IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 24, 2019, accepted July 30, 2019, date of publication August 5, 2019, date of current version August 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933032

Structured Pruning of Convolutional Neural Networks via L1 Regularization

CHEN YANG[1,2], ZHENGHONG YANG[1,2], ABDUL MATEEN KHATTAK[2,3], LIU YANG[1,2], WENXIN ZHANG[1,2], WANLIN GAO[1,2], AND MINJUAN WANG[1,2]

[1]Key Laboratory of Agricultural Informatization Standardization, Ministry of Agriculture and Rural Affairs, Beijing 100083, China
[2]College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
[3]Department of Horticulture, The University of Agriculture, Peshawar 25120, Pakistan

Corresponding authors: Wanlin Gao (wanlin_cau@163.com) and Minjuan Wang (minjuan@cau.edu.cn)

This work was supported by the Project of Scientific Operating Expenses from Ministry of Education of China under Grant 2017PT19.

ABSTRACT Deep learning architecture has achieved amazing success in many areas with the recent advancements in convolutional neural networks (CNNs). However, real-time applications of CNNs are seriously hindered by the significant storage and computational costs. Structured pruning is a promising method to compress and accelerate CNNs and does not need special hardware or software for an auxiliary calculation. Here a simple strategy of structured pruning approach is proposed to crop unimportant filters or neurons automatically during the training stage. The proposed method introduces a mask for all filters or neurons to evaluate their importance. Thus the filters or neurons with zero mask are removed. To achieve this, the proposed method adopted L1 regularization to zero filters or neurons of CNNs. Experiments were conducted to assess the validity of this technique. The experiments showed that the proposed approach could crop 90.4%, 95.6% and 34.04% parameters on LeNet-5, VGG-16, and ResNet-32respectively, with a negligible loss of accuracy.

INDEX TERMS Convolutional neural networks, regularization, structured pruning.

I. INTRODUCTION

During the recent years, convolutional neural networks (CNNs) [1] have accomplished successful applications in many areas such as image classification [2], object detection [3], neural style transfer [4], identity authentication [5], information security [6], speech recognition and natural language processing. However, these achievements were made through leveraging large-scale networks, which possessed millions or even billions of parameters. Those large-scale networks heavily relied on GPUs to accelerate computation. Moreover, devices with limited resources, such as mobile, FPGA or embedded devices, etc. have difficulties to deploy CNNs in actual applications. Thus, it is critical to accelerate the inference of CNNs and reduce storage for a wide range of applications [7].

According to the studies done so far, the major approaches for compressing deep neural networks can be categorized into four groups, i.e. low-rank decomposition [8], parameter quantization [9], knowledge distillation [10]–[13], and network pruning [14]. For the deep neural networks (DNN) that have been trained, the low-rank decomposition technology decomposes and approximates a tensor to a smaller level to achieve compression. The low-rank decomposition achieves efficient speedup because it reduces the elements of the matrix. However, it can only decompose or approximate tensors one by one within every layer, and cannot discover the redundant parameters of DNN. Besides, more research has been focused on network module designs, which are smaller, more efficient and more sophisticated. These models, such as SqueezeNet [15], MobileNet [16] and Shufflenet [17], are basically made up of low resolutions convolution with lesser parameters and better performance.

At present, network pruning is a major focus of research, which not only accelerates DNN, but also reduces redundant parameters. Actually, using a large-scale network directly may provide state-of-the-art performance, so learning a large-scale network is needed. However, optimum network architecture may not be known. Thus, a massive redundancy exists in large neural networks. To combat this problem, network pruning is useful to remove redundant parameters, filters, channels or neurons, and address the over-fitting issue.

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Shen.

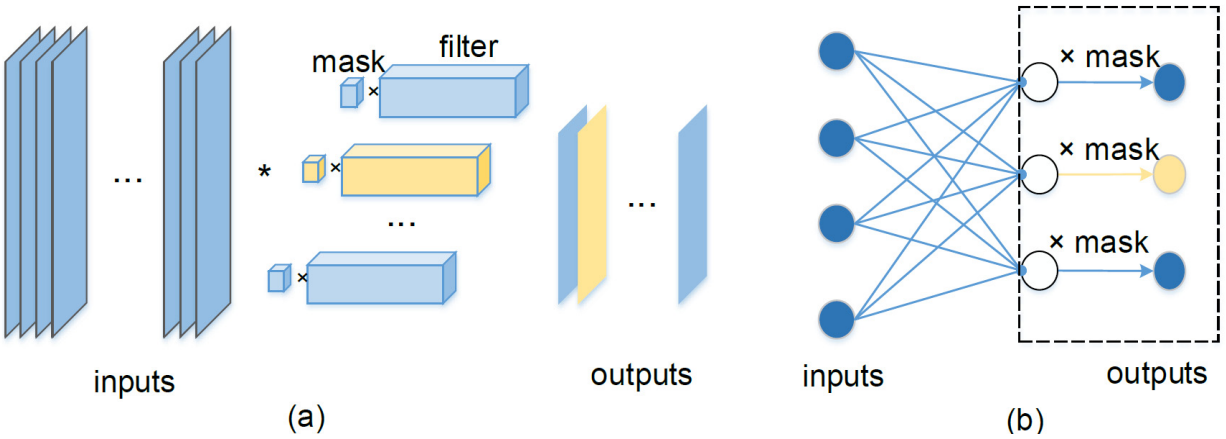

FIGURE 1. The architecture of the layer with the mask. (a) The architecture of a convolutional layer with the mask. (b) The architecture of a fully-connected layer with the mask. The proposed approach chooses the unimportant filters and neurons (highlighted in yellow) by the order of magnitude of mask value.

