https://doi.org/10.20965/jaciii.2013.p0443

Paper:

A Neural Network Structure Decomposition Based on Pruning and its Visualization Method

Atsushi Shibata, Jiajun Lu, Fangyan Dong, and Kaoru Hirota

Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology
G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan
E-mail: {shibata, lu, tou, hirota}@hrt.dis.titech.ac.jp

[Received December 2, 2012; accepted March 25, 2013]

To decompose neural network structures for composite tasks, a pruning method and its visualization method are proposed. Visualization by placing the neurons on a 2D plane clarifies the structure related to each composited task. Experiments on a composite task using two tasks from a UCI dataset show that the neural network of the composite task contains more than 80% of neurons. The proposed methods target the transfer learning of robot motion, and results of an adaptation experiments are also referred.

Keywords: neural network, pruning, visualization, UCI dataset, 6 leg robot

1. Introduction

The classification of surrounding states and acquisition of corresponding motions for robots that carry out composite tasks are the main issues when a flexible response to a rapidly changing environment is considered. Among the studies of neural networks in robotics applications for classification and motion learning purposes are methods like the preparation of multiple learners for different states [1] and the learning of motions from automatically classified input [2]. With the increase in new states and acquired motions, the ballooning of data size becomes a salient issue. The optimization of network structures is studied for alleviating computation cost. Different pruning methods by evaluating the averaged output of neurons [3, 4], the sensitivity of teacher signals [5], or normalized terms [6] have been well studied for the structural optimization of neural networks. All of these methods of pruning are designed mainly for single tasks, whereas tasks in the real world, in contrast, are usually composite combination of single tasks with unknown relationships. When these pruning methods are introduced to networks dealing with composite tasks, there is the possibility that learned network structures may be damaged. To solve this problem, the portion of network structures related to individual tasks has to be modified by removing unwanted neurons and connections.

We propose a pruning method and its visualization method. The proposed pruning decomposes neural network structures that correspond to each task, and the proposed visualization shows its relationship via the arrangement of neurons in 2D space. Concretely, rewards and lifetimes are defined for pruning purposes. The generated reward is only given to the neurons that are related to the generation of reward. The lifetime of neurons is updated using the given reward so that neurons and connections that hinder the generation of rewards are removed. The concept of neuron space is also defined and neurons are allocated in the neuron space. Concretely speaking, neurons converge or spread according to their mutual relevance so that structures of neural networks related to specific tasks are displayed in 2D neuron space. Using the pruning method of rewards and lifetimes, structures that are locally suited from a composite task to individual decomposed tasks are likely to be formed. Also, by viewing the placement of neurons in neuron space after learning, the network structure of a composite task is understood intuitively.

To verify that the proposed pruning method does not hinder the process of learning, a comparison of convergence between the Back Propagation (BP) method and the BP method with the proposed pruning method is carried out. Comparison experiments are also done between the proposed pruning method and the Dynamic Node Decaying Method (DNDM) [3, 4] to determine the accuracy of the proposal in the learning process of neural networks. The UCI dataset [7] is used as the benchmark for these experiments. To verify the visualization of composite tasks by the proposal, two tasks are selected from the UCI dataset and combined to form a composite task. The composite task is learned using the BP method with proposed pruning method. The composite task is evaluated using averaged task ratios of single tasks in each cluster formed by k-means. Based on the BP method with the proposed pruning method and the visualization method of neurons, an experiments on the learning of walking motion using a six-legged robot that is installed in the environment of PhysX^{TM} [8] is also carried out to confirm the applicability of the proposal in the area of autonomic robot learning.

In Section 2 of this paper, the visualization of the structure of neural networks and pruning based learning methods using rewards and lifetimes are introduced. Experiments on UCI datasets and the learning of a simulated six-legged robot are done in Section 3. Conclusions are presented in Section 4.

