Interspeech 2018
2-6 September 2018, Hyderabad

# Automatic DNN Node Pruning Using Mixture Distribution-based Group Regularization

*Tsukasa Yoshida*[1,2], *Takafumi Moriya*[1], *Kazuho Watanabe*[2],
*Yusuke Shinohara*[1], *Yoshikazu Yamaguchi*[1], *Yushi Aono*[1]

[1]NTT Media Intelligence Laboratories, NTT Corporation, Japan.
[2]Deptertment of Computer Science and Engineering, Toyohashi University of Technology, Japan.
tyoshida@lisl.cs.tut.ac.jp, takafumi.moriya.nd@hco.ntt.co.jp

## Abstract

In this paper, we address a constrained training for deep neural network-based acoustic model size reduction. While the L2 regularizer is used as a modeling approach to shrinking parameters, we cannot cut down the unimportant parts because it does not assume any group structure. The Group Lasso regularizer is used for the model size reduction approach. Group Lasso can set arbitrary group parameters (e.g. the column vector norms of the parameter matrices) as unimportant parts, and make the parameters sparse. Therefore, we can prune the unimportant parameters whose group parameter norm is nearly zero. However, Group Lasso does not suggest a clear rule for separating parameters close to zero and large in the group parameter space and hence is unsuitable for the model size reduction. To solve these problems, we propose a mixture distribution-based regularizer which assumes distributions of norms in the group parameter space. We evaluate our method on a NTT real recorded voice search data containing 1600 hours. Our proposal achieves 27.0% reduction compared to the pruned model by Group Lasso while keeping recognition performance.

**Index Terms**: automatic speech recognition, deep neural network, group regularization

## 1. Introduction

In recent years, the Deep Neural Network (DNN) has attracted attention as an artificial intelligence technology because of its excellent generalization performance; its application in various fields has been studied [1]. DNN is also being actively studied in the field of speech recognition, and it has been observed that it can achieve high recognition performance if allowed to form complicated models such as convolutional neural networks and recurrent neural networks with long short-term memory [2, 3, 4]. The high generalization performance of DNN comes from its deep layer structure and wealth of parameters.

However, the forward calculation process of DNN incurs enormous computation costs since the sum of the products of parameters and inputs is calculated in each layer. If the model is made more complicated, the response time becomes even longer. In addition, the parameter storage requirements can be excessive. The high computation costs are a problem when incorporating DNN into small terminals such as smart phones and wearable devices.

Although methods for reconstructing DNNs with fewer parameters have been proposed [5, 6, 7], they demand re-training after the ordinary fine-tuning process. Our approach is to focus on constrained training because the trained model can be combined with pruning or compression type operations such as low-rank matrix factorization using singular value decomposition (SVD) or decomposition methods, node pruning, a special network structure, and combinations of node-pruning and quantization [8, 9, 10, 11, 12, 13].

A node pruning method that dispenses with re-training was proposed by using Group Lasso [14]. The parameters of DNN are given in matrix form. They regard a column or a row vector of the matrix as a group, and add a regularization term to the loss function for minimizing each group norm. As a result, all parameters belonging to an unnecessary group approach zero simultaneously in ordinary parameter training. After training, node pruning is executed by deleting the nodes corresponding to the groups whose norms are close to zero. However, Group Lasso does not suggest a clear rule for separating parameters close to zero and large in the group parameter space and hence is unsuitable for model size reduction. Moreover, the effectiveness of Group Lasso against real problems remains doubtful because it has yet to be applied to large-scale datasets.

In this paper, we propose a novel group regularization method by modifying Group Lasso. In this regularization method, the group parameter norm is considered to be a random variable and its prior distribution is assumed. We demonstrate that further reduction in the number of parameters can be achieved by using a mixture of two Gaussian distributions. We use 1600 hours of various speech data including actually recorded voice segments in one of NTT's voice search databases, and show the superiority of the proposed method over Group Lasso.

