# Visualization of topographical internal representation of learning robots

Shiori Kuramoto
Dept. of Applied Physics
Waseda University
Tokyo, Japan
kuramoto@sawada.phys.waseda.ac.jp

Hideyuki Sawada
Dept. of Applied Physics
Waseda University
Tokyo, Japan
sawada@waseda.jp

Pitoyo Hartono
School of Engineering
Chukyo University
Nagoya, Japan
hartono@sist.chukyo-u.ac.jp

***Abstract*— The objective of this study is to understand the learned-strategy of neural network-controlled robots in relation to their physical learning environments by visualizing the internal layer of the neural network. During the past few years, neural network-controlled robots that are able to learn in physical environments are becoming more common. While they can autonomously acquire strategy without human supervisions, it is becoming difficult to understand their strategy, especially when the robots, their environments and their tasks are complicated. In the critical fields that involve human safety, as in self-driving vehicles or medical robots, it is important for human to understand the strategies of the robots. In this preliminary study, we propose a hierarchical neural network with a two-dimensional topographical internal representation for training robots in physical environments. The 2D representation can then be visualized and analyzed to allow us to intuitively understand the input-output strategy of the robots in the context of their learning environments. In this paper, we explain about the learning dynamics of the neural network and the visual analysis of some physical experiments.**

***Keywords— self-organizing map, explainable AI, hierarchical neural networks, reinforcement learning, autonomous robots***

## I. INTRODUCTION

The primary objective is the study to build a compact neural network for training autonomous robot that can be intuitively understood by human. In recent few years, the applications of neural networks have been proliferating in various fields. While the performances of some of them have exceeded human experts, most often they have to be treated as a black-box, in that their input-output characteristics are uninterpretable to human. The unexplanability and non-accountability of the neural networks can be problematic for their applications in some fields that involve human welfare and safety, as in self-driving vehicles, medical diagnostics and surgeries. Recently many studies are being conducted in explaining the behaviors of trained neural networks in a new field of Explainable AI (XAI) [1,2].

Many attempts to explain the strategic characteristics such as the relation between sensory inputs to a robot and its reactions that are governed by neural networks that often is difficult to understand for human, many study for understanding the input-output causality of neural networks are based on rule extractions, as in [3,4,5]. While they are successful in some cases, when the structures of the neural networks or the tasks to be learned are complicated, the rule extraction methods often generate complicated rules that are still uninterpretable for human. Hence, they do not help in increasing the accountability and transparency of the neural networks.

Other attempts are to visualize some aspect of the neural networks. In this approach, the activations of parts of the neural networks, for example some of the hidden layers in hierarchical neural networks, are treated as high dimensional vectors that to some extent explain the input-output relation of the neural networks. By applying some dimensionality reduction methods [6,7,8], they can be visualized. While the visualizations do not directly generate logical rules or mathematical functions, they allow human to have intuitive understanding on the input-output relation of the neural network. The visual interpretability often generates better understandability than complicated logical rules. However, in those past studies, the methods of dimensionality reduction are often detached from the learning algorithms of the neural networks to be interpreted. The detachment of dimensionality reduction methods from the learning algorithms reduces the relation between the information to be visualized and the actual context to be explained. In this study, we experimented on learning by building a hierarchical neural network in a PC to interactively communicate with an autonomous robot in physical environment. This hierarchical neural network has a topographical hidden layer. The topographical hidden layer is two-dimensional, so that it can be visualized and intuitively analyzed, and thus allowing human to understand the strategy of the robot in relation to its learning environment. As opposed to the previous studies where the learning algorithm and the dimensionality reduction method are detached, in this study the topological dimensionality reduction is integrated into the reinforcement learning mechanism, so that the visualization reflects the actual characteristics of the neural networks. The neural network model in this study can be implemented to a small physical robot that executes a real time reinforcement learning. This study is based on the past study, in which it was reported that topological initialization, in the form of Self Organizing Maps (SOM) [9, 10], allows better learning for robots in real world environments [11]. The proposed learning method here is different from the past method, in that in the past method the topological structure is used for initializing the neural network, while in this study, the topographical self-organization is integrated with reinforcement learning mechanism, resulting in a more comprehensive visualization. The reinforcement learning in this study is based on the previous study in [12,13]. This study is different from the previous ones, in that while the reinforcement learning mechanism is identical in the previous study, the hierarchical neural network for implementing the reinforcement learning does not have low dimensional internal layer, and thus cannot be visually interpreted.

In this study, we deal with collision avoidance learning of a real robot in various physical environments in real time. After the learning progressed, we visually analyzed the resulting internal low dimensional map.