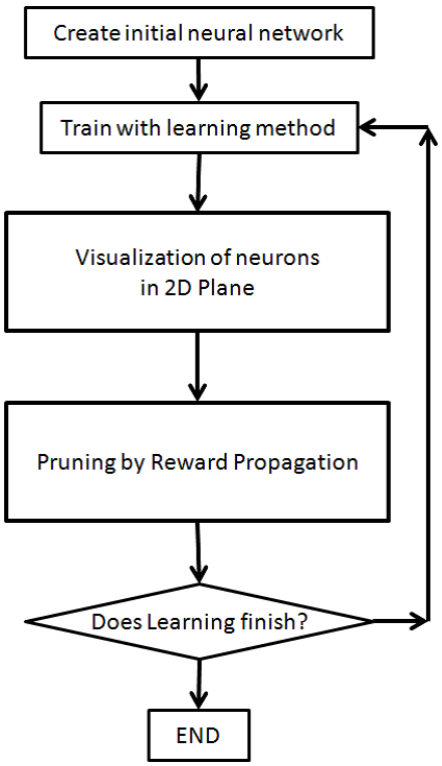


Fig. 1. Flowchart for neural networks with two proposed methods.

2. Pruning-Based Structure Decomposition and Visualization

Two proposed methods, visualization and pruning for network decomposition, are explained in this section. A flowchart of the system used in this experiment is depicted in **Fig. 1**. The proposed methods can be applied among existing learning methods.

2.1. Visualization of Neurons in 2D Plane

2D Euclidean space in which neurons are allocated is defined as neuron space. To visualize network structures, each neuron is given position vector $\mathbf{x}$, and arranged in neuron space. For human visual perception, $\mathbf{x}$ has two elements, $\mathbf{x}^1$ and $\mathbf{x}^2$.

Initial location $\mathbf{x}_{i(0)}$ of neuron i is given randomly. And location $\mathbf{x}_{i(t)}$ at time t is updated by adding $\Delta\mathbf{x}_{i(t)}$ as

$$\mathbf{x}_{i(t+1)} = \mathbf{x}_{i(t)} + \Delta\mathbf{x}_{i(t)}, \quad \ldots\ldots\ldots \quad (1)$$

$$\Delta\mathbf{x}_{i(t)} = \delta \sum_{m \in M} \left(\gamma \frac{e_{r_{im}}}{|r_{im}(t)|} - |w_{im}| r_{im}(t) \right), \quad \ldots \quad (2)$$

$$r_{im}(t) = \mathbf{x}_{i(t)} - \mathbf{x}_{m(t)}, \quad \ldots\ldots\ldots \quad (3)$$

where M denotes the total number of neurons in the neuron space, r_{im} is a vector from neuron i to neuron m in the neuron space, $e_{r_{im}}$ represents a unit vector in the same direction as r_{im}, and $|w_{im}|$ is the absolute value of weight between i and m. Note that if there is no connection between two neurons, 0 is assigned as the value of $|w_{im}|$. The quantity of $\Delta\mathbf{x}_{i(t)}$ is altered by adjusting two parameters, that is, attraction ratio γ and magnification ratio of movement δ. When two neurons overlap, one is moved randomly to a neighboring location.

By applying this idea, neurons with strong connections are made more adjacent and in contrast, neurons with no connection are kept far away from each other. Neurons with a commonness in processing information therefore are concentrated in one place, while neurons with different processing information are placed away from each other. In addition, the distribution range of neurons is adjusted by γ and δ. When a new neuron is added to neuron space, connections between this new neuron and existing ones are determined by distance in neuron space.

2.2. Pruning by Reward Propagation

The proposed pruning method is divided into two parts. In the first part, it is determined whether neurons and connections are removed or not. In the second part, reward to be used to determine this are propagated by the rules detailed in the subsections that follow.