Network pruning techniques can also be broadly categorized as structured pruning and non-structured pruning. Non-structured pruning aims to remove single parameters that have little influence on the accuracy of networks and non-structured pruning is efficient and effective for compacting networks. Nonetheless, non-structured pruning is difficult to be widely used in practical applications. Actually, the operation of convolution is reformulated as a matrix-by-matrix multiplication in many prevalent deep learning frameworks. This requires additional information to represent pruned locations in non-structured pruning method. Therefore, special hardware or software is needed to assist with the calculation, which may increase computation time. Instead, structured pruning directly removes the entire filters, channels or neurons. Thus, the remaining network architecture can be used directly by the existing hardware. For example, Anwar *et al.* [18] employed particle filtering to structured sparsity convolutional neural network at channel-wise, kernel-wise, and intra-kernel stride levels. At present, several structured pruning methods [24], [25], [27] are mainly based on the statistical information of parameters or activation outputs. These methods do not consider the loss and are unable to remove parameters during training. In addition, some methods, such as those mentioned by [19], [20], require layer-by-layer iterative pruning and recovery accuracy, which involves enormous calculations. On the contrary, the proposed approach links pruning with minimization of loss and can be implemented during the training.

It is inspiring that the filters who's weights are all zero can be safely removed, because, whatever the input, they would not extract any features. This study presents a scheme to prune filters or neurons of fully-connected layers based on L1 regularization [21] to zero out the weights of some filters or neurons. Similar to this method, Wen *et al.* [31] adopted group LASSO regularization [40] to zero out filters. However, all the weights are required to compute an extra gradient, which is computationally expensive for a large-scale network. Contrarily, in the proposed method, a mask is introduced to address this issue and the regularization term only is the l_1-norm of the mask, which easily calculates the gradients of the mask. In this method, the parameters of filters or neurons are multiplied by a mask to pick unimportant filters or neurons, and once the mask is zero the corresponding filter or neuron will be removed. Here, though a mask is introduced for filters or neurons, the method does not change the architecture of the network. This allows for other compression methods to be used with the proposed technique. Similar to the proposed method, Lin *et al.* [32] also adopted a mask to identify unimportant filters or neurons, but the value of the mask could not be changed by training. In addition, removing unimportant filters or neurons may temporarily degrade accuracy, but the network can be retrained for recovery performance. FIGURE 1 shows the framework of the proposed method.

In this article, a structured pruning technology is presented, which allows for simultaneously learning and removing unimportant filters or neurons of CNNs. The main contributions are as follows:

- A simple yet effective method based L1 regularization is presented to compress CNNs model during the training stage.
- A threshold is adopted to solve the optimization problem of l_1-norm. In this approach, only some mask values are required to be near zero, though not completely zero. The detail is provided in the following section.

II. PREVIOUS WORK

The importance of compressing deep learning models before the application is self-evident, especially for expanding the application scenarios of deep learning [11]. For example, a compressed deep learning model can be combined with edge computing [12] to enable Internet of things devices understand data. In this section, we will review the contributions of others.

Le Cun *et al.* [14] first proposed a saliency measurement method called Optimal Brain Damage (OBD) to selectively delete weights by second-derivative information of error function. Later, Hassibi and Strok [22] proposed the Optimal Brain Surgeon (OBS) algorithm based on OBD. The OBS not only removed unimportant weights but also automatically adjusted the remaining weights, which improved accuracy and generalization ability. All these methods are based on Taylor expansion (even OBD and OBS are required to compute Hessian matrix), which may be computationally intensive especially for large networks. In addition, they use a criterion of minimal increase in error on the training data. Guo *et al.* [23] introduced a binary matrix to dynamically choose important weights. Han *et al.* [24], [25] directly removed weights with values lower than a predefined threshold to compress networks, then followed by retraining to recover accuracy. Considering most filters in CNNs that tended to be smooth in the spatial domain, Liu *et al.* [26] extended Guo's work to the frequency domain by implementing Discrete Cosine Transform (DCT) to filters in the spatial domain. However, these non-structured pruning technologies were hard to use in real applications, because extra software or hardware was required for the calculation.

Directly cropping a trained model by the value of weight is a wide method. Normally it is used to find an effective evaluation to judge the importance of weights and to cut the unimportant connection or filter to reduce the redundancy of a model. Hu *et al.* [27] thought the activation outputs of a significant portion of neurons were zero in a large network, whatever inputs the network received. These zero activation neurons were unimportant, so they defined the Average Percentage of Zeros (ApoZ) to observe the percentage of activations of a neuron and cropped the neurons with fewer activations. Li *et al.* [28] introduced a structured pruning method by measuring the norm of filters to remove unimportant filters. Luo *et al.* [29] took advantage of a subset of input channels to approximate output for compressing convolutional layers. Changpinyo *et al.* [30] proposed a random method to compress CNNs. They randomly connected the output channel to a small subset of input channels to compress CNNs. Though successful to an extent, their method did not directly relate to the loss, hence it was necessary to retrain the network for the recovery of accuracy. On the other hand, such a scheme could only be used layer-by-layer. Thus, it was essential to iterate over and over to prune, which would result in massive computation costs.

