# An Application of Automaton Neural Networks to Artificial Agents

| | |
|---|---|
| メタデータ | 言語: English<br>出版者: 琉球大学工学部<br>公開日: 2007-09-16<br>キーワード (Ja):<br>キーワード (En): Automata, Neural networks, Intelligent agents<br>作成者: Kawano, Yoji, Nakao, Zensho, Chen, Yen Wei, 仲尾, 善勝, 陳, 延偉<br>メールアドレス:<br>所属: |
| URL | http://hdl.handle.net/20.500.12000/1965 |

# An Application of Automaton Neural Networks to Artificial Agents †

Yoji Kawano*, Zensho Nakao**, & Yen Wei Chen**

E-mail: {kawano, nakao, chen}@augusta.eee.u-ryukyu.ac.jp

**Keywords:** Automata, Neural networks, Intelligent agents

## Abstract

There is presented a model that transfers artificial intelligence into an intelligent Neural Network, which is called AUtomaton Neural Network (AUNN), and is composed of two algorithms: an automaton algorithm and a neural network algorithm.

The model was applied to artificial agents to provide them with intelligence, and its applicability was demonstrated by computer simulation.

## 1. Introduction

Today, there are a lot of intelligent systems which are applied to various fields, where the research on artificial agents which possesses intelligence is making rapid progress. In this paper, we present a model that transfers artificial intelligence into an intelligent Neural Network. We call the model AUtomaton Neural Network (AUNN). AUNN is composed of two algorithms: an automaton algorithm and a neural network algorithm[1][2][3].

We apply this model to artificial agents to provide them with intelligence. The agent searches for a goal and keeps on moving to get to the goal in a field. If the agent does not have AI, it will be unable to act adaptively when it is faced with an unexpected. For example, when the goal is a moving one, the agent is going to lose its way. But the agent having AI resolves such problems. It will think of what to do by itself and look for an optimal way.

We have adopted AUNN for agents' brain, and consider its applicability by showing several experimental results

## 2. AUtomaton Neural Networks (AUNN)

### 2.1. The structure of AUNN

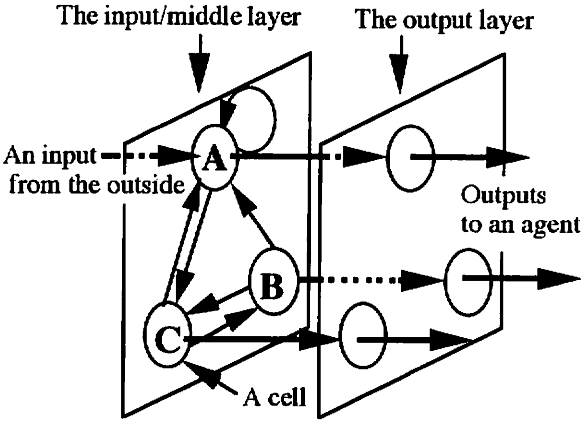


Fig. 1 An automaton neural network

AUNN works as intelligence of an agent. When AUNN develops or grows, the agent will behave like a creature which is learning. The idea of AUNN is depicted in Fig.1. The AUNN system is composed of

Recieved on : December 1, 1998.

* Graduate Student, Dept. of Electrical and Electronics Eng.

** Dept. of Electrical and Electronics Engineering, Fac. of Eng.

† Part of the paper was presented at KES'98 (Adelaide, Australia), April 21-23,1998.

two layers: one is an input/middle layer, the other is an output layer. It looks like a neural network, but a process in the input/middle layer is similar to that of an automaton. The input/middle layer (Fig. 2) has many cells. Each cell has several inputs and two outputs, and is connected with one output cell which has information to determine an action of an agent at binary data from one input, and then classifies the data into two kinds, 0 or 1. In Fig. 2, paths for 0 and 1 data are designated. For example, in Fig. 2, if cell A gets a data 0, then A outputs a signal 0 to itself (cell A); if cell A gets 1, then A outputs a signal 1 to cell C. When cells B and C, get a binary data, a similar process is followed. Repeating this process excites many cells. Finally, an output from an output cell connected with the last excited input/middle cell commands the agent to perform an appropriate action.

