Article

PRP: Hardware-Oriented Pattern Replacement Pruning for Deep Neural Networks

Baoyu Chen [1,‡,*], Diyu Gui [2,‡] and Yidu Wu [3]

[1] College of Mechanical and Electrical Engineering, Hainan Vocational University of Science and Technology,Haikou 571126, China; cby222@126.com
[2] Yangtze Del Region Transformation Center for Adavanced Technological Achevements; dyangzhou22@126.com
[3] Hong Kong Baptist University; wuyidou@gmail.com
* Correspondence: cby222@126.com; Tel.: +86-1834-468-6114 (F.L.)
‡ These authors contributed equally to this work.

Abstract: For wide applications, previous works have developed many model pruning methods to help the deployment of convolutional neural networks (CNNs) in resource-limited platforms. However, the widely unstructured and pattern pruning schemes need extra hardware support to be utilized, while structured pruning ones suffer from accuracy degradation for modifications in network architectures. Thus, we propose a hardware-oriented pattern replacement pruning (PRP) method to compress CNNs considering both compression ratio, model accuracy, and implementation complexity. Compared with previous pruning methods, we have two main advantages: (1) Hardware-friendly kernel replacement: Instead of pruning kernels directly, the PRP approach replaces part of computations with more compact operators originating from the backbone. It compress the network without squeezing the network architecture or introducing any extra operations, leading to better model accuracy and lower implementation complexity compared to prior pruning methods. (2) AutoML-methods for better compression: The meta-learning based neural architecture search (NAS) is introduced into the PRP approach for searching optimal replacement patterns for each convolutional layer, which is conducted by the develop SearchNet. It could enumerate all pruned candidates and evaluate their performances without retraining the large network. To prove the effectiveness, we perform the PRP approach on widely-used ResNet. With a similar accuracy loss, the PRP approach have better compression compared to prior filter pruning algorithms on both CIFAR-10 and ILSVRC-2012 data-sets. Besides, we evaluate the PRP-ResNet on an accelerator supporting sparse CNNs. The proposed PRP approach can achieve better energy efficiency compared to other pruned ResNets with similar sparsity.

Keywords: Deep neural networks (DNNs); Model Compression; Meta Learning; Pruning, Accelerator

Citation: Chen, B.; Gui, D.; Wu, Y. Title. *Journal Not Specified* **2023**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. Introduction

Convolutional neural networks (CNNs) have been extensively applied in multiple computer vision tasks [1–3]. To increase the accuracy, model sizes of CNNs also enlarge drastically, leading to growing computation and storage complexities. It is hard to deploy large CNNs in resource-limited platforms. Pruning schemes, including unstructured pruning [4,5] and structured pruning, is an effective method for model compression. The structured pruning can further be classified into filter pruning [6] and pattern pruning [7,8].

However, the current pruning schemes have different disadvantages in various application scenarios. Unstructured pruning schemes can drastically reduce parameters of the model, but the irregularity of pruned weights harms the performance of hardware implementations, and makes it hard to transfer the reduction of parameters into the processing speedup of algorithms. Filter pruning schemes is able to reduce the model size in a regular way, but the modification of network architecture may lead to unacceptable

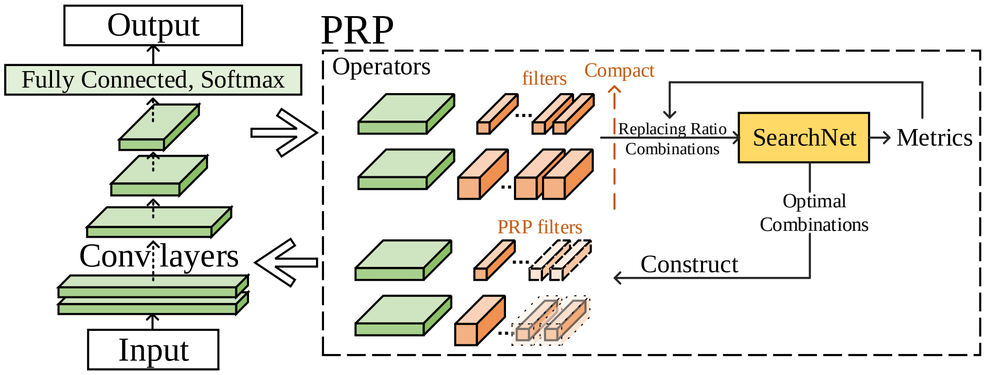


Figure 1. Pattern Replacement Pruning Approach. *Conv layers* is the short for convolutional layers. The green boxes denotes activation of the convolutional layers, and the orange boxes represent filters. We mark the pruned filters as the orange dashed boxes. The proposed SearchNet takes random generated replacing ratio combinations as input, and evaluates the pruned candidates with pre-set metrics. Finally, the optimal combinations can be selected by the SearchNet and utilized to construct the PRP compressed network.