Ding *et al.* [37] applied a customized L2 regularization to remove unimportant filters and simultaneously stimulate important filters to grow stronger. Lin *et al.* [32] proposed a Global & Dynamic Filter Pruning (GDP) method, which could dynamically recover the previously removed filters. Liu *et al.* [33] enforced channel-level sparsity in the network to compress DNNs in the training phase. In addition, Gordon *et al.* [39] iteratively shrank and expanded a network targeting reduction of particular resources (e.g. FLOPS, or the number of parameters).

III. The APPROACH OF STRUCTURED PRUNING FOR CNNs

A. NOTATIONS

First of all, notations are clarified in this section. CNN is a multi-layer deep feed-forward neural network, which is composed of a stack of convolutional layers, pooling layers, and full-connected layers. In an l-layer CNNs model, $W_l^k \in \mathbb{R}^{d\times d\times C_{l-1}}$ represents the k-th filter of l layer, C_{l-1} denotes the number of feature maps in l-1 layer and d indicates the kernel size. Let us denote feature maps in the l layer by $Z_l \in \mathbb{R}^{H_l\times W_l\times C_l}$, where $H_l \times W_l$ is the size, C_l is the number of channels, and Z_l is the output of l-1 layer. In addition, Z_l^k represents the k-th feature map of l layer. The output feature map Z_l^k can be computed as:

$$Z_l^k = f(Z_{l-1}^* W_l^k + b_l^k), \tag{1}$$

where f is a non-linear activation function, * is the convolutional operation and b_l^k is the bias. $D = \{X = \{x_1, x_2, \ldots, x_N\}, Y = \{y_1, y_2, \ldots, y_N\}\}$ represents the training set, where x_i and $y_i$represent the training sample and label respectively, and N indicates the number of samples.

B. THE PROPOSED SCHEME

The goal of pruning is to remove those redundant filters or neurons, which are unimportant or useless for the performance of the networks. Essentially, the main role of the convolutional layer filters is to extract local features. However, once all the parameters of a filter are zeroed, the filter is confirmed unimportant. Whatever the inputs for the filter, the outputs are always zero. Under the circumstance, the filters are unable to extract any information. When the filters are multiplied by zero, all the parameters of the filters become zero. Based on this observation, a mask is introduced for every filter to estimate its importance. This can be formulized as:

$$Z_l^k = f(Z_{l-1}{}^*(W_l^k \times m_l^k) + b_l^k), \tag{2}$$

where m_l^k represents the k-th mask of l-layer.

Therefore, the problem of zeroing out the values of some filters can be transformed to zero some mask. For this purpose, the following optimization solution is proposed:

$$\begin{aligned} &\min_W \ L(Y, F(X; W, m)) \\ &s.t. \ \ \| \mathrm{m} \|_0 \leq C, \end{aligned} \tag{3}$$

where $L(\cdot)$ is a loss function, such as cross-entropy loss, $F(\cdot)$ is the output of CNNs and C is a hyper-parameter that controls the number of pruned filters. Equation (3) is the core of the proposed method. Once the optimal solution of the equation is obtained, the pruning is achieved.

In addition, this method can also remove redundant neurons in a fully-connected layer. The inference of fully-connected layer can be represented by:

$$Z_l = f(W_l Z_{l-1} + b_l), \tag{4}$$

where $W_l \in \mathbb{R}^{m\times n}$ is a weight matrix and $Z_{l-1} \in \mathbb{R}^{n\times 1}$ is the input of l-th layer. Here, when fully-connected layers introduce mask, the inference of these layers can be reformulated as:

$$Z_l = f(W_l Z_{l-1} {}^{\circ} m_l + b_l), \tag{5}$$

where $m_l \in \mathbb{R}^m$ is a mask vector and $\circ$ is Hadamard product operator.

Equation (3) can be transformed into the following form based Lagrange multiplier:

$$\min_W L(Y, f(X, W, m)) + \alpha \parallel m \parallel_0, \tag{6}$$

where α is a coefficient associated with C. Equation (6) is an NP-hard problem because of the zero norm. Thus, it is quite difficult to obtain an optimal solution with equation (6). Therefore, l_1-norm is adopted to replace l_0-norm, as:

$$\min_W L(Y, f(X, W, m)) + \alpha \parallel m \parallel_1 . \tag{7}$$

Equation (7) can be solved by SGD in practical application, so the proposed method is simple and easy to implement. We just need to introduce a mask for each layer and train the network. Though the proposed method introduces mask, the network topology will be preserved because the mask can be absorbed into weight.