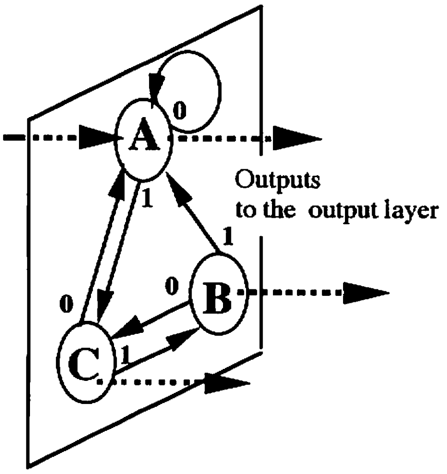


Fig. 2 An input/middle layer

## 2.2. Learning Algorithm

The AUNN is attached an energy value which expresses how far the agent is from the goal. The neural network is changed by a learning method when the system energy gets lower, i.e., the agent gets farther from the goal. The learning method changes condition of the combination of cells in the input/middle layer and cells in the output layer. In other words, the method changes a direction of an arrow lying between the input/middle layer and the output layer shown in Fig. 3. In Fig. 3, if the last excited cell is A connected with cell A' in the output layer and an output from the cell makes the system's energy be lower, i.e., the agent gets far from the goal, then the head of an arrow (which cell A has) changes to point. from cell A to cell B'. This operation is made every time the system's energy gets lower, and results in the neural network being more robust.

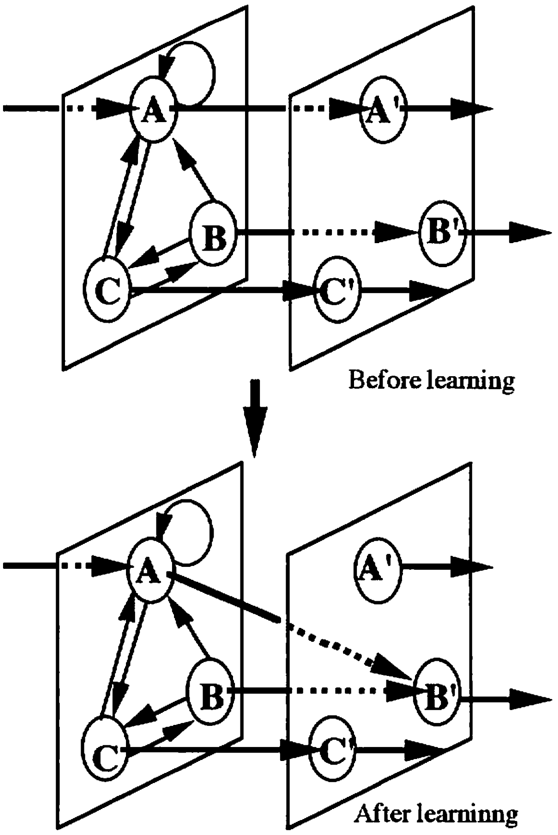


Fig. 3 The learning method

# 3. The system

We describe the system which makes an agent move to a goal in a field. The above algorithm is used for AI of the agent. The agent having AI moves, searches the goal, and gets there in the field[4][5].

## 3.1. The field

The field where an agent moves around has

1. An outer wall,
2. Several walls to block an agent's way, and
3. An agent and one goal.

The outer wall prevents an agent from leaving the field, and other several walls prevent the agent from getting at a goal too easily, i.e., it will have to avoid several obstacle walls.

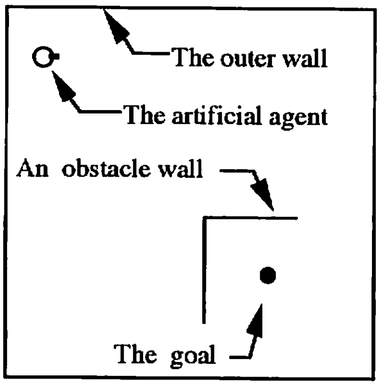


Fig. 4 An example field

### 3.2. The goal

The goal is a destination for an agent, so the agent moves to reach the goal. The goal which is placed in the field keeps on sending signals that are binary numbers to the entire field. The signal is strongest at the goal and gets weaker as the distance is farther from the goal. The agent gathers this signal for its input and learns to close in to the goal.

