Reconstruction of Visual Sensory Space on the Hidden Layer in Layered Neural Networks

Katsunari Shibata and Koji Ito
Email:shibata@ito.dis.titech.ac.jp

Dept. of Computational Intelligence and Systems Science,
Interdisciplinary Graduate School of Science and Engineering, Tokyo Inst. of Technology,
4259 Nagatsuta, Midori-ku, Yokohama 226-8502, JAPAN

ABSTRACT

In layered neural networks, the input space is reconstructed on the hidden layer through the connection weights from the input layer to the hidden layer and the output function of each hidden neuron. The connection weights are modified by learning and realize the transformation to emphasize necessary information and to degenerate unnecessary one for calculating the output. In this paper, visual sensory signals are adopted as the input. In order to examine the reconstruction, (1)supervised or reinforcement learning is applied to a layered neural network at first, (2)all the connection weights from the hidden layer to the output layer are reset to 0, (3)another supervised learning using some training data is applied, and finally (4)the output for the test data is compared to that when the first learning was not applied. It is shown that the necessary information to generate the desired output in the first learning was extracted on the hidden layer.

KEYWORDS: Layered Neural Networks, Space Reconstruction, Generalization, Visual Sensor, Direct-Vision-Based Reinforcement Learning

1. Introduction

The autonomous learning ability of reinforcement learning(RL) has been focused in these days[1]. In RL, appropriate motions are obtained by learning through trial and errors. However, when a complicated task is given to a robot, it may require huge trials. When we look back at our living creatures, three methods to solve this problem can be thought of. One of them is that much information on genes is inherited from our parents such that a horse just after birth can walk. The second one is to utilize for the present learning what we learned in the past. For example, the spatial recognition ability is necessary in many tasks we do. We do not learn it from the beginning in each task, but utilize the information we obtained through the past tasks. In other words, the ability can be obtained through the tasks that need it. Finally the third one is generalization ability of the neural networks based on the smooth output function of each neuron.

In this paper, from the second viewpoint, the reconstruction of input signals on the hidden layer after learning is examined. As a typical example, the visual sensory space is adopted as input space. In order to observe the reconstructed space on the hidden layer, (1)all the hidden-output connection weights are reset to 0.0 after supervised or reinforcement learning, (2)another supervised learning using some training data sets is applied, and (3)the output for the test data is compared with that when no learning was applied beforehand.

A part of this research was supported by "The Japan Society for the Promotion of Science" as "Biologically Inspired Adaptive Systems" (JSPS-RFTF96I00105) in "Research for the Future Program".

2. Space Reconstruction on the Hidden Layer

The role of hidden neurons is reconstruction of the input space by the linear transformation through the weight matrix from the input layer to the hidden layer, and by the non-linear transformation through the output function that is often sigmoid function. By applying the supervised learning based on BP (Back Propagation) learning[2], the space reconstruction is processed so that the necessary information is emphasized and unnecessary information is degenerated. When we think the spatial recognition, the location and size of the projected object have to be extracted from many visual sensory signals, each of which reflects the information only in a local receptive field. For example, a visual sensor assumes to catch an object as shown in Fig. 1, and only the object location assumes to be varied between the left and right edge of the visual sensor. In this case, the sensor has 10 sensory cells, and then the dimension of the sensory space is 10, even if the degree of freedom is only 1 (x). Accordingly, x is emphasized in hidden neurons' space through some learning in which the output depends on x. If the hidden layer is used in another learning in which x is also necessary to calculate the output, information about x does not need to be learned again from scratch. The generalization ability which mentioned in the first section, is rather effective on the hidden neurons' space than on the input space. This means that even if input patterns are not close, if the hidden states become close by the transformation from the input layer to the hidden layer, the outputs become close. In the case of Fig. 1, the distance between input data set (a) and (b) is the same as that between (a) and (c). However, if x assumed to be extracted on the hidden layer, the distance between (a) and (b) is smaller than that between (a) and (c). If the hidden neurons are used in another learning, generalization on the hidden neurons' space is expected to accelerate the learning.

