, including convolutional layer information distance pruning and weight association, will be widely integrated and added to the dynamic pruning model to form a new pruning method that is diversified and lightweight It will be the future development trend of the pruning algorithm model.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Song conducted the research; Xia analyzed the data; Song and Xia wrote the paper; all authors approved the final version.

Acknowledgment

The authors thank Dr. Lu from Institute of Computer Technology, Chinese Academy of Sciences for support for the title.

References

[1] Chen, S.-L., Tian, S., Ma, J.-W., Liu, Q., Yang, C., Chen, F., & Yin, X.-C. (2021). End-to-end trainable network

for degraded license plate detection via vehicle-plate relation mining. *Neurocomputing, 446*, 1–10.

[2] Gangwar, A., Gonzlez-Castro, V., Alegre, E., & Fidalgo, E. (2021). Attm-cnn: Attention and metric learning based cnn for pornography, age and child sexual abuse (csa) detection in images. *Neurocomputing, 445*, 81–104.

[3] Liang, Y., Qin, G., Sun, M., Yan, J., & Jiang, H. (2021). Mafnet: Multi-style attention fusion network for salient object detection. *Neurocomputing, 422*, 22–33.

[4] Liu, Y., Shen, J., & He, H. (2020). Multi-attention deep reinforcement learning and reranking for vehicle re-identification. *Neurocomputing, 414*, 27–35.

[5] Cao, C., Cao, Z., & Cui, Z. (2020). Ldgan: A synthetic aperture radar image generation method for automatic target recognition. *IEEE Trans. Geosci. Remote Sens, 58(5)*, 3495–3508.

[6] Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*, 436–444.

[7] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Proceedings of the Advances in Neural Information Processing Systems (NIPS)* (pp. 1097−1105). Tahoe: IEEE.

[8] Sercu, T., Puhrsch, C., Kingsbury, B., & Le, C. Y. (2016). Very deep multilingual convolutional neural networks for LVCSR. *Proceedings of the Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4955–4959). Shanghai: IEEE.

[9] Liang, T., Glossner, J., Wang, L., Shi, S., & Zhang, X. (2021). Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing, 461*, 370–403.

[10] Augasta, M. G., & Kathirvalavakumar, T. (2013). Pruning algorithms of neural networks – A comparative study. *Open Computer Science, 3*, 105–115.

[11] Lei, W., Chen, H., & Wu, Y. (2017). Compressing deep convolutional networks using k means based on weights distribution. *Proceedings of the 2nd International Conference on Intelligent Information Processing*, New York, USA: ACM Press.

[12] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *Proceedings of ICLR 2015*.

[13] Zhang, M. M., Lu, Q. N., Li, W. Z., & Song, H. (2021). Deep neural network compression algorithm based on joint dynamic pruning. *Computer Application, 41(06)*, 1589-1596.

[14] Yan, Y. C., Guo, R. Z., & Yang, J. X. (2021). Pruning method of convolutional neural network model based on weight association. *Microcomputer System, 42(07)*, 1500-1504.

[15] Cai, H., Gan, C., Wang, T., Zhang, Z., & Han, S. (2019). Once-for-all: Train one network and specialize it for efficient deployment. *Proceedings of ICLR 2020* (pp. 1–15).

[16] Fang, Z. Y., Shi, S. D., Zheng, J. Q, & Hu, J. D. (2021). A pruning method of convolutional neural network model with weak layer penalty. *Computer Engineering*, 1-8.

Qidi Song was born in 2000. He is an undergraduate student at International Education College, Changchun University of Technology. He is major in electrical and electronic engineering.

Xuanze Xia was born in 2000. He is an undergraduate student at SHU-UTS SILC Business School, Shanghai University. He is major in information system and information management.