Invited Paper

OPTICAL MULTILAYER NEURAL NETWORKS

Demetri Psaltis and Yong Qiao

California Institute of Technology
Department of Electrical Engineering
Pasadena, CA 91125

ABSTRACT

In order to implement fully adaptive optical multilayer neural networks, a number of issues involving both the learning algorithms and the device technologies need to be addressed. In this paper, we present some important modifications to existing learning algorithms that serve to simplify optical architectures and allow the use of simple optical devices.

1. INTRODUCTION

Feedforward multilayer neural networks represent a class of powerful learning systems, manifested by the fact that such networks with as few as one hidden layer are capable of approximating any measurable function to any desired degree of accuracy.[1] It is well known that optics is particularly applicable to the implementation of multilayer neural networks because it can provide a large number of neural interconnections relatively simply.[2] The limitations of the current optics and electro-optics technologies, however, may not allow or favor optical implementation of arbitrary multilayer neural learning algorithms. For example, although a holography-based optical implementation scheme[3] exists for the well known Backward Error Propagation (BEP) multilayer learning algorithm,[4] quite a few problems are yet to be solved before a practical implementation becomes possible. These problems, involving both learning algorithms and device technologies, include the realization of backward propagation of error signals through the network, the squaring effect of detectors and spatial light modulators (SLMs) that simulate neurons, and the decay of dynamic holograms used to implement synapses. In this paper, we will focus on the algorithmic aspects of optical multilayer neural networks. Specifically, we will discuss multilayer learning algorithms and their modifications that are suitable for optical implementation.

Partially adaptive multilayer networks and related learning algorithms are relatively easier to implement. By definition, only some of the interconnections of such networks are adaptive or fully trained. A typical example is a fixed feature-extracting layer followed by an adaptive layer trained by the Perceptron[5] or Widrow-Hoff Madaline rule[6]. An even simpler example is Kanerva's Sparse, Distributed Memory (SDM)[7], which consists of a fixed, random interconnection first layer cascaded with a second layer trained by the sum-of-outer-products rule. This network was recently implemented optically and trained for handwritten character recognition.[8]

Since the degrees of freedom offered by multilayer networks are not fully utilized when they are partially trained, usually the performance of such networks is relatively poor. This calls for the use of fully adaptive multilayer networks, trained by algorithms such as BEP. In this case, the hardware implementation is complicated by the need to realize error backpropagation through the network. To overcome this problem, we introduce here an anti-Hebbian local learning rule for two-layer networks. With this rule, weight updates for a certain layer depend only on the inputs and outputs of that layer and a global, scalar error signal. We show that this learning procedure still guarantees that the network is trained by energy descent. The fact that error signals need not back-propagate through the network makes this local rule easy to implement either optically or electronically.

Another problem that exists in the optical implementation of neural networks is that the available optical devices for simulating neurons are usually square-law detectors (i.e., they detect optical intensity, which is the square of the optical signal field). This means that the sign of the signal, which is crucial for

0-8194-0692-9/91/$4.00

learning algorithms such as BEP and Perceptron, cannot be detected. In order to allow the use of simple square-law detectors in optical neural networks, we add a coherent bias to the optical signal arriving at the detector. This method can be straightforwardly combined with the original local learning rule to yield a local learning rule for training multilayer networks with square-law neurons.

Section 2 describes the local learning rule for fully adaptive multilayer networks. Sections 3 presents the method for training networks with square-law neurons.

2. ANTI – HEBBIAN LOCAL LEARNING RULE

We will describe the local learning rule with a feedforward two-layer network shown in Fig. 1. The numbers of neurons for the input, first and second layers, are N_0, N_1 and N_2, respectively. The presynaptic inputs to the nth layer are:

$$s_j^{(n)} = \sum_{i=1}^{N_{n-1}} w_{ji}^{(n)} o_i^{(n-1)}, \tag{1}$$

where $w_{ji}^{(n)}$ is the weight of interconnection between the jth neuron in the nth layer and the ith neuron in the previous layer, and $o_i^{(n)}$ is the output of the ith neuron in the nth layer ($o_i^{(0)}$ being that of the ith input neuron). For the ith input neuron, $o_i^{(0)} = i_i$, where i_i is the input signal. The first and second layers of neurons perform a soft thresholding operation on the presynaptic inputs, forming the outputs:

$$o_j^{(n)} = f[s_j^{(n)}], \tag{2}$$

where the f-function is chosen as $f[x] = tanh[x]$.