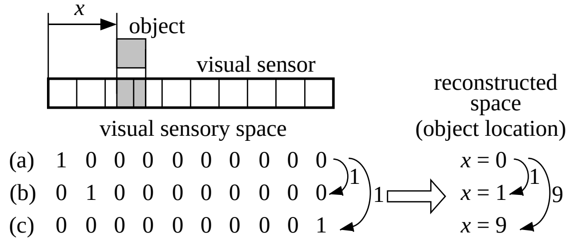

Figure 1: Example of reconstruction of visual sensory space

3. Simulation of Space Reconstruction by Supervised Learning

In this section, it is examined how the input space is reconstructed on the hidden layer by supervised learning. At first, a visual sensor, which has 10 sensory cells arranged in a row that is same as that in the previous section, was prepared as shown in Fig. 2. Then an object, whose size was just same as one sensor cell, was presented. Each sensory cell gave the output as the area ratio occupied by the projected object. The output values were continuous from 0 to 1, and were put into a neural network directly. The output function of each neuron was sigmoid and its value range was from 0 to 1. The training signal was 0.1 at $x = 0.0$ (left edge), 0.91 at $x = 9.0$ (right edge), and proportional to the object location x. x was chosen randomly at every time. After the learning, all the hidden-output connection weights were reset to 0, and the second learning was applied to the same network. In the second learning, the same object was presented only at the edge of the visual field ($x = 0.0, 9.0$) and the network was trained by the same training signals as the first learning. Then interpolation ability was observed. The number of hidden neurons was 10 and the network was trained for 1000 presentations with a small learning rate in each learning.

After both learnings, the average of the difference between the output values and the training signals given in the first learning for the 19 object locations at intervals of 0.5 was calculated. Then the average, maximum and minimum value when the initial connection weights are varied, are shown in Table 1. The first column shows the result when the first learning was not applied. The second shows that after the both learnings. The third shows that when the input-hidden connection weights were fixed in the second learning. The fourth shows that when the second learning was not applied. Though the result depends deeply on the initial condition, it can be said that the difference after the second learning was reduced by applying the first learning. The difference was reduced more by fixing the input-hidden weights.

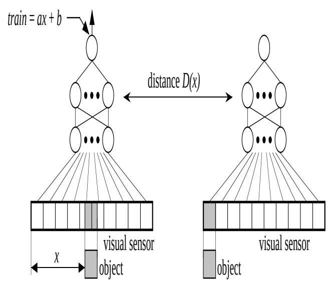

Figure 2: Simulation when the visual sensory cells are arranged in a row

The typical output pattern is shown in Fig. 3. x axis shows the object location and y axis shows the output. The thick straight line shows the training signals. From this figure, it is known that the outputs become close to the training signals by the second learning in which the object was presented only at the edge of the visual sensor when using the hidden neurons trained by the first learning. The outputs for the object locations at which the object was not presented in the second learning, are slightly closer to 0.5 than the training signals. This is because the locality and the generalization of

Table 1 Difference from the expected value in the case of interpolation

$\times 10^{-2}$	no 1st learning	both learnings	fixed i→h weights	no 2nd learning
average	4.327	0.328	0.168	0.003
max	8.602	1.035	0.858	0.005
min	0.880	0.076	0.022	0.001

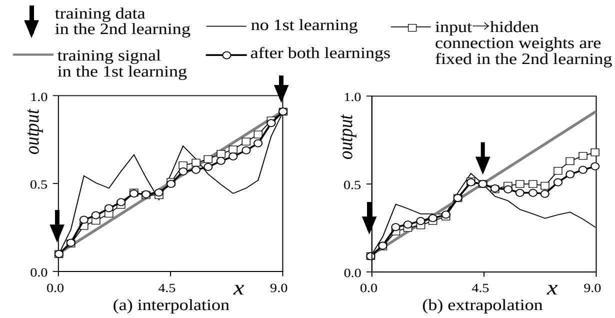

Figure 3: Typical output pattern in the case of interpolation

Table 2 Difference from the expected value in the case of extrapolation

$\times 10^{-2}$	no 1st learning	both learnings	fixed i→h weights	no 2nd learning
average	8.587	2.882	1.571	0.003
max	12.778	6.361	5.590	0.005
min	5.324	1.299	0.169	0.001

