Adding lateral inhibition to a simple feedforward network enables it to perform exclusive–or.

Leslie S. Smith
Centre for Cognitive and Computational Neuroscience
Departments of Computing Science and Psychology
University of Stirling
Stirling FK9 4LA, Scotland, UK

Abstract

A simple laterally inhibited recurrent network which implements exclusive–or is demonstrated. The network consists of two mutually inhibitory units with logistic output function each receiving one external input, and each connected to a simple threshold output unit. The mutually inhibitory units settle into a point attractor. We investigate the range of steepness of the logistic, and the range of inhibitory weights for which the network can perform exclusive–or.

1 Background

It is well known that exclusive–or cannot be solved by a single layer feedforward network with units whose output function is monotonic [Minsky and Papert 69], but can be solved using hidden units [Rumelhart et al. 86]. We show that it can be solved using lateral inhibition to form a simple recurrent network.

2 Method

The network used is shown in figure 1a. We write I_L and I_R for the left and right inputs, S_L and S_R for the stimulation of the hidden units, L and R for the outputs of the hidden units, S_Y for

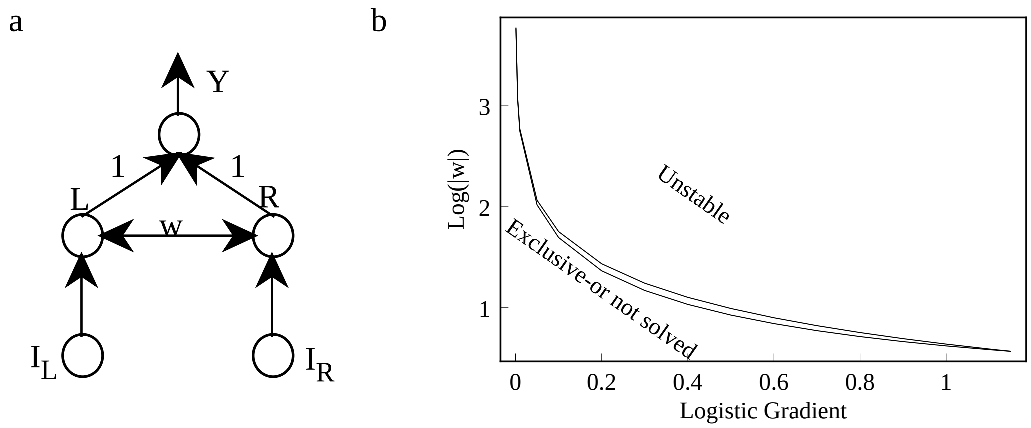

Figure 1: a: The network used. All weights are 1, except for the lateral weight, w. b: Results using a logistic output function at the hidden units. Y axis is $\log(\mid w \mid)$. The network solves the exclusive-or problem for values of w between the two lines. Actual values of w become very large for small β.

the stimulation of the output unit and Y for its output. We use a simple threshold output unit with variable threshold θ. Since the problem we are trying to solve is symmetric, we make the network symmetric. Because we choose θ, we can set hidden-to-output weights to be 1. Writing F for the monotonically increasing output function for the hidden units,

$$\begin{aligned} S_Y &= L + R \\ &= F(I_L + wR) + F(I_R + wL) \end{aligned} \tag{1}$$

This is a recurrent equation, describing a recurrent network: such a network may or may not settle to a stable state.

In order that the network can solve the exclusive–or problem, we need to be able to set θ so that

$$S_Y(0,0), S_Y(1,1) < \theta \leq S_Y(0,1), S_Y(1,0) \tag{2}$$

In addition all the values in equation 2 must be stable.

It is clear that equation 2 cannot hold for $w = 0$. An increase in I_L will result in an increase in the S_L leading (by monotonicity), to an increase (or no change) in L. For $w > 0$, this leads to an increase in S_R, and hence an increase (or no change) in R. Thus changing I_L from 0 to 1 cannot decrease either L or R, making equation 2 unsatisfiable. Equation 2 can only be satisfied if $w < 0$.

If the hidden units are linear, the stable state of the value of the stimulation of the output unit, $S_Y = L + R$, depends linearly on the inputs. Thus satisfying equation 2 requires F to be nonlinear.

The problems of satisfying equation 2 and of maintaining stability arise primarily at $I_L = I_R = 1$. We need to choose F and to make w sufficiently negative so that

$$L(0,1) + R(0,1) > L(1,1) + R(1,1) \tag{3}$$

while retaining stability. When $I_L = I_R = 1$ the fixed point is at

$$L = F(1 + wR) = F(1 + wF(1 + wL)) \tag{4}$$

We use lockstep parallel updating so that if the networks settles, $L = R$ by symmetry. We can find the fixed point by solving

$$L = F(1 + wL) \tag{5}$$

and check for stability by considering

$$L'(L) = F(1 + wF(1 + wL)) \tag{6}$$

The fixed point is stable if $\mid \frac{dL'}{dL} \mid < 1$ [Phillips and Taylor 73]. We can proceed by choosing some F, then iteratively solving equation 5 varying w to find the point at which $\mid \frac{dL'}{dL} \mid = 1$ in order to find the value for w at which the solution becomes unstable. We can then check equation 2 to see if the solution is valid. If so, we can then reduce w to find (by simulation) the least negative value for w for which the solution remains valid. If not, there are no symmetric solutions.

3 Results

We applied the above technique using the logistic function $F(x) = \frac{1}{1+\exp(-\beta x)}$. The results are shown in figure 1b. For small β there is a range of values of w such that the lateral inhibition is sufficient to fulfil equation 3 before oscillation occurs. Above the upper line the network is unstable. Below the lower line the network does not solve the exclusive–or problem. As w moves towards the upper line from below, the number of iterations can be quite large. The top line (boundary between solving exclusive–or and instability) was found analytically, and checked by simulation. The bottom line (boundary between not solving and solving exclusive–or) was found by simulation. For $\beta > 1.153$ there are no solutions.

4 Discussion

We have shown that exclusive–or can be solved using a small laterally inhibited network. The implementability and stability of this relation have been shown to depend on the steepness of the transfer function and the magnitude of the inhibitory weight. Lateral inhibition usually results in one unit winning, or in increasing contrast between output units (reviewed in [Anderson 95]). One can interpret its effect here as an indirect result of contrast enhancement. When I_L and I_R differ (say $I_L = 1$ and $I_R = 0$), the lateral inhibition enhances the contrast by decreasing R more than L. When $I_L = I_R = 1$, both L and R are decreased by the same amount. Exclusive–or is solved when the overall result of these decreases is that equation 3 is fulfilled.

When the symmetric solution discussed here exists, it is the unique fixed point for $I_L = I_R = 1$, and is found whatever the initial values of L and R, and whatever update scheme is used. As w is made more negative, so that $\mid \frac{dL'}{dL} \mid > 1$, this fixed point splits into two stable and one unstable fixed points. This results in asymmetric solutions in which for $I_L = I_R = 1$ either $L > R$ or $R > L$, and the behavior of the network depends on the initial values of L and R, and on the update scheme.

References

[Anderson 95] J.A. Anderson. *An Introduction to Neural Networks*. MIT Press, 1995.

[Minsky and Papert 69] M.L. Minsky and S.A. Papert. *Perceptrons*. MIT Press, 1969.

[Phillips and Taylor 73] G.M. Phillips and P.J. Taylor. *Theory and applications of numerical analysis*. Academic Press, 1973.

[Rumelhart et al. 86] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, volume 1, chapter 8, pages 318–362. MIT Press, Cambridge, 1986.