2.2.1. Removal of Neurons and Connections Using Lifetimes

For both neurons and connections, $life(t)$ quantity is defined as time t. The lifetime of a firing neuron and the connection on its output side is updated as

$$life(t+1) = I(\alpha \cdot life(t) + \beta \cdot reward(t)), \quad \ldots \quad (4)$$

$$I(x) = \begin{cases} 1 & (x \geq 1) \\ x & (1 \geq x > \theta) \\ 0 & (\theta \geq x) \end{cases}, \quad \ldots\ldots\ldots \quad (5)$$

where parameter $\alpha\,(\in [0,1])$ is the attenuation rate of a lifetime, parameter $\beta\,(>0)$ is the recovering rate of lifetime due to a reward, $reward(t)$ (≥ 0) denotes the reward of a neuron propagating through connection at time t, and parameter $\theta\,(\in [0,1])$ is the threshold for a lifetime. If lifetime *life* is below threshold θ, then related neurons and connections are deleted.

In cases where there is a large deviation in training data, it is necessary to adjust parameters α, β, and θ. By setting α to a value of 0.9, neurons and connections gradually attenuate if the reward is low or no reward is obtained at firing. For the same reason, β should be more than 1 and θ should be less than 0.1.

2.2.2. Propagation Rules of Rewards

Rewards are generated in accordance with the output of a neuron network and distributed to neurons in the output layer. A neuron that has obtained its reward imparts the reward to neurons that have contributed to the output of which the reward is generated. A brief example that shows the propagation of reward is given in **Fig. 2**. As is depicted in **Fig. 2**, i is the neuron that gets the reward. The set of neurons of which output y is connected to i is denoted as N. j and k are two neurons included in set N. When the reward is propagated from i, candidates for

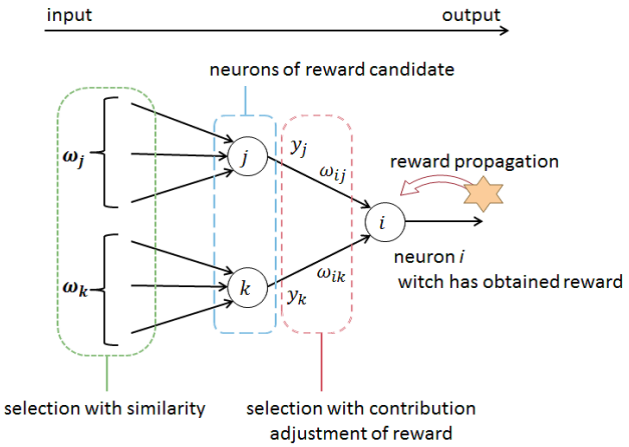


Fig. 2. Example for reward propagation.

Table 1. Selection of reward candidate using firing state and weight.

	j is firing		j is not firing	
	$w_{ij} \geq 0$	$w_{ij} < 0$	$w_{ij} \geq 0$	$w_{ij} < 0$
i is firing	○			○
i is not firing		○	○	

receiving the reward are chosen from set N. Candidate neurons are selected and get rewards according to three steps, i.e., their firing states, the weight of connections, and the similarity of firing conditions. The value of the imparted reward depends on the input to neuron i.

In the first step, connections that contribute to the reward acquisition of neuron i are selected (**Table 1**). If neuron i is firing when getting the reward, then connections with non-negative weight and with firing signal and connections with negative weight and without firing signals are selected. If neuron i is not firing when getting the reward, then opposite methods for connection selection are used.

In the second step, among selected connections, those with a high degree of similarity are removed. According to [9], the degree of similarity S_{jk} between neuron j and k is figured out as follows,

$$S_{jk} = \frac{\left| w_j^T w_k \right|}{|w_j|\,|w_k|} \quad , \quad \ldots\ldots\ldots\ldots \quad (6)$$

where w_j is the vector of connection weight as the input to neuron j. A connection weight that is not in common with neuron k is set to 0. If calculated S_{jk} exceeds the threshold for the degree of similarity, the corresponding neuron is removed from among candidates to whom the reward is propagated.

