USC-SIPI REPORT #176

Neural Networks Based on the Incoherent Optical Neuron Model

by

Chein-Hsun Wang and B. Keith Jenkins

April 1991

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
Powell Hall of Engineering
University Park/MC-0272
Los Angeles, CA 90089 U.S.A.

Contents

Neural Networks based on the Incoherent Optical Neuron Model

Chein-Hsun Wang and B. Keith Jenkins

Signal and Image Processing Institute, Department of Electrical Engineering

University of Southern California, Los Angeles, CA 90089-2564

Abstract

To fully use the advantages of optics in parallel implementations of neural networks, an *incoherent optical neuron* (ION) model can be used to optically implement neurons with both excitatory and inhibitory inputs. The main purpose of this model is to provide for the requisite subtraction of signals at each neuron unit without the phase sensitivity of a fully coherent optical system and without the cumbrance of photon-electron conversion and electronic subtraction. The ION model, in conjunction with coherent or incoherent optical weighted interconnections, can be used to implement arbitrarily connected neural networks. This chapter describes techniques for implementation of both analog and binary inner product neuron units as well as mass action law neuron units. Potential use of the ION model in implementing the neocognitron model, multilayer networks, selective attention for winner-take-all networks, and simple cells of the visual cortex are discussed. In addition, experimental results on the optical implementation of a 2-D array of IONs, and of sequential feature detection operations of the visual cortex, are reviewed.

To be published in Sing H. Lee and C. Lee Giles, Eds., Optical Computing, Vol 1, Academic Press, Boston, 1991.

1 Introduction

Artificial intelligence approaches to problems require intensive computation to search for optimal or suboptimal solutions. Neural networks can potentially provide parallel, real-time solutions. For example, a real-time automatic planning system needs to compute an optimal allocation of currently available resources to respond to real-time dynamics of the environment. Problems of this type belong to the class of non-deterministic polynomial complete (NP complete) problems, which require parallel searching to find even sub-optimal solutions in a reasonable amount of time. Neural networks provide a possible approach for this type of problem. Other applications include machine vision, pattern recognition, speech recognition, and associative memory. These problems use a limited number of samples to train the neural networks. For recognition and classification tasks, the expected result is for the network (after training is completed) to find the nearest class belonging to the unknown inputs.

1.1 The Use of Optics

The most crucial problem in the hardware realization of large-scale neural networks is to implement the massive interconnections between the neurons. As pointed out by Goodman *et. al.*, electronic very large scale integration (VLSI) has certain unfavorable scaling properties in the implementation of massive interconnections [Goodman84]. For example, the planar interconnections can easily consume most of the area of a chip, and propagation delay (measured per unit length) increases as we scale down the device size, so that the delay, even measured per scaled down circuit-to-circuit length, does not decrease as the device size is scaled down. On the other hand, the inherent parallelism and 3-D free-space interconnection capability of optics provide a promising approach for the implementation of large-scale neural networks with higher computational throughput.

In addition to optical interconnection efficiency, the incorporation of *analog* optical signal representation provides additional advantages. Some optical materials, for example photorefractive crystals, can potentially be used to implement massively parallel, adaptive synaptic weights, pro-

viding analog storage as well as rapid analog multiplication. Finally, rapid analog addition of optical signals can be obtained by an appropriately configured optical superposition. Thus analog optical neural networks may play a critical role in the solution of artificial intelligence problems of a size likely to be encountered in realistic situations.

For example, consider a neural computation performed on a digital electronic machine. Figure 1 shows a sample procedure to compute the weighted sum of the membrane potential of a neuron based on a single digital processor. We see that over half of the time (55 %) is spent on moving data and adjusting the pointer. Although pipeline and multiprocessor techniques can be used to speed up the computation, bottlenecks still exist in moving data between memories and registers. For a neural model comprising N fully connected neurons, $O(N^2)$ multiplications and summations are required. If N processors are used so that each processor corresponds to one neuron, the computation time is $O(N)$ at best. A fully parallel analog system can do this in O(1) time. In the electronic case the partitioning of the computation in hardware also causes limitations. This results in a trade-off that depends on the computation overhead, hardware complexity, power dissipation and speed up. Because of these factors, it is pertinent to explore the use of analog optics, which has the potential of overcoming these scale-up problems. Fortunately, analog optical processing accuracy is acceptable in many neural network applications. In addition, for pattern recognition and machine vision tasks, the inputs are light intensity. Optical neural networks can potentially process these inputs directly without any serial electronic conversion, increasing the likelihood of a fast, efficient system.

1.2 Electronic, Coherent Optical, and Incoherent Optical Neurons

There are four main arithmetic operations in a conventional neural network: multiplication, addition, subtraction and nonlinear thresholding. Optics can provide analog multiplication and addition in real time, while nonlinear thresholding can be performed by an optical modulator. Implementation of subtraction in an optical neural network is a key issue. Coherent and incoherent techniques for subtraction differ markedly.

Psaltis and Farhat [Farhat85] used a hybrid electro-optical scheme to implement a Hopfield net with electronics performing the subtraction. In many such approaches, the use of electronics to provide subtraction can create bottlenecks that slow down the speed of the system. For example, if the signals are electronically routed off of a 2-D parallel detector to a parallel subtractor, and then to a 2-D modulator array, this will require either: (1) time serial multiplexing of the signals, which causes a bottleneck, or (2) a massive number of parallel electronic wires bonded over a 2-D array, which will likely have substantial reliability, crosstalk, and signal delay problems. Such disadvantages could in principle be avoided by using a 2-D optoelectronic spatial light modulator (SLM) with integrated detectors, electronics, and modulators, keeping all electronic communication local. But such a technology is currently still in the basic research stage, and it will likely be some time before it is realizable in a full-scale 2-D array appropriate for neural networks implementations. Thus, it is pertinent to develop a model for all-optical neural networks.

A fully coherent optical system can subtract signals directly, using differences in the phase, or path length, of the optical beams. An example of such a system is in [Anderson87]. The subtraction in this type of system is very efficient; the tradeoffs are that the system must keep relative path lengths stable to within much less than one wavelength, and (assuming the signal levels are represented by electric field amplitude) the detection process yields the *square* of the weighted sum as the neuron unit input [Jenkins91]. In addition, the phases of components, including sources or modulators, in the system must be accurately controlled.

An incoherent system is more robust in terms of stability, position accuracy requirements, and noise immunity. Signals are typically encoded as light intensities; this provides real, nonnegative quantities which must at some point be subtracted. Examples of previous incoherent optical neural networks include [Farhat85,Shariv89]. Recently, Shariv and Friesem [Shariv89] demonstrated an all optical neural network with only inhibitory neurons for the case of a Hopfield type network.

A variety of techniques for *linear* incoherent optical subtraction have been demonstrated by others [Ebersole75,Marom77]. Many of these techniques result in an absolute value of the difference $| Y - X |$, where Y and X are the image operands. The technique described in [Marom77] uses

a liquid crystal light valve (LCLV) and results in the difference image added to a constant bias image, i.e. $Y - X + B$, where B is a constant bias. The incoherent optical neuron (ION) model modifies and extends this concept to provide directly a *nonlinear* function of a linear subtraction, which includes a threshold below zero and above some user-defined value (per neuron model). In this case, there is no need for an output bias, and in our model there is none; this enables direct cascadability as required in a neural net.

1.3 Paradigm of an Optical Neural Network

Figure 2 shows a paradigm for an optical neural network, which uses incoherent optical neurons combined with optical interconnections. Here, incoherent means the phase of the input light to the neuron unit is not being used for subtraction; the neuron unit outputs may still be coherent light. A hologram or holograms can be used to emulate synaptic weights in the optical neural network. If the network is to be adaptive, the interconnections should be dynamic during the training phase. Psaltis et. al. [Psaltis88a,88b] , for example, have discussed several learning and recalling issues in photorefractive crystals. As shown in Fig. 2, the incoherent optical neuron units process the weighted sums from the interconnection network; they may also serve as an input transducer. In the case of an adaptive network, the inputs of the neurons are also fed to the interconnection network to form correlations with the outputs of the neuron units during the learning phase. Then the modified interconnection strength is stored in the interconnection network.

In many neural net models, the inputs to a neuron unit can be excitatory and inhibitory, but the output of a neuron unit is always nonnegative. Our incoherent optical neuron (ION) model, described below, applies directly to such *unipolar* neuron units. In some neural net models, the output of a neuron unit can take on both positive and negative values; such *bipolar* neuron units can be implemented using a complementary ION model, also described below. Both unipolar and bipolar neuron units typically function with analog inputs and outputs; in some neural net models they output only binary values (e.g., 0 and 1 for the unipolar case and -1 and 1 for the bipolar case). The ION model can implement both binary and analog neuron units.

2 Incoherent Subtraction Techniques for Optical Neural Networks

Techniques for *linear* incoherent optical subtraction have been presented in the literature, and were briefly discussed above in section 1.2. Here we focus on incoherent techniques applied to neural networks, in which case the requisite nonlinearity, varied connectivity and direct cascadability impact the subtraction method.

