Emergence of Communication for Negotiation By a Recurrent Neural Network

Katsunari SHIBATA and Koji ITO

Dept. of Comp. Intell. & Sys. Sci., Interdisciplinary Grad. School of Sci. & Eng., Tokyo Instute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 226-8502, JAPAN
shibata@ito.dis.titech.ac.jp

Abstract

We believe that communication in multi-agent system has two major meanings. One of them is to transmit one agent's observed information to the other. The other meaning is to transmit what an agent is thinking. Here we focus the latter and aim to the emergence of the autonomous and decentralized arbitration through communication among some agents. The communication contents, strategy and representation are not prescribed and are acquired by learning using a reinforcement signal which is given to the agent after its action. The reinforcement signal is not shared with the other agents. In order to realize this learning, the agent often has to make a decision not only from the present communication signals but also from the past signals. Accordingly the system architecture using recurrent type (Elman) neural network is proposed. The ability of this architecture was examined by two and four agents negotiation problems. A variety of negotiation strategies emerged among the agents through the learning to avoid some conflict after their decisions.

1. Introduction

In multi-agent system, communication is very effective for cooperation or arbitration among the agents. However, if we have designed in advance what the agents communicate and how the agents communicate, the agents cannot modify the communication contents, strategy and representation according to the change of the environment, and then they may lose the flexibility. There are also many cases that the most effective communication contents, strategy and representation are not known beforehand. Furthermore the communication contents, strategy and representation among our living creatures do not seem to be given from the outside of us, but are generated autonomously among us by the individual learning through experiences. For these reasons, the emergence of communication is focused recently.

We believe that communication has two major meanings. One of them is to transmit one agent's observed information to the other. That is useful for making up a lack of the receiver's observations. The famous G. M. Werner et al's work[1] belongs to this category. In their work dealing with the mate finding problem, there are some females which cannot move but have an eye, and some blind males which can move. The females can transmit some communication signals to the males. The transformation from the observed information to the communication signals in the females and the transformation from the communication signals to the action in the males are done by the recurrent neural network whose connection weights are given by the value of their genes. When a male finds a female, they can produce a son and a daughter. The offspring is produced by the standard genetic operations of crossover and mutation. Here, the communication from the females to the males is utilized in order that the females inform the nearby blind male as to the information about the relative location between them that the blind male cannot know by itself. The pioneering work by K. Nakano et al.[2] shows the learning method to generate some common words for the objects which exist in the environment. The common word is useful with respect that the agent can know the existence of an object by the communication signals from the other agents. Accordingly it can be said that the communication in this work is also utilized to transmit the observed information to the other.

The other meaning of communication is to represent what an agent is intending. The agents become able to cooperate and also to avoid some conflict through communication. Many works about negotiation are done at this standpoint in the distributed artificial intelligence field[3][4], but in these researches, the communication contents, strategy and representation were prescribed in advance, and were evaluated.

Now we focus on "emergence of communication" and "communication of intention". We propose a learning architecture for the multi-agent which can generate an appropriate transformation from the past communication signals received from the other agents to the present communication signal or action. Then note that the communication contents, strategy and representation are not prescribed. Some reward is given to the agents in the case of the appropriate communication and action, and some penalty is given in the case of conflicts. These reinforcement signals (reward and penalty) are not shared with the others. It is also an important point of this research that the negotiation among some agents can be realized without sharing the reinforcement learning. As an example, a negotiation problem is tries to be solved here. In negotiation problems, an agent represents its intention at first. However, when some conflict in the final decision is predicted, the agents have to change its intention according to the communication signals from the other agents. . Sometimes the past series of communication signals are required to decide the change of its intention. To achieve this function, Elman-type recurrent neural network[5] is employed.

The proposed learning architecture is described in the next section, and two simulations are shown in the following section. The first simulation is “two agents negotiation problem”. It is examined that one-to-one negotiation, that is the simplest case, can be realized by the learning. In this problem, there is only one solution. The basic ability of the system is evaluated by whether the solution is obtained or not in spite of many combinations of communication signals and decision. The second one is “four agent negotiation problem”. This problem is more difficult because it cannot be solved by the relation to only one opponent. Each agent has to make a decision according to the other three agents and their locations. It is difficult to design such agents manually. It is shown whether the agents have the abilities to negotiate with more than two agents autonomously and decentralizedly without sharing reinforcement signals.