C. THRESHOLD

L1 regularization is a widely used sparse technology, which pushes the coefficients of uninformative features to zero. So a sparse network is achieved by solving equation (7). However, there is a problem in solving equation (7). Here the mask value cannot be completely zeroed in practical application, because the objective function (7) is non-convex and the global optimal solution may not be obtained. A strategy is adopted in the proposed method to solve this problem. If the order of magnitude of the mask value is small enough, it can be considered almost as zero. Thus, to decide whether the mask is zero, a threshold is introduced. However, considering only the value of the mask is meaningless if the mask is not completely zero. Because there is a linear transformation between mask and convolution. One can shrink the masks while expanding the weights to keep the product of them the same. Hence, considering the mask and weight simultaneously is necessary. The average value of the product of the mask and the weight is used to determine whether the mask is exactly zero or not? The specific definition can be presented as:

$$m_l^k = \begin{cases} m_l^k & if\ abs\left(E(m_l^k w_l^k)\right) \geq \beta \\ 0 & if\ abs\left(E(m_l^k\, w_l^k)\right) < \beta, \end{cases} \tag{8}$$

where β is a pre-defined threshold and $E(\cdot)$ is the average operation. This strategy is efficient and reasonable, which can be proved by the results of the experiment.

Algorithm 1 The Proposed Pruning Approach

Input: Train data D, CNNs model, threshold β, penalty factor C, mask m.

DO:

1. Initializing weight W and mask $m = 1$
2. Training the CNNs with the mask, for suitable C
3. Pruning the filters or neuron based the value of the mask
4. Fine-tuning the network by retraining

End

Merging weights and masks and then removing the mask layer.

Return the pruned network architecture and preserved weights.

D. FINE-TUNING AND OTHER REGULARIZATION STRATEGIES

Pruning may temporarily lead to degradation in accuracy, so fine-tuning is necessary to improve accuracy. Furthermore, the proposed method can be employed iteratively to obtain a narrower architecture. Actually, a single iteration of proposed method is enough to yield noticeable compaction. The method is elaborated in Algorithm 1.

Essentially, the purpose of this approach is to adjust some masks to adequately small order of magnitude. Therefore, L2 regularization can also serve as a regularization strategy in this approach.

IV. EXPERIMENTS

The approach was primarily evaluated through three networks: LeNet-5 on MNIST dataset, VGG-16 on CIFAR-10 dataset and ResNet-32 on CIFAR-10 dataset. The implementation of this approach was accomplished through the standard Keras library. All experiments were conducted through Intel E5-2630 V4 CPU and NVIDIA 1080Ti GPU.

A. DATASETS

1) MNIST

MNIST dataset of handwritten digits from 0 to 9 is widely applied to evaluate machine learning models. This dataset owns 60000 train samples and 10000 test samples.

2) CIFAR-10

The CIFAR-10 dataset [41] has a total of 60000 images consisting of 10 classes, each having 6000 images with 32×32 resolution. There are 50000 training images and 10000 test images. During training, a data augmentation scheme was adopted, which contained random horizontal flip, rotation, and translation. The input data was normalized using the means and standard deviations.

B. NETWORK MODELS

1) LENET-5

LeNet-5 is a convolutional neural network designed by LeCun *et al.* [34]. It has two convolutional and two

TABLE 1. The result of lenet-5 on mnist.

Layer	Baseline		α=0.0001		α=0.01		α=0.05		α=0.12	
	Maps	Parameters	Maps	Pruned	Maps	Pruned	Maps	Pruned	Maps	Pruned
Conv1	6	0.15k	6	0%	6	0%	4	33.33%	3	50.0%
Conv2	16	2.41k	14	12.5%	7	56.25%	5	68.75%	3	81.25%
Fc1	120	30.84k	25	79.17%	25	79.17%	11	90.83%	12	90.00%
Fc2	84	10.16k	27	67.85%	27	67.85%	25	70.23%	20	76.19%
Fc3	10	0.85k	10	67.85%	10	67.85%	10	70.23%	10	76.19%
Total	**44.42k**		**80.18%**		**88.84%**		**95.46%**		**97.01%**	
Test Accuracy	**99.03%**		**99.19%**		**98.77%**		**98.46%**		**97.59%**	

"Baseline" represents the original CNNs, "α" is a hyper-parameter controlling the number of pruned parameters, "Maps" denote the number of convolution kernels in convolutional layers or the number of neurons in fully-connected layers, and "Pruned" indicates the percentage of removed parameters.

full-connected layers. This network has 44.2K learnable parameters. In this network, dropout is used in the full-connected layer.

2) VGG-16

The original VGG-16 [35] has thirteen convolutional and two fully-connected layers and has 130M learnable parameters. However, VGG-16 is very complex for CIFAR-10 dataset. So the fully-connected layers were removed. Moreover, Batch Normalization was used after each convolution operation. The modified model has 14.7M learnable parameters.

3) RESNET-32

Deep residual network (ResNet) [42] is a state-of-the-art multiple CNNs architecture. In this paper, ResNet-32 was implemented to evaluate the proposed method. The used ResNet-32 had the same architecture as described in [42], which contained three stages of convolutional, one global average pooling after last convolutional layer and one fully-connected layer. In addition, when the dimensions increased, 1×1 convolution was adopted as identity mapping to match the dimensions. This network has 0.47M learnable parameters.

C. THE DETAIL OF TRAINING, PRUNING, AND FINE-TUNING

To obtain the baseline of accuracy in the experiments, we trained LeNet-5 on MNIST, VGG-16 on CIFAR-10, and ResNet-32 on CIFAR-10 from scratch. Then, the pruning was performed on the basis of the trained network and the strategy of regularization was chosen as L1 regularization, with the mask initialized to 1. Later, we would retrain the pruned network for the recovery of accuracy.

