Channel Pruning for Efficient Convolution Neural Networks

Wei Wang¹,²* and Liqiang Zhu¹,²
¹School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China
²Key Laboratory of Vehicle Advanced Manufacturing, Measuring and Control Technology (Beijing Jiaotong University), Ministry of Education, Beijing 100044, China
*Corresponding author E-mail: 16116336@bjtu.edu.cn

Abstract. In this paper, we propose a novel pruning method using feature maps Ln-norm to compress deep convolutional neural networks (CNNs) with large model size and long inference time. Different from existing pruning methods, which are mainly based on convolution kernels L1-norm, the proposed method uses output feature maps Ln-norm, where n is a variable. This method can accurately identify the redundant convolution filters and has good compression performance. The experiments on ImageNet show that the proposed method outperforms previous methods and is generally suitable for any CNNs models.

1. Introduction
In recent developments, convolutional neural networks (CNNs) have been a dominant choice in computer vision tasks, e.g., image classification [1], object detection [2] and semantic segmentation [3]. With the help of parallel computing platforms, CNNs [4, 5, 6] have become more resource-hungry algorithms with large demands in run-time memory and storage. The deployment of CNNs in real-time systems, such as video surveillance systems or mobile devices, is mostly constrained by run-time memory and large model size. The costs required by large CNNs can be excessive in terms of resource consumption, hardware cost and processing delay.

Many works have been proposed to reduce model size and accelerate the inference time. Tai et al. [7] propose the low-rank tensor decomposition for model compression in convolutional filters. Courbariaux et al. [8] use binary weights to train CNNs models. Hu et al. [9] propose to compute Average Percentage of Zeros (ApoZ) to identify and prune redundant filters. It is noteworthy that the pruning techniques are easy to balance acceleration and accuracy loss, compatible with other compression methods, and have the ability to prevent over-fitting.

In this paper, we propose a novel reliable pruning method to compress CNN models. Our approach uses feature maps Ln-norm to identify unimportant kernels. The experimental results on ImageNet dataset show that the proposed criterion outperforms the previous methods. The main contributions of this paper are two-fold: Firstly, we propose a channel-level pruning based on the feature map norms, other than the widely-used kernel norms, to evaluate the usefulness of filters; Secondly, we show that convolution layers in different depth focus on features at different abstract levels and propose a layer-wised Ln-norm criterion to facilitate information transfer and feature abstraction across layers.

2. Related Work
Pruning-based approaches can be divided into filter pruning and weight pruning. For weight pruning, LeCun et al. [10] prove that network pruning for reducing the network complexity and over-fitting is a valid method. Tan et al [11] propose to cluster weights into an associated fuzzy quantity space (FQS)
and remove the unimportant connections by thresholding. The weight pruning approaches require specially hardware and only achieve model acceleration with specially designed libraries or hardware.

For kernel pruning, Hu et al. [9] propose Average Percentage of Zeros (APoZ) to measure the percentage of zero activations of a neuron. This criterion is mainly applicable to the deeper convolution layers and fully-connected layers with zero activation neurons in large CNNs models. Li et al. [12] compute convolution filters L1 norms in each layer. The filters with small L1 norm are discarded and the pruned CNN is fine-tuned. As shown later in this paper, this criterion may lead to some important convolution kernels being pruned, which affects the output accuracy.

Different from the previous pruning methods, we propose a pruning method based on feature maps Ln-norm. In addition, in order to obtain better performance, layer-wised Ln-norms with variable n on different convolutional layers are proposed.

3. Layer-Wised Pruning Method

This section describes the proposed pruning method. We first discuss the general process of convolution operation and the pruning method for CNN models. We then propose the pruning criterion based on feature maps norms for the single convolutional layer. Finally, we present the feature map Ln-norms method for the whole network.

3.1. The Criterion of Feature Map Norm

CNN usually consists of many layers, and each convolutional layer contains a large number of convolutional filters. The 2D convolution operation can be described as:

\[ F(x, y) = \sum_{i,j} X(i, j, k, l) \times K(i, j, k, l) \]

Where \( K \) is the weight matrix of a convolution kernel in a layer, \( X \) is the input feature map of this layer with \( x \) as the input sample image and \( F \) is the output feature map.