2. Learning Architecture

Fig. 1 shows the proposed architecture of two agents. The agent receives the other agent’s previous communication signal and its own previous communication signal as the inputs to its own neural network. The neural network is Elman-type, in which the present outputs of the hidden neurons are used as the inputs at the next time step. The output function of each neuron except the input layer is sigmoid function whose value range is from -0.5 to 0.5. The neural network has two outputs, one of which is the output for its communication and the other is for its action. The actual communication signal and action outputs are stochastically determined from -1 or 1. The probability p that each of them is 1, is the sum of the output of the neural network and 0.5. The agents exchange their communication signals synchronously with each other three times, and then make a final decision of the action. At the beginning of each negotiation, all the inputs including the feedback inputs from the hidden neurons are

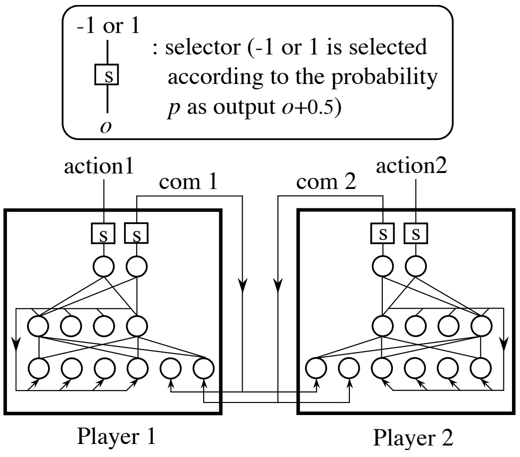

-1 or 1
s
o
: selector (-1 or 1 is selected according to the probability p as output o+0.5)
action1
com 1
com 2
action2
Player 1
Player 2

Figure 1. Proposed architecture of the agent for communication using a recurrent neural network.

set to be 0.0. Accordingly the probability that the first communication signal is 1, does not depend on the other agent’s communication signal.

The agent learns its communication and action by the reinforcement signal which is obtained after the action. When the agent obtained the reward, the probabilities of the action and a series of the communication signals become large, and when it obtained the penalty, the probabilities become small. The neural network is trained by normal BP (Back Propagation) learning algorithm or BPTT (Back Propagation Through Time) learning algorithm[6]. Both are typical supervised learnings for neural networks. In BPTT, the error signal propagates backward through time until the beginning of the negotiation. On the other way, in normal BP, the error signal propagates backward through the network, but does not propagate through time, in other words, does not propagate from the input layer to the hidden layer. The training signal is given to the communication output or action output at each time step as follows,

$$x_{ideal} = x + \eta \cdot r \cdot x' \cdot o$$

where x_{ideal} : training signal, $x=f(u)=1/(1+\exp(-u))-0.5$: output of the neural network, u : internal state of the output neuron, η : learning rate (0.1 is employed here), r : reinforcement signal. Here 1 is employed in the case of reward, and -5 in the case of penalty. What is different from the popular reinforcement learning is that the reinforcement signals are not discounted for the training of the past communication signal. $x'=dx/du=(0.5-x)(0.5+x)$, and o : actual (stochastically determined) communication signal or action that is different from the raw output of the neural network. x' is added to make the output stable when the probability p is close to 0 or 1. The initial connection weights from the input layer to the hidden layer are set to be small random numbers, and those from the hidden layer to the output layer are set to be 0.0. Then the probabilities of all the actual outputs are just 0.5 before the learning.

3. Simulation

3.1. Two Agents Negotiation Problem

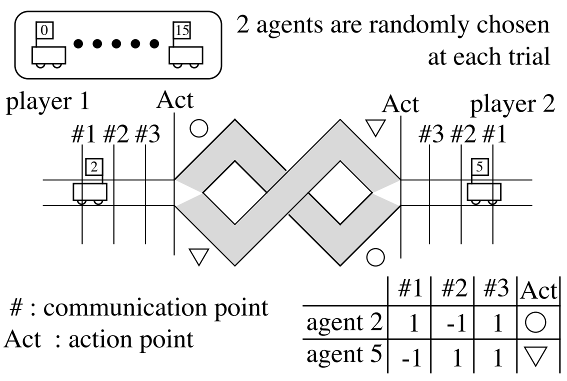
3.1.1. Setting. At first, two agents negotiation problem is described. Figure 2 shows the simulation environment. Two players are randomly chosen from 16 agents, and the player cannot know the opponent directly. Three communication chances, #1, #2, #3, are given to the players. The players decide the communication signal (-1 or 1) at each chance according to the probability decided by the communication output as described in the previous section. At the chance #1, the player has to decide the first communication signal without knowing the other's signal. At the chance #2, it decides the second communication signal depending on its and the other's first signals. At the last chance #3, it can decide the third signal from its and the other's first and second signals. Then note that the information about the first communication signals are supposed to be kept indirectly in the hidden layer as the context information through the hidden-input feedback connections. Finally it decides which route of ○ (action=1) or ▽ (action=-1) the player goes to. If both players select the same route, they receives penalty (*r*=-5), and if they select the different route, they receives reward (*r*=1). Then they learns its three communication signals and action from this reinforcement signal as described in the previous section. In this problem, since there are three communication chances, two of 2^3=8 agents can be distinguished using the communication patterns. Even if all the communication signals are the same between two agents, the agents can make the different actions. Then each of 2^4=16 agents can go through the different route from the other's ideally.