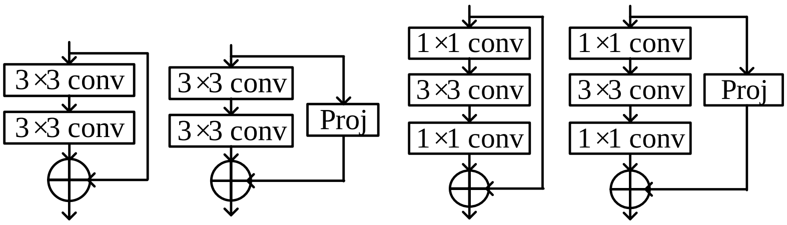
accuracy loss. To achieve the balance among hardware implementation complexity, model compression, and model accuracy, pattern pruning schemes are proposed. With a pre-set sparse pattern, the pruning scheme helps the algorithm deployment and maintains the model accuracy. However, previous pattern pruning algorithms are closely connected with hardware implementation, while laborious training procedures are introduced to attain tiny networks with complex sparse patterns.

Hence, we design a pattern replacement pruning (PRP) method, utilizing the available compact operators to replace part of original computations. With the pattern replacement, the new network becomes more compact and all computation patterns are selected from the original model, which can be implemented on hardware without extra operator support. Taking ResNets [9] as an example, there are 7 × 7, 3 × 3, and 1 × 1 convolutions. The PRP method replaces 3 × 3 filters with 1 × 1 filters, which reduces the model computation complexity without extra support in the hardware implementation.

As indicated in [10], the architecture left after pruning is more important than the fine-tuned sparse weights, that is to say a pruned network could achieve the same accuracy without inheriting original weights. Thus, the PRP approach aims to find an efficient pruned network architecture, which can achieve the balance between complexity reduction and model accuracy. We introduce neural architecture search (NAS) technique to achieve this goal. Compared to pruning schemes based on human-designed polocies [11], the AutoML-based PRP method can reduce manual works and achieve better performance. Apart from model accuracy and parameter reduction, hardware implementation metrics, such as latency, are also applied as the feedback to search networks [12].

Inspired by AutoML-based researches, especially the One-Shot model [13,14] and the Meta-learning method [15], we develop a SearchNet to evaluate all potential pruned architectures quickly. With the pattern replacement, the reconstructed compact network is generated from the original network. The proportion and position of replacement filters are searched and evaluated by the proposed SearchNet. Fig. 1 illustrates the process of the proposed PRP approach.

To prove the effectiveness, we perform the PRP approach on widely-used ResNet. With a similar accuracy loss, the PRP approach have better compression compared to prior filter pruning algorithms on both CIFAR-10 and ILSVRC-2012 data-sets. Besides, we evaluate the PRP-ResNet on a prior accelerator supporting sparse CNNs. The proposed PRP approach can achieve better energy efficiency compared to other pruned ResNets with similar sparsity.



(a) Basic Block Structures (b) Bottleneck Block Structures

Figure 2. Block Structures of ResNets. The *Proj* denotes the identity projection utilized in blocks to align the channel numbers of the input and the output.

2. Methodology

2.1. Pattern Replacement Pruning (PRP)

Pattern pruning is a suitable alternative for filter pruning due to its flexibility. [7] uses bank-level sparse pattern and [8] gives the constrain that paralleled lines should share the same sparsity. Different from filter pruning, pattern pruning will not cause a direct reduction in the model FLOPs unless supported by a specially hardware.

To collect different information during processing, CNNs usually constructed by filters of multiple kernel sizes. The proposed PRP approach utilizes this property to compress the network architecture by replacing operators with the available compact ones. The replacement can preserve the network architecture compared to the filter pruning approach, minimizing the accuracy loss. Besides, FLOPs reductions of PRP-based networks can be utilized by the existing hardware architecture, for the computation patterns of pruned networks are not extended compared to the original ones.

Take ResNet [9] as an example to illustrate the PRP approach. For different applications, ResNets are usually constructed by block structures shown as Fig 2, and consists 1×1, 3×3, and 7×7 filters in the network architecture. Utilizing the PRP approach, we mainly replace the 3×3 filters with 1×1 filters in ResNets, which could achieve 50% FLOPs reduction with negligible accuracy loss.

For 3×3 filters, the ideal FLOPs reduction of the PRP approach equals to that of the 1-in-9 pattern pruning method, which is achieved by pre-setting 8 zero positions in filters. However, the pruning pattern needs to be selected. For a layer with N 3×3 filters, M filters are pre-set to be pruned. For the PRP method, only the suitable replacement ratio, i.e. M, needs to be considered. While for the pattern pruning, M^9 different pruning patterns also needs to be selected, which is impossible to find the optimal solution manually. Also, with our experiments, models after the PRP could converge faster and achieve a higher accuracy compared to 1-in-9 pattern pruned ones.

When processing with compact operators such as 1×1 filters, the PRP approach would replace them with 0×0 filters, i.e. filter pruning. Combining with the filter pruning, the PRP approach can achieve the target FLOPs reduction.

