Hierarchical Group Sparse Regularization for Deep Convolutional Neural Networks

Kakeru Mitsuno
School of Engineering
Hiroshima University
Higashi Hiroshima, Japan
mitsunokakeru@gmail.com

Junichi Miyao
Department of Information Engineering
Hiroshima University
Higashi Hiroshima, Japan
miyao@hiroshima-u.ac.jp

Takio Kurita
Department of Information Engineering
Hiroshima University
Higashi Hiroshima, Japan
tkurita@hiroshima-u.ac.jp

***Abstract*—In a deep neural network (DNN), the number of the parameters is usually huge to get high learning performances. For that reason, it costs a lot of memory and substantial computational resources, and also causes overfitting. It is known that some parameters are redundant and can be removed from the network without decreasing performance. Many sparse regularization criteria have been proposed to solve this problem. In a convolutional neural network (CNN), group sparse regularizations are often used to remove unnecessary subsets of the weights, such as filters or channels. When we apply a group sparse regularization for the weights connected to a neuron as a group, each convolution filter is not treated as a target group in the regularization. In this paper, we introduce the concept of hierarchical grouping to solve this problem, and we propose several hierarchical group sparse regularization criteria for CNNs. Our proposed the hierarchical group sparse regularization can treat the weight for the input-neuron or the output-neuron as a group and convolutional filter as a group in the same group to prune the unnecessary subsets of weights. As a result, we can prune the weights more adequately depending on the structure of the network and the number of channels keeping high performance. In the experiment, we investigate the effectiveness of the proposed sparse regularizations through intensive comparison experiments on public datasets with several network architectures.**

***Index Terms*—group sparse regularization, convolutional neural network, image classification, pruning**

I. Introduction

Interest in methods of enforcing the network sparsity is increasing in the field of deep neural networks (DNN). By making the network sparse, we can reduce the necessary computational resources, and improve the generalization performance of the trained network.

Tibshirami [1] proposed the most simple non-structural sparse regularization lasso. Zou and Hastie [2] also proposed an elastic net that combined L2 regularization and L1 regularization as a weighted sum. Yuan and Lin [3] and Schmidt [4] proposed group lasso regularization to neglect a group of parameters in the model. Kim and Xing [5] proposed tree-guided group lasso, which is based on group lasso, but groups are defined for a tree structure for a sparse multi-task regression. Friedman et al. [6] and Simon et al. [7] proposed sparse group lasso for linear regression, which combines L1 regularization and group lasso.

This work was partly supported by JSPS KAKENHI Grant Number 16K00239.

Recently, some methods of pruning unnecessary weights of deep neural networks were proposed by many researchers. Wen et al. [8] proposed a structured sparsity learning (SSL) method to regularize the structures of deep neural networks. SSL can learn a compact structure from a bigger DNN to reduce computation cost, obtain a hardware-friendly structured sparsity of DNN, and regularize the DNN to improve classification accuracy.

Alvarez and Salzmann [9] introduced an approach to automatically determine the number of neurons in each layer of a DNN during learning by using sparse group regularization. This method can reduce the number of parameters by up to 80% while retaining or even improving the network accuracy. Scardapane et al. [10] also proposed group sparse regularization for deep neural networks.

Zhou et al. [11] and Kong et al. [12] proposed exclusive lasso. Exclusive lasso introduces competition among variables in the same group. Yoon and Hwang et al. [13] proposed a combined group and exclusive sparsity (CGES) for deep neural networks. CGES enforces the network to be sparse and removes any redundancies in the features to fully utilize the capacity of the network.

Xu et al. [14] [15] [16] proposed $L_{1/2}$ regularization. $L_{1/2}$ regularization can enforce the network to be more sparse than L1 regularization and much simpler than L0 regularization. Fan et al. [17] [18] applied $L_{1/2}$ regularization for sparsification of hidden layers of feed forward neural networks. Li et al. [19] [20] proposed group $L_{1/2}$ regularization for feed forward neural networks.

Li et al. [21] proposed Out-In-Channel Sparse Regularization (OICSR) for compact deep neural networks. Ma et al. [22] proposed non-convex integrated transformed L1 regularization for learning sparse deep neural networks. This method simultaneously promotes connection-level and neuron-level sparsity for DNNs.

These regularizations can make the weights of a deep neural network sparse at the individual weight level and the grouped weights level. In a convolutional neural network (CNN), we can consider the weights of a neuron as a group. However, we can also consider each convolution filter as a group. To treat these groupings simultaneously, we have to introduce the concept of the hierarchical grouping. In this paper, we propose

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

several hierarchical group sparse regularization criteria for deep neural network pruning and evaluate the performance of regularization criteria through intensive comparison experiments.

II. Related Works

In this section, we review previous works on weight pruning methods of deep neural networks in terms of the pruning criteria.

Assume that we have a training set with N instances $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ is a d-dimensional input feature vector and $y_i \in \{1, \ldots, K\}$ is a class label from one of the K classes. Then the objective function with sparse regularization for a deep neural network, especially for classification by convolutional neural networks (CNNs), can be represented as

$$J(W) = loss(W|D) + \lambda \sum_{l=1}^{L} R(W^l) \tag{1}$$

where $loss(W|D)$ is the standard loss for the CNN, W is the set of trainable weights for all L layers in the CNN and $R(W^l)$ is the regularization term at l^{th} layer for pruning the set of weights, $\{W^l\}$. The parameter λ is used to balance the loss and the pruning criterion. If the l^{th} layer is fully connected, we assume that the weight is given by $W^l \in \mathbb{R}^{oc_l \times ic_l}$, where oc_l and ic_l are the dimensions of W^l along the axes of out-channel and in-channel respectively. Also, we assume the weights as $W^l \in \mathbb{R}^{oc_l \times ic_l \times H_l \times W_l}$ when the l^{th} layer is a convolutional layer, where H_l and W_l are the height and width of the kernel respectively.

