SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Volume 6 No. 1
ISSN 2054-7390

Structural Optimization of Deep Belief Network by Evolutionary Computation Methods including Tabu Search

[1]Tomohiro Hayashida, [1]Ichiro Nishizaki, [1]Shinya Sekizaki, [1]Masanori Nishida, [1]Murman Dwi Prasetio
[1]Graduate School of Engineering, Hiroshima University, Hiroshima, JAPAN;
hayashida@hiroshima-u.ac.jp

ABSTRACT

This paper proposes structural optimization method of a Deep Belief Network (DBN) which consists of multiple Restricted Boltzmann Machines (RBMs) and a single Feedforward Neural Network (FNN) using several kinds of evolutionary computation methods and modularization. The performance, accuracy of data classification or data prediction, should strongly depend on the structure of the network. Concretely, the number of RMBs, the number of nodes in the hidden layer of RMB. The result of the experiments using some benchmarks for image data classification problems by DBN optimized by the proposed method, DBN without any structural optimization, and some other data classification methods indicate that our proposed method defeats other existing classification methods.

Keywords: Structural optimization; Deep Belief Network; Tabu search; Modularization; Evolutionary Computation.

1 Introduction

A neural network consists of a number of units which have simple nonlinear transfer functions and approximation capability for a number of kinds of complex problems comparative small number of calculation. Therefore, neural networks are applied to data analysis, data mining and data classification. A sufficient learning cannot be performed if the size of the network is too small. Adversely, overfitting occurs to the learning data and it loses generalization ability if the size is too large. Therefore, the appropriate structure of the neural network is required to be determined for each target problem for higher performance of a neural network. Traditionally, structure of a neural networks are determined through a trial and error procedure based on the experiences of a designer of the neural networks. However, a huge computation time is required by such determination process.

In recent years, several structural optimization methods of the neural networks simultaneously with learning are proposed. Delgado et al. [4] proposed a simultaneous optimization method of learning and structure using an evolutionary multi-objective optimization methods, SPEA 2 and NSGA II with the learning error and the number of units of the hidden layers of neural networks as the objective functions. The target is a recurrent neural network (RNN), and the structural information of RNNs are encoded in gene format, such as the number of intermediate layer and the presence or absence of connection between the units of neighboring layers. Katagiri et al. [15] improve the procedure of Delgado et al. for extended Multi-Context Recurrent Neural Network (exMCRNN) which include eliminate unnecessary connection between nodes and the elite preserving strategy. Hayashida et al. [8] propose a structural

DOI: 10.14738/tmlai.61.4048
Publication Date: 30th December, 2017
URL: http://dx.doi.org/10.14738/tmlai.61.4048

optimization method for Recurrent Neural Network (RNN) by introducing two stage taboo search, one of the meta-heuristic solutions. Here, they define that the structure of a RNN is determined by the number of inputs, the number of intermediate layer units, the number of feedback layers. Hayashida et al. [7] proposed a structural optimization for a combined neural network model of a Feedforward Neural Network (FNN) and an Auto Encoder (AE) which performs dimension compression to remove extra data and redundant data. Their procedure optimize the number of input data, the number of units of the middle layer of AE, and the number of units of the hidden layer of FNN by using tabu search.

Because of the performance of a neural network should be measured based not only on the learning accuracy but also the generalization capability, a neural network is evaluated based on the degree of error between both the training data and data for verification of the generalization capability in above mentioned optimization method. In order to evaluate a neural network, it is necessary to divide the known data into training data and for verification of the generalization capability. However, even in a method such as cross validation, the data may be biased and the network cannot be evaluated well. Nishida et al. [20] has improved the method of Hayashida et al. [7], they use the Self Organization Map (SOM) to convert the data mapped onto a 2 D plane by k-means method and divide them into training data and data for verification of the generalization capability. Though such data generation method, they succeed in reducing the bias of features between divided data, and improving learning accuracy and improving generalization capability.