## 2. Grouping DNN parameters

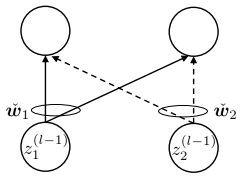


Figure 1: *The role of column vector $\vec{\boldsymbol{w}}$ of weight matrix $\boldsymbol{W}$. This paper adopts column weight vector norm of $\boldsymbol{W}$ as the group parameters.*

We consider a fully connected neural network with $L$ layers. Let the number of nodes in each layer be $N_l$. The output of the neural network can be obtained by calculating the following from $l = 1$ to $l = L - 1$ :

$$\boldsymbol{x}^{(l+1)} = \boldsymbol{W}^{(l)}\boldsymbol{z}^{(l)} + \boldsymbol{b}^{(l)},\ \boldsymbol{z}^{(l+1)} = \boldsymbol{\sigma}\left(\boldsymbol{x}^{(l+1)}\right), \quad (1)$$

where $\boldsymbol{z}^{(l)}$ is the output vector from the $l$-th layer, $\boldsymbol{z}^{(1)}$ is the


10.21437/Interspeech.2018-2062

input vector into the $(l+1)$-th layer of the DNN, $\boldsymbol{W}^{(l)}$ is the weight matrix between the $l$-th and $(l+1)$-th layers, $\boldsymbol{b}^{(l)}$ is the bias vector added to the $(l+1)$-th layer, and $\boldsymbol{\sigma}(\boldsymbol{x})$ is the vector-valued function that applies an activation function to each element of a vector : That is, for $\boldsymbol{x}^{(l)} = [x_1, x_2, \ldots, x_{N_l}]^{\mathrm{T}}$, $\boldsymbol{\sigma}(\boldsymbol{x}^{(l)}) = [\sigma(x_1), \sigma(x_2), \ldots, \sigma(x_{N_l})]^{\mathrm{T}}$, where $\sigma(x)$ indicates the use of sigmoid as the activation function.

$\check{\boldsymbol{w}}_i^{(l)}$ denotes the $i$-th column vector of $\boldsymbol{W}^{(l)}$ in Eq.(1). The role of $\check{\boldsymbol{w}}_i^{(l)}$ is shown in Fig. 1, using the example of a neural network with two nodes in the $(l-1)$- and $l$-th layers. Thus, if the norm of $\check{\boldsymbol{w}}_i^{(l)}$ is close to 0, we can consider the $i$-th node in the $(l-1)$-th layer unnecessary. Refer to Section 3.4 for detailed deletion procedures.

# 3. Regularization and node pruning

Regularization adds a penalty function, called the "regularization term", related to the parameters to a general error function. The loss function, $L$, has the following expression when regularization is applied :

$$L(\mathcal{W}) = E(\mathcal{W}) + R(\mathcal{W}), \tag{2}$$

where $\mathcal{W}$ is a set of weight matrices : $\mathcal{W} = \{\boldsymbol{W}^l\}_{l=1}^{L-1}$, $E(\mathcal{W})$ is the error function and $R(\mathcal{W})$ is the regularization term. DNNs that perform classification generally use the cross entropy function for $E(\mathcal{W})$ [15].

By considering the $\mathcal{W}$ that can minimize equation (2), learning can be done with constraints on parameters. Adding a penalty term is closely related to Bayesian estimation. The addition of $R(\mathcal{W})$ to the loss function corresponds to the introduction of prior probability.

### 3.1. L2 Regularization

The regularization term of L2 Regularization is defined as follows :

$$R_{\mathrm{L2}}(\mathcal{W}) = \frac{\lambda_{\mathrm{L2}}}{2} \sum_{l=1}^{L-1} \|\boldsymbol{W}^{(l)}\|_F^2, \tag{3}$$

where $\lambda_{\mathrm{L2}}$ is a regularization parameter and $\|\cdot\|_F$ is the Frobenius norm. Since L2 regularization takes as its penalty the sum of squares of elements of $\boldsymbol{W}$, learning proceeds by making all elements of $\boldsymbol{W}$ in each layer approach zero as much as possible. L2 regularization is also called Ridge regularization. L2 regularization corresponds to Bayesian estimation whose prior distribution is a Gaussian distribution.