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

Our focus is to understand the obstacle avoidance strategy of the robot related to its physical learning environments. We report that the resulting visual information allows us to some extent understand the input (an array of proximity sensors) and output (robot's actions) characteristics of the robots within its current environmental conditions. Although still in preliminary stage, we consider that our approach can contribute to giving better explainability and accountability especially to neural network-controlled robots that learn in physical environments.

This paper is organized as follows. Section 2 describes the learning visualization method used in this study. Section 3 describes the system used in this experiment. Section 4 describes the experimental results and considerations learned by this learning method. Finally, Section 5 gives a summary and future work.

## II. NEURAL NETWORK'S STRUCTURE AND DYNAMICS

The structure of the neural network (NN) in this study is shown in Fig. 1. The neural network is a three-layered network with an input layer, a two dimensional topographical hidden layered an output layer. The input layer receives sensory inputs from the robot, in which one neuron is associated with a particular sensor. The sensory inputs in this layer are propagated to the hidden layer to be self-organized while retaining the inputs topographical order, similar to that of Kohonen's Self Organizing Maps (SOM) [9, 10]. Different from SOM, the hidden neurons in this layer generate output based on the input's topographical similarities. Those outputs are then propagated to the output layer. As shown in Fig. 1, each neuron in the output layer is associated with a particular action that can be executed by the robot. It can be seen here, as in the conventional hierarchical neural networks, the neural network forms a function that reflects the strategy of the robot to convert sensory inputs into its movements. However, while in conventional neural networks the strategy is expressed as freely distributed activations of hidden neurons, here the hidden neurons are topographically constrained. The low dimensional topographical constrained of the hidden layer allows human user to visually and intuitively gain information on input-output characteristics of the robot.

While in conventional supervised neural networks, learning and running phases are separated, in the proposed neural network these two processes are integrated.

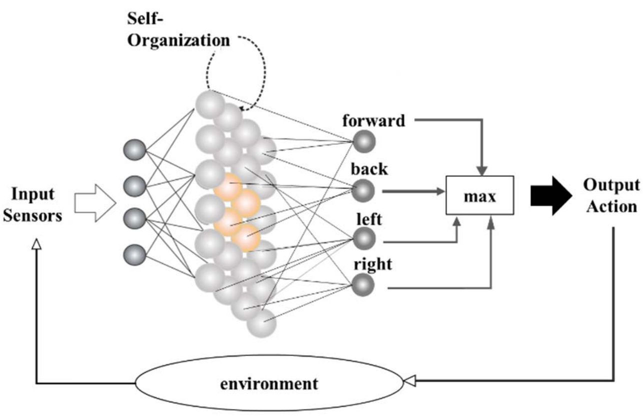



Fig.1. Outline of neural network

The dynamics are explained as follows. Receiving an array sensory inputs $X(t) \in \mathbb{R}^d$ at time $t$, the NN select a winner node, $win$, among the hidden neurons, as shown in Eq. (1), where $W_j(t) \in \mathbb{R}^d$ is the reference vector associated with the $j$th hidden neuron while $d$ is the number of sensors.

$$win = arg\min_j \left|X(t) - W_j(t)\right|^2 \quad (1)$$

Once the winner node is determined, the vector is modified according to Eq. (2), where the neighborhood function $\sigma(win, i, t)$ is defined as in Eq. (3).

$$W_i(t+1) = W_i(t) + \eta_1(\mathrm{t})\sigma(win, i, t)\big(X(t) - W_i(t)\big) \quad (2)$$

$$\sigma(win, i, t) = \exp\left(-\frac{dist(win, i)}{s(t)}\right) \quad (3)$$

Here, $dist(win, i)$ represents the Euclidean distance between the winner and the $i$th node on the map, and $s$ and $\eta$ are annealing and learning coefficients.

