A Min-Max Optimization Framework for Multi-task Deep Neural Network Compression

Jiacheng Guo, Huiming Sun, Minghai Qin, Hongkai Yu, Tianyun Zhang

Cleveland State University, USA

2024 IEEE International Symposium on Circuits and Systems (ISCAS) | 979-8-3503-3099-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISCAS58744.2024.10557958

***Abstract*—Multi-task learning is a subfield of machine learning in which the data is trained with a shared model to solve different tasks simultaneously. Instead of training multiple models corresponding to different tasks, we only need to train a single model with shared parameters by using multi-task learning. Multi-task learning highly reduces the number of parameters in the machine learning models and thus reduces the computational and storage requirements. When we apply multi-task learning on deep neural networks (DNNs), we need to further compress the model since the model size of a single DNN is still a critical challenge to many computation systems, especially for edge platforms. However, when model compression is applied to multi-task learning, it is challenging to maintain the performance of all the different tasks. To deal with this challenge, we propose a min-max optimization framework for the training of highly compressed multi-task DNN models. Our proposed framework can automatically adjust the learnable weighting factors corresponding to different tasks to guarantee that the task with worst-case performance across all the different tasks will be optimized.**

***Index Terms*—multi-task learning, deep learning, weight pruning, model compression**

I. Introduction

Multi-task learning is a subfield of machine learning in which the data is trained with a shared model to solve different tasks simultaneously [1], [2]. There are various advantages for the use of multi-task learning. Multi-task learning improves the generalization of the model, reduces the training time and it also reduces the computational requirement. The last aspect becomes critical when machine learning models are implemented in platforms with limited computational resource, such as edge platforms. Instead of training multiple models corresponding to different tasks, we only need to train a single model with shared parameters by using multi-task learning. Multi-task learning highly reduces the number of parameters in the machine learning models and thus reduces the computational and storage requirements. For example, there are multiple tasks to be done in real-time in self-driving cars, including object detection and depth estimation. If we train multiple models to solve each task individually, it will include a high computational burden on the platform. However if we use multi-task learning to train a single model with shared parameters for these tasks, we can highly reduce the model size and speed up the inference for the tasks.

Currently, deep neural networks (DNNs) are known as state-of-the-art machine learning models. DNNs have achieved great performance in many different tasks such as image classification [3]–[5], object detection [6], [7] and speech recognition [8], [9]. Meanwhile, the large model size of DNNs becomes a significant burden to the current computing systems. When we apply multi-task learning on DNNs, we need to further compress the model since the model size of a single DNN is still a critical challenge to many computation systems, especially for edge platforms. Model compression techniques such as weight pruning and weight quantization have become an effective way to compress the model size of DNNs. For single-task learning, model compression can highly reduce the size of a DNN model with acceptable performance degradation. However, when model compression is applied on multi-task learning, it is challenging to maintain the performance of all the different tasks. For some easy tasks, they can still perform well when the compression rate of the model is high. But things are different for some difficult tasks, there will be a huge performance degradation on them when we pursue a high model compression rate. It becomes a great challenge on how to balance the performance of different tasks in a highly compressed multi-task DNN model.

The most common method to train a multi-task DNN model is to parameterize the aggregated loss function as a weighted sum of the task-specific loss functions [2]. There are several different approaches to determine the weighting factor corresponding to the loss function of each task, such as weighting by uncertainty [10], [11], weighting by learning speed [12]–[15] and weighting by performance [16], [17]. These approaches perform well on an uncompressed multi-task DNN model. However, there is no approach which can guarantee to balance the performance of different tasks on a highly compressed multi-task DNN model. To deal with this challenge, we propose a min-max optimization framework for the training of multi-task sparse DNN models. In our proposed framework, multi-task learning is defined as a min-max optimization problem in which the object function is the weighted sum of the task-specific loss functions. Different from prior multi-task learning approaches, in our proposed framework the weighting factors associated with different tasks are learnable parameters. The min-max optimization framework can automatically adjust the learnable weighting factors corresponding to different tasks to guarantee that the task with worst-case performance across all the different tasks will be optimized.

