Solving maze problems by cellular neural networks

[1]Uğur Halıcı and [2]Ufuk Yaranlı

[1,2] Department of Electrical and Electronics Engineering, 06531, METU, Ankara, Turkey,
[1] email: halici@trmetua.bitnet

Abstract

In this study, the problem of orientation in maze like patterns is aimed to be solved by using neural networks in which the neurons are organized as cellular arrays. The maze patterns are coded in the array of the neurons by considering the corresponding cells in the maze. A probabilistic algorithm that handles the loops in a way similar to Tremaux's algorithm is proposed to find out a path from the initial cell to the final one by using only the local informations. The paths found in this way are not necessarily the shortest. A competitive learning algorithm is introduced by which the network learns one of these paths. The path learned in this may not be the shortest one but the learning strategy favors shorter paths as confirmed in the experimental results.

1. INTRODUCTION

A maze or a labyrinth is a network of paths through which it is hard to find one's way [5]. In greek mythology, Labyrinthos is the name a building with intricate passageways designed by Daedalus for King Minos of Crete to bewilder the uninitiated . The Minotaur, a monster with a bull's head and a man's body was kept there and fed with human flesh. Theseus, the chief hero of Athen, killed the Minotaur and escaped from the Labyrinth with the help of a silken cord given him by Ariadne, the doughter of King Minos [1]. Architectural labyrinths of this sort were not uncommon in the ancient world. The temple in Egypt, in the east to the Moiris lake, that contained 3.000 chambers, 1500 of which is under ground, is an example of this kind. The labyrinth in Lemnos having 150 columns, and the labyrinth in Closium, Italy, containing rooms on within another, are the other examples. Throughout the Middle Ages the walls and floors of many catedrals in Continental Europe, were decorated with mazes that symbolized the snakelike twists of sin and

the diffuculty of keeping on the true path. In England, mazes were cut in the turf outside the churches, and they were traversed as a part of a religious rutal . Garden mazes made of high hedges and intended solely for amusement became fashionable during the late Reneissance. The one in the Hampton Court Palace, England, and the other in Versailles Palace, France, are the famous gardens with labyrinths[1,3].

From the mathematical standpoint a maze is a problem in topology. A maze can be solved quickly on paper by shading all the blind alleys until only the direct routes remain. But in the case a map of the maze is not available, such a method is not applicable. If the maze has one entrance, and the object is to find the way to the only exit, it can always be solved by placing your hand against the right (or left) wall and keeping it there as you walk. In such a solution, to reach the exit is guaranteed, although the route is not likely to be the shortest one. The same method can be applied for the mazes in which the goal is within the labyrinth, provided there is no route by which you can walk around the goal and back to where you started. If the goal is surrounded by one or more such closed circuits, the hand-on-wall method simply takes the searcher around the largest circuit and back out of the maze [3].

Mazes having no closed circuits are called 'simply connected' (Figure 1.a). If a maze is simply connected, then it means that it has no detached walls. Mazes with detached walls contain closed circuits, they are called 'multiply connected ' (Figure 1.b). The Tremaux Algorithm formulated by Edouard Lucas in 1882 is able to solve all mazes, including multiply connected ones with closed loops that surrounds the goal. As the maze is walked through , a line is drawn on one side of the path, say the right side. When a new junctures of the paths is confronted, any path can be chosen. If in walking along a new path, a previously visited junction is confronted or a dead end is reached, turn around and go back the way you came. If in walking along an old path, which are already marked on the left, a previously visited juncture is reached, then a new path is taken , if one is available, otherwise an old path is taken. A path marked on both sides is never entered [3] .