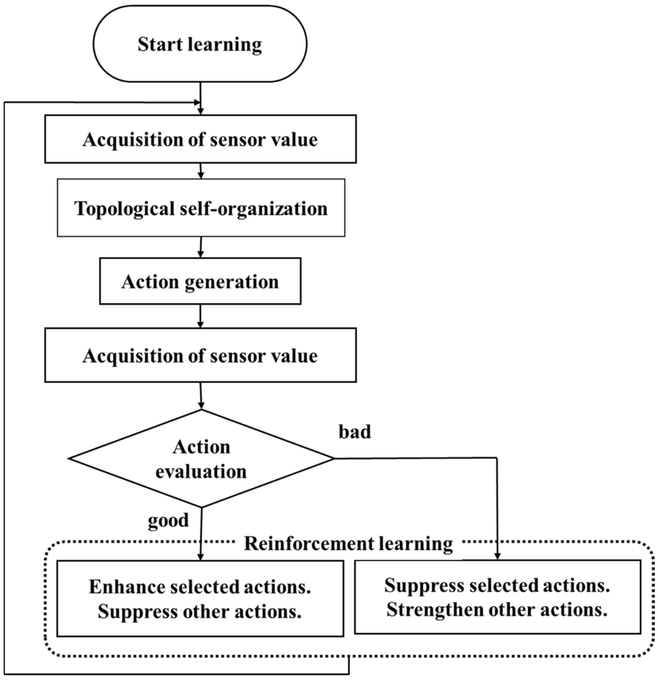



Fig.2. Learning process flow

Next, the potential of the $i$th hidden neuron, $I_i^h(t)$, is calculated in Eq. (4), and the output of that neuron, $O_i^h(t)$, is calculated in Eq. (5). Here, it is obvious that as the hidden layer preserves the topological structure of the high dimensional sensory input, the output of the hidden layer also encodes this structure, in that neurons that are in the proximity of the winning neurons are activated while other neurons are inhibited. Consequently, an input activates a cluster of hidden neurons located close to each other. This topological activation characteristic ensures selective learning, as dropout often utilized in deep neural network and known to prevent overfitting.

$$I_i^h(t) = |X(t) - W_i(t)|^2 \quad (4)$$

$$O_i^h(t) = exp\left(-I_i^h(t)\right)\sigma(win, i, t) \quad (5)$$

Next, the output $O_k(t)$ of the $k$ th output neuron is calculated as in Eq. (6). Where, $v_{ik}$ is the connection weight between the $i$th hidden neuron and the kth output neuron, and $\theta_k$ is the bias of the $k$th neuron. The function $f$ in equation (7) is sigmoid function defined in Eq. (8).

$$O_k(\mathrm{t}) = f\left(\sum_i v_{ik}(t) O_i^h(t) - \theta_k(t)\right) \tag{6}$$

$$f(x) = \frac{1}{1+e^{-x}} \tag{7}$$

Once the values of the output neurons are calculated, the neural network selects a winning neuron, $w(t)$ as in Eq. 8, and executes an action, $a(t)$, associated with the winning neuron as in Eq. (9).

$$w(t) = \max_j O_j(t) \tag{8}$$

$$a(t) = act\big(w(t)\big) \tag{9}$$

In Eq. (9), $act\big(w(t)\big)$ denotes a function to associate the winning neuron with its physical action.

After the action is executed, the action is then evaluated according to the objective of the robot's task and allowing reinforcement learning to be executed. In this study, the purpose of the robot is to learn a strategy for obstacle avoidance in a real world environment.

The evaluation of the action performed by the robot is denoted by $U\big(a(t)\big)$, which varies depending on the environment and the robot's task. This detail is described in Section 4.

When $U\big(a(t)\big) > 0$, it means that the executed action has moved the robot away from the obstacle, and that action is considered a good action. When $U\big(a(t)\big) < 0$, it means that the robot has moved toward the obstacle, and thus its action is considered bad. Action selection in Eq. (8) and (9) is based on competition among all possible actions. Hence, for reinforcement learning, the good action should be reinforced, while the other unselected actions should be inhibited, so that similar input in the future will trigger the same action. Contrastly, if the selected action is bad, that action should be inhibited while any other actions should be reinforced. For this neural network, we transferred this reinforcement learning to the supervised mechanism by defining the cost function as follows.

$$E(t) = \frac{1}{2}\sum_k \big(O_k(t) - T_k(t)\big)^2 \tag{10}$$

Here, the components of the teacher signal are set as follows using the evaluation function $U\big(a(t)\big)$.

$$\begin{aligned} &\text{when } U\big(a(t)\big) > 0 \\ &\qquad T_w(t) = 1 \\ &\qquad T_j(t) = 0(\forall j \neq w) \\ &\text{when } U\big(a(t)\big) < 0 \\ &\qquad T_w(t) = 0 \\ &\qquad T_j(t) = 1(\ \forall j \neq w) \end{aligned} \tag{11}$$

The learning mechanism is a type of reinforcement learning because the ideal action for the robot cannot be explicitly specified, especially in the case when the selected action is bad.