Our major contributions in this work can be summarized as follows:

- We propose a min-max optimization framework to bal-

Corresponding author: Tianyun Zhang (t.zhang85@csuohio.edu).

ance the performance of all the tasks when the multi-task DNN model is being highly compressed. The learnable weighting factors corresponding to different tasks can guarantee that the task with worst-case performance across all the different tasks will be optimized.
- In the experiments, we evaluate our proposed method on the multi-task attention network (MTAN) [13] for NYUv2 dataset [18]. Experimental results demonstrate that our proposed method achieves 40× pruning rate on MTAN with negligible precision degradation compared with unpruned model. When the MTAN model is pruned by 60×, our method performs much better on balancing the precision of all the tasks compared with prior methods.

II. Related Work

A. Multi-task Learning

Multi-task learning is a subfield of machine learning in which the data is trained with a shared model to solve different kind of tasks simultaneously. The major advantages of multi-task learning include improving the generalization of the model, reducing the training time and reducing the computational requirement. A general multi-task learning framework is shown in Figure 1.

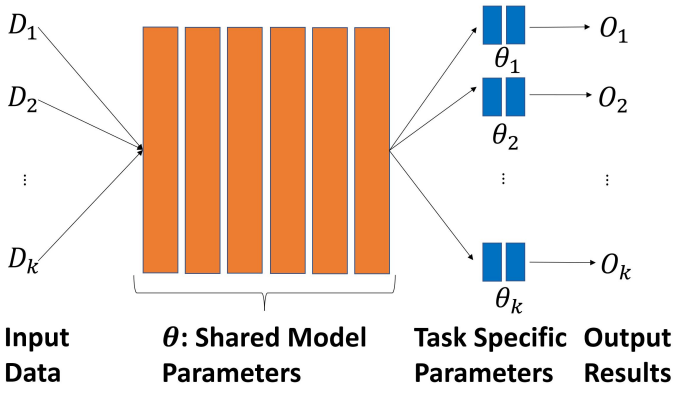

Fig. 1. A general multi-task learning framework.

The most common method to train a multi-task DNN model is to parameterize the aggregated loss function as a weighted sum of the task-specific loss functions [2]. There are several different approaches to determine the weighting factors corresponding to the loss function of each task. The first approach is weighting by uncertainty. For this approach, [10] proposes to treat the multi-task network as a probabilistic model, and it derives the weighting factors associated with different models by maximizing the likelihood of the ground truth output. Based on [10], [11] proposes a revised uncertainty-based method by enforcing positive regularization values in the regularization terms. The second approach is weighting by learning speed. For this approach, [12], [13], [15] set the weighting factor of a task using the ratio of the current loss to the previous loss corresponding to the task. And [14] set the weighting factor of a task by decreasing the weighting factor as learning speed increases. The third approach is weighting by performance. For this approach, [16] proposes to assign the weighting factor to each task according to the defined performance metrics. And [17] proposes to assign the weighting factors based on connection between the training loss and task schedule.

B. Weight Pruning for DNNs

Weight pruning reduces the number of weight parameters in DNNs. A pioneering method of weight pruning is proposed in [19]. This work iteratively prunes the weights with small magnitude and retrain the DNN. The limitation of this work is that it cannot achieve high weight pruning rate with acceptable precision degradation. To address this limitation, several works [20]–[22] propose the optimization-based methods to improve the precision of the DNNs in a high weight pruning rate. And a recent work [23] proposes a weight regularization method based on reweighted l_1 [24] to promote sparsity. This method achieves the state-of-the-art performance on weight pruning and it is applicable to different kind of DNN models, including multi-task DNNs. In this work, we use reweighted l_1 as the weight pruning method to evaluate the performance of different multi-task learning methods under a high weight pruning rate.