### 3.2. Group Lasso

When Group Lasso is used for DNN node pruning, regularization term $R_{GL}$ is defined as follows :

$$R_{\mathrm{GL}}(\mathcal{W}) = \lambda_{\mathrm{GL}} \sum_{g \in \mathcal{G}_{\mathrm{GL}}} \|\boldsymbol{w}_g\|_2 + \frac{\kappa_{\mathrm{GL}}}{2} \|\boldsymbol{W}^{(1)}\|_F^2, \tag{4}$$

where $\lambda_{\mathrm{GL}}, \kappa_{\mathrm{GL}}$ are regularization parameters, $\|\cdot\|_2$ is the L2 norm, and $\mathcal{G}_{\mathrm{GL}}$ is a set of groups for Group Lasso. Element $g$ in $\mathcal{G}_{\mathrm{GL}}$ corresponds to the individual columns of weight matrix $\boldsymbol{W}_l$ for $l = 2, \ldots, L-1$. $\boldsymbol{w}_g$ is the vector of parameters belonging to group $g$. In this case $\boldsymbol{w}_g$ is the column vector $\check{\boldsymbol{w}}$ of weight matrix $\boldsymbol{W}$.

Group Lasso takes as its penalty the norm of each group parameter. When DNN learns the minimization of the error function and $R_{\mathrm{GL}}$ simultaneously, the norms of the unnecessary groups approach zero. That is, $\check{\boldsymbol{w}}$ approaches the 0 vector. The Lp norm can be used as the regularizer in the first term of equation (4), but the L2 norm is most popular.

In equation (4), L2 regularization is applied to give weak constraints to just the first layer. This is to prevent the loss of the important information of the input layer.

### 3.3. Group Regularization (Proposed)

The group regularization that we propose is a variant of Group Lasso. Its regularization term is defined as follows :

$$R_{\mathrm{P}}(\mathcal{W}) = -\lambda_{\mathrm{P}} \sum_{g \in \mathcal{G}_{\mathrm{P}}} \log\Big(\tilde{p}(\|\boldsymbol{w}_g\|_2)\Big), \tag{5}$$

where $\lambda_{\mathrm{P}}$ is the regularization parameter and $\tilde{p}(x)$ is an arbitrary function taking a positive value.

In particular, in this study, we propose $\tilde{p}(x)$ for node pruning as follows :

$$\tilde{p}(x) = A_1 \exp\left(-\frac{\Lambda_1}{2}x^2\right) + A_2 \exp\left(-\frac{\Lambda_2}{2}(x-\mu)^2\right), \tag{6}$$

where $\mu, A_1, A_2, \Lambda_1, \Lambda_2 > 0$ are parameters. This is based on an idea similar to the spike and slab model for variable selection [16]. This definition corresponds to taking the truncated mixture distribution of the two Gaussian distributions whose means are 0 and $\mu$ as the prior distribution in Baysian estimation. $\Lambda_1, \Lambda_2$ are the reciprocals of the variances of the Gaussian distributions. $A_1, A_2$ correspond to the mixture ratio, but they do not need to satisfy $A_1 + A_2 = 1$. For node pruning, $\mathcal{G}_{\mathrm{P}}$ is defined as a set of all columns of weight matrix $\boldsymbol{W}_l$ for $l = 1, \ldots, L-1$.

This paper uses "bimodal Group Ridge (bGR)" to refer to the group regularization based on equation (6). These definitions mean that the penalty will increase for groups whose group parameter norms deviate from 0 or $\mu$. When DNN learns the minimization of the error function and $R_{\mathrm{P}}$ simultaneously, the norms of unnecessary groups approach zero, and the norms of necessary groups approach $\mu$. That is, $\check{\boldsymbol{w}}$ are automatically divided into two sets.

### 3.4. Node pruning procedure

We test the use of column vectors as groups. The algorithm for node pruning is as follows :

Repeat the following steps from $l = 1$ to $l = L-1$.

I Find the set $\mathcal{S}^{(l)}$ of the column number such that the norm of $\check{\boldsymbol{w}}$ in $\boldsymbol{W}^{(l)}$ is less than threshold value $\theta$.