The suitable replacement ratio for each convolutional layer is searched and evaluated by the AutoML methods without human design. Also, the search procedure also prevents the pruning scheme from being trapped in the local optimal. With selected replacement ratio combinations, the PRP compressed network can be reconstructed.

2.2. Training the SearchNet

Based on the Meta-learning method, we develop a SearchNet to find the suitable replacement ratio by evaluating all the potential pruned architecture quickly. Also, the SearchNet takes compression ratio as input and generates weights for the compressed model. Here, the compression ratio denotes the replacement ratio for 3×3 filters or the filter pruning ratio for 1×1 filters. When searching and evaluating the pruned network, only the SearchNet is evolved, which reduces the retraining efforts of the pruned network.

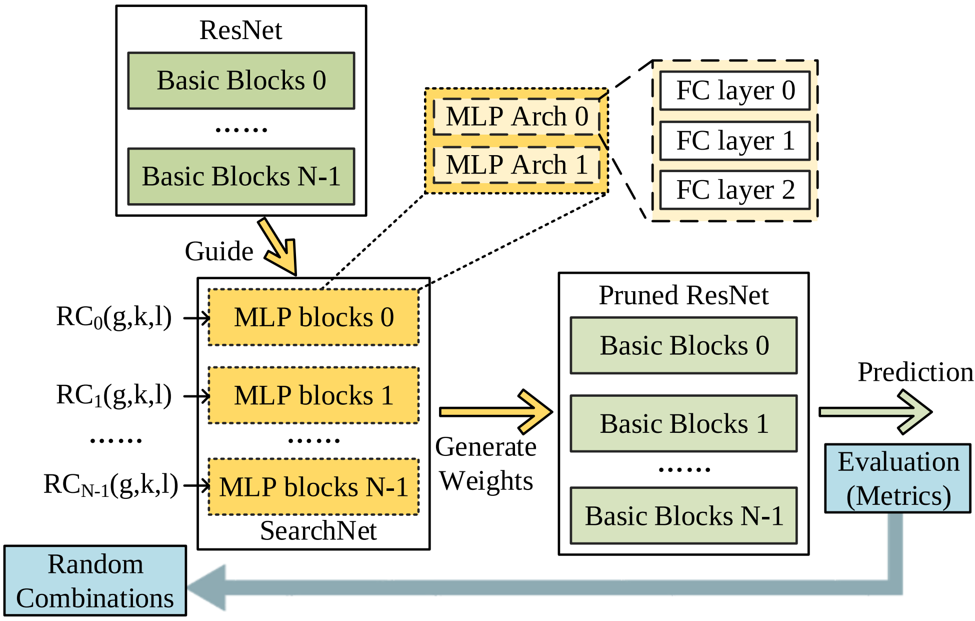


Figure 3. Illustration of the SearchNet, as exampled by a basic block constructed network. *RC* denotes for the replacement ratio for 3 × 3 filters.

Under the conduct of the original network architecture, the SearchNet is built up by replacing convolutional layers by multi-layer perceptron (MLP) architectures. Each MLP architecture contains three fully connected (FC) layers, aiming to generate weights of the corresponding convolutional layer. To ensure the efficiency of the SearchNet, the structure of the MLP architectures are dedicated designed. For block-constructed ResNets, the MLP architectures in the SearchNet are also combined as MLP blocks, which helps to squeeze the search space and increase the training stability.

Take the basic block as an example to illustrate the details of the SearchNet as shown in Fig.3. For a basic block with $64 \times 64 \times 3 \times 3$ filter size, the corresponding MLP architecture is set as three FC layers with sizes of 3×16, 16×32 and 32×36864. Each MLP block receives an input vector composed by compression ratios of adjacent convolutional layers, consisting previous layer and layers of the current block. The included compression ratio of the previous layer is to enlarge the input of the MLP block. With the input replacement ratio vector of (0.1, 0.2, 0.6), the replaced filters are utilized for inference while only weights of the MLP are updated in the backward pass. The update of pruned net can be omitted for all operations are differentiable. The feasibility of this strategy comes from the over-parameter of the MLP structure, allowing the SearchNet to generate corresponding sub-networks according to different compression ratios.

Furthermore, the proposed PRP method can prune shortcut layers freely compared with previous filter pruning methods. The shortcut layer is utilized in ResNets to address the accuracy degradation problem [9], which is performed by a projection from the block input to its output. For filter pruning methods, extra computations will be introduced in the projection when filter pruning ratios varies in a block. For the proposed PRP approach, when processing with a basic block, we only replace 3×3 filters with 1×1 filters, and output channels remain unchanged compared to the original network. As for a bottleneck block, we keep the 1×1 filters' prune ratio unchanged unless the block contains a downsampling layer. We handle this problem by pruning the downsampling layer and the 3rd layer of the bottleneck block at the same ratio.