We propose a pruning method for evaluating the significance of convolution filters based on the norm of their feature maps. For a sample \( x \) in training images, the feature map Ln-norm outputted by a convolution filter is calculated. Then the mean value of Ln-norm which is averaged over all training images is assigned to the filter. The filters are sorted by the corresponding output channels Ln-norms. Those filters with smaller feature map norm are discarded for accelerating the CNN model. In general, in order to reduce the calculation, a small part of images in the training dataset can be randomly selected to estimate the averaged Ln-norm:

\[ \bar{L_n} = \frac{1}{N} \sum_{i=1}^{N} \| F_i \|_n \quad (n = 0, 1, 2, \ldots, \infty) \]

Where \( \| F_i \|_n \) is the Ln-norm calculated by the corresponding output channel of the kernel \( K \), and \( x_i \) denotes the \( i \)th sample image which is fed into the CNN model.

Figure 1 shows the channel pruning method. When the feature maps channels in orange are pruned, its corresponding kernels and next connections in orange are removed. Symbol * denotes convolution operation. For the CNNs with fully-connected layers, unnecessary connections which are less than a pre-specified threshold can be removed. After one round of pruning is completed, the pruned CNN need to be fine-tuned. This step can be repeated to compress the model recursively and speed up the calculation. Besides, the GAP [13] layer can be used instead of fully-connected layers to further reduce the costs of model storage and calculation.
3.2. Different Norms on Different Layers

For better performance, we also propose to utilize output channels Ln-norms with variable n on different layers. To further understand the necessity of this step, an input sample image of an airplane, shown in Figure 2, into a VGG16 classification network, pre-trained on ImageNet which contains the class of airplane in Figure 2 among 1000 object classes. As shown in Figure 3(a), for the low layers, e.g., the first convolutional layer in VGG16 model (conv1-1), the simple information, such as shape, direction and corners can be obtained. For middle layers, as shown in Figure 3(b), the obtained features consist of relatively complicated objects and components on conv3-3. For deep layers, as shown in Figure 3(c), the valid target information of the airplane is obtained on conv5-3.

The probability distribution in normalized output feature maps on convolutional layers of VGG16 is shown in Figure 4. In the lower layers, the distributions are flat and the number of larger element points in feature maps is larger. This is because simple information features, obtained in these convolutional layers, are abundant. The feature activations at different positions should be treated equally while evaluating the effectiveness of filters, to conserve enough feature information for the deeper layers. Accordingly, L1 norm, absolute value summation of all feature elements, is more suitable to identify effective kernels because of more equally treating for all elements in a feature map. In deeper layers, the abstract patterns represented by the effective convolution filters are more concentrated in the output feature maps. In other words, feature maps with small number and concentrated large elements are outputted by effective filters, and most of positions should be near zero on the output feature maps, which can be significantly expressed by feature map Ln-norm where n>1. Generally, with the increase of network layer depth, Ln-norm with variable n could be available to concentrate the focus on different convolutional layers. In a word, for very deep CNNs models, the criterion with Ln-norms can be used to obtain better performance.
4. Experiments

The method proposed in this paper is applicable to the acceleration of any neural network with convolutional operations. In this section, we evaluate our pruning method on several widely used CNNs, i.e. VGG Net and Alex Net.

4.1. Dataset

The dataset used for transfer learning is ImageNet, a large visual database for visual object recognition and containing 1000 classes of objects. In our experiments, 10 classes are selected from ImageNet dataset to form a new and smaller dataset. Obviously, a CNN model, pre-trained on the 1000-class ImageNet, should have a large number of redundant convolution kernels for the new 10-class dataset, since the classification task on the latter is much simpler. In experiment, the shorter side of images is resized to 256 in proportion and the augmentation for retraining includes random crop of 224×224 and mirror. When the VGG16 model [5], pre-trained on all 1000 classes of images in ImageNet dataset, is directly used for the new classification task, test error of the model is 0.2%. A forward passing runs 26.6ms on our computer with Intel Xeon E5-2667 and Tesla K80 GPU.