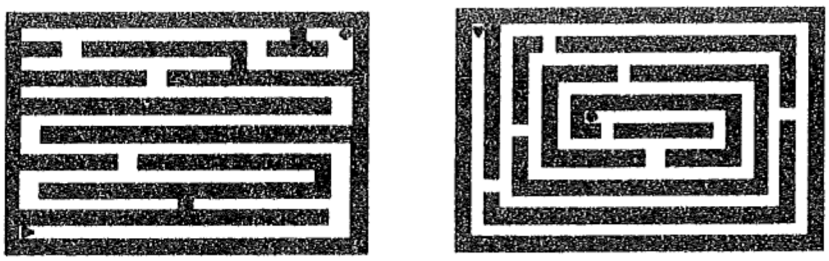

Figure 1. a) Simply connected maze b) Multiply connected maze

Psychology and computer are the fields of science in which interest in mazes is high. Psychologists uses mazes for several decades to study the learning behaviour of men and animals. For the computer point of view, they can be used for the development of the learning machines. In this paper, cellular neural networks are used to solve mazes. For this purpose the maze pattern is placed in the cellular array. In such a maze, while some of the cells permit to pass through them, some others forbid. Therefore the set of cells that permit to pass through them constitutes some passages in the maze and the elements that forbid to pass constitutes the walls.

Given a maze pattern, the aim is to find out a path in the passages connecting a predetermined initial cell to a final one, and meanwhile to learn a reasonable short path as the experiment on the maze is repeated for several times.

A probabilistic algorithm for cellular neural networks, that handles the loops in a way similar to Tremaux's algorithm is proposed to find out a path from the initial cell to the final. A path found in this way is not necessarily the shortest one. But a competitive learning algorithm is introduced by which the network learns the paths favoring the shorter paths. It has been observed through the simulations that most of the time the network learns the shortest or the next shortest paths.

2. CELLULAR NEURAL NETWORKS

The structure of the cellular neural networks is similar to the cellular automata. Each unit is a neuron and it is called cell. Any cell in the network is connected only to its neighbor cells and they can interact directly with each other. Cells which are not neighbours can affect each other indirectly because of the propogation effects of the network [2,4]. The structure of the cellular neural network used in this paper is shown in Figure 2.

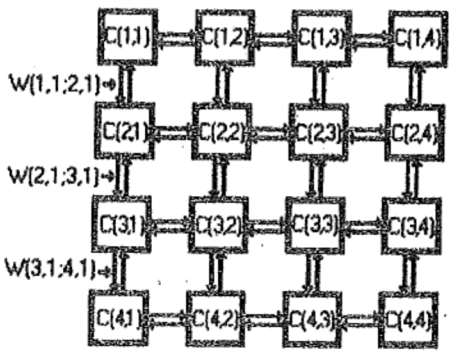

Figure 2. The structure of the two dimensional cellular net used for solving maze problem

Here C(i,j) is used to denote the cell in row i, column j. For a cell C(i,j) the neighbors are

$$N(i,j)=\{C(i-1,j),C(i+1,j),C(i,j-1),C(i,j+1)\} \quad (1)$$

which are the cells in the left, right, up and down respectively. The notation (i,j;k,l) is used to denote the connection from C(i,j) to C(k,l), and W(i,j;k,l) to denote its weight. In this paper, the connection weights are not assumed to be symmetric, that is W(i,j;k,l) is not necessarily equal to W(k,l; i,j) . Furthermore, it is guaranteed that the summation of the weights of the connections from a cell to the neighbors is equal to one, that is for each C(i,j) in the network:

$$W(i,j;i+1,j)+W(i,j;i-1+j)+W(i,j;i,j-1)+W(i,j;i,j+1)=1 \quad (2)$$

Here, the cellular neural network is not used in the classic meaning. The structures of the units and the connections are almost the same as before, but the connections are activated instead of the units for describing a path from the initial cell to the final. Only the strengths of the active connections are updated when the network is forced to learn this path. The outputs of the neuorons are either 0 or 1. The connection weights are effective in choosing the next neuron in the path, whose output is to be set to 1.

For a cell C(i,j) the connection weights are assigned initial values as follows: W(i,j;k,l) is set to 0 and never changed if C(k,l) is a cell in the wall, else W(i,j;k,l) is set to 1/n where n is the number of neighbors that are in the passages. The weights of the connections are changed by learning.

