HFP: Hardware-Aware Filter Pruning for Deep Convolutional Neural Networks Acceleration

Fang Yu$^{1,2}$, Chuanqi Han$^{1,2}$, Pengcheng Wang$^{1,2}$, Ruoran Huang$^{1,2}$, Xi Huang$^1$, Li Cui$^1*$$^\textsuperscript{*}$

$^1$Institute of Computing Technology, Chinese Academy of Sciences, $^2$University of Chinese Academy of Sciences
Background

- **Burden of CNNs**
  - Computationally demanding and memory intensive
  - Burden to be deployed on the hardware devices

- **Benefit of filter pruning**
  - Reduces the FLOPs and storage usage
  - Accelerates the CNNs inference
Background

Filter pruning Compress and accelerate the CNNs
Background

- The majority of pruning approaches prune networks by defining the important filters or training the networks with a sparsity prior loss.
- However, these pruning methods cannot prune a network while respecting a actual budget on the target hardware, such as latency, power or energy.
- These works adopt hardware-agnostic metrics such as floating-point operations (FLOPs) to estimate the CNNs’ efficiency.
Hardware-aware Filter Pruning

- We propose a hardware-aware filter pruning (HFP) method which can directly control the latency of pruned networks on the hardware platform.
- In our method, we propose a greedy pruning criterion based on information gain to evaluate the filter importance, which efficiently simplifies the pruning optimization problem.
- We propose the Opti-Trim pruning framework, which can decrease the accuracy degradation of pruning process and accelerate the pruning procedure.
Problem formulation

- For classification task, to minimize the accuracy drop while meeting the budget of latency on hardware, we define the pruning problem as:

\[
k^* = \arg \min_k L_{CE}(Y, P(Y|X, \theta_k^+)) \quad \text{s.t.} \quad \text{LAT}(\theta_{k^*}^+) < \text{Bud},
\]

where \( L_{CE} \) is cross-entropy loss, \( \text{LAT}(\cdot) \) evaluates the actual latency of pruned network consumed on the hardware, and \( \text{Bud} \) is the budget about latency.
Greedily pruning via information gain

- The information gain (IG) of filter quantifies the influence of filter removal on class probability distribution of network output.

- The more information gain of a certain filter, the more information is gained by this filter.

- Filters with the minimum IG carry little information, whose removal will not incur much information loss.
Opti-Trim pruning framework

- To decrease the accuracy degradation of pruning process and accelerate the pruning procedure, we proposed Opti-Trim pruning framework.
  - Opti phase: fine-tune the pruned network using L1 group regularization and compute the IG of filters
  - Trim phase: prune filters, achieve the budget on hardware and tighten the resource constraint
  - The Opti and Trim phase alternately work m times.

```
Algorithm 1: Algorithm Description of HFP
Input: Pre-trained network: Θ; Desired budget: Bud;
       Iteration number: m; Training set: {X, Y}
Output: Pruned network: θ_k^+

/* Initialization */
1 Build up a lookup table on the target hardware;
2 Obtain the base latency B;
3 Obtain Δ = (B − Bud)/m;
/* Opti-Trim pruning framework */
4 for i ∈ [0, m] do
    /* Opti phase */
    foreach {x,y}∈{X,Y} do
        Fine-tune the remaining filters in the network via Eq. (9);
        Calculate the IG of filter via Eq. (6) or Eq. (7);
    end
    /* Trim phase */
    repeat
        Prune a filter with the minimum IG across all layers;
        Obtain the current latency LAT(θ_k^+) of pruned network via Eq. (8);
        until LAT(θ_k^+) < B − i * Δ;
    end
End?
```

Pre-trained model → Build up a lookup table → [Opti]: Fine-tune model and compute IG → [Trim]: Prune filters to meet the budget → Pruned model

loop for m times

tighten constraint

End?
# Experiment on VGG-16

## TABLE I

<table>
<thead>
<tr>
<th>Uniform Baselines</th>
<th>HFP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ratio</strong></td>
<td><strong>Accuracy</strong></td>
</tr>
<tr>
<td>1×</td>
<td>93.73%</td>
</tr>
<tr>
<td>0.75×</td>
<td>92.80%</td>
</tr>
<tr>
<td>0.5×</td>
<td>91.89%</td>
</tr>
<tr>
<td>0.25×</td>
<td>89.06%</td>
</tr>
</tbody>
</table>

Fig.1. Number of filters at each layer of pruned VGG-16 on CIFAR-10.
Experiment on ResNet

Fig. 2. Comparison with MIL [37], PFEC [14], SFP [16], FPGM [6] and uniform baselines varying different FLOPs reduction rates on CIFAR-10.

(a) Results of pruning ResNet-32
(b) Results of pruning ResNet-56
(c) Results of pruning ResNet-110
Thank you for your attention!