2.3. Architecture Search

We look for the optimal subnet with the SearchNet. The search and evaluation procedures are processed in an iterative way. In each iteration, $N - 10$ random compression ratio combinations for the whole model are generated under the preset pruning ratio, which form the search space. Then, the SearchNet constructs eligible subnets and generates corresponding weights directly, which omits the duplicate retraining of the original network. These generated subnets are evaluated by multiple metrics, such as model accuracy, parameters, and estimated latency of its deployment on hardware. 10 combinations with best performances are selected and combined with the random generated $N-10$ candidates for the next iteration. The whole process ends when the best performance hardly changes, then outputs the optimal subnet.

3. Experimental Results

3.1. Experiments Settings

ResNets [9] are widely applied nowadays, and most computer visions take residual structures as their backbone. Thus, to prove the effectiveness of our PRP method, we apply it on the widely utilized ResNets and evaluate the PRP-Reset on two benchmarks: CIFAR-10 and ILSVRC-2012. The PRP method could also be extended to other off-the-shelf networks and applied in a variety of deep learning scenarios, which will be explored in the future.

During every mini batch of the training stage, random compression ratio combinations are generated as input vectors of the SearchNet. With forward and backward propagation, the SearchNet could gradually gaining the ability of providing reasonable weights for sub-networks.

3.2. Results and Comparisons

To compare the PRP method with normal filter pruning methods, the metrics of model FLOPs and accuracy are utilized as the comparison standards. Compared to network sparsity, the reduction of FLOPs can better reveal the performance of pruning algorithms. As illustrated in the Section, ResNets can be categorized into two types, Type-I: basic-block-based ResNets and Type-II: bottleneck-block-based ResNets. We will demonstrate our works on these two types, separately.

Basic-block-based ResNets: For ResNet-20, 30, and 110 for CIFAR-10 and ResNet-18 for ILSVRC-2012, networks are mainly built by basic blocks with two 3×3 convolutional layers. Besides, several 1×1 filters are also employed in stride=2 blocks for the identical projection from input to output. The PRP method compacts the Type-I ResNets by replacing 3×3 filters with 1×1 ones.

Bottleneck-block-based ResNets: When processing with larger data-sets, instead of basic blocks, ResNets are built by bottleneck blocks to increase network depth with a more compact structure. 1×1 filters account for a large portion in Type-II ResNets. Thus, to reduce more FLOPs of the network, the PRP method compresses 1×1 filters by filter pruning, along with replacing 3×3 filters. That is to say, 1×1 filters are selected and replaced by 0×0 ones in the PRP method for Type-II ResNets.

Then we provide a brief description of the dataset we will be using:

The CIFAR-10 dataset: The CIFAR-10 dataset shown in fig 4 consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

The ILSVRC2012 dataset: ImageNet is a dataset that Feifei Li, a professor at Stanford University, led the construction of in order to solve the problems of overfitting and generalisation in machine learning. Until now, this dataset is still one of the most commonly used datasets for image classification, detection, and localisation in the field of deep learning. ILSVRC2012 shown in fig 5 refers to a subset of ImageNet-based competitions, i.e., a subset of ImageNet, of which the most commonly used is the 2012 dataset, denoted ILSVRC2012.

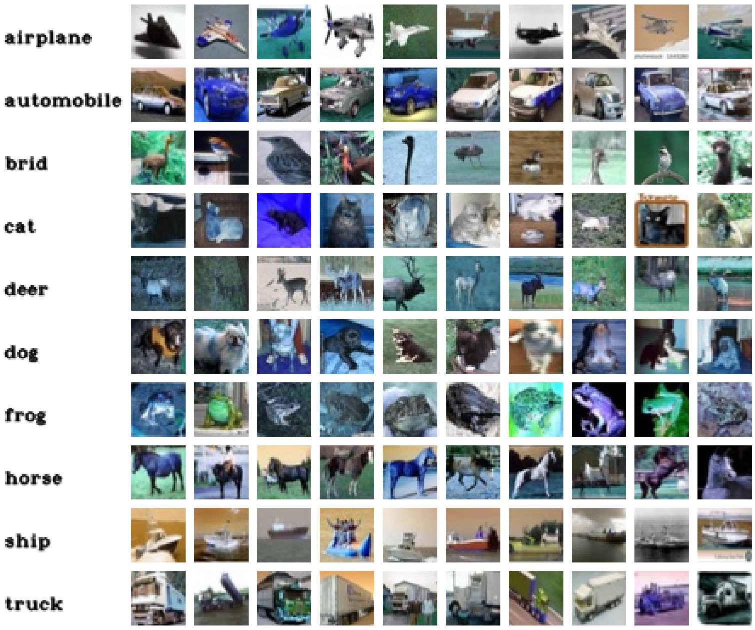


Figure 4. Overview of the Dataset CIFAR-10

The ILSVRC2012 dataset has 1000 classifications with about 1000 images per classification. The total number of these images used for training is about 1.2 million, in addition to a number of images used as a validation set and a test set.ILSVRC2012 contains 50,000 images as a validation set and 100,000 images as a test set.