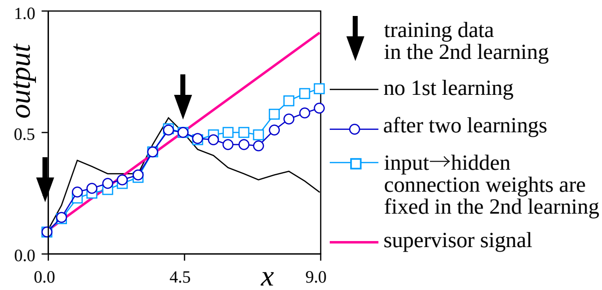

Figure 4: Typical output pattern in the case of extrapolation

the second learning were mixed. If the connection weights are fixed in the second learning, the effect of the locality in the second learning is reduced, and the effect of the first learning becomes larger relatively.

Table 2 shows the error when the object was presented only at $x = 0.0, 4.5$ in the second learning. In this case, extrapolation ability is observed. Fig. 4 shows the typical output distribution. It is known that generalization ability along x axis is not so effective like the interpolation case, but data is generalized even when the test data is not within the two training data sets.

Next, 5×5 visual sensory cells was arranged in a square as shown in Fig. 5. Then an object, whose size was just same as one sensor cell, was presented. The training signal that was proportional to the object location x or y was given to the

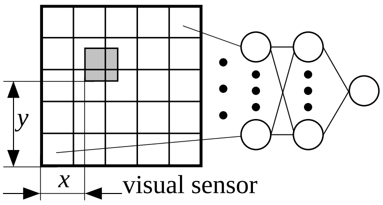

Figure 5: Simulation when the visual sensory cells are arranged in a square

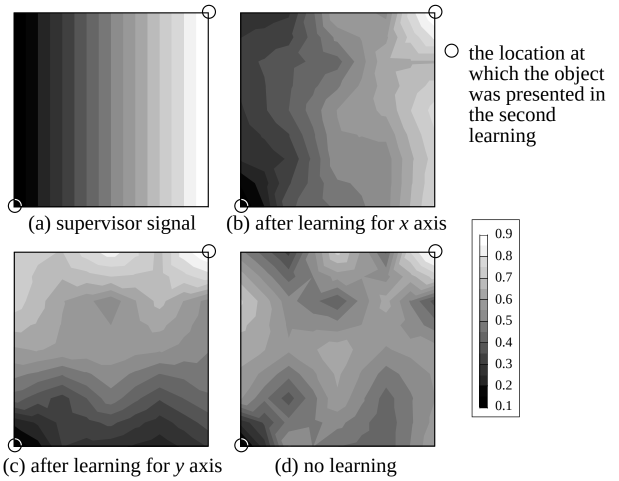

Figure 6: Comparison of the output distribution after both learnings according to the first learning

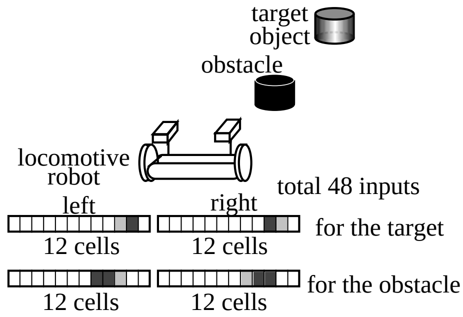

Figure 7: Simulation environment of "going to a target" task with an obstacle

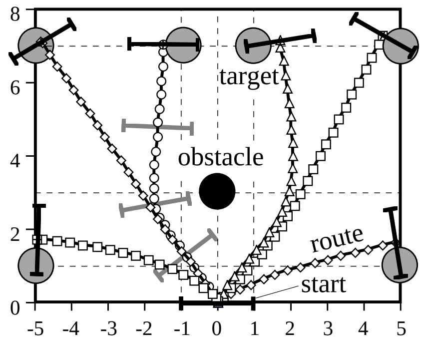

Figure 8: Comparison of the robot's routes after learning between in the task with an obstacle and with no obstacles.

network. After the first learning and resetting the hidden-output connection weights, the object was presented only at the two corner of the sensor $((x, y) = (0, 0), (4, 4))$ and the network was trained by the same training signals as the first learning. The structure of the network and training parameter was the same as the previous learning.