The desired response for the input i_i, presented at the input of the network in a certain machine cycle, is given by a target vector t_k, which we take to be binary $\{1, -1\}$. The logarithmic energy function measuring the network output error is defined as:

$$E = \sum_{k=1}^{N_2} \{(1+t_k) ln \frac{1+t_k}{1+o_k^{(2)}} + (1-t_k) ln \frac{1-t_k}{1-o_k^{(2)}}\}. \tag{3}$$

It reaches its minimal value of zero only when the network output is the same as the desired response. We chose this form of error measure instead of the quadratic error function because we found that for our learning procedure this energy function gave better performance.

The BEP rule changes the weights via gradient descent, i.e.,

$$\Delta w_{kj}^{(2)} \propto -\frac{\partial E}{\partial w_{kj}^{(2)}} = 2\delta_k o_j^{(1)}, \tag{4}$$

$$\Delta w_{ji}^{(1)} \propto -\frac{\partial E}{\partial w_{ji}^{(1)}} = 2(1 - o_j^{(1)^2}) o_i^{(0)} \sum_{k=1}^{N_2} \delta_k w_{kj}^{(2)}, \tag{5}$$

where $\delta_k = t_k - o_k^{(2)}$ is the output error signal.

Since the non-local nature of the BEP algorithm is due to the $\sum_{k=1}^{N_2} \delta_k w_{kj}^{(2)}$ factor in Eq. (5), we can simply approximate it with something more accessible. This idea leads to the following local learning rule for the first layer:

$$\Delta w_{ji}^{(1)} \propto \gamma \frac{o_j^{(1)}}{(1 - o_j^{(1)^2})} o_i^{(0)}, \tag{6}$$

where $\gamma = \sum_{k=1}^{N_2} \delta_k s_k^{(2)}$. This rule implies that the weight update for the first layer depends only on the input and output of that layer and a global, scalar error signal γ which can be easily evaluated at the output

stage. Error backpropagation through the network is no longer needed; therefore it is a local rule. The learning rule for the second layer can just follow gradient descent since it is already a local rule.

This new local learning rule is obviously not a gradient descent rule any more. It is, however, still an energy descent rule. Using Eqs. (5) and (6), and assuming that the weights of interconnections between any input neuron and all hidden neurons are updated simultaneously (true in most practical situations), we obtain:

$$\Delta E = \sum_{j=1}^{N_1} \frac{\partial E}{\partial w_{ji}^{(1)}} \Delta w_{ji}^{(1)} \propto -\gamma^2 o_i^{(0)^2} \leq 0, \tag{7}$$

which proves our claim.

Consider a single output neuron, i.e. $N_2 = 1$. If the sign of the network output is different from that of the desired response, then $\gamma < 0$. Since $(1 - o_j^{(1)^2})$ is always positive, we obtain $\Delta w_{ji}^{(1)} \propto -o_j^{(1)} o_i^{(0)}$, which is different from the Hebbian rule in its sign and therefore is called an anti-Hebbian rule. It is understandable because if there is a sign error in the output, it can be corrected by flipping the sign of the internal representation. The anti-Hebbian rule implies exactly that in such cases we should train the first layer with the flipped internal representation as its target. This anti-Hebbian part of the local learning rule is in some respects similar to the Learning by Choice of Internal Representations (CHIR) algorithm developed by Grossman *et al.*,[9] but our rule does not require additional memory to store internal representations of all the training patterns and the computation involved is simple. If the network output and the desired response have the same sign but different magnitude, γ becomes positive and the learning rule for the hidden layer changes to the Hebbian type, which will enhance the internal representation and increase the magnitude of the network output in the right direction. For multiple output neurons, the local rule becomes mixed with Hebbian and anti-Hebbian learning. But the overall effect is still guaranteed to reduce the energy function.

Tests of the anti-Hebbian local learning rule were made on a network of 100 input neurons, 10 hidden neurons and 5 output neurons. The network was trained to perform classification on 5 classes of handwritten digits: **0**, **1**, **2**, **3**, and **4**. Each output neuron responds to only one class. 40 digit patterns, with 8 patterns from each class, were used as training samples. Using the same step size, it took 1,100 learning cycles (presentations of the whole training set) for the BEP algorithm to achieve the desired network response, and 16,000 cycles for the anti-Hebbian local rule. However, taking into account the actual amount of computation involved, the anti-Hebbian rule is only about 5 times slower than the BEP algorithm in terms of computational efficiency. Considering the great hardware implementation advantage offered by the local rule, this cost seems reasonable.

