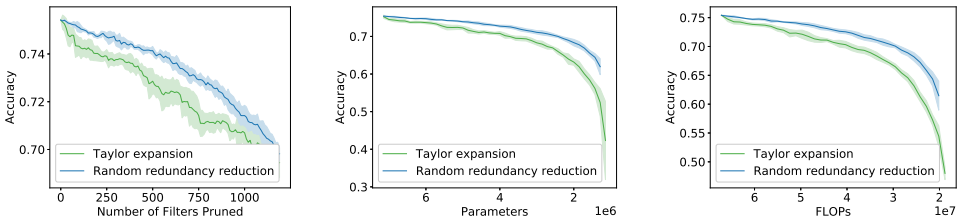
Appendix

A. Examples to validate the theoretical analysis

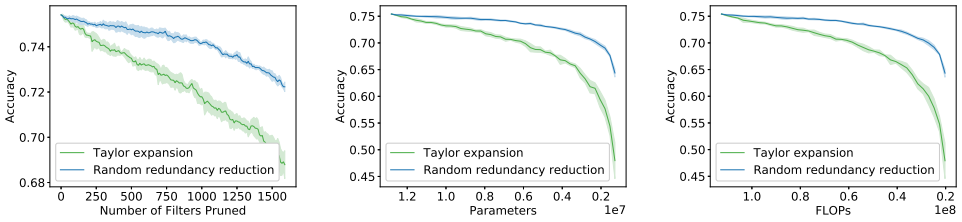
We first provide a number of intuitive examples to support our theoretical analysis in Section 3. For this purpose, we create redundant layers in several benchmark structures by manually increasing the number of filters in certain layers. We use a popular layer-adaptive filter ranking criterion, i.e., Taylor expansion, to prune the least important filters across all layers as the baseline. The Taylor expansion approach has been proved in their paper to have better performance than a number of other widely used filter ranking criteria, such as minimum weight, mean activation, and APoZ. We compare the performance of the baseline with randomly pruning filters in the layer with the most number of filters. We use progressive pruning for demonstration so that the performance of both approaches can be continuously observed. We use AlexNet on CIFAR-10 and VGG-16 on Birds-200 as the examples. For AlexNet, we increase the number of filters in the third and fourth convolutional layers from 382 and 256 to 1536 and 2048, respectively. For VGG-16, we increase the number of filters in the ninth and twelfth convolutional layers from 512 to 2048, respectively. We prune 10 and 50 filters for AlexNet and VGG-16 in each iteration and fine-tune the remaining networks for 500 mini-batches with a learning rate of $1e^{-4}$.

The performance comparison of both approaches are presented in Fig. 1 to Fig 4. It is obvious that for all four experiments, even simply randomly pruning filters from the layer with the most number of filters outperforms pruning the least important filters ranked by a popular criterion, i.e., Taylor expansion, across all layers. These results validate our theoretical claim that if a layer has much higher redundancy than others, randomly pruning filters in that layer outperforms pruning the least important filters across all layers.



(a) Number of filters (b) Parameters (c) FLOPs

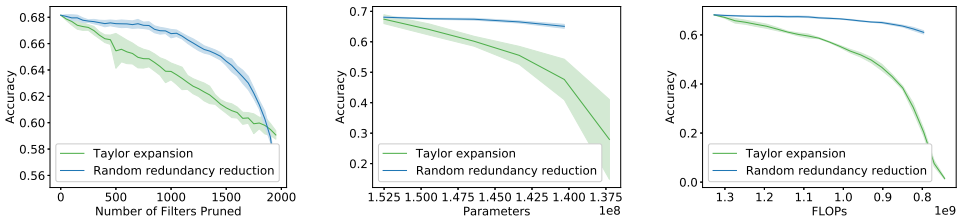
Figure 1: Performance comparison between random pruning in the layer with the most filters and pruning the least important filters ranked by Taylor expansion across all layers (AlexNet, number of filters in the third convolutional layer increased to 1536).



(a) Number of filters (b) Parameters (c) FLOPs

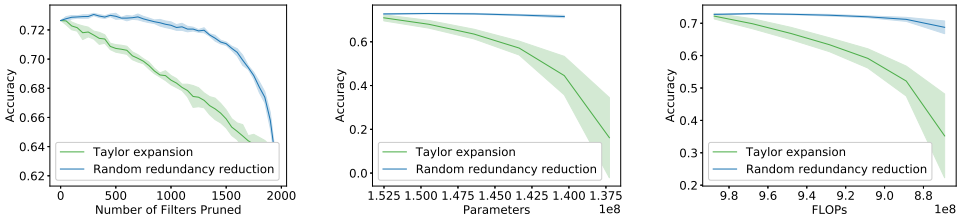
Figure 2: Performance comparison between random pruning in the layer with the most filters and pruning the least important filters ranked by Taylor expansion across all layers (AlexNet, number of filters in the fourth convolutional layer increased to 2048).