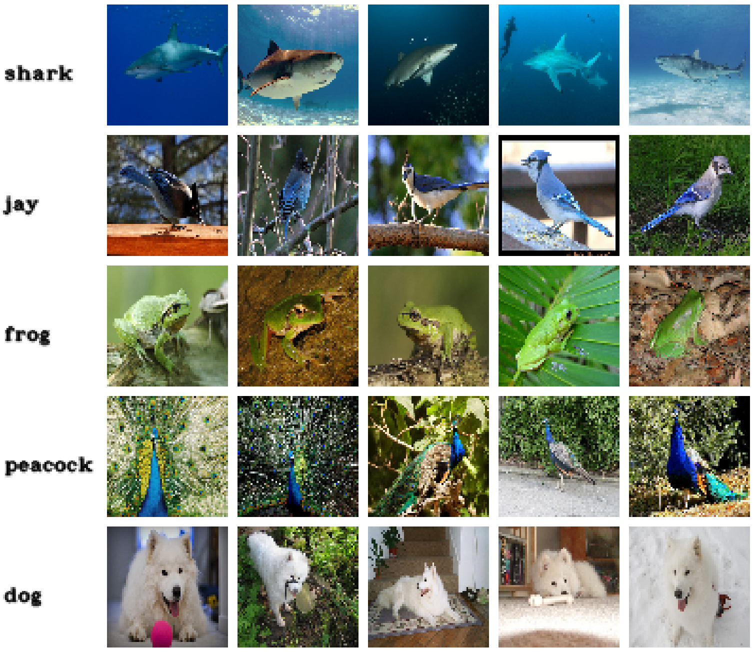


Figure 5. Overview of the Dataset ILSVRC2012

TABLE 1 and TABLE 2 present performances of the PRP method deployed on different data-sets. When evaluating on CIFAR-10, the PRP-method can reduce FLOPs to half with merely accuracy loss for ResNet-56, the network can still maintain considerable accuracy when aggressively pruned by about 80%. As for ResNet-110, After subtracting 60% flops, the accuracy loss of the model is only 0.05. In fact, the pruned ResNet-110 has achieved better performance than the original ResNet-56 with similar FLOPs, this also confirms that

PRP is an effective pruning scheme, which can obtain small models with higher accuracy and fewer FLOPs from large basic models.

Table 1. Experiments on CIFAR-10

Model	Result	Base	Pruned	Top-1	FLOPs
		Top-1	Top-1	↓%	↓%
RseNet-56	AMC [16]	92.8	91.9	0.9	50
	FPGM [17]	93.59	93.26	0.33	52.6
	SFP [6]	93.59	93.35	0.24	52.6
	LFPC [18]	93.59	93.24	0.35	52.9
	TRP [19]	93.14	91.62	1.52	77.82
	Ours	**93.65**	**93.61**	**0.04**	**54.3**
	Ours	**93.65**	**92.54**	**1.11**	**78.6**
RseNet-110	Li et al. [20]	93.53	93.30	0.23	38.60
	SFP [6]	93.50	92.74	0.76	48.5
	HRank [21]	93.50	93.36	0.14	58.2
	Ours	**93.94**	**93.89**	**0.05**	**61.43**

The results for ILSVRC-2012 also demonstrate the effectiveness of the PRP method. We carefully adjusted the proportion of pruned FLOPs to about 50%, which is most commonly used in practical applications, and 60%, which is more aggressive. We can see that PRP maintains good accuracy under the condition of about half FLOPs reduction, which well proves the effectiveness and generalization of PRP on larger data-sets. At the same time, the experimental results also show that adding only very few additional FLOPs pruning on the basis of 50% will have a harmful impact on model performance. We speculate that this is because the parameters left are not enough to accommodate the knowledge of ILSVRC-2012, which leads to an accelerated decline in model accuracy.

Compared to the other filter pruning schemes, the proposed PRP method can achieve a better balance between the model complexity and accuracy. As shown in TABLE 1 and TABLE 2, with a similar FLOPs reduction, the PRP method outperforms prior works in model accuracy. Besides, more FLOPs reduction can be achieved withe a similar accuracy loss. Thus, the PRP method is more flexible and soft, which is very important for model compression.

After our analysis, the PRP method benefits from the *replacement* idea, which compresses the network without squeezing its original network architecture. Also, the compact operators comes from the original network and does not introduce any additional operators, which enables the compressed network be processed by previous accelerators only with a little modification on hardware control.

3.3. Case Study

In this section, we conduct ablation studies for further analyzing the proposed pruning method.

