A Comprehensive Study on the Application of Structured Pruning methods in Autonomous Vehicles

Ahmed Hamed*
Valeo Egypt AI Research
ahmed.mohamed-hamed@valeo.com

Ibrahim Sobh*
Valeo Egypt AI Research
ibrahim.sobh@valeo.com

Abstract

Deep neural networks (DNNs) have achieved huge successes in many autonomous driving tasks. However, existing deep neural networks are computationally and memory expensive, making it difficult to be deployed on embedded systems with limited processing power and low memory resources. In this work, a detailed network structured pruning study is conducted for autonomous driving tasks, including object detection, skin semantic segmentation and steering angle prediction. This study considers the network performance for different algorithms at different pruning sparsity levels. We conclude this paper by comparing and discussing the experimental results and proposing the recommended structured pruning algorithms for these tasks.

1 Introduction

Autonomous Vehicles are expected to significantly reduce accidents [1]. Autonomous Vehicles systems have two main approaches, the end-to-end approach, where the deep network predicts the driving signals directly from raw sensor data, and the pipeline approach where the deep networks are components of a larger system. Despite the notable successes of Deep neural networks, the existing models are computationally expensive and memory intensive. Deployment of these models on low memory target devices or for applications with strict latency requirements such as autonomous driving is a challenging task. Therefore, a natural solution is to reduce the size of these models and accelerate them significantly without decreasing the performance. During the past few years, huge progress has been made in this area. Network Quantization compresses the original network by reducing the number of bits used to represent the model's weights, for example from 32-bit floating point numbers to 16-bit [2], or int 8-bit [3], or even binary weights represented as 1-bit [4]. This results in a smaller model size and faster computation, usually with marginal or negligible effect on the performance of the model. Pruning, on the other hand, works by removing weights within a model that have a minor impact on its predictions. This makes pruning a useful technique for reducing model size. Optimal Brain Damage (OBD) [5] was introduced as a technique for reducing the size of a network by selectively deleting weights, where the saliency for each weight is computed using a diagonal Hessian approximation, and the low-saliency parameters are removed, and the network is retrained. More recently, magnitude-based weight pruning methods have become more popular for network pruning. In [6], a method is proposed for pruning redundant connections and retraining the network to fine tune the weights of the remaining connections. Deep compression method was proposed in [7], where the network is pruned by learning only the important weights, then the weights are quantized, and finally Huffman coding is applied. Convolutional neural networks (CNNs) are widely used in computer vision tasks such as classification, object detection and segmentation. Large CNNs with 10s or 100s layers enabled state of the art performance for computer vision tasks. However, deployment of such networks is challenging. In this work, we review one of the most effective

*Both authors contributed equally.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

compression methods for deep neural networks which is structured pruning. Structured pruning removes entire channels or filters completely from the network, instead of removing individual weights, this method does not require any special hardware. This paper is organized as follows: First we discuss different algorithms for applying structured pruning on CNNs: $l1$ Filter pruning [8], filter pruning via Geometric Median (FPGM) [9], Taylor pruning [10]. Then, these algorithms are applied on different applications related to autonomous driving. LiDAR sensor Object detection and camera sensor skin semantic segmentation for autonomous driving pipeline approach, steering angle prediction for camera sensor end-to-end approach. Finally, we conclude this work by comparing and discussing the pruning sparsity levels and the corresponding performance.

2 Structured Pruning methods

In this section, the structured pruning methods are described briefly.

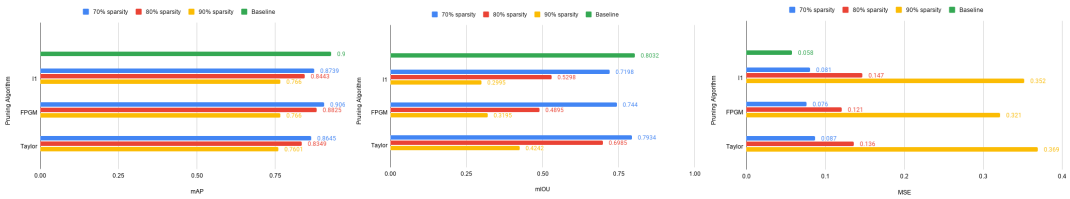
$l1$ **Filter Pruning:** Magnitude-based pruning of weights reduces the number of parameters from fully connected layers, but it is not suitable for reducing the required computation cost in convolutional layers. $l1$ Filter [8] is a structured pruning algorithm that prunes the filters of a CNN network with the smallest $l1$ norm of the weights. Accordingly, by removing whole filters along with their corresponding feature maps, the computation costs are reduced significantly. This approach does not result in sparse connectivity patterns. To prune m filters from the ith convolutional layer, for each filter $F_{i,j}$, the sum of its absolute kernel weights is calculated and the filters are sorted accordingly. Then, the m filters with the smallest sum values are pruned. The kernels in the next convolutional layer corresponding to the pruned feature maps are also removed. Finally, a new kernel matrix is created for both the ith and $i+1$th layers, and the remaining kernel weights are copied to the new model.