There are a variety of possible techniques for incoherent subtraction in optical neural networks. Some are straight forward, and some developed by others are less straight forward and have fewer disadvantages. However, they all have at least one out of several possible significant drawbacks that will preclude their use in general optical neural networks. These drawbacks include a buildup of bias, lack of cascadability, a model-induced threshold that varies from neuron unit to neuron unit, a bias that depends on the weights connected to a neuron unit, and a restriction on connectivity such as applying only to fully connected networks.

We model the recall or computation process of a single layer of a neural network as

$$\hat{V}_i = \psi[\sum_{j=1}^{N} W_{ij} V_j] \tag{1}$$

where $\hat{V}_i$ is the output of neuron i, V_j are the signal inputs, W_{ij} is the synaptic weight from neuron j to neuron i, and $\psi(\cdot)$ is the output nonlinear function of the neuron, which is a nondecreasing function of the neuron inputs and has a finite range. Generally, W_{ij} can be positive, zero, or negative; $\hat{V}_i$ and V_j are nonnegative in many neuron models but can take on negative values in other models.

Conceptually, the interpretation of an inhibitory signal can be either positive weight / negative signal or negative weight / positive signal. Of course, a bias can be added to either the input signals or the weights, yielding subtraction in a straightforward manner. First, a weight-bias method uses negative weights and positive outputs to code the inhibitory signals. The output of the neuron i is

$$\begin{aligned} \hat{V}_i &= \psi[\sum_{j=1}^{N}(W_{ij}+W_b)V_j] \\ &= \psi[\sum_{j=1}^{N}W_{ij}V_j + W_b\sum_{j=1}^{N}V_j] \end{aligned} \tag{2}$$

where W_{ij} represents the connection strength from the j-th neuron to the i-th neuron, which can be either positive or negative and is normalized to be between -0.5 and 0.5. When W_{ij} is positive, it represents an excitatory connection; when it is negative, it is inhibitory and is subtracted from the excitatory inputs. W_b is the bias in weight, usually 0.5, and V_j is the output of the j-th neuron, which is positive and is between 0 and 1. The nonnegative values $(W_{ij} + W_b)$ and V_j can then be implemented using incoherent optics. The second term $W_b \sum_{j=1}^{N} V_j$ is an *input dependent* bias, which is impractical for realization due to the required dynamic threshold of the neurons.

Alternatively, a second technique biases the input signals and uses positive weights and negative inputs for inhibitory signals, and can be written as

$$\begin{aligned} \hat{V}_i &= \psi[\sum_{j=1}^{N}W_{ij}(V_j+V_b)] \\ &= \psi[\sum_{j=1}^{N}W_{ij}V_j + V_b\sum_{j=1}^{N}W_{ij}]. \end{aligned} \tag{3}$$

In this case the nonnegative quantities W_{ij} and (V_j+V_b) are represented physically using incoherent optics. The second term $V_b \sum_{j=1}^{N} W_{ij}$ is a weight dependent bias term. Since the weights of a neural network are typically changed from time to time to adapt to their environment by learning, this term is difficult to implement. We need to calculate the sum of weights into each neuron, which increases the implementation complexity, particularly if volume holograms are used. Nevertheless, this input signal bias method can potentially be used in the special case of neural networks that assume conservation of the sum of weights into a neuron. Such networks have been described, for example, by von der Malsburg [Malsburg73]. In this case the second term, $V_b \sum_{j=1}^{N} W_{ij}$, is a constant and can be treated as a fixed threshold of the optical neuron.

A similar technique called bias-subtraction has been discussed by Gmitro and Gindi [Gmitro87]. In their approach, the output of the neuron is positive and the weight can be either positive or negative to represent an excitatory or inhibitory connection efficiency. During implementation, two channels are used to process positive and negative weights separately; the negative channel is implemented by positive weights and complementary input signals, which are also positive:

$$\begin{aligned} \hat{V}_i &= \psi[\sum_k W_{ik}V_k + \sum_j W_{ij}(1 - V_j)] \\ &= \psi[\sum_k W_{ik}V_k - \sum_j W_{ij}V_j + \sum_j W_{ij}] \end{aligned} \tag{4}$$

where W_{ik} and W_{ij} are the weights to the positive (excitatory) and negative (inhibitory) inputs of the i-th neuron. They are positive quantities during optical implementation and are represented by the transmittance of a mask or diffraction efficiency of a hologram. The bias term, $\sum_j W_{ij}$, must be canceled out by adjusting the threshold of each neuron. This increases implementation complexity.

Another technique, proposed by Te Kolste and Guest [Te Kolste87], is a hybrid of the previous two:

$$\begin{aligned} \hat{V}_i &= \psi[\sum_{j=1}^{N}(W_{ij} + \frac{1}{2})V_j + \sum_{j=1}^{N}\frac{1}{2}(1 - V_j)] \\ &= \psi[\sum_{j=1}^{N} W_{ij}V_j + \frac{N}{2}] \end{aligned} \tag{5}$$

The bias term $N/2$ is independent of input as well as weight, which simplifies the requisite hardware considerably. This approach applies to unipolar neurons and fully connected networks only.

3 Overview of the Incoherent Optical Neuron (ION) Model

In this section, we describe the incoherent optical neuron (ION) model [Jenkins87,88,Wang88,89a,89b,90]. It is a generic model that can be used in a variety of neural networks. A device requirement analysis

and a variant of this model that incorporates bipolar neuron outputs is also discussed. Section 5 will give several examples of applications of the ION model to different neural networks.

3.1 The ION Model

From a biological point of view, the inhibitory site of a neuron often utilizes a distinct mechanism as compared to the excitatory site, e.g. neurotransmitters vs. chemical-selected receptors [Shepherd78,Stevens79,Wang-Freeman87]. The inhibitory site then accepts a positive input to produce a negative effect on the membrane of the neuron. The ION model follows this approach; it uses spatially and physically distinct control mechanisms to emulate the excitatory and inhibitory signal processing in a biological neuron.

The ION comprises two elements: an inhibitory (I) element and a nonlinear output (N) element. The inhibitory element provides an inversion of the sum of the inhibitory signals; the nonlinear element operates on the excitatory signals, the inhibitory element output, and an optical bias to produce the output. The inhibitory element is linear; the nonlinear threshold of the neuron is provided entirely by the nonlinear output device. Figure 3(a,c) show the characteristic curves of the I and N elements respectively. The structure of the ION model is illustrated in Fig. 3(d). The output of the normalized I element is given by

$$I_{out}^{(I)} = 1 - I_{inh} \tag{6}$$

and of the N element is

$$I_{out}^{(N)} = \psi(I_{out}^{(I)} + I_{exc} + I_{bias} - \alpha) \tag{7}$$

where I_{inh} and I_{exc} represent the total (weighted) inhibitory and excitatory neuron unit inputs respectively, and include any lateral feedback signals as well as inputs from other layers; α is the nominal device threshold of the N element. I_{bias} is the bias term for the N element, which can be varied to change the threshold, and α is the offset of the characteristic curve of the N element. $\psi(\cdot)$

denotes the nonlinear output function of the neuron. If we choose I_{bias} to be $\alpha - 1$, the output of the N element is

$$I_{out}^{(N)} = \psi(I_{exc} - I_{inh}) \tag{8}$$

which is the desired subtraction.

The features of the ION model include a bias that is essentially independent of input weight and signals, a dynamically and globally variable threshold, the capability of implementing a sigmoid or binary threshold function for different neuron models, cascadability and ease of implementation. Due to the separation of the control mechanisms in the ION model, it can implement more complex neuronal functions. For example, it can implement global inhibition by using the output of one inhibitory element to control the reading beam intensity of many excitatory elements. Distinct from Gmitro and Gindi's bias subtraction technique, this model inverts only the sum of inhibitory inputs to a neuron.

3.2 Analysis of the ION Model

Many factors impact the operation of ION. The incorporation of an unnormalized I element, positive and negative neuron thresholds, and a single device for implementation of both I and N elements is discussed below. Device requirements dictated by the ION model, and limitations of fan-in and fan-out due to device and system parameters are also given. The effects of device imperfections such as nonuniformities, drift, and undesired nonlinearities have been analyzed via simulation of ION in a neural network; results are summarized in [Wang88,Wang90].

In general, the I element will not be normalized (Fig 3(b)). In this case the offset and slope of its response can be adjusted using I_{bias} and an attenuating element (ND filter), respectively, again enabling proper subtraction. More quantitatively, the characteristic curve of the unnormalized I and N elements (Fig. 3(b,c)) can be modeled as

$$
\begin{aligned}
I_{out}^{(I)} &= -\frac{b_1}{a_1} I_{inh} + b_1 \quad \text{for } 0 \leq I_{inh} \leq a_1 \\
I_{out}^{(N)} &= \psi(I_{in}^{(N)} - \alpha) \quad \text{for } \alpha \leq I_{in}
\end{aligned}
\tag{9}
$$

where $I_{in}^{(N)}$ denotes the sum of all inputs to the N element, and a_1 and b_1 are the maximum input and output of the I element, respectively. The output of the I element is attenuated by an ND filter (Fig. 3(d)); the intensity transmittance of the attenuator is equal to the magnitude of the slope of the characteristic curve of the I element. Of course, the unnormalized I element must have gain greater than or equal to 1. In this case, the input to the N element becomes

$$
I_{in}^{(N)} = \frac{a_1}{b_1} I_{out}^{(I)} + I_{exc} + I_{bias} \tag{10}
$$

and the bias, I_{bias}, is now set to $\alpha - a_1$. Upon substitution into the second Eqns. (9), this again yields exactly the desired subtraction given by Eqn. (8).