Deep Neural Network (DNN) consists of a lot of multiple layers, and the data analysis performance such as data prediction, data classification, or data mining is dramatically improved compared with conventional neural networks such as Feedforward Neural Network (FNN). Therefore, a lot of applications of DNN are reported in the various study fields. A neural network composed of many layers is difficult to learn properly by back propagation, however, the learning procedure of DNN is constructed for appropriate learning by applying apply pretraining [9], drop out [27] and so forth. Various models of DNN such as Convolutional Neural Network (CNN) [13], Deep Belief Network (DBN) [25], are proposed. This paper focuses on DBN which has a structure with multiple layers of Restricted Boltzmann Machine (RBM). DBN has succeeded in acquiring higher data analysis capability by effectively incorporating a feature extraction process which is conventionally performed by trial and error. In DBN, multiple RBMs were incorporated into the learning process as feature extractors. DBN performs feature extraction with unsupervised learning called Pre-training and supervised learning called Fine-tuning are performed based on the extracted features.

From the structural characteristics of DBNs, it can be considered that there exists a great relationship between the structure of DBM, the number of hidden layers and units constituting each layer, and the performance in data classification or prediction. Performance improvement is expected by giving an appropriate structure corresponding to input data. This paper proposes a new method for highly accuracy and efficient structure optimization for DBNs. Additionally, this paper compares the proposed method and the conventional methods by the numerical experiments, and verifies the effectiveness of the proposed method.

The rest of this paper is constructed as follows: Section 2 introduces the explanation of the neural networks and the conventional methods. Section 3 outlines the proposed method and Section 4 describes the design and result of the numerical experiments. Finally, Chapter 5 summarizes this paper.

2 Neural Networks and Structural Optimization

2.1 Neural Networks

This section introduces several models of neural networks.

FNN, Recurrent Neural Network (RNN) are neural networks with a transfer function as a sigmoid function: $f(s) = 1/\{1 + \exp(-s)\}$. An FNN has a layered structured units, and it consists of an input layer, an output layer, and a layer disposed between them called a hidden layer. Signals from the input layer to the output layer are transmitted in only one direction. In general, the number of hidden layers is not necessarily 1. A neural network with a signal feedback structure added to FNN is called a RNN. There are types of RNN such as Jordan Network [14] and Elman Network [5] depending on its structure. RNN is applied to prediction of time series data analysis.

Generally, NN with a number of layers is called Deep Neural Network (DNN). Appropriate learning of DNN based on back propagation (BP) is difficult. Several learning methods corresponding to DNN such as pretraining [10], drop out [27] have been proposed. There exist various models of DNN such as Deep Belief Network (DBN) [9], Convolutional Neural Network (CNN) [13], and so forth.

2.1.1 Deep Belief Network (DBN)

DBN [9] is a type of DNN that performs data classification or data prediction with high-precision by performing feature extraction by using a network in which a plurality of Restricted Boltzmann Machines (RBM) are concatenated. In the feature extractor part, unsupervised learning called Pre-training is performed, and Fine-tuning which is supervised learning is performed in the remaining process of it [9, 16]. Figure 1 shows DBN consisting of three layers of RBM and one layer of FNN as an example of DBN.

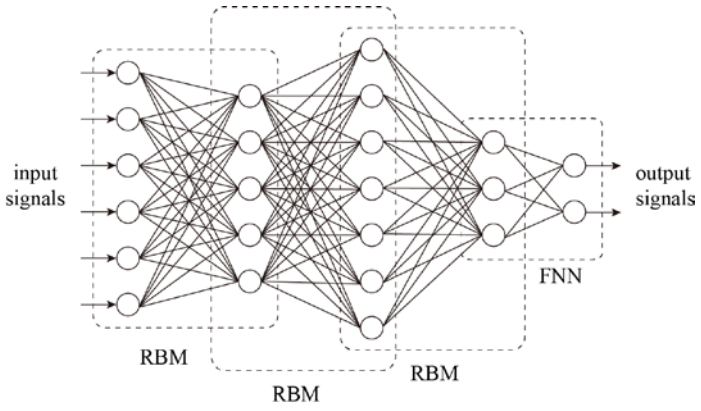

Figure 1. Deep Belief Network (DBN).

2.1.2 Restricted Boltzmann Machine (RBM)

A RBM is a Boltzmann machine with an undirected bipartite graph consisting of a visible layer and a hidden layer, and there is no connection between the units in a same layer. The connecting weights and the thresholds are updated so that the hidden layer extracts the feature amount of the input data of the

visible layer. A RBM can compress dimensionality of input data, feature learning, or collaborative filtering. Figure 2 shows an example of RBM.

