Exploring the Differences in Pruning Methods for Convolutional Neural Networks

Romanela Lajić, Peter Peer, Žiga Emeršič
University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia
romanela.lajic@fri.uni-lj.si, peter.peer@fri.uni-lj.si, ziga.emersic@fri.uni-lj.si

With the rising computational and memory cost of deep neural networks there is more effort to reduce the size of these models, especially when their deployment on resource constrained devices is the goal. New methods of compressing neural networks are being constantly developed with the goal of minimizing the drop in accuracy. In this paper we focus on pruning techniques as a way of compression. We present a comparison of different pruning criteria and analyze the loss in accuracy for the case of a simple non-iterative pruning procedure. We provide the comparison between cases when these criteria are applied to different architectures of convolutional neural networks.

DOI
https://doi.org/10.18690/um.feri.2.2025.2

ISBN
978-961-286-960-1

Keywords:
convolutional neural networks,
model compression,
model pruning,
deep learning,
deep neural networks

University of Maribor Press

DOI
https://doi.org/10.18690/um.feri.2.2025.2

ISBN
978-961-286-960-1

Ključne besede:
konvolucijske nevronske mreže,
kompresija modelov,
obrezovanje modelov,
globoko učenje,
globoke nevronske mreže

Univerzitetna založba Univerze v Mariboru

RAZISKOVANJE RAZLIK MED METODAMI OBREZOVANJA ZA KONVOLUCIJSKE NEVRONSKE MREŽE

ROMANELA LAJIĆ, PETER PEER, ŽIGA EMERŠIČ
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija
romanela.lajic@fri.uni-lj.si, peter.peer@fri.uni-lj.si, ziga.emersic@fri.uni-lj.si

Zaradi naraščajočih računskih in pomnilniških zahtev globokih nevronskih mrež je vedno več truda usmerjenega v zmanjšanje velikosti teh modelov, še posebej kadar je cilj njihova uporaba na napravah z omejenimi viri. Nenehno se razvijajo nove metode za stiskanje nevronskih mrež, katerih cilj je čim manjši padec natančnosti. V tem članku se osredotočamo na tehnike obrezovanja (pruning) kot način stiskanja. Predstavimo primerjavo različnih kriterijev obrezovanja in analiziramo izgubo natančnosti pri enostavnem, neiterativnem postopku obrezovanja. Podamo primerjavo primerov, kjer so ti kriteriji uporabljeni pri različnih arhitekturah konvolucijskih nevronskih mrež.

1 Introduction

Since the surge of popularity of deep neural networks in the area of computer vision has started, we have seen a growing trend when it comes to size and number of parameters of these models. This has also led to a rise in their memory and computational cost which makes their deployment on resource constrained devices, such as mobile or edge devices, challenging.

While there is a lot of effort invested in creating more lightweight models (Iandola, 2016; Buotros et al., 2022; Sandler et al., 2018), they can rarely achieve the accuracy comparable to the one of deep models. Compression attempts to reduce the number of parameters of larger models while maintaining the original accuracy or minimizing its reduction.

There are several different techniques of compressing neural networks. Some techniques such as quantization (Cai & Vasconcelos, 2020; Jacob et al., 2018; Zhou et al., 2017) focus on reducing the memory cost of a neural network by reducing the number of bits required for parameter representation. Other methods such as knowledge distillation (Hinton, 2015; Li et al., 2023; Park et al., 2019) attempt to train lightweight models so that they mimic the behavior of a larger architecture. Third group of methods, such as low-rank matrix decomposition (Lin et al., 2018) attempt to reduce computational cost of inference by reducing the number of operations in the network.

Pruning focuses on removing redundant connections or filters from a neural network. Connections which are considered redundant are the ones carrying either a low amount of information or less important information and are determined by a specific metric called the pruning criterion. The matter of choosing a pruning criterion is a topic of a large number of works. Criterion is mostly chosen in such a way that the reduction in the accuracy of the network is minimized, but can also be chosen in regards to a particular, specialized task such as reduction of bias in biometric models (Lin et al., 2022), improving discriminative power of the network (Liu et al., 2021) or enhancing generalization ability (Zimmer et al., 2023).