FPGM: In the norm-based methods, it is assumed that smaller norm filters are less important and accordingly, filters with the largest norm are kept, others are pruned. In contrast, the Filter Pruning via Geometric Median (FPGM) [9] proposed a method that prunes filters with redundant information in the network. Accordingly, instead of pruning filters with relatively less contribution, FPGM prunes the most replaceable filters containing redundant information, by calculating the Geometric Median of the filters in the same layer. FPGM explicitly considers the mutual relations between filters and shows to achieve good performances when norm-based methods fail.

Taylor Pruning: Taylor First order pruning (TaylorFO) [10] is a pruning criterion which iteratively removes the least important set of filters from the model. However, instead of the estimating the importance of a filter as the squared difference in loss after removing a filter from the network, which is extremely expensive for large networks, this pruning criterion approximates it with a Taylor expansion, making this pruning criterion more practical and easier to implement. The importance of a filter can be defined as $I_m = (E(D, W) - E(D, W|w_m = 0))^2$ where W is the neural network parameters, D is the dataset, E is the error function, W_m is the weight of filter m and I_m is the importance of filter m. This criteria proposed an approximation as $I_m = (g_m w_m)^2$ Where g_m is the gradient of the error function E with respect to the weight of filter W_m.

3 Experimental setup

In this section, the details of the experiments are described. The experiments are conduced across three different applications: LiDAR based object detection, skin semantic segmentation and front camera based steering angle prediction. Automated Gradual Pruning(AGP) [11] is a scheduling algorithm for pruning filters in deep neural networks where the network is trained to gradually increase the sparsity while allowing it to recover from any induced loss in accuracy. A binary mask, that determines which of the weights participate in the forward execution, is added for every layer chosen to be pruned. The AGP algorithm removes filters starting from an initial sparsity to a final sparsity level over a number of pruning steps, starting at training epoch, where the binary masks are updated at a pruning frequency. In all of the following experiments, the AGP scheduling, where the initial sparsity is 0, staring epoch is 50 and update frequency is 1. Finally, in order to inspect the effect of pruning on the performance, three pruning sparsity levels are considered at 70%, 80% and 90%. All the pruning algorithms are applied across all the mentioned applications. Ending epoch for object detection is 100, for skin semantic segmentation is 90 and for steering angle prediction is



(a) Object Detection (b) Semantic Segmentation (c) Steering Angle Prediction

Figure 1: Comparison of the performance on the test data for pruning algorithms with different sparsity levels

100. In this work, Neural Network Intelligence (NNI) [12] toolkit is used for model pruning. For all of the experiments the number of FLOPS is estimated using the NNI tooklit, the toolkit identifies the remaining filters according to its mask, which does not take into consideration the pruned input channels. Accordingly, the estimated FLOPS are larger than the expected.

Object detection is one of the most important computer vision tasks for autonomous driving. LiDAR sensors enable capturing a 3D point cloud representation of the environment with accurate distance measures, which is crucial for reliable perception and safety considerations. In this application, we choose a one-shot object detector, adapted from YOLO3D [13], to produce oriented bounding boxes and the corresponding class labels in the 2D Bird Eye View (BEV). KITTI dataset [14] is used for the object detection experiments, where it contains LiDAR BEV images from different driving scenarios. For simplicity, a single but common class from the dataset is used which is the car class. This dataset contains 8k training images and 1.4k test images.

Semantic Segmentation is another important computer vision task where the output is pixel-wise classification of the input image. Semantic Segmentation provides a simpler abstraction of the environment in terms of the predefined classes. In this application skin semantic segmentation is conducted as an example of semantic segmentation task, that can be applied for pedestrians and interior monitoring including the passengers and the driver. This task can be used for higher level applications such as gesture recognition and activity recognition. In this experiment, UNet [15] is used with ResNet101 [16] as backbone. Moreover, three classes are considered, the first class is skin class that corresponds to the exposed skin of a person. The second class is clothes class that a person is wearing. The third class is the background. Two datasets are used for the skin segmentation task. The first dataset is MHP dataset [17] that contains multiple persons captured in real-world scenes with pixel-level semantic annotations. For our application we reduced the 58 classes to the three classes. This dataset contains 15k training images and 5k test images. The second dataset is CIHP dataset [18] that contains images with semantic annotations of 19 human-part labels, captured from a different range of viewpoints. Similarly, all of these 19 classes are reduced to the three classes. This dataset contains 28k training images and 5k test images.

Steering angle prediction is one of the classical applications of autonomous vehicles where it is usually based on the front camera image. We follow the Nvidia's self driving car convolutional neural network architecture [19], used to predict the steering angles for a self-driving car directly from the raw pixels of a front camera. For this application, a publicly available dataset [20] is used where it contains images recorded from a car dashcam with labeled steering angles. This dataset contains 36.4k training images and 9k test images.