### 3.3. The agent

The agent has several functions such as:

1. The directions it can face are up, down, left and right,
2. It can move forward but not backward,
3. It can get data in front of it, and
4. It cannot go through any walls.

As described above in 3 or shown in Fig. 5, the agent can get data in front of it to move. It gets the data in there as an input to AUNN, acts, and learns.

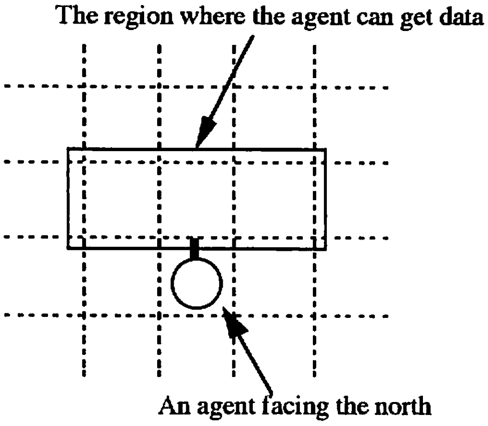

The region where the agent can get data
An agent facing the north

Fig. 5 The region which the agent can get data.

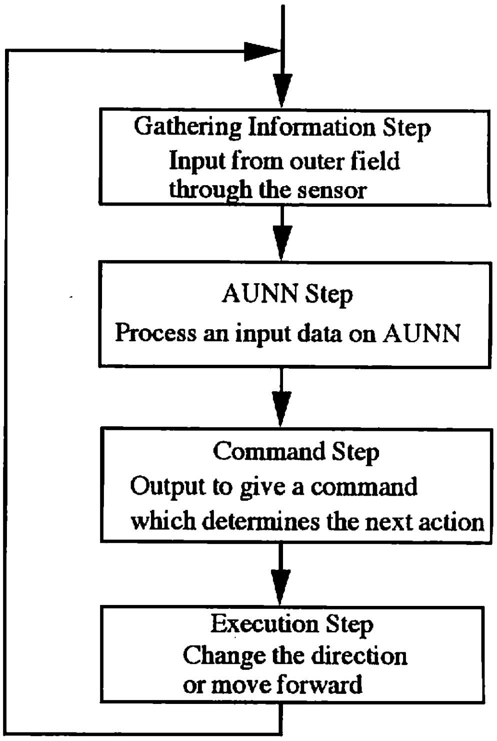


Fig. 6 A flow chart of the system

The agent can remember a value of data, and where he is and he was before he moves one step. Comparing two data, the agent decides whether he has to learn or not. That is to say, when the value of data which the agent got at a previous step is bigger than that at a present step, the agent adapts and improves the AUNN.

The agent gets some data from around it in the field to learn. In Fig. 6 is shown a flow of the processes

## 4. The Experiments

We performed computer simulation under several conditions. The field dimensions for all experiments are of 60x60.

1. There is a goal and no walls in the field.

As shown in Fig. 7(a), there is a goal which is fixed and no walls in the field. The location where the agent is placed is fixed. The agent begins to move from there, and it is placed at the same point when it succeeds getting to the goal. We count the number of the steps for it to move from the start to the goal, and repeat it.

2. There is a goal and are several walls in the field. The second experiment is similar to the first one without obstacle walls. The experiment includes several walls as shown in Fig. 7(b), so the agent has to avoid these walls to get to the goal. As in the first experiment, when the agent reaches the goal, it is placed back at the start point.

3. There is a randomly appearing goal in the field. In the third experiment, a goal appears randomly in the field. When the agent gets at the goal, another goal appears randomly in the field. The agent is not placed back at the start point when it reaches the goal. The agent searches for a goal which appears in the field. Fig. 7(c) shows the third one, where the numbers above black dots indicate an order of appearance of the goals in the field.