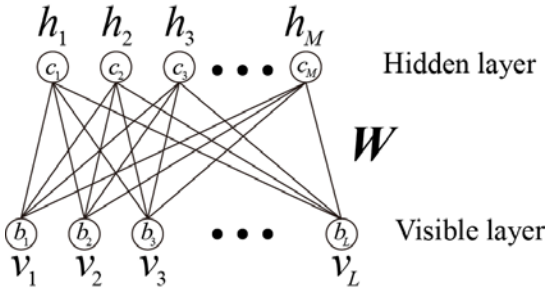

Figure 2. Restricted Boltzmann Machine (RBM).

Let L, M be numbers of units of a visible layer and a hidden layer shown in Figure 2, respectively. Let $\boldsymbol{v} = (v_1, v_2, \dots, v_L), v_i \in \{0,1\}, i = 1,2, \dots, L$ be a set of units of a visible layer, and $\boldsymbol{h} = (h_1, h_2, \dots, h_M), h_j \in \{0,1\}, j = 1,2, \dots, M$ be a set of units of a hidden layer, $b_i, i = 1,2, \dots, L$ be a bias of a unit i in a visible layer, $c_j, j = 1,2, \dots, M$ be a bias of a unit j in a hidden layer, and $\boldsymbol{W} = \{W_{ij}, i = 1,2, \dots, L, j = 1,2, \dots, M\}$ be the connecting weights between visible layer and hidden layer.

The conditional probability of hidden elements conditioned with visible elements is

$$p(h_j = 1|\boldsymbol{v}) = sigmoid(c_j + \textstyle\sum_i v_i W_{ij}). \quad (1)$$

And the conditional probability of visible elements conditioned with hidden elements is

$$p(v_i = 1|\boldsymbol{h}) = sigmoid(b_i + \textstyle\sum_j W_{ij} h_j), \quad (2)$$

where, $sigmoid(s) = 1/(1 + \exp(-s))$. The weight parameters W_{ij} and the biases $b_i, c_j, i = 1,2, \dots, L$, $j = 1,2, \dots, M$ are updated based on the maximum likelihood estimation method.

2.2 Structural Optimization of the neural networks

A structure of a neural network is characterized by number of hidden layers and number of units of each layer, and the performance is greatly affected by these characteristic parameters. That is, in order to obtain the appropriate property for the data classification of or data prediction, it is necessary to find the optimum characteristic parameters.

Delgado et al. [4] focus on a RNN called Elman Network with a feedback layer which only connects to the hidden layer, and they propose a structural optimization method based on genetic algorithm (GA), and they demonstrate the usefulness of structure optimization by numerical experiments. Katagiri et al. [15] focus on a Multi Context RNN (MCRNN) [11, 12] which consists of multiple feedback layers. And they propose an improved structural optimization method of Delgado et al. [4] for extended MCRNN by adding a structure of Time Delay NN (TDNN) to MCRNN.

Hayashida et al. [7] indicate the performance for data analysis of a neural networks with similar structures to be similar, and they propose a structural optimization method based on Tabu Search (TS) [6] for a combined neural network of a Sandgrass Type neural network and FNN named ST-FNN. Numerical experiments show that TS is more effective for a structural optimization method of NN than GA. Additionally, Hayashida et al. [8] proposed a structural optimization method of MCRNN by applying TS. In

order to deal with the problem that the number of structural parameters of MCRNN is many and the search space for structure evaluation becomes enormous. Therefore, by dividing the search space into a plurality of small regions, the searching process is performed by introducing a short term memory and a long term memory for the purpose of improving efficiency of comprehensive search of each small region and extensive search of the entire search space.

3 Structural Optimization of Deep Belief Networks (DBN)

This paper proposes a structure optimization method with parameters of each hidden layer and unit numbers of each layer of RBMs constituting DBN. The proposed method includes local search based on tabu search for structural optimization, modularization for improving of RBMs the learning efficiency which is required for structural evaluation of DBN and enormous calculation time. Furthermore, number of hidden layers and the number of units are optimized separately to reduce the search space.

3.1 Outline of the Structural Optimization

The outline of the DBN structure optimization method proposed in this paper is shown below.