3. MULTILAYER NETWORKS WITH SQUARE – LAW NEURONS

The next issue is the effect of square-law SLMs or detectors, which are often used to simulate neurons in optical neural networks, on learning algorithms. In coherent optical networks, neuron responses are represented by optical amplitudes. So are the weighted sums of neuron responses. However, the neuron-simulating optical devices such as liquid crystal light valves (LCLVs) usually detect optical intensity, which is the square of the optical amplitude.[10] This operation loses information about the signs of the weighted sums, and it could cause ambiguity in weight update.

Consider a single layer network with a square-law output neuron and N input neurons as shown in Fig. 2, where $\underline{x}$ and $\underline{w}$ are the bipolar input vector and the bipolar weight vector, respectively. The total signal arriving at the output neuron (i.e., the weighted sum) is

$$s = \underline{w} \cdot \underline{x}. \tag{8}$$

For a square-law neuron, s will be squared to form the signal u :

$$u = s^2, \tag{9}$$

which is then thresholded to produce o, the network output. As can be seen, the sign of s cannot be uniquely determined from u or o.

One solution is to provide a coherent positive bias b to the total signal arriving at each neuron. As long as the weighted sum s is less than b in magnitude, the biased weighted sum $s + b$ is always positive, so that the magnitude and sign of s can be unambiguously determined from u or o. This eliminates the weight update ambiguity due to the squaring operation.

With a positive bias b, the presynaptic inputs to the nth layer of the multilayer network shown in Fig. 1 now become

$$v_j^{(n)} = s_j^{(n)} + b. \tag{10}$$

The first and second layers of square-law neurons perform a soft thresholding operation on the square of the presynaptic inputs, forming bipolar outputs of the neurons:

$$o_j^{(n)} = tanh[\alpha v_j^{(n)^2} - d], \tag{11}$$

where α is a positive scaling constant and d is a bias to make neuron outputs bipolar. The operation described by Eq. (11) can be realized by an LCLV combined with some polarizing optics, in which case the sign of the neuron output is encoded in the phase of the linearly polarized light exiting from the LCLV module. The d parameter is set such that $o_j^{(n)} = 0$ when $s_j^{(n)} = 0$. This gives $d = \alpha b^2$. Then

$$o_j^{(n)} = tanh[\alpha(s_j^{(n)^2} + 2bs_j^{(n)})]. \tag{12}$$

A plot of this input-output relationship of optical square-law neuron is shown in Fig. 3 for $\alpha = 1$ and $b = 2$. As can be seen, if the presynaptic input $s_j^{(n)}$ is restricted between $-b$ and b, the input-output relationship is a *tanh*-like sigmoid function which maps a bipolar input to a thresholded bipolar output.

A local learning rule can be derived for multilayer networks with biased square-law neurons. It is slightly different from its standard form and is given by:

$$\Delta w_{kj}^{(2)} \propto v_k^{(2)} \delta_k o_j^{(1)}, \tag{13}$$

$$\Delta w_{ji}^{(1)} \propto \gamma' \frac{o_j^{(1)}}{v_j^{(1)}(1 - o_j^{(1)^2})} o_i^{(0)}, \tag{14}$$

where $\delta_k = t_k - o_k^{(2)}$ is the output error signal, and $\gamma' = \sum_{k=1}^{N_2} \delta_k v_k^{(2)}(v_k^{(2)} - b)$. As long as $s_j^{(n)} > -b$, $v_j^{(n)}$ remains positive and can be uniquely determined from the neuron output $o_j^{(n)}$. There will be no ambiguity in the direction of weight update. Furthermore, similar to Eq. (7), we can prove that this is again an energy descent rule.

The most crucial requirement for this learning network is that $v_j^{(n)}$ should be always positive, or equivalently, the magnitude of the weighted sum $s_j^{(n)}$ should not exceed b. For this purpose, the magnitudes of interconnection weights must be prevented from becoming too large. This is a typical dynamic range problem and one solution is weight normalization. There exist optical systems that do precisely that[11,12]

4. ACKNOWLEDGEMENTS

This work was supported by DARPA and the Air Force Office of Scientific Research.

5. REFERENCES

1. K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks **2**, 359(1989).
2. D. Psaltis, D. Brady, X. -G. Gu, and S. Lin, "Holography in artificial neural networks," Nature **343**, 325(1990).