In this paper we will do a comparison and analysis of three different pruning criteria, based on different indicators in the neural network, such as outputs of layers, filter weights and batch normalization parameters. We will test the criteria on different architectures of convolutional networks and present results obtained on different datasets.

2 Related work

One of the early works presenting neural network pruning (LeCun et al., 1989) was published all the way back in 1989. The very idea for pruning stems from biology. At a young age children develop a large number of neural connections in order to make learning more efficient. Later on in life when a lot of the tasks learned in the earlier years become standard a lot of these connections are deactivated (Calderia et al., 2025). By the same logic, having a large number of layers and connections in a neural network makes information flow and learning easier, but during inference a lot of those connections are proven to be redundant and can be removed.

Pruning once again became popular with the development of deep learning, when the size of models began to significantly increase. At first, most of the work focused on finding the optimal pruning criterion, which would determine which of the connections can be removed from the network without a significant drop in accuracy. One of the first notable works around this time (Li et al., 2016) proposes computing L^1 norm of filter weights in each layer of a convolutional network. Filters with the lowest value of the norm are then removed under the assumption that they have the least effect on the output. Some methods focus on using the change in the loss function when a filter is pruned as an indicator of importance. Method presented in (You et al., 2019) proposes using a first-order Taylor expansion to estimate the change in the loss caused by setting a filter to zero Certain papers such as (Liu et al., 2017) are based on the concept of sparsity regularization. The mentioned paper proposes using batch normalization layers as indicators and applies L^1 regularization to batch normalization weights before using their values to choose which filters to prune.

Several recent works exploring pruning focus on finding a new, more efficient pruning metric, which remains one of the biggest problems of the method. Shang et al. (Shang et al., 2022), propose breaking down the pruning procedure into layer-

level problems and solving them cooperatively. By assuming that the removal of a filter mostly affects the filters in the same layer, they propose using an evolutionary algorithm to choose a subpopulation of filters to keep for each layer. On the other hand, the work in (Liu et al., 2021) focuses more on improving the dicriminative power of the network by introducing discriminative-aware losses such as cross-entropy to intermediate layers of the network, and combining them with feature-reconstruction error. Basha et al. (Basha et al., 2024) propose looking at the training history of the network. The hypothesis is that if the difference between the filters does not change significantly through the training epochs, those filters can be considered redundant. They suggest measuring the difference between the L^1 norms of different filter pairs during the training procedure and pruning one of the filters in the pairs with the smallest sums of absolute differences.

Li et al. (Li et al., 2022) argue that the structure of a network is just as important as the weights and that random channel pruning has the ability to reach performance levels of more complex pruning criterion. Although simply randomly choosing channels to prune cannot achieve competitive performance, the authors propose two setups based on random pruning. One is randomly choosing filters in a layer, then pruning them based on a certain criterion, such as L^1. The other method is randomly choosing network configurations and training them in parallel.

Work in (Fang et al., 2023) addresses the issues with structural pruning. When performing structural pruning the architecture of the network is changed and interdependence between the parameters can oftentimes be violated. For this reason, the design often needs to be architecture specific. The authors attempt to find a way of automating structural pruning by representing the network as a graph and performing pruning by taking these dependencies into consideration.

Zimmer et al. (Zimmer et al., 2023) focuses on trying to enhance generalization ability of pruned networks by averaging out the parameters of different models. Since averaging the parameters of differently pruned models could actually increase sparsity of the final model, the authors propose so-called sparse model soups. This entails pretraining and pruning a larger model, then forming different models by changing other hyperparameters which can then be averaged. This allows the sparsity level to remain intact.

Some works have attempted to combine pruning with other compression techniques. In (Li et al., 2023) pruning is combined with mixed-precision quantization for a more efficient hardware acceleration. Iterative quantization is performed until the redundant weights are completely pruned and the rest of the network is quantized with a different bit-width.

