Supplementary Materials for Learning Compact Neural Networks via Generalized Structured Sparsity

Paper #1766

Three main parts are presented in the supplement:

- Discussion on GSS_{121} and Deep GSS_{121}
- Detailed configuration of the experiments
- More experimental results
  - Results on Fashion-MNIST and MNIST
  - Quantitative analysis on Fashion-MNIST and MNIST
  - Convergence analysis

1 Supplement for GSS

1.1 Formulation of GSS_{121}

In the main paper, we propose a novel sparsity regularizer (namely $\textbf{GSS}_{212}$ in this supplement) that exploits a multi-level grouping structure to enforce hierarchical structured sparsity based on ℓ_2-ℓ_1-ℓ_2-norm. Here we develop an another form of GSS based on ℓ_1-ℓ_2-ℓ_1 and term it $\textbf{GSS}_{121}$. For GSS_{121}, at the first-level ($\mathcal{G}$), inter-group competition and intra-group cooperation are promoted based on the ℓ_1- and ℓ_2-norm, respectively, in order to select or remove an entire group $\mathbf{w}_g$ as a whole; at the second-level ($\mathcal{G}_g$), where $\mathcal{G}_g$ is a subset of the partition of variables in g, competition is promoted within groups g's ($\mathcal{G}_g \subseteq \mathcal{P}(g), \forall g' \in \mathcal{G}_g$) by the ℓ_1-norm, while cooperation is encouraged between groups g's by the ℓ_2-norm. Thus, the $\textbf{GSS}_{121}$ regularizer can be defined as:

$$\Omega_{GSS}(\mathbf{w}) = \sum_{g\in\mathcal{G}} \sqrt{\sum_{g'\in\mathcal{G}_g} ||\mathbf{w}_{g,g'}||_1^2}. \tag{1}$$

Unlike group lasso and exclusive lasso, GSS_{121} in (1) enforces ℓ_1-ℓ_2-ℓ_1 hierarchical structured sparsity[1] based on a multi-level grouping structure. A comparison of group lasso, exclusive lasso and GSS_{121} is illustrated in Fig. 6. Similar with GSS_{212}, as shown in Table 1, GSS_{121} can also flexibly unify different sparse regularizations under different group definitions, making it highly adaptable and suitable for various scenarios.

Table 1. Some widely used regularizations unified by GSS_{121}.

Regularization	Condition in (1)	$\Omega(\mathbf{w})$
Group lasso	$\lvert g'\rvert = 1, \forall g' \in \mathcal{G}_g$	$\sum_g \lvert\lvert\mathbf{w}_g\rvert\rvert_2$
Exclusive lasso	$\lvert\mathcal{G}\rvert = 1$	$\sqrt{\sum_g \lvert\lvert\mathbf{w}_g\rvert\rvert_1^2}$
The ℓ_2 norm	$\lvert\mathcal{G}\rvert = 1, \lvert g'\rvert = 1\,, \forall g' \in \mathcal{G}_g$	$\lvert\lvert\mathbf{w}\rvert\rvert_2$
The ℓ_1 norm (lasso)	$\lvert\mathcal{G}_g\rvert = 1, \forall g \in \mathcal{G}$	$\lvert\lvert\mathbf{w}\rvert\rvert_1$

[1] Unfortunately, the ℓ_1-ℓ_2-ℓ_1 structure prevents us from deriving a similar general form as shown in Theorem 1 of the main paper.

1.2 Deep GSS_{212}

Based on GSS_{121}, we can develop Deep GSS_{121} to simultaneously enforce element-wise sparsity and structured sparsity on two types of DNNs[2], including Multi-Layer Perception (MLP) and Convolutional Neural Networks (CNN).