3.1.2. Results. The right hand side of Table 1 shows the negotiation examples after a successful learning, which means that all the pairs of the agents could go without any collisions after learning. In the almost all cases after the successful learning, the probabilities of all the communication signals and action were close to 1.0 or 0.0. The first three examples are explained as follows.

Example 1: The agent 0 and agent 1 were chosen as players. Both agents continued to output the communication signal 1, and finally the agent 0 went to the route ○, and the agent 1 went to the route ▽.

Example 2: The agent 0 and 2 were chosen. The agent 0 made the same sequence of communication signals and action (select the same route) as the example 1. The agent 2 made the communication output 1 twice, changed it to -1 at the third chance, and finally went to the route ▽.

Example 3: The agent 1 and 2 were chosen. The agent 1 continued to output the communication signal 1, and went to the route ○. The action is different from that in example 1. The agent 2 made the same communication signals and action as the example 2.



	#1	#2	#3	Act
agent 2	1	-1	1	○
agent 5	-1	1	1	▽

Figure 2. Simulation environment of two agents negotiation problem

Here, although the meaning of the communication signals had not been given to the agents before the learning, we put the meaning on them after the learning. There were two agents which always went through the same route not depending on the opponent agent like the agent 0 in the above examples. The sequences of such agents' communication signals were always the same not depending on the opponent. These agents are supposed to have persisted on going through the selected route. Accordingly the meaning that the communication signal is 1, is defined as the assertion to go through the route ○ in the above examples. Then the agent 1 in the example 1 is supposed to persist on the route ○, but to change its action to the route ▽ finally.

Now Table 1 shows the negotiation results for all the pairs of players. The order of the agents in this table is sorted by the probability of selecting the route ○. The filled circle ● shows that the agent continued to persist on going along the route ○. The circle ○ with superscript number means that the agent persisted on the route ▽ until the superscript number of communication chance, and changed its intention to the route ○ at the next chance like the agent 12 in the example 7. The circle ○ with subscript number means that the agent persisted on the route ○ at first, it changed its mind to the route ▽ at the subscript number of chance, and finally it returned its intention to the route ○. Such agent can be seen as the agent 4 in the example 5 and the

Table 1. Communication signals and action corresponding to the agent pair after learning when Elman-type neural network is used in the agent.

Agent No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
1	▽^{3}		●	●	●	●	●	●	●	●	●	●	●	●	●	●
2	▽^{2}	▽^{2}		○$_{3}$	●	●	●	●	●	●	●	●	●	●	●	●
3	▽^{2}	▽^{2}	▽^{2}		●	●	●	●	●	●	●	●	●	●	●	●
4	▽^{1}	▽^{1}	▽^{1}	▽^{1}		○$_{2}$	○$_{2}$	○$_{2}$	●	●	●	●	●	●	●	●
5	▽^{1}	▽^{1}	▽^{1}	▽^{1}	▽$_{13}$		○$_{2}$	○$_{2}$	●	●	●	●	●	●	●	●
6	▽^{1}	▽^{1}	▽^{1}	▽^{1}	▽^{1}	▽^{1}		○$_{23}$	●	●	●	●	●	●	●	●
7	▽^{1}	▽^{1}	▽^{1}	▽^{1}	▽^{1}	▽^{1}	▽^{1}		●	●	●	●	●	●	●	●
8	▼	▼	▼	▼	▼	▼	▼	▼		○^{1}	○^{1}	○^{1}	○^{1}	○^{1}	○^{1}	○^{1}
9	▼	▼	▼	▼	▼	▼	▼	▼	▽$_{23}$		○^{1}	○^{1}	○^{1}	○^{1}	○^{1}	○^{1}
10	▼	▼	▼	▼	▼	▼	▼	▼	▽$_{2}$	▽$_{2}$		○$_{13}$	○^{1}	○^{1}	○^{1}	○^{1}
11	▼	▼	▼	▼	▼	▼	▼	▼	▽$_{2}$	▽$_{2}$	▽$_{2}$		○^{1}	○^{1}	○^{1}	○^{1}
12	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼		○^{2}	○^{2}	○^{2}
13	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▽$_{3}$		○^{2}	○^{2}
14	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼		○^{3}
15	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	