The ION model can be implemented using separate devices for the I and N elements (heterogeneous case), or by using a single device with a nonmonotonic response to implement both elements (homogeneous case). Possible devices include bistable optical arrays [Lentine88,Miller85,Walker86] and spatial light modulator (SLMs) such as liquid crystal light valves (LCLVs) [Bleha78,Jenkins84a]. A single Hughes liquid crystal light valve can be used to implement both elements (Fig. 4). A positive neuron threshold (θ) can be implemented in ION by decreasing the bias by the same amount θ to $I_{bias}^{(+)}$ (Fig. 4). Similarly, a negative threshold is realized by increasing the bias by θ to $I_{bias}^{(-)}$.

A detailed analysis for implementing the ION model is given in [Wang90], which describes the device requirements and threshold implementation as well as constraints on fan-in, fan-out and gain. Here we summarize the results in the following.

Device Requirements

To guarantee the proper operation of the ION model, the input signals and the device characteristics must satisfy the following inequalities:

$$0 \leq I_{inh} \leq a_1 \tag{11}$$

$$0 \leq I_{exc} \leq a_1 \tag{12}$$

$$\alpha \geq a_1 \quad \text{(heterogeneous case)} \tag{13}$$

$$\alpha \geq a_1 + a_2 \quad \text{(homogeneous case)} \tag{14}$$

where a_1 is the maximum input of the I and N element, and a_2 is defined as an I element input that corresponds to the residual neuron output, I_r (*cf.* Fig. 3(c) and Fig. 4).

Threshold Implementation

For a positive neuron threshold θ, the constraints (13) and (14) are modified to become

$$\alpha \geq a_1 + \theta \quad \text{(heterogeneous case)} \tag{15}$$

$$\alpha \geq a_1 + a_2 + \theta \quad \text{(homogeneous case)} \tag{16}$$

and θ is in the range $0 \leq \theta \leq a_1$.

There are no extra constraints for negative threshold. The requirements (11)-(14) above are sufficient.

Fan-in and Fan-out

If the neural net model requires a maximal change of signal level on any one input line to a neuron to be readily distinguishable (typical of neural nets with small to moderate fan-in), the maximum fan-in for the ION model is limited by the extinction ratio of the device:

$$N_{in} \leq \frac{\Delta I_s}{I_r} \tag{17}$$

where N_{in} is the total fan-in, I_r is the residual output, and ΔI_s is the output difference due to a state change of the N element (*c.f.* Fig. 3(c) and Fig. 4). If instead, the neuron model only requires a maximal change of signal level on a constant fraction β of the input lines to a neuron

unit to be readily distinguishable (typical of neural nets with large fan-in), then β is limited by the extinction ratio of the device

$$\frac{1}{\beta} \leq \frac{\Delta I_s}{I_r} \tag{18}$$

and if this condition is met, the fan-in, *per se*, is not limited by device characteristics.

The fan-out, N_{out}, comes from the N element only, and is bounded by the overall optical system loss, γ_s, and device and model parameters by

$$\frac{N_{out}}{N_{in}^{(exc)}} \leq \gamma_s \bar{w}_{ij} \frac{\Delta I_s}{\Delta a_N} \tag{19}$$

where Δa_N is the differential input required to turn the N element full ON (Fig. 4), $\bar{w}_{ij}$ is the mean value of the weights, and $N_{in}^{(exc)}$ is the fan-in to the excitatory site. Thus, the ratio of the fan-out to the excitatory fan-in is bounded above by a measure of the differential gain of the N element, scaled by loss factors γ_s and $\bar{w}_{ij}$.

The device requirements given by Eqns. (11) and (12) can also be re-written in terms of the fan-out and fan-in, which are more likely to be known for a given neural model:

$$\begin{aligned} \Delta I_s \frac{N_{in}^{(inh)}}{N_{out}} &\leq a_1 \\ \Delta I_s \frac{N_{in}^{(exc)}}{N_{out}} &\leq a_1 \end{aligned} \tag{20}$$

where $N_{in}^{(inh)}$ is the fan-in to the inhibitory site.

3.3 The Complementary ION Model: Incorporation of Bipolar Neuron Outputs

The ION model emulates a biological neuron by using spatial coding for the sign of the *input* signals. In order to accommodate some of the artificial neuron models that use bipolar neuron *outputs*, a variant of the ION model, the complementary ION, can be used which uses complementary inputs and weights. It is given by

$$\frac{1}{2}(1+\hat{V}_i) = \psi[\sum_{j=1}^{N}\frac{(1-W_{ij})}{2}\frac{(1-V_j)}{2}+\sum_{j=1}^{N}\frac{(1+W_{ij})}{2}\frac{(1+V_j)}{2}] \quad (21)$$
$$= \psi[\sum_{j=1}^{N}\frac{W_{ij}V_j}{2}+\frac{N}{2}]$$

The terms $(1-V_j)/2$, $(1+V_j)/2$, $(1-W_{ij})/2$ and $(1+W_{ij})/2$ are positive. The $(1+V_i)/2$ and $(1-V_i)/2$ terms can be generated in some complementary devices by using orthogonal polarizations (e.g. LCLV) or by reflected and transmitted beams (e.g. some bistable optical devices); this makes the model directly cascadable. In this case, the input to the net must be inverted, and the signal and its complement are preserved throughout the network. No other I elements are necessary. If such complementary devices are not available, the neuron unit input and output can be left in the form $(1+V_i)/2$. Then an I element can be used at each neuron unit to generate $(1-V_i)/2$ from $(1+V_i)/2$.

For example, an Amari (or Hopfield) net [Amari72,Hopfield82] uses binary bipolar neurons, and its retrieval operation is given by

$$\hat{V}_i = \Phi[\sum_{j=1}^{N} W_{ij}V_j] \quad (22)$$

where $\hat{V}_i \in \{+1,-1\}$ is the output state and $W_{ij} \in [-1,1]$ is the normalized weight from neuron j to neuron i. The nonlinear output function $\Phi(x)$ is equal to 1 for $x \geq 0$, otherwise it is -1. The net is fully connected. Our complementary model can implement this directly, as shown in Fig. 5.

In this complementary ION model, there is no spatial distinction between excitatory and inhibitory channels. Restricting the model to fully connected networks (as in Eq.(21)) keeps the threshold neuron-independent; this yields the simplest hardware implementation. Alternatively, by summing in Eq. (21) only over the inputs to each neuron unit, a partially connected network can be implemented, at the expense of an increase in hardware complexity due to the neuron-dependent threshold.

4 Demonstration of ION Array

To demonstrate the concept of the ION model, a single liquid crystal light valve (LCLV) is used in a homogeneous experimental implementation of the I and N elements (cf. Fig. 4). In this demonstration, a 2-D array of IONs effectively subtracts two spatial patterns, by sending one spatial pattern to an array of inhibitory inputs, and the other spatial pattern to an array of excitatory inputs. The two arrays are implemented as spatially distinct regions within the active area of the same LCLV.

Figure 6 shows the experimental setup for implementing and testing an array of ION's. Three input beams are used to provide N element bias, I element inputs and N element inputs, which are controlled by polarizer pairs P1, P2 and P3 respectively. The I element input path, SI-BS5-BS7-L2-BS8-LCLV, is imaging with magnification factor of 0.8. The same magnification factor is applied to the N element input path, SN-BS7-L2-BS8-LCLV. Two feedback paths are implemented, one for the I to N connection, which is BS9-BS10-L3-BS11-L5-BS12-BS8-LCLV. The other feedback path through mirror M5, M6 is for the N to N self-feedback connection. Each feedback path images from the LCLV output plane to the LCLV input plane; the I to N feedback path also shifts the image to the N element input. Mask MK2 is used to block the bias beam to the I elements. Masks MK3 and MK4 block the N and I element outputs in the I to N and N to N feedback paths, respectively.

Figure 7 (a) and (b) give the input/output characteristics of the I and N element for an applied voltage of 5.0 volts rms at a frequency of 1.5 Khz. In our case the Hughes LCLV used has a twisted nematic liquid crystal and a CdS photoconductor. Self-feedback for the N element is necessary to fulfill the constraints of the ION model.