1) LENET-5 ON MNIST

The original network was normally trained from scratch, for a total of 30 epochs, by Adam [43] with a batch sizes of 128. The learning rate was initialized to 0.001, the weight decay was set to 0.0005. The momentum was set to 0.9 and the dropout rate was set to 0.5 for the fully-connected layer. While implementing the pruning training, only the epochs was modified. The epochs was set at 10 and the threshold mentioned above to select pruned filters was set at 0.01. The pruned network was then retrained to compensate for the loss of accuracy. We adopted the same hyper-parameter setting as in normal training.

2) VGG-16 ON CIFAR-10

To get the baseline accuracy, the network was normally trained from scratch by SGD with a batch size of 128. The total epochs were set to 60. The initial learning rate was set to 0.01 and then scaled up by 0.1 every 20 epochs. The weight decay was set at 0.0005 and the momentum at 0.9. While implementing the pruning training, epochs was set to 30 , the learning rate was scaled by 0.1 every 10 epochs and other settings remained the same, while the threshold was set at 0.01. Finally, the pruned model was retrained following the same pre-processing and hyper-parameter settings as the normal training.

3) RESNET-32 ON CIFAR-10

Generally, the network was trained from scratch by SGD as the baseline with a batch size of 128. The weight decay was set at 0.0001, the epochs were set at 120, and the momentum was set at 0.9. The initial learning rate was set at 0.1 and then scaled by 0.1 at 60 and 100 epochs. Here, for pruning training, the epoch was set at 30, the learning rate was scaled by 0.1 every 10 epochs and the other settings remained the same. After pruning, the network was retrained from scratch. The epochs was modified to 60 and the learning rate was scaled by 0.1 every 20 epochs.

D. RESULTS OF THE EXPERIMENTS

1) LENET-5 ON MNIST

As per the results in TABLE 1, 88.84% of the parameters were removed without any impact on performance. Based on the proposed method, 95.46% of the parameters were discarded as well with an accuracy loss of 0.57%.

TABLE 2. Result of VGG-16 on CIFAR-10 datasets.

α	Strategy	Test Accuracy %	Remained Parameters	Pruned
0 (Baseline)	—	91.04	14.73 M	0
0.003	A	90.28	1.55 M	89.5%
	B	**90.72**		
0.005	A	89.79	0.818 M	94.4%
	B	**90.53**		
0.008	A	89.04	0.417 M	97.16%
	B	**89.44**		
0.01	A	88.7	0.328 M	97.76%
	B	**89.0**		

"Strategy" denotes the retraining strategy, "A" represents a fine-tuning pruned network, "B" represents retraining the pruned network with randomly initialized weights.

TABLE 1 also reveals that there was enormous redundancy in fully-connected layers because at least 90% parameters of fully-connected layers could easily be dropped. According to the form, the proposed method may indeed seek important connections. The reasons can be summarized in two points. First, when parameters of 83.83% are removed, the accuracy doesn't change. This indicates that the pruned parameters are unimportant for maintaining the accuracy of the network. Second, it is difficult to remove some filters or neurons, especially the neurons of fully-connected layers, when the pruning rate gradually increases. So the remaining connections are crucial.

In addition, the convolutional layer, especially the first one, is hard to prune in comparison with the next layer. The possible explanation could be that the proposed method automatically selects the unimportant filters through a backpropagation algorithm. However, the backpropagation algorithm will cause the previous layer to suffer gradient vanishing problem. That is why the former layers are hard to prune compared to the later ones.

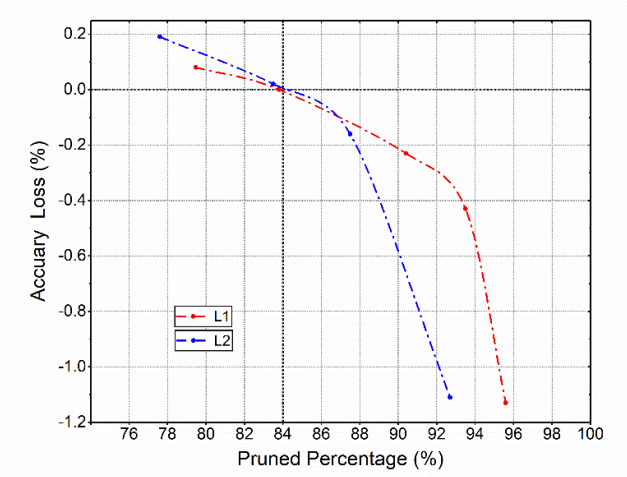

FIGURE 2. Comparison of L1 regularization and L2 regularization. "accuracy loss" represents the difference of accuracy between pruned CNNs and original CNNs. A positive value indicates the improvement of network accuracy after pruning, while a negative value indicates the decrease of accuracy.

2) VGG-16 ON CIFAR-10

As depicted in TABLE 2, over 94.4% of parameters could be removed with a negligible accuracy loss of 0.51%. It can also be observed that the loss of accuracy was only 2.04% when prune parameters of 97.76%. The proposed method proved to be effective again in reducing redundancy.

