Local Normalization Based BN Layer Pruning

Yuan Liu[1], Xi Jia[1], Linlin Shen[1*], Zhong Ming[2], and Jinming Duan[3]

[1]Computer Vision Institute, Shenzhen University, Shenzhen, Guangdong, China

[2]Big Data Institute, Shenzhen University, Shenzhen, Guangdong, China

[3]School of Computer Science, University of Birmingham, England

ABSTRACT

Compression and acceleration of convolutional neural network (CNN) have raised extensive research interest in the past few years. In this paper, we proposed a novel channel-level pruning method based on gamma (scaling parameters) of Batch Normalization layer to compress and accelerate CNN models. Local gamma normalization and selection was proposed to address the over-pruning issue and introduce local information into channel selection. After that, an ablation based beta (shifting parameters) transfer, and knowledge distillation based fine-tuning were further applied to improve the performance of the pruned model. The experimental results on CIFAR-10, CIFAR-100 and LFW datasets suggest that our approach can achieve much more efficient pruning in terms of reduction of parameters and FLOPs, e.g., 8.64× compression and 3.79× acceleration of VGG were achieved on CIFAR, with slight accuracy loss.

Our Method

In this paper, we developed a Batch Normalization (BN) layer based network pruning approach including three contributions:

1. For the approach proposed by Liu et al. [6] with high pruning ratio, most and even all of the channels in the deep layers could be pruned and this is so called ”overpruning”. To address such issue and make the pruning more balanced, we proposed local gamma (scaling paramters) normalization and selection.

2. To relieve the potential loss brought by ignoring and simply removing the corresponding beta of pruned channels, we proposed ablation based beta (shifting paramters) transfer.

3. To further improve the performance of pruned neural network models, we proposed knowledge distillation based fine-tuning.

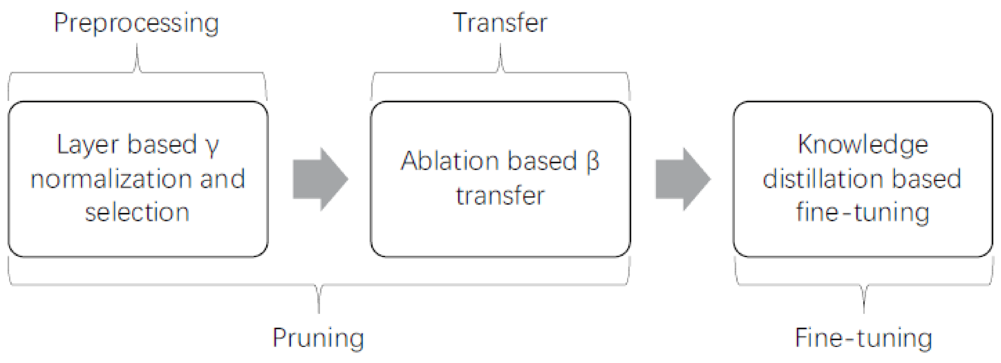


Fig. 2: Flow chart of our method

Local Normalization and Selection

$$\gamma^{l}_{\text{normalized}} = \frac{\gamma^{l} - \gamma^{l}_{\min}}{\gamma^{l}_{\max} - \gamma^{l}_{\min}}$$

Ablation based transfer

1. If the subsequent convolution layer is followed by a non-BN layer:

$$x^{l+1} = \sigma\left(w^{l+1} * x^{l} + b^{l+1}\right)(1)$$

$$b^{l+1}_{new} = \sum^{a}(I(\beta^{l}) \cdot \sigma(\beta^{l}) \sum^{i}\sum^{j} w^{l+1}_{:,a,i,j}\ (2)$$