To demonstrate a 2-D array of neuron units, a spatially continuous case was implemented, i.e. no isolation between neuron units. Figure 8(a) and (b) show an experimental result for binary subtraction. Two character sets, each 6 mm square, are chosen for the N (left side) and I (right side) inputs (Fig. 8(a)). All four possible cases are included (corresponding to 1 or 0 for the N element input, and 1 or 0 for the I element input). A bias is added to the N element inputs as

described in Sec. 3.1. Figure 8 (b) shows the I element outputs (right side) and the final neuron outputs (left side; these are also the N element outputs). The ideal result is the residual leg of the character "R" (right top) and the full character "T" (right bottom) in the N element area, and agree with Fig. 8(b); these regions correspond to a 1 on the excitatory inputs and a 0 on the inhibitory inputs.

A key advantage of such a homogeneous implementation is the sufficiency of just one active device. In addition, an ultimate system based on a homogeneous implementation would likely be simpler than one based on a heterogeneous implementation, because in the homogeneous case only one set of input interconnections are required, and a single feedback loop for lateral and layer-to-layer interconnections should be sufficient in an optimized, compact system design. The tradeoffs are that the device requirements are more stringent, and that an initial system demonstration is more difficult to implement. In particular, meeting the device requirements outlined in Sec. 3.2 with two LCLVs in a heterogeneous implementation is likely much simpler than meeting the requirements with a single LCLV in a homogeneous implementation.

5 Application Examples

In this section, we will discuss several potential applications of the ION model to neural networks. A physical implementation of the neurons in these networks could be either heterogeneous or homogeneous. First as a conceptually straight forward example, a single layer feedback net based on ION is described; then its use in the implementation of a sample general purpose multilayer network is outlined. Then follow applications using ION for Grossberg's mass action type of neurons. Normalization of inputs is very important in analog pattern recognition; the ION structure can serve in a preprocessing stage for input normalization. Finally, we describe a selective attention network based on the ION model.

As an example of the use of the original ION model in a neural net, a conceptual diagram of an implementation of a single layer feedback net is shown in Fig. 9. It utilizes a single 2-D spatial

light modulator to implement the array of neurons in a homogeneous ION implementation for both I and N elements. The spatial light modulator is spatially divided into two regions, one each for the array of I elements and the array of N elements. The output of the I element array is imaged onto the input of the N element array, after passing through a ND filter as the (uniform) attenuation. A uniform bias beam is also input to the N element array. The N element array output is fed back through an interconnection hologram to the inputs of both I and N element arrays, representing inhibitory and excitatory lateral connections, respectively. The inhibitory and excitatory neural net inputs are also sent to the I and N element array inputs, respectively.

Heretofore our discussion of the ION model has considered the implementation of only single layer networks and only conventional inner-product neurons; inner-product neurons perform a weighted sum of their input signals. The ION concept can also be applied to other types of neuron, for example that based on mass action. These neurons use a mass action law to model the neuron behavior, which tends to cause local competition for the limited membrane sites. The following two sections will briefly discuss two types of multilayer neural network models, those due to Fukushima and Grossberg, and their implementation using ION.

5.1 Multilayer Networks

Cooperative and competitive interactions [Amari82,Grossberg80] are two main neural mechanisms of information processing in the human brain. The macroscopic behavior of a neural network typically exhibits a cooperative property, but at the same time it may locally execute competitive operations. Here we consider networks that can exhibit both of these properties. Structurally, these two mechanisms comprise interactions of excitatory and inhibitory signals through feedforward, lateral, and feedback connections in the neural network.

Competitive neural networks can be used in feature extraction, pattern recognition and associative memory [Carpenter87a,87b,Ellias75,Fukushima75,80,84,86,87,Grossberg73,76a-c,Miyake84]. Fukushima's neocognitron is a multilayer feedforward neural network used for pattern recognition. His more recent models are bidirectional and can serve as feature extractor, pattern recognizer and associa-

tive memory [Fukushima80,84,86,87,Miyake84]. The interaction relationship of his models can be written as

$$V_i = \psi\{ \sum_{j \in A_{pre}} a_{ij} V_j + \sum_{k \in A_{post}} a_{ik} V_k - d_i \sum_{j \in A_{pre}} \tilde{c}_{ij} V_j - b_i \sum_{k \in A_{post}} \tilde{c}_{ik} V_k - \sum_{l \in A_l} \tilde{e}_{il} V_l \} \qquad (23)$$

where tilded weights ($\tilde{c}_{ij}$, $\tilde{c}_{ik}$ and $\tilde{e}_{il}$) are fixed weights with a Gaussian-type distribution with respect to the distance between the current cell and the input cell; the other weights (a_{ij}, a_{ik}, b_i, and d_i) are modifiable subject to winner-take-all learning, i.e. within a specified region only the neuron with the strongest output can modify its weights; A_{pre}, A_{post} and A_l denote the interconnection region from the previous layer, post layer, and lateral inhibition area, respectively; and V_m represents the output signal of neuron unit m. The first two terms in Eq.(23) are excitatory inputs from previous and post layers, and the third and fourth terms are used to provide adaptive level control. The last term in Eq.(23) is the lateral inhibition (i.e. inhibitory connections within the layer). All the weights shown in Eq. (23) are positive. The ION model can implement these by putting the last three terms into the I element, with the first two terms going directly to the N element. Typically, only a subset of these terms are present in any one of Fukushima's models; for example, the cognitron and neocognitron models [Fukushima75,80] use only the first, third, and fifth terms, while his hierarchical associative memory [Fukushima84] uses all but the last term.

5.2 Mass Action Neurons

Another cooperative competitive type of neural network is Grossberg's on-center off-surround enhancement and adaptive resonance theory [Carpenter87a,87b,Ellias75,Grossberg73,76a-c]. Grossberg uses a mass action law in his models, which can be described as

$$\begin{aligned} \dot{x}_i &= -Ax_i + (B - x_i) I_{exc} - x_i I_{inh} \\ I_{exc} &= \sum_{j=1}^{n_1} \psi(x_j) C_{ij} + I_i \end{aligned} \qquad (24)$$

$$I_{inh} = \sum_{j=1}^{n_2} \phi(x_j) D_{ij} + J_i$$

where x_i is the membrane potential of neuron i, and I_{exc} and I_{inh} denote the total excitatory and inhibitory inputs to neuron i. $\psi(x_j)$ and $\phi(x_j)$ are the output of neuron j and of its inhibitory interneuron respectively; ψ and ϕ are sigmoid functions in most cases. I_i and J_i are the total excitatory and inhibitory inputs from other layers, respectively. A and B are the decay constant and maximum membrane potential, respectively. C_{ij} and D_{ij} are the interconnection weights within this layer, and n_1 and n_2 are the total number of excitatory and inhibitory inputs, respectively. The above equation can be grouped into two terms, one term each to be implemented by I and N elements.

For the discrete case, we can rewrite the first of Eqs. (24) as

$$x_i(k+1) = x_i(k)[1 - (A + I_{exc} + I_{inh})] + BI_{exc}. \tag{25}$$

This is a shunting model. The crucial difference from an implementation point of view is the product between the neuron inputs and the neuron potential in the first term of Eq. (25). To implement this term, an I element with adaptive gain is needed; or the I element read beam can be modulated by x_i to provide the multiplication. The second term is the excitatory part, which can be fed to the N element directly. For the "lumped" case of Grossberg's model, there is no delay between the output of the neuron and its interneurons. We assume that the output characteristic functions $\psi(\cdot)$ and $\phi(\cdot)$ are the same to simplify the complexity of the neuron.

Figure 10 shows a conceptual implementation of Grossberg's cooperative competitive network. In the figure, masks MK1 and MK3 are used to provide read beams for the N and I elements, respectively, on LCLV1. Each mask is in an image plane of LCLV1. The read beam of the I element derives from the N element output through mask MK3, BS2, and BS4. The optics also provide a shift so that the N element output is incident on the I element region of the LCLV1 output. The N element response of LCLV1 is linear. The feedback path through mask MK4 provides the requisite I element output to N element input connections, per the conventional ION

model. The LCLV2 provides the nonlinear threshold function of the neuron units.

Figure 10 implements Eqs. (24) and (25) as follows. On the input side of LCLV1, input to the I element array is $(A + I_{exc} + I_{inh})$, and input to the N element array is BI_{exc}. On the output side, the N element outputs the neuron potential, x_j. The path through mask MK3 sends this potential to LCLV2, which provides the neuron output $\psi(x_j)$, as well as the interneuron output $\phi(x_j)$. The lateral connections given by C_{ij} and D_{ij} of Eq. (24) are implemented in interconnection unit H, which in turn sends I_{exc} signals to both I and N element inputs, and sends I_{inh} to only I element inputs, per Eq. (25). The decay constant, A, is sent as an additive bias to the I element input via mask MK2. (The same mask also provides the ION model's bias to the N element input.) External inputs are provided via input device E, with E_1 providing a sum of excitatory and inhibitory inputs, and E_2 providing only excitatory inputs.

