The Cellular Simultaneous Recurrent Network Adaptive Critic Design for the Generalized Maze Problem Has a Simple Closed-Form Solution

Donald Wunsch
Applied Computational Intelligence Laboratory
University of Missouri - Rolla
dwunsch@umr.edu

Abstract

The generalized maze problem has been considered as an interesting testbed by various researchers in AI and neural networks. The most significant results, from a neural networks point of view, were:

1. Simultaneous recurrent networks are necessary if a neural network-based cellular automaton approach to the problem is to be successful.
2. These networks can be designed so that convergence to a correct solution is assured.

Here, a simple closed-form solution for the critic is shown, making adaptation unnecessary. Furthermore, it is shown that the design converges to the correct solution in only J steps, and the worst case convergence speed for an N x N mesh is derived.

Introduction

The generalized maze problem can be simply specified, in matrix or graph notation. Consider a mesh, as shown in Figure 1. Each node in the mesh is a target node, a pathway node, or a barrier. Each edge has a cost associated with it, infinite for a barrier, or one, otherwise. The J function of the critic is the expected total optimal cost-to-go from a specific node. The maze is generalized because the graph can be any size, and any node may be a target, pathway, or barrier. Any currently occupied node may be considered as the starting point.

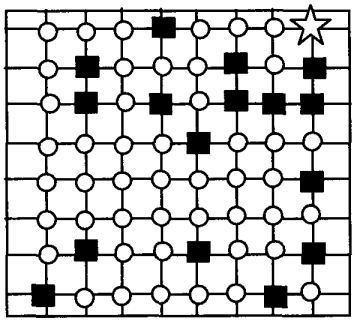

Figure 1. Graph representation of the generalized maze problem. The star indicates the target, nodes with circles are pathway nodes, and nodes with black squares are barriers. The border can be considered to consist entirely of barriers.

In [1], Werbos and Pang show the noteworthy result that this problem may be solved by a cellular simultaneous recurrent neural network (SRN) structure, of fixed architecture, repeated at each node. The weights, once learned, may also be repeated at each node. They also show that the cellular neural net approach will not work with a feedforward design, an argument that may also be derived from [2]. It is noteworthy that the architecture is fixed, regardless of maze size or complexity. The price for this, of course, is the repetition of the entire network at each node of the maze. Still, the design simplicity is attractive.

0-7695-0619-4/00 $10.00 © 2000 IEEE

The structure of the cellular SRN is shown in Figure 2. The eleven input nodes consist of a code for whether the current node is a target, barrier, or pathway (node inputs 1 and 2), the current J function output for each of four neighbor nodes (node inputs 3-6), and feedback inputs from the hidden layer (node inputs 7-11). There are five hidden nodes, and one additional product node at the output.

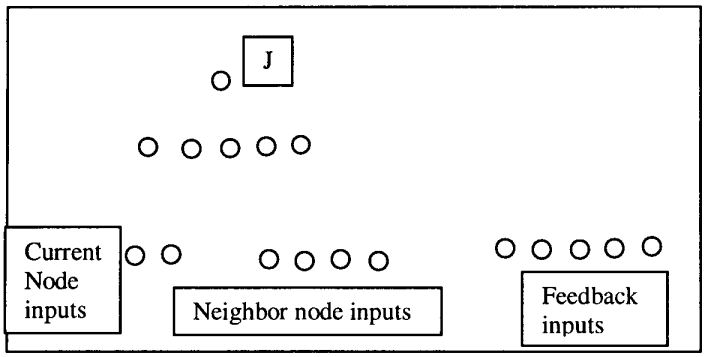

Figure 2. Cellular SRN structure. Connections are not shown, but each layer is globally connected to the following layer, and additional feedback connections exist between the hidden layer and the feedback inputs of the bottom layer. This is known to be trainable to solve the generalized maze problem.

In [3], Werbos shows that architectures of this class can be trained to converge to a correct solution. Here, a closed form solution is solved for. To do this, we will number the nodes $x_{i(a,b)}$. We will normally suppress the (a,b) subscripts, since the architecture is repetitive over the grid, but for certain nodes we will need them. Nodes x_1 through x_{11} are as defined above. Nodes 12 - 15 are the hidden nodes. Node 16 is the output node. We will assign the first node values as:

x_1 = { 0 if an obstacle, 1 otherwise}
x_2 = { 0 if a target, 1 otherwise}
x_3 = $x_{16(a,b-1)}$
x_4 = $x_{16(a-1,b)}$
x_5 = $x_{16(a,b+1)}$
x_6 = $x_{16(a+1,b)}$

Now we require for the output node:

$$x_{16} = (x_2 / x_1)[\min\{x_6, x_5, x_4, x_3\} + 1]. \quad (1)$$

(Of course, in practical implementations, we will use a very small value for x_1, rather than 0, if it is an obstacle.) To do this, we can use an on-center-off-surround feedback OCOS network. An excellent general survey treatment of OCOS networks and how their parameterizations affect their properties is in [4]. For our purposes here, a simple special case of that architecture will suffice. We will use the name MAXNET, as discussed in [5]. This architecture selects the maximum among its inputs via lateral inhibition, providing a value of 1 for that node, and zero for the others. We can now write:

$$x_{16} = (-1)\,(x_2 / x_1)[\text{Arg}(\text{MAXNET}\{-x_6, -x_5, -x_4, -x_3\}) + 1], \quad (2)$$

where Arg(.) is the value of the maximal input. This can be achieved by a simple sigma-pi modification of MAXNET, where each input is fed along with the corresponding MAXNET output to a pi (product) neuron. These product neurons are nodes x_{11} to x_{14}. Node x_{15} inserts the (+1) bias term in (2). Node x_{16} only needs to weight the result by (x_2 / x_1) to achieve the desired result. The MAXNET and product node portion of this architecture are shown in Figure 3.

