ResNet Can Be Pruned 60×: Introducing Network Purification and Unused Path Removal (P-RM) after Weight Pruning

Xiaolong Ma1, Geng Yuan1, Sheng Lin1, Zhengang Li1, Hao Sun2, Yanzhi Wang1
1Department of Electrical and Computer Engineering, Northeastern University
2Department of Civil and Environmental Engineering, Northeastern University
1{ma.xiaol, yuan.geng, lin.sheng, li.zhen}@husky.neu.edu, yanz.wang@northeastern.edu
2h.sun@northeastern.edu

Abstract—The state-of-art DNN structures involve high computation and great demand for memory storage which pose intensive challenge on DNN framework resources. To mitigate the challenges, weight pruning techniques has been studied. However, high accuracy solution for extreme structured pruning that combines different types of structured sparsity still waiting for unraveling due to the extremely reduced weights in DNN networks. In this paper, we propose a DNN framework which combines two different types of structured weight pruning (filter and column prune) by incorporating alternating direction method of multipliers (ADMM) algorithm for better prune performance. We are the first to find non-optimality of ADMM process and unused weights in a structured pruned model, and further design an optimization framework which contains the first proposed Network Purification and Unused Path Removal algorithms which are dedicated to post-processing an structured pruned model after ADMM steps. Some high lights shows we achieve 232× compression on LeNet-5, 60× compression on ResNet-18 CIFAR-10 and over 5× compression on AlexNet. We share our models at anonymous link http://bit.ly/2VJ5ktv.

I. INTRODUCTION

Structured weight pruning [4] techniques are developed to facilitate weight compression and computation acceleration to solve the high demand for computation and storage resources. In this work, a structured pruning technique is utilized to compress the DNN models, which provides potential advantages for high-parallelism implementation in hardware by eliminating the required weight indices compared with irregular pruning.

However, the accuracy loss problem in structured pruning is inevitable. By adopting ADMM [1], one can achieve the state-of-art structured weight pruning results without post-processing optimization. During post-processing procedure, we propose a novel algorithm to detect and remove the redundant weights which slip away from ADMM pruning, which is caused by the lacks the guarantee on solution feasibility (non-optimality) of ADMM due to the non-convex nature of objective function (loss function).

- We adopt ADMM for efficiently optimizing the non-convex problem and successfully utilized this method on structured weight pruning.
- We design a novel Network Purification and Unused Path Removal (P-RM) algorithm focused on post-processing an ADMM structured pruned model to boost compression rate while maintain accuracy.

II. ADMM MODEL COMPRESSION

Consider an \( N \)-layer DNNs, sets of weights of the \( i \)-th CONV layer are denoted by \( W_i \), respectively. And the loss function associated with the DNN is denoted by \( f(\{W_i\}_{i=1}^N) \). In this paper, \( \{W_i\}_{i=1}^N \) characterize the set of weights from layer 1 to layer \( N \). The overall problem is defined by

\[
\begin{align*}
\text{minimize} \quad & f(\{W_i\}_{i=1}^N), \\
\text{subject to} \quad & W_i \in P_i, \ W_i \in Q_i, \ i = 1, \ldots, N.
\end{align*}
\]

Given the value of \( \alpha_i \), the constraint set is denoted by \( P_i = \{W_i|\text{card}(\text{supp}(W_i)) \leq \alpha_i\} \), where “card” refers to cardinality and “supp” refers to the support set. Elements in \( P_i \) are the solution of \( W_i \) satisfying the number of non-zero elements in \( W_i \) is limited by \( \alpha_i \) for layer \( i \).

The standard ADMM regularized optimization steps are shown as follow, consider a indicator function is utilized to incorporate \( P_i \) into objective functions, which is

\[
g_i(W_i) = \begin{cases} 0 & \text{if } W_i \in P_i \\ +\infty & \text{otherwise} \end{cases} \quad i = 1, \ldots, N
\]

Then original problem (1) can be equivalently rewritten as

\[
\begin{align*}
\text{minimize} \quad & f(\{W_i\}_{i=1}^N) + \sum_{i=1}^N g_i(W_i), \\
\text{subject to} \quad & W_i = Y_i, \ i = 1, \ldots, N.
\end{align*}
\]

Auxiliary variables \( Y_i \) and dual variables \( U_i \) are imported. ADMM decompose problem (3) into simpler subproblems and solve subproblems iteratively until convergence. The augmented Lagrangian formation of problem (3) is

\[
f(\{W_i\}_{i=1}^N) + \sum_{i=1}^N \sum_{k=1}^K \frac{\rho_i}{2}\|W_i - Y_i + U_i\|^2_F
\]

The first term in problem (4) is the differentiable loss function of the DNN, and the second term is a quadratic regularization term of the \( W_i \), which is differentiable and convex, and \( \| : \|^2_F \) denotes Frobenius norm. As a result, subproblem (4) can be solved by stochastic gradient descent algorithm as the original DNN training.