Step 1. Optimize number of hidden layers**.**

Step 1-1. Let $n \in [\underline{n}, \bar{n}]$ be the number of hidden layers, and let $n = \underline{n}$ as the initial value. Let $n^* = n$ and $E_n^* = 0$.

Step 1-2. A DBN with n hidden layers is evaluated based on the training and generalization capability. Here, the number of units of each layer is 500. (The structural evaluation procedure is explained in the following subsection.)

Step 1-3. If $E_n > E_n^*$, then update the best solution as $E_n^* = E_n, n^* = n$.

Step 1-4. If $\bar{n} > n$, then let $n = n + 1$ and return Step 1-2. Otherwise let n^* be number of hidden layers.

Step 2. Optimize number of units of each layer (Rough search)

Step 2-1. Let m_i be number of units of i-th layer, $i = 1,2, \dots, n^*$, and let $t = 0$.

Step 2-2. Divide search range of number of units of i-th hidden layer, $[\underline{x}_i, \bar{x}_i]$ into k_i subranges. Let $d_i^{j_i} \equiv [\underline{x}_i^{j_i}, \bar{x}_i^{j_i}]$ be j_i-th subrange, $j = 1,2, \dots, k_i$, $\underline{x}_i^1 = \underline{x}_i, \bar{x}_i^{k_i} = \bar{x}_i$. n^*-dimensional subrange is represented as $D_j \equiv \Pi_{i=1}^{n^*} d_i^{j_i}$. Here, $j \in [1, \Pi_{i=1}^{n^*} k_i]$.

Step 2-3. Let $(\dot{x}_1^j, \dot{x}_2^j, \dots, \dot{x}_{n^*}^j)$ be the center of gravity of the subrange j, and let $\boldsymbol{x^j} \equiv (\hat{x}_1^j, \hat{x}_2^j, \dots, \hat{x}_{n^*}^j) = (\lfloor \dot{x}_1^j + 0.5 \rfloor, \lfloor \dot{x}_2^j + 0.5 \rfloor, \dots, \lfloor \dot{x}_{n^*}^j + 0.5 \rfloor)$ be the representative point of the subrange j.

Step 2-4. Evaluate the structure of the neural network corresponding to the representative point $\boldsymbol{x^j}$. Let E^j be the evaluate value of the subspace j.

Step 2-5. Choose a subspace with highest evaluation value, and let the selected subspace θ.

Step 2-6. If $\bar{x}_i^\theta - \underline{x}_i^\theta \leq k_i, \forall i$ or $t = T_l$, then go to Step 3. Otherwise, Generate next search range with vertices at the representative points $[x_i^{j-1}, x_i^{j+1}]$ in the neighboring subspace centered on θ, and go to Step 2-2 with update the number of iteration as $t = t + 1$.

Step 3. Optimize number of units of each layer (Detailed search using Tabu search)

Step 3-1. Randomly generate an initial solution $\boldsymbol{x^0} = (x_1, x_2, \ldots, x_{n^*})$ from the subspace θ, let $\boldsymbol{x}^* = \boldsymbol{x^0}$ be the current best solution, and E^* be the evaluation value for $\boldsymbol{x}^*$. The solution $\boldsymbol{x^0}$ is recorded in the tabu list. Set the number of iteration as $t = 0$.

Step 3-2. Evaluate each neighbor of the current solution $\boldsymbol{x}^t$ which are not included in the tabu list, and let $\hat{\boldsymbol{x}}^t$ solution with highest evaluation value in the evaluated solutions, and $\hat{E}^t$ be the evaluation value of $\hat{\boldsymbol{x}}^t$.

Step 3-3. Add the solution $\hat{\boldsymbol{x}}^t$ in the tabu list. If the number of solutions recorded in the tabu list is larger than the predetermined size of tabu list, the earliest recorded solution in the list is deleted instead of $\hat{\boldsymbol{x}}^t$.

Step 3-4. If $\hat{E}^t > E^*$, then let $E^* = \hat{E}^t$, and $\boldsymbol{x}^* = \hat{\boldsymbol{x}}^t$.

Step 3-5. If $t = T_{tb}$, then let $\boldsymbol{x}^*$ be the best solution and terminate the structural optimization procedure. Otherwise, let $t = t + 1$ and go to Step 3-2.

The rest of this section describes the procedure of structure evaluation, optimization of number of layers, number of units of each layer, and modularization of RBM for effective learning of DBN.