Fig. 6 shows the output distribution after both learnings. Fig. 6(a) shows the training signals that was proportional to x. Fig. 6(b) shows the output when the training signals were given as shown in Fig. 6(a) in the first learning. Fig. 6(c) shows the output when the training signals those were proportional to y were given. Fig. 6(d) shows the output when the first learning was not applied. It can be known that the similar output distribution to the training signal distribution in the first learning is obtained only by giving the two sample input-output pairs in the second learning. It can be said that after the first learning, the representation that is required to make the output close to the training signals is obtained in the hidden layer in some degree.

4. Simulation of Space Reconstruction by Reinforcement Learning (RL)

Here we gave a "going to a target" task to a locomotive robot with two wheels[3]. The robot has to reach the target while avoiding an obstacle as shown in Fig. 7. It has total of 4 visual sensors. Two of them catch only the target object and the others catch only the obstacle. It was assumed that they can catch the target or obstacle even if it hides behind the other one. One sensor for the target and one for the obstacle were attached on the left wheel and the other pair of the sensors were attached on the right wheel. Each visual sensor has 12 visual sensory cells which coveres 180 degree of visual field without overlapping. The total of 48 visual signals are given to the input layer of the neural network. The robot cannot go through the obstacle. This means that the robot stopes its motions when it collids with the obstacle even if the motion signals are not 0. But it has no penalty for the collisions. The diameters of the target and the obstacle are both 1.0 and the length of the robot is 2.0. The obstacle location is chosen randomly in the range of $-5 \leq x \leq 5, 0 \leq y \leq 7$ that is the same as the target location range at every trial. However it is not located at the area where the distance from the target is smaller than 2.0. In the early phase of the learning, the object is located near the robot, and the location is spreaded gradually. Furthermore, the obstacle is not located until the initial target location is spreaded to the final range. The neural network has two hidden layers. The lower one has 30 neurons and the upper one has 20 neurons. Here the range of the output is from -0.5 to 0.5. The number of output neurons is 3, and the first is for state evaluation and the other two are for motion signals, each of which corresponds to the motion signal for the right or left wheel. The first output neuron is trained to change its value with a constant slope k from the initial location to the target location during one trial as

$$s_1(t-1) = x_1(t) - k \qquad (1)$$

where $x_1(t)$ is the $1st$ output at time t, $s_1(t-1)$ is the training signal for $x_1(t-1)$, and k is calculated from necessary times to arrive at the target in the past trials. This learning is similar to TD learning[4]. When the robot gets the target, $s_1(t-1) = 0.4$ is given, and when the robot misses it, $s_1(t-1) = -0.4$ is given. The other two outputs are trained to make the next x_i become large. This means that random vector $rnd_i(t)$ is added to the outputs, and the robot generates motions according to the sum. Then the outputs are trained by the training signal as

$$s_i(t-1) = \{x_1(t) - x_1(t-1)\}rnd_i(t-1) + x_i(t-1), \quad (2)$$

where $i = 2, 3$. Details can be seen in [3].

Figure 8 shows the robot's routes after 167000 trials (the best data until 200000) when an obstacle is located in front of the robot's start position. It can be seen that the robot arrives at each target while avoiding the obstacle. The routes seems to be close to the optimal ones.

The reconstruction of the visual sensory space in hidden neurons is examined. Here, the three output neurons were removed after the RL, and one output neuron, whose connection weights from the hidden layer are all 0.0, was added Then the generalization ability about the recognition of the state in which the target hides behind the obstacle, between the different locations of the target and the obstacle, is observed. As shown in Fig. 9, seven locations for the target (i=0,..,6) and seven for the obstacle (j=0,..,6) are prepared respectively. The distance from the robot to each target is 5, and that to each obstacle is 3. Then the training signal was given 0.3 in the case of $i = j$, 0.0 in the case of $|i - j| = 1$, and -0.3 otherwise. $i = j$ means the target exists just behind the obstacle. Then the target location and the obstacle location were chosen randomly except for the case of $i = j = a$ where $a = 0, .., 6$, and the neural network was trained by the training signal. a was fixed during the learning. After the learning, the visual signals when the both target and obstacle locations were a ($i = j = a$), were put into the input layer of the network as a test data set and the output was observed.