The connection weight $v_{ik}(t)$ of the $i$th neuron in the hidden layer of the neural network and the $k$th neuron of the output layer, and the threshold $\theta_k$ of the $k$th output neuron can then be modified by calculating the cost function's gradient as in Backpropagation [16] as shown in Eq. (12) and Eq.(13), in which $\eta_1$ is the learning rate.

$$v_{ik}(t+1) = v_{ik}(t) - \eta_1 \frac{\partial E(t)}{\partial v_{ik}(t)} \tag{12}$$

$$\theta_k(t+1) = \theta_k(t) - \eta_1 \frac{\partial E(t)}{\partial \theta_k(t)} \tag{13}$$

## III. Robot's Configuration

In this study, we built a learning robot for our experiments and visualization analysis.

The dimensions of the robot are 25cm in length and 15cm in width as shown in Fig. 3. The robot is equipped with sensors and motors. There are four proximity sensors for observing the surroundings in four directions: front, rear, left and right. The sensors were installed at a height of 8 cm.

It has four wheels and four motors for motion generation. Therefore, each wheel can be moved independently, and the robot can move forward, backward, turn right, and turn left. The specifications of the robot are shown in Table □.

The devices shown in Table □. were used. Table □ shows the specifications of the sensor (Fig. 4). The measurement range is 2 to 400 cm, and a measurement error will occur if it is out of the measurement range.

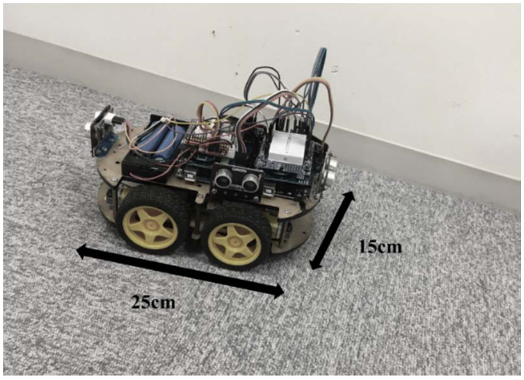



Fig.3. Robot appearance

TABLE I. ROBOT SPECIFICATIONS

| | |
|---|---|
| gross weight [g] | 1140 |
| total length [cm] | 25 |
| width [cm] | 15 |
| sensor height [cm] | 8 |

TABLE II. EQUIPMENT SPECIFICATIONS

| ***machine*** | ***type*** |
|---|---|
| Proximity Sensor | Four ultrasonic sensor (HC-SR04) |
| Motor | Four DC motor |
| Microcomputer (For motor and sensor) | Arduino Uno |
| Power | lithium battery |
| Communication equipment | XBee |

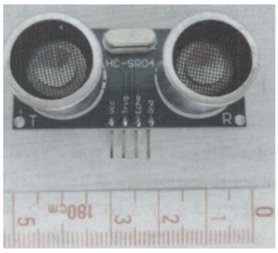


Fig.4. Ultrasonic sensor

TABLE III. ULTRASONIC SENSOR SPECIFICATIONS

| | |
|---|---|
| Measurement range [cm] | 2-400 |
| Operatingfrequency [kHz] | 40 |
| Size[cm] | 45×20×15 |

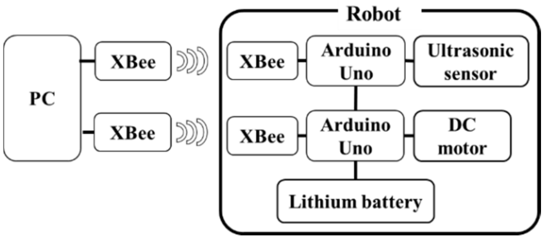



Fig.5. System configuration

Two microcomputers are used, one for distance measurement and one for motor movement control.

Figure 5 shows the system configuration. Here, the learning process was performed on the PC, taking the sensor values sent by microcomputer 1 as input. Receiving the input, the neural network computes the output(action) that should be executed by the robot, and sends the information to microcomputer 2. After the action execution, the distances to the obstacles are re-measured for generating the action evaluation and then the cost function for weight modifications.

## IV. EXPERIMENTS

First, an experiment was performed on the one-dimensional environment as shown in Fig. 6, in which the possible actions of the robot are "move forward" or "move backward". In this case, the robot has to acquire a strategy to avoid the walls on both ends, and position itself around the middle point between the walls.

