# Channel Pruning for Deep Neural Networks via a Relaxed Groupwise Splitting Method

Biao Yang, Jack Xin
*University of California, Irvine*
Irvine, CA, USA.
{biaoy1, jack.xin}@uci.edu

Jiancheng Lyu, Shuai Zhang, Yingyong Qi
*Qualcomm AI Research**
San Diego, CA, USA.
{jianlyu, shuazhan, yingyong}@qti.qualcomm.com

***Abstract*—A relaxed groupwise splitting method (RGSM) is developed and evaluated for channel pruning of deep neural networks. Experiments with VGG-16 and ResNet-18 architectures on CIFAR-10/100 image data show that RGSM can achieve much higher channel sparsity than group Lasso method, while keeping comparable accuracy.**

***Index Terms*—deep neural networks, relaxed groupwise splitting, channel pruning.**

## I. Introduction

Deep convolutional neural networks (CNNs) have made significant advances in computer vision tasks such as image classification, semantic segmentation and object detection. In resource limited situations however, light weight networks are desirable for which pruning methods have been actively studied [3].

In this paper, we propose a Relaxed Group-wise Splitting Method (RGSM) extending the Relaxed Variable Splitting Method (RVSM) [1] to group sparsification of network weights, especially channel pruning. The RGSM utilizes the thresholding formulas of group-Lasso (GLasso) [6], and group-$\ell_0$. We also found that blending RGSM with the direct GLasso [5] can help zero out small weights more effectively than each individual method. Our main contributions are:

- Formulation of RGSM for structured network pruning.
- General applicability of RGSM for discontinuous penalty $\ell_0$ and others with closed form proximal operators.
- Blending RGSM and direct GLasso [5] into an efficient group sparsity method.

The rest of the paper is organized as follows. In Section 2, we discuss some related works. In Section 3, we show our algorithms. The experimental results are in Section 4.

## II. Related Work

In network pruning [3], a major line of work is on structured pruning [5] via group sparsity penalties, most notably Glasso [6]. Besides direct implementation in gradient descent [5], primal-dual like approaches can bring additional efficiency in pruning. In [7], the alternating direction method of multipliers (ADMM) is applied for unstructured weight pruning. In [1], a relaxed variable splitting method (RVSM) is proposed for unstructured sparsity and its convergence is analyzed in a regression problem. In RVSM, thresholding and gradient descent are efficiently integrated to handle non-smooth (discontinuous) penalties for network training. The RVSM is much simpler than ADMM, and so is more computationally appealing for deep network training.

The work was partially supported by NSF grant IIS-1632935, DMS-1854434, a Qualcomm Faculty Award, and Qualcomm AI Research.

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

## III. Algorithm

Let $w = \{w_1, ..., w_g, ..., w_G\}$ be the grouped weights of convolutional layers of a deep network, where $G$ is the total number of groups. Let $I_g$ be the indices of $w$ in group $g$. The GLasso penalty [6] is: $\|w\|_{GL} := \sum_{g=1}^{G} \|w_g\|_2$. Similarly, the group-$\ell_0$ penalty (G-$\ell_0$) is: $\|w\|_{G\ell_0} := \sum_{g=1}^{G} 1_{\|w_g\|_2 \neq 0}$. We obtain the GLasso proximal operator by solving:

$$y_g^* = \text{argmin}_{y_g} \lambda \|y_g\|_2 + \sum_{i \in I_g} \frac{1}{2} \|y_{g,i} - w_{g,i}\|_2^2, \quad (1)$$

and G-$\ell_0$ proximal (projection) operator by solving:

$$y_g^* = \text{argmin}_{y_g} \lambda 1_{\|y_g\|_2 \neq 0} + \frac{1}{2} \sum_{i \in I_g} \|y_{g,i} - w_{g,i}\|_2^2. \quad (2)$$

The solution of $(1)$ is a soft-thresholding operation:

$$y_g^* = \text{Prox}_{GL,\lambda}(w_g) := w_g \ \max(\|w_g\|_2 - \lambda, 0)/\|w_g\|_2 \quad (3)$$

and the solution of (2) is the hard-thresholding operation:

$$y_g^* = \text{Prox}_{G\ell_0,\lambda}(w_g) := w_g \ 1_{\|w_g\|_2 > \sqrt{2\lambda}}. \quad (4)$$