Figure 10(a) shows the output as a function of the target location when the obstacle is fixed at the location 3 ($j = 3$) when $a = 3$. The solid line with squares shows the output after both RL and supervised learning. The line with circles shows the output when the RL was not applied. The thin line shows the expected output for $i = 3$ and the training signals for otherwise. Because the supervised learning is applied except for $i = 3$, the output is almost the same as the training signal at $i = 0, 1, 2, 4, 5, 6$. When only one of the target and the obstacle is located at 3, the training signal is not larger than 0.0. Therefore when the both locations are 3, the possibility that the output is smaller than 0.0 is large in general for the generalization in visual sensory space. Actually the output at $i = 3$ in the case of no learning hidden neurons is far smaller than 0.0. However, the output in the case of the trained hidden neurons is larger than 0.0. This means that the new space was constructed on the hidden layer, and the generalization was more effective on the space. Some information about the state that the target hides behind the obstacle, became to be extracted by the RL. Figure 10(b) shows the outputs as a function of a. The pair $(i, j) = (a, a)$ of the target and obstacle location, was not trained and used as the test data set. It can be seen that the outputs are larger than 0.0 when the a is 2, 3 or 4 at the case of trained hidden neurons. The reason why the outputs when a is not 2, 3 or 4, are not so large, can be thought that the generalization ability is not effective for the biased positions.

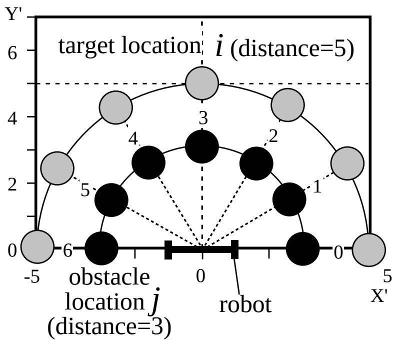

Figure 9: The target and obstacle locations in the simulation to examine the generalization ability of the hidden neurons information.

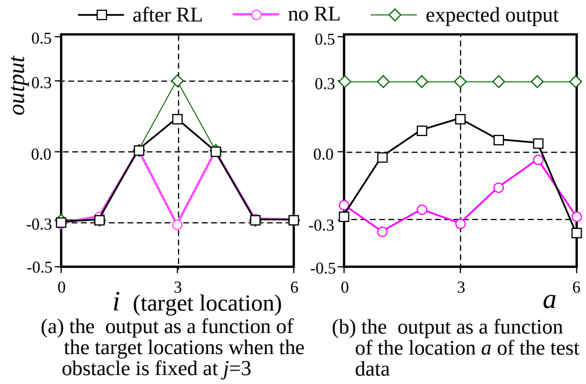

Figure 10: Comparison of the output after supervised learning between the trained hidden neurons and no trained hidden neurons

5. Conclusion

The reconstruction of visual sensory space on the hidden layer after supervised learning or reinforcement learning is observed. It was known that the space which was thought to be useful to realize the desired output was constructed on the hidden layer.

References

[1] Kakazu, Y., et al., *Annual Report of Research on Theory of Adaptive and Reinforcement Learning, "System Theory of Function Emergence" under the Grant-in Aid for Scientific Research on Priority Area supported by the Ministry of Education, Science, Sports and Culture of Japan* (1998).

[2] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., "Learning representation by back-propagating errors", Nature, Vol. 323, No. 9, pp. 533-536 (1986).

[3] Shibata, K., Ito, K. and Okabe, Y., "Direct-Vision-Based Reinforcement Learning in "Going to a Target" Task with an Obstacle and with a Variety of Target Sizes", Proc. of NEURAP '98, pp. 95-102 (1998).

[4] Barto., A. G., Sutton, R. S. and Anderson C. W., "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems", *IEEE Trans. on SMC*, **13**, pp. 835-846 (1983).