II For all $i \in \mathcal{S}^{(l)}$, delete the $i$th column of $\boldsymbol{W}^{(l)}$, the $i$th row of $\boldsymbol{W}^{(l-1)}$ and the $i$th element of $\boldsymbol{b}^{(l-1)}$.

# 4. Experiment

### 4.1. Data

The training data consisted of 1600 hours of Japanese utterances recorded in various acoustic environments, and consisted of real data of voice search application, call center recordings, Corpus of Spontaneous Japanese (academic presentations) [17], and Japanese Newspaper Article Sentences (reading newspapers) [18]. The test data sets were composed to cover six tasks, distant talk (1.5 and 2.5m), voice search task (Child, Elder and Adult) and Reading speech, and each amounted to 2.5 hours in total.

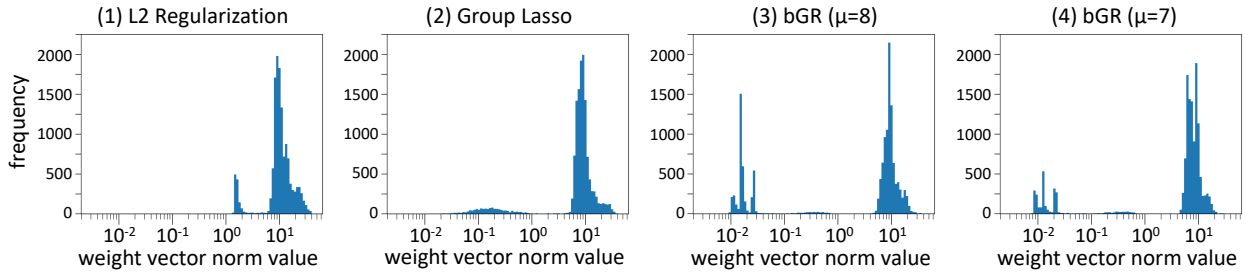


Figure 2: *Histogram of each column vector norm value in all weight matrices as the model parameters. The distributions towards the left side are unneeded so the corresponding nodes can be pruned. The proposed bGRs method can safely prune DNN nodes than the other approaches.*

Table 1: *CERs [%] of Japanese real recorded speech tasks (1600 hour-training data). We evaluate our proposal "bGR" on six field data sets: distant talk (1.5 and 2.5m), voice search task (Child, Elder and Adult) and Reading speech. Each column of θ indicates the threshold for DNN node pruning decided by the distribution in group parameter space in Fig. 2. The nodes with values less than θ are pruned.*

| **Evaluation data** | **Regularization** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **L2** | | | **Group Lasso** | | | **bGR1** ($\mu = 8$) | | | **bGR2** ($\mu = 7$) | | |
| | Not pruned | threshold $\theta$ 0.1 | 2 | Not pruned | threshold $\theta$ 0.1 | 2 | Not pruned | threshold $\theta$ 0.1 | 2 | Not pruned | threshold $\theta$ 0.1 | 2 |
| distant talk (1.5m) | 12.1 | 12.1 | 13.3 | 12.6 | 12.7 | 14.6 | 12.7 | 12.8 | 12.7 | 12.1 | 12.0 | **11.9** |
| distant talk (2.5m) | 16.2 | 16.2 | 18.0 | 16.4 | 16.2 | 18.8 | 16.8 | 16.9 | 16.7 | 15.8 | 15.8 | **15.7** |
| voice search (Child) | **10.9** | **10.9** | 11.0 | 11.1 | 11.1 | 11.0 | 11.5 | 11.5 | 11.5 | **10.9** | **10.9** | **10.9** |
| voice search (Elder) | 9.3 | 9.3 | 9.5 | 9.3 | 9.3 | **9.1** | 9.6 | 9.5 | 9.5 | 9.2 | 9.2 | 9.2 |
| voice search (Adult) | 13.6 | 13.6 | **13.2** | 13.6 | 13.5 | 13.7 | 14.0 | 14.0 | 14.1 | 13.3 | 13.3 | 13.4 |
| Reading speech | 3.2 | 3.2 | 4.4 | 3.1 | 3.1 | 2.9 | 3.4 | 3.3 | 3.0 | **2.6** | **2.6** | 2.7 |