The most often used sparse regularization is L2 regularization, defined as $\|W^l\|_2^2$. This regularization is often used in deep neural networks as weight decay to suppress over fitting.

Tibshirami [1] proposed a simple non-structural sparse regularization as an L1 regularization for a linear model, which is defined as $\|W^l\|_1$. L1 regularization prevents overfitting by neglecting individual parameters in both convolution layers and fully connected layers. However, with L1 regularization, it is difficult to remove subsets of weights such as filters or channels in a CNN.

A. Group Lasso Regularization

Yuan and Lin [3] and Schmidt [4] proposed group lasso regularization. In order to reduce subsets of weights like filters or channels, it is necessary to treat the subsets as groups in the regularization criterion. Yuan and Lin [3] and Schmidt [4] proposed this regularization for a linear model that can treat sets of parameters as a group in the criterion. Group lasso forces subsets of unnecessary parameters to be simultaneously zero. The regularization criterion of group lass is defined as

$$R_{GL}(W^l) = \sum_{g \in G} \|W_g^l\|_2 = \sum_{g \in G} \sqrt{\sum_i {w_{g,i}^l}^2}, \tag{2}$$

where $g \in G$ is a group in the set of groups G, W_g^l is the weight matrix or the weight vector for the group g that is a sub matrix or sub vector in W^l and $w_{g,i}^l$ is a weight with index i in the group g. Group lasso introduces sparseness at the group level and can reduce the number of active neurons or active filters. Alvarez et al. [9] proposed an approach to automatically determine the number of neurons in each layer of a DNN during learning, and they showed that group lasso regularization could reduce the number of parameters and even improve network accuracy. Wen et al. [8] proposed a structured sparsity learning (SSL) method to regularize the structures of deep neural networks by group lasso as structured sparse regularization. They introduced several structures of group lasso.

B. Sparse Group Lasso Regularization

Friedman et al. [6] and Simon et al. [7] proposed sparse group lasso by combining L1 regularization and group lasso, applied to linear regression. Sparse group lasso forces parameters to be zero at both the group and the individual feature level. Scardapane et al. [10] proposed to use sparse group lasso for deep neural networks. The criterion of the sparse group lasso is written as

$$R_{SGL}(W^l) = \alpha \sum_{g \in G} \|W_g^l\|_2 + (1-\alpha)\|W^l\|_1, \tag{3}$$

where α is a balancing parameter to control strength of both group lasso and L1 regularization. By this combination, unnecessary parameters in the network can be pruned at both the group level and the individual feature level.

C. Exclusive Sparse Regularization

Zhou et al. [11] and Kong et al. [12] proposed exclusive lasso for multi-task feature selection. Exclusive lasso introduces competition among parameters in the same group and can prune neurons in neural networks. It is also called exclusive sparsity and the regularization criterion is defined as

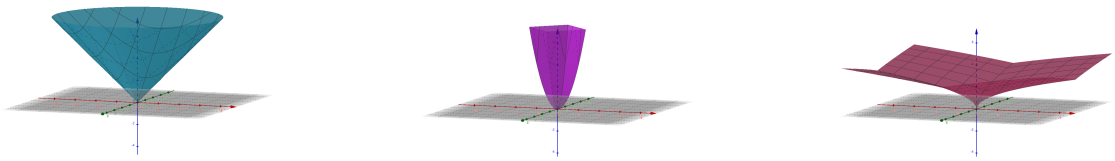
$$R_{ES}(W^l) = \frac{1}{2} \sum_{g \in G} \|W_g^l\|_1^2 = \frac{1}{2} \sum_{g \in G} \left(\sum_i |w_{g,i}^l| \right)^2. \tag{4}$$

D. Combined Group and Exclusive Sparse Regularization

Yoon and Hwang et al. [13] proposed a pruning criterion called combined group and exclusive sparsity (CGES) for deep neural networks, which combines group lasso and exclusive sparse regularization. The authors claim that CGES can make the network sparse and also remove any redundancies among the features to fully utilize the capacity of the network.

E. Group $L_{1/2}$ Regularization

$L_{1/2}$ regularization, proposed by Xu et al. [14] [15] [16], can make the network to be more sparse than L1 regularization and much simpler than L0 regularization. Fan et al. [17] [18] applied $L_{1/2}$ regularization for pruning the neurons in the hidden layer of feedforward neural networks. Li et al. [19] [20] also applied a group $L_{1/2}$ regularization for feedforward neural networks. $L_{1/2}$ regularization can make not only the redundant hidden nodes to be zero but also the redundant



(a) group lasso (b) exclusive sparsity (c) group $L_{1/2}$ regularization

Fig. 1: The shape of three different regularization terms (a) group lasso regularization (b) exclusive sparse regularization (c) group $L_{1/2}$ regularization

weights of the surviving hidden nodes of the neural networks to be zero. In this paper, we define the criterion of the group $L_{1/2}$ regularization for deep neural network as

$$R_{GL_{1/2}}(W^l) = \sum_{g \in G} \|W_g^l\|_1^{1/2} = \sum_{g \in G} \sqrt{\sum_i |w_{g,i}^l|}. \quad (5)$$

F. Out-In-Channel Sparse Regularization

Li et al. [21] proposed Out-In-Channel Sparse Regularization (OICSR) for compact deep neural networks. In OICSR, the correlations between successive layers are taken into consideration to keep the predictive power of the compact network.

III. Proposed Method

A. Structured sparse regularization

In this paper, we investigate the effectiveness of the structured sparse regularization criteria such as group lasso, exclusive sparsity, and group $L_{1/2}$ regularization for the convolutional neural network through intensive comparison experiments. The definitions of these regularization criteria are shown as equations (2), (4), and (5) respectively. The visualization of these functions are shown in Fig. 1.