We first consider to design a sparsity regularizer based on GSS_{121} in (1) for MLP. Let $\mathbf{W}^{(l)} \in \mathbb{R}^{d_{l-1}\times d_l}$ denote the weights of the lth layer, where d_{l-1} and d_l are the number of neurons in the $(l-1)$th and lth layers, respectively. For the jth neuron in the lth layer, its input weights in $\mathbf{w}_{:j}^{(l)}$ or output weights in $\mathbf{w}_{j:}^{(l+1)}$ should compete with each other to survive, while neither $\mathbf{w}_{:j}^{(l)}$ nor $\mathbf{w}_{j:}^{(l+1)}$ should be penalized to zero, otherwise the jth neuron will be removed. It aims to enforce exclusive sparsity in the weights and meanwhile selects or removes the jth neuron from the network by treating its input or output weights as a whole. Therefore, the proposed **Deep** $\textbf{GSS}_{121}$ for MLP is defined as

$$\Omega(\{\mathbf{W}^{(l)}\}) = \sum_{l=1}^{L-1}\sum_{j=1}^{d_l} \sqrt{||\mathbf{w}_{:j}^{(l)}||_1^2 + ||\mathbf{w}_{j:}^{(l+1)}||_1^2}. \tag{2}$$

For CNNs, let $\mathcal{W}^{(l)} \in \mathbb{R}^{d_{l-1}\times d_l\times h_l\times w_l}$ denote the convolutional kernel in the lth layer, where d_{l-1} and d_l are the number of channels in the $(l-1)$th and lth layers, respectively. Similar with (2), we propose to enforce exclusive element-wise sparsity in both $\mathcal{W}_{:j::}^{(l)}$ and $\mathcal{W}_{j:::}^{(l+1)}$, and meanwhile promote cooperation between $\mathcal{W}_{:j::}^{(l)}$ and $\mathcal{W}_{j:::}^{(l+1)}$ to prevent their supports from being zero. In addition, the competition is encouraged between the neurons belonging to the same layers, resulting in the elimination of irrelevant neurons. Thus, the proposed **Deep** $\textbf{GSS}_{121}$ for CNN is defined as

$$\Omega(\{\mathcal{W}^{(l)}\}) = \sum_{l=1}^{L-1}\sum_{j=1}^{d_l} \sqrt{||\mathcal{W}_{:j::}^{(l)}||_1^2 + ||\mathcal{W}_{j:::}^{(l+1)}||_1^2}. \tag{3}$$

In terms of grouping structure, at the first level, we have d_l groups by treating each neuron in the lth layer as a group; and at the second level, we have two groups by treating $\mathbf{w}_{:j}^{(l)}$ ($\mathcal{W}_{:j::}^{(l)}$) or $\mathbf{w}_{j:}^{(l+1)}$ ($\mathcal{W}_{j:::}^{(l+1)}$) as a group. A comparison of Deep GSS_{212} and Deep GSS_{121} for MLP is shown in Fig.2.

2 Detailed Configuration of the Experiments

For experiments on MNIST and Fashion-MNIST, we use LeNet-300-100 and LeNet-5. LeNet-300-100 is a fully connected network with

[2] Similar idea can be applied to other DNN models, such as RNN [5], LSTM [2] and Transformers [4].

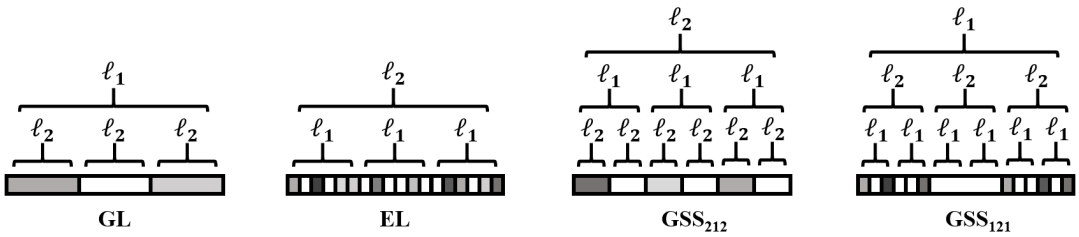
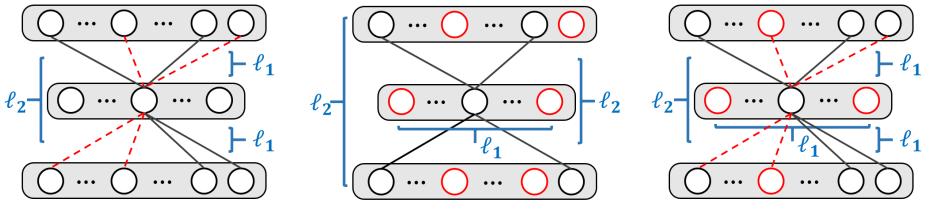


Figure 1. A comparison of Group Lasso (GL), Exclusive Lasso (EL), Generalized Structured Sparsity (GSS) with ℓ_2-ℓ_1-ℓ_2 and GSS with ℓ_1-ℓ_2-ℓ_1. Unlike the single-level grouping structure adopted by GL and EL, GSS uses a multi-level structure to promote cooperation and competition at different levels.



