PERGAMON

Neural Networks 12 (1999) 1321–1323

Neural Networks

www.elsevier.com/locate/neunet

Neural Networks Letter

Solving the N-bit parity problem using neural networks ☆

Myron E. Hohil, Derong Liu*, Stanley H. Smith

Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA

Received 11 May 1999; received in revised form 6 July 1999; accepted 6 July 1999

Abstract

In this letter, a constructive solution to the N-bit parity problem is provided with a neural network that allows direct connections between the input layer and the output layer. The present approach requires no training and adaptation, and thus it warrants the use of the simple threshold activation function for the output and hidden layer neurons. It is previously shown that this choice of activation function and network structure leads to several solutions for the 3-bit parity problem obtained using linear programming. One of the solutions for the 3-bit parity problem is then generalized to obtain a solution for the N-bit parity problem using $\lfloor N/2 \rfloor$ hidden layer neurons. It is shown that through the choice of a "staircase" type activation function, the $\lfloor N/2 \rfloor$ hidden layer neurons can be further combined into a single hidden layer neuron.

Keywords: N-bit parity problem; Exclusive-OR problem; Neural networks

1. Introduction

The XOR/parity problem has a long history in the study of neural networks. The N-bit parity function is a mapping defined on 2^N distinct binary vectors that indicates whether the sum of the N components of a binary vector is odd or even.

Stork and Allen (1992) show that the N-bit parity problem can be solved with a standard feedforward neural network using just two hidden layer neurons. The activation function used in both hidden units is

$$f(u) = \frac{1}{N}\left(u - \frac{\cos(\pi u)}{\alpha\pi}\right) \tag{1}$$

where the choice of $\alpha > 1$ ensures that the function is monotonically increasing. One of the hidden units has a constant bias of -1 while the other has zero bias. The output unit is a linear threshold unit with a constant bias of $-1/N$. All connection weights are equal to 1 except the weight from the first hidden unit to the output unit which is -1. When direct connections between the input layer and the output layer are introduced, it is shown in Brown (1993), that the problem can be solved with a structure that requires only one hidden layer neuron. The activation function used in the hidden layer neuron is the continuously differentiable "step" function

$$f(u) = \lfloor u \rfloor + \sin^K\left(\frac{\pi(u - \lfloor u \rfloor)}{2}\right) \tag{2}$$

where $K > 1$ and $\lfloor\cdot\rfloor$ stands for rounding towards $-\infty$. The implementation of the N-bit parity problem using a different network structure is proposed in Minor (1993). The implementation leads to a neural network with an output $y = u - 2\eta$, where

$$\eta = \sum_{j=1}^{[N/2]} \frac{1}{1 + e^{-40(u-2j+0.15)}} \tag{3}$$

$u = \sum_{i=1}^{N} x_i$ and $[\cdot]$ indicates truncation to the nearest integer. The connection weights from the inputs to u and inputs to the hidden layer neurons are equal to 1. It is also noted in Minor (1993) that the node u can be replaced by direct connections from the input nodes to the output node.

2. Solutions to the parity problem

Based on the structure given in Fig. 1, several solutions (listed in Table 1) for the 3-bit parity problem are obtained in Hohil, Liu and Smith (1998) using a linear programming technique. The activation function used for the output neuron (φ_1) and the hidden neuron (φ_2) is assumed to be

☆ This work was supported by the National Science Foundation under Grant ECS-9732785.

* Corresponding author. Present address: EECS Department (MC 154), University of Illinois at Chicago, Chicago, IL 60607, USA. Tel.: +1-312-996-3422; fax: +1-312-413-0024.

E-mail address: dliu@ieee.org (D. Liu)

0893-6080/99/$ - see front matter © 1999 Elsevier Science Ltd. All rights reserved.
PII: S0893-6080(99)00069-6

Nomenclature

$\lfloor\cdot\rfloor$ Round towards $-\infty$, e.g. $\lfloor(N + 0.5)/2\rfloor$

the threshold transfer function given by

$$\varphi(u) = \varphi_1(u) = \varphi_2(u) = \begin{cases} 1 & \text{if } u > 0 \\ 0 & \text{if } u \le 0 \end{cases}. \quad (4)$$

Solution no. 1 in the table can be generalized to obtain a general solution to the N-bit parity problem shown in Fig. 2. The result in Fig. 2 indicates that the N-bit parity problem can be solved using $\lfloor N/2\rfloor$ hidden layer units, assuming the threshold activation function for both the output and the hidden neurons.

A careful analysis of the role of the hidden layer neurons in Fig. 2 reveals that they can be further combined into one hidden layer neuron through the use of the transfer function

$$\varphi_2(u) = g(u) = \left\lfloor \frac{u + 0.5}{2} \right\rfloor$$

where $u = \sum_{i=1}^{N} x_i$. For example, when $N = 8$ or 9, the required transfer function g is shown in Fig. 3.