SSL proposed by Wen et al. [8] introduces various ways of grouping for structured sparse regularization. In this paper, we also investigate the effectiveness of the ways of grouping through intensive comparison experiments. In the following explanations, we will show the ways of grouping by using the criteria for group lasso, but we can also define the criteria for exclusive sparsity and group $L_{1/2}$ regularization.

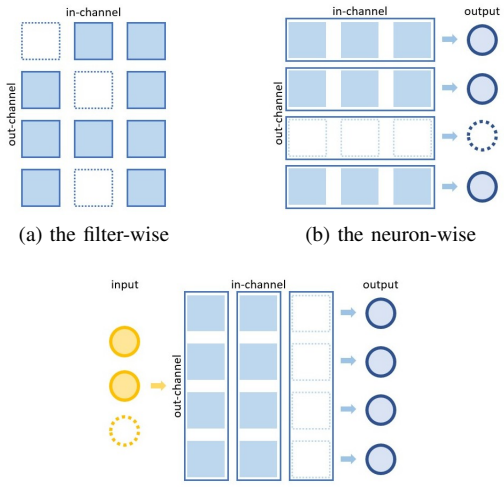
In the case of a convolutional layer, we can consider three types of grouping for structured sparse regularization. The way of grouping for a convolutional layer are shown in Fig. 2. The first one is the filter-wise grouping which is defined as

$$R_{GL}(W^l) = \sum_{i=1}^{oc_l} \sum_{j=1}^{ic_l} \sqrt{\sum_{h=1}^{H_l} \sum_{w=1}^{W_l} {w_{i,j,h,w}^l}^2}. \quad (6)$$

This criterion prunes unnecessary filters in the convolution layers.

The second one is the neuron-wise grouping which is defined as

$$R_{GL}(W^l) = \sum_{i=1}^{oc_l} \sqrt{\sum_{j=1}^{ic_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} {w_{i,j,h,w}^l}^2}. \quad (7)$$



(a) the filter-wise (b) the neuron-wise

(c) the feature-wise

Fig. 2: The way of grouping for convolutional filters. (a) Each filter is considered as a group. We call this grouping the filter-wise grouping. By this grouping, we can prune unnecessary filters. (b) The weights connected to a output neuron are consider as a group. We call this grouping the neuron-wise grouping. By this grouping, we can prune unnecessary output neurons. (c) The weights connected to a input neuron are considered as a group. We call this grouping the feature-wise grouping. By this grouping, we can prune unnecessary the output channels in $(l-1)^{th}$ layer (the input channels in $(l)^{th}$ layer).

This criterion prunes unnecessary output neurons at each convolution layer. As a result, the number of out-channels is reduced in each convolution layer.

Last one is the feature-wise grouping which is defined as

$$R_{GL}(W^l) = \sum_{j=1}^{ic_l} \sqrt{\sum_{i=1}^{oc_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} {w_{i,j,h,w}^l}^2}. \quad (8)$$

This criterion prunes unnecessary input neurons of the convolutional layer. As a result, we can remove unnecessary out-channel in $(l-1)^{th}$ layer by making the unnecessary input neurons in l^{th} layer zero.

B. Hierarchical Group Sparse Regularization

In the fully connected layer, weights are not structured, and we can apply sparse regularization to prune unnecessary input neurons or output neurons.

On the other hand, the weights of the convolutional layer are structured as convolution filters, and there are three types of grouping, namely the filter-wise grouping, the neuron-wise grouping and the feature-wise grouping.

However, mutual interaction between filters in the group is not taken into account in the neuron-wise grouping or the feature-wise grouping. To introduce such interactions in the sparse regularization criterion, we propose hierarchical group sparse regularization.

There are several possibilities to define the hierarchical interactions between filters in the group for structured sparse regularization. In this paper, we consider two ways of the integration, namely the square root of the sub-groups and the square of the sub-groups. Thus, we can propose a set of hierarchical group sparse regularization criteria using group lasso regularization, exclusive sparsity, and group $L_{1/2}$ regularization based on the neuron-wise grouping or the feature-wise grouping. In the following, we explain the hierarchical group sparse regularization criteria based on the feature-wise grouping. However, we can easily define the hierarchical group sparse regularization criteria based on the neuron-wise grouping.

The hierarchical group lasso regularization criterion based on the feature-wise groupings is defined as

$$R_{HSQRT-GL}(W^l) = \sum_{j=1}^{ic_l} \sqrt{\sum_{i=1}^{oc_l} \sqrt{\sum_{h=1}^{H_l} \sum_{w=1}^{W_l} {w^l_{i,j,h,w}}^2}}. \quad (9)$$

In this criterion, the square root of the sub-groups (the feature-wise groupings) are taken to defined the sparse regularization criterion.

The hierarchical group lasso regularization criterion based on the feature-wise groupings is also defined by taking the square of the sub-groups as

$$R_{HSQ-GL}(W^l) = \sum_{j=1}^{ic_l} \left(\sum_{i=1}^{oc_l} \sqrt{\sum_{h=1}^{H_l} \sum_{w=1}^{W_l} {w^l_{i,j,h,w}}^2} \right)^2. \quad (10)$$

It is expected that these hierarchical group lasso criteria can prune unnecessary output neurons and input neurons simultaneously.

