A: FSKD with different # BCD iterations

In our FSKD algorithm, we can apply the block-coordinate descent for several iterations. However, we do not observe noticeable gains for the iteration number \( T > 1 \) over \( T = 1 \) as shown in Figure 1, so that we set \( T = 1 \) in all our following experiments. This may be due to the reason that in each iteration, the sub-problem is a linear optimization problem so that we can find exact minimization, which is consistent with the finding by [3].

![Figure 1: Accuracy vs #iterations of FSKD on CIFAR-10. Student-net is Prune-B by filter pruning.](image)

B: FSKD-BCD vs. FSKD-SGD

<table>
<thead>
<tr>
<th>VGG-16</th>
<th>Acc. (%)</th>
<th>#Samples</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prune-B + FSKD-BCD</td>
<td>90.17</td>
<td>100</td>
<td>3.7</td>
</tr>
<tr>
<td>Prune-B + FSKD-SGD</td>
<td>89.41</td>
<td>100</td>
<td>18.4</td>
</tr>
<tr>
<td>Prune-B + FSKD-BCD</td>
<td>91.21</td>
<td>500</td>
<td>19.3</td>
</tr>
<tr>
<td>Prune-B + FSKD-SGD</td>
<td>90.76</td>
<td>500</td>
<td>50.5</td>
</tr>
</tbody>
</table>

Table 1: Performance comparison between FSKD-BCD and FSKD-SGD by student-nets from filter pruning of VGG-16 with pruning scheme B on CIFAR-10.

In this section, we compared two FSKD optimization algorithms: FSKD-BCD uses the BCD algorithm on block-level and FSKD-SGD optimizes the loss all together with the standard SGD algorithm. We valuate both methods on VGG-16 models trained on CIFAR-10 and compressed using filter pruning (prune-B). As shown in Table 1, FSKD-BCD achieves better accuracy than FSKD-SGD while significantly improves time efficiency.

C: Proof of Theorem 1

Proof. When \( W \) is a point-wise convolution with tensor \( W \in \mathbb{R}^{n_a \times n_i \times 1 \times 1} \), both \( W \) and \( Q \) are degraded into matrix form. It is obvious that when condition \( c1 \sim c3 \) satisfied, the theorem holds with \( W' = Q \ast W \) in this case, where \( \ast \) indicates matrix multiplication.

When \( W \) is a regular convolution with tensor \( W \in \mathbb{R}^{n_a \times n_i \times k \times k} \), the proof is non-trivial. Fortunately, recent work on network decoupling [1] presents an important theoretical result as the basis of our derivation.

**Lemma 1.** Regular convolution can be exactly expanded to a sum of several depth-wise separable convolutions. Formally, \( \forall W \in \mathbb{R}^{n_a \times n_i \times k \times k} \), \( \exists \{P_k, D_k\}_{k=1}^K \), where \( P_k \in \mathbb{R}^{n_a \times n_i \times 1 \times 1} \), \( D_k \in \mathbb{R}^{1 \times n_i \times k \times k} \), \( s.t. \quad (a)K \leq k^2; \)

\[
(b) W = \sum_{k=1}^K P_k \circ D_k, \tag{1}
\]

where \( \circ \) is the compound operation, which means performing \( D_k \) before \( P_k \).

Please refer to [1] for the details of proof for this Lemma. When \( W \) is applied to an input patch \( x \in \mathbb{R}^{n_i \times k \times k} \), we obtain a response vector \( y \in \mathbb{R}^{n_o} \) as

\[
y = W \odot x, \tag{2}
\]

where \( y_o = \sum_{i=1}^{n_o} W_{o,i} \odot x_i, o \in \{n_o\}, i \in \{n_i\}, \) and \( \odot \) here means convolution operation. \( W_{o,i} = W[o,i, \ldots] \) is a tensor slice along the i-th input and o-th output channels, \( x_i = x[i, \ldots] \) is a tensor slice along the i-th channel of 3D tensor \( x \).
When point-wise convolution $Q$ is added after $Q$ without non-linear activation between them, we have

$$y' = Q \circ (W \otimes x).$$

With Lemma-1, we have

$$y' = (Q\circ \sum_{k=1}^{K} P_k \circ D_k) \otimes x = (\sum_{k=1}^{K} Q \circ P_k \circ D_k) \otimes x$$

As both $Q$ and $P_k$ are degraded into matrix form, denoting $P'_k = Q \circ P_k$ and $W' = \sum_{k=1}^{K} P'_k \circ D_k$, we have $y' = W' \circ x$. This proves the case when $W$ is a regular convolution.

**D: Algorithm for iterative pruning and FSKD**

Algorithm-1 describes the iteratively pruning and FSKD procedure to achieve extremely compression rate based on [2, 4, 5].