The figure can implement a single layer net, or can be viewed as a multi-layer functional net implemented on a single physical layer with feedback [Farhat87a,Jenkins91]. Viewed as a single layer net, inputs from other layers, I_i and J_i, also come from input location E. Viewed as a multiple layer net, connections from other functional layers, I_i and J_i, are also implemented in the interconnection unit H. The most powerful interconnection unit would utilize holograms in a volume medium.

With some re-arranging of Eq. (25), we can show that a key element of its proposed implementation has been successfully demonstrated previously. For the *steady-state* case, the membrane potential (from Eq. (25) or the first of Eqs. (24)) is

$$x_i = \frac{BI_{exc}}{A + I_{exc} + I_{inh}} \tag{26}$$

which is essentially the division of two terms. An optical implementation of pixel-by-pixel division has been described and demonstrated by Efron et. al. [Efron85]; they used the same technique we propose in Fig. 10, that is the use of a positive-going device (linear N element) output as the read beam of a negative-going device (I element), with the numerator input to the N element,

denominator input to the I element, and the I element output being added to the N element input. This circuitry is imbedded into Fig. 10. Efron et. al. ran this feedback system to steady state to achieve the desired division; here we use the same technique more generally to achieve both the desired *transient* behavior as well as the steady-state behavior.

For the unlumped system, which is more common in the biological neural network, the inhibitory signal comes from an interneuron with a different characteristic. The interaction can be formulated as

$$\begin{aligned} \dot{x}_i &= -Ax_i + (B_i - x_i)[\sum_j^{n_1} \psi(x_j)C_{ij} + I_i] - x_i[\sum_j^{n_2} \phi(y_j)D_{ij} + J_i] \\ \dot{y}_i &= -Ey_i + \sum_j^{n3} x_j F_{ij} \end{aligned} \tag{27}$$

where y_i is the activation state (potential) of interneuron i, which has decay constant E and receives excitatory signals from a total of n_3 excitatory neurons; n_1 and n_2 are total number of excitatory and inhibitory inputs, respectively. F_{ij} represents the interconnection strength from excitatory neuron j to interneuron i. D_{ij} is the weight from the output of the interneuron j to its neighboring excitatory neurons i, and C_{ij} is the excitatory weight from neuron j to neuron i. For a full linear neuron with potential $x_i(k)$, we need one I element with adaptive gain and one linear N element. The interneuron, which has potential $y_i(k)$, is implemented by one N element only. As before, for nonlinear neuron outputs, an additional nonlinear N element is incorporated.

5.3 Input Normalization

Normalization of input patterns is often critical for the correct operation of a neural network. Consider the example of a pattern classification task. The inner-product neuron can provide a very efficient means to perform nearest-neighbor classification. Let two stored prototypes be $\underline{x}_1$ and $\underline{x}_2$, and the current input pattern be $\underline{x}$. We assume that the input $\underline{x}$ is closer to prototype $\underline{x}_1$ than prototype $\underline{x}_2$. If the Euclidean distance is used as the classification metric, then

$$| \underline{x} - \underline{x}_1 | < | \underline{x} - \underline{x}_2 | . \tag{28}$$

Assuming all the patterns are normalized, i.e. $\| \underline{x} \| = \| \underline{x}_1 \| = \| \underline{x}_2 \|$, the above inequality can be represented instead using the inner-product operation

$$\underline{x}^T \cdot \underline{x}_1 > \underline{x}^T \cdot \underline{x}_2 \tag{29}$$

which can be much simpler to implement, and maps readily into a neural architecture.

In addition to pattern classification, input normalization is important in other tasks. Typically, information from the physical world is presented to sensors in an analog format. The important features of the input pattern often reside in the ratio between components instead of in their absolute quantity. For example, the cone photo receptors (R,G,B) in our retina perform a *log* operation on input light intensity, and then the ratios R/G and R/B are used to provide information for color vision operations.

A mass action type neuron can be used to perform input normalization as follows. From the second of Eqns. (27), we have

$$\frac{dy_i}{dt} = I_{exc} - y_i I_{inh} \tag{30}$$

where I_{inh} is used in place of the decay constant E. At steady state, $dy_i/dt = 0$, so $y_i = I_{exc}/I_{inh}$. Let $I_{exc} = x_i$ be the external input, and $I_{inh} = \sqrt{\sum_{i=1}^{N} x_i^2}$. Then $y_i = x_i/\sqrt{\sum_{i=1}^{N} x_i^2}$ is the normalized output of the i^{th} component of input pattern $\underline{x}$.

Figure 11 shows an ION structure that performs input normalization. An input, x_i, that could be (for example) one component of an input pattern, is sent to 3 linear N elements, N_1, N_2, and N_3. The output of N_2 is used as the read beam of N_1, providing a multiplication $x_i \cdot x_i$, which is sent to nonlinear N element N_4; N_4 receives as its input the sum over x_i^2, and its nonlinearity provides an approximate square root. The I element and linear element N_3 provide the mass action neuron response given by Eqn. (30), using techniques described above. To get an accurate

normalization is very difficult in the neural network due to the inherent moderate accuracy of neural networks. Nevertheless, the neural network still can perform pattern classification very well when input normalization is combined with a classification technique such as selective attention [Fukushima86].

If we let the total inhibitory input to a mass action neuron be the norm of the input vector, the neuron will in effect perform an automatic gain control operation on its inputs. If we set B (maximum potential output) of Eq. (26) to 1 and assume the leakage (A) to be very small so that it can be neglected, then Eq. (26) can be rewritten as

$$x_i = \frac{1}{1 + \frac{I_{inh}}{I_{exc}}} \tag{31}$$

showing that under these conditions the neuron responds only to the *ratio* of its inhibitory and excitatory inputs. Figure 12 shows the steady-state potential output of a mass action neuron vs. total excitatory to inhibitory input ratio for the case of very small leakage ($A = 0.01$). Thus the potential of a mass action neuron is dependent on the ratio of its excitatory component to its inhibitory component rather than on their absolute value or difference. This is a very important feature for analog pattern classification. Usually such input normalization is applied as a preprocessing stage of a network, as in, for example, Adaptive Resonance Theory (ART) II network [Carpenter87b].

5.4 Selective Attention for Winner-Take-All Networks

Consider again the example of a pattern classification task. For a classifier implemented using local representation in a neural network, a grandmother cell can represent a class of learned prototypes. The winner-take-all network is very important in this type of neural network to find the nearest matched class. Examples of such networks are Fukushima's hierarchical net for associative memory [Fukushima86] and Grossberg and Carpenter's ART net [Carpenter87a,87b].

The ART net comprises two layer of neurons and some ancillary circuitry. The lower (F1) layer of neurons receives the input patterns, and the upper (F2) layer of neurons provides the classification

result, with each F2 neuron corresponding to one particular class. Connections between these two layers are bidirectional, thereby providing feedback. A winner-take-all algorithm is applied to the upper layer. The particular winning neuron firing in this layer represents the class closest to the input pattern. The top-down trace from this winning neuron feeds the corresponding (previously learned) prototype pattern back to the input (F1) layer, where it is summed with the input pattern. An error measure circuit is used to compare the resulting output at the F1 layer with the input. If the error exceeds a preset level, then a reset action is sent to the winner-take-all network to inhibit the current winner. The rest of the neurons in the F2 layer will compete again to find the next closer class. A similar technique called selective attention has been discussed in Fukushima's hierarchical neural network for associative memory [Fukushima86].

The F1 and F2 neurons themselves can be implemented with ION using techniques similar to those described above. Additional circuitry is needed, however, for such functions as selective attention, which provides enable and reset (disable) commands to each neuron in the F2 layer. Here we describe how the ION model can implement a sample and hold subcircuit that can be used for this application. If there is no inhibitory input to a mass action neuron and we assume the maximum potential (B) to be 1, then, from Eq. (26), the steady state potential of the mass action neuron can be written as

$$x_i = \frac{1}{1 + \frac{A}{I_{exc}}}. \tag{32}$$

If the leakage (A) is zero, the mass action neuron will output 1 and hold to this value even though the excitatory input has been relinquished. A small leakage will make the output decrease slowly to zero after the release of the excitatory input. Thus the mass action neuron can implement a sample and hold function, which in discrete form can be written (from Eq. (25)) as

$$x_i(t+1) = (1 - A - I_{exc})x_i(t) + I_{exc}. \tag{33}$$

Based on this equation, the ION model can implement a sample and hold network by an I and

an N element as illustrated in Fig. 13. The N element is biased below the threshold α, so that a set signal is sufficient to fire the element. An excitatory input to the I and N element will fire the N element. The feedback from the N element output serves as the read beam of the I element that keeps the N element firing after releasing the excitatory input. The I element has a very small input bias to simulate the leakage term A. A strong input bias to the I element will reset the network. For the binary case, the input I_{exc} must be sufficiently large to fire the N element. This requirement can be met by choosing the N element bias at an appropriate level relative to the N element threshold. For this binary case, the network has the same operation as an S-R flip flop. The excitatory input is the set (S) signal and the strong leakage input is the reset (R) signal.