Similarly, the hierarchical exclusive sparse regularization criterion is define as

$$R_{HSQRT-ES}(W^l) = \sum_{j=1}^{ic_l} \sqrt{\sum_{i=1}^{oc_l} \left(\sum_{h=1}^{H_l} \sum_{w=1}^{W_l} |w^l_{i,j,h,w}| \right)^2} \quad (11)$$

and

$$R_{HSQ-ES}(W^l) = \sum_{j=1}^{ic_l} \left(\sum_{i=1}^{oc_l} \left(\sum_{h=1}^{H_l} \sum_{w=1}^{W_l} |w^l_{i,j,h,w}| \right)^2 \right)^2 \quad (12)$$

The hierarchical group $L_{1/2}$ regularization criterion is also defined as

$$R_{HSQRT-GL_{1/2}}(W^l) = \sum_{j=1}^{ic_l} \sqrt{\sum_{i=1}^{oc_l} \sqrt{\sum_{h=1}^{H_l} \sum_{w=1}^{W_l} |w^l_{i,j,h,w}|}} \quad (13)$$

and

$$R_{HSQ-GL_{1/2}}(W^l) = \sum_{j=1}^{ic_l} \left(\sum_{i=1}^{oc_l} \sqrt{\sum_{h=1}^{H_l} \sum_{w=1}^{W_l} |w^l_{i,j,h,w}|} \right)^2. \quad (14)$$

It is also possible to combine the L1 regularization with the hierarchical group sparse regularization criteria in order to prune unnecessary individual weights.

In the next section, we investigate the effectiveness of the each criterion through intensive comparison experiments.

IV. Experiments

A. The Sparse Regularization Criteria

We have performed experiments with the convolutional neural network to compare the effectiveness of the sparse regularization criteria explained in this paper. They are summarized in the table I. The regularization is applied to the weights except for the bias term in all convolutional layers.

TABLE I: Summary of the sparse regularization criteria

abbreviation	sparse regularization criteria
L2	L2 regularization
L1	L1 regularization [1]
GL	Group lasso regularization [3] [4]
ES	Exclusive sparse regularization [11] [12]
$GL_{1/2}$	Group $L_{1/2}$ regularization [19] [20]
SGL	Sparse group lasso regularization [10]
$SGL_{1/2}$	Combined $GL_{1/2}$ and L1
CGES	CGES regularization [13]
OICSR-GL	Combined OICSR and GL [21]
HSQRT-GL	Hierarchical square rooted GL
HSQ-GL	Hierarchical squared GL
HSQRT-ES	Hierarchical square rooted ES
HSQ-ES	Hierarchical squared ES
$HSQRT\text{-}GL_{1/2}$	Hierarchical square rooted $GL_{1/2}$
$HSQ\text{-}GL_{1/2}$	Hierarchical squared $GL_{1/2}$
$SHSQRT\text{-}GL_{1/2}$	Combined $HSQRT\text{-}GL_{1/2}$ and L1
$SHSQ\text{-}GL_{1/2}$	Combined $HSQ\text{-}GL_{1/2}$ and L1

B. Networks and Datasets

To confirm the robustness to the variations of the characteristics of the data, we have performed experiments using five datasets MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100, and STL-10. A simple CNN, AlexNet [23], ResNet [24] and VGG nets [25] are used as the base network and they are trained from the scratch. The number of channels of

the network at each layer is adjusted to prevent overfitting, depending on each dataset.

MNIST contains 70,000 grayscale images of handwritten digits. So the number of classes is 10. The size of the image is 28×28 pixels. They are divided into 60,000 training images and 10,000 testing images. The simple CNN with two convolutional layers and two fully connected layers is trained by using the training images of MNIST dataset.

Fashion-MNIST contains 70,000 grayscale images of ten different fashion items. The size of the image is 28×28 pixels. They are divided into 60,000 training images and 10,000 testing images. Similar to the MNIST, the simple CNN with two convolutional layers and three fully connected layers is trained by using the training images of Fashion-MNIST datasets.

CIFAR-10 contains 60,000 color images of ten different animals and vehicles. The size of the image is 32×32 pixels. They are divided into 50,000 training images and 10,000 testing images. For CIFAR-10, we trained AlexNet with 5 convolutional layers and three fully connected layers and ResNet18 with 17 convolutional layers and one fully-connected layer with batch normalization layers. The number of channels of AlexNet is set to [8, 16, 32, 16, 16, 128, 128, 10] and [8, 8, 8, 8, 8, 16, 16, 16, 16, 32, 32, 32, 32, 64, 64, 64, 64, 10] for ResNet18.

CIFAR-100 contains 60,000 color images of 100 different categories. The size of the image is 32×32 pixels. They are divided into 50,000 training images and 10,000 testing images. For CIFAR-100, we trained VGG11bn, that has eight convolutional layers and three fully connected layers with batch normalization layers. The number of channels is set to [16, 32, 64, 64, 128, 128, 128, 128, 128, 128, 10].

STL-10 contains 13,000 color images of animals and vehicles. (airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck). The size of the image is 96×96 pixels. They are divided into 5,000 training images and 8,000 testing images. For STL-10 dataset, we trained AlexNet with five convolutional layers and three fully connected layers and ResNet18 with 17 convolutional layers and one fully-connected layer with batch normalization layers. The number of channels is set to [16, 32, 64, 32, 32, 512, 512, 10] for AlexNet, and [8, 8, 8, 8, 8, 16, 16, 16, 16, 32, 32, 32, 32, 64, 64, 64, 64, 10] for ResNet18.

C. Experimental Setting

All the base networks are trained by using SGD optimizer with a momentum of 0.9. Also, we used the weight decay with the strength of 10^{-4} to prevent overfitting. For MNIST and Fashion-MNIST, the networks are trained for 30 epochs with the sparse regularization using a mini-batch size 256. For CIFAR-10 and CIFAR-100, we trained the networks for 100 epochs with the sparse regularization using a mini-batch size 128. For STL-10, the network is trained for 100 epochs with the sparse regularization using a mini-batch size 64.