We turn gradient descent update: $w^{t+1} = w^t - \eta \nabla f(w^t)$ via a relaxed group splitting into:

$$u_g^t = \text{Prox}_\lambda(w_g^t), \ \ g = 1, \cdots, G, \quad (5)$$

$$w^{t+1} = w^t - \eta \nabla f(w^t) - \eta \beta (w^t - u^t), \quad (6)$$

where the last term with $\beta$ in (6) is due to relaxation of $u$ into $w$ to facilitate gradient descent as in [1]. Let $\eta$ be the learning rate, $\lambda_1 = \lambda$ and $\lambda_2$ be the GLasso *blending parameter*. The general RGSM is summarized in Alg. 1. If $\lambda_2 = 0$ ($\neq 0$), Alg. 1 is called RGSM (RGSM+GL) for short. The RGSM can be RGSM(GL) or RGSM(G-$\ell_0$) depending on using GLasso or G-$\ell_0$ penalty. The output $u^t$ ($t = max\_epoch$) gives the pruned weights. The $\{w^t\}$'s are auxiliary weights to help compute $\{u^t\}$'s.

978-1-7281-4087-2/19/$31.00 ©2019 IEEE
DOI 10.1109/AI4I.2019.00032

**Algorithm 1:** Relaxed Group-wise Splitting Method

Set hyper-parameters: $\beta, \lambda_1, \lambda_2$.
Define objective function: $f(w) = \text{loss}(w) + \lambda_2 \|w\|_{GL}$.
Randomly initialize $w^0$, define $u^0$, and iterate as:
**for** *g=1,2,...,G* **do**
  $u_g^0 = \text{Prox}_{\lambda_1}(w_g^0)$
**end**
**for** *t =0,1,2,...,max_epoch* **do**
  **for** *batch = 0,1,2,...,max_batch* **do**
    $w^{t+1} = w^t - \eta \nabla f(w^t) - \eta\beta(w^t - u^t)$;
    **for** *g=1,2,...,G* **do**
      $u_g^{t+1} = \text{Prox}_{\lambda_1}(w_g^t)$.
    **end**
  **end**
**end**

TABLE I
ACCURACY (%) AND SPARSITY (%) OF VGG-16 ON CIFAR-10.

| Model | $\beta$ | $\lambda_1$ | $\lambda_2$ | *Accuracy* | *Sparsity* |
|---|---|---|---|---|---|
| Original | 0 | 0 | 0 | 93.94 | 0 |
| GL | 0 | 0 | 1e-4 | 93.62 | 65.9 |
| RGSM(GL) | 1 | 1e-3 | 0 | 93.68 | 69.0 |
| RGSM(GL)+GL | 1 | 1e-3 | 1e-6 | 93.61 | 70.1 |
| RGSM(G-$\ell_0$) | 1 | 4e-2 | 0 | 93.77 | 67.8 |
| RGSM(G-$\ell_0$)+GL | 1 | 4e-2 | 1e-6 | 93.64 | 70.1 |

## IV. Experiments and Results

We compare Alg. 1 with GLasso [5] on CIFAR-10 dataset through VGG-16 [4] and ResNet-18 [2], and on CIFAR-100 through ResNet-18. In training, $\lambda_1$ controls the threshold, and is found to be larger for RGSM(G-$\ell_0$) to be effective.

### A. VGG-16 on CIFAR-10

We train VGG-16 model in 200 epochs, and use SGD as optimizer with momentum 0.9, weight decay 5e-4 and initial learning rate 0.1. The learning rate decays by a factor of 0.1 at the 100th and 160th epochs. We apply Alg. 1 to pruning convolutional layers of the model. The sparsity is measured as the percentage of all channels with $\ell_2$-norm less than 1e-15. Table I shows that both RGSM(GL) and blended RGSM(GL) with Glasso (GL) achieved higher channel sparsity than GL while maintaining the original network accuracy.