To optimize the structure of a DBN, considering the number of hidden layers and the number of units simultaneously is required. However, in this paper, after optimizing the number of hidden layers of DBN first, optimize the number of units of each hidden layer. Even when structural optimization is conducted in such order, verification experiments on the relation between DBN structure and data prediction accuracy are conducted to verify whether same structure are obtained or not, compared to a optimization procedure such that both are taken into consideration simultaneously. In the experiment, the accuracy for unknown data D_3 is calculated by using DBN where the number of units of each n hidden layer is fixed to 500. Additionally, the accuracy for D_3 is calculated by using DBN where the number of units of each n hidden layer is optimized by the method described in the next section. The 3-category image data is used for the experiment, and the classification accuracy for the data D_3 is set as the verification result. The experimental results are shown in Figure 3.

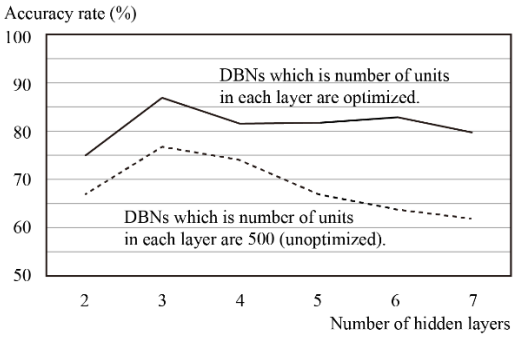

Figure 3. Relation between optimized and unoptimized number of units of each hidden layer.

From Figure 3, the data classification accuracy is highest when the number of hidden layers is 3 in both of two types of variation experiments indicated by a broken line and a solid line. In other words, DBN showed

that the superiority and inferiority relationship of performance based on the number of hidden layers is the same irrespective of whether the number of units of each hidden layer is optimized. In the experiment corresponding to Figure 3, only experiments using 3-category image data are shown. However, similar relationships are observed in other prior experiments. Therefore, the optimal number of hidden layers of DBN is determined that is with the highest performance fixing the number of each layer as 500 first. Subsequently, the numbers of units of all hidden layers are optimized.

3.2 Structure Evaluation

As the number of components of the neural network increases, the accuracy for learning data increases. However, the generalization ability for unknown data decreases. On the contrary, if the constituent elements of a neural network are small, the features of the target data cannot be properly learned. Therefore, it is desirable that the structure of a neural network is not only the prediction error with respect to the learning data, but also the prediction error with respect to the data not used for learning. In this paper, the target data is divided into three and evaluate the network structure by the following procedure.

At first, divide the target data D into three dataset (D_1, D_2, D_3) for learning, generalization verification, and test. Error back propagation is conducted using data D_1 and the verify generalization capability is evaluated based on output error when input data of D_2 is given to the learned neural network. Let e_{tr} be training error for data D_1 and e_{ve} be the output error for data D_2, i.e., e_{ve} represents the generalization capability. Let T be the number of data, $M\ be$ the number of units of the output layer, $O_j(t)$ be the output value from the j-th unit in the output layer of the neural network at the period t, and $Y_j(t)$ be the j-th factor of the target value. The error is defined by the mean square error between the target value and the output of neural network output as

$$e_A = \frac{1}{T}\sum_{j=1}^{M}\sum_{i=1}^{T}\left(O_j(i) - Y_j(i)\right)^2, A = tr, ve. \tag{3}$$

Based on these criteria of error, structure of a neural network A^k is evaluated by

$$E(A^k) = \frac{1}{e_{tr} + e_{ve}}. \tag{4}$$

3.3 Optimization of Number of Hidden Layers

Let $\underline{n}$ and $\overline{n}$ be minimum and maximum number of hidden layers, respectively. In the related literature [16], 500 is employed for the number of units of each hidden layer of DBN. Similarly, this paper employs 500 for the number of units of each layer in Step 1. Set the initial number of hidden layers be $\underline{n}$ and the number of hidden layers is added one by one up to $\overline{n}$. DBNs with n $(n = \underline{n}, \underline{n}+1, \ldots, \overline{n}-1, \overline{n})$ hidden layers such that the number of units of each hidden layer is 500 is evaluated based on the evaluation function (4). A DBN with highest evaluation value is selected and let n^* be the corresponding number of hidden layers.