The hyper-parameter λ, which balances the cross-entropy loss and the sparse regularization criterion, is experimentally determined by grid search in the range from 10^{-1} to 10^{-6}. For SGL, SGL$_{1/2}$, SHSQRT-GL$_{1/2}$, SHSQ-GL$_{1/2}$ and OICSR-GL, we set the parameter α, which balances the L1 regularization and the group sparse regularization criterion, to be 0.5. Also, we set $m = 0.8$ for CGES.

D. Preliminary Experiments using MNIST and Fashion-MNIST datasets

TABLE II: Accuracy and sparsity with simple CNN on MNIST and Fashion-MNIST datasets. Top 2 sparsity are shown in boldface.

Dataset	MNIST		Fashion-MNIST	
Method	Accuracy	Sparsity	Accuracy	Sparsity
L2(Baseline)	99.03%	1.81%	87.90%	1.73%
L1	99.18%	39.29%	89.15%	95.39%
GL	99.20%	1.82%	89.50%	68.66%
ES	99.14%	52.98%	88.32%	98.76%
GL$_{1/2}$	99.16%	50.33%	89.10%	96.61%
SGL	99.22%	19.62%	88.38%	99.20%
SGL$_{1/2}$	99.24%	21.70%	88.26%	99.45%
CGES	99.05%	58.94%	89.69%	62.42%
OICSR-GL	99.24%	1.89%	89.50%	67.21%
HSQRT-GL	99.22%	2.20%	89.40%	87.95%
HSQ-GL	99.17%	20.55%	88.86%	86.05%
HSQRT-ES	99.09%	58.80%	88.78%	96.23%
HSQ-ES	99.10%	19.75%	88.05%	80.98%
HSQRT-GL$_{1/2}$	99.25%	26.83%	87.97%	**99.67%**
HSQ-GL$_{1/2}$	99.13%	**77.60%**	88.61%	**99.54%**
SHSQRT-GL$_{1/2}$	99.20%	21.09%	88.39%	99.43%
SHSQ-GL$_{1/2}$	99.04%	**94.08%**	88.82%	99.38%

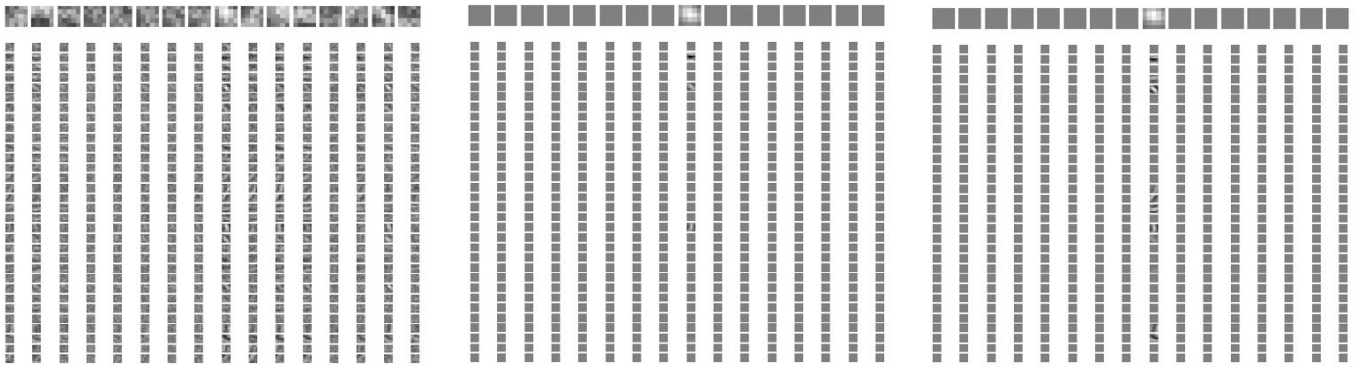
At first, we have performed preliminary experiments to investigate the effectiveness of the proposed hierarchical group sparse regularization for the simple CNN using MNIST and Fashion-MNIST datasets, which include gray-scale images.

Results of the simple CNN for MNIST and Fashion-MNIST with the hierarchical group sparse regularizations and the other sparse regularizations are shown in Tab. II. The ratio of the zero weights is calculated by assuming the weights whose absolute value is less than 10^{-3} are zero to evaluate the sparsity of the trained network.

From this table, it is noticed that all the test accuracies are higher than the baseline (L2) after the sparse regularizations are introduced. For the MNIST dataset, the sparse regularizations SHSQ-GL$_{1/2}$ and HSQ-GL$_{1/2}$ achieved the sparsity of 94.08% and 77.60%. For the Fashion-MNIST dataset, the sparse regularizations HSQRT-GL$_{1/2}$, HSQ-GL$_{1/2}$, SGL$_{1/2}$, SHSQRT-GL$_{1/2}$, and SHSQ-GL$_{1/2}$ achieved the sparsity more than 99%. These results show the effectiveness of the proposed hierarchical group sparse regularizations. Also, it is noticed that the group $L_{1/2}$ base regularizations are effective in increasing the sparseness.

From these results, we can say that the parameters of the CNN are very redundant, and more than 90% of the weights are not necessary to achieve the classification accuracy of the baseline CNN without pruning.