**Algorithm 1** Iteratively pruning and FSKD Algorithm

1. prune-ratio-list $\{r_k\}_{k=1}^{K}$, number of iterations $T$
2. for $t = 1 : T$
3. $q_{max} = 0$
4. for $k = 1 : K$
5. Prune $s$ with ratio $r_k$ to obtain student-net $t$
6. Run FSKD with $s$, $t$ and $\{X_i\}_{i=1}^{N}$, output $s'$
7. Evaluation $s'$ on validation set to obtain score $q_k$
8. if $q_k > q_{max}$ then
9. $q_{max} = q_k$
10. $s_{max} = s'$
11. end if
12. end for
13. Update teacher $t = s_{max}$
14. end for

**E: Training only PW conv-layer is enough**

People may challenge that learning $1 \times 1$-conv may loss representation power and ask why the added $1 \times 1$ convolution works so well with such few samples. According to the network decoupling theory (Lemma-1), any regular conv-layer could be decomposed into a sum of depthwise separable blocks, where each depthwise separable block consists of a depthwise (DW) convolution (for spatial correlation modeling) followed by a pointwise (PW) convolution (for cross-channel correlation modeling). The added $1 \times 1$ conv-layer is absorbed/merged into the previous PW layer finally. The decoupling yields that the number of parameters in PW-layer occupies most ($> 80\%$) parameters of the whole network. We argue that learning only $1 \times 1$-conv is still very powerful, and make a **bold hypothesis** that PW conv-layer is more critical for performance than DW conv-layer. To verify this hypothesis, we conduct experiments on VGG16 and ResNet50 on CIFAR-10 and CIFAR-100 under below different settings.

(1) We train the network from random initialization with 160 epochs with learning-rate decay 1/10 at 80, 120 epochs from 0.01 to 0.0001.

(2) We start from a random initialized network (VGG16 or ResNet50), and do full rank decoupling ($K = k^2$ in Eq. 1) so that channels in DW layers are orthogonal, and PW layers are still fully random. Note that Lemma-1 ensures the network before and after decoupling are equivalent (i.e., able to transfer back and force from each other). We keep all the DW-layers fixed (with orthogonal basis from random data), and train only the PW layers with 160 epochs. We denote this scheme as ND-1*1.

<table>
<thead>
<tr>
<th>Model</th>
<th>CIFAR-10(%)</th>
<th>CIFAR-100(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
<td>93.00</td>
<td>73.35</td>
</tr>
<tr>
<td>VGG-16 (ND-1*1)</td>
<td>93.91</td>
<td>73.61</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>92.64</td>
<td>69.93</td>
</tr>
<tr>
<td>ResNet-50 (ND-1*1)</td>
<td>93.51</td>
<td>70.83</td>
</tr>
</tbody>
</table>

Table 2: Results by two schemes (1) full training (2) only training pointwise conv-layers (ND-1*1).
Note that except the setting explicitly described, all the other configurations (including training epochs, hyperparameters, hardware platform, etc) are kept the same on both experimental cases. Table 2 lists the experimental results on these two cases on both datasets with two different network structures. It is obvious that the 2nd case (ND-1*1) clearly outperforms the 1st case. Figure 2 further illustrates the test accuracy at different training epochs, which clearly shows that the 2nd case (ND-1*1) converges faster and better than the 1st case. This experiment verifies our hypothesis that when keeping DW channels orthogonal, training only the pointwise (1 × 1) conv-layer is accurate enough, or even better than training all the parameters together.

**F: Filter Visualization**

In this section, we try to answer why FSKD works so well that it can provide almost the same results as that of fine-tuning with full training set. We conduct experiments based on VGG-13 on CIFAR-10. For a given VGG-13 network, We first decouple a conv-layer to obtain one DW conv-layer and one PW conv-layer, as is done in network decoupling [1]. Then we visualize the PW conv-layer of the decoupled layer. For simplicity, we only visualize the PW conv-layer of the first decoupled layer. We do the visualization on three VGG-13 network with different parameters:

1. Initialize the VGG-13 network with the MSRA initialization (Figure 3 left).
2. Run SGD based fine-tuning on 500 samples for VGG-13 with random initialization until convergence (Figure 3 middle).
3. Run FSKD on 500 samples for VGG-13 with SGD based initialization (Figure 3 right). The teacher network is also a VGG-13 trained on full CIFAR-10 training set.

It clearly shows that the PW conv-layer before fine-tuning is fairly random on the value range, the one after fine-tuning is less random, while the one after FSKD further starts to show some regular patterns, which demonstrates that FSKD can distill the knowledge from the teacher-net to student-net effectively with few samples.

**References**