3. K. Wagner and D. Psaltis, "Multilayer optical learning networks," Appl. Opt. **26**, 5061(1987).
4. D. E. Rumelhart and J. L. McClelland, eds., *Parallel Distributed Processing, Vol. 1,* MIT Press, Cambridge, Mass, 1986.
5. F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms,* Spartan Books, Washington, D.C., 1962.
6. B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, Madaline, and Backpropagation," IEEE Proc. **78**, 1415(1990).
7. P. Kanerva, "Parallel structures in human and computer memory," in *Neural Networks for Computing,* J. S. Denker, ed., New York: Am. Inst. Phys., 1986, pp. 247-258.
8. D. Psaltis and Y. Qiao, "Optical neural networks," Opt. & Photonics News, Vol. 1, No. 12, 17(1990).
9. T. Grossman, R. Meir, and E. Domany, "Learning by choice of internal representations," Complex Systems **2**, 555(1988).
10. W. P. Bleha *et al.*, "Application of the liquid crystal light valve to real-time optical data processing," Opt. Eng. **17**, 371(1978).
11. D. Brady, K. Hsu and D. Psaltis, "Periodically refreshed multiply exposed photorefractive holograms," Opt. Lett. **15**, 817(1990).
12. Y. Qiao *et al.*, "Phase-locked sustainment of photorefractive holograms using phase conjugation," in *Conference on Lasers and Electro-Optics, 1991* (Opt. Soc. Am., Washington, D.C., 1991), pp. 328-329.

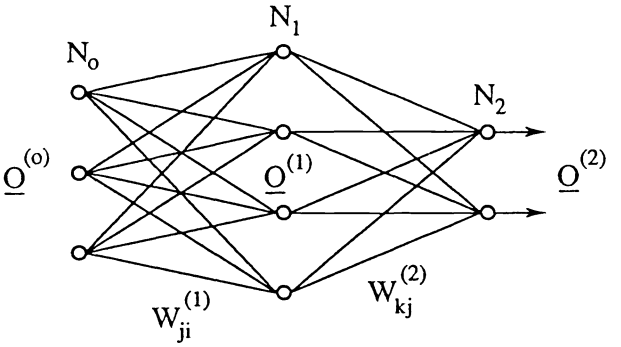

Figure 1. Schematic diagram of a feedforward two-layer neural network.

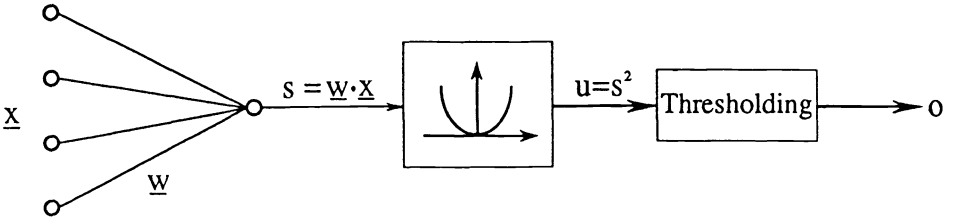

Figure 2. A single layer network with a square-law output neuron.

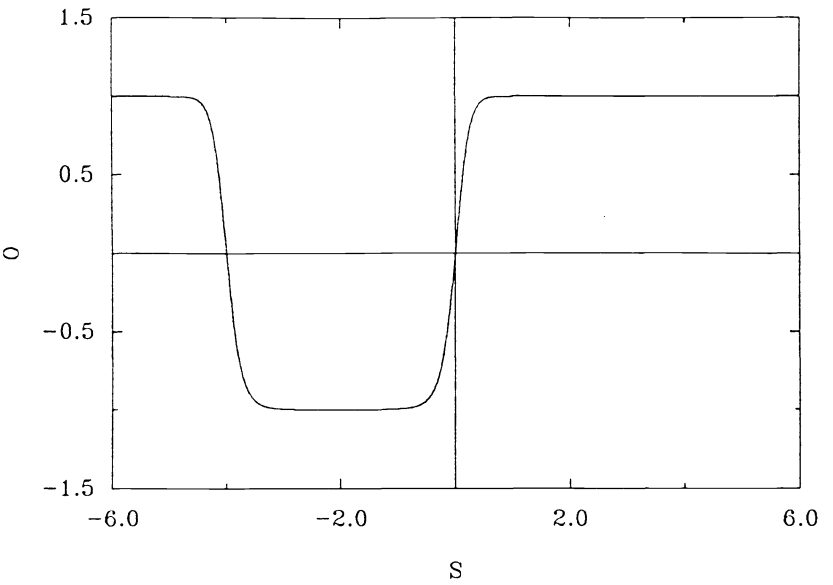

Figure 3. The input-output relationship of the bipolar optical square-law neuron with a positive bias.