3.3.1 Optimization of Number of Units of Each Layer

After the number of hidden layers n^* of DBN is determined, the number of units of each hidden layer should be determined. Let $(x_1, x_2, \ldots, x_{n^*})$ be a n^* dimensional solution in the solution space $X = \prod_{i=1}^{n^*}[\underline{x}_i, \overline{x}_i]$, where $x_i \in [\underline{x}_i, \overline{x}_i]$ is the number of the hidden layer. There exist numerous solutions in the space X. Therefore, the search space is divided and generate small areas to search and realize efficient

search by the following procedure. Here, let k_i be the number of division of the dimension of the solution space corresponding to the i-th hidden layer.

Let $d_i^{j_i} \equiv \left[\underline{x}_i^{j_i}, \overline{x}_i^{j_i}\right], j_i = 1,2,\dots,k_i, \underline{x}_i^1 = \underline{x}_i, \overline{x}_i^{k_i} = \overline{x}_i$ be the j-th interval of a subspace of i-th dimension x_i of the divided solution space, and $D_{\boldsymbol{j}} \equiv \Pi_{i=1}^{n^*} d_i^{j_i}$ be the divided subspace, where $\boldsymbol{j} = (j_1, j_2, \dots, j_{n^*})$. Select the center of gravity $(x_1^{\boldsymbol{j}}, x_2^{\boldsymbol{j}}, \dots, x_{n^*}^{\boldsymbol{j}})$ as a representative point of subspace $D_{\boldsymbol{j}}$. A subspace with the highest evaluation value (4) among the representative points of $\prod_i k_i$ subspaces is defined as $\boldsymbol{j}^* = (j_1^*, j_2^*, \dots, j_{n^*}^*)$.

Let a superior rectangular parallelepiped whose vertices are $x_1^{j^*-1}, x_1^{j^*+1}, x_2^{j^*-1}, x_2^{j^*+1}, \dots, x_{n^*}^{j^*-1}, x_{n^*}^{j^*+1}$ be new search space, and repeat the above steps until $\overline{x}_i^{j_i} - \underline{x}_i^{j_i} \leq k_i, \forall i$ is satisfied. If the condition $\overline{x}_i^{j_i} - \underline{x}_i^{j_i} \leq k_i, \forall i$ is satisfied, in other words, the division of the solution space is completed, the optimal solution of the network structure is searched by using taboo search described in the next section. As an example, the procedure of division of the solution space with $n^* = 2$ is shown in Figure 4.

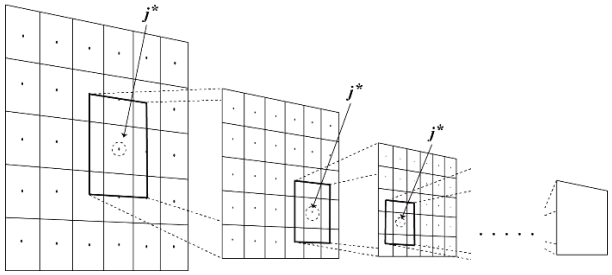

Figure 4. Division of the Solution Space ($n^* = 2$)

3.3.2 Structural Optimization by Tabu Search

In general, a pair of neural networks which have similar structures have similar performance to each other. In this paper, the optimal network structure is chosen by tabu search [6] which is one of evolutionary computation methods based on the neighbor search, from the solution subspace divided according to the above mentioned procedure.

3.4 Modularization of Structure of RBM

The DBN consists of connected plurality of RBMs and FNN. The learning process of the parameters such as connection weight and biases are performed in order from the RBMs closer to the input layer. In this paper, in order to avoid the redundancy of the structure evaluation in the structure optimization procedure of the DBN, a RBM utilizes the past learning information of another RBM which has common structure partially, input information to the RBM, number of units of a hidden layer, and the number of hidden layers, by following procedure.

For example, consider DBN1 including η_1 hidden layers which is the learning process is completed and DBN2 including η_1 hidden layers which is not completed. Let $\eta \geq \min\{\eta_1, \eta_2\}$, assume that the number of units of each hidden layer from the first layer to $(\eta - 1)$-th layer of DBN1 and DBN2 are all the same, and the number of units of the hidden layer of the η-th layer is different. Learning from the remaining the η-th to the η_2-th layer of DBN2 is performed by using the connection weights and biases of each hidden

layer from the first to the $(\eta-1)$-th layer of DBN1 in the learning process of DBN2. This mechanism improves learning efficiency of DBNs with different network structures.