### 4.2. System configuration

The input feature for all DNNs was 40 dimensional FBANK with the temporal context of 11 frames; dynamic features ($\Delta$ and $\Delta\Delta$) were used. The DNN architectures were six fully connected layers with 2048 nodes and the number of the output layer units was 3072; parameters were randomly initialized and trained without pre-training. The 3-gram language model was used in all conditions; it has 520K size vocabulary and was trained on various text corpora with a total of 2.3G words. Decoding was performed by the WFST-based decoder VoiceRex [19, 20]. We evaluated performance in terms of character error rate (CER).

### 4.3. Parameter settings for each regularizer

We set the coefficient hyperparameter of each regularizer as follows; $\lambda_{\mathrm{L2}} = \lambda_{\mathrm{GL}} = \lambda_{\mathrm{P}} = 2 \cdot 10^{-7}$ and $\kappa_{\mathrm{GL}} = 2 \cdot 10^{-8}$ (see in Section 3). These parameters were set by validation so that the CER of Group Lasso equaled the CER without regularizations as much as possible. We evaluated the bGR method using two values of $\mu$ ; "bGR1" is $\mu = 8$, while "bGR2" is $\mu = 7$. The values of $\mu$ were set by validation. These bGR use $(\Lambda_1, \Lambda_2) = (30, 1)$ as the variances of distributions, $(A_1, A_2) = (1, 1)$ as the weight of distributions. The value of $\Lambda_1$ is set at an appropriate integer value such that the probability that the group norm belonging to the distribution with mean 0 becomes less than 0.5 is more than 99%.

Threshold $\theta$ for pruning the model parameters was set at either $\theta = 0.1 or 2$. We can arbitrarily select $\theta$ to suit the group parameter distribution in Fig. 2 as described later.

### 4.4. Result

Fig. 2 shows, for each approach, a histogram of the weight vector norms in the converged parameters. The horizontal axis plots the norm value of the group (column vector) of the weight matrix, and the vertical axis plots the appearance frequency. Note that Fig. 2 is semilogarithmic, so the histogram of the proposed method does not show the shape expected of a normal distribution. The L2 regularization histogram, which does not have a group whose norm value is close to zero, indicates that groups considered unnecessary are densely distributed between 1 and 2. Group Lasso, bGR1 ($\mu = 8$) and bGR2 ($\mu = 7$) have groups whose norm value is close to zero. Group Lasso has a distribution of groups that are considered unnecessary widely spread from $10^{-2}$ to 2. On the other hand, bGRs densely gathered the groups considered to be unnecessary in the area with norm values less than $10^{-1}$. However, there is also a small norm group near 1 from $10^{-1}$. From these results, we set the thresholds for pruning to $\theta = 0.1, 2$.

Table 1 shows the CERs of each task before pruning and after pruning, and Table 2 shows the pruning reduction rates. Comparing the column of "Not pruned" for each method in Table 1, bGR2 ($\mu = 7$) yields better results than the other methods. The CERs before pruning take the order of bGR2 $>$ L2 $>$ Group Lasso $>$ bGR1. In addition, comparing the column of after pruning with $\theta = 0.1$ with the column of "Not pruned" for each method in Table 1, we see that there is little difference among the methods. It is considered that threshold $\theta$ was sufficiently small and did not significantly affect the CERs after pruning. Looking at the reduction rate with $\theta = 0.1$ (Table 2), L2 regularization has no reduction nodes at all. While Group Lasso achieved only a 7.1% reduction rate, bGR1 ($\mu = 8$) and

