

# Optical-processor architectures for alternating-projection neural networks

R. Jackson Marks II, Les E. Atlas, Seho Oh, and Kwan F. Cheung

*Interactive Systems Design Laboratory, FT-10, University of Washington, Seattle, Washington 98195*

Received July 6, 1987; accepted March 21, 1988

Optical-processor architectures for various forms of the alternating-projection neural network are considered. Required iteration is performed by passive optical feedback. No electronics or slow optics (e.g., phase conjugators) are used in the feedback path. The processor can be taught a new training vector by viewing it only once. If the desired outputs are trained to be either  $\pm 1$ , then the network can be configured to converge in one iteration.

Optical neural network architectures have been proposed by a number of researchers.<sup>1-5</sup> Using the alternating-projection neural network (APNN) algorithm developed elsewhere,<sup>6-9</sup> we propose two architectures for a corresponding optical implementation. (A contrast of APNN performance to other neural network architectures is in Ref. 9.) Required iteration is accomplished by using only guided or free-space propagation. No electronics or slow optics such as phase conjugators are used in the feedback path. Learning can be performed by viewing each training vector only once. The network has been shown to scale well.<sup>6-9</sup> The number of training vectors that can be stored is of the order of the total number of neurons minus the number of floating (or output) neurons. The APNN's storage capacity can be increased by the use of additional neurons in the hidden layer. If the output takes on only values of  $\pm 1$ , then the network can be configured to converge in one iteration.

We begin by presenting a terse overview of the APNN. More-detailed explanations are available elsewhere.<sup>6-9</sup> Consider a set of  $N$  continuous-level library vectors of length  $L > N$ : ( $f_n | 1 \leq n \leq N$ ). We form the library matrix  $\mathbf{F} = [f_1 | f_2 | \dots | f_N]$  and the interconnect matrix  $\mathbf{T} = \mathbf{F}(\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T$ . The  $L$  neurons are divided into two sets: one in which the neural states are known and the remainder in which the states are unknown. Let  $S_k(m)$  denote the state of the  $k$ th neuron at time  $m$ . If the  $k$ th neuron falls into the known category, then its state is clamped to the known value. Otherwise the state remains floating and is equal to the sum of the neural inputs. Assume without loss of generality that the states of neurons 1 through  $P < L$  are known and the remaining  $Q = L - P$  are not. Let  $\mathbf{f}$  denote a library vector of which we know the first  $P$  elements. Define the corresponding node operator,  $\eta$ , on an arbitrary vector  $\mathbf{i}$  by

$$\begin{aligned} \eta \mathbf{i} &= \eta(i_1, i_2, \dots, i_P | i_{P+1} \dots i_L)^T \\ &= (f_1, f_2, \dots, f_P | i_{P+1} \dots i_L)^T, \end{aligned}$$

where  $f_k$  is the  $k$ th element of  $\mathbf{f}$ . Then, in synchronous form, the network performs the operation

$$\mathbf{S}(m+1) = \eta \mathbf{T} \mathbf{S}(m), \quad (1)$$

where  $\mathbf{S}(m)$  is the vector of neural states at time  $m$ . Convergence to  $\mathbf{f}$  is ensured if the first  $P$  rows of  $\mathbf{F}$  form a matrix of full column rank. Subsumed in this is the criterion that  $P \geq N$ .

In a layered neural network the same neurons are always used to stimulate the network, and the same set is always floating. In addition, there can be a hidden layer of neurons whose principal purpose is to increase the storage capacity and classification diversity of the network. Use of a hidden layer also increases the convergence rate of the network and decreases its sensitivity to the inexactitude of analog multiplication.<sup>9</sup>

In order for the APNN to converge to the proper solution, the number of clamped states must equal or exceed the number of stored library vectors. The number of clamped neural states, however, can be artificially increased by using new hidden neurons, the states of which are a function of the known portion of the input library vector. The hidden states can, in general, be any nonlinear combination of the clamped states. A technique commonly used with neural networks is to run a linear combination of the clamped states through a sigmoidal nonlinearity to determine the hidden states. Alternately, products of clamped states could be used.<sup>10</sup> Once established, the hidden states are treated as clamped states in the previously discussed analysis. Although the choice of the nonlinearity does not affect the response of the network to training data, it does affect the manner in which the network generalizes.

A basic architecture for optical implementation of the layered APNN is shown in Fig. 1. The point-source array elements corresponding to the clamped and hidden layers provide the input to a Stanford matrix-vector multiplier (astigmatic focusing optics are not shown).<sup>11</sup> The point source array for the floating layer is used only when one is training the network. The same is true for the detector array at the output. Indeed, the only neurons of interest are the floating ones. The states corresponding to the floating neurons are introduced at the right to a fiber bundle.