3 Methodology

Pruning of filters in convolutional neural network in most cases consists of three steps: choosing the least important filters which can be removed, removing the filters and fine-tuning the model. The metric which implies which filters are the least important is called the pruning criterion, while the percentage of the filters chosen to be removed is referred to as pruning sparsity. We will be examining three different pruning criteria and comparing their performance.

First examined algorithm is based on the output of the filters. We consider a greedy algorithm which chooses filters based on their output norm. The algorithm removes filters one by one by choosing the filter with the lowest output norm after the removal of the previous filter until we achieve the desired sparsity.

Other two methods determine redundant filters based on network parameters. The first method determines the least important filters by calculating their L^1 norm under the assumption that the filters with the lowest norm contribute less to the output. The second method, proposed in (Liu et al., 2017), looks at weights of a batch normalization layer which is typically placed after the convolutional layer. The method applies L^1 regularization to batch normalization layers, after which it chooses the filters to prune based on the corresponding batch norm weights.

We evaluate the three techniques applied to three architectures of convolutional networks, VGG-16, ResNet-18 and ResNet-50, on CIFAR-10 and CIFAR-100 datasets. We also provide results of the VGG-16 network on the ImageNet dataset.

4 Experiments and Results

When it comes to CIFAR datasets, all three networks have been trained from scratch after which pruning has been applied using the three described criteria with the same sparsity level of 40%. Pruned networks are then retrained on epochs, with a batch

size 32, using an SGD optimizer starting at a learning rate of 1e-3 which is then reduced to values 1e-4 and after that 1e-5. When using the CIFAR-10 dataset, the models are retrained on 10 epochs, and on the CIFAR-100 dataset, on 20 epochs. Random rotation and random horizontal flip are applied to the training images. The results are shown in the Table 1 and Table 2. When it comes to the slimming technique the original paper applies L^1 regularization to the batch normalization layers during training. Considering that the tests on the ImageNet dataset are done using a pretrained model, the regularization is omitted in the other two datasets as well. This kind of test will give us an idea of how well batch normalization weights function as an indicator of importance on their own, without any additional preparation, which might prove to be useful in cases where training the original model from scratch is simply not possible for various reasons.

Table 1: Results on CIFAR-10

Model	Method	Accuracy
VGG-16	**Pre-pruning**	**93.18%**
	Greedy	91.01%
	L1	90.67%
	Slimming	90.66%
ResNet-18	**Pre-pruning**	**94.41%**
	Greedy	93.37%
	L1	93.38%
	Slimming	93.54%
ResNet-50	**Pre-pruning**	**94.96%**
	Greedy	93.69%
	L1	93.99%
	Slimming	92.63%

From the tabels we can see that on most combinations of architecture and dataset the pruning criteria give comparable results. There is also a notable drop in performance in most of the cases, in some being more prominent than the others. The results imply that the drop in performance of the pruned models is affected by many factors, including the original architecture, the dataset and the retraining procedure.

VGG-16 is also tested on the ImageNet dataset. Retraining a pruned network on ImageNet is a more challenging task and some works use iterative retraining procedures where the network is retrained after pruning each of the layers, in order to get minimize the loss of accuracy as much as possible. For the sake of direct

comparisson, the network is retrained on ImageNet in the similar way as on CIFAR datasets. The network is retrained on 10 epochs after completely pruning all of the convolutional layers, using the same parameters as described for the CIFAR datasets. The training images are center-cropped to the appropriate size and random horizontal flip is applied. No other augmentations are added to the images. The results are shown in the Table 3.

Table 2: Results on CIFAR-100

Model	Method	Accuracy
VGG-16	**Pre-pruning**	**71.50%**
	Greedy	68.26%
	L1	67.26%
	Slimming	67.92%
ResNet-18	**Pre-pruning**	**75.72%**
	Greedy	74.11%
	L1	75.08%
	Slimming	74.95%
ResNet-50	**Pre-pruning**	**78.76%**
	Greedy	76.87%
	L1	75.66%
	Slimming	77.40%

Table 3: Results on ImageNet

Model	Method	Accuracy
VGG-16	**Pre-pruning**	**73.42%**
	Greedy	68.39%
	L1	68.77%
	Slimming	68.50%

In the case of ImageNet there is also a comparable performance between the different pruning criteria. Performance loss in this case is more significant, since complex datasets require more complex procedures, such as iterative retraining after pruning each of the layers, as was previously mentioned. This allows the network to gradually adjust to the loss of information.