III. Problem Formulation and Proposed Framework

A. Problem Formulation

Consider a multi-task DNN with K different tasks, in which the shared parameters are denoted by θ, and the parameters for the i-th specific task are denoted by θ_i. The loss function corresponding to the i-th task can be presented as $F_i(\theta, \theta_i)$. In this work, we propose a min-max optimization framework for multi-task learning, in which we formulate the problem as

$$\underset{\theta,\theta_i}{\text{minimize}} \ \underset{\mathbf{w}\in\mathbf{P}}{\text{maximize}} \sum_{i=1}^{K} w_i F_i(\theta,\theta_i) - \frac{\gamma}{2}\|\mathbf{w}-\mathbf{1}/K\|_2^2, \tag{1}$$

where w_i denotes the learnable weighting factor associated with the i-th task, the vector $\mathbf{w}$ denotes the collection of w_i, $\mathbf{P}$ denotes the probability simplex $\mathbf{P} = \{\mathbf{w} \,|\, \mathbf{1}^T\mathbf{w} = 1, w_i \in [0,1], \forall i\}$, $\mathbf{1}$ denotes an all-ones vector and $\gamma > 0$ is a regularization parameter. The constraint set $\mathbf{P}$ makes sure that every weighting factor is in the range of 0 to 1, and the sum of all the weighting factors equals to 1. The reason to include a regularization term in problem (1) is that it improves the generalizability to different tasks. If the regularization term is not included, the solution of $\mathbf{w}$ is always a one-hot vector. This one-hot coding only focuses on one task and it reduces the generalizability to other tasks and induces instability of the learning procedure in practice.

In order to prune the shared parameters in a multi-task DNN, we add a regularization term in the min-max problem to promote sparsity, the problem becomes

$$\begin{aligned}\underset{\theta,\theta_i}{\text{minimize}} \ \underset{\mathbf{w}\in\mathbf{P}}{\text{maximize}} \quad & \sum_{i=1}^{K} w_i F_i(\theta,\theta_i) - \frac{\gamma}{2}\|\mathbf{w}-\mathbf{1}/K\|_2^2 \\ & +\lambda R(\theta),\end{aligned} \tag{2}$$

where $R(\cdot)$ is the sparsity regularization term and $\lambda > 0$ is the regularization parameter corresponding to it. Our proposed method is applicable to different kinds of sparsity regularization terms. In this work, we choose reweighted l_1 [24], [25] as the regularization term as it achieves the state-of-the-art performance on weight pruning.

B. Problem Solving Algorithm

We apply the alternating projected gradient descent-ascent (APGDA) method [26] to solve the problem (2). Specifically, we use stochastic gradient descent to solve the outer minimization problem, and we use projected gradient ascent to solve the outer maximization problem. The application of APGDA on solving the problem (2) is summarized in Algorithm 1.

Algorithm 1 APGDA to solve the min-max problem

1: Input: given $\mathbf{w}^{(0)}$, $\theta^{(0)}$, $\theta_i^{(0)}$ and epoch numbers T.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: *outer minimization*: fixing $\mathbf{w} = \mathbf{w}^{(t-1)}$, update $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta_i}^{(t)}$ with stochastic gradient descent
4: *inner maximization*: fixing $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$, $\boldsymbol{\theta_i} = \boldsymbol{\theta_i}^{(t)}$, update $\mathbf{w}^{(t)}$ with projected gradient ascent
5: **end for**

In algorithm 1, when $\mathbf{w}$ is fixed, the outer minimization problem is similar to the regular training of a multi-task DNN model with stochastic gradient descent. For the inner maximization problem, $\mathbf{w}$ is updated by

$$\mathbf{w}^{(t)} = \mathrm{proj}_{\mathcal{P}}\Big(\mathbf{w}^{(t-1)} + \beta \nabla_{\mathbf{w}} L(\mathbf{w}^{(t-1)})\Big),$$

where $L(\mathbf{w}) = \sum_{i=1}^{K} w_i F_i(\theta, \theta_i) - \frac{\gamma}{2}\|\mathbf{w} - \mathbf{1}/K\|_2^2$, β is the learning rate, and $\mathrm{proj}_{\mathcal{P}}(\cdot)$ denotes the Euclidean projection onto the simplex set $\mathcal{P}$, the closed-form solution of this kind of Euclidean projection is derived by [27].