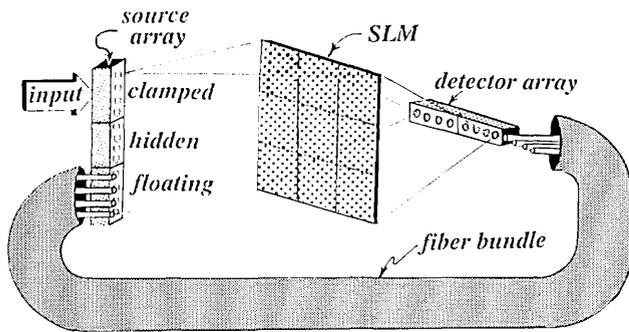


Fig. 1. An architecture for performing a layered APNN. In practice, the architecture requires augmentation as in Ref. 11 to allow for the required bipolar operations. The states of the hidden layers are nonlinear functions of the clamped layers and are generated electronically.

These intensities are fed back to the input as shown, and the process corresponding to Eq. (1) is repeated iteratively until convergence. Alternately, mirrors can be used to provide the feedback.<sup>12</sup>

The astute reader will have noticed two implementation problems associated with the architecture in Fig. 1. First, there is no provision to detect the output. Second, there is no apparent provision for compensating for absorptive, coupling, and other losses in the feedback path. Each of these problems has a straightforward solution. To detect the output, we simply place a highly transmitting pellicle in the feedback path and focus the reflected portion onto a detector array (not shown). This contributes more to the problem concerning losses in the feedback path.

The magnitude of the elements of  $\mathbf{T}$  is generally quite small when  $N \ll L$ . For example, if the library matrix consists of only  $\pm 1$ 's and the library vectors are orthogonal, then the maximum value of the magnitude of the elements in  $\mathbf{T}$  is  $N/L$ . (We would expect such orthogonal library vectors in the statistical sense if all elements of  $\mathbf{F}$  were chosen by a 50-50 coin flip.) When each element of  $\mathbf{T}$  is small, feedback losses can be compensated for by scaling the intensity transmittance up to its maximum passive value of unity. Note that the storage capacity of the network is then in part a function of the ability to minimize feedback losses.

When every neuron in the network can either be clamped or floating, the APNN is said to be homogeneous. In this form the network stimulus can be provided by different neurons from application to application.

An architecture for a homogeneous APNN is shown in Fig. 2. Clamped neural states are provided by the point-source array. The five darkened dots on the array correspond to the five clamped neural states in this example. These provide the input-vector intensities for the portion of the Stanford matrix-vector multiplier corresponding to the upper spatial light modulator (SLM). The light is collected at the output and, as before, is fed back through a fiber-optics bundle to the input. The light from the fiber output is fed through the lower portion of the Stanford matrix-vector multiplier, whose SLM transmittance compen-

sates for feedback loss. The processor output is detected as before.

Note that there should be no input from fibers corresponding to clamped neurons. This can be accomplished with either an optic-optic or electro-optic toggle that turns off the fibers corresponding to the locations of the clamped neurons. Such switches can operate in the gigahertz range with small attenuation.<sup>13</sup>

Note that the feedback is totally optical. In essence, the nonlinearities of the neural network have been placed at the input rather than in the feedback path.

The interconnect matrix equation [Eq. (1)] for most cases is computationally unacceptable. In the spirit of learning, the interconnect matrix can be constructed one vector at a time by using a Gram-Schmidt procedure.<sup>7,8</sup> If, for example, we wish to include a new library vector  $\mathbf{f}$  in an established APNN with an interconnect matrix  $\mathbf{T}$ , the revised interconnect matrix is

$$\mathbf{T}_+ = \mathbf{T} + \mathbf{e}\mathbf{e}^T/(\mathbf{e}^T\mathbf{e}), \quad (2)$$

where  $\mathbf{e} = (\mathbf{I} - \mathbf{T})\mathbf{f}$ .

Consider, then, training the homogeneous APNN in Fig. 2. A new training vector  $\mathbf{f}$  is input on the source array. Since the fibers are turned off, the vector  $\mathbf{T}\mathbf{f}$  will be read by the output detector (not shown). The output is subtracted electronically from the input to give the vector  $\mathbf{e}$ . The SLM is then updated in accordance with Eq. (2).

If the new library vector  $\mathbf{f}$  is a linear combination of the previous library vectors, then  $\mathbf{e}$  will contain all zeros. Owing to the computational inexactitude of analog computations is such a case,  $\mathbf{e}$  will be close to but not exactly equal to the zero vector. This motivates us to compare the energy  $\mathbf{e}^T\mathbf{e}$  to a small threshold before deciding whether Eq. (2) should be applied.