We visualized the convolutional filters of each network after the training for the MNIST dataset with the structured sparse regularizations. Fig. 3 shows the filters in the 1_{st} convolutional layer and the 2_{nd} convolutional layer for the baseline CNN and



(a) Filters of trained network without regularization (b) Filters of trained network with neuron-wise HSQRT-GL (c) Filters of trained network with feature-wise HSQRT-GL

Fig. 3: Visualization of the 1_{st} convolutional layer filer (showed at above) and 2_{nd} convolutional layer filter for a input neuron (showed at below) from the network trained on MNIST dataset. The numbers of channels of each layer are [16, 32, 128, 10]. All filters are shown in 1_{st} convolutional layer and 2_{nd} convolutional layer. (a) Filters of trained network without regularization. Sparsity of the filters at the 1_{st} convolutional layer and 2_{nd} convolutional layer are 00.25%, 1.86% respectively. (b) Filters of trained network with neuron-wise HSQRT-GL. Sparsity of the filters at the 1_{st} convolutional layer and 2_{nd} convolutional layer are 63.50%, 99.22% respectively. (c) Filters of trained network with feature-wise HSQRT-GL. Sparsity of the filters at the 1_{st} convolutional layer and 2_{nd} convolutional layer are 93.75%, 97.75% respectively.

the networks trained with the neuron-wise HSQRT-GL and the feature-wise HSQRT-GL.

As shown in Fig. 3a, the weights of the trained filters without sparse regularization becomes active at almost all locations and show various patterns. However, the filters trained with the hierarchical group sparse regularization are sparse in which many of the weights become almost zero, as shown in Fig. 3b and Fig. 3c. Notably, only one filter remains active in the 1_{st} convolutional layer, and this filter works as a blurring filter. Also, only a few filters are survived in the 2_{nd} convolutional layer. These filters work as directional edge detection filters to the blurred input image processed by the filter in the 1_{st} convolution layer. Interestingly, the effectiveness of the features obtained by the combinations of the directional edge filers and blurring is well known in character recognition, and there is a correspondence with the network trained with the structured sparse regularizations. We can get this structure automatically by training with sparse regularizations.

Similar results are also obtained for the MNIST dataset and the Fashion-MNIST dataset by using the other structured sparse regularizations. These results show that the network structure of a combination of the blurring filter and edge filters is fundamental for MNIST and Fashion-MNIST. It is interesting to consider the reason why this network structure is fundamental for gray image classification tasks.

Fig. 3b and Fig. 3c show the visualizations of the filters obtained by the neuron-wise grouping with HSQRT-GL and the filters by feature-wise grouping with HSQRT-GL. In the neuron-wise grouping, the weights connected to a output neuron are considered as a group, and the structured sparse regularization removes the neuron if the neuron is not necessary for the classification task. On the other hand, the weights to a input neuron are considered as a group, and the structured sparse regularization removes the neuron if the neuron is not necessary for the classification task.

From the experiments, in both grouping methods, we found that the network trained with structured sparse regularization can remove unnecessary neurons enforcing the subset of the weights to be zero. Thus, the trained filters with the neuron-wise grouping and the feature-wise grouping are similar. In the following experiments, we show the results for the case of the feature-wise grouping.

E. Hierarchical vs Non-Hierarchical

The training results of AlexNet, ResNet18, and VGG11bn with the sparse regularizations for the CIFAR-10, the CIFAR-100, and the STL-10 datasets are shown in Tab. III.

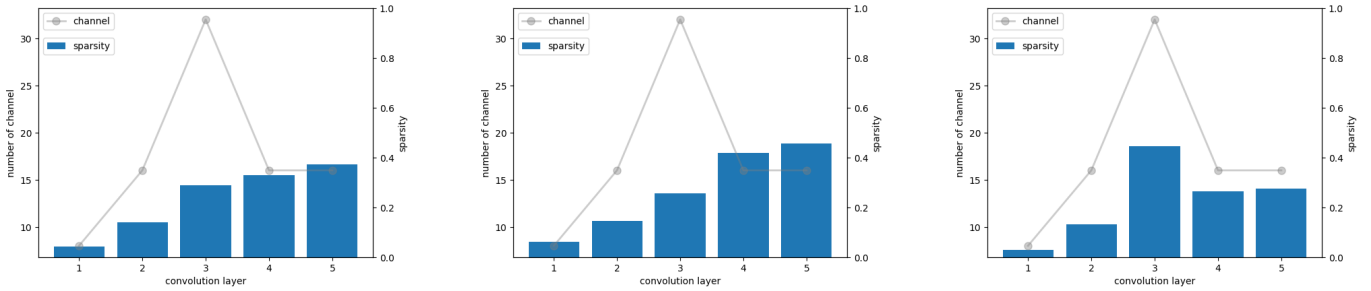
By comparing the proposed hierarchical group sparse regularizations (HSQRT-GL, HSQ-GL, HSQRT-ES, HSQ-ES, HSQRT-GL$_{1/2}$, and HSQ-GL$_{1/2}$) with the corresponding non-hierarchical group sparse regularization (GL, ES, and GL$_{1/2}$), it is noticed that the networks trained with the hierarchical group sparse regularization are sparser than the non-hierarchical group sparse regularizations. Especially, the hierarchical squared sparse regularizations can make the trained networks more sparse by keeping the accuracy the same as the baseline network.

For all datasets and all network structures, HSQ-GL$_{1/2}$ shows the highest sparsity keeping high accuracy. These results show that HSQ-GL$_{1/2}$ is the most effective in terms of sparseness.