example

	agent	#1	#2	#3	Act
(1)	⓪	1	1	1	○
	①	1	1	1	▽
	agent	**#1**	**#2**	**#3**	**Act**
(2)	⓪	1	1	1	○
	②	1	1	-1	▽
	agent	**#1**	**#2**	**#3**	**Act**
(3)	①	1	1	1	○
	②	1	1	-1	▽
	agent	**#1**	**#2**	**#3**	**Act**
(4)	⓪	1	1	1	○
	⑧	-1	-1	-1	▽
	agent	**#1**	**#2**	**#3**	**Act**
(5)	④	1	-1	1	○
	⑤	1	-1	-1	▽
	agent	**#1**	**#2**	**#3**	**Act**
(6)	⑥	1	-1	-1	○
	⑦	1	-1	-1	▽
	agent	**#1**	**#2**	**#3**	**Act**
(7)	⑫	-1	-1	1	○
	⑬	-1	-1	1	▽

agent 6 in the example 6 at the right hand side in Table 1. Each meaning of ▼ and ▽ is as well as the circle case, but the final selection is the route ▽. It is known that each agent was ordered and arbitrated autonomously and decentralizedly. It was found in other simulations that the communication sequence acquired by the learning when the agent persisted on the route ○ was one of (-1, -1, -1), (-1, 1, -1), (1, 1, 1) and (1, -1, 1). The acquired sequence of the four depends on the initial connection weights and the stochastic factor in the choice of agents in the simulation. The reason why the sequence was not (1, 1, -1), (1, -1, -1), (-1, -1, 1) or (-1, 1, 1) is that it is easy to repeat the same communication signal at all times or to change the communication signal at all times. That is because the previous signal is one of the inputs of the neural network. When the number of agents is reduced to 8, the solution could be found more easily. In this case, no agents exist which change its intention more than once like the agent 2, 4, 5, 6, 9, 10, 11 and 13 in Table 1.

Next, the connection weights of the trained neural network with two hidden neurons in the agent 2 in Table 1 is shown in Fig. 3 and the time series of the hidden neurons' outputs for 4 types of opponent agents are shown in Fig. 4. The agents could solve this problem by learning only with two hidden neurons, but the probability of success was very small. The basic strategy of the agent can be interpreted as follows,

1. The communication output is always close to the action output by the similar connection weights from the hidden layer to the output layer.
2. The initial communication output is 1 by the positive bias of the hidden 2 neuron and the positive connection weight of the hidden 2 -> the output 1.
3. If the opponent agent's communication signal is -1, both outputs become 1 by the positive connection weights of the hidden 2 -> the output 1 and the hidden 2 -> the output 2, and negative connection weight of the input 1 -> the hidden 2. This can be observed in Fig. 4 (b), (c) and (d).
4. The agent tries to keep the output value by the negative connection weights of the input 2 -> the hidden 1 and the hidden 1 -> the output 1, and also by keeping the hidden 2 neuron's output based on the positive feedback connection weight of the input 4 -> the hidden 2.
5. If the opponent agent's communication signal is 1, the hidden 2 value is decreased by the negative connection weight of the input 1 -> the hidden 2. If the state, that both of the agent's and the opponent agent's communication signals are 1, continues for two time steps, the communication output becomes -1. That is because the hidden 2 neuron's output becomes less than

0 for the latter reason of the item 4. This can be observed in Fig. 4 (a) and (b).

The reason why the situation of the item 5 can be realized is that the neural network is recurrent and the context information can be stored by the feedback connection weights. When the normal layered neural network was used instead of the Elman network, the agents which changed their intentions at the chance #3 like 2, 3, 12 and 13 in Table 1 did not appear.

