Discrimination-aware Channel Pruning for Deep Neural Networks

Zhuangwei Zhuang*, Mingkui Tan*†, Bohan Zhuang*, Jing Liu*, Yong Guo, Qingyao Wu, Junzhou Huang, Jinhui Zhu†

Neural Information Processing Systems Foundation

Background and Motivation

Channel pruning reduces the model size and speeds up the inference by **removing redundant channels** directly. Existing methods include:

- **Training-from-scratch methods**: select channels to minimize the **cross-entropy loss** with sparsity regularization [1].
- **Reconstruction-based methods**: select channels to minimize the **reconstruction error** of feature maps between the pruned model and a pre-trained model [2].

Limitations of existing channel pruning methods:

- **Training-from-scratch methods**: are **difficult to converge**.
- **Reconstruction-based methods**: ignore the **discriminative power**.
- Both methods result in **apparent drop** in accuracy.

Our solution: propose a **discrimination-aware channel pruning (DCP)** scheme to choose channels with **true discriminative power**.

Contributions

- We propose a **discriminative-aware channel pruning (DCP)** scheme to choose the channels with **true discriminative power**.
- We formulate the channel selection problem as an **$\ell_{2,0}$-norm constrained optimization problem** and propose a **greedy method** to solve the resultant optimization problem using SGD.
- Extensive experiments demonstrate the effectiveness of DCP.

Problem Definition

- **Channel Pruning** prunes those redundant channels in $\mathbf{W}$ to save the model size and accelerate the inference speed in Eq. (1)

$$\mathbf{O}_{i,j,:,:} = \sum_{k=1}^{c} \mathbf{X}_{i,k,:,:} * \mathbf{W}_{j,k,:,:}, \quad (1)$$

where $\mathbf{X}_{i,k,:,:}$ is the input feature map, $\mathbf{W}_{j,k,:,:}$ denotes the parameters and $\mathbf{O}_{i,j,:,:}$ is the output feature map.

- **$\ell_{2,0}$-norm constraint** on $\mathbf{W}$ to choose channels

$$||\mathbf{W}||_{2,0} = \sum_{k=1}^{c} \Omega(\sum_{j=1}^{n} ||\mathbf{W}_{j,k,:,:}||_F) \leq \kappa_l, \quad (2)$$

where $\Omega(a) = 1$ if $a \neq 0$, $\Omega(a) = 0$ if $a = 0$, $||\cdot||_F$ represents the Frobenius norm, and κ_l denotes the desired number of channels at the layer l. Given a predefined pruning rate $\eta \in (0, 1)$, we can calculate $\kappa_l = \lceil \eta c \rceil$.

DCP Architecture

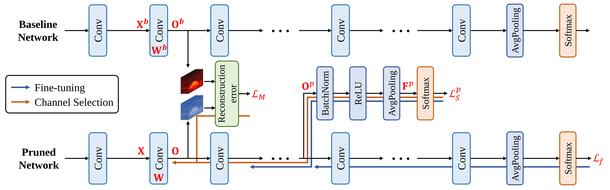


Figure 1: The architecture of discrimination-aware channel pruning.

- We insert P **discrimination-aware losses $\{\mathcal{L}_S^p\}_{p=1}^P$ (cross-entropy loss)** evenly and split the network into $P+1$ stages.
- For each stage, we do fine-tuning using $\mathcal{L}_S^p$ and $\mathcal{L}_f$ to improve the **discriminative power** of intermediate layers.
- We conduct **discrimination-aware channel pruning** for each layer involved in the considered stage by solving following problem:

$$\min_{\mathbf{W}} \quad \mathcal{L}(\mathbf{W}) = \mathcal{L}_M(\mathbf{W}) + \lambda \mathcal{L}_S^p(\mathbf{W}), \quad \text{s.t. } ||\mathbf{W}||_{\mathbf{2,0}} \leq \kappa_l, \quad (3)$$

where λ balances the two terms, $\mathbf{W}$ is the model parameters of a considered layer, $\mathcal{L}_M(\mathbf{W})$ is the **reconstruction error**, and $\mathcal{L}_S^p(\mathbf{W})$ is the **cross-entropy loss**.