In the previous experiments, we manually increase the number of filters in certain layers of two benchmark architectures, i.e., AlexNet and VGG-16. Here we show that for some real benchmark architectures (without manual operation on the number of filters in the convolutional layers), our theoretical claim still holds.



(a) Number of filters (b) Parameters (c) FLOPs

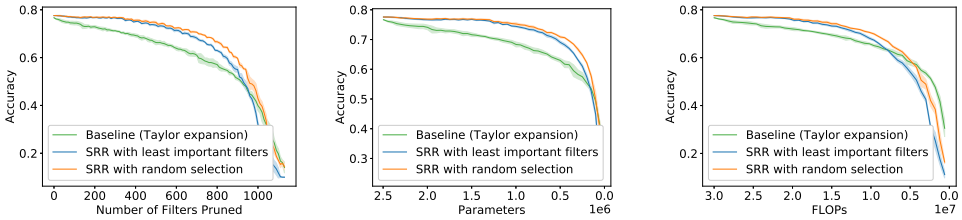
Figure 3: Performance comparison between random pruning in the layer with the most filters and pruning the least important filters ranked by Taylor expansion across all layers (VGG-16, number of filters in the ninth convolutional layer increased to 2048).



(a) Number of filters (b) Parameters (c) FLOPs

Figure 4: Performance comparison between random pruning in the layer with the most filters and pruning the least important filters ranked by Taylor expansion across all layers (VGG-16, number of filters in the twelfth convolutional layer increased to 2048).

We use AlexNet as an example, with the same pruning configurations as the previous experiments. We compare the following three strategies: (1) pruning the least important filters across all layers, ranked by Taylor expansion (baseline), (2) randomly pruning filters in the layer(s) with the most number of filters, and (3) pruning the least important filters in the layer(s) with the most number of filters, ranked by Taylor expansion. The results are presented in Fig. 5. We observe that both strategy (2) and (3) show better performance compared to the baseline. The performance are similar when pruning filters in the layer(s) with the most filters with different filter selection strategies (random or least important). These results are consistent with the conclusion in the theoretical analysis section, which indicates that $p_g \leq p_{\eta r} \leq p_{\underline{\eta}}$.



(a) Number of filters (b) Parameters (c) FLOPs

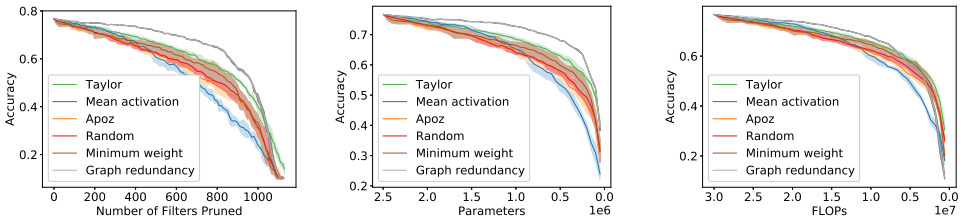
Figure 5: Performance comparison between pruning the least important filters ranked by Taylor expansion across all layers, random pruning in the layer with the most filters, and pruning the least important filters in the layer with the most filters.

For more sophisticated architectures that contain less redundancy, such as ResNet, using the number of filters as the criterion for redundancy measurement is not the best choice (see Section 6.4). However, with a well designed metric to measure the layer redundancy, we show that pruning the least important filters in the layer(s) with larger redundancy can still outperform pruning the least important filters across all layers with different filter selection strategies (see the experiment section).

B. Progressive pruning results

Because progressive pruning usually takes much more time than single-shot pruning, we validate the performance of our approach with AlexNet on the CIFAR-10 dataset. We re-implement all of the methods with the same configuration. We prune 10 filters in each iteration and fine-tune the remaining network for 500 mini-batches with a learning rate of 0.0001. During experiments, we discover that as more filters are removed, there exists less redundancy in the graphs. Slightly increasing γ results in better performance. In our experiments we set $\gamma_{i+1} = \gamma_i \times 1.01$ at each time step i. We plot the accuracy after each step of pruning and fine-tuning, in terms of the number of filters, parameters, and FLOPs pruned from the original network.

We first pre-train an AlexNet on the CIFAR-10 dataset and achieve an accuracy of 76.67%. Results in Fig. 6 show that our approach outperforms other methods when reducing the same number of filters or parameters. For example, when pruning 600 (out of 1152) filters from AlexNet, we achieve an accuracy of 69.85%, while the highest performance among the other method is 63.89% (Taylor). In terms of FLOPs, our approach also achieves better performance than other methods.



(a) Number of filters (b) Parameters (c) FLOPs

Figure 6: Progressive pruning results of AlexNet on CIFAR-10.