4. There is a moving goal and no walls in the field. In the fourth experiment, a goal escapes from the agent in a random direction in the field when the agent gets nearer. In this case, the agent is not placed at the starting point.

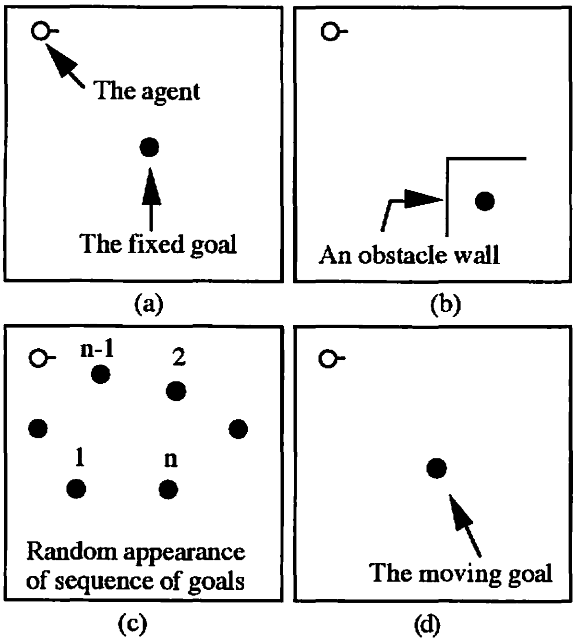


Fig. 7 (a)The 1st experiment, (b) The 2nd experiment, (c) The 3rd experiment, and (d) The 4th experiment.

## 5. Simulation Results

1. The first experiment.

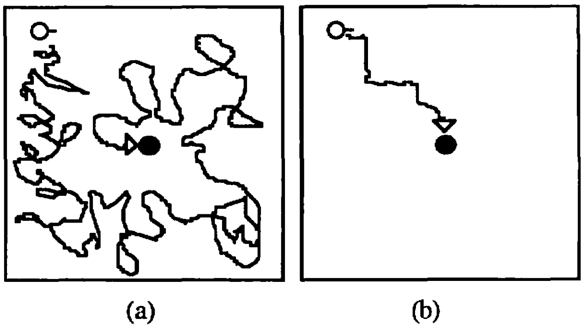


Fig. 8 Performance of the agent in the 1st experiment (a) before enough learning, and (b) after enough learning.

As shown in Fig. 8, before learning, the agent looked like losing its way. But it found a better way after enough learning.

2. The second experiment.

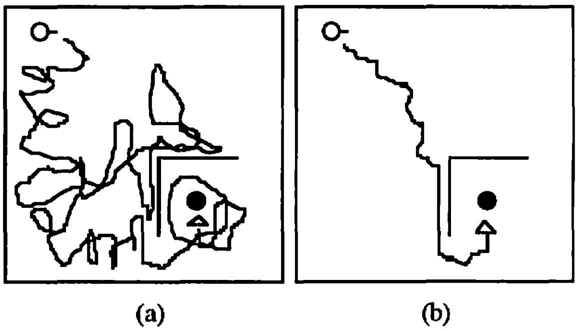
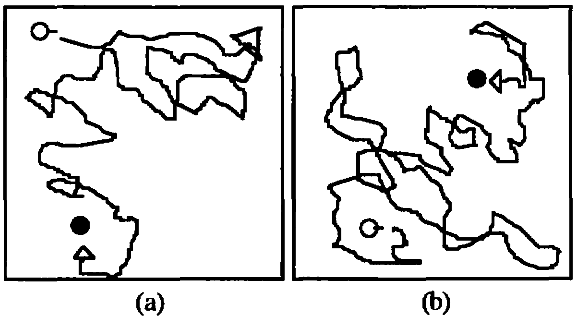
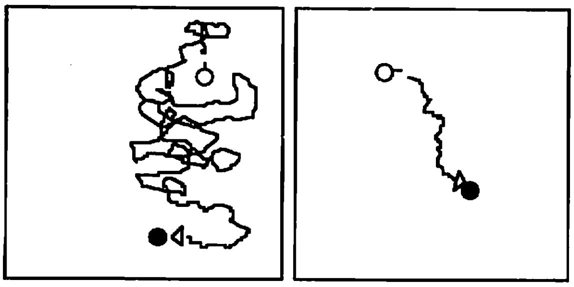


Fig. 9 Performance of the agent in the 2nd experiment (a) before enough learning, and (b) after enough learning.