Convexity of the Loss Function

Proposition 1. (Convexity of the loss function) *Let $\mathbf{W}$ be the model parameters of a considered layer. Given the mean square loss and the cross-entropy loss, then the joint loss function $\mathcal{L}(\mathbf{W})$ is **convex** w.r.t. $\mathbf{W}$.*

Discrimination-aware Channel Pruning

Algorithm 1 Discrimination-aware channel pruning (DCP)

Input: Pre-trained model M, training data $\{\mathbf{x}_i, y_i\}_{i=1}^N$, and parameters $\{\kappa_l\}_{l=1}^L$.
for $p \in \{1, ..., P+1\}$ **do**
  Construct loss $\mathcal{L}_S^p$ to layer L_p as in Figure 1.
  Learn $\boldsymbol{\theta}$ and **Fine-tune** M with $\mathcal{L}_S^p$ and $\mathcal{L}_f$.
  for $l \in \{L_{p-1}+1, ..., L_p\}$ **do**
    Do **Channel Selection** for layer l using Algorithm 2.
  end for
end for

- DCP introduces P **discrimination-aware losses** and updates the model to increase the **discriminative power** of intermediate layers.
- DCP performs channel pruning with $(P+1)$ stages.

Greedy Algorithm

Algorithm 2 Greedy algorithm for channel selection

Input: Training data, model M, parameters κ_l, and ϵ.
Output: Selected channel subset $\mathcal{A}$ and model parameters $\mathbf{W}_{\mathcal{A}}$.
Initialize $\mathcal{A} \leftarrow \emptyset$, and $t = 0$.
while (stopping conditions are not achieved) **do**
  Compute gradients of $\mathcal{L}$ w.r.t. $\mathbf{W}$: $\mathbf{G} = \partial\mathcal{L}/\partial\mathbf{W}$.
  Find the channel $k = \arg\max_{j \notin \mathcal{A}} \{||\mathbf{G}_j||_F\}$.
  Let $\mathcal{A} \leftarrow \mathcal{A} \cup \{k\}$.
  Solve Problem (4) to update $\mathbf{W}_{\mathcal{A}}$.
  Let $t \leftarrow t + 1$.
end while

- Instead of solving problem (3), DCP uses a **greedy algorithm** to optimize $\mathbf{W}$ w.r.t. the selected channels by minimizing:

$$\min_{\mathbf{W}} \quad \mathcal{L}(\mathbf{W}), \quad \text{s.t.} \quad \mathbf{W}_{\mathcal{A}^c} = \mathbf{0}, \quad (4)$$

where $\mathbf{W}_{\mathcal{A}^c}$ denotes the submatrix indexed by $\mathcal{A}^c$ which is the complementary set of $\mathcal{A}$.

Stopping Conditions

- Given a predefined parameter κ_l, Algorithm 2 will be stopped if $||\mathbf{W}||_{2,0} > \kappa_l$.
- Since $\mathcal{L}$ is **convex**, $\mathcal{L}(\mathbf{W}^t)$ will **monotonically decrease** with iteration index t in Algorithm 2. The number of selected channels can be **automatically determined** by following stopping condition:

$$|\mathcal{L}(\mathbf{W}^{t-1}) - \mathcal{L}(\mathbf{W}^t)| / \mathcal{L}(\mathbf{W}^0) \leq \epsilon,$$

where $\mathcal{L}(\mathbf{W}^t)$ is the joint loss function with iteration t and ϵ is a tolerance value.

Results on CIFAR-10 and ILSVRC-12

Table 1: Comparisons on CIFAR-10. "-" denotes that the results are not reported.