### B. ResNet-18 on CIFAR-10 & CIFAR-100

We implemented Alg. 1 on CIFAR-10 and CIFAR-100 with ResNet-18 under the same training condition as VGG-16. In Table II, the blended RGSM(G-$\ell_0$) and GL garnered the highest sparsity under 1% loss of the original accuracy. This can be explained by the observation: while the splitting procedure zeros out channels with $\ell_2$-norm under certain threshold, the blended GLasso helps promote channel differences so more channels with small $\ell_2$-norm appear. Fig. 1 shows the number of channels of each layer in ResNet-18 trained on CIFAR-10.

TABLE II
ACCURACY (%) AND SPARSITY (%) OF RESNET-18 ON CIFAR-10/100.

| Dataset | Model | $\beta$ | $\lambda_1$ | $\lambda_2$ | *Accuracy* | *Sparsity* |
|---|---|---|---|---|---|---|
| CIFAR-10 | Original | 0 | 0 | 0 | 94.97 | 0 |
| | GL | 0 | 0 | 1e-4 | 95.13 | 29.7 |
| | RGSM(GL) | 1 | 1e-3 | 0 | 94.74 | 45.8 |
| | RGSM(GL)+GL | 1 | 1e-3 | 5e-6 | 94.74 | 46.1 |
| | RGSM(G-$\ell_0$) | 1 | 1e-2 | 0 | 95.19 | 35.9 |
| | RGSM(G-$\ell_0$)+GL | 1 | 1e-3 | 5e-6 | 94.87 | 49.7 |
| CIFAR-100 | Original | 0 | 0 | 0 | 77.76 | 0 |
| | GL | 0 | 0 | 1e-4 | 77.52 | 11.2 |
| | RGSM(GL) | 1 | 1e-3 | 0 | 77.03 | 11.1 |
| | RGSM(GL)+GL | 1 | 1e-3 | 5e-6 | 77.47 | 12.7 |
| | RGSM(G-$\ell_0$) | 0.1 | 5e-2 | 0 | 76.93 | 19.7 |
| | RGSM(G-$\ell_0$)+GL | 0.1 | 5e-2 | 1e-6 | 76.88 | 20.3 |

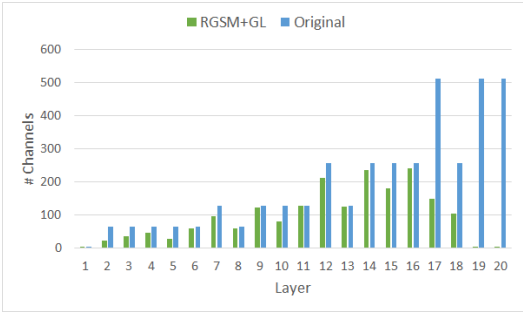


Fig. 1. Layer-wise channel numbers in ResNet-18 before and after RGSM(G-$\ell_0$)+GL pruning on CIFAR-10.

## V. Conclusion

RGSM is developed for structured channel pruning. It outperformed GLasso [5] in the number of pruned channels while maintaining network accuracy. The blended $\ell_0$-version, viz. RGSM(G-$\ell_0$)+GL, achieved most channel sparsity while keeping loss of accuracy under one percent for pruning ResNet-18 on both CIFAR-10 and CIFAR-100. The blending of group-wise splitting and GLasso is found to be effective. In future work, we plan to apply RGSM to object detection neural networks in combination with quantization.

## References


[1] T. Dinh and J. Xin, Convergence of a relaxed variable splitting method for learning sparse neural networks via $\ell_1$, $\ell_0$, and transformed-$\ell_1$ penalties, arXiv:1812.05719, 12 2018.
[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In Proc. CVPR, 2016.
[3] Z. Liu, M. Sun, T. Zhou, G. Huang, T. Darrell, "Rethinking the value of network pruning," ICLR 2019.
[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," In Proc. ICLR, 2015.
[5] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in NIPS, 2016.
[6] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," Journal of the Royal Statistical Society, Series B, 68(1):4967, 2007.
[7] 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," European Conference on Computer Vision (ECCV), 2018.