As shown in Fig. 9, before learning, the agent looked like losing its way. But it found a better way after learning like the first one

.3. The third experiment.



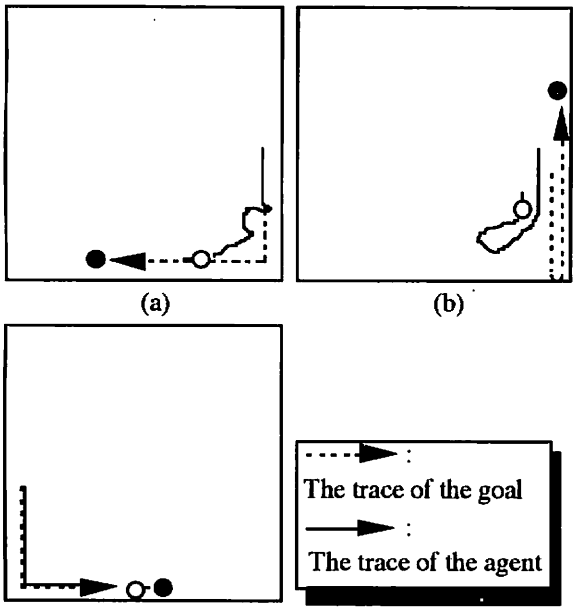


(c) (d)

Fig. 10 Performance of the agent in the 3rd experiment. (a), (b), and (c) in an early learning stage, and (d) after the learning stage.

As shown in Fig. 10, before learning the agent acted like it did not know where the goal was; however, after enough learning it could find the goal as soon as it appeared.

4. The fourth experiment.



(a) (b)

(c)

Fig. 11 Performance of the agent in the 4th experiment. (a) chasing the moving goal, (b) reacting to an unexpected move, and (c) after enough learning.

In the fourth experiment, the agent displayed similar actions as the first, second and third ones in an early stage. As shown in Fig. 11, after repeated learning, the agent chased the goal. In Fig. 11(a), it tried to take a short cut; however, in Fig. 11(b), it lost the way because the goal took an unexpected action. Repeating processes such as those, it finally chased just behind the goal.

All these experimental results demonstrate that the agent acts like a natural creature. But we observed that it loses its way easily when there are several obstacles in the field.

## 6. Conclusion

We adopted AUNN as an artificial agent's brain. The computer simulation results show the applicability of the system.

In this paper the learning method changes the condition of combination between the input/middle layer and the output layer. If, however, it changes the condition of the input/middle layer, i.e., it changes the condition of automaton, the system will perform better, and it is a new learning method that we are investigationg now.

## 7. References

[1] H Ogura: *Introduction to Formal Languages and Finite Automata - Discrete Mathematics with Examples and Problems* (in Japanese), Corona Publishing Co., Tokyo, 1996.

[2] N Baba, F Kojima, & S Ozawa: *Foundations and Applications of Neural Networks* (in Japanese), Kyoritsu Publishing, Tokyo, 1994.

[3] K Wada: *Evolution of Digital Creatures* (in Japanese), Iwanami, 1994.

[4] Ah Chung Tsoi: The role of self organization in evolutionary computations, *Proceedings of the 1st Korea - Australia Joint Workshop on Evolutionary Computation* (KAIST, Taejon, Korea), pp.129-157, 1995.

[5] T Miyagi, & Y Kawano: Self-organization in evolutionary computation, *B.E. Thesis*, University of the Ryukyus, Okinawa, Japan, 1997.