In the third step, the exact value of the reward propagated from neuron i to the receiving neuron j is computed as

$$reward_{ji}(t) = \frac{\left| w_{ij} y_i \right|}{\sum_{n \in N'} w_{jn} y_n} reward_i(t) \quad , \quad \ldots \quad (7)$$

where w_{jn} is the weight of the connection between neuron j and neuron n that receives a reward, y_n is the output of neuron n, N' is the set of candidate neurons for receiving rewards, and $reward_i(t)$ denotes the reward for neuron i at time t. $reward_{ji}(t)$ is considered to be the total amount of the reward that neuron j receives from neuron i. The reward of neuron j is obtained as

$$reward_j(t) = \sum_{m \in M} reward_{jm}(t) \quad , \quad \ldots\ldots \quad (8)$$

where m denotes a neuron that gives j some reward, and M is the set of m.

Using the rules explained in Section 2.2.2, the generated reward is propagated from the output layer to the input layer, and only neurons that contributed to the current task can restore their lifetimes.

3. Experiments on UCI Dataset and Six-Legged Robot

Three experiments are conducted for evaluating our two proposed methods. In Section 3.1, the proposed pruning method is confirmed and compared to the existing pruning method. In Section 3.2, the two proposed methods confirm the pruning method's visualization on 2D neuron plane. In Section 3.3, the proposed methods are applied to adaptation experiments on robot motion.

In these experiments, the proposal is incorporated into a three layered feed-forward neural network, using a sigmoid function and a BP method.

3.1. Pruning and Learning of Single Task in UCI Dataset

To confirm the performance of the proposed pruning method, the influence of pruning on learning is evaluated in experiments. Comparative experiments between the DNDM [3, 4] and the proposed pruning method were also carried out.

In these experiments, the UCI dataset, which is widely utilized as the benchmark for classification problems, was chosen as the single task. Three decomposed datasets were chosen from UCI dataset: a WDBC dataset (with 699 samples), a glass dataset (with 214 samples), and an iris dataset (with 150 samples).

Before experiments, the initial parameters for the neural network are set, i.e., α was set to 0.9, β was set to 1, and θ was set to 0.1. In order to match with the previous studies [3, 4], the learning rate is set to 0.05, and the initial region for weights is set to ± 0.1 for generating initial neurons. Each time output is generated, 0.8 times the total number of neurons in the neural network is used as the size of the reward. The reason for relating the size of the reward to the total number of neurons is that the averaged number of connections for each neuron is maintained, independent of neural networks.

To show that the proposed pruning method does not hinder the learning process of neural networks, the learn-

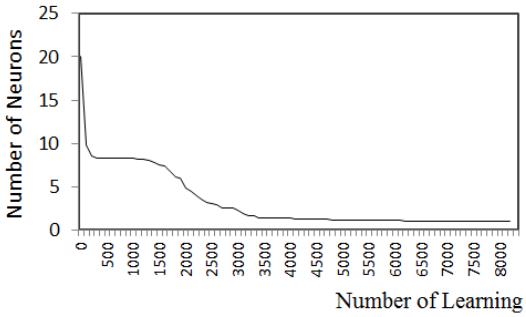


Fig. 3. Change of the number of neurons in hidden layer using the proposed pruning.

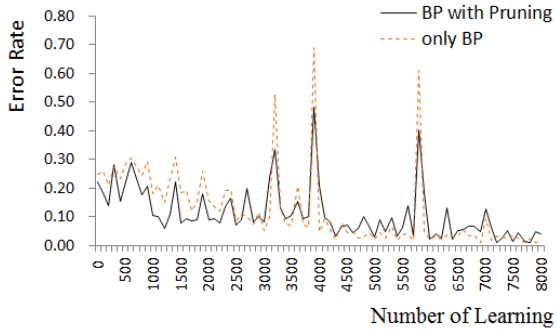


Fig. 4. Error rate comparison between BP method with proposed pruning and BP method alone.

Table 2. Comparison of number of neurons in hidden layers.