The layered APNN in Fig. 1 can be similarly trained. As with the case above, the input to the system from the fibers is suppressed. The vector  $\mathbf{f}$  is input on the source array. The output is read by the detector array shown, and the detector array is used to read the output in the recall mode (not shown). Updating the SLM transmittance is done as before. The energy of the error corresponding to the output neurons only, however, should be used to determine whether to use Eq. (2).

Note that for both the layered and homogeneous

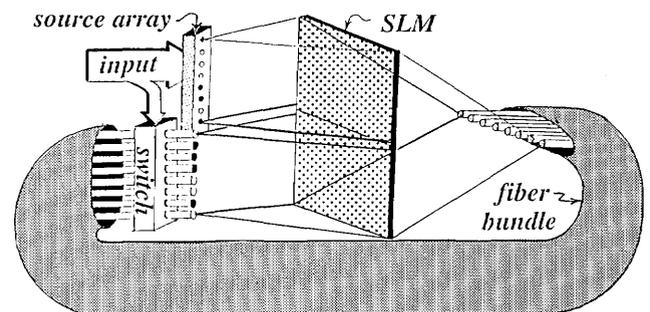


Fig. 2. An architecture for performing a homogeneous APNN.

cases, a significant portion of the SLM is used only in the learning process.

The layered APNN can be configured to converge in a single iteration when the output is allowed to take on values of only  $\pm 1$ . Consider an APNN with a single output neuron that has been trained as either  $\pm 1$ . Then the sign of the output neural state after one iteration is the correct result.<sup>9</sup> Visualize  $Q$  single-output neuron neural networks with input identical to hidden-layer interconnects. Assume that the number of neurons in the hidden layer is sufficient so that the network can be trained with no clamped-to-floating interconnects. Each net trains its output to be either  $\pm 1$  by using different criteria on the same input data. Clearly, the  $Q$  nets can be superimposed into a single net with the common input to hidden interconnects. The hidden neurons are connected to the  $Q$  outputs as they were in the individual net case. The composite net will converge in sign in one iteration. Such performance is clearly desirable since no feedback is required, and, consequently, minor absorptive losses in the optical processor are no longer an issue. Training algorithms and corresponding implementation of such a neural network are now under investigation.

This research was supported by the Strategic Defense Initiative Organization/Innovative Science and Technology through the U.S. Office of Naval Research in conjunction with the Optical Systems Laboratory at Texas Tech University. Support was also provided by the Washington Technology Center at the University of Washington and the Boeing High Technology Center in Bellevue, Washington. The research of L. E.

Atlas was also supported by a National Science Foundation Presidential Young Investigator Award.

## References

1. N. Farhat, D. Psaltis, A. Prata, and E. Paek, *Appl. Opt.* **24**, 1469 (1985).
2. D. Psaltis and N. Farhat, *Opt. Lett.* **10**, 98 (1985).
3. B. Macukow and H. H. Arsenault, *Appl. Opt.* **26**, 924 (1987).
4. N. Farhat, *Opt. Lett.* **12**, 448 (1987).
5. Special issues on optical artificial intelligence and neural networks, *Appl. Opt.* **26**, 1827-1958, 4909-5111, 5189-5190 (1987).
6. R. J. Marks II, *Appl. Opt.* **26**, 2005 (1987).
7. K. F. Cheung, S. Oh, R. J. Marks II, and L. E. Atlas, in *Proceedings of the IEEE First International Conference on Neural Networks* (Institute of Electrical and Electronics Engineers, New York, 1987), p. III-245.
8. R. J. Marks II, L. E. Atlas, and K. F. Cheung, in *Proceedings of the Fourteenth Congress of the International Commission for Optics* (International Commission for Optics Secretariat, Quebec City, Quebec, 1987), pp. 29-30.
9. R. J. Marks II, L. E. Atlas, S. Oh, and J. A. Ritcey, "The performance of convex set projection based neural networks," presented at the IEEE Conference on Neural Information Processing Systems, Denver, Colo., 1987.
10. T. Maxwell, C. L. Giles, and Y. C. Lee, in *Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics* (Institute of Electrical and Electronics Engineers, New York, 1986), p. 627.
11. J. W. Goodman, A. R. Dias, and L. M. Woody, *Opt. Lett.* **2**, 1 (1978).
12. R. J. Marks II, *Appl. Opt.* **19**, 1670 (1980).
13. See, e.g., H. Haga, *IEEE J. Quantum Electron.* **QE-22**, 902 (1986).