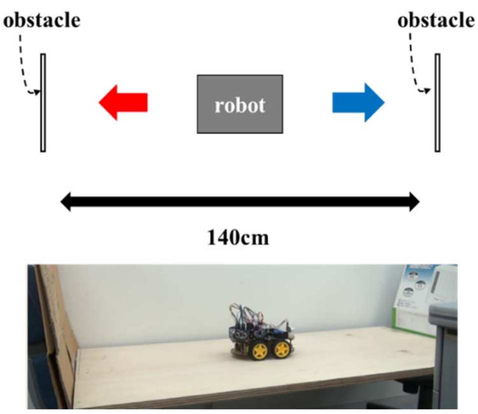



Fig.6. Experiment (□) environment

In experiment (□), the distance between obstacles is set to 140cm, and no other obstacles in between. Here, although the robot is equipped with four sensors, only two of them, in front and back, were utilized. In this experiment, data obtained by performing 30 random walks to initialize the NN, and then real world sequential learning (Fig. 2) was performed for 80 steps.

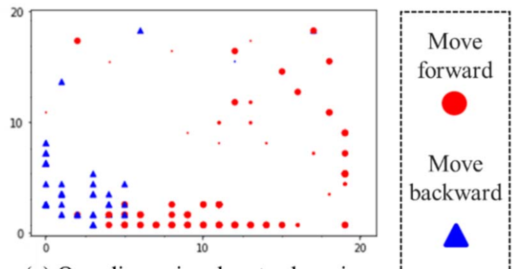
The evaluation of the action performed by the robot is represented by the difference between the distance from the robot to the obstacle before and after the executed action. In this experiment, $U\big(a(t)\big)$ used for learning evaluation (equation (11)) is calculated by the following equation (14), where $d(t)$ is the smaller sensor value of the two sensor values when the time $t$, and after an action, the values recorded as the smaller sensor value are $d(t+1)$.

$$U\big(a(t)\big) = d(t+1) - d(t) \tag{14}$$

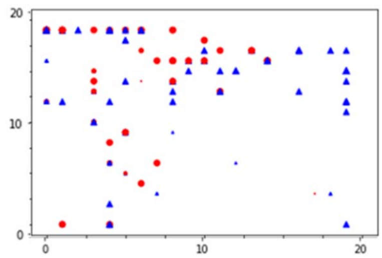
Figure 7 shows the internal topographical representation for the robot trained on experiment (□). Figure 7(a) shows the representation for the successful learning, in which the robot was able to acquire a good strategy, to position itself between the wall, while Fig. 7 (b) shows the internal representation of a rare case when the robot was not able to learn well, and random walking near one of the wall, due to the unresolved initial states of the reference vectors and connection weights.

It is clear that the topographical internal representation in the successful case is well organized, in which clusters of neurons responsible for one action is distinctively separated with the clusters of neurons for other action. The border between the two possible actions is also obvious. The border represents physical environments around the middle point between the two walls. Here, it is important to notice that there is a strong correlation between the visual appearances of the internal topographical organization, the behavior of the robot and its learning environment.

Next, experiment (□) and experiment (□) were performed on two-dimensional environment and motion (Fig. 8).




(a) One-dimensional motor learning : success




(b) One-dimensional motor learning : failure

Fig.7. Topographical Representation for Experiment (□)

In experiment (□) and (□), the robot was surrounded by four walls, without any other obstacles. The four proximity sensors (front, rear, left and right) were utilized and the possible outputs are four actions (“move forward”, “move backward”, “ turn right”, “ turn left”). The distance between obstacles is set to 150 cm, and no other obstacles in between.

In these experiments, the robot has to acquire a strategy to avoid the walls in four directions, and position its self around the middle point between the walls.

In the experiment performed on the two-dimensional environment, the evaluation equation $U\big(a(t)\big)$ was different from the one in experiment (□).

When time step $t$, $d_1(t)$ and $d_2(t)$ are respectively the smallest value and second smallest value in the four sensor values. After the executed action, the values are respectively re-acquired and denoted as $d_1(t+1)$ and $d_2(t+1)$. These values are used for learning evaluation. In two-dimensional experiment, $U\big(a(t)\big)$ used in equation (11) is calculated by the following equation (15).

$$U\big(a(t)\big) = \sqrt{\sum_{i=1}^{2}(d_i(t+1))^2} - \sqrt{\sum_{i=1}^{2}(d_i(t))^2} \quad (15)$$

In experiment (□), the sensory values after 60 random walks were given to initialize the NN in advance, and real world sequential learning (Fig. 2) was performed for 20 steps. Here, the four proximity sensors in the front, rear, left and right of the robot were utilized as input to generate four possible actions, “move forward”, “move backward”, “turn left”, “turn right”.

The resulting internal topographical representations from this experiment are shown in Fig. 9. The move forward action is indicated by ○, the move backward action is indicated by △, the right-turn action is indicated by ☆, and the left-turn action is indicated by □.