After we use APGDA to solve problem (2), we remove the weight parameters which are close to zero (set them to zero), and then retrain the multi-task DNN model using the remaining non-zero weights. In the retraining stage, we solve problem (1) via APGDA.

C. Comparisons between Our Method and Prior Methods

Compared with the prior methods on multi-task learning, the proposed min-max optimization framework is expected to achieve a higher compression rate for an acceptable worst-case performance because it always focuses on optimizing the task with worst performance across all the tasks. The advantage of the proposed min-max framework on improving the worst-case performance will be notable when the DNN compression rate is extremely high. The reason is that prior methods do not include a strategy to maintain the worst-case performance across all the tasks. As the compression rate increases, the easy tasks can still perform well, but the difficult tasks will suffer high performance degradation. For the proposed min-max optimization framework, it always focuses on maintaining the worst-case performance and automatically strikes a balance between different tasks. As the compression rate increases, the min-max framework will sacrifice the performance of some easy tasks in order to maintain the performance of the worst-case task. The model compression rate can be increased until the performances of all the tasks are close to the acceptable thresholds. This leads to a much higher potential in the compression rate with acceptable performance for all the tasks compared with prior approaches.

IV. Experiments

We evaluate our proposed method on the multi-task attention network (MTAN) [13] for NYUv2 dataset [18]. And we compare our method with the prior multi-task learning methods, such as Equal Weights, Weighting by Uncertainty [10], and Dynamic Weighting Average (DWA) [13]. For both our method and the prior methods, we first add a reweighted l_1 regularization term and train the multi-task DNN for 180 epochs in order to promote weight sparsity, with the learning rate set to 10^{-4}. Then a specific ratio of weights is pruned from the MTAN model, and then the model is retrained by 200 epochs for all the methods. At this stage, the learning rate is 10^{-4} at the beginning and it is reduced by half at the 100-th epoch. For each method, we show the average precision of the last 5 epochs in the retraining stage for the fair comparison.

A. Architecture of Multi-task Attention Network

The multi-task attention network (MTAN) is established based on VGG-16 [28] which has two basic components: a shared network which has shared features among all the tasks containing a shared feature pool, and K task-specific attention networks which train the specific features of each individual task. A set of attention modules are settled in each task-specific network which builds the connection between each task and the shared network. Therefore, the features could be learned jointly in the shared network and soft attention masks among multiple tasks, improving the performance of specific tasks simultaneously. Each layer of the shared network has a soft-attention mask to learn task-specific features, which ensures the designed network to be learned in an end-to-end manner.

B. Dataset

The NYUv2 dataset consists of video sequences of various indoor scenes recorded by Microsoft Kinect's RGB and depth cameras. It contains 1449 densely labeled pairs of aligned RGB and depth images, 464 new scenes taken from 3 cities together with 407,024 new unlabeled frames. Each object is labeled with a class and an instance number. In our experiments, three learning tasks are evaluated on MTAN for the NYUv2 dataset: 13-class semantic segmentation, true depth data estimation, and surface normal prediction. The first task is defined in [29] and the last two tasks are provided in [30].

Semantic segmentation is the task of assigning a class label to every single pixel of an input image, such that pixels belonging to the same object or category have the same label. Semantic segmentation operates at the pixel level, providing a fine-grained understanding of the objects and regions within

TABLE I

THE RESULTS OF 3 TASKS (I.E. 13-CLASS SEMANTIC SEGMENTATION, DEPTH ESTIMATION, AND SURFACE NORMAL PREDICTION) ON THE NYUV2 VALIDATION DATASET, WITH 40× PRUNING RATE ON MTAN. THE BEST PERFORMANCE AMONG THE ARCHITECTURES IS HIGHLIGHTED IN BOLD. "(↑)" MEANS HIGHER BETTER AND "(↓)" MEANS LOWER BETTER.

