EE5142/921 U9360

高等積體電路設計 Advanced Integrated Circuit Design Homework 3

馬咏治 kane@access.ee.ntu.edu.tw
羅翊誠 alan@access.ee.ntu.edu.tw

Preliminaries:

Deep Learning models have become the dominant approach in several areas due to their high performance. Unfortunately, the size and hence computational requirements of operating such models can be considerably high. Therefore, this constitutes a limitation for deployment on memory and battery constrained devices such as mobile phones or embedded systems. To address these limitations, several pruning techniques is proposed for reducing the #(model parameters) and #Ops.

Global Pruning

A novel and simple pruning method that compresses neural networks is proposed by removing entire filters and neurons according to a global threshold across the network without any pre-calculation of layer sensitivity. All layers are considered simultaneously for pruning; considering layers simultaneously was first introduced by to prune individual weights and was referred as class-blind pruning.

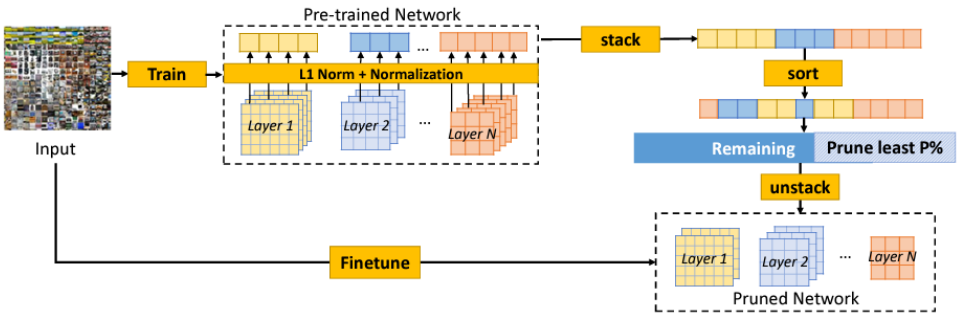


Figure 1

Overview of global pruning: as the first step, the L1-norm of each filter in each layer is calculated, then the calculated L1-norm of each filter is normalized according to its number of kernel weights, followed by stacking all normalized norms of all filters from all layers. As a last step, we perform sorting and pruning procedures of the filters corresponding to the least p% of normalized norms.

Layer-wise Pruning

Difference from global pruning, layer-wise pruning applies the same rate to each layer while global applies it on the whole network at once, which illustrated in Figure 2.

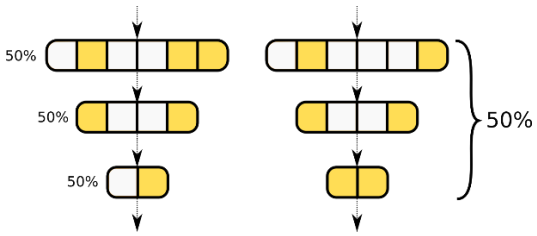


Figure 2

Unstructured Pruning and Structured Pruning (https://towardsdatascience.com/neural-network-pruning-101-af816aaea61**)**

Unstructured pruning is an intuitive way to removing parameters themselves. Directly pruning parameters has many advantages. First, it is simple, since replacing the value of their weight with zero, within the parameter tensors, is enough to prune a connection. The greatest advantage of pruning connections remains yet that they are the smallest, most fundamental elements of networks and, therefore, they are numerous enough to prune them in large quantities without impacting performance. However, this method presents a major, fatal drawback: most frameworks and hardware **cannot accelerate sparse matrices' computation**, meaning that no matter how many zeros you fill the parameter tensors with, it will not impact the actual cost of the network.