For the maze problem, one of the cells is predetermined as the initial, and one another as the final cell. A maze pattern is represented by clamping the outputs of the cells in the walls to 0. However the outputs of the other cells are not clamped, so they may have output value either 0 or 1. A cellular array to represent the maze pattern given in Figure 1 b, is shown in Figure 3. In the figure, the connections are not shown for the simplicity and the dark cells, corresponds to the neurons whose output are clamped to 0 since they are in the walls.. The other cells corresponds to the passages, and the shaded part represent the cells taking place in a path from the initial cell to the final one. For the given example, the cell C(2,2) is the initial cell and C(8,10) is the final. In the figure several different paths connecting the initial node to the final are shown.

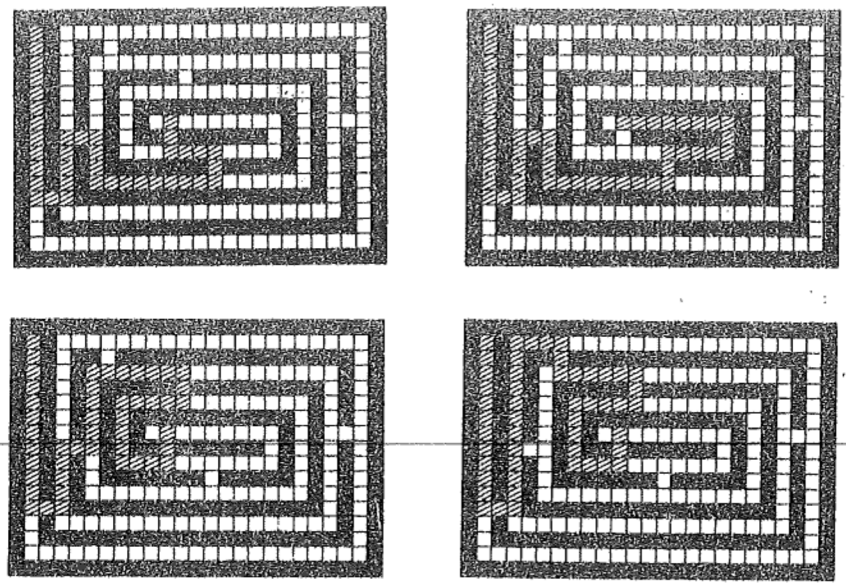

Figure 3. The cellular array representing the maze of Figure 1.b. and some possible paths having length a) 38 units b) 46 units c) 46 units d) 50 unit

3. WALKING IN THE PASSAGES

At a time, only one of the cells has output value 1 indicating where we are in the maze. For the next time, one of the neighbors of the current cell becomes on and the previous node becomes off. Such a cell in the neighborhood is chosen probabilistically by considering the connection weights. The neighbor having connection with larger weigth is more probably to be chosen. For this purpose, each neighbor is assigned a random value between 0 and the weight of the corresponding connection. Then, the one being assigned the largest value is chosen as the next node to be visited. Since the connection weights to the wall cells are initially set to 0 and never changed, such a cell is never selected as the next one.

In the beginning, the neuron for the initial cell has output value 1, and all the others are 0. Finally it is expected that the final node will be on and it will remain as 1, unless a new experiment has not been started. Since the wall cells are clamped to 0, they never have value 1, that means it is not possible to pass through a wall cell.

Within an experiment, the network is assumed to be able to remember the activated path in its short term memory. When a cell C(k,l) in N(i,j) is chosen just after cell C(i,j) then the connection (i,j;k,l) is activated if connection (k,l;i,j) is not already active, otherwise the connection C(k,l;j,i) is deactivated instead of activating C(i,j;k,l). The first part corresponds to propagating in a passage and the second

corresponds to return. Special care should be taken for the loops of the passages. Such a situation is implied if one of the outgoing connection say C(i,j;k,l) is already active when we arrive a cell C(i,j) and it can be checked by using local information. If existence of such a loop is distinguished by the active connection (i,j;k,l) when we arrive in C(i,j), the algorithm directs to choose C(k,l) as the next cell and deactivate the connection (i,j;k,l). Therefore the loop is erased by passing through it for a second time in the same direction. When the final node is reached, which is the end of the experiment, we have a path made of active connections whose information is stored in short term memory.