		WDBC	Glass	Iris
Proposed Pruning Method	Mean	1.0	6.0	2.1
	Variance	0.0	4.0	0.8
	Min	1.0	3.0	1.0
	Max	1.0	9.0	4.0
DNDM	Mean	2.4	4.0	2.7
	Variance	1.4	1.1	1.5
	Min	1.0	3.0	1.0
	Max	6.0	7.0	7.0

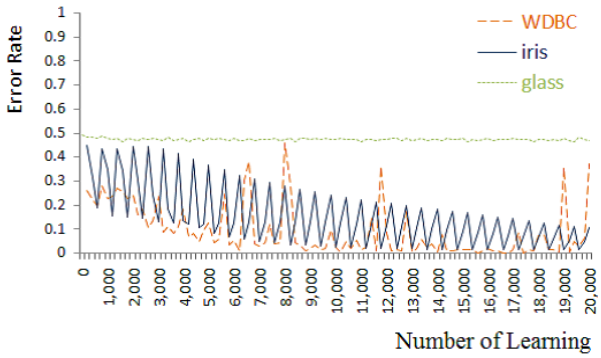


Fig. 5. Error in learning process for three datasets.

ing time of the BP method is compared to that of the modified BP method into which the pruning method is incorporated. The WDBC dataset is used for learning. The learning is conducted for 8000 times averaged in 10 trials, and the average is evaluated.

Figures 3 and **4** show results of experiments for the single task. **Fig. 3** shows that the number of neurons in the hidden layer is reduced to about 8 in the early stage of the experiment, which indicates that the proposed pruning method tends to remove neurons that do not work well in learning. In contrast, **Fig. 4** shows that the error rate of the proposal keeps a low level compared to that of the BP method alone. The proposed pruning method thus improves the learning process of neural networks.

The BP method with the proposed pruning is compared with DNDM [3, 4] to confirm the accuracy of the proposed pruning method. Learning for the three datasets – WDBC, glass, and iris [7] – for both methods is done in 10 trials, and the mean value of numbers of neurons in the hidden layer and variance are chosen to be the evaluation criteria as shown in **Table 2**.

The BP method with proposed pruning outperforms the DNDM in terms of neurons remaining in the hidden layer for the WDBC dataset. For the glass dataset, the DNDM works better, while for the iris dataset, the proposed pruning method again has a slightly better performance.

The reason why the proposal did not work well for the glass dataset is explained using **Fig. 5**. The error curve of learning for the glass dataset shows that the convergence of the learning process is slow compared to that of the other two datasets, which is in turn due to an imbalance in teacher signals for the glass dataset.

3.2. Visualization of Composite Task in UCI Dataset

To confirm decomposition of the network structure, a neural network learned a composite task, combined from the UCI dataset with proposed pruning and visualization. Evaluation is done based on neuron placement and connection status. The remaining number of neurons in hidden layers after learning is also considered. The WDBC dataset and the iris dataset used in Section 3.1 are highly mutually independent. The two datasets are combined to produce a new composite task containing 480 samples. The same initial parameters as those used in Section 3.1 are applied for generating a neural network. Parameters for the proposed pruning are carefully tuned using an attenuation rate of lifetime α set to 0.98, the recovering rate of lifetime β set to 5.0, and the lifetime threshold set to 0.1, respectively. The initial number of neurons in the hidden layer is set to 20. The learning process terminates when average square error between output and the teacher signal is lower than 0.05. Each learning process is carried out for 4 trials. Results for the number of neurons in the hidden layer after learning are shown in **Table 3**, together with the results of single tasks from **Table 2**. When a composite task (WDBC+iris) is used to learn the BP method with the proposed pruning, accuracy becomes much lower in comparison to single task settings.

Table 3. Performance of pruning for a composite task.

	WDBC+iris	WDBC	Iris
Mean	10.0	1.0	2.1
Variance	5.5	0.0	0.8
Min	8.0	1.0	1.0
Max	14.0	1.0	4.0

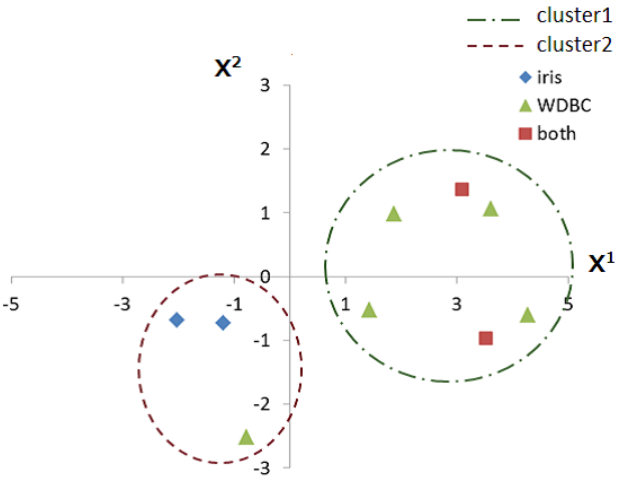


Fig. 6. Example of clustering in neuron space.