(a) Element-wise Deep GSS_{212} (b) Structured Deep GSS_{212} (c) Deep GSS_{121}

Figure 2. Illustration of Deep GSS_{212} and Deep GSS_{121} for MLP. (a) Weights of a neuron within the same layer compete with each other (pruned weights highlighted in red), while cooperating between successive layers. (b) Neurons within the same layer compete to survive (pruned neurons highlighted in red), while cooperating across all layers.

two hidden layers, each with 300 and 100 neurons. LeNet-5 [3] is a convolutional neural network consisting of two convolutional layers of 5×5 filters with 20 and 50 output channels, two fully-connected hidden layers with 800 and 500 neurons, and two subsampling layers. Both models uses the ReLU (Rectified Linear Unit) as activation function. For model initialization, we use the Kaiming [1] distribution to initialize the weight parameters and use the uniform distribution to initialize the bias parameters.

LeNet-300-100 and LeNet-5 are firstly pretrained without the sparsity-inducing regularizer, where they achieve the testing accuracy of 97.91% and 99.32%, respectively. Then the models are further trained for 250 epochs with the proposed GSS regularizer applied in the objective. Finally, we prune the weights of each layer with a threshold proportional to the standard derivation (std) of each layer's weights. The threshold/std ratio is chosen to achieve the highest sparsity with little performance loss. All weight elements with an absolute value smaller than the threshold are set to zero and are fixed during the final finetuning. The pruned model is finetuned for another 50 epochs without sparsity regularizers, and the best testing accuracy achieved is reported.

For experiments on the CIFAR-10 dataset, we implemented the ResNet56 model and pretrain the it for 164 epochs. During pretraining, the learning rate is initially set to 0.1 and decays by 0.1 at epoch 81 and epoch 122. The pretrained ResNet-56 model achieved a testing accuracy of 93.30%. We start with the pretrained models and train using the GSS-structure regularizer. Both pretraining and training use same learning rate schedule as the experiments on MNIST.

3 Experiments on MNIST and Fashion-MNIST

3.1 Element-Wise Pruning

Tables 2 and 3 summarize the results of LeNet-300-100 on the Fashion-MNIST dataset and LeNet-5 MNIST in terms of prediction accuracy and the number of remained parameters after model pruning. The best results are highlighted in boldface. As shown in Table 2, GSS outperforms comparing methods in testing accuracy and significantly reduces the number of irrelevant weights (by $72\times$). From Table 3, we can see that GSS achieves the highest degree of sparsity (by $149.3\times$) with a slight loss in performance. Among all methods, GSS achieves competitive prediction accuracy and the highest overall sparsity.

3.2 Structure Pruning

Tables 4 and 5 report the results of LeNet-300-100 on the Fashion-MNIST dataset and LeNet-5 MNIST in terms of testing accuracy, remained FLOPs and model structure after pruning, with the best results highlighted in boldface. As shown in the Tables, GSS-structure reduces FLOPs by $8.2\times$ and $11.7\times$ on LeNet-300-100 and LeNet-5, respectively, achieves higher accuracy than most comparing methods, and has a significant compact effect. Group-GReg needs to model the dependency between layers, which introduces additional computational overhead, and when the network scale is large, the dependency graph becomes too complex, making it difficult to achieve effective pruning. Considering that GSS requires only sparse auxiliary selection variables and has linear complexity w.r.t. the data size, we conclude that the GSS regularizer can achieve sparse DNN models at lower cost.

3.3 Quantitative analysis on MNIST and Fashion-MNIST

3.3.1 Evaluation of Pruning Effect

Figure 3 shows the performance of five comparing methods by varying the remained number of parameters and remained FLOPs during model pruning.