4. LEARNING THE PATHS

Initially, the weights of the connections going out a cell is set inversely proportional to the number of neighbors that are in the passages but not in the walls. When an experiment terminates, only the weights of the connections going out from the cells that are on the path are updated. Say, C(i,j) is such a cell then W(i,j,k,l) is updated by the formula:

$$W(t+1)=\begin{cases} W(t) + c/L & \text{if connection (i,j;k,l) is activated in the path} \\ W(t) - c/Ln, & \text{connection (i,j;k,l) is not activated,but C(k,l) is not on a wall} \\ \text{no change} & \text{if C(k,l) is on a wall.} \end{cases} \quad (3)$$

where W(t) is the weight of the connection (i,j;k,l) at trial t, L is the length of the path, n is the number of neighbors of C(i,j) that are not on a wall. In the formula, c is constant about Lmax/100 where Lmax is the length of the longest path in the chosen maze pattern. Notice that after the update of these weights, still the summation of the weights of the outgoing connections is 1. In general, updating weights means storing information, that is correlated with the path in the last trial, in the long term memory as a result of learning.

5. EXPERIMENTAL RESULTS

A cellular neural network of size 17x25 as shown in Figure 3 is used for the solution of the maze given in Figure 1.b. The network is tested for 30 different random seeds.

For each seed the following procedure is applied:
1. Clamp the outputs of the neurons corresponding to the walls of the maze to 0,
2. Set initial weights as explained in section 2,
3. Find a random path as explained in section 3,
4. Update the weights as explained in section 4,
5. Goto step 3 for the next trial (repeated for 300 times in the experiments)

Figure 4. shows the results for some different seed values in which the time elapsed in finding a path, the length of these paths and the time/length ratio for the consecutive trials are demonstrated. Figure 5. is the histogram showing the distribution of the paths that are learned for 30 different seeds.