Table 4. Task ratios for the two clusters.

Trial	Cluster 1		Cluster 2	
	R_{WDBC}	R_{iris}	R_{WDBC}	R_{iris}
1	1.0	0.0	0.0	0.8
2	0.6	0.0	0.0	0.9
3	0.9	0.0	0.0	1.0
4	1.0	0.0	0.3	0.7

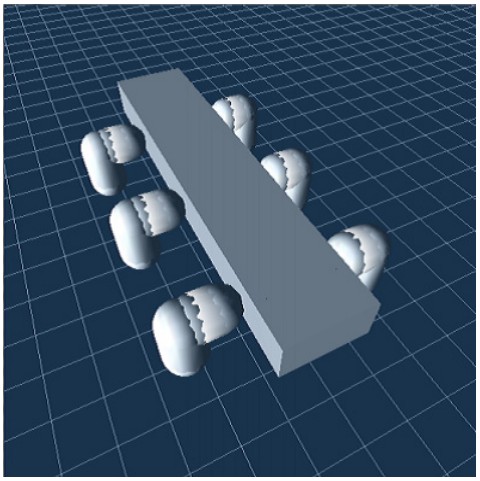


Fig. 7. Profile of a six-legged robot.

After learning, neurons in the hidden layer are decomposed into two clusters based on location in neuron space. To confirm these neurons' express network structure, we compare clusters and categories of neurons. Neurons are divided into three categories by input, namely WDBC, iris, and WDBC+iris (the composite task) and clustered into two clusters by k-means of OpenCV [10]. An example of neuron location in the neuron space after clustering using k-means is shown in **Fig. 6**. In both clusters, numbers of neurons whose input comes from a single task surpass those from the other single task.

Task ratios for single tasks are defined as the ratios of neurons whose input is from a single task (WDBC or iris) in one cluster. The task ratio for the WDBC dataset and the iris dataset are denoted as R_{WDBC} and R_{iris}. R_{WDBC} and R_{iris} are computed as

$$R_{WDBC} = \frac{N_{WDBC}}{N_{cluster}}, \quad \ldots\ldots\ldots\ldots (9)$$

$$R_{iris} = \frac{N_{iris}}{N_{cluster}}, \quad \ldots\ldots\ldots\ldots (10)$$

where N_{WDBC} denotes the number of neurons whose input comes only from the WDBC dataset, N_{iris} represents the number of neurons whose input come only from the iris dataset, and $N_{cluster}$ denotes the number of neurons in one cluster. A dataset that is related to a larger number of neurons in one cluster is considered the task for that cluster. Neural network structure decomposition is evaluated by task ratios. Average task ratios for both clusters for 4 trials are shown in **Table 4**. The averaged task ratio for both clusters is 0.8, which is explained the same as that for the neural network of a composite task composed of more than 80% neurons in each task.

3.3. Visualization of Tasks in a Robot Motion

We expend our proposal to visualization experiments on robot motion based on a simulated six-legged robot installed in PhysX™ [8].

The profile of the six-legged robot is shown in **Fig. 7**. The robot has six legs – three on each side of its body. The three pairs are positioned on either side of the front, middle, and back of its body. Each leg has two joints – one connecting the leg and body, and the other connecting the upper and lower part of the leg. The movable angle of each joint is set at 60°.

To assist in the motion of the robot, the legs are move alternately. Concretely speaking, the left leg at the front, the right leg at the middle, and the left leg at the back are designed to share the same motion, as do the remaining three legs. The four torque values for the two forelegs are considered to be input for the robot and the other four legs move accordingly.