Table 2. Element-wise pruning results of LeNet-300-100 on Fashion-MNIST.

Method	Accuracy(%)	Total	FC1	FC2	FC3
Original	90.29	266.2k	235.2k	30k	1k
l_1-norm	**89.20**	7.2k(2.70%)	5.7k(2.42%)	1.3k(4.33%)	136(13.6%)
Hoyer	89.12	5.0k(1.88%)	4.8k(2.04%)	**179(0.60%)**	**50(5.00%)**
Hoyer Square	89.16	4.7k(1.78%)	4.0k(1.70%)	553(1.84%)	118(11.8%)
Transformed ℓ_1	89.19	5.9k(2.52%)	2.6k(22.60%)	3.1k(4.00%)	150(15.00%)
Spred	89.11	5.4k(2.03%)	4.9k(2.09%)	218(0.73%)	282(28.20%)
GSS-element	**89.20**	**3.7k(1.39%)**	**2.3k(0.98%)**	1.1k(0.37%)	148(14.80%)

Table 3. Element-wise pruning results of LeNet-5 on MNIST.

Method	Accuracy(%)	Total	CONV1	CONV2	FC1	FC2
Original	99.32	430.5k	500	25k	400k	5k
Hoyer	99.20	5.0k(1.16%)	**60(30.00%)**	744(5.76%)	4.0k(1.00%)	166(3.32%)
Hoyer Square	99.24	3.5k(0.82%)	82(16.40%)	829(3.32%)	2.4k(0.59%)	245(4.90%)
Transformed ℓ_1	**99.33**	3.7k(0.86%)	91(18.20%)	622(2.48%)	2.8k(0.69%)	193(3.86%)
ADMM-Pruning	99.12	6.1k(1.42%)	100(20.00%)	2k(8.00%)	3.6k(0.90%)	350(7.00%)
SNIP	98.98	21.5k(5.00%)	391(78.2%)	6.3k(25.30%)	12.9k(3.21%)	2.0k(40.10%)
Spred	99.09	4.1k(0.95%)	96(19.2%)	675(2.70%)	3.0k(0.76%)	213(4.26%)
GSS-element	99.26	**2.9k(0.67%)**	95(19.00%)	**281(1.12%)**	**2.3k(0.57%)**	**175(3.50%)**

Table 4. Structure pruning results of LeNet-300-100 on Fashion-MNIST.

Method	Accuracy(%)	#FLOPs	Pruned Structure
Original	90.29	266.2k	784-300-100
Group Lasso	**89.56**	94.2k	775-121-13
Group HS	89.43	50.8k	400-92-24
Group-GReg	89.52	53.1k	784-67-7
Group Pruner	89.38	52.6k	784-64-33
GSS-structure	89.51	**32.6k**	760-39-60

Table 5. Structure pruning results of LeNet-5 on MNIST.

Method	Accuracy(%)	#FLOPs	Pruned Structure
Original	99.32	2293K	20-50-800-500
Group HS	99.23	233.3K	9-7-199-24
Bayes ℓ_{1trim}	99.00	334.0K	8-17-53-19
PFPnet	99.16	177.1K	5-9-139-82
SoftNet	99.18	252.2K	5-37-9-206
SHSQ-GL$_{1/2}$	98.99	731.1k	16-35-18-231
Group-GReg	99.20	220.9k	4-20-320-107
Group Pruner	**99.24**	224.6k	5-18-288-29
GSS-structure	**99.24**	**168.1k**	4-15-240-58

3.3.2 Sensitivity Analysis on λ

Fig. 4 shows the results of sensitivity analysis of GSS on Fashion-MNIST and MNIST. The hyperparameter λ of GSS controls the strength of the sparisty regularization. Larger λ increases the sparsity of the model but significantly decreases the accuracy. On the other hand, a lower λ has a smaller impact on model accuracy but also does not introduce too much sparsity and the accuracy will sharply decrease when the sparsity increases. In general, it is recommended to set $\lambda \leq 10^{-5}$ and $\lambda \geq 10^{-6}$ on Fashion-MNIST and MNIST.