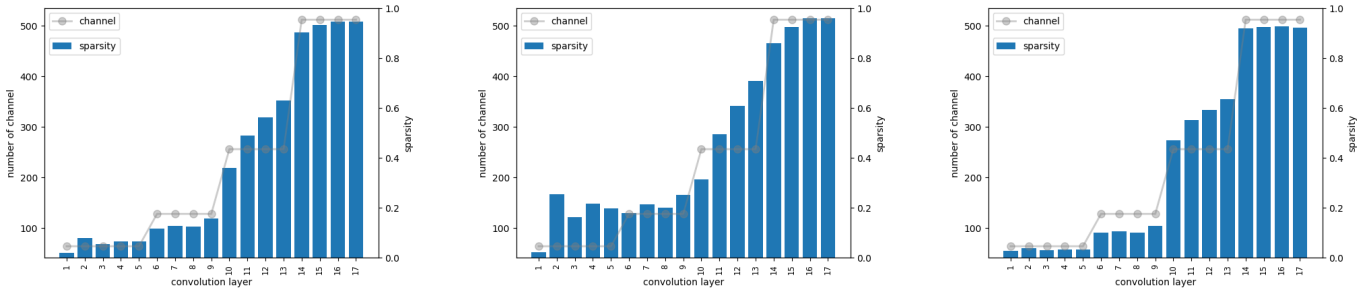
We also compare the ways of grouping with filter-wise grouping that the simplest grouping, feature-wise grouping that

TABLE III: Results for AlexNet, ResNet18 and VGG11bn nets on CIFAR-10/100 and STL-10 dataset. Ave rank shows the average of the ranks of the sparsity. The best sparsity is shown in boldface.

Network	AlexNet				ResNet18				VGG11bn		Ave rank
Dataset	CIFAR-10		STL-10		CIFAR-10		STL-10		CIFAR-100		
Method	Accuracy	Sparsity	Accuracy	Sparsity	Accuracy	Sparsity	Accuracy	Sparsity	Accuracy	Sparsity	
L2(Baseline)	75.50%	0.50%	66.90%	1.52%	72.43%	1.59%	70.69%	1.13%	57.20%	1.09%	-
L1	76.00%	5.53%	69.92%	12.34%	72.98%	72.82%	73.22%	73.75%	58.30%	39.32%	7.4
GL	76.03%	0.67%	69.53%	1.68%	73.44%	42.72%	74.66%	22.52%	58.59%	1.63%	14.4
ES	75.70%	10.05%	69.39%	14.05%	72.79%	78.10%	72.84%	64.65%	58.87%	13.68%	7.8
$GL_{1/2}$	75.76%	9.38%	69.33%	29.62%	73.29%	24.29%	71.62%	15.63%	57.44%	49.47%	8.8
SGL	75.91%	2.20%	67.36%	71.93%	73.48%	64.57%	72.12%	53.27%	58.56%	19.49%	9
$SGL_{1/2}$	75.55%	2.58%	67.04%	76.98%	73.16%	67.09%	71.56%	62.42%	58.49%	21.88%	7.4
CGES	76.04%	2.46%	68.56%	30.21%	72.48%	88.14%	72.26%	10.01%	59.10%	1.29%	10
OICSR-GL	76.23%	0.61%	69.54%	1.59%	73.69%	47.68%	75.02%	24.92%	59.03%	1.78%	14.2
HSQRT-GL	76.02%	0.88%	69.49%	6.03%	73.03%	71.02%	70.94%	1.37%	58.91%	3.49%	13
HSQ-GL	76.12%	3.16%	69.04%	23.11%	73.98%	67.96%	73.72%	40.52%	59.40%	7.35%	9.8
HSQRT-ES	75.80%	13.04%	70.10%	14.65%	73.02%	76.93%	71.40%	79.50%	58.46%	28.80%	5.6
HSQ-ES	75.60%	10.25%	68.05%	17.66%	73.54%	58.49%	72.64%	25.22%	57.31%	14.97%	9.4
$HSQRT\text{-}GL_{1/2}$	75.77%	5.63%	69.01%	28.02%	73.07%	83.54%	71.33%	9.66%	58.11%	43.61%	7.4
$HSQ\text{-}GL_{1/2}$	75.54%	**26.30%**	67.91%	63.20%	72.89%	**89.05%**	70.71%	**93.73%**	57.69%	**62.79%**	**1.6**
$SHSQRT\text{-}GL_{1/2}$	75.80%	2.51%	67.06%	**78.56%**	73.20%	68.11%	71.25%	67.39%	58.63%	21.79%	6.8
$SHSQ\text{-}GL_{1/2}$	75.67%	12.72%	69.10%	43.28%	72.80%	84.01%	72.67%	89.23%	57.72%	39.40%	3.4



(a) filter-wise $GL_{1/2}$ (b) feature-wise $GL_{1/2}$ (c) feature-wise $HSQ\text{-}GL_{1/2}$



(d) filter-wise $GL_{1/2}$ (e) feature-wise $GL_{1/2}$ (f) feature-wise $HSQ\text{-}GL_{1/2}$

Fig. 4: The sparsity of each of the layers in the trained network with sparse regularization. (a)-(c) Sparsity of the trained AlexNet on CIFAR-10 with the group-wise $GL_{1/2}$. The number of channels of the network are [8, 16, 32, 16, 16, 128, 10]. The sparsity of the trained AlexNet is around 30%. The accuracy and sparsity are 78.23%, 0.56%. (d)-(f) Sparsity the trained ResNet18 on STL-10 with group-wise $GL_{1/2}$. The number of channels of the network are [64, 64, 64, 64, 64, 128, 128, 128, 128, 256, 256, 256, 256, 512, 512, 512, 512, 10]. The sparsity is around 80%. The accuracy and sparsity are 75.97%, 3.93%. (a) AlexNet on CIFAR-10 with the filter-wise $GL_{1/2}$ regularization, accuracy and sparsity are 78.99%, 30.04%. (b) AlexNet on CIFAR-10 with the feature-wise $GL_{1/2}$ regularization, accuracy and sparsity are 78.68%, 33.67%. (c) AlexNet on CIFAR-10 with the feature-wise $HSQ\text{-}GL_{1/2}$ regularization, accuracy and sparsity are 79.25%, 31.46%. (d) ResNet on STL-10 with the filter-wise $GL_{1/2}$ regularization, accuracy and sparsity are 78.48%, 81.07%. (e) ResNet on STL-10 with the feature-wise $GL_{1/2}$ regularization, accuracy and sparsity are 78.36%, 82.20%. (f) ResNet18 on STL-10 with the feature-wise $HSQ\text{-}GL_{1/2}$ regularization, accuracy and sparsity are 78.51%, 80.85%.

can consider the weight for the input neuron as a group and hierarchical feature-wise grouping that can consider the weight for the input neuron as a group and convolutional filter as a group in the same group.