A neural network is used, that has four neurons in the output layer and eight neurons in the input layer. Neurons in the output layer correspond to four inputs for the robot, and neurons in the input layer receive eight angles – current and previous four forelegs angles – as input.

The BP method with proposed pruning is utilized to construct the neural network for controlling robot walk-

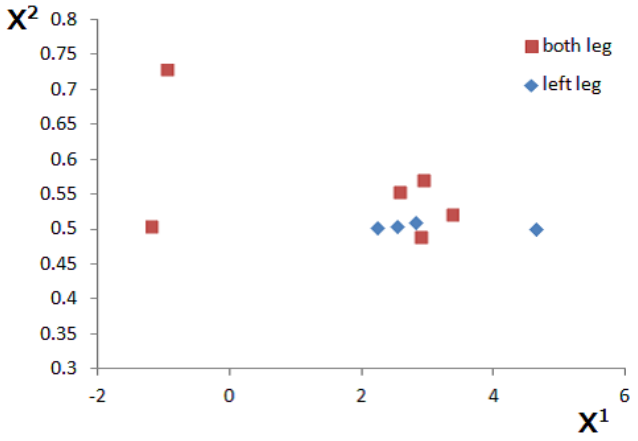


Fig. 8. Locations of neurons in neuron space for learning walking.

ing. Input to the forelegs is set as tasks and locations of neurons in neuron space are evaluated.

An example of neuron locations in neuron space after walking motions is learned is shown in **Fig. 8**. In this case, no neurons correspond only to the right foot. In addition, neurons are densely assembled along the $\mathbf{x}^1$-axis. In other cases, neurons are centralized in a small region.

From these results, tasks for learning walking are considered mutually associated.

4. Conclusions

Experiments on learning using our proposal of a pruning method and its visualization method have been conducted using a PC with a dual core processor (2.5 GHz) and 2 GB memory. And programming language is in C++. The physics engine used for the simulation of a six-legged robot is PhysX^{TM} SDK 2.8.1.

Experiments on single tasks – WDBC, glass, and iris dataset – from the UCI dataset using the BP method with our proposed pruning method have shown that the pruning method tends to remove neurons that do not work well in learning. The BP method with pruning leads to a lower error rate in comparison with the BP method alone. When the WDBC and iris datasets are learned using the BP method with pruning, the mean numbers of neurons in the hidden layer are reduced by 1.4 and 0.6 in comparison with those of the DNDM method. In the case of the glass dataset, the number of neurons in the hidden layer is increased by 2.0, which was mainly due to the imbalance of the teacher signal in this dataset.

Experiment on the composite task using the WDBC+iris dataset has shown that when composite task is used for learning the BP method with our proposed pruning method, accuracy drops noticeably in comparison to that of single task settings. When neurons in the hidden layer after learning are clustered into two clusters in neuron space, the average task ratio for both clusters is over 0.8. Concretely, task ratios for the WDBC and the iris datasets are 0.88 and 0.85, which is evidence showing that the structure of neural networks for this composite task is well decomposed by applying our proposal.

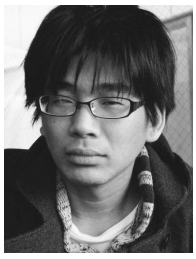
Results of the adaptation experiments on learning walking using a simulated six-legged robot have shown that neurons in the hidden layer after learning are centralized in a small region in the neuron space, which is mainly owing to relevance between tasks. The potential of our proposal in learning robot motion is also shown in these results.

Acknowledgements

This work was supported by the Japan Society for the Promotion of Science (JSPS) under grant Kakenhi C23500272.