100 simulations were done varying the initial connection weights of the neural network of each agent. The number of the hidden neurons is 4. When BPTT was applied, all the combinations of two agents could go through the route without collisions in 41 simulations. In the case of normal BP, all the agents went through the route without collisions in 52 simulations. We had expected that successful number is more in the case of BPTT than in the case of normal BP, but it was the opposite result. The connection weights from the context inputs to the hidden neurons are used for both keeping the necessary context information and reflecting the information to the outputs. The learning for both keeping and reflecting the context information can be done by BPTT, but by normal BP, the learning only for reflecting the context is done. In this case, it might be thought that the learning for reflecting the context information to the outputs is useful to keep the necessary context information.

3.2. Four Agents Negotiation Problem

3.2.1. Setting. Next four agents negotiation problem is presented. The previous two agents negotiation problem can be solved by ordering all the agents and deciding each route based on the order between two players. However, four agents negotiation problem is more difficult. Because, even if an agent has negotiated successfully with another agent, it might conflict to the other agents, and nevertheless the reinforcement signal of one agent is not shared with the others. Furthermore, to solve the problem described in the following, it is not enough only to know what are the chosen agents, but also to know where the other chosen agents are arranged relatively.

Figure 5 shows the environment of the four agents negotiation problem. 4 players are chosen randomly at each trial among 8 agents, and located at the entrance of each route. The players can transmit a communication signal, -1 or 1, to all the players and can receive 4 communication signals three times. The first received signal is its own signal, the second one is transmitted from the opposite side agent, the third one is from the neighbor agent through the route ○, and the last is from the other neighbor agent through the route ▽. But the agent does not know the following three matters, the source of each signal, which are the chosen agents, and the locations

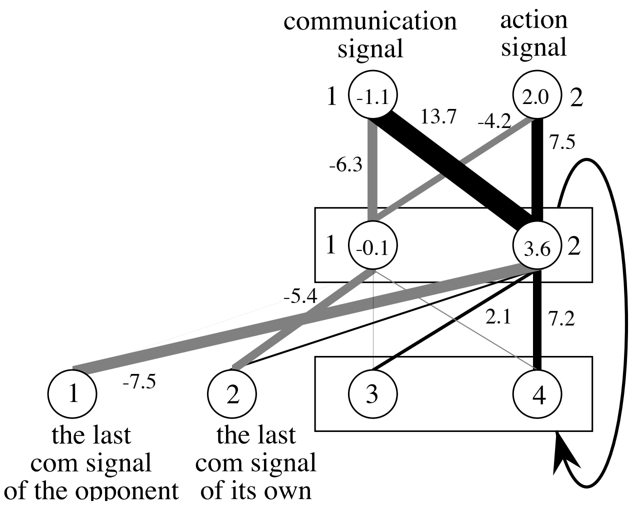


Figure 3. Connection weights of the neural network in the agent 2 after learning.

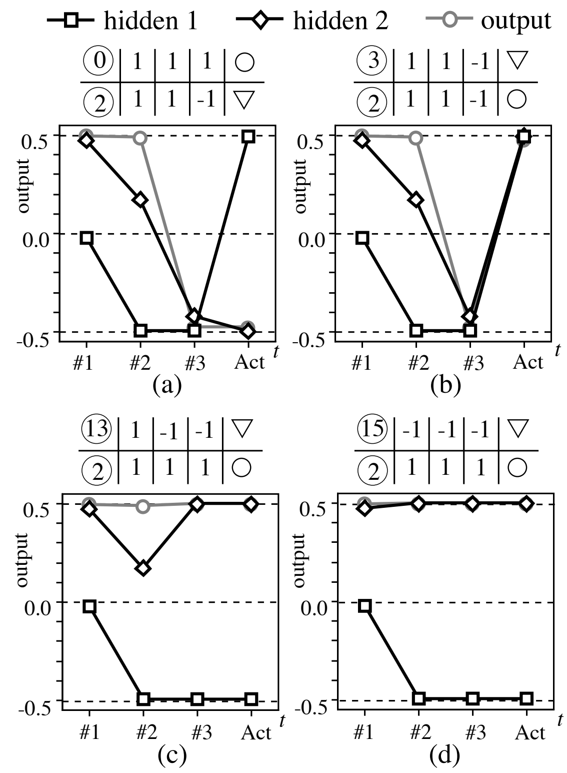


Figure 4. Time series of the hidden neurons' outputs of the agent 2 according to the opponent player.

of the chosen agents. The agents have to decide whether it goes on the route ○ or the route ▽. If the agent can go through one route without collisions with the neighbor, it can obtain the reward *r*=1. If the agent collides, the penalty *r*=-5 is given to the agent. The reward and penalty is not shared with the other agents. The architectures of the agents are almost the same as shown in Fig. 1, but the number of communication inputs are 4 instead of 2. The learning of each agent is same as the previous simulation. The problem is solved when all the agents go clockwise or counterclockwise at each trial through the route arranged in a diamond shape as shown in Fig. 5. In order that all the agents go clockwise or anti-clockwise, the agents have to select the same action (route) as the opposite side agent, and the different action from the neighbors. It is very difficult for us to design the communication strategy of each agent.