In fact, preserving the remaining architecture without retaining the parameters (training the pruned network from scratch) is also a strategy to fine-tune network. This strategy was adopted here to retrain the network and the results were promising, as shown in TABLE 2. The results reveal that a better effect can be achieved through directly retraining the pruned network from scratch. Perhaps the significance of the proposed method is that it furnishes the facility to discover excellent architectures, as mentioned by Liu *et al.* [36] as well. Nevertheless, training a pruned network from scratch is expensive in terms of computation cost, especially in case of large-scale datasets and networks.

3) RESNET-32 ON CIFAR-10

Pruning ResNet-32 based on the order of magnitude of the mask may result in different output map dimensions in the residual module. So a 1×1 convolution is needed as identity mapping to match dimensions. However, this operation brings about extra parameter and computation. To avoid this problem, a percentile was defined to remove filters of the same proportion in every convolutional layer. TABLE 3 shows that the proposed method removed 34% parameters with accuracy loss of 0.65%. Moreover, over 62.3% of parameters could also be discarded with an accuracy loss of 1.76%. Thus, it was confirmed that the proposed method could reduce the redundancy of complex network, i.e. ResNet.

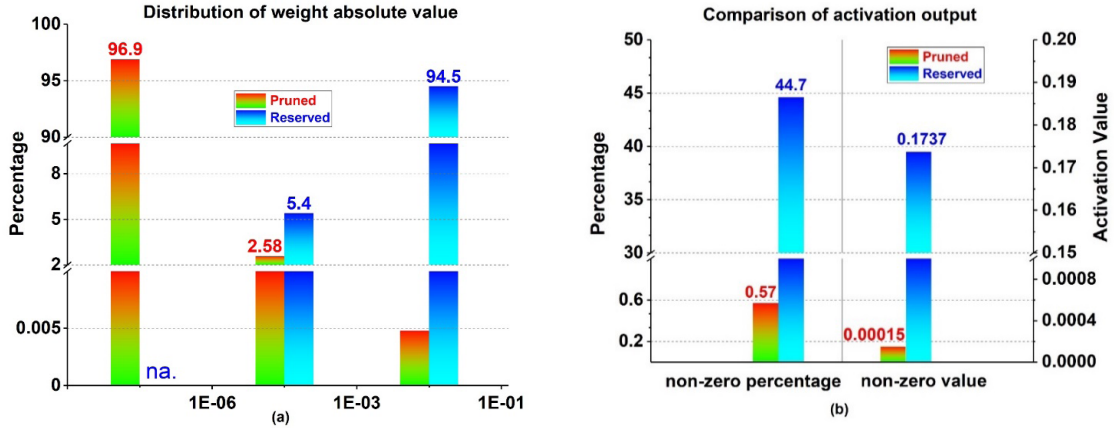

FIGURE 3. The comparison of pruned and reserved filters. (a) The comparison of parameters order of magnitude between pruned and reserved filters. The x-axis represents the distribution interval and the y-axis represents the percentage of the parameter in the interval. (b) The comparison of non-zero activations. The left bar represents average non-zero activation percentage, and the right bar represents average non-zero activation value.

TABLE 3. Restlt of RESNET-32 on CIFAR-10 datasets.

α	Proportion	Parameters	Test Accuracy(%)
0(baseline)	-	0.47M	90.03
0.001	20%	0.31M	89.38
	40%	0.177M	88.27

V. ANALYSIS

A. L2 REGULARIZATION

L2 regularization was also explored as a regularization strategy in this study. As shown in FIGURE 2, the LeNet-5 can also be compressed without degrading accuracy based L2 regularization. Nevertheless, there is some difference between L1 regularization and L2 regularization. Both L1 and L2 regularizations can improve accuracy when pruning rate is less than 84%, but the effect of L2 regularization is better. The main reason is that regularization techniques can prevent overfitting and improve the generalization ability. Moreover, with the pruning rate increasing, L1 regularization can achieve a greater compression effect in the same accuracy. As per Han *et al.* [24], L1 regularization pushes more parameters closer to zero, so it can prune more parameters. Having studied the difference between L1 regularization and L2 regularization, the inclination is more towards the L1 regularization from the perspective of compression and accuracy trade-off.

B. THE EFFECT OF PRUNING

To better describe the effect of the proposed method, a comparison was made between the pruned filters and reserved filters. The CONV3-1 layer of VGG-16, which owned 256 filters, was chosen while the α set at 0.008. Based on the above setting, 125 filters of CONV3-1 layer could be removed. Empirically, a weak filter or neuron always has lower activation outputs, lower activation frequency, and lower weight value. Hence weight values and activation outputs were chosen here to evaluate the difference between pruned and preserved filters.

As shown in Figure 2, the bulk of values of pruned parameters, with a percentage of 96.9, are less than 10^{-6}, in terms of the weight absolute values. However, most of the values of reserved parameters, with a percentage of 94.5, are greater than 0.001. The results indicate an enormous distribution difference between the values of the pruned and the reserved parameters. Therefore, the present approach can effectively reduce the order of magnitude of the pruned parameters.

In addition, the test set was chosen as a sample to calculate the average non-zero activation values and percentage of CONV3-1. As obvious from Figure 3, both the average percentage of non-zero activation and the average values of non-zero activation of the pruned filters was much lower than those of the reserved filters. From the activation perspective, the pruned filters were weak, because the output and weight values of pruned filters were negligible compared with the reserved filters and could be completely ignored. Thus, using the order of magnitude of the mask to determine pruned filters or neurons was reasonable.