Task	Semantic Segmentation		Depth Estimation		Surface Normal Prediction				
Metrics					Angle Distance		Within t°		
	mIoU(↑)	Pix Acc(↑)	Abs Err(↓)	Rel Err(↓)	Mean(↓)	Median(↓)	11.25(↑)	25(↑)	30(↑)
Unpruned	0.2685	0.5572	0.6902	0.3004	29.9930	25.7138	0.2026	0.4439	0.5762
Equal Weights	0.2324	0.5306	0.6827	0.2833	32.2727	27.8334	0.1683	0.4074	0.5407
Uncertainty [10]	0.2474	0.5416	0.6628	0.2938	31.5592	27.2784	0.1808	0.4168	0.5490
DWA [13]	0.2440	0.5404	0.6484	0.2830	32.1461	28.0508	0.1672	0.4024	0.5361
Min-Max	**0.2502**	**0.5495**	**0.6333**	**0.2681**	**30.1683**	**25.5968**	**0.2034**	**0.4453**	**0.5778**

TABLE II

THE RESULTS OF 3 TASKS (I.E. 13-CLASS SEMANTIC SEGMENTATION, DEPTH ESTIMATION, AND SURFACE NORMAL PREDICTION) ON THE NYUV2 VALIDATION DATASET, WITH 60× PRUNING RATE ON MTAN. THE BEST PERFORMANCE AMONG THE ARCHITECTURES IS HIGHLIGHTED IN BOLD. "(↑)" MEANS HIGHER BETTER AND "(↓)" MEANS LOWER BETTER.

Task	Semantic Segmentation		Depth Estimation		Surface Normal Prediction				
Metrics					Angle Distance		Within t°		
	mIoU(↑)	Pix Acc(↑)	Abs Err(↓)	Rel Err(↓)	Mean(↓)	Median(↓)	11.25(↑)	25(↑)	30(↑)
Unpruned	0.2685	0.5572	0.6902	0.3004	29.9930	25.7138	0.2026	0.4439	0.5762
Equal Weights	0.1574	0.4122	0.8761	0.3123	36.8045	33.5375	0.1060	0.3109	0.4446
Uncertainty [10]	0.1518	0.3978	0.8973	0.3188	36.9686	33.4708	0.1334	0.3344	0.4548
DWA [13]	0.1902	0.4715	0.7443	**0.3126**	33.1194	28.6406	0.1747	0.3999	0.5248
Min-Max	**0.2288**	**0.5319**	**0.7363**	0.3372	**31.7579**	**27.4788**	**0.1811**	**0.4147**	**0.5445**

an image. Each pixel is assigned to one and only one class label, typically representing object categories.

Depth estimation task aims to determine the distance (or depth) of objects in a scene from a 2D image or a sequence of images. The goal of depth estimation is to create a depth map, which assigns a depth value to each pixel in the image, indicating how far that pixel is from the camera. Darker pixels typically represent objects closer to the camera, while lighter pixels represent objects farther away.

Surface normal prediction is a computer vision task that involves estimating the surface normals of objects and scenes in images or 3D point clouds. Surface normals are vectors that are perpendicular to the surface of an object at each point. They describe the orientation of a surface or the direction it faces at every location, which is essential for understanding the geometry and shape of objects in a scene. Surface normal vectors are typically 3D vectors that provide information about the orientation of a surface. In the context of 2D images, surface normals are often represented as vectors pointing out of the image plane.

C. Experimental Results

Table I shows the performance of each task when the weights of the MTAN model are pruned by 40×. We can observe that all the methods perform well on the depth estimation task. This task is not in high demand on the number of parameters. A moderate pruning rate can even improve the performance of this task because overfitting is mitigated. For the semantic segmentation and surface normal prediction tasks, our method performs better than other methods. Moreover, we prune the weights for 40× with negligible precision degradation compared with the unpruned MTAN model.