Table 2: *The total rate of pruned DNN nodes and that of reduced parameters. Each column of $\theta$ indicates the threshold for DNN node pruning decided by the boundary of distribution in group parameter space in Fig. 2. Our proposed bGRs could reduce the model parameters while keeping CERs (see in Table 1)*

| **Removal rate** | **Regularization** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **L2** | | **Group Lasso** | | **bGR1** ($\mu = 8$) | | **bGR2** ($\mu = 7$) | |
| | $\theta = 0.1$ | $\theta = 2$ | $\theta = 0.1$ | $\theta = 2$ | $\theta = 0.1$ | $\theta = 2$ | $\theta = 0.1$ | $\theta = 2$ |
| Group (DNN nodes) reduction rate | 0% | 8.3% | 3.0% | 10.8% | 28.0% | **30.3**% | 15.6% | 18.2% |
| Parameter reduction rate | 0% | 19.4% | 7.1% | 25.1% | 49.4% | **52.1**% | 25.5% | 28.9% |

Table 3: *The number of the pruned group vector norms corresponding to DNN nodes in each layer. Each column of $\theta$ indicates the threshold for DNN node pruning decided by the boundary of distribution in group parameter space in Fig. 2. The bGRs could prune more nodes than the other approaches, especially in middle hidden layers. The total node reduction ratio corresponds to group reduction rate in Table 2.*

| **Layer** $l$ | **Regularization** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **L2** | | **Group Lasso** | | **bGR1** ($\mu = 8$) | | **bGR2** ($\mu = 7$) | |
| | $\theta = 0.1$ | $\theta = 2$ | $\theta = 0.1$ | $\theta = 2$ | $\theta = 0.1$ | $\theta = 2$ | $\theta = 0.1$ | $\theta = 2$ |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 3 | 6 | 15 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 728 | 811 | 723 | 825 |
| 4 | 0 | 0 | 0 | 0 | 753 | 898 | 809 | 1,046 |
| 5 | 0 | 0 | 0 | 0 | 796 | 876 | 594 | 605 |
| 6 | 0 | 1,132 | 409 | 1,457 | 1,535 | 1,538 | 0 | 0 |

bGR2 ($\mu = 7$) achieved 28.0% and 15.6%, respectively. Comparing the column of after pruning with $\theta = 2$ with the column of "Not pruned" for each method in Table 1, we see that the CERs of L2 regularization and Group Lasso have deteriorated, whereas the CERs of bGR1 ($\mu = 8$), bGR2 ($\mu = 7$) are almost unchanged. The CERs after pruning with $\theta = 2$ is take the order of bGR2 $>$ bGR1 $>$ L2, Group Lasso. Looking at the reduction rate with $\theta = 2$ in Table 2, L2 regularization and Group Lasso have greatly increased reduction rate compared with $\theta = 0.1$. On the other hand, while the reduction rates of bGR1 ($\mu = 8$) and bGR2 ($\mu = 7$) have increased, the change is not large. However, the reduction rates of bGRs are very high compared with the results of L2 regularization and Group Lasso. The difference in parameter reduction rate between Group Lasso and bGR1 ($\mu = 8$) is 27.0%, the difference between Group Lasso and bGR2 ($\mu = 7$) is 3.8%.

Table 1 also shows the results wherein the CER increases after pruning in all methods. This is because the retention of an originally useless node yields an adverse effect like noise, so performance is expected to be improved by deleting useless nodes.

Table 3 shows the number of groups deleted in each layer. Looking at Table 3, while L2 regularization and Group Lasso achieve great reduction in only the last layer, bGR1 ($\mu = 8$) and bGR2 ($\mu = 7$) can achieve reductions in 4 layers and 3 layers respectively. Group Lasso is a regularization method similar to L2 regularization and so simply makes the parameters smaller; it does not change the structure of the parameter distribution of the original DNN. On the other hand, it can be seen that the proposed method completely changes the structure of the distribution, it can identify unnecessary parameters that remain hidden from simple reduction. In L2 regularization and Group Lasso, groups whose norm values are less than 2 are concentrated in the last layer after learning; deleting them all at once had a significant impact on the CER. On the other hand, bGRs was not significantly influenced by their deletion because the groups whose norm values is less than 2 were dispersed among the other layers. There are also extreme differences between bGR1 ($\mu = 8$) and bGR2 ($\mu = 7$) such as deleting the last layer or not. This implies that bGR1 ($\mu = 8$) considered the groups in the last layer unnecessary, which normally have a small norm value without regularization, whereas bGR2 ($\mu = 7$) considered them necessary.