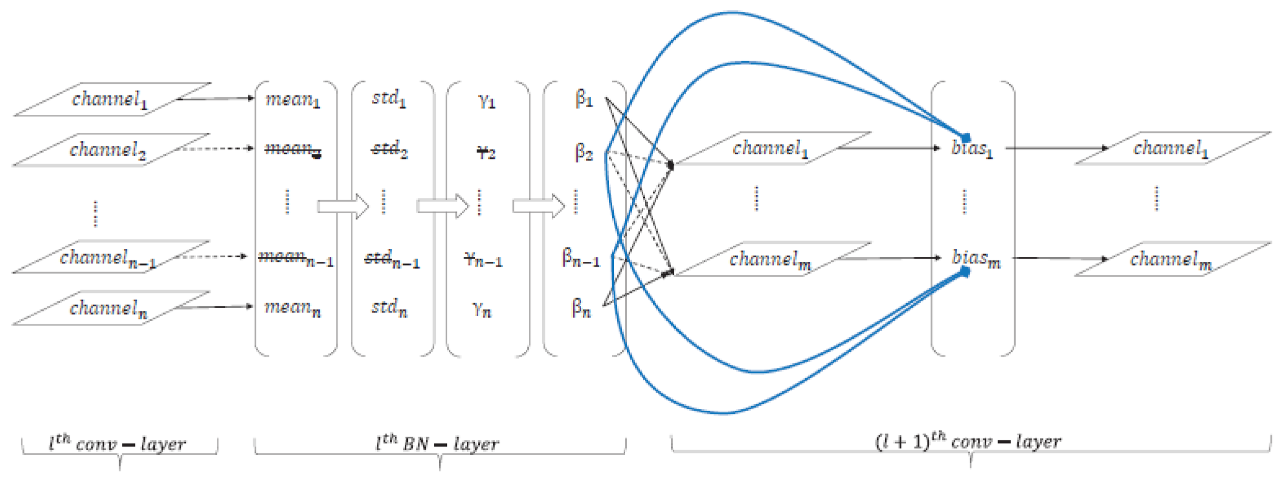
$$x^{l+1} \approx \sigma_{\neg I(\beta^{l})}\left(w^{l+1} * x^{l} + b^{l+1}_{new}\right)(3)$$

2.If the subsequent convolution layer is followed by a BN layer, then the convolution layer’s bias doesn’t work. Therefor, we absorb into running mean of next BN layer:

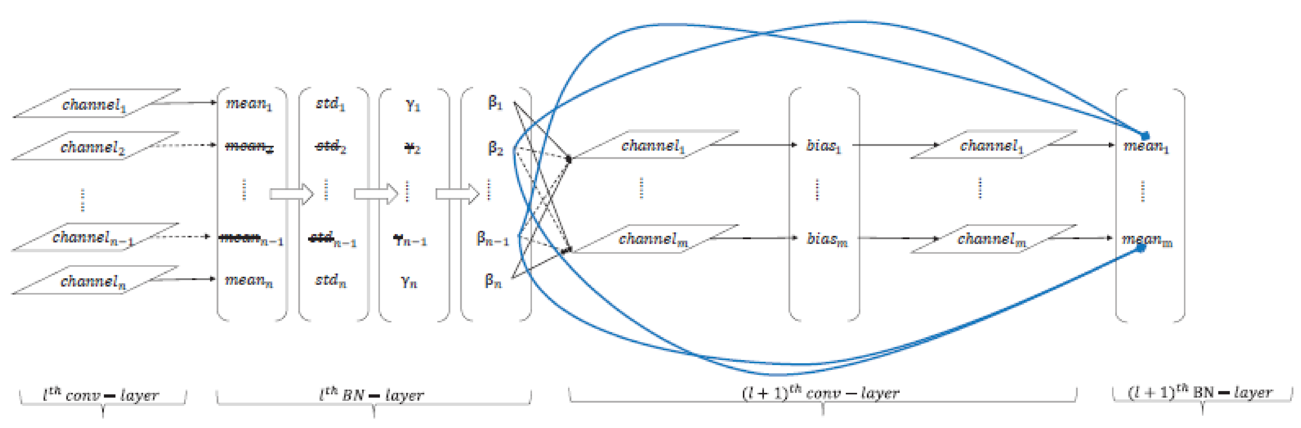
$$x^{l+2} = \sigma(\gamma^{l+2} \cdot BN_{u^{l+2}}\left(w^{l+1} * x^{l} + b^{l+1}\right) + \beta^{l+2})(1)$$

$$\mu^{l+2}_{new} = \mu^{l+2} - \sum^{a}(I(\beta^{l}) \cdot \sigma(\beta^{l}) \sum^{i}\sum^{j} w^{l+1}_{:,a,i,j}\ (2)$$

$$x^{l+2} \approx \sigma_{\neg I(\beta^{l})}\left(\gamma^{l+2} \cdot BN_{u^{l+2}}\left(w^{l+1} * x^{l} + b^{l+1}\right) + \beta^{l+2}\right)(3)$$



(1) If the subsequent convolution layer is followed by a non-BN layer



(2) If the subsequent convolution layer is followed by a BN layer

Knowledge Distillation based Fine-tuning

In fine-tuning process, we use knowledge distillation to help the pruned model restore the accuracy as much as possible.

