Supplementary Material: ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression

Jian-Hao Luo[1], Jianxin Wu[1], and Weiyao Lin[2]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]Shanghai Jiao Tong University, Shanghai, China
luojh@lamda.nju.edu.cn, wujx2001@nju.edu.cn, wylin@sjtu.edu.cn

Appendix A: Equivalence of Eq. 5 and Eq. 6

Note that $S \cup T = \{1, 2, \ldots, C\}$ and $S \cap T = \emptyset$, Eq. 3 can be rewritten as:

$$\hat{y} = \sum_{j \in S} \hat{x}_j + \sum_{j \in T} \hat{x}_j. \tag{1}$$

Hence, for each training examples $\{(\hat{\mathbf{x}}_i, \hat{y}_i)\}$, we have

$$\hat{y}_i - \sum_{j \in S} \hat{\mathbf{x}}_{i,j} = \sum_{j \in T} \hat{\mathbf{x}}_{i,j}. \tag{2}$$

Which means Eq. 5 is equivalent to Eq. 6.

Appendix B: Sparse coding solution

If we reformulate the optimization objective function in a matrix form, it becomes

$$\arg\min \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2, \tag{3}$$

where $\mathbf{y} \in \mathbb{R}^{m \times 1}$ and $\mathbf{X} \in \mathbb{R}^{m \times C}$ denote the m training examples. $\mathbf{w} \in \mathbb{R}^{C \times 1}$ is a sparse vector, whose entries are zeros except those preserved channels. According to the theory of compressed sensing, the sparse solution can be obtained through the following ℓ^1-minimization problem:

$$\hat{\mathbf{w}} = \arg\min \|\mathbf{w}\|_1, \quad \text{s.t.} \quad \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2 \leq \epsilon. \tag{4}$$

This optimization problem can be efficiently solved using the homotopy method. And the sparsity of $\mathbf{w}$ can be controlled by some hyper-parameters. Hence, we use the binary search strategy to obtain a sparse vector with exactly $C \times r$ nonzero items. Based on $\mathbf{w}$, we remove those channels associated with the zeros entries, and regard the nonzero items as the scaling factors which is introduced in Section 3.2.3.

We empirically compare the performance of sparse coding on VGG-16 using the ThiNet-Conv pruning framework. As shown in Table 1, the proposed simple greedy approach is slightly better than sparse coding.

Further improvement would be obtained if we can solve the optimization problem more efficiently, which should be explored in the future.

Table 1. Performance comparison between sparse coding and greedy solution using ThiNet-Conv pruning framework.

Strategy	Top-1	Top-5
ThiNet-Conv with sparse coding	69.34%	89.27%
ThiNet-Conv with greedy solution	**69.80%**	**89.53%**

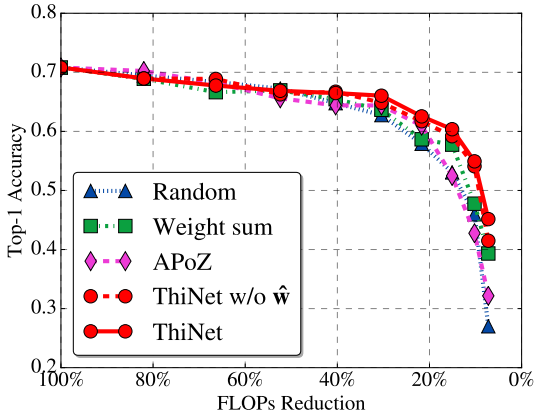

Figure 1. Performance comparison of different channel selection methods: the VGG-16-GAP model pruned on Indoor-67 with different compression rates. (This figure is best viewed in color and zoomed in.)

Appendix C: Comparison on Indoor-67

In Section 4.1, we have revealed the performance of different channel selection methods on CUB-200. Here, we would present the comparison results on Indoor-67, which is shown in Figure 1.

Unlike CUB-200, different heuristic criteria have similar performance on this dataset. However, ThiNet can still achieve significantly and consistently higher accuracy, showing much stronger generalization ability.