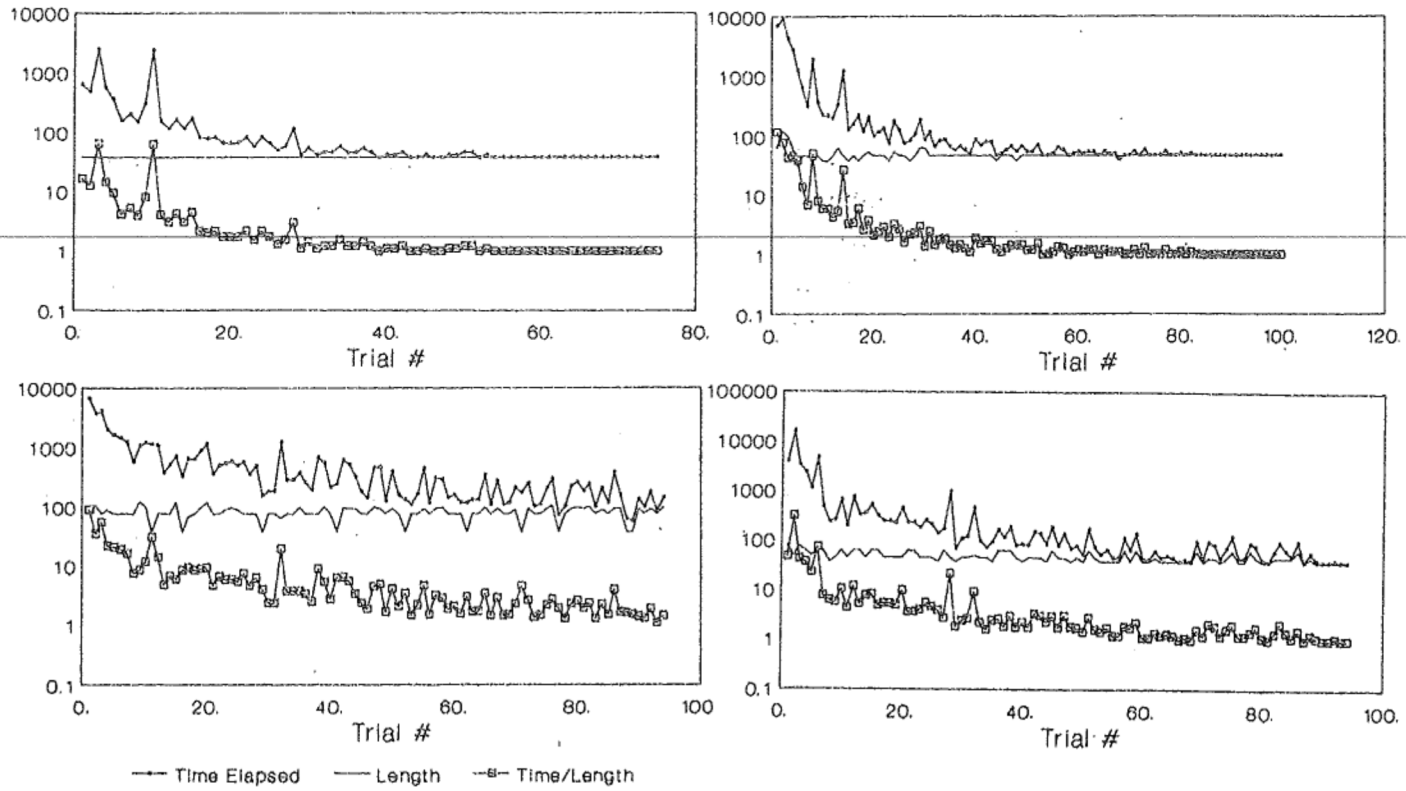

Figure 4. Experimental results for some different seeds a) seed 0 b) seed1 c) seed 7 d) seed 15

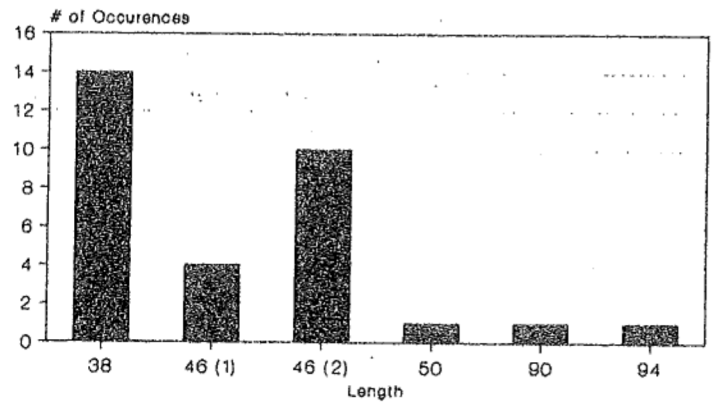

Figure 5. Distribution of paths that are learned for 30 different seeds.

Through these experiments it has been observed that:
1. Initially it takes a long time to find a path. However as the system learns in each trial, the time/length ratio converges to 1 indicating that the path is directly followed without going back and forward unneccessarily.
2. The learning strategy favors the shorter paths, but does not guarantee that it will be the shortest one.

6. CONCLUSION

The problem we put forward is to find a correct path which connects an initial cell to a desired final one in a cellular neural network representing a maze. A probabilistic algorithm based on cellular neural networks is proposed to find a path as the solution of the problem and a strategy to learn these paths is also given. In such a schema it is not guaranteed to learn the shortest path, but it favors the shorter paths. The experimental results are quite reasonable indicating the existence of such a learning.

As a feature work, we are going to search the effect of forgetting some prefix of a path when it is to be learned and also the effect of partially remembering the previous paths on the performance of learning the shortest path. Furthermore, a genetic algorithm will be developed aiming to find out the shortest path as a result of the mutations on the paths that generated previously.

REFERENCES

1. Labirent, *AnaBrittanica* ,(Anayayıncılık A.Ş. and Encyclopedia Brittanica Inc., 1989).
2. Chua L., Yang L., Cellular Neural Networks: Theory, *IEEE Trans. on Circ. Sys*,35(10), October 1988.
3. Gardner M., Mazes, *More Mathematical Diversions*, (Pelican Books, 1961, pp 88-93)
4. Savran M.E. and Morgül Ö., On the Design of Hopfield and Cellular Neural Networks, in *Proceedings of ISCIS VI*, November 1991, pp 899-905
5. Thorndike E.L., Barnhart C.L, *High School Dictionary*, (Doubleday & Company Inc., Newyork, 1941)