C. COMPARISON WITH OTHER METHODS

In this section, two classical structured prune methods were compared with the proposed method. First, in LeNet-5 on MNIST-10 dataset, the proposed method was compared with that of Wen *et al.* [31]. In this experiment, both the proposed and Wen *et al.* [31] methods adopted the same coefficient of sparsity regularization ($\alpha = 0.03$). The results (TABLE 5) show that both the methods were analogous in terms of accuracy and compression effect. However, the proposed method is simpler and costs less computation in practice. Further, the proposed method was also compared with that

TABLE 4. Compare of VGG-16 on CIFAR-10.

	Liu[33]	Proposed
Pruned Parameters	13.52M (91.79%)	13.912M (94.47%)
Test Accuracy Error(%)	-0.92	-1.25

TABLE 5. Compare of LENET-5 on MNIST.

	SSL[31]	Proposed
Pruned Parameters (%)	93.3	95.46
Baseline Accuracy (%)	99.03	99.03
Test Accuracy Error (%)	-0.88	-0.57

of Liu *et al.* [33] in VGG-16 on CIFAR-10. Again, the same sparsity regularization coefficient ($\alpha = 0.005$) was adopted for both the methods. However, Liu *et al.* [33] adopted a fixed percentage threshold setting, whereas, the scheme of threshold setting of proposed method was different from Liu. The results (in TABLE 4) reveal that the proposed method was superior in terms of compression efficiency, although there was a slight loss of accuracy. In general, the proposed method can not only generate sparsity but also achieve better pruning effect with its improved threshold.

Nevertheless, some shortcomings were also observed with this approach. One is that though this approach doesn't change the existing CNNs architecture, the added mask layer essentially increases the number of layers in the network. This may increase optimization difficulty. However, this problem can be solved by Batch Normalization (BN [38]). The other is that, as this method introduces a threshold, the pruning effect may not be smooth. The pruning rate may change drastically with small changes in the α, which is not conducive to finding the best α.

VI. CONCLUSION

In this article, a structured pruning technology is proposed to automatically tailor redundant filters or neurons based on regularization. A mask is introduced to remove unimportant filters or neurons by zeroing the values of some masks during training. In addition, to deal with the problem that the mask cannot be completely zeroed in practice, a threshold is designed to zero the mask. Experimentation with multiple datasets has proved that the proposed method can effectively remove parameters with a negligible loss of accuracy. In the future, establishing a relation between the hyper-parameter α and the pruning rate will be considered to facilitate the adjustment of hyper-parameter α.

ACKNOWLEDGMENT

All the mentioned support is gratefully acknowledged.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[5] C. Shen, Y. Li, Y. Chen, X. Guan, and R. Maxion, "Performance analysis of multi-motion sensor behavior for active smartphone authentication," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 48–62, Jan. 2018.
[6] C. Shen, Y. Chen, X. Guan, and R. Maxion, "Pattern-growth based mining mouse-interaction behavior for an active user authentication system," *IEEE Trans. Dependable Secure Comput.*, to be published.
[7] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2017, *arXiv:1710.09282*. [Online]. Available: https://arxiv.org/abs/1710.09282
[8] C. Tai, T. Xiao, Y. Zhang, X. Wang, and E. Weinan, "Convolutional neural networks with low-rank regularization," 2015, *arXiv:1511.06067*. [Online]. Available: https://arxiv.org/abs/1511.06067
[9] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2285–2294.
[10] Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing deep convolutional networks using vector quantization," 2014, *arXiv:1412.6115*. [Online]. Available: https://arxiv.org/abs/1412.6115
[11] Z. Tian, S. Su, W. Shi, X. Du, M. Guizani, and X. Yu, "A data-driven method for future Internet route decision modeling," *Future Gener. Comput. Syst.*, vol. 95, pp. 212–220, Jun. 2018.
[12] Z. Tian, W. Shi, Y. Wang, C. Zhu, X. Du, S. Su, Y. Sun, and N. Guizani, "Real-time lateral movement detection based on evidence reasoning network for edge computing environment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4285–4294, Jul. 2019.
[13] R. Liu, N. Fusi, and L. Mackey, "Teacher-student compression with generative adversarial networks," 2018, *arXiv:1812.02271*. [Online]. Available: https://arxiv.org/abs/1812.02271
[14] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 598–605.
[15] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: https://arxiv.org/abs/1602.07360
[16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861
[17] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
[18] S. Anwar, K. Hwang, and W. Sung, "Structured pruning of deep convolutional neural networks," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 3, p. 32, 2017.
[19] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 1389–1397.
[20] J.-H. Luo and J. Wu, "An entropy-based pruning method for CNN compression," *arXiv:1706.05791*, 2017. [Online]. Available: https://arxiv.org/abs/1706.05791
[21] R. Tibshirani, "Regression selection and shrinkage via the lasso," *J. Roy. Stat. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.
[22] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Proc. Adv. Neural Inf. Process. Syst.*, 1993, pp. 164–171.
[23] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient DNNs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1379–1387.
[24] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1135–1143.
[25] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*. [Online]. Available: https://arxiv.org/abs/1510.00149