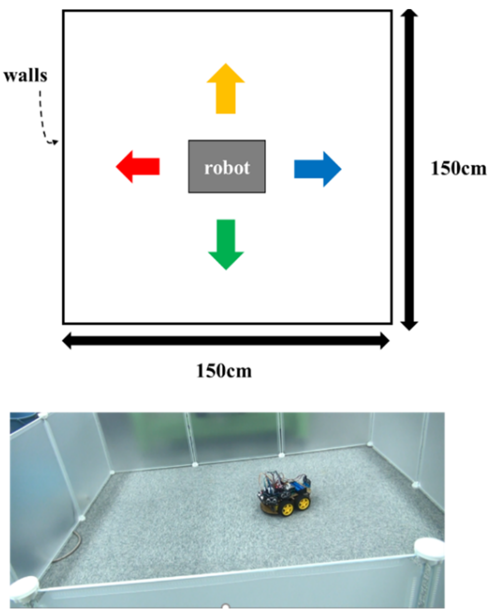


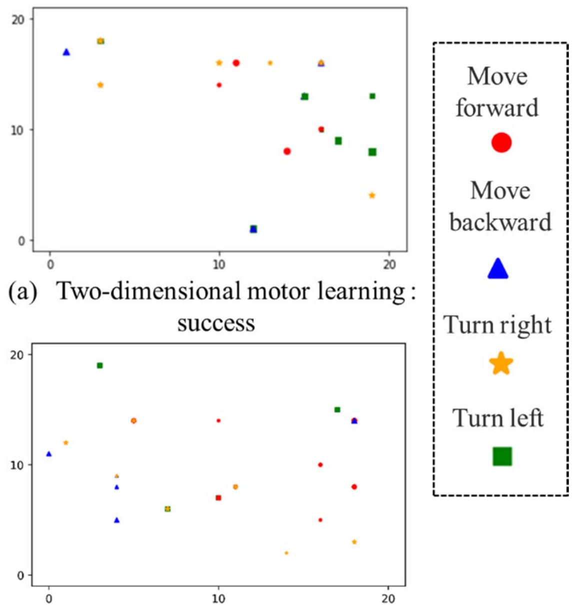

Fig.8. Experiment (Ⅱ)・(Ⅲ) environment

Figure 9(a) shows the topographical representations where the robot succeeded in generating a good strategy for its given task. Here, it can be observed that different movements are distinctly represented. Some movements are represented in multiple nodes due to the execution of those movements in different part of the environment.

We can also learn that the right-turn action was widely distributed, indicating that it was close to both moving forward action and backward action. It is an indication that they are some similar sensory inputs that triggered those different actions. The distribution of left-turn action appeared near the moving forward action.

In the case of successful learning, we can also learn that the robot distinctively distinguishes between moving-forward and moving-backward actions. This means the sensory inputs that triggered those actions are significantly different, so that the robot is not confused in deciding those two actions. Likewise, the distribution of turning right actions and turning left actions are distinctively split. This means that the robot recognized the actions of left-turn and right-turn as different actions.

Here, the right-turn was closer to the move forward action and move backward action, and the left-turn action was closer to the move forward action. This is because the robot recognized that right-turn actions are similar to moving forward and backward actions, and left-turn actions are near moving forward action. This is due to the way the robot arrives at the intermediate point between walls.




(a) Two-dimensional motor learning : success

(b) Two-dimensional motor learning : failure

Fig.9. Topographical Representation for Experiment (Ⅱ) in 20 steps

In this experiment, learning was started from a point close to one wall, to expose the robot to other walls from the early stage of its learning process. At first, the robot was placed so that the values of the front and left sensors were minimized. At this time, the robot recognized that the right-turn action and the moving backward action were triggered by similar sensory inputs. The robot then, rotated in the same position and recognized that the right-turn action and the moving forward action were also triggered by similar sensory inputs. After that the robot found that the left-turn was similar to the action of moving forward while moving to the middle of the walls.

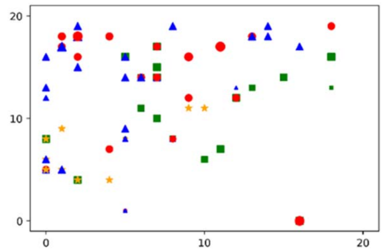
In the case of failed learning (Fig. 9 (b)), it can be seen that the four types of outputs moving forward, moving backward, right-turn and left-turn are overlapped. It indicates that the robot could not distinguish the action to be taken.

Furthermore, using the same environment, increasing the number of steps of learning of two-dimensional movement by 10 times will greatly change the results. The results are shown below when sequential learning is performed 200 steps.