A selective attention network based on the above subnetwork is illustrated in Fig. 14. This network can serve to hold the output status of the winner-take-all network. The S-R flip flop is built from the ION circuit as shown in Fig. 13. At the beginning, a global set command is sent to enable all the grandmother cells for competition during the retrieval phase. Flip-flop 1 (FF1) is used to enable an individual grandmother cell for competition. The winner signal from the winner-take-all net is gated with this signal to set flip-flop 2 (FF2), which represents the winner status S_i. The S_i signal is used to inhibit all other classes (grandmother cells). If the error due to the winning class exceeds a threshold value, a global reset command is issued and is gated with S_i to reset FF2 and FF1. Thus the current cell would not be able to compete with the other grandmother cells until it is enabled by another global set command. The gating function in Fig. 14 can be realized by an N element with bias $\alpha - 1$. It needs both inputs to be high (1) to get a high output. Of course, the network of Fig. 14 does not have to be operated in the complete binary mode. For example, if a gradual but continuous (in time) leakage is desired, the "global reset" can take on an intermediate value.

5.5 Combinatorial Optimization

An attractive application of neural networks is in parallel searching for optimal or suboptimal solutions of problems. Pattern recognition and associative memory can be formulated as special

cases of this type of application. A given problem can be mapped into the weights of the network; this then defines a criterion function that determines the quality of a solution, which is usually represented by a stable state of the neurons in the network. A example is given by Hopfield and Tank [Hopfield85].

Simulated annealing [Kirpatrick83] is a technique that provides a globally optimal solution to such problems. To simulate a simulated annealing algorithm on a conventional computer is extremely time consuming. Parallel hardware implementation seems to be more plausible. An electronic realization of this algorithm for stochastic learning was discussed by Alspector and Allen [Alspector87]. Farhat and Shae [Farhat87b] proposed a noise threshold scheme for optical realization.

Using the ION model, the neuron threshold can be globally adjusted by varying the optical bias of the N element. If we introduce a time-varying amplitude-controlled noise source for the bias term, the ION can potentially be used in an optical simulated annealing by noisy threshold implementation.

6 Neural Networks that Implement Visual Cortex Operations

In this section we describe the use of the ION model in optical neural net implementations of some feature detection operations of the visual cortex. It is well known that the visual cortex can perform some low level visual processing and transformation operations such as orientation or line detection, edge enhancement, and direction of motion selection. These operations have been modeled using ION, and have been demonstrated experimentally using two Hughes liquid crystal light valves and a dichromated gelatin hologram; some of these results are summarized in the following subsections. Additional results and more detail are given in [Wang91].

6.1 Introduction to Visual Information Processing

In human visual processing, the optical signal detected by the retinal photoreceptors is processed by several layers in the retina before reaching ganglion cells, which function essentially as the output layer of the retina. There are three types of ganglion cells [Stone79], X (static), Y (transient), and W (static or transient); they are differentiated by their physical appearance, receptive field size, output signal destination cell type, signal propagation speed, and response to static or dynamic stimuli.

Most of the ganglion cell outputs are fed to lateral genticulate nucleus (LGN) cells, which serve as an intermediary between the ganglion cells and the visual cortex. The LGN cells combine the binocular views and then project them to the visual cortex [Levine85]. Areas 17, 18, and 19 are generally defined as visual cortex and perform low level processing, which includes several types of feature detection such as line, edge, and motion detection. The higher level processing, perception, visual discrimination, and learning, are done in areas 20 and 21 (inferotemporal cortex).

Figure 15 shows the cell types and features in the visual cortex of cat. Hubel and Wiesel [Hubel62] classify the visual cortical cells into three different functional groups: simple cells, complex cells and hypercomplex cells. Each cell has a corresponding receptive field. Stimulation of the retinal photoreceptors within the receptive field activates the corresponding cell. The receptive field of some cells can be divided into subfields, by their affect on the cell's activation. Compared to complex cells, simple cells have distinct excitatory and inhibitory subfields, smaller receptive fields, less spontaneous activity and prefer slower motion [Schiller76]. Each simple cell responds to a small range of *orientation* and *locations* of an edge or line within its receptive field. Some simple cells also respond to motion in a specific direction. Schiller [Schiller76] further divided simple cells into seven subclasses according to their response to directional movement. Simple cells can generally be used as building blocks to model higher level functions in complex and hypercomplex cells [Hubel62].

In the visual cortex, the cell connections are generally local and space invariant for a given

type of feature detection. A group of simple cells may respond to lines of the same orientation in spatially overlapping set of identical receptive fields. To implement these cells, we generally need an optical neural network with space variant and space invariant connections. Nonadaptive holographic optical systems can implement these interconnections. The incoherent optical neuron (ION) model can implement the requisite neuron units.

6.2 Model and Implementation of Visual Cortex Cells

In this subsection, we will discuss functional architectures for the implementation of some ganglion and simple cells in the visual cortex. The simple cells have subfields of excitatory and inhibitory regions and their connections are local and space invariant, i.e. the same connection patterns are applied to the same type of simple cells. "Simple" cells might actually exhibit very complex behavior as reported by Schiller [Schiller76]; here we model the primary functions of transient and direction-sensitive motion cells. The direction sensitive cells respond to spatial motion within their receptive field. The transient (Y) cells are ganglion cells and their outputs feed the inputs of direction sensitive cells. We will first describe the function and implementation of transient cells, and subsequently that of direction sensitive cells.

The transient (Y) cells are populated in the ganglion cells and are more sensitive to large and moving stimuli than are the static (X) cells, and it has been suggested that the Y cells play an important role in the visual discrimination of movement [Stone79]. There are two types of transient cells: those that detect *on* transients and those that detect *off* transients. Several biological mechanisms may contribute to the transient behavior of these two types of cells. These mechanisms include a delay in an interneuron, a difference in the propagation speed within different axons, and different reaction times of excitatory and inhibitory neurotransmitters. The transient cell outputs can be viewed as a nonlinear function of the *temporal* difference of incoming signals. In general, this nonlinear function exhibits a higher degree of nonlinearity than that of static cells.

An approximation to the on-transient and off transient cell responses can be implemented using the neuron connections shown in Fig. 16. Each transient cell receives its input just from one input

node. The temporal responses for these implementations are also shown in Fig. 16; the temporal responses shown assume each N element has one time unit of delay and each I element has negligible time delay. (This is an abstraction of the temporal responses obtained in our experiment using LCLVs, as discussed below.) The ION circuits shown in Fig. 16(a) and (c) do not impose any requirements on the relative delay of the I and N elements; the circuit of Fig. 16(b) requires the N element to have a longer delay.

Hubel and Wiesel [Hubel59,62] showed that some orientation sensitive simple cells have directionally asymmetric characteristics. Berman, Wilkes and Payne tested 138 unit cells in the cat striate cortex [Berman87], and suggested a resulting classification of cells as bidirectional (exhibiting little direction preference), directional bias (moderate direction preference) and directional (strong direction preference). It also has been observed that the direction sensitive cells respond to a limited range of speeds and thus might more precisely be called velocity sensitive cells [Baker88]. The preferred speed can range by a factor of 100 among different cells.

Emerson and Gerstein [Emerson77] proposed two mechanisms to account for direction selectivity: lateral inhibition and excitation. There is experimental evidence that the mechanism of lateral inhibition is crucial in obtaining directional selectivity. The inhibitory transmitter γ - *aminobutyric acid* (GABA) plays an important role in direction selectivity. It was shown that iontophoretic application of bicuculline, which blocks the function of GABA by binding to inhibitory receptor sites, reduced or abolished direction selectivity [Sillto77].

Several models for direction selectivity have been reviewed by Ruff et. al. [Ruff87]. Our implementation of the direction sensitive cells is based on the facts of speed preference *and* a lateral inhibition mechanism. Figure 17(a) shows the interconnections of direction sensitive cells for the one-dimensional case. For left motion, the cell inhibits all interneuron cells to its left except the one two cells over. This prefers a moderate speed (on the order of two input cell-to-cell distances per unit cell delay time) in the left direction. The cell also inhibits all interneuron cells to its right and thus no response is generated for right motion. Direction sensitive cell inputs are fed from transient cell outputs. An example of the 2-D lateral connections of direction sensitive

cells is shown in Fig. 17(b) illustrating different speed preferences. A unidirectional (classified as bidirectional, above) motion sensitive cell can also be implemented, by making the connections approximately circularly symmetric, with the radius of the excitatory region being proportional to the speed preference.

Figure 18 shows a conceptual optical implementation of direction sensitive cells based on the ION model. Since in our implementation the I elements had a much faster response than the N elements, an extra N element is used to provide sufficient delay in the upper path. The preferred speed can be changed for different direction sensitive cells by changing the interconnection pattern of Fig. 18. The inputs of direction sensitive cells are fed from transient cells; in our experiment these inputs were provided by a computer-controlled input device. As described above, the transient cells can also be implemented by the ION model, so the two implementations can be cascaded to form complete direction sensitive cells.