Table 1. Network pruning configurations of VGG16.

| Network Layer | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---------------|---|---|---|---|---|---|---|---|---|---|
| conv1-1~conv1-2 | 64 | 61 | 58 | 55 | 52 | 49 | 46 | 43 | 40 | 40 |
| maxpooling | | | | | | | | | | |
| conv2-1~conv2-2 | 128 | 119 | 110 | 101 | 92 | 83 | 74 | 65 | 56 | 46 |
| maxpooling | | | | | | | | | | |
| conv3-1~conv3-3 | 256 | 231 | 206 | 181 | 156 | 131 | 106 | 81 | 56 | 42 |
| maxpooling | | | | | | | | | | |
| conv4-1~conv4-3 | 512 | 384 | 288 | 216 | 162 | 122 | 91 | 68 | 51 | 42 |
| maxpooling | | | | | | | | | | |
| conv5-1~conv5-3 | 512 | 384 | 288 | 216 | 162 | 122 | 91 | 68 | 51 | 42 |
| maxpooling | | | | | | | | | | |
| GAP | | | | | | | | | | |
| fc6~fc7 | / | / | / | / | / | / | / | / | / |
| fc8 | | | | | | | | | | |

4.2. VGG16 on ImageNet

VGG16 is a convolution neural network with 16 layers of single branch, with a total of 13 convolution layers and 3 fully-connected layers. A version of VGG16 [5] pre-trained on ImageNet dataset is used in the experiments. Noticing that parameters in the fully-connected layers fc6 and fc7 account for nearly 90% of the total network model memory, a GAP layer is used to replace Layers fc6 and fc7 before pruning. The experiment results show that this replacement has little effect on the accuracy of the model for the two datasets used in this study. In order to analyse the impact of the pruning process on network performance, VGG16 is pruned recursively using the strategy shown in Table 1. The column of pruning times 0 shows the number of convolution kernels of the original VGG16 network, and other columns give the number of convolution kernels remaining after each round of pruning.
100 pictures are selected randomly from the 10-class dataset to estimate the Ln-norm of feature maps. For the feature-map L1-L2-L∞-norm criterion, L1-norm is utilized in lower layers, such as conv1-1 and 1-2. L2-norm is utilized in middle layers, such as conv2-1 to conv5-2. L∞-norm is utilized in last layer, such as conv5-3. After nine pruning rounds, the model's memory and a forward-pass time are shown in Figure 5 and Figure 6 respectively. As shown in Table 2, a forward-pass time per image is reduced by 77.4%, and the network model only takes 770KB disk space, compressing about 680 times. The test accuracy of feature-map-L1-L2-L∞ criterion remains at a high level of 98.6%.

### Table 2. Performance of VGG16 after pruning.

| Model                              | Test-error (%) | GPU time (ms) | Model size (MB) |
|------------------------------------|----------------|---------------|-----------------|
| Original (Baseline)               | 0.2            | 26.6          | 512             |
| Kernel L1-norm [12]               | 2.8            | 6.01          | 0.76            |
| APoZ (Average Percentage of Zeros) [9] | 1.8            | 6.01          | 0.76            |
| Feature map L1-norm               | 1.6            | 6.01          | 0.76            |
| Feature map L1-L2-L∞-norm         | 1.4            | 6.01          | 0.76            |

In the experiment, eight pruning criteria are compared, and the results are shown in Figure 7. It can be seen that, with the increase of the pruning times, the accuracy on the validation set and test set are slowly decreasing. Obviously, the performances of feature map Ln-norm-based criteria are better than kernel norm criterion, and the accuracy of the feature map L1-L2-L∞-norm criterion is consistently highest at each round of pruning.