Direct 1-in-9 pattern pruning There are two main hinders for implementing the direct 1-in-9 pattern pruning: 1) Select suitable pruning positions for each filter, which includes 9 combinations in a 3×3 kernel; 2) Whether to prune one filter or not. For the first problem, we sum the absolute value of all channels and use the largest one as the maintaining for the filter. For the second obstacle, we set a fixed layer-wise sparse ratio and choose to prune filters with smaller L1-norms. Experiments show that under similar FLOPs, direct pattern pruning tends to suffer a 1% top-1 accuracy decrease. We summarize two reasons for the accuracy drop: 1) Measuring the importance of a single channel could be quite hard, as it is

Table 2. Experiments on ILSVRC-2012

Model	Result	Base Top-1	Base Top-5	Top-1↓	Top-5↓	FLOPs↓%
ResNet-50	SFP [6]	76.15	92.87	1.54	0.81	41.8
	GAL-0.5 [22]	76.15	92.87	4.20	1.93	43.03
	NISP [23]	-	-	0.89	-	44.01
	HP [24]	76.01	92.93	1.14	0.50	50
	MetaPruning [25]	76.6	-	1.2	-	51.10
	Autopr [26]	76.15	92.87	1.39	0.72	51.21
	GDP [27]	75.13	92.30	3.24	1.59	51.30
	FPGM [17]	76.15	92.87	1.32	0.55	53.5
	Ours	**76.15**	**92.87**	**0.16**	**0.08**	**54.32**
	C-SGD (extension) [28]	76.15	92.87	0.86	0.48	55.44
	C-SGD [28]	75.33	92.56	0.79	0.47	55.76
	ThiNet [29]	75.30	92.20	3.27	1.21	55.83
	SASL [30]	76.15	92.87	1.00	0.40	56.10
	Ours	**76.15**	**92.87**	**0.43**	**0.18**	**57.12**
	TRP [19]	75.90	92.70	3.21	1.29	56.52
	LFPC [18]	76.15	92.87	1.69	0.55	60.8
	HRank [21]	76.15	92.87	4.17	1.86	62.10
	Ours	**76.15**	**92.87**	**1.21**	**0.43**	**61.36**

strongly related to the input feature. Judging by sum of absolute value is not reasonable enough. 2) L1-norm may represent the price of a certain filter, but if most of the greater values occasionally share the same position at kernel-level across differ channels, it could be safely pruned. With PRP, we transform these problems into a combinatorial optimization problem, providing theoretical basis for PRP's validity.

Mixed replacement Inception [31] shows that fusing different size of convolution filters could bring performance improvement for CNNs. Consider a layer with 32 3×3 filters with the replacing ratio $r = 0.5$. Instead of replacing half of the 3×3 filters with 1×1 filters, we could replace them by a mixed of ten 1×1 filters and six 5×5 depth-wise filters. This solution is feasible at the algorithm level, but due to hardware limitations, we did not experiment with this strategy in this work.

4. Hardware Implementation

In this section, we evaluate the proposed PRP method on the existing hardware accelerator to prove its effectiveness. With the same architecture, the proposed PRP approach can achieve the effective speedup compared to the original network and the unstructured pruning approach with a similar sparsity. The acceleration system can also outperforms other structured pruning systems on the overall performance.

4.1. Hardware Architecture

The architecture proposed in [32] is designed to support the acceleration of unstructured pruned networks. The architecture can be configured by software to support the acceleration of different networks. The overall architecture is shown as Fig. 6.

To skip pruned calculations, each PE is driven by the same weight in serial. Paralleled PEs can be configured to process weights from different channels and filters. Due to the irregular distribution of pruned weights in unstructured pruned networks, load

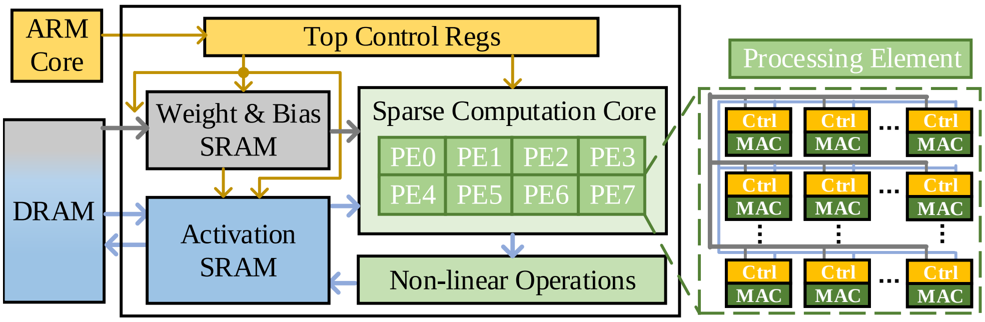


Figure 6. The Overall Architecture of the Implemented Accelerator. The yellow squares and lines denote the hardware control, and will be reconfigured when processing different networks.

Table 3. The Implementation Performance Comparison of Different Pruning Schemes

Works	[33]	[32]	Ours	
Pruning Scheme	ADMM-Structured	ADMM-Unstructured	PRP	None
Sparsity	45%			100%
Quantization	16-bit	8-bit	8-bit	8-bit
Accuracy Loss (top-5)	No Significant Loss	No Loss	No Loss	No Loss
Power	15.4	4.6	4.6	4.6
Images/J	3.70	5.00	5.68	3.04
Computation Efficiency	-	57.90%	65.37%	65.07%

imbalance and conflict memory accesses occurs in hardware implementation, leading to the decrease of hardware efficiency. [32] proposes an off-chip weight pre-processing (WPP) algorithm for pruned weights to improve hardware performance and simplify hardware control. However, extra zeros are introduced in unstructured pruned models after the WPP algorithm, which results in a lower acceleration.