3.3.3 Comparison of Different Variants of Deep GSS

In this experiment, we compare four variants of Deep GSS by selecting $p, q \in \{1, 2\}$ in the general form formulated in (7). Note that p, q actually controls the strength of sparsity of Deep GSS. Fig. 5 shows the results for GSS-structure over LeNet-300-100 trained on MNIST. Generally, the model performs the best in accuracy when $p = 1, q = 2$, which shows the importance on learning a well-performing deep model with an appropriate degree of sparsity.

4 Convergence Analysis

In this experiment, we analyze the empirical convergence rate of our regularized network on MNIST and Fashion-MNIST, and set the maximum epoch as 250. As shown in 6, the algorithm can converge to a stable state with a high convergence rate.

References

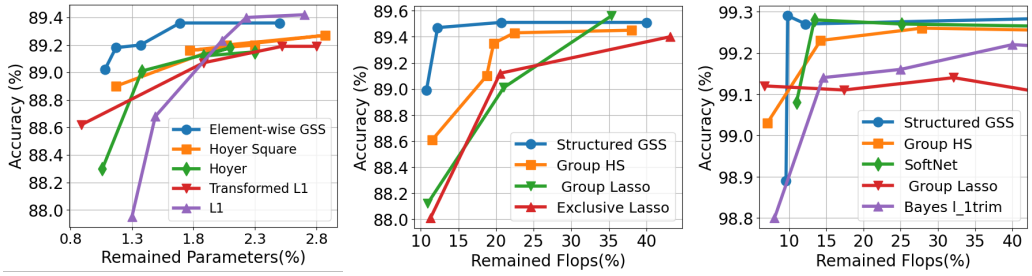
[1] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[2] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[3] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

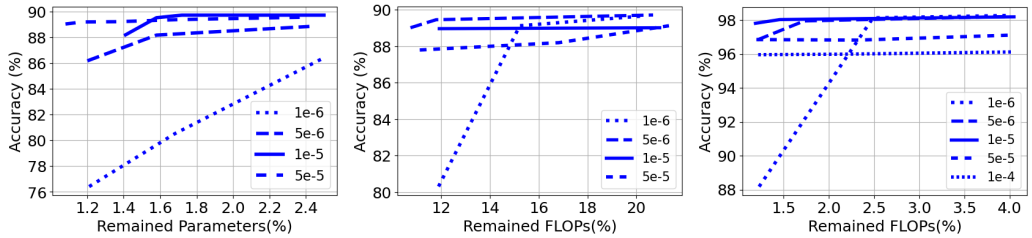
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[5] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015.



(a) GSS-element on Fashion-MNIST (b) GSS-structure on Fashion-MNIST (c) GSS-structure on MNIST

Figure 3. Illustration of accuracy-efficiency trade-off on Fashion-MNIST by varying the number of parameters (for element-wise sparsity) in (a) and FLOPs (for structured sparsity) in (b) and on MNIST in (c).



(a) GSS-element on Fashion-MNIST (b) GSS-structure on Fashion-MNIST (c) GSS-structure on MNIST

Figure 4. Hyperparameter sensitivity analysis of GSS on Fashion-MNIST.

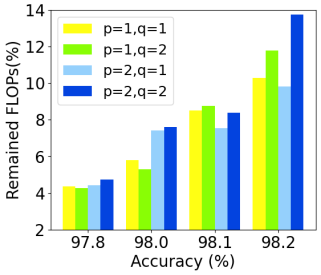
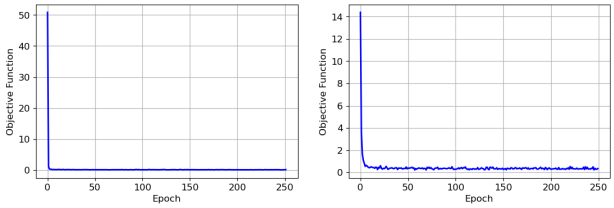


Figure 5. Comparison of four variants of GSS in terms of structured pruning over LeNet-300-100 trained on MNIST.



(a) MNIST (b) Fashion-MNIST

Figure 6. Convergence analysis on the MNIST and Fashion-MNIST datasets. The LeNet-300-100 model is regularized by the element-wise Deep GSS with $\lambda = 10^{-5}$.