4 Results

Object Detection: After pruning the object detection network using $l1$, FPGM and TaylorFO algorithms at different sparsity levels of 70%, 80% and 90%, as shown in the Figure 1a, the best performing pruning algorithm is the FPGM that gives mean average precision (mAP) of 0.9067, 0.8825 and 0.766 respectively, where the baseline mAP without pruning is 0.928. The baseline model has 9.2B FLOPS, while the pruned models have relative FLOPS corresponding to the sparsity levels. To visualize the effect of the pruning using different pruning algorithms and different sparsity levels on the performance, figure 2 shows ground truth samples and the corresponding outputs. As expected, at 70% sparsity, most of the pruned networks detect objects that are visually almost identical to the

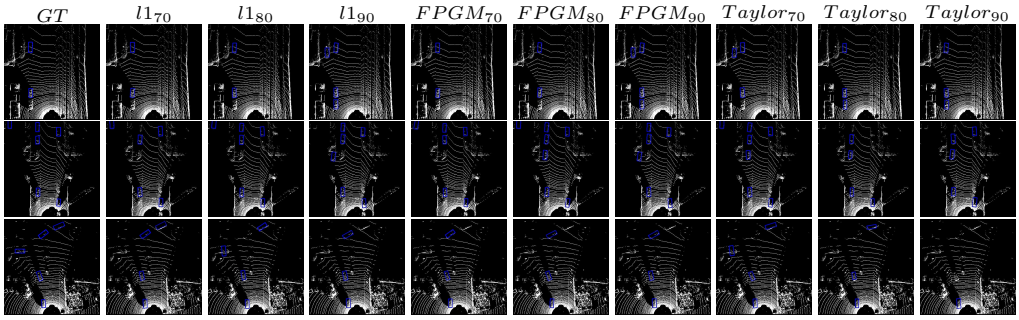


Figure 2: Visual samples for LiDAR object detection.

Figure 3: Visual samples for skin semantic segmentation.

ground truth. However, as the sparsity level is increased, the number of false positives increases in many cases, leading to decreased mAP.

Skin Segmentation: After pruning the network, as shown in Figure 1b, the best performing pruning algorithm is the TaylorFO that gives mean intersection over union (mIOU) of 0.7934, 0.6985 and 0.4242 respectively, where the mIOU without pruning is 0.8032. The baseline model has 53.3B FLOPS. As depicted in Figure 3, we can see an overlay for the segmentation mask with two different colors, green for skin class and blue for the clothes class. At 70% sparsity the semantic segmentation is affected by the pruning, but it is still performing visually well. For the 80% sparsity the number of misclassified pixels increases marginally. However, for the 90% sparsity the model misclassifies most of the pixels including the pixels of the skin and clothes. Furthermore, as shown in the figure, the third sample is affected heavily by the pruning even at 70% sparsity.

Steering angle prediction: After pruning the network, as shown in Figure 1c, the best performing pruning algorithm is the FPGM with mean squared errors (MSE) of 0.076, 0.121 and 0.321 respectively, where the MSE without pruning is 0.058. The baseline model has 26.9M FLOPS. It is noticed that at sparsity 90% the error increased dramatically mainly because the network is relatively small.

5 Conclusions and future work

In this work, structured pruning methods are applied on autonomous driving tasks at different sparsity levels. Generally speaking, by increasing the sparsity levels, the performance of all the applications is decreased as expected. However, object detection seems to be more robust compared to semantic segmentation, mainly because of the nature and complexity of these application and related training data and network architectures. Furthermore, FPGM did not harm the performance of both object detection and steering angle prediction applications, compared to other pruning methods. On the other hand, TaylorFO is found to be the best for skin semantic segmentation. For future work, it is planned to extend this study to conduct more experiments considering more applications, different sensors, more network architectures, multi task learning, sensor fusion and more deep network compression and acceleration methods.

References

[1] WHO. Global status report on road safety 2018, "world health organization. https://apps.who.int/iris/bitstream/handle/10665/276462/9789241565684-eng.pdf. [accessed October-2020].

[2] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International Conference on Machine Learning*, pages 1737–1746, 2015.

[3] Vincent Vanhoucke, Andrew Senior, and Mark Z Mao. Improving the speed of neural networks on cpus. 2011.

[4] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

[5] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

[6] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015.

[7] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[8] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[9] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2019.

[10] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11264–11272, 2019.

[11] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

[12] Microsoft. Neural network intelligence. https://github.com/microsoft/nni. [accessed October-2020].

[13] Waleed Ali, Sherif Abdelkarim, Mahmoud Zidan, Mohamed Zahran, and Ahmad El Sallab. Yolo3d: End-to-end real-time 3d oriented object bounding box detection from lidar point cloud. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Jianshu Li, Jian Zhao, Yunchao Wei, Congyan Lang, Yidong Li, Terence Sim, Shuicheng Yan, and Jiashi Feng. Multiple-human parsing in the wild. *arXiv preprint arXiv:1705.07206*, 2017.

[18] Ke Gong, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 932–940, 2017.

[19] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[20] Sully Chen. Driving datasets. `https://github.com/SullyChen/driving-datasets`. [accessed October-2020].