Compressed models after the proposed PRP method have regular distribution of pruned weights, and can be mapped to the architecture without the WPP algorithm. Also, compared to other solution with the co-design of algorithm and hardware, the proposed PRP method is independent from the hardware design and can be implemented on various accelerators only with slightly modification of hardware control.

4.2. Results and Comparisons

Evaluated on the the architecture proposed in [32], the PRP method achieves performances shown as Table 3 with the same sparsity of 45% and similar accuracy loss. Compared to work [33], the proposed method can achieve an improvement of 1.53× in power efficiency, which gains from the hardware architecture and the specific pattern of the PRP method. Besides, evaluated on the same accelerator, the processing speed of the proposed PRP method is 1.14× faster than that of the unstructured alternating direction method of multipliers (ADMM)-based [5] pruning algorithm.

The hardware inefficiency of [32] is mainly caused by the unsuitable tiling scheme, the under utilization when processing stride=2 convolutions, and communication overhead with external memories. Weights in stride=2 blocks pruned by the PRP approach are more efficient than unstructured pruned ones, leading to the improved hardware efficiency. As shown in TABLE 3, the computation efficiency of the PRP approach is similar to the dense network, which are substantially higher than that of unstructured pruned ones.

Implemented with the architecture shown as [32], the hardware efficiency comparisons of different network blocks are shown as Fig. 7.

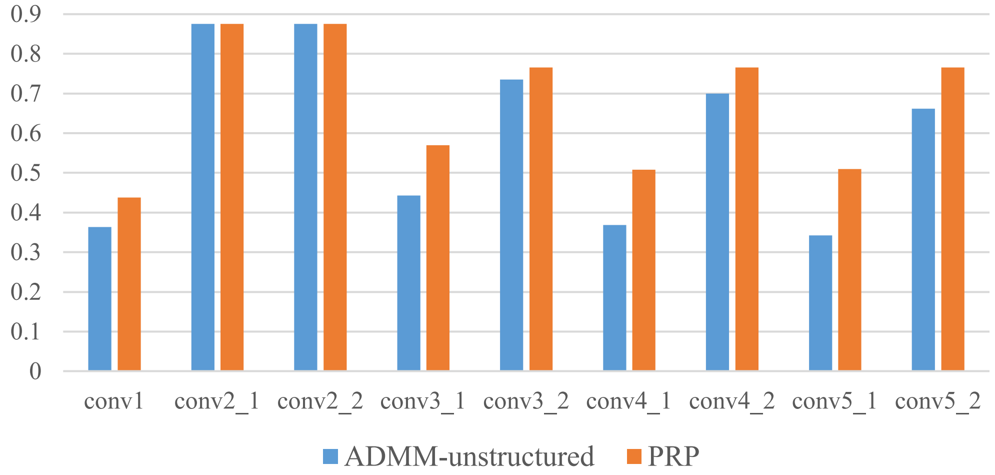


Figure 7. The Computation Efficiency Comparison of Different Blocks Among Different Pruning Schemes for ResNet-50, where *conv* represents different bottleneck blocks.

5. Conclusion

In this paper, we propose a pattern replacement pruning (PRP) approach for hardware-oriented network compression. With the developed pattern replacement idea, the PRP could compact networks without network architecture modification or inducing extra operator, which ensures the model accuracy and helps the hardware deployment of compressed networks. Besides, Auto-ML methods are combined in the PRP approach to save human efforts and achieve better performance. With experiments on ResNets and comparison with prior works, the PRP approach proves its validation by achieving higher model accuracy at a similar FLOPs reduction. After evaluating on an existing CNN acceleration architecture, the proposed PRP method is able to achieve 14% higher energy efficiency compared to other pruning schemes.

References

1. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv:1804.02767* **2018**.
2. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the Proceedings of the European Conference on Computer Vision, 2018.
3. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
4. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv:1510.00149* **2015**.
5. Zhang, T.; Ye, S.; Zhang, K.; Tang, J.; Wen, W.; Fardad, M.; Wang, Y. A Systematic DNN Weight Pruning Framework Using Alternating Direction Method of Multipliers. *Lecture Notes in Computer Science* **2018**, p. 191–207.
6. He, Y.; Kang, G.; Dong, X.; Fu, Y.; Yang, Y. Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. In Proceedings of the Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 2234–2240.
7. Wang, S.; Li, Z.; Ding, C.; Yuan, B.; Qiu, Q.; Wang, Y.; Liang, Y. C-LSTM: Enabling Efficient LSTM Using Structured Compression Techniques on FPGAs. In Proceedings of the Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2018, p. 11–20.
8. Han, S.; Kang, J.; Mao, H.; Hu, Y.; Li, X.; Li, Y.; Xie, D.; Luo, H.; Yao, S.; Wang, Y.; et al. ESE: Efficient Speech Recognition Engine with Compressed LSTM on FPGA. In Proceedings of the Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2018.
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
10. Frankle, J.; Carbin, M. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In Proceedings of the International Conference on Learning Representations, 2019.
11. He, Y.; Zhang, X.; Sun, J. Channel pruning for accelerating very deep neural networks. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1389–1397.