References:

[1] K. G. Jolly, R. Sreerama Kumar, and R. Vijayakumar, "Intelligent task planning and action selection of a mobile robot in a multi-agent system through a fuzzy neural network approach," Engineering Applications of Artificial Intelligence, Vol.23, pp. 923-933, 2010.
[2] M. Okada, D. Nakamura, and Y. Nakamura, "Self-organizing Symbol Acquisition and Motion Generation based on Dynamics-based Information Processing System," The Japanese Society for Artificial Intelligence, Vol.20, No.3, SP-A, pp. 177-186, 2005.
[3] M. Shahjahn and K. Murase, "A Dynamic Node Decaying Method for Pruning Artificial Neural Networks," IEICE Trans. Inf. and Syst., Vol.E86-D, No.4, pp. 736-751, April 2003.
[4] M. Shahjahn and K. Murase, "A Pruning Algorithm for Training Cooperative Neural Network Ensembles," IEICE Trans. Inf. and Syst., Vol.E89-D, No.3, pp. 1257-1269, March 2006.
[5] S. Kikuchi and N. Nakamura, "Recurrent neural network with short-term memory and fast structural learning method," System and Computer in Japan, Vol.34, No.6, pp. 69-79, 2003.
[6] Z. Zhang and J. Qiao, "A Node Pruning Algorithm for Feedforward Neural Network Based on Neural Complexity," Int. Conf. on Intelligent Control and Information Processing, Dalian China, pp. 406-410, August 13-15, 2010.
[7] C. B. D. J. Newman, S. Hettich, and C. Merz, "UCI Repository of Machine Learning Databases," 1998. http://archive.ics.uci.edu/ml/
[8] http://www.geforce.com/hardware/technology/physx
[9] N. Higa and H. Shirotsuchi, "Studies on An Algorithm for Reducing Redundant Units Based on Geometrical Approach," The Institute of Electronics, Information and Communication Engineers, technical report, pp. 43-48, June 2002 (in Japanese).
[10] http://opencv.org/



Name:
Atsushi Shibata

Affiliation:
Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology

Address:
G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan

Brief Biographical History:

2005-2009 B.E., Aoyama Gakuin University
2010-2012 M.E., Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology
2012- Ph.D. Student, Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology

Name:
Jiajun Lu

Affiliation:
Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology

Address:
G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan

Brief Biographical History:
2007-2011 B.E., Shanghai Jiao Tong University
2011-2012 M.E., Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology
2012- Ph.D. Student, Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology

Main Works:
• J. Lu, F. Dong, and K. Hirota, "Non-Photorealistic Rendering for High Dynamic Range Images," The 5th Int. Symposium on Computational Intelligence and Industrial Applications (ISCIIA 2012), August 2012.

Name:
Fangyan Dong

Affiliation:
Assistant Professor, Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology

Address:
G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan

Brief Biographical History:
2006- Assistant Professor, Tokyo Institute of Technology

Main Works:
• F. Dong, K. Chen, E. M. Iyoda, H. Nobuhara, and K. Hirota, "Solving Truck Delivery Problems Using Integrated Evaluation Criteria Based on Neighborhood Degree and Evolutionary Algorithm," J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.8, No.3, pp. 336-345, 2004.
• F. Dong, K. Chen, and K. Hirota, "Computational Intelligence Approach to Read-world Cooperative Vehicle Dispatching Problem," Int. J. of Intelligent Systems, Vol.23, pp. 619-634, 2008.

Membership in Academic Societies:
• Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
• The Japanese Society for Artificial Intelligence (JSAI)

Name:
Kaoru Hirota

Affiliation:
Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology

Address:
G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan

Brief Biographical History:
1982-1995 Professor, College of Engineering, Hosei University
1995- Professor, Tokyo Institute of Technology

Main Works:
• M. L. Tangel, C. Fatichah, M. R. Widyanto, F. Dong, and K. Hirota, "Multiscale Image Aggregation for Dental Radiograph Segmentation," J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.16, No.3, pp. 388-396, May 2012.
• M. Dai, F. Dong, and K. Hirota, "Fuzzy Three-Dimensional Voronoi Diagram and its Application to Geographical Data Analysis," J. of Advanced Computational Intelligence and Intelligent Informatics, Vol.16, No.2, pp. 191-198, March 2012.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)
• International Fuzzy Systems Association (IFSA), Fellow, Immediate-Past-President
• Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT), Past-President