Exploring Group Sparsity using Dynamic Sparse Training

Geunhye Jo[1], Gwanghan Lee[2], and Dongkun Shin[1]
[1]Department of Electrical and Computer Engineering, [2]Department of Artificial Intelligence
Sungkyunkwan University, Suwon, Korea
{shurui, ican0016, dongkun}@skku.edu

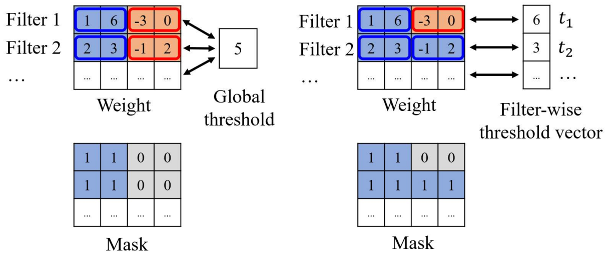
Abstract

Group-level pruning is a model compression method that can accelerate models on general hardware while maintaining the accuracy of models even at high sparsity. Existing group-level pruning studies have a limitation in that pruning is performed by evaluating the importance of the weight group based on a single global threshold, and thus the accuracy loss is significant. In this paper, we propose Grouped DST, which applies Dynamic Sparse Training to group-level pruning to learn the criteria for each filter to determine the importance of weight groups. We validate our technique at ResNet-20 on CIFAR-10. The model compressed with Grouped DST achieve 3.77% a lower accuracy loss compared to that pruned with a global pruning threshold at 95% of sparsity.

Keywords: Deep Learning, Model Compression, Group-Level Pruning

1. Introduction

Deep Convolutional Neural Networks (CNNs) show the state-of-the-art performance in many visual tasks, but as the memory and computation requirement of Deep CNNs is enormous, it has become difficult to deploy Deep CNNs to mobile or edge devices. Weight pruning [1] is one of the most widely used model compression methods addressing this problem, which removes unimportant neural weights. Typically, pruning is done in fine-grained manner (i.e. individual weight) or coarse-grained manner (i.e. convolutional filter). Fine-grained pruning can reduce most of parameters and computation without severe loss of accuracy, but it is difficult to lead to actual acceleration on general hardware. In contrast, coarse-grained pruning can directly result in acceleration, but suffers significant loss of accuracy as sparsity of weights increases.



(a) Group-level pruning with global threshold

(b) Grouped DST

Fig. 1. Comparison between group-level pruning scheme. Group size is 2.

Group-level pruning [2] is to group and prune consecutive weights together considering load behavior of SIMD unit. For a 4-way SIMD unit, for instance, four consecutive weights are determined together whether to be pruned. This approach hits a sweet spot between fine- and coarse-grained pruning. Models pruned in groups can be easily accelerated than models pruned in individual weights. Accuracy loss can be kept lower by pruning models in groups than in filters because the weaker regularity allows more of important weights to survive.

However, existing group-level pruning studies figure out whether each weight group is important by comparing it to global pruning threshold as in Fig. 1(a). It cannot consider differences in importance for layers and filters to which the weights belong; thus, the ability of group-level pruning to maintain model accuracy is underestimated.

In this paper, we propose Grouped Dynamic Sparse Training (DST) to learn different pruning thresholds for each filter to determine which weight groups are to be pruned by applying DST [3] in group-level pruning. It can reduce the accuracy loss of group-level pruned model with high sparsity.

2. DST

DST [3] is a method to make the model increasingly sparse by repeating pruning and recovering for each step

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.IITP-2017-0-00914, Software Framework for Intelligent IoT Devices)

in the training process. This is done by training the filter-wise pruning thresholds.