Experimental Results

Table 2: Performance on CIFAR-10

	VGG16				ResNet164			
	ρ (%)	Error (%)	Params (M)	FLOPs ($\times 10^8$)	ρ (%)	Error (%)	Params (M)	FLOPs ($\times 10^8$)
Liu [6]	0	6.34	20.04	7.97	0	5.42	1.70	4.99
	70	6.20	**2.30**	3.91	40	5.08	1.44	3.81
					60	5.27	1.10	2.75
Ours	0	6.33	20.04	7.97	0	5.32	1.72	5.00
	70	**5.96**	2.32	**3.83**	40	**4.96**	1.00	2.28
					60	5.07	**0.61**	**1.33**

Table 3: Performance on CIFAR-100

	VGG16				ResNet164			
	ρ (%)	Error (%)	Params (M)	FLOPs ($\times 10^8$)	ρ (%)	Error (%)	Params (M)	FLOPs ($\times 10^8$)
Liu [6]	0	26.74	20.08	7.97	0	23.37	1.73	4.99
	50	26.52	5.00	5.01	40	22.87	1.46	3.33
					60	23.91	1.21	2.47
Ours	0	26.42	20.08	7.97	0	23.37	1.74	5.00
	50	**25.87**	**4.94**	**4.07**	40	**22.51**	1.02	2.09
					60	23.31	**0.63**	**1.32**

Table 4: Comparison with other channel-level pruning on CIFAR-10 without accuracy loss

	VGG16			
Method	Pruning Ratio or Policy	Error (%)	Params (M)	Speedup
Li [15]	0	6.75	15.00	1×
	Pruned-A	6.60	5.40	1.52×
Luo [17] (our impl.)	0	6.31	14.99	1×
	50	6.24	4.09	2.04×
Ours	65	**6.20**	**1.98**	**2.24**×

	ResNet56			
Method	Pruning Ratio or Policy	Error (%)	Params (M)	Speedup
Li [15]	0	6.96	0.85	1×
	Pruned-B	6.94	0.73	1.37×
Luo [17] (our impl.)	0	6.00	0.86	1×
	40	5.99	0.51	1.62×
Ours	50	**5.96**	**0.42**	**1.99**×

Table 5: Performance on LFW

Model	ρ (%)	Error (%)	Parameters (M)	FLOPs ($\times 10^8$)
SphereFace20	0	1.00	22.68	35.04
	30	1.20	19.55	24.23
	40	1.27	18.34	21.53
	50	1.33	17.23	18.65

Conclusion

We have proposed a Batch Normalization based channel-level pruning method with local normalization and selection, ablation based transfer and knowledge distillation based fine-tuning. Our approach firstly normalize the values of at each BN layer and then prune the channels whose values are smaller than a layer adaptive threshold.

After channel pruning, ablation based transfer, and knowledge distillation based fine tuning are also applied to further improve the performance of pruned model. The experimental results on CIFAR-10, CIFAR-100 and LFW clearly suggest that our approach can achieve much efficient pruning in terms of reduction in parameters and FLOPs. Take ResNet for example, when pruning ratio is set as 60%, the sizes of our pruned model for CIFAR-10 and CIFAR-100 are 0.61M and 0.63M, respectively, which are roughly half the size of the models pruned by Liu’s approach. Similar conclusions can also be suggested for FLOPs. Compared to other channal-level pruning [15] [17] without accuracy loss on VGG16 and ResNet56, our method achieves 86.79% and 51.46% reduction in parameters while 55.25% and 49.87% reduction in FLOPs, respectively.

Contact

Liu Yuan: liuyuan20162@email.szu.edu.cn

Xi Jia: jiaxi@email.szu.edu.cn

Linlin Shen: llshen@szu.edu.cn

ICANN19

28th International Conference on Artificial Neural Networks