12. He, Y.; Lin, J.; Liu, Z.; Wang, H.; Li, L.J.; Han, S. Amc: Automl for model compression and acceleration on mobile devices. In Proceedings of the Proceedings of the European Conference on Computer Vision, 2018, pp. 784–800.
13. Bender, G.; Kindermans, P.J.; Zoph, B.; Vasudevan, V.; Le, Q. Understanding and simplifying one-shot architecture search. In Proceedings of the International Conference on Machine Learning, 2018, pp. 549–558.
14. Cai, H.; Gan, C.; Wang, T.; Zhang, Z.; Han, S. Once-for-All: Train One Network and Specialize it for Efficient Deployment. In Proceedings of the International Conference on Learning Representations, 2020.
15. Liu, Z.; Mu, H.; Zhang, X.; Guo, Z.; Yang, X.; Cheng, T.K.T.; Sun, J. MetaPruning: Meta Learning for Automatic Neural Network Channel Pruning. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision, 2019.
16. He, Y.; Lin, J.; Liu, Z.; Wang, H.; Li, L.J.; Han, S. Amc: Automl for model compression and acceleration on mobile devices. In Proceedings of the Proceedings of the European conference on computer vision (ECCV), 2018, pp. 784–800.
17. He, Y.; Liu, P.; Wang, Z.; Hu, Z.; Yang, Y. Filter pruning via geometric median for deep convolutional neural networks acceleration. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4340–4349.
18. He, Y.; Ding, Y.; Liu, P.; Zhu, L.; Zhang, H.; Yang, Y. Learning filter pruning criteria for deep convolutional neural networks acceleration. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2009–2018.
19. Xu, Y.; Li, Y.; Zhang, S.; Wen, W.; Wang, B.; Qi, Y.; Chen, Y.; Lin, W.; Xiong, H. Trp: Trained rank pruning for efficient deep neural networks. *arXiv preprint arXiv:2004.14566* **2020**.
20. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710* **2016**.
21. Lin, M.; Ji, R.; Wang, Y.; Zhang, Y.; Zhang, B.; Tian, Y.; Shao, L. Hrank: Filter pruning using high-rank feature map. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 1529–1538.
22. Lin, S.; Ji, R.; Yan, C.; Zhang, B.; Cao, L.; Ye, Q.; Huang, F.; Doermann, D. Towards optimal structured cnn pruning via generative adversarial learning. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 2790–2799.
23. Yu, R.; Li, A.; Chen, C.F.; Lai, J.H.; Morariu, V.I.; Han, X.; Gao, M.; Lin, C.Y.; Davis, L.S. Nisp: Pruning networks using neuron importance score propagation. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 9194–9203.
24. Xu, X.; Park, M.S.; Brick, C. Hybrid pruning: Thinner sparse networks for fast inference on edge devices. *arXiv preprint arXiv:1811.00482* **2018**.
25. Liu, Z.; Mu, H.; Zhang, X.; Guo, Z.; Yang, X.; Cheng, K.T.; Sun, J. Metapruning: Meta learning for automatic neural network channel pruning. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 3296–3305.
26. Luo, J.H.; Wu, J. Autopruner: An end-to-end trainable filter pruning method for efficient deep model inference. *Pattern Recognition* **2020**, *107*, 107461.
27. Lin, S.; Ji, R.; Li, Y.; Wu, Y.; Huang, F.; Zhang, B. Accelerating Convolutional Networks via Global & Dynamic Filter Pruning. In Proceedings of the IJCAI. Stockholm, 2018, Vol. 2, p. 8.
28. Ding, X.; Ding, G.; Guo, Y.; Han, J. Centripetal sgd for pruning very deep convolutional networks with complicated structure. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4943–4953.
29. Luo, J.H.; Wu, J.; Lin, W. Thinet: A filter level pruning method for deep neural network compression. In Proceedings of the Proceedings of the IEEE international conference on computer vision, 2017, pp. 5058–5066.
30. Shi, J.; Xu, J.; Tasaka, K.; Chen, Z. SASL: Saliency-adaptive sparsity learning for neural network acceleration. *IEEE Transactions on Circuits and Systems for Video Technology* **2020**, *31*, 2008–2019.
31. Szegedy.; Christian.; Vanhoucke.; Vincent.; Ioffe.; Sergey.; Shlens.; Jonathon.; Wojna.; Zbigniew. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016.
32. Xie, X.; Lin, J.; Wang, Z.; Wei, J. An Efficient and Flexible Accelerator Design for Sparse Convolutional Neural Networks. *IEEE Transactions on Circuits and Systems I: Regular Papers* **2021**, pp. 1–14.
33. Zhu, C.; Huang, K.; Yang, S.; Zhu, Z.; Zhang, H.; Shen, H. An Efficient Hardware Accelerator for Structured Sparse Convolutional Neural Networks on FPGAs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **2020**, *28*, 1953–1965.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.