We now briefly comment the similarities and differences between our results and those of Brown (1993), Minor (1993), and Stork and Allen (1992). In Minor (1993), a structure that is functionally equivalent to Fig. 2 is used to solve the N-bit parity problem. The summing node u can be replaced by direct connections between the input and output neurons. The solution requires the same number of hidden layer units depending on the value of N. The main difference is that Minor (1993) used an activation function of the form (3), and the present approach uses the simple threshold function. Note that the summation is multiplied by a connection weight of -2 from each hidden layer neuron. The transfer function in Eq. (3) is similar to the function (1) used for the two hidden layer neurons in Stork and Allen (1992) and to the function (2) used in Brown (1993) in that all three are monotonically increasing functions. The activation functions in Eqs. (1)–(3) all resemble the "staircase" function used in the present work. In addition to this similarity, structures in Brown (1993) and Minor (1993) also allow direct connections between the input layer and the output layer. In cases where practical issues arise concerning hardware realization, we note that the transfer function described in this work is easier to implement.

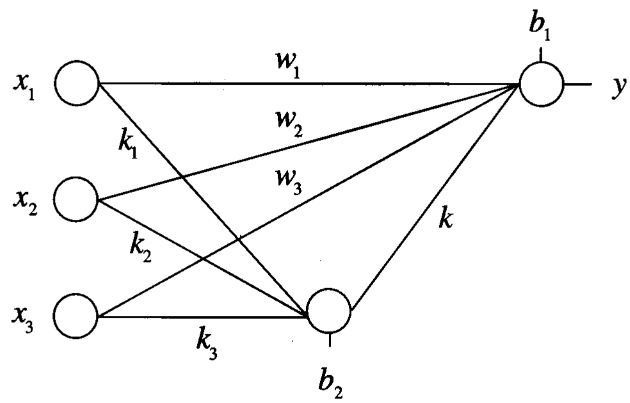

Fig. 1. Neural network structure for the 3-bit parity problem.

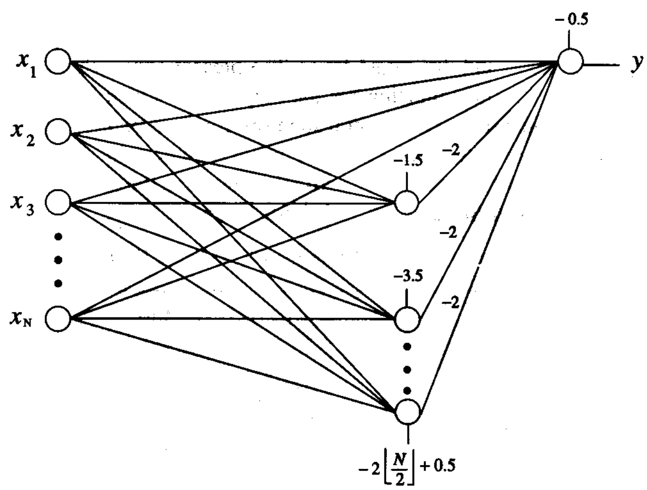

Fig. 2. Structure used to solve the N-bit parity problem.

3. Conclusion

We have shown that when direct connections are allowed between the input and output layers, the N-bit parity problem can be solved using neural networks requiring $\lfloor N/2\rfloor$ hidden layer neurons, where the simple threshold activation function is used in the output and hidden layer neurons. It is demonstrated that through the choice of a "staircase" type activation function, the $\lfloor N/2\rfloor$ hidden layer neurons can be further combined into a single hidden layer neuron. Our solution to the N-bit parity problem is generalized from one of the solutions obtained earlier for the 3-bit parity problem using linear programming. The present

Table 1
Solutions for the 3-bit parity problem

	w_1	w_2	w_3	b_1	k	k_1	k_2	k_3	b_2
1	1	1	1	−0.5	−2	1	1	1	−1.5
2	1	−1	−1	1.5	−2	1	−1	−1	0.5
3	−1	1	−1	1.5	−2	−1	1	−1	0.5
4	−1	−1	1	1.5	−2	−1	−1	1	0.5

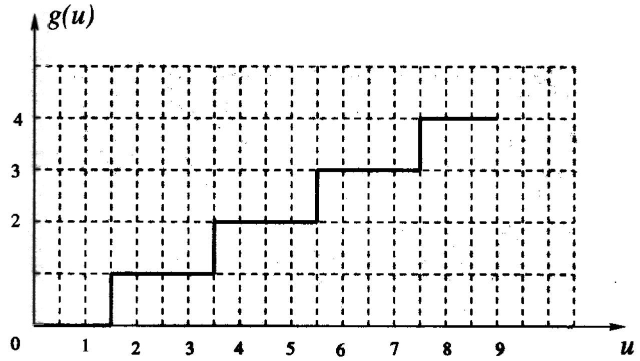

Fig. 3. Transfer function for the single hidden layer neuron when $N = 8, 9$.

approach is constructive and it requires no training and adaptation. In cases where practical issues arise concerning hardware realization, we note that the transfer function described in this work is easier to implement. We also mention that the linear programming technique developed in Hohil et al. (1998) can be used for the constructive design of neural networks implementing binary mappings such as the N-bit parity problem.

References

Brown, D. A. (1993). *N*-bit parity networks. *Neural Networks*, *6*, 607–608.

Hohil, M.E., Liu, D., & Smith, S.H. (1998). *N*-bit parity networks: a simple solution. *Proceedings of the 32nd Annual Conference on Information Sciences and Systems*, Princeton University, Princeton, NJ, March 1998 (pp. 695–700).

Minor, J. M. (1993). Parity with two layer feedforward nets. *Neural Networks*, *6*, 705–707.

Stork, D. G., & Allen, J. D. (1992). How to solve the *N*-bit parity problem with two hidden units. *Neural Networks*, *5*, 923–926.