The standard ADMM algorithm steps proceed by repeating, for \( k = 0, 1, \ldots \), the following subproblems iterations:

\[
W_i^{k+1} := \arg \min_{W_i} L_p(\{W_i\}, \{Y_i^k\}, \{U_i^k\})
\]

\[
Z_i^{k+1} := \arg \min_{Z_i} L_p(\{W_i^{k+1}\}, \{Y_i\}, \{U_i^k\})
\]

\[
U_i^{k+1} := U_i^k + W_i^{k+1} - Y_i^{k+1}
\]

which (5) is the proximal step, (6) is projection step and (7) is dual variables update. However, due to the non-convexity of the DNN loss function rather than the quadratic term in our method, the global optimality cannot be guaranteed.
III. NETWORK PURIFICATION AND UNUSED PATH REMOVAL (P-RM)

ADMM weight pruning can significantly reduce weights while maintaining high accuracy. However, does the pruning process really remove all unnecessary weights? From our analysis on the data flow through a network, we find that if a whole filter is pruned, then after General Matrix Multiply (GEMM), the generated feature maps by this filter will be “blank”. If those “blank” feature maps input to next layer, then corresponding input channel weights become abandonable. By the same token, a pruned channel also causes abandonable filter in previous layer.

Suppose $\Lambda_i$ is the number of columns per channel in layer $i$, and $\eta_{i,j}$ is the emptiness ratio. We have

$$\eta_{i,j} = \left[ \sum_{k=1}^{\delta} \| column_k \|^2_F = 0 \right] / \delta \delta \in \Lambda_i$$

If $\eta_{i,j}$ exceed a pre-defined threshold, we can assume that this channel is empty. But this indiscriminate criterion has its limitation. If we remove all columns that satisfy $\eta$, disastrous accuracy drop will occur and hard to recover by retraining because some relatively “important” weights might be removed. To mitigate this problem, we design Network Purification algorithm targeted on dealing with the non-optimality problem of the ADMM process. We set-up an criterion constant $\sigma_{i,j}$ to represent channel $j$’s importance score, which is derived from an accumulation procedure:

$$\sigma_{i,j} = \sum_{k=1}^{\delta} \| column_k \|^2_F / \delta \delta \in \Lambda_i$$

One can think of this process as if collection evidence for whether each channel that contains one or several columns need to be removed. Network Purification also works on purifying remaining filters and thus remove more unused path in the network. Algorithm 1 shows our generalized method of the P-RM method where $Th_1 \ldots Th_4$ are hyper-parameter thresholds values.

Algorithm 1 Network purification & Unused path removal

Result: Redundant weights and unused paths removed

Load ADMM pruned model

$\delta = \text{numbers of columns per channel}$

for $i = 1$ until last layer do

for $j = 1$ until last channel in layer do

for each: $k \in \delta$ and $\| column_k \|^2_F < Th_1$ do

| calculate: equation (8), (9) |

end

if $\eta_{i,j} < Th_2$ and $\sigma_{i,j} < Th_3$ then

| prune(channel$_{i,j}$) |

| prune(filter$_{i-1,j}$) when $i \neq 1$ |

end

for $m = 1$ until last filter in layer do

if filter$_m$ is empty or $\| filter_m \|^2_F < Th_4$ then

| prune(filter$_{i,m}$) |

| prune(channel$_{i+1,m}$) when $i \neq$ last layer index |

end

end

IV. EXPERIMENTAL RESULTS

Figure 1 proves that ADMM’s non-optimality exists in a structured pruned model. By purifying the redundant weights, we can further optimize the loss function. All of the results are based on non-retraining Network Purification process. The purification along with removal of unused path (P-RM) process has great compression boost effect on a deep network.

V. CONCLUSION

In this paper, we provide an ADMM regularized method to achieve highly compressed DNN models with combination of different weight pruning structures, and maintain the network in a high level. We further investigate the post-process of ADMM pruning to solve the non-optimal solution caused by non-convex DNN loss function. We proposed Network Purification and Unused Path Removal that increase our model compression rate significantly.

REFERENCES

[1] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning, 2011.

[2] He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., and Han, S. AMC: AutoML for model compression and acceleration on mobile devices. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 784–800, 2018.

[3] C. Min, A. Wang, Y. Chen, W. Xu, and X. Chen. 2PFPCE: Two-phase filter pruning based on conditional entropy. arXiv preprint:1809.02220, 2018.

[4] Wen, W., Wu, C., Wang, Y., Chen, Y., and Li, H. Learning structured sparsity in deep neural networks. In NeurIPS, pp. 2074–2082, 2016.