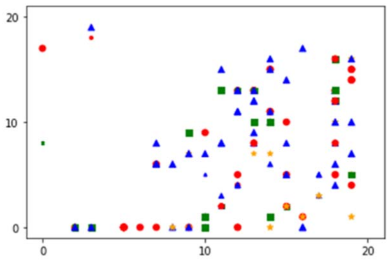
Two different cases of learning in the same experiments were shown in Fig. 10. The difference is due to the learning and position initialization of the robot in the environment.

The internal representations of learning case 1 (Fig. 10 (a) and learning case 2 (Fig. 10(b)) look different because the output that changes depends on the learning process at that time. However, the topographical relation for these two internal representations is similar. They also indicate that the robot can distinctively execute different actions given different sensory inputs. We also learned that the action at the start of learning is easy to distinguish from other actions, and this is due to the higher convergence for the topographical self-organization than the subsequent reinforcement learning. In experiment (□), we added new obstacles in the environment as shown in Fig. 11.

Figure 12 shows the results of experiment (□) performed in the environment shown in Fig. 11. The results in Fig. 12 are significantly different from the results when no obstacle is placed (Fig. 10). This indicates that the environment changes the internal representation of the robot. The robot formed different clusters of a set of actions depending on its location in the environment.

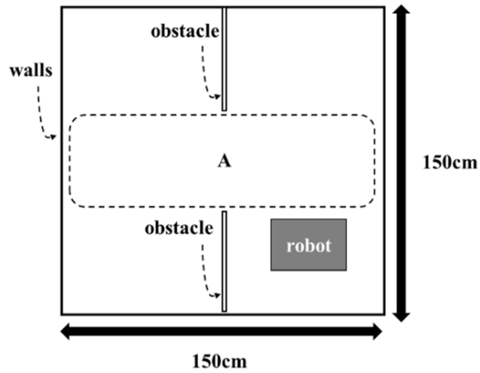


(a) Two-dimensional motor learning
Learning Case 1



(b) Two-dimensional motor learning
Learning Case 2

Fig.10. Topographical Representation for Experiment (□)
in 200 steps



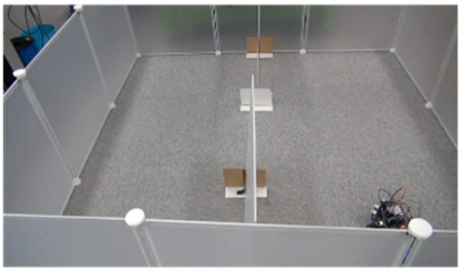
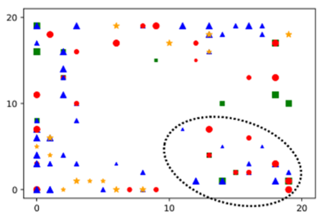



Fig.11. Experiment (□) environment

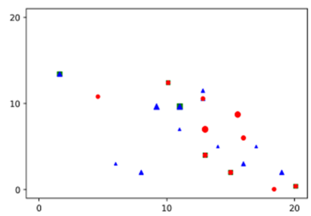
We can learn that the robotic actions formed a cluster when it is located at position A. Similarly in the case 2 and 3 shown in Figures (c) and (d), the learned behaviors in the area A formed a cluster.

We can learn that the robotic actions formed a cluster when it is located at position A. Similarly in the case 2 and 3 shown in Figures (c) and (d), the learned behaviors in the area A formed a cluster.

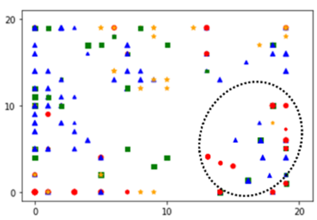
The fact that the cluster is divided at the position of A and other positions means that the robot recognized its position in the environment. From this result, it can be understood that the recognition of the position is more important than the direction of the obstacle for the robot.



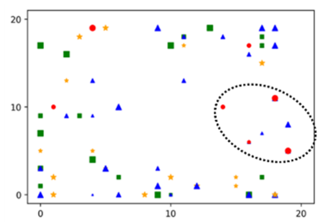
(a)Two-dimensional motor learning
Learning Case 1



(b)(a) Partial data visualization



(c) Two-dimensional motor learning
Learning Case 2



(d) Two-dimensional motor learning
Learning Case 3

Fig12. Topographical Representation for Experiment (□)

## V. Conclusion and Future Works

In this preliminary study, we proposed a hierarchical neural network with a topographical hidden representation that can be utilized to train a robot in real world environment. The low dimensionality of the internal representation allows us to visualize the representation and to promote intuitive understanding on the input-output characteristics of the robot in relevance to their learning environment.