5 Conclusion

In this paper we have explored different pruning criteria based on both outputs of layers and network parameters. We have tested and compared these criteria on three different architectures of convolutional neural networks using three different

datasets. From the results we can conclude that the metrics give comparable results for most cases but that there is a certain drop in accuracy when networks are reduced to 60% of their convolutional filters. In order for the pruned models to give accuracy which matches the one of the full-sized model, more complex retraining procedures must be applied even for simple datasets, while for more challening cases both more complex training procedures and more advanced pruning criteria.

References

Basha, S. S., Farazuddin, M., Pulabaigari, V., Dubey, S. R., & Mukherjee, S. (2024). Deep model compression based on the training history. *Neurocomputing*, *573*, 127257.

Boutros, F., Siebke, P., Klemt, M., Damer, N., Kirchbuchner, F., & Kuijper, A. (2022). Pocketnet: Extreme lightweight face recognition network using neural architecture search and multistep knowledge distillation. *IEEE Access*, *10*, 46823-46833.

Cai, Z., & Vasconcelos, N. (2020). Rethinking differentiable search for mixed-precision neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2349-2358).

Caldeira, E., Neto, P. C., Huber, M., Damer, N., & Sequeira, A. F. (2025). Model compression techniques in biometrics applications: A survey. *Information Fusion*, *114*, 102657.

Fang, G., Ma, X., Song, M., Mi, M. B., & Wang, X. (2023). Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16091-16101).

Hinton, G. (2015). Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Iandola, F. N. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam H., Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2704-2713).

LeCun, Y., Denker, J., & Solla, S. (1989). Optimal brain damage. *Advances in neural information processing systems*, *2*.

Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2016). Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.

Li, J., Guo, Z., Li, H., Han, S., Baek, J. W., Yang, M., ... & Suh, S. (2023). Rethinking feature-based knowledge distillation for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 20156-20165).

Li, Y., Adamczewski, K., Li, W., Gu, S., Timofte, R., & Van Gool, L. (2022). Revisiting random channel pruning for neural network compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 191-201).

Liu, J., Zhuang, B., Zhuang, Z., Guo, Y., Huang, J., Zhu, J., & Tan, M. (2021). Discrimination-aware network pruning for deep model compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(8), 4035-4051.

Lin, S., Ji, R., Chen, C., Tao, D., & Luo, J. (2018). Holistic cnn compression via low-rank decomposition with knowledge transfer. *IEEE transactions on pattern analysis and machine intelligence*, *41*(12), 2889-2905.

Lin, X., Kim, S., & Joo, J. (2022, October). Fairgrape: Fairness-aware gradient pruning method for face attribute classification. In *European Conference on Computer Vision* (pp. 414-432). Cham: Springer Nature Switzerland.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., & Zhang, C. (2017). Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision* (pp. 2736-2744).

Li, Z., Gong, Y., Zhang, Z., Xue, X., Chen, T., Liang, Y., ... & Wang, Z. (2023). Accelerable lottery tickets with the mixed-precision quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4604-4612).

Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3967-3976).

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

Shang, H., Wu, J. L., Hong, W., & Qian, C. (2022). Neural network pruning by cooperative coevolution. *arXiv preprint arXiv:2204.05639*.

You, Z., Yan, K., Ye, J., Ma, M., & Wang, P. (2019). Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. *Advances in neural information processing systems*, *32*.

Zhou, S. C., Wang, Y. Z., Wen, H., He, Q. Y., & Zou, Y. H. (2017). Balanced quantization: An effective and efficient approach to quantized neural networks. *Journal of Computer Science and Technology*, *32*, 667-682.

Zimmer, M., Spiegel, C., & Pokutta, S. (2023). Sparse model soups: A recipe for improved pruning via model averaging. *arXiv preprint arXiv:2306.16788*.