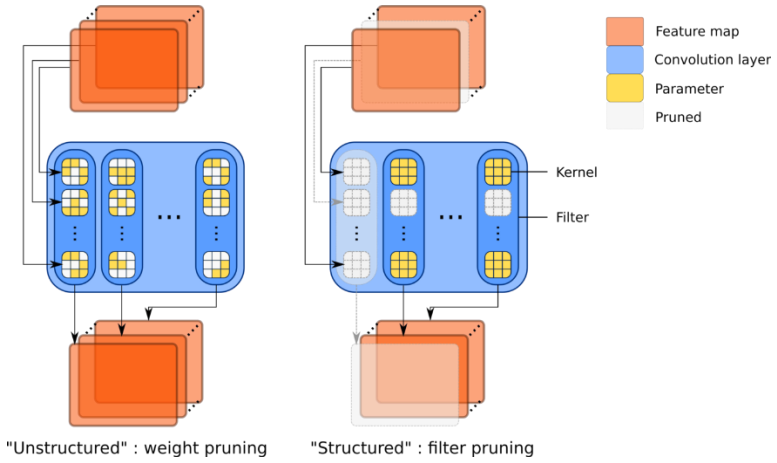


Figure 3: Difference between Unstructured Pruning and Structured Pruning

Structured pruning focused on pruning larger structures, such as whole neurons or, for its direct equivalent within the more modern deep convolutional networks, convolution filters. Not only does removing such structures result in sparse layers that can be directly instantiated as thinner ones, it does **help accelerate sparse matrices' computation**. However, structured pruning could bring some danger, which is altering the input and output dimensions of layers can lead to some discrepancies.

Code Description

- main.py
  - Pytorch CIFAR10 model training and pruning
- utils.py
  - Utilities function

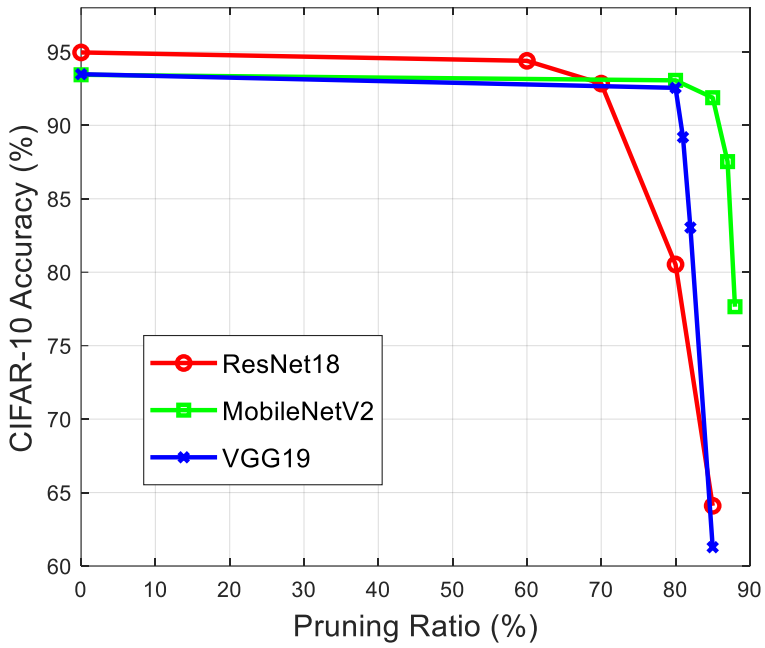
Please check following documentation for further information:

- https://pytorch.org/tutorials/intermediate/pruning_tutorial.html
- https://github.com/kuangliu/pytorch-cifar

Reference:

[1] Salama, A., Ostapenko, O., Klein, T., & Nabi, M. (2019). Pruning at a glance: Global neural pruning for model compression. *arXiv preprint arXiv:1912.00200*.

HW 3 Questions:



1. (45%)
   The **code is ready** for reproducing the figure above, please plot the **Compression Ratio-Accuracy** pareto frontier using Global Pruning with the following network:
   - ResNet18
   - MobileNetV2
   - VGG19

2. (55%)
   Try to manually fine-tune the pruning policy to improve the pareto frontier based on 1.), using Layer-wise Pruning.

3. (Bonus, 40%)
   Use **AMC*** to automatically find the pruning policy for 2.)
   * He, Y., Lin, J., Liu, Z., Wang, H., Li, L. J., & Han, S. (2018). Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 784-800).

Submission requirement:

1. The report should be merged as a single **pdf file** and **uploaded to NTU COOL**.
   Example of filename: AVLSI_HW3_d09943011.pdf
   Note that you have to replace d09943011 with your student ID number.
2. **Deadline: 2022/10/14 23:59**

The homework will be graded ONLY IF the filename of your submission is correct!