4 Numerical Experiments

Caltech101 [17] are used as benchmark of image classification in many related literature. The image data of the Caltech 101 are grayscale 20×20 grids images, and each grid is scaled in the range of $[0,1]$. Images are classified into 4 categories "airplane", "cat", "face", "dolphin". There are 65 images per a category. This paper performs the following 2 kinds of experiments using Caltech101.

- 3-category classification experiment using 3 categories, “airplane”, “cat”, and “face”.
- 4-category classification experiment using 4 categories.

As a comparative methods, DBN without structural optimization, FNN, structural optimized FNN [20], k-means method are employed. Here, the number of hidden layers of a DBN without structural optimization is set to 3, and the number of each hidden layer unit is 500, 500, 2000. The number of hidden layer of a FNN without structural optimization is set to 3, and the number of hidden layer units was set to 200, 200 and 800, respectively. For a data classification problem with m categories, the number of units of an output layer is set to m, and that data is classified into a certain category corresponding to the unit such that output value is maximum in all output units. The experiments are conducted 10 trials, and the average value of classification accuracy is shown in Table 1 as experimental results.

Table 1. Image Classification Test: Result (accuracy (%))

Method	3−category	4-category
DBN with structural optimization (Proposed method)	**85.0**	**74.7**
DBN without structural optimization	77.1	62.2
FNN with structural optimization [20]	75.2	61.8
FNN without structural optimization	59.8	40.1
k-means method	58.6	37.4

From Table 1, the proposed method has the highest performance, and this experimental result indicates that the proposed method succeed to discover the appropriate structure of DBNs to increase the data classification accuracy. In the case of 3-category classification, structure of all DBNs obtained by the proposed method have 3-layer structure in all 10 trials. The average value of the number of units of the hidden layers are 454.7, 1834.5, and 2935.9 from the closer to the input layer, respectively. In each trial, numbers of units of hidden layers are similar to each other. Also, in the case of 4-category classification, DBNs with a 5-layer structure are obtained in all 10 trials. The average value of the number of units of the hidden layers are 457.0, 212.9, 2046.9, 1109.5, and 5974.9 from the closer to the input layer, respectively. Same as 3-category classification, almost same structure are obtained in all trials.

In the structure optimization of DBN by the proposed method, the solution space are divided into multiple subspaces first and solution search procedure are performed intensively in the promising regions. Such searching process can realize both diverse and intensive solution search and stably discover appropriate

structure. Additionally, it is also successful to improve the computational efficiency by modularization focusing on that the DBN has a structure in which a plurality of RBMs are superimposed.

5 Conclusion

This paper proposes structure optimization method for a DBN (Deep Belief Network) which consists of multiple RMBs (Restricted Boltzmann Machine) and a FNN (Feedforward Neural Network). The features of the proposed method are that it realizes searching both in wide range of solution areas by division of solution space and intensive search by tabu search, introduces modularization of RBMs to improve the calculation efficiency drastically by reducing the calculation amount in solution search. Numerical experiments using multiple categories image data indicates that it succeed in obtaining appropriate structure of DBN with high data classification accuracy by the proposed structural optimization method for DBNs. To develop a network structure optimization method that supports data analysis for high dimensional time series data such as voice data can be one of the future works.

REFERENCES

[1] [AgrawalEtal08] S. Agrawal, Y. Dahora, M.K. Tiwari and Y.-J. Son, "Interactive particle swarm: a Pareto-adaptive metaheuristic to multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38, pp. 258-277, 2008.

[2] [Bengio09] Y. Bengio, "Learning deep architectures for AI", M. Jordan eds*., Foudations and Trends in Machine Learning*, 2, Berkley, CA, USA, pp. 1-127, 2009.

[3] [Chen08] E. Chen, X. Yang, H. Zha, R. Zhang and W. Zhang, "Learning object classes from image thumbnails through deep neural networks", *International Conference on Acoustics, Speech and Signal Processing*, pp. 829-832, 2008.