Fig. 4 shows the sparsity at each layer in AlexNet for CIFAR-10 and ResNet18 for STL-10 after training with the filter-wise $\mathrm{GL}_{1/2}$, the feature-wise $\mathrm{GL}_{1/2}$, and the feature-wise HSQ-$\mathrm{GL}_{1/2}$.

In AlexNet, the number of channels increases from the first layer to the third layer, and then it decreases at the forth layer. The network trained with the filter-wise $\mathrm{GL}_{1/2}$ or the feature-wise $\mathrm{GL}_{1/2}$ regularizations becomes sparse only in the later layers. On the other hand, the middle layers of the network trained with the feature-wise HSQ-$\mathrm{GL}_{1/2}$ are also sparse. These results suggest that we do not need to increase the number of channels in the later layers in AlexNet.

ResNet18 consists of four blocks of the layers. In each block, the number of channels is the same in each layer. The number of channels increases as the block becomes closer to the output layer. The sparseness of each layer in the network trained with the filter-wise $\mathrm{GL}_{1/2}$ regularization or the feature-wise $\mathrm{GL}_{1/2}$ regularization increases as the layer is closer to the output. Also, the sparseness of the layers in each block is different. On the other hand, the sparseness of the layers is almost the same within each block when the network was trained with HSQ-$\mathrm{GL}_{1/2}$ regularization. This result also shows that we do not need to increase the number of channels in ResNet18.

From these results, we think that the proposed hierarchical group sparse regularizations can effectively prune the unnecessary subsets of weights more adequately depending on the structure of the network and the number of channels.

F. Comparison with the previous works

From Tab. II and Tab. III, it is obvious that the sparseness of the network trained with the proposed hierarchical group sparse regularization criteria is higher than the previous sparse regularization criteria, including CGES and OICSR-GL. It is noticed that higher sparsity keeping better accuracy than the base networks is obtained by the hierarchical group sparse regularization criteria, which use L1 regularization in the hierarchical grouping. Especially HSQ-$\mathrm{GL}_{1/2}$ gives the highest sparsity for almost all network architectures and datasets.

REFERENCES

[1] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[2] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

[3] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[4] Mark Schmidt. Graphical model structure learning with l_1- regularization. *University of British Columbia*, 2010.

[5] Seyoung Kim and Eric P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, page 543–550. Omnipress, 2010.

[6] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010.

[7] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245, 2013.

[8] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in neural information processing systems*, pages 2074–2082, 2016.

[9] Jose M Alvarez and Mathieu Salzmann. Learning the number of neurons in deep networks. In *Advances in Neural Information Processing Systems*, pages 2270–2278, 2016.

[10] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

[11] Yang Zhou, Rong Jin, and Steven Chu-Hong Hoi. Exclusive lasso for multi-task feature selection. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 988–995, 2010.

[12] Deguang Kong, Ryohei Fujimaki, Ji Liu, Feiping Nie, and Chris Ding. Exclusive feature learning on arbitrary structures via $\ell_{1,2}$-norm. In *Advances in Neural Information Processing Systems*, pages 1655–1663, 2014.

[13] Jaehong Yoon and Sung Ju Hwang. Combined group and exclusive sparsity for deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3958–3966. JMLR. org, 2017.

[14] Zongben Xu, Hai Zhang, Yao Wang, XiangYu Chang, and Yong Liang. $l_{1/2}$ regularization. *Science China Information Sciences*, 53(6):1159–1169, 2010.

[15] Zongben Xu, Xiangyu Chang, Fengmin Xu, and Hai Zhang. $l_{1/2}$ regularization: A thresholding representation theory and a fast solver. *IEEE Transactions on neural networks and learning systems*, 23(7):1013–1027, 2012.

[16] Jinshan Zeng, Shaobo Lin, Yao Wang, and Zongben Xu. $l_{1/2}$ regularization: Convergence of iterative half thresholding algorithm. *IEEE Transactions on Signal Processing*, 62(9):2317–2329, 2014.

[17] Wei Wu, Qinwei Fan, Jacek M Zurada, Jian Wang, Dakun Yang, and Yan Liu. Batch gradient method with smoothing $l_{1/2}$ regularization for training of feedforward neural networks. *Neural Networks*, 50:72–78, 2014.

[18] Qinwei Fan, Jacek M Zurada, and Wei Wu. Convergence of online gradient method for feedforward neural networks with smoothing $l_{1/2}$ regularization penalty. *Neurocomputing*, 131:208–216, 2014.

[19] Feng Li, Jacek M Zurada, and Wei Wu. Smooth group $l_{1/2}$ regularization for input layer of feedforward neural networks. *Neurocomputing*, 314:109–119, 2018.

[20] Habtamu Zegeye Alemu, Junhong Zhao, Feng Li, and Wei Wu. Group $l_{1/2}$ regularization for pruning hidden layer nodes of feedforward neural networks. *IEEE Access*, 7:9540–9557, 2019.

[21] Jiashi Li, Qi Qi, Jingyu Wang, Ce Ge, Yujian Li, Zhangzhang Yue, and Haifeng Sun. Oicsr: Out-in-channel sparsity regularization for compact deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7046–7055, 2019.

[22] Rongrong Ma, Jianyu Miao, Lingfeng Niu, and Peng Zhang. Transformed ℓ_1 regularization for learning sparse deep neural networks. *Neural Networks*, 119:286–298, 2019.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[24] 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.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv: 1409.1556*, 2014.