3.2.2. Results. The number of agent, 8, is not the maximum for solving this problem. Many kinds of solutions could be found in the simulations on the contrary to the previous two agents negotiation problem. In some simulations, one agent did not change the sequence of its communication signals and action, whatever the other agents' communication signals were. The agent seems to have the initiative, and all the other agents seem to try to know the location of the initiating agent and decide their actions according to the location if the initiating agent is chosen as a player. In the most cases, all the agents changed their communication signals and actions

Though the possibility of the successful learning is very small, this problem could be solved by the network with only two hidden neurons. The details of the solution are described as follows. Table 2 shows the typical negotiation examples in the solution. In Table 2 (a), agents 0, 2, 3, 7 were chosen. The first and second communication signals for 4 agents were the same, but only the agent 7 changed its signal at the communication chance #3. Finally, the agent 0 and 3 took the action 1 (route ○) and the others took the action -1 (route ▽). Table 3 shows the probability of each communication signal and action, and the number of the communication-action patterns of each agent on all the combinations of the agents after learning. The order of the agents is not sorted. Since the values in the table is 2*p*-1 where *p* is the probability that the communication signal or action is 1, the sign shows the agent's preferable direction and the absolute value shows the insistence degree of the agent. It is known that the agents can change its communication signals adaptively and with variety. Especially the agent 0 took seven communication-action patterns among all the eight patterns. Since the communication signal at the chance

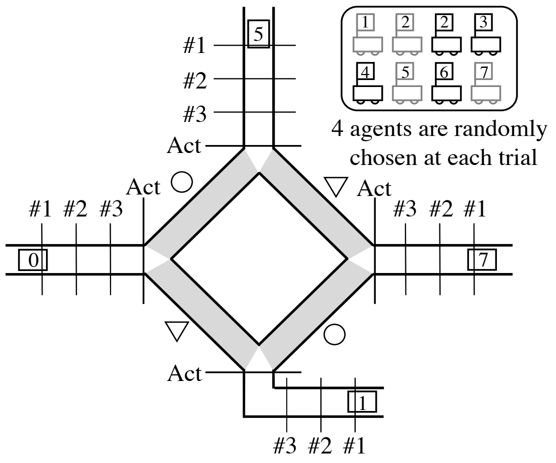


Figure 5. Simulation of the four agents negotiation problem.

#1 is always the same, the number of all the patterns is 8. In all the agents, the signs of the communication signals at three chances are the same, while the sign of the action is always different from that of the communication signals. It can be interpreted that the communication signal 1 denotes to insist on the action -1(▽), and the communication signal -1 denotes to insist on the action 1(○). When we looked at the results for all the combination of the agents, some rules could be found as follows.

(Case 1) When the preferable route of the four chosen agents is ○, e.g. when the chosen agents are 0, 2, 3, and 7 as shown in Table 2 (a), the agent 7 always changed the communication signal from -1 to 1 at the chance #3, and took the action –1. That does not depend on the arrangement of the agents. The opposite side agent did not change its communication signal, but took the action -1. The others did not change their signals and took the action 1 according to its preference.

(Case 2) When the preferable route of three chosen agents is ○ and the other's is ▽ as shown in Table 2 (b), the agent who prefers ▽ and its two neighbors did not change their signals, and only the opposite side agent always changed the signal at the chance #2 and took the action ▽.

(Case 3) When the preferable route of the half of the chosen agents is ○, the situation can be divided into two cases. In one case, where the agents have the same preference as the opposite side agent as shown in Table 2 (c), all the agents did not change the signal and took the preferable actions.

Table 2. Typical communication patterns after successful learning in four agents negotiation problem.