[4] M. Delgado, M.P. Cuellar and M.C. Pegalajar, "Multiobjective hybrid optimization and training of recurrent neural networks," *IEEE Transactions on Systems, Man, And Cybernetics-Part B: Cybernetics*, 38, pp. 381-403, 2008.

[5] J.L. Elman, "Finding structure in time," *Cognitive Science*, 14, pp. 179-211, 1999.

[6] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.

[7] T. Hayashida, I. Nishizaki and T. Matsumoto, "Structural optimization of neural network for data prediction using dimensional compression and tabu search," 2013 IEEE 6th International Workshop on Computational Intelligence & Applications Proceedings (IWCIA 2013), Hiroshima, Japan, pp. 85-88, 2013.

[8] T. Hayashida, I. Nishizaki and A. Suemune, "Structural optimization of recurrent neural networks using tabu search," *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, 27, pp. 638-349, 2015.

[9] G.E. Hinton, S. Osindero and Y.-W. Teh, "A fast lerning algorithm for deep belief nets," *Neural Computation*, 18, pp. 1527-1554, 2006.

[10] G.E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, 313, pp. 504-507, 2006.

[11] B.Q. Huang, T. Rashid and M.-T. Kechadi, "A new modified network based on the elman network," *Proceedings of IASTED International Conference on Artificial Intelligence and Application*, 1, pp. 379-384, 2004.

[12] B.Q. Huang, T. Rashid and M.-T. Kechadi., "Multi-context recurrent neural network for time series application", *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 1, pp. 3073-3082, 2007.

[13] Y. LeCun, L. Bottou, L. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, pp. 1-46, 1998.

[14] M.I. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, E. Cliffs, NJ: Erlbaum, pp. 531-546, Reprinted in IEEE Tutorials Series, New York: IEEE Publishing Services, 1990.

[15] H. Katagiri, I. Nishizaki, T. Hayashida and T. Kadoma, "Multiobjective evolutionary optimization of training and topology of recurrent neural networks for time-series prediction," *The Computer Journal*, 55, pp. 325-336, 2011.

[16] Y. Liu, S. Zhou and Q. Chen,``Discriminative deep belief networks for visual data classification'', *Pattern Recognition*, 44, pp. 2287-2296, 2011.

[17] F.-F. Li, R. Fergus and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories", *Computer Vision and Image Understanding*, 106, pp. 59-70, 2007.

[18] [Martens10] J. Martens, "Deep learning via Hessian-free optimization," *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

[19] [McCullochPits43] W.S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, 5, pp. 115-133, 1943.

[20] M. Nishida, T. Hayashida, I. Nishizaki, and S. Sekizaki, "Structual optimization of neural network and training data selection based on SOM," *2014 IEEE SMC Hiroshima Chapter Young Researchers' Workshop Proceedings*, pp. 117-120, 2014.

[21] [Park13] D.-C. Park, "Structure optimization of bilinear recurrent neural networks and its application to ethernet network traffic prediction," *Information Sciences, 237, pp. 18-28, 2013.*

[22] [Prokhorov98] D.V. Prokhorov, E.W. Saad and D.C. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," *IEEE Transactions on Neural Networks*, 9, pp. 1456-1470, 1998.

[23] [QuocEtal11] V. Le Quoc, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow and A.Y. Ng, "On optimization methods for deep learning", *Proceedings of the 28 th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

[24] [SrivastraSalakhutdinov12] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," *Representation Learning Workshop*, 2012.

[25] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, Clearwater Beach, Florida, USA, pp. 448-455, 2009.

[26] [ScullerEtal11] B. Sculler, S. Steidl, A. Batliner, F. Schiel and J. Krajewski, *The interspeech 2011 Speaker State Challenge,* Florence, Italy, 2011.

[27] N. Srivastave, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Drop out: A simple way to prevent neural networks from overfitting", *Journal of Machine Learning Research*, 15, pp. 1929-1958, 2014.

[28] [VriesPrincipe90] B. de Vries and J.C. Principe, "A theory for neural networks with time delays," *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems,* 3, pp. 162-168, 1990.

[29] [DoSSweb] "DoSS@d", http://mo161.soci.ous.ac.jp/@d/index.html, (accessed 2017-12-13).

[30] [UCweb] "UC Irvine Machine Learning Repositor", http://archive.ics.uci.edu/ml/, (accessed 2017-12-13).