Model		ThiNet	CP	Sliming	WM	WM+	Random DCP	DCP	DCP-Adapt
VGGNet (Baseline 6.01%)	#Param. ↓	1.92×	1.92×	8.71×	1.92×	1.92×	1.92×	1.92×	**15.58×**
	#FLOPs ↓	2.00×	2.00×	2.04×	2.00×	2.00×	2.00×	2.00×	**2.86×**
	Err. gap (%)	+0.14	+0.32	+0.19	+0.38	+0.11	+0.14	**-0.17**	**-0.58**
ResNet-56 (Baseline 6.20%)	#Param. ↓	1.97×	-	-	1.97×	1.97×	1.97×	1.97×	**3.37×**
	#FLOPs ↓	1.99×	2×	-	1.99×	1.99×	1.99×	1.99×	1.89×
	Err. gap (%)	+0.82	+1.0	-	+0.56	+0.45	+0.63	**+0.31**	**-0.01**

Table 2: Comparisons on ILSVRC-12. The **top-1 and top-5 error (%)** of the pre-trained model are **23.99 and 7.07**, respectively. "-" denotes that the results are not reported.

Model		ThiNet	CP	WM	WM+	DCP
ResNet-50	#Param. ↓	2.06×	-	2.06×	2.06×	2.06×
	#FLOPs ↓	2.25×	2×	2.25×	2.25×	2.25×
	Top-1 gap (%)	+1.87	-	+2.81	+2.41	**+1.06**
	Top-5 gap (%)	+1.12	+1.40	+1.62	+1.28	**+0.61**

- DCP achieves the best performance under the same acceleration rate.

Exploring Pruning Rate and λ

Table 4: Comparisons on ResNet-18 and ResNet-50 with different pruning rates. We report the top-1 and top-5 error (%) on ILSVRC-12.

Network	η	Top-1/Top5 err.
ResNet-18	0% (baseline)	30.36/11.02
	30%	**30.79/11.14**
	50%	32.65/12.40
	70%	35.88/14.32
ResNet-50	0% (baseline)	23.99/7.07
	30%	**23.60/6.93**
	50%	25.05/7.68
	70%	27.25/8.87

- The performance of the pruned models go worse with the increase of pruning rate.
- ResNet-50 with pruning rate of 30% outperforms the pre-trained model.

Table 5: Pruning results on ResNet-56 with different λ on CIFAR-10.

λ	Training err.	Testing err.
0 ($\mathcal{L}_M$ only)	7.96	12.24
0.001	7.61	11.89
0.005	6.86	11.24
0.01	6.36	11.00
0.05	4.18	9.74
0.1	3.43	8.87
0.5	2.17	8.11
1.0	**2.10**	**7.84**
1.0 ($\mathcal{L}_S$ only)	2.82	8.28

- The performance of the pruned model improves with increasing λ.
- Both the reconstruction error and the cross-entropy loss contribute to better performance.

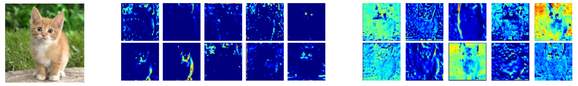
Effect of the Stopping Condition

Table 5: Effect of ϵ for channel selection over VGGNet on CIFAR-10.

Loss	ϵ	Testing err. (%)	#Param. ↓	#FLOPs ↓
$\mathcal{L}$	0.1	12.68	**152.25×**	27.39×
	0.01	6.63	31.28×	5.35×
	0.001	**5.43**	15.58×	2.86×

- A smaller ϵ leads to better performance of the pruned model.

Visualization of Feature Maps



(a) Input image (b) Feature maps of the pruned channels (c) Feature maps of the selected channels

- Feature maps of the pruned channels are **less informative**.

References

[1] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, 2017.

[2] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *ICCV*, 2017.

Contact Information

- Correspondence to: Prof. Mingkui Tan
- Email: mingkuitan@scut.edu.cn
- School: South China University of Technology