For the direction sensitive cell implementations described above, a complete cell would actually include one additional layer. This layer has one neuron unit per receptive field, and serves to combine all outputs (from a given receptive field) of a given cell type, so that the final output responds to the aggregate over a particular receptive field. This could be implemented via another interconnection stage using a hololens similar to the one used in our experimental setup. This additional layer is included in the schematic diagram for the case of a direction sensitive cell in Fig. 17.

6.3 Experimental System

Figure 19 shows the primary components of the experimental setup used to implement the simple and ganglion cells discussed above. Two liquid crystal light valves are used; one (LCLV1) for input conversion, which converts incoherent cathode ray tube output to a coherent beam, and the other (LCLV2) to implement the optical neuron units. An Argon laser is used as the optical source for all optical beams except for the write beams of LCLV1. The input to the network is an image generated from a three inch cathode ray tube, and serves as the write beam of LCLV1.

A multiplexed dichromated gelatin hologram [Chang79,80] serves as a holographic optical element (HOE) [Owen85] that performs space invariant point spread functions for the network interconnections. The HOE is a multiplexed hololens that images the input pattern on LCLV1, through the recorded (space-invariant) interconnection pattern, to the input plane of LCLV2. The HOE was exposed optically using a computer-controlled system. The system used multiple sequential exposures to provide fanout in the resulting reconstruction.

The LCLV2 implements arrays of I and N elements, per the ION model. The output of LCLV2 splits into two feedback paths (through L1 and L2); one for the connections from the I element outputs to the N element inputs and the other path for the self-feedback of the N-elements to increase their sensitivity. Both paths image the LCLV2 output plane onto the LCLV2 input plane. The positive self-feedback used for all N elements in our system increases their response time, causing the I and N element response times to differ. The pixel size for each I and N element is 0.25 mm in diameter with center-to-center spacing of 0.50 mm; the LCLV2 is divided into three regions with each region containing 35 pixels organized into 7 rows and 5 columns. The three regions from left to right (viewing the output side of LCLV2) are operated as N, I, and I elements for the off-transient cells and as N, I, and N elements for the direction sensitive motion cells.

The read beam of LCLV2 is provided by a collimated beam through mask MK1 and BS4. Mask MK1 defines the area of the LCLV2 used to realize the ION components including pixelation to individual neuron units; the output plane of LCLV2 is in an image plane of MK1. The bias beam for the N elements is supplied by the beam path through MK2 and BS5. There is one mask in front of lens L1 to block out the unnecessary self-feedback for the I elements. Each feedback path includes a polarizer for control of the overall feedback connection strengths.

6.4 Experimental Results and Discussion

Figure 20 shows outputs of the implemented transient cells. They are designed to respond only to off transients. The input and output regions are each 7x5 arrays. Due to the optical setup, the input image as shown in all figures is inverted spatially about its center relative to the output

images. In this experiment, a block of bright input patterns is moving from top to bottom (shown as bottom to top in the input frames). The responses shown from left to right are sampled every 66 ms in time to show their temporal responses.

Figure 21(a) and (b) show the on-transient and off-transient temporal responses of the fourth row respectively. In Fig. 21(a), the fourth row of the output corresponds to the on-transient row of the input (top bright row of the final input as shown in Fig. 21). As expected, there is no *on* response in the output. In Fig. 21(b), the fourth row of the output corresponds to the off-transient row of the input (bottom bright row of the initial input as shown in Fig. 21(b)). As expected, an *on* response is obtained in the output. Similarly, Fig. 22 shows the outputs of transient cells corresponding to an ON-OFF pattern in the input plane. The recorded images are continuous from left to right, taking (a) first and (b) second, with 66 ms interval in successive frames. Again, a positive response to the off transient is seen, without such a response to the on transient, as desired.

For the direction sensitive motion cells, Fig. 23(a) and (b) show the results for an upward and downward moving line respectively (again, the input shown in the figure is spatially inverted). The output for each shown frame is three 7x5 regions on LCLV2, corresponding to N, I, and N element arrays (from left to right). The final output of each frame is the N element array on the right. The inputs of the direction sensitive motion cells excite corresponding cells and inhibit the neighboring cells as shown in Fig. 17(b); arranged as shown in Fig. 18(b), they will suppress downward motion and horizontal motion. Thus, a response is seen only for upward motion as in Fig. 23(a).

As outlined above (Sec. 3.2) and detailed in Refs. [Wang88,Wang90], the distinguishable fan-in is essentially limited by the extinction ratio of the optical devices. Some spatial light modulators may have insufficient extinction ratio to support the requisite distinguishable fan-in of a massively connected neural network. A pyramid structure can be used to permit a larger effective fan-in (Fig. 24). For example, a two-level pyramid structure with a fan-in of 36 for each interneuron can realize a total fan-in of 1296. This kind of structure has two advantages during the implementation of the optical visual cortex: an increase in the fan-in or receptive field size, and the ease of aggregating the outputs of simple cells to form a complex cell. Since most of the visual cortex cells are locally

and regularly connected, a space invariant hololens is an efficient means for implementing the interconnections. With such an implementation, the structure in Fig. 24(b) yields a simpler interconnection than that of Fig. 24(a) due to its space invariant property. The device area utilization can be maximized by incorporating instead a multiplexed hybrid space-variant / space-invariant optical system for the interconnections (see for example, Ref. [Jenkins84b]).

7 Conclusion

The ION model can subtract inhibitory from excitatory neuron inputs by using two device responses. Functionally it accommodates positive and negative weights, excitatory and inhibitory inputs, nonnegative neuron outputs, and can be used in a variety of neural network models. An extension is given to include bipolar neuron outputs in the case of fully connected networks. Two different responses are generally required to realize a complete neuron with ION. One response is sometimes sufficient to implement an interneuron, for example to achieve the requisite delay in a transient cell.

The features of the ION model include a bias that is essentially independent of input weights and signals, a dynamically and globally variable threshold, the capability of implementing a sigmoid or binary threshold function for different neuron models, cascadability and ease of implementation. For example, this technique can in principle implement conventional inner-product neuron units and Grossberg's mass action law neuron units.

The ION model, in conjunction with optical weighted interconnections, can be used to implement arbitrarily connected neural networks. We have outlined the use of ION in single layer feedback networks and multilayer networks including Fukushima's neocognition models. Since ION provides the capability of both excitatory and inhibitory inputs, a variety of models and features can in principle be implemented merely by incorporating additional optical circuitry. For example, ION can implement Grossberg's mass action law neuron units by incorporating additional optical feedback paths. Other features, for example input normalization and selective attention for

winner-take-all networks, used for example in adaptive resonance theory (ART) networks, can also be implemented using ION, as described above.

An LCLV can be divided into several regions to implement various I and N elements, and in conjunction with a feedback loop it can realize a multistage neural network. We have used these techniques in an optical implementation of ganglion cells and simple cells of the visual cortex. Experimental results of transient cells and direction sensitive cells were presented. Experimental results of other simple cells are given in [Wang91]. In these experiments, we demonstrated some of the key mechanisms in the visual cortex. These present many of the mechanisms that, when combined, can be used to realize complete simple and complex cortical cells.

8 Acknowledgements

The help of J. M. Wang in the optical visual cortex experiments is gratefully acknowledged. The authors would like to thank the faculty of the USC Center for the Integration of Optical Computing for numerous stimulating discussions. Parts of this work were supported by AT & T Corp., AFSOR University Research Initiative under contract F49620-87-C-007, National Science Foundation under Presidential Young Investigator Award No. MIP-88-58094, and by AFSOR under grant No. 86-0196.

9 Reference

[Alspector87] Alspector, J. and Allen, R. B. (1987). "A Neuromorphic VLSI learning system", In Proc. of the 1987 Stanford Conf., Advanced Research in VLSI, (Paul Losleben ed.). MIT press 1987.

[Amari72] Amari, S-I. (1972). IEEE Trans. on Computers C-21, 1197-1206.

[Amari82] Amari, S-I. (1982). In Competition and Cooperation in Neural Nets (Amari S-I. and Arbib M. A. eds.) Spring Verlog press, New York, 1-28.

[Anderson87] Anderson, D. Z. and Lininger, D. M. (1987). Appl. Optics 26, 5031-5038.

[Baker88] Baker, Jr. C. L. (1988). J. of Neurophysiol. 59, 1557-1574.

[Berman87] Berman, Nancy E. J., Wilkes, M. E. and Payne, B. R. (1987). J. of Neurophysiol. 58, 676-699.

[Bleha78] Bleha, W. P. et. al. (1978). Optical Engineering, 17, 371-384.

[Carpenter87a] Carpenter, G. A. and Grossberg, S. (1987). Computer Vision, Graphics and Image Processing 37, 54-115.

[Carpenter87b] Carpenter, G. A. and Grossberg, S. (1987). Appl. Optics 26, 4919-4930.

[Chang79] Chang, B. J. and Leonard, C. D. (1979). Appl. Optics 18, 2407-2417.

[Chang80] Chang, B. J. (1980). Optical Engineering 19, 642-648.

[Ebersole75] Ebersole, J. F. (1975). Optical Engineering 14, 436-447.