The proposed method can achieve a large reduction rate like bGR1 ($\mu = 8$), and it was confirmed that the CER and reduction amount can be adjusted by altering $\mu$. In addition, while we performed experiments with$(A_1, A_2)$ fixed this time, it is considered that the reduction rate can be freely changed to some extent by adjusting these values which corresponds to the mixing proportion of prior probabilities. Furthermore, it can be thought that unnecessary parameters can be brought closer to 0 by using the Laplace distribution with mean 0 instead of the Gaussian distribution.

## 5. Conclusion

We have proposed a constrained DNN training method that uses a group regularizer for pruning model parameters. The proposal, bimodal Group Ridge (bGR), is based on a mixture distribution-based regularizer which assumes distributions of norms in the group parameter space; it has hyperparameters such as the mixture weight, mean and variance. bGR regularizes the group parameters by the assumed distribution so that classification of needed or unneeded group members becomes possible. We showed the effectiveness of the proposal in various Japanese speech recognition tasks; it achieved a higher rate of model parameter reduction than existing regularizers while keeping recognition performance high, the reduction rate was 52%. Future work includes tuning the hyperparameters and application of the proposal to convolutional neural networks and recurrent neural networks with long short-term memory.

# 6. References


[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "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.

[2] O. Abdel Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1533–1545, 2014.

[3] L. C. Jain and L. R. Medsker, *Recurrent Neural Networks: Design and Applications*, 1st ed. CRC Press, Inc., 1999.

[4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size dnn with output-distribution-based criteria," in *Proceedings Interspeech*, 2014, pp. 1910–1914.

[6] T. He, Y. Fan, Y. Qian, T. Tan, and K. Yu, "Reshaping deep neural network for fast decoding by node-pruning," in *Procceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014, pp. 245–249.

[7] R. Takeda, K. Nakadai, and K. Komatani, "Acoustic model training based on node-wise weight boundary model for fast and small-footprint deep neural networks," *Comput. Speech Lang.*, pp. 461–480, 2017.

[8] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition." in *Proc. of INTERSPEECH*, 2013, pp. 2365–2369.

[9] T. He, Y. Fan, Y. Qian, T. Tan, and K. Yu, "Reshaping deep neural network for fast decoding by node-pruning," in *Proc. ofICASSP*, 2014, pp. 245–249.

[10] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets." in *Proc. of ICASSP*, 2013, pp. 6655–6659.

[11] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Proc. of NIPS*, 2015, pp. 1135–1143.

[12] A. See, M.-T. Luong, and C. D. Manning, "Compression of neural machine translation models via pruning," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 291–301.

[13] V. Sindhwani, T. N. Sainath, and S. Kumar, "Structured transforms for small-footprint deep learning," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'15, 2015, pp. 3088–3096.

[14] T. Ochiai, S. Matsuda, H. Watanabe, and K. Shigeru, "Automatic node selection for deep neural networks using group lasso regularization," in *Procceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017, pp. 5485–5488.

[15] C. M. Biship, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[16] T. J. Mitchell and J. J. Beauchamp, "Bayesian variable selection in linear regression," *Journal of the American Statistical Association*, vol. 83, no. 404, pp. 1023–1032, 1988.

[17] S. Furui, K. Maekawa, and H. Isahara, "A Japanese national project on spontaneous speech corpus and processing technology," in *Proc. ASR'00*, 2000, pp. 244–248.

[18] K. Itou, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, and S. Itahashi, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," *Journal of the Acoustical Society of Japan (E)*, pp. 199–206, 1999.

[19] H. Masataki, D. Shibata, Y. Nakazawa, S. Kobashikawa, A. Ogawa, and K. Ohtsuki, "VoiceRex – spontaneous speech recognition technology for contact-center conversations," *NTT Technical Review*, vol. 5, no. 1, pp. 22–27, 2007.

[20] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," *IEEE Trans. Audio, Speech & Language Processing*, vol. 15, no. 4, pp. 1352–1365, 2007.