While the proposed method doesn't produce logical rules that explain how the neural network works and require human observation and interpretation, it often leads to better explainability. In the experiments, we show we can to some extent understand why and when the robot will generate good actions, and why it fails. Although still in the preliminary stage, we believe that our proposed method has the potential to give better accountabilities for machine learnings, allowing them to be utilized in more critical applications.

For the future works, we plan to execute more experiments using robots with various morphologies in various environments against various tasks. We also plan to implement a topographical representation that better include the context of the learning as proposed in [19], that can potentially increase the relevance between the representation and the strategy of the robot.

## References


[1] W.Samek, T.Wiegand, and K.Muller, "Explainable artificial intelligence: understanding, visualizing and interpreting deep learning models," CoRR, vol. abs/1708.08296, 2017.

[2] Amina Adadi and Mohammed Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)", IEEE Acess, vol.6, pp. 52138 – 52160, 2018.

[3] B. Letham, et al., "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model, Ann. Appl. Statist., Vol. 9, No. 3, pp. 1350-1371, 2015.

[4] R. Andrews, et al., "Survey and critique of technique for extracting rules from trained artificial neural networks", Knowl.-Based Syst., Vol.8, No. 6, pp. 373-389, 1995.

[5] T. Gopikrishna, "Evaluation of rule extraction algorithms", Int. J. Data Mining Knowl. Manage. Process, Vol. 4, No. 3, pp. 9-19, 2014.

[6] G. Hinton, R. Salakhutdinov, "Reducing the dimensionality of data with neural network", Science, Vol. 28, No. 5786, pp. 504-507, 2006.

[7] G. E. Hinton and Sam T. Roweis, "Stochastic neighbor embedding", in Advances in Neural Information Processing Systems, vol. 15. Cambridge, MA, USA: MIT Press, 2002, pp. 833–840.

[8] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE", J.Mach. Learn. Res., vol. 9, no. 85, pp. 2579–2605, 2008.

[9] T. Kohonen, "Self-organized formation of topologically correct feature maps," Biol. Cybern., vol. 43, no. 1, pp. 59–69, 1982.

[10] T. Kohonen, "Essentials of the self-organizing map," Neural Netw., vol. 37, pp. 52–65, Jan. 2013.

[11] Pitoyo Hartono and Thomas Trappenberg, "Internal topographical structure in training autonomous robot", IEEE International Conference on Systems, Man, and Cybernetics, pp.239-243, 2011.

[12] P. Hartono, and S. Kakita, Fast reinforcement learning for simple physical robots, Memetic Computing, Vol. 1, No.4, pp. 305-313, 2009.

[13] Yuji Shimamoto, Hartono Pitoyo, Takanobu Miwa, Hideyuki Sawada, Shuji Hashimoto, "Visualization of learning state of mobile robot using neural network", J-STAGE, Abstracts of Robotics and Mechatronics Conference, Session ID 1A1-D16, p .1A1-D16-, 2018.

[14] N. Srivastana, et al., Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research, Vol. 15, pp. 1929-1958, 2014.

[15] K.Pearson, "On lines and planes of closest fit to systems of points in space", Philosophical Magazine, Series 6, 2(11), pp. 559–572, 1901.

[16] Daniel L. Marino, Chathurika S. Wickramasinghe, Milos Manic, "An adversarial approach for explainable AI in intrusion detection systems.", IEEE Industrial Electronics Society, pp.3237 - 3243, 2018.

[17] T. Trappenberg, P. Hartono, and D. Rasmusson, "Top-down control of learning in biological self-organizing maps," in Proc. 7th Int. Workshop Self-Org. Maps (WSOM), vol. 5629. 2009, pp. 316–324.

[18] T. Trappenberg, Fundamentals of Computational Neuroscience, 2nd ed. London, U.K.: Oxford Univ. Press, 2010.

[19] Pitoyo Hartono, Paul Hollensen, and Thomas Trappenberg, "Learning-regulated context relevant topographical map.", IEEE Transactions on Neural networks and learning systems, vol. 26, no. 10, pp. 2331-2332, 2015.

[20] Georgios Pierris and Torbjørn S. Dahl, Member, "Learning robot control using a hierarchical SOM-based encoding", IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, VOL. 9, NO. 1, pp.30-43, 2017.

[21] Sutton, R.S. and Barto, A.G., "Reinforcement learning: An Introduction", MIT Press, Cambridge, MA , 1998.

[22] Rumelhart, David E. Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533.