[26] Z. Liu, J. Xu, X. Peng, and R. Xiong, "Frequency-domain dynamic pruning for convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1043–1053.
[27] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures," 2016, *arXiv:1607.03250*. [Online]. Available: https://arxiv.org/abs/1607.03250
[28] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convNets," 2016, *arXiv:1608.08710*. [Online]. Available: https://arxiv.org/abs/1608.08710
[29] J.-H. Luo, J. Wu, and W. Lin, "ThiNet: A filter level pruning method for deep neural network compression," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 5058–5066.
[30] S. Changpinyo, M. Sandler, and A. Zhmoginov, "The power of sparsity in convolutional neural networks," *arXiv:1702.06257*. [Online]. Available: https://arxiv.org/abs/1702.06257
[31] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2074–2082.
[32] S. Lin, R. Ji, Y. Li, Y. Wu, F. Huang, and B. Zhang, "Accelerating convolutional networks via global & dynamic filter pruning," in *Proc. IJCAI*, 2018, pp. 2425–2432.
[33] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 2736–2744.
[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[36] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," 2018, *arXiv:1810.05270*. [Online]. Available: https://arxiv.org/abs/1810.05270
[37] X. Ding, G. Ding, J. Han, and S. Tang, "Auto-balanced filter pruning for efficient convolutional neural networks," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 6797–6804.
[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167
[39] A. Gordon, E. Eban, O. Nachum, B. Chen, H. Wu, T.-J. Yang, and E. Choi, "MorphNet: Fast & simple resource-constrained structure learning of deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1586–1595.
[40] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc., B (Statist. Methodol.)*, vol. 68, no. 1, pp. 49–67, 2006.
[41] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 4, 2009.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

CHEN YANG is currently pursuing the master's degree with the Department of College of Information and Electrical Engineering, China Agricultural University, Beijing, China. His research is about general deep learning and machine learning but his main research interest includes deep models compression.

ZHENGHONG YANG received the master's and Ph.D. degrees from Beijing Normal University, in 1990 and 2001, respectively. He is currently a Professor with the College of Science, China Agricultural University. He has presided two projects of National Natural Science Foundation. He has written two teaching and research books and has published more than 40 academic papers in domestic and foreign journals, among them, about 30 are cited by SCI/EI/ISTP. His major research interests include the matrix theory, numerical algebra, image processing, and so on. He is a member of Beijing and Chinese Society of Computational Mathematics.

ABDUL MATEEN KHATTAK received the Ph.D. degree in horticulture and landscape from the University of Reading, U.K., in 1999. He was a Research Scientist in different agriculture research organizations before joining the University of Agriculture, Peshawar, Pakistan, where he is currently a Professor with the Department of Horticulture. He has conducted academic and applied research on different aspects of tropical fruits, vegetables, and ornamental plants. He has also worked for Alberta Agriculture and Forestry, Canada, as a Research Associate, and Organic Agriculture Centre of Canada as a Research and Extension Coordinator, for Alberta province. There he helped in developing organic standards for greenhouse production and energy saving technologies for Alberta greenhouses. He is a Professor with considerable experience in teaching and research. He is currently a Visiting Professor with the College of Information and Electrical Engineering, China Agricultural University, Beijing. He has published 59 research articles in scientific journals of international repute. He has also attended and presented in several international scientific conferences. His research interests include greenhouse production, medicinal, aromatic and ornamental plants, light quality, supplemental lighting, temperature effects on greenhouse crops, aquaponics, and organic production.

LIU YANG is currently pursuing the master's degree with the College of Information and Electrical Engineering, China Agricultural University, Beijing, China. Her research interests include the application of image recognition and intelligent robots in the field of agriculture.

WENXIN ZHANG is currently pursuing the master's degree with the School of Information and Electrical Engineering, China agricultural university, Beijing, China. Her research interest includes pose estimation methods about pig based on deep learning for timely access to pig information.

WANLIN GAO received the B.S., M.S., and Ph.D. degrees from China Agricultural University, in 1990, 2000, and 2010, respectively. He is the currently the Dean of the College of Information and Electrical Engineering, China Agricultural University. He has been the principal investigator (PI) of over 20 national plans and projects. He has published 90 academic papers in domestic and foreign journals, among them, over 40 are cited by SCI/EI/ISTP. He has written two teaching materials, which are supported by the National Key Technology Research and Development Program of China during the 11th Five-Year Plan Period, and five monographs. He holds 101 software copyrights, 11 patents for inventions, and eight patents for new practical inventions. His major research interests include the informationization of new rural areas, intelligence agriculture, and the service for rural comprehensive information. He is a member of Science and Technology Committee of the Ministry of Agriculture, a member of Agriculture and Forestry Committee of Computer Basic Education in colleges and universities, and a Senior Member of Society of Chinese Agricultural Engineering, etc.

MINJUAN WANG received the Ph.D. degree from the School of Biological Science and Medical Engineering, Beihang University, under the supervision of Prof. Hong Liu, in June 2017. She was a Visiting Scholar with the School of Environmental Science, Ontario Agriculture College, University of Guelph, from October 2015 to May 2017. She is currently a Postdoctoral Fellow with the College of Information and Electrical Engineering, China Agricultural University. Her research interests mainly include bioinformatics and the Internet of Things key technologies.