[Efron85] Efron, U., Marom, E. and Soffer, B. H. (1985). In *Topical meeting on optical computing* (OSA technical digest) 1985.

[Ellias75] Ellias, E. A. and Grossberg, S. (1975). Biol. Cybernetics 20, 69-98.

[Emerson77] Emerson, R. C. and Gerstein, G. L. (1977). J. of Neurophysiol. 40, 136-155.

[Farhat85] Farhat, N. H., Psaltis, D., Prata, A. and Paek, E. (1985). Applied Optics 24, 1469-1475.

[Farhat87a] Farhat, N. H. (1987). Applied Optics 26, 5093-5103.

[Farhat87b] Farhat, N. H. and Shae, Z. Y. (1987). In IEEE first Int. Conf. on Neural Networks III, 565-575.

[Fukushima75] Fukushima, K. (1975). Biol. Cybernetics 20, 121-136.

[Fukushima80] Fukushima, K. (1980) Biol. Cybernetics 36, 193-202.

[Fukushima84] Fukushima, K. (1984). Biol. Cybernetics 50, 105-113.

[Fukushima86] Fukushima, K. (1986). Biol. Cybernetics 55, 5-15.

[Fukushima87] Fukushima, K. (1987). Appl. Optics 26, 4985-4992.

[Gmitro87] Gmitro, A. F. and Gindi, G. R. (1987). In IEEE First Int. Conf. on Neural Networks III, 599-606.

[Goodman84] Goodman, J. W., Leonberger, F. J., Kung, S-Y. and Athale, R. A. (1984). Proc. of IEEE 72, 850-866.

[Grossberg73] Grossberg, S. (1973). Studies in Appl. Math. 52, 213-257.

[Grossberg76a] Grossberg, S. (1976). Biol. Cybernetics 21, 145-159.

[Grossberg76b] Grossberg, S. (1976). Biol. Cybernetics 23, 121-134.

[Grossberg76b] Grossberg, S. (1976). Biol. Cybernetics 23, 187-202.

[Grossberg80] Grossberg, S. (1980). Proc. Natl. Acad. Sci. USA 77, 2338-2342.

[Hopfield82] Hopfield, J. J. (1982). Proc. Natl. Acad. Sci. USA 79, 2554-2558.

[Hopfield85] Hopfield, J. J. and Tank, D. W. (1985). Biol. Cybernetics 52, 141-152.

[Hubel59] Hubel, D. H. and Wiesel, T. N. (1959). J. Physiol. 148, 574-591.

[Hubel62] Hubel, D. H. and Wiesel, T. N. (1962). J. Physiol. 160, 106-154.

[Hubel65] Hubel, D. H. and Wiesel, T. N. (1965). J. Neurophysiol. 28, 229-289.

[Jenkins84a] Jenkins, B. K., Sawchuk, A. A., Strand, T. C., Forchheimer, R. and Soffer, B. H. (1984). Appl. Optics 23, 3455-3464.

[Jenkins84b] Jenkins, B. K., Chavel, P., Forchheimer, R., Sawchuk, A. A. and Strand, T. C. (1984). Appl. Optics 23, 3465-3474.

[Jenkins87] Jenkins, B. K. and Wang, C. H. (1987). J. of Opt. Soc. Amer. (A) Optics & Image Science, 4, p127A.

[Jenkins88] Jenkins, B. K. and Wang, C. H. (1988). Optics Letters 13, 892-894.

[Jenkins91] Jenkins, B. K. and Tanguay, Jr. A. R. (1991). "Photonic implementations of neural networks", In Neural Networks and Fuzzy Systems: A dynamical approach to machine intelligence (Kosko B. ed.) Prentice Hall, New York, 1991.

[Kirpatrick83] Kirpatrick, S., Gelatt, Jr. C. D. and Vecchi, M. P. (1983). Science 220, 671-680.

[Lentine88] Lentine, A. L. and Hinton, H. S., Miller, D. A. B., Henry, J. E., Cunningham, J. E. and Chirovsky, L. M. F. (1988). Appl. Phys. Lett. 52, 1419-1421.

[Levine85] Levine, M. D. (1985). In Vision in man and machine, chapter 3 & 5, Mcgraw-Hill Book Comp., 1985.

[Malsburg73] Malsburg, Chr. von der (1973). Kybernetik 14, 85-100.

[Marom77] Marom, E. and Grinberg, J. (1977). Appl. Optics 16, 3086-3087.

[Miller85] Miller, D. A. B., Chemla, D. S., Damen, T. C., Wood, T. H., Burrus, Jr. C. A., Gossard, A. C. and Wiegmann, W. (1985). IEEE J. of Quantum Electronics QE-21, 1462-1476.

[Miyake84] Miyake, S. and Fukushima, K. (1984). Biol. Cybernetics 50, 377-384.

[Owen85] Owen H. (1985). SPIE Proceeding 523, 296-301.

[Psaltis88a] Psaltis, D., Brady, D. and Wagner, K. (1988). Appl. Optics 27, 1752-1759.

[Psaltis88b] Psaltis, D., Gu, X-G and Brady, D. (1988). In Optical Computing 88 SPIE, 963, 468-474.

[Ruff87] Ruff, P. I., Rauschecker, J. P. and Palm, G. (1987). Biol. Cybernics 57, 147-157.

[Schiller76] Schiller, P. H., Finlay, B. L. and Volman, S. F. (1976). J. Neurophysiol. 39, 1320-1333.

[Shariv89] Shariv, I. and Friesem, A. A. (1989). Optics Letters 14, 485-487.

[Shepherd78] Shepherd, G. M. (1978). Scientic American Feb., 93-103.

[Sillto77] Sillto, A. (1977). J. Physiol. 271, 699-720.

[Stevens79] Stevens, C. F. (1979). Scientic American Aug., 55-65.

[Stone79] Stone, J., Dreher, B. and Leventhal, A. (1979). Brain Research Review 1, 345-394.

[Te Kolste87] Te Kolste, R. D. and Guest, C. C. (1987). IEEE First Int. Conf. on Neural Network, III, 625-629.

[Walker86] Walker, A. C. (1986). Applied Optics 25, 1578-1585.

[Wang88] Wang, C. H. and Jenkins, B. K. (1988). Proc. first ICNN II, 403-410.

[Wang89a] Wang, C. H. and Jenkins, B. K. (1989).
Proc. IEEE 3rd annual Parallel Processing Symposium, Fullerton, March 1989.

[Wang89b] Wang, C. H. and Jenkins, B. K. (1989). annual meeting of OSA, paper WU1, Orlando, Florida, Oct. 1989.

[Wang90] Wang, C. H. and Jenkins, B. K. (1990). Applied Optics 29, 2171-2186.

[Wang91] Wang, C. H., Jenkins, B. K. and Wang, J. M. (1991). "Visual cortex operations and their implementation using the incoherent optical neuron model," submitted to Applied Optics, information processing division, Feb. 1991.

[Wang-Freeman87] Wang, M. and Freeman, A. (1987). In Neural function, Little, Brown and Company press, 1987.

```
calc()
register  a;      / * weight * /
register  b;      / * input_value * /
register  c;      / * store weighted sum * /
register  d=n;    / * count index * /
          c=0;
  while   (d=0)   (7 or 3)
          {
          a=      weight[d]; (4)
          b=      input_value[d]; (4)
          a=      a * b; (9)
          c=      c + a; (2)
          d=      d - 1; (2)
          }
          end;
```

Figure 1: A sample procedure to calculate membrane potential based on a uniprocessor. The value shown in parenthesis is the number of clock cycles for an Intel 80386 processor. Clock period is 50 ns. Only 45% of the time is used in actual computation.

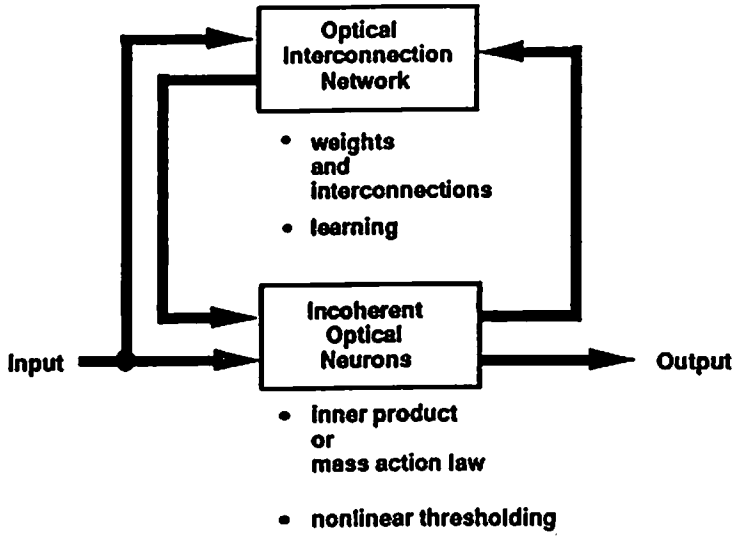


Figure 2: A paradigm for an optical neural network.