The weight of the convolutional layer can be represented by the matrix $W \in \mathbb{R}^{c_o \times z}$, where $z = c_i \times w \times h$ is the filter shape, c_o is the number of output channels, c_i is the number of input channels, w and h are the width and the height of the kernel. DST [3] sets a threshold parameter vector $\boldsymbol{t} \in \mathbb{R}^{c_o}$ to prune the weight matrix. Then the pruning mask $\{M_{ij}\} \in \mathbb{R}^{c_o \times z}$ is made as

$$M_{ij} = step(|W_{ij}| - t_i) \qquad (1)$$

where $step(.)$ is a function that returns 1 when the input is equal to or greater than 0, and returns 0 when the input is less than 0. Since the $step(.)$ function is undifferentiable, thresholds are trained using a long-tailed estimator [4] in the backpropagation process in DST [3]. And we introduce a sparse regularization term L_s that can raise the pruning rate of weights by increasing thresholds.

$$L_s = \sum_{i=1}^{N} \sum_{j=1}^{c_o^i} \exp(-t_j^i) \qquad (2)$$

where N is the total number of layers, c_o^i is the number of output channels of the i-th layer, and t_j^i is the threshold of the j-th filter in the i-th layer. In the total loss $L = L_{cls} + \alpha L_s$, where L_{cls} represents the cross-entropy loss. The ratio of remaining parameters is adjusted through hyperparameter α.

3. Grouped DST

We modified the method of creating a mask to prune the weights in groups as

$$M_{ik} = step\left(\sum_{j=Gk-G+1}^{Gk} |W_{ij}| - t_i \right), 0 < k < \frac{C_i}{G} \qquad (3)$$

where G is the group size and $M_{ik}$is obtained one by one for each group. According to this equation, if the L1-norm of the weights included in the group is smaller than the threshold, the mask is set to 0, and the entire group is pruned.

Fig. 1(b) shows the creation of a pruning mask by applying Grouped DST. Because different thresholds were applied in Filter 1 and Filter 2, it can be seen that a group pruned in Fig. 1(a) survives in Fig. 1(b).

4. Experiment Results

We evaluated Grouped DST with ResNet-20 [5] at CIFAR-10, an image classification task. The group size is 4. The first and the last layers of the model were not pruned. For comparison, we used a group-level pruned model [2] obtained by progressively pruning pretrained model based on the global threshold.

Fig. 2 shows the accuracy of pruned models against the pruning rate. Overall, the model pruned with Grouped DST has higher accuracy than the model to which the global threshold is applied. And when the pruning rate increases, the difference in performance between the models using the Grouped DST and the global threshold tends to increase. The performance of the model pruned with Grouped DST was 0.57% higher when the pruning rate was about 0.9 and 3.77% higher around 0.95. This result suggests that as the magnitude of the surviving weight decreases, it is more appropriate to judge the importance of the weight group by considering the relative magnitude within the filter to which the weight group belongs rather than to judge the importance of the weight group only by its absolute magnitude.

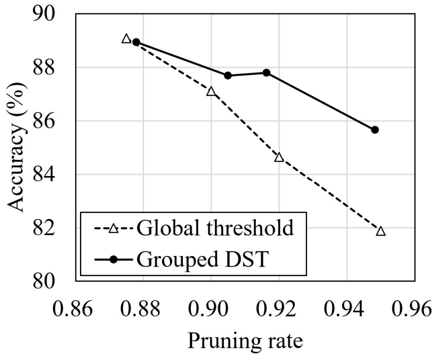


Fig. 2. Group-level pruning results.

5. Conclusion and Future Work

In this paper, we proposed to apply Dynamic Sparse Training in group level pruning to determine the importance of weight groups through training threshold for each filter. the CIFAR-10 classification error was reduced by up to 3.77% compared to the model pruned with the same threshold in all layers at high pruning rate. In the future, we plan to conduct research to find sparse weight structures at unaligned group-level by developing advanced sparse training.

References

[1] 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.*, pp. 1135-1143, 2015.

[2] K. Lee, H. Kim, H. Lee, and D. Shin, “Flexible group-level pruning of deep neural networks for on-device machine learning,” *in IEEE 2020 Des., Automat. & Test in Eur. Conf. & Exhib.,* pp. 79-84 March 2020.

[3] J. Liu, Z. Xu, R. Shi, R. Cheung, and H. So, “Dynamic Sparse Training: Find Efficient Sparse Network from Scratch with Trainable Masked Layers,” *arXiv preprint arXiv:2005.06870*, 2020.

[4] X. Zhe, and R. Cheung. "Accurate and Compact Convolutional Neural Networks with Trained Binarization," *30th Brit. Mach. Vision Conf.*, 2019.

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *In Proc. IEEE conf. Comput. Vision and Pattern Recognit.*, pp.770–778, 2016.