	#1	#2	#3	Act
Agent 0	-1	-1	-1	1 ○
Agent 2	-1	-1	-1	-1 ▽
Agent 3	-1	-1	-1	1 ○
Agent 7	-1	-1	1	-1 ▽

(a)

	#1	#2	#3	Act
Agent 2	-1	-1	-1	1 ○
Agent 7	-1	1	1	-1 ▽
Agent 0	-1	-1	-1	1 ○
Agent 5	1	1	1	-1 ▽

(b)

	#1	#2	#3	Act
Agent 2	-1	-1	-1	1 ○
Agent 5	1	1	1	-1 ▽
Agent 7	-1	-1	-1	1 ○
Agent 4	1	1	1	-1 ▽

(c)

	#1	#2	#3	Act
Agent 2	-1	-1	-1	1 ○
Agent 4	1	1	1	-1 ▽
Agent 5	1	1	1	1 ○
Agent 7	-1	-1	1	-1 ▽

(d)

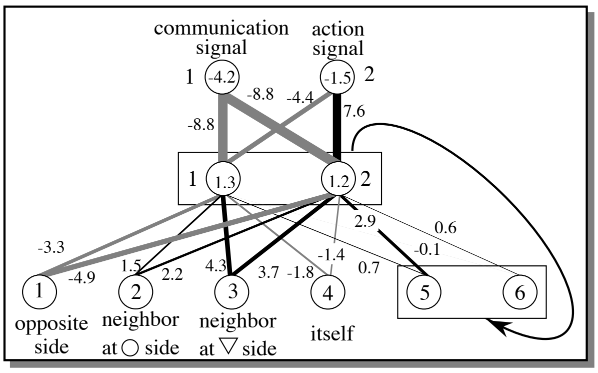
Table 3. The probability of the communication signals and action of each agent after learning and the number of the communication patterns. The values in the table are 2*p*-1 where *p* is the probability that the output is 1.

	com #1	com #2	com #3	Action	num of patterns
Agent 0	-1.0	-0.429	-0.457	0.352	7
Agent 1	1.0	0.029	0.286	-0.257	4
Agent 2	-1.0	-1.000	-1.000	0.581	2
Agent 3	-1.0	-0.086	-0.343	0.181	5
Agent 4	1.0	1.000	1.000	-0.571	2
Agent 5	1.0	0.771	0.533	-0.486	5
Agent 6	1.0	0.429	0.324	-0.286	5
Agent 7	-1.0	-0.771	-0.552	0.486	4

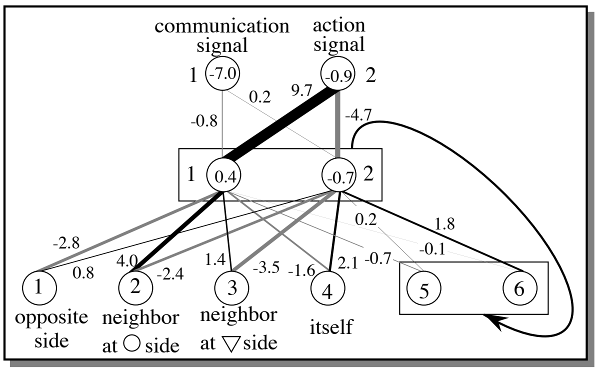
(Case 4) In the other case, where the two agents, who have the same preferable direction, are the neighbors as shown in Table 2 (d), there are many combinations and any simple rules could not be found.

As well as this simulation, it had been expected also in other simulations that the half of 8 agents preferred the route ○ than the route ▽, and the other 4 agents preferred the route ▽. However, it could be found that only two agents preferred the route ○, and the other 6 agents preferred the route ▽ as the most asymmetrical case.

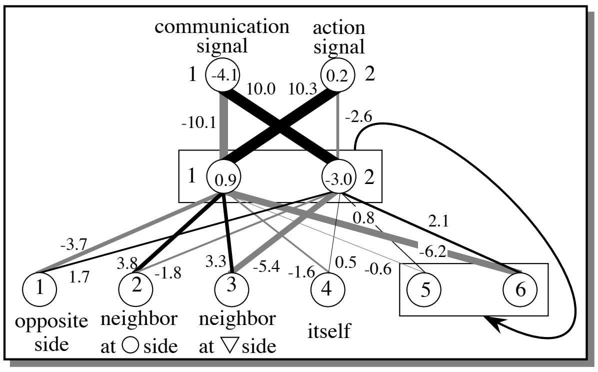
Figure 6 shows the connection weights of the recurrent neural network of the agent 0, 2 and 7. The characteristics of each agent is described in the following.