This completes the specification of the closed form solution, except to note that, for an arbirtrary NxN grid, we should begin with initial values:

$$x_{16} \geq N^2 \qquad (3)$$

for all nodes except the target node, where

$$x_{16} \equiv 0. \qquad (4)$$

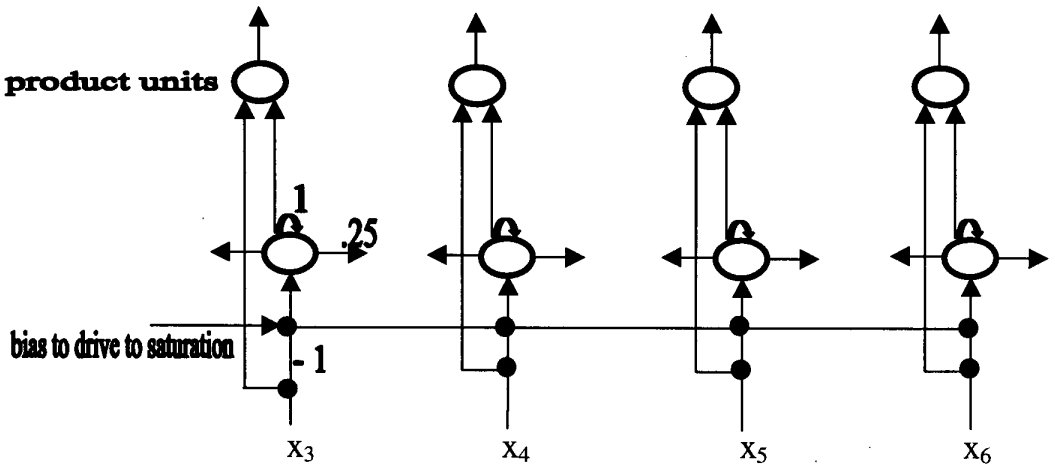

Figure 3. The MAXNET and product node portion of the architecture. Exactly one of the outputs will be nonzero, and will be equal to $\min(x_6, x_5, x_4, x_3)$.

To analyze convergence speed, consider the smallest nontrivial case, N = 5. The worst case maze, from a convergence speed point of view, is given in Figure 4.

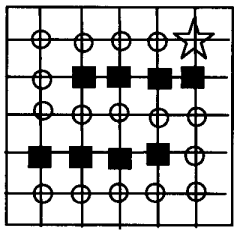

Figure 4. A 5x5 grid to illustrate convergence analysis. Recall that outside border points are considered barriers.

In any properly initialized configuration, the target node will have the correct value of J at all times, since all other nodes will have $J \geq 0$. Any move incurs unit cost, so any path node adjacent to J will, in one time step, attain J = 1, by (2). Subsequently, any node of optimal distance 2 from the target will attain the correct value of J in two steps, and so on.

Finally, the worst case convergence time (WCT) can be computed. For the 5 x 5 grid, we can easily see that it is:

$$WCT = N^2 - 2N + N - 3 = N^2 - N - 3. \qquad (5)$$

We show this is true in general by induction. If it is true up to arbitrary N, then for M = N +1, the worst case is that we must traverse all extra points except the target (which must be one of the extra points to achieve the worst case). This gives an extra 2N points to traverse. Now

$$M^2 - M - 3 = (N+1)^2 - (N+1) - 3 = N^2 + N - 3 = N^2 - N - 3 + 2N, \quad (6)$$

and we therefore have shown that (5) is the correct worst case convergence time for all N x N grids.

Conclusion

The generalized maze problem has been reexamined, a closed form solution has been obtained requiring no adaptation, and worst case convergence time has also been obtained. The results are consistent with previous research showing that an SRN-based critic is necessary.

References

1. P.Werbos & X.Z.Pang, "Generalized maze navigation: SRN critics solve what feedforward or Hebbian nets cannot.," Proc. Conf. Systems, Man and Cybernetics (SMC) (Beijing), IEEE, 1996. (An earlier version appeared in WCNN96.)
2. Minsky, Marvin, and Papert, Seymour, *Perceptrons*, MIT Press, 1969. (An expanded edition was published in 1988.)
3. X.Z.Pang & P.Werbos, "Neural network design for J function approximation in dynamic programming," Math. Modelling and Scientific Computing (a Principia Scientia journal), Vol. 5, NO.2/3, 1996 (physically 1998). Available also as adap-org 9806001 from xxx.lanl.gov/form, using the "Other Groups" field set to "nlin-sys."
4. Grossberg, Stephen and Michael Kuperstein, *Neural Dynamics of Adaptive Sensory-Motor Control: Ballistic Eye Movements*, Elsevier / North Holland, Amsterdam, 1986.
5. Lippmann, Richard P., "An introduction to computing with neural nets," IEEE ASSP Magazine, IEEE Acoustics, Speech, and Signal Processing Society, Vol. 4, No. 2, April 1987, pp. 4-22.