Playing With Mazes

David B. Suits
Department of Philosophy
Rochester Institute of Technology
Rochester NY 14623

david.suits@rit.edu

5. A Neural Net Maze Solver

The solution methods I have discussed involve putting various identifying marks in the cells of the maze (or, what comes to the same thing, "marking" a memory image of the maze). The memory — the record — which helps guide you through the maze is imprinted upon the maze (or a copy of the maze) itself. In order to determine what to do next, at any given position, you need only consult the marks in the present cell and apply an appropriate rule. If there is an independent memory in use, it is minimal: perhaps a counter, for example.

Imagine, however, that you are trying to move through a maze without the benefit of being able to leave identifying clues (or clews). Your own memory serves only as a more or less (un)reliable "feel" — a sense of context, perhaps, which grows with experience of the maze. That is, as you move through the maze you more and more confidently identify patterns of kinds of cells. You may learn, for example, that there is one section of the maze with a very long corridor, each cell of which has a door to the right leading to a cul-de-sac. If you have knowledge of such a corridor, then you might recognize it whenever you move through it, and this identification might serve adequately as a means of orientation. Such a memory is a behavior pattern.

The problem now in solving a previously learned maze under the constraint of not being able to leave marks in the cells is to first orient oneself. Once oriented, it is a matter of following one's internalized "feel" in order to remain oriented.

This is usually inherently serial. One might not clearly anticipate the proper moves in advance; that is, you might not be able to recall, in advance, that at a certain point you will have to turn right. Rather, you will simply wait until the pattern seems to emerge on its own in response to the present context.

An artificial neural net embodies at least some of the properties we need in order to create such a memory. Neural nets do not store explicit memory images; rather, they contain a network of interactions which, given the proper stimulation, will *recreate* certain kinds of responses. We may say that the neural net's memory is its ability to engage in such recreations.

But most research on artificial neural networks has not focused on the time domain, whereas I am searching for a neural net architecture which will tend to serialize its recreations. I propose a sketch of a possible neural network to solve mazes. It is based roughly on an idea for serial learning proposed by Jordon (1986) and later used by Todd (1989). But whereas Jordon's network learned by means of back-propagation (wherein mistakes made by the net are corrected by an all-knowing teacher), my proposal is for a network which teaches itself.

Figure 22 shows the components of the network. The general idea is that the network will receive information about the type of cell it is in (i.e., which sides of the cell have doors and which have walls), make a decision as to the direction of motion out of the cell (to simplify matters, I use compass directions instead of an "egocentric" orientation), and then predict the kind of cell it will be moving into. This prediction is then compared to the actual cell type moved into, and on the basis of that comparison, various of the net's weights will be affected. Also, if the goal cell is ever moved into, some of the net's weights are affected. The four move nodes constitute a "winner-take-all" subnet, activated by the "decision" subnet and strongly inhibited by various of the

"actual cell type" nodes in order to ensure that a move chosen is a possible move. The move nodes are mutually inhibitory so that as one node's activation tends to grow, the others tend to diminish, with the result that eventually one and only one node will be active as the chosen direction.

The activity of the decision subnet is fed into the predictor subnet. Active nodes in this subnet are taken to represent the one (or more) cell types which the node expects to see after making the move indicated by the active node. They also act as a kind of "context" which is verified or falsified by the action of weight adjustments which take place after comparing the predictor's predictions with the actual new cell type. If this idea of "context" works, then the net will choose moves in a serial order based in some degree upon past modes and some degree on the present situation. That is, the net will "learn" a route from a given start cell to a given goal cell.

Once having learned a route, the start cell and/or the goal cell may be changed and the net allowed to investigate anew. The hope is that eventually the net will have a "feel" for the entire maze, and that it will be able roughly to orient itself anywhere in the maze after a number of moves.

An interesting feature about this proposed net is that it is given no knowledge about the size of the maze; the net, if it works at all, will work on different sized mazes. (There is obviously some limit to the maze complexity of course.)

I have been talking about the net as a proposal rather than as an accomplished machine, because at this stage it is only partially implemented. The basic structure of most of the nodes has been worked out, but it will be some time before all the pieces can be put together for some serious trials.

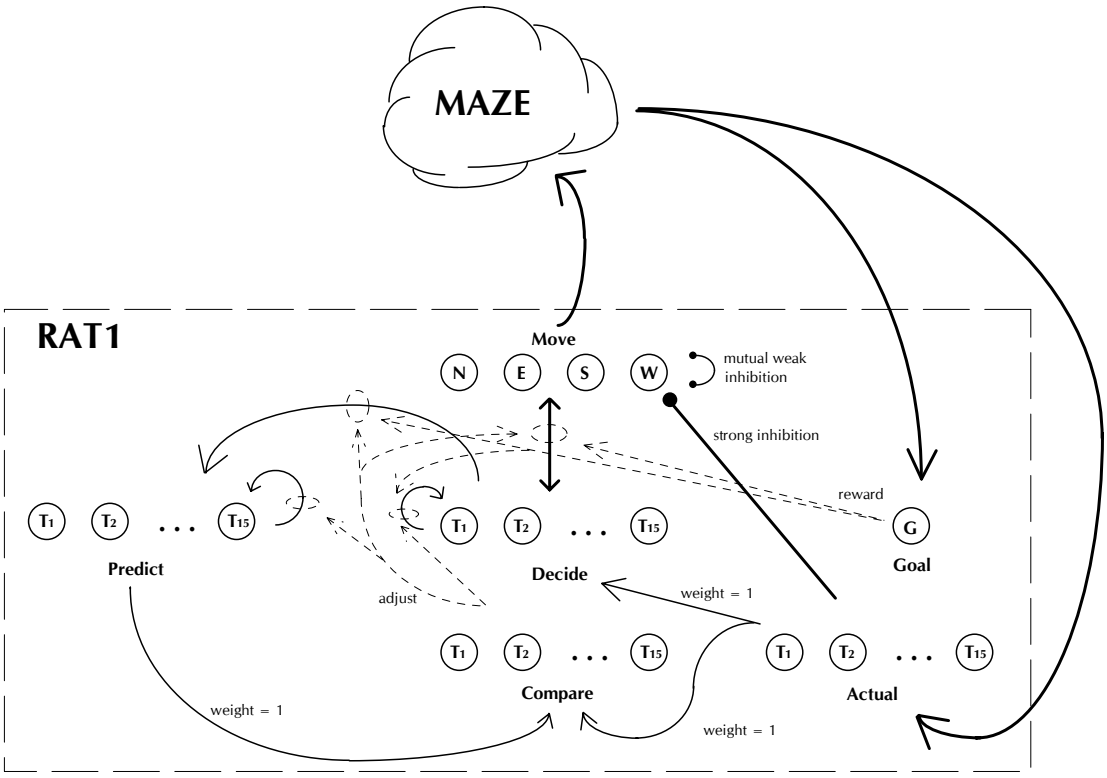
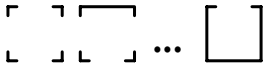


Figure 22. A proposal for a maze-learning net.

T_1, T_2, ..., T_{15} are 15 maze cell types:



Broken lines are weight adjustment lines.

1. If predicted cell type = actual cell type, certain weights are adjusted away from zero.

2. If predicted cell type ≠ actual cell type, certain weights are adjusted toward zero.

3. If goal cell is reached, certain weights are rewarded (moved further from zero).