(a) Agent 0



(b) Agent 2



(c) Agent 7

Figure 6. Connection weights of recurrent neural network after learning

[All agents] In the neural network of all the agents, it is easily found that the signs of the connection weights to the hidden neurons from the input 1 (the last communication signal from the opposite side agent) and input 4 (the last communication signal of the agent itself) is opposite to those from the input 2 and input 3. That is because the agent in this simulation has to make the same decision with the opposite side agent and make the different decision from the neighbor agents. Accordingly all the 8 agents have the same tendency in their strategy decided by their connection weights.

[Agent 0] The agent 0 can generate seven sequences of communication signals and action as shown in Table 3 by the recurrent neural network as Fig. 6 (a). The absolute value of the connection weights from the input 1, 2, 3 are large and those from the input 4, 5, 6 are small. This means that the agent 0 does not have strong intention and tries to adjust its action to the others. That is the reason why the agent 0 generates a variety of sequences of communication signals.

[Agent 2] On the contrary, the agent 2 does not change its communication signal as shown in Table 3. That is also known from Fig. 6 (b) in which the connection weights from the hidden neurons to the communication output is close to 0.0 and the bias of the output is very small.

[Agent 7] The agent 7 has an ability to change the communication output according to some contexts using recurrent neural network. In Table 2 (a) and (d), the received communication signals at the chance #1 is the same as those at #2, but the communication signal of the agent 7 at chance #2 is different from that at #3. The ability can be explained roughly from the connection weights as shown in Fig. 6(c). Since the hidden 2 neuron is very small according to the bias at the chance #1, the hidden neuron 1 becomes large and the communication output becomes small at the chance #2 by the connection weight from the input 6 to the hidden 1 and the weight from the hidden 1 to the output 1. However, since the hidden neuron 2 becomes large by the connection weight from the input 3 to the hidden 2 at the chance #2, the hidden neuron 1 becomes small and the communication output becomes large at chance #3.

In the case of the recurrent neural network with 4 hidden neurons, 100 simulations were done varying the initial connection weights of each agents. When BPTT was applied, in 42 simulations, all the combinations of two agents can go through the route without collisions. In the case of normal BP, all the agents go through the route in 26 simulations. BPTT is slightly useful in this simulation.

4. Conclusion

We proposed to divide communication among multiple agents into two classes with respect to its meaning. The first one is to transmit the observed information, and the second one is to transmit the agent's intention. We also proposed the architecture using Elman type recurrent neural network for the learning of the latter communication autonomously and decentralizedly from the reinforcement signal. The problem in which the agents avoid some collision by negotiation was adopted as example. Although the communication contents, strategy and representation were not given to the agents beforehand, they became to be able to avoid the collision through the communication. It was known that the recurrent neural network kept the past information as occasion demands, and the agent negotiated adaptively according to the other agents. Among the agents, there were differences in the degree of the persistence of its intention. We think that it can be said individuality. This individuality emerged even if the learning of all the agents are the same. Furthermore, in the four agents negotiation problem, which cannot be solved only by one-to-one negotiation, the solution could be obtained without sharing reinforcement signal to the other agents. This process is similar to the optimizing process by Hopfield network with respect that each element descends a potential surface of the whole system by a decentralized way through the interaction with the others. This architecture is useful particularly for the emergence of communication to transmit its intention, but it can be applied to the emergence of communication in multi-agent system generally.

5. Acknowledement

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", and also by the Grant-in Aid for Scientific Research (No. 10450165) supported by the Ministry of Education, Science, Sports and Culture of Japan.

6. Reference

[1] Werner, G. M. & Dyer, M. G., "Evolution of Communication in Artificial Organisms", Proc. of Artificial Life II, pp.1-47 (1991)

[2] Nakano, N., Sakaguchi, Y., Isotani, R. & Ohmori, T., "Self-Organizing System Obtaining Communication Ability", Biological Cybernetics, 58, pp.417-425 (1988)

[3] Davis, R. & Smith, R. G., "Negotiation as a Metaphor for Distributed Problem Solving", Artificial Intelligence, Vol. 20, No. 1, pp. 63-109 (1983)

[4] Kreifelts, T. & von Martial, F., "A Negotiation Framework for Autonomous Agents, Demazeau, Y. & Muller, J.-P. eds., Decentralized A. I. 2, pp.71-88 (1991)

[5] Elman J. L., "Finding Structure in Time", Technical Report CRL 8801, Center for Research in Language, Univ. of California, San Diego (1988)

[6] Rumelhart, D. E., Hinton, G. E. & Williams, R. J., "Learning internal representations by error propagating", Parallel Distributed Processing, Vol. 1, MIT Press, pp. 318-362 (1986)