![Figure 5. Model size after each round of pruning.](image1)

![Figure 6. A forward-pass time on GPU after each pruning.](image2)

![Figure 7. Accuracy curves of eight pruning criteria on ImageNet.](image3)
Table 3. Performance of Alex Net after pruning.

| Solution                      | Test-error (%) | Model size (MB) |
|-------------------------------|----------------|-----------------|
| Original (Baseline)           | 0.9            | 232.5           |
| Kernel L1-norm [12]           | 2.4            | 1.6             |
| Feature map L1-norm           | 2.0            | 1.6             |
| Feature map L1-L2-L∞-norm     | 1.6            | 1.6             |

4.3. Alex Net on ImageNet

Unlike VGG16's single-branch architecture, Alex Net has a two-group convolution architecture with a total of 5 convolution layers and 3 fully-connected layers. In the experiment, the shorter side of images is first resized to 256 in proportion and the augmentation for retraining is random crop of $227 \times 227$ and mirror. Considering that the fully-connected fc6 and fc7 layer parameters account for nearly 89% of the total network model memory, a GAP layer is used to replace fc6 and fc7 layers before pruning. When the Alex Net model [1], pre-trained on all 1000 classes of images in ImageNet dataset, is directly used for the new classification task, the test error of this model is 0.9%. As shown in Table 3, with feature map L1-L2-L∞-norm criterion, the Alex Net model can be compressed from 232.5MB to 1.6MB, compressed by 145× and the loss accuracy is only 0.7%.

5. Conclusion

In this paper, we propose a channel pruning algorithm to accelerate large CNN models. During the process of acceleration, the convolutional layer is pruned recursively by utilizing the proposed feature map norm criterion, and the pruned CNN models can still maintain high accuracy. Our pruning method can be combined with other existing compression methods for further acceleration. It can meet the requirements of small run-time memory and real-time processing for mobile devices or video surveillance systems.

6. Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2016YFB1200401.

7. References

[1] Krizhevsky A, Sutskever I, Hinton GE, Imagenet classification with deep convolutional neural networks, in Proc. NIPS, 2012, pp. 1097–1105.
[2] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C.-Y, Berg AC. SSD: single shot multibox detector, in Proc. ECCV, 2016, pp. 21–37.
[3] Noh H, Hong S, Han B, Learning deconvolution network for semantic segmentation, in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 1520-1528.
[4] LeCun Y, Bottou L, Bengio Y, Haffner P, Gradient-based learning applied to document recognition, Proc. IEEE, Vol.66, pp. 2278–2324, Nov. 1998.
[5] Simonyan K, Zisserman A, (Sep. 2014). Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556.
[6] He K, Zhang X, Ren S, Sun J, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[7] Tai C, Xiao T, Zhang Y, Wang X, E W (Nov. 2015). “Convolutional neural networks with low-rank regularization.” [Online]. Available: https://arxiv.org/abs/1511.06067.
[8] Courbariaux M, Bengio Y, David J-P, BinaryConnect: Training deep neural networks with binary weights during propagations, in Proc. of the 28th International Conference on Neural Information Processing Systems, NIPS’15, 2015, pp. 3123–3131.
[9] Hu H, Peng R, Tai Y.-W, Tang C.-K. (Jul. 2016). Network Trimming: A data-driven neuron pruning approach towards efficient deep architectures. [Online]. Available: https://arxiv.org/abs/1607.03250.
[10] LeCun Y, Denker J. S, Solla S. A, Optimal brain damage. In: Advances in neural information processing systems, 1989, pp 598–605.

[11] Tan W.R, Chan C.S, Aguirre H.E, Tanaka K, Fuzzy qualitative deep compression network, Neurocomputing, vol. 251, pp. 1–15, Aug. 2017.

[12] Li H, Kadav A, Durdanovic I, Samet H, Graf H.P. (Aug. 2016). Pruning filters for efficient convnets, [Online]. Available: https://arxiv.org/abs/1608.08710.

[13] Lin M, Chen Q, Yan S. (Dec. 2013). Network In Network. [Online]. Available: https://arxiv.org/abs/1312.4400.