The experiment results with 60× weight pruning rate on the MTAN model are shown in Table II. Unlike that every method achieves acceptable performance with 40× pruning rate, Equal Weights and Weighting by Uncertainty approaches lead to huge performance degradation for semantic segmentation and surface normal prediction tasks when the pruning rate is 60×. Dynamic Weighting Average (DWA) performs better than the above two approaches, but it still results in notable precision degradation in the segmentation task. Our method performs better than DWA on balancing the three tasks. We achieve higher precision on the semantic segmentation and surface normal prediction tasks by slightly sacrificing the precision of the depth estimation task. Compared with the unpruned MTAN model, our precision degradation is acceptable for all the three tasks when the pruning rate is 60×.

V. CONCLUSION

In this paper, we propose a min-max optimization framework for the training of highly compressed multi-task DNN models. The learnable weighting factors corresponding to different tasks can guarantee that the task with worst-case performance across all the different tasks will be optimized. As a result, we can balance the performance of all the tasks when the multi-task DNN model is being highly compressed. In the experiments, our proposed method achieves 40× weight pruning rate on the MTAN model with negligible precision degradation. When the weights of the MTAN model are pruned by 60×, our method performs much better on balancing the precision of all the tasks compared with prior methods.

VI. ACKNOWLEDGMENT

This research was supported by the National Science Foundation under award CNS-2245765.

REFERENCES

[1] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[2] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," *arXiv preprint arXiv:2009.09796*, 2020.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," *Advances in neural information processing systems*, vol. 26, 2013.

[7] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[8] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2011.

[9] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[10] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[11] L. Liebel and M. Körner, "Auxiliary tasks in multi-task learning," *arXiv preprint arXiv:1805.06334*, 2018.

[12] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *International conference on machine learning*. PMLR, 2018, pp. 794–803.

[13] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1871–1880.

[14] F. Zheng, C. Deng, X. Sun, X. Jiang, X. Guo, Z. Yu, F. Huang, and R. Ji, "Pyramidal person re-identification via multi-loss dynamic training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8514–8522.

[15] S. Liu, Y. Liang, and A. Gitter, "Loss-balanced task weighting to reduce negative transfer in multi-task learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 9977–9978.

[16] M. Guo, A. Haque, D.-A. Huang, S. Yeung, and L. Fei-Fei, "Dynamic task prioritization for multitask learning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 270–287.

[17] S. Jean, O. Firat, and M. Johnson, "Adaptive scheduling for multi-task learning," *arXiv preprint arXiv:1909.06434*, 2019.

[18] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*. Springer, 2012, pp. 746–760.

[19] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.

[20] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient dnns," in *Advances In Neural Information Processing Systems*, 2016, pp. 1379–1387.

[21] T. Zhang, S. Ye, K. Zhang, J. Tang, W. Wen, M. Fardad, and Y. Wang, "A systematic dnn weight pruning framework using alternating direction method of multipliers," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 184–199.

[22] H. Yang, W. Wen, and H. Li, "Deephoyer: Learning sparser neural network with differentiable scale-invariant sparsity measures," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rylBK34FDS

[23] T. Zhang, S. Ye, X. Feng, X. Ma, K. Zhang, Z. Li, J. Tang, S. Liu, X. Lin, Y. Liu *et al.*, "Structadmm: Achieving ultrahigh efficiency in structured pruning for dnns," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 5, pp. 2259–2273, 2021.

[24] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier analysis and applications*, vol. 14, no. 5, pp. 877–905, 2008.

[25] T. Zhang, X. Ma, Z. Zhan, S. Zhou, C. Ding, M. Fardad, and Y. Wang, "A unified dnn weight pruning framework using reweighted optimization methods," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 493–498.

[26] J. Wang, T. Zhang, S. Liu, P.-Y. Chen, J. Xu, M. Fardad, and B. Li, "Adversarial attack generation empowered by min-max optimization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 020–16 033, 2021.

[27] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations (ICLR)*, 2015.

[29] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," *International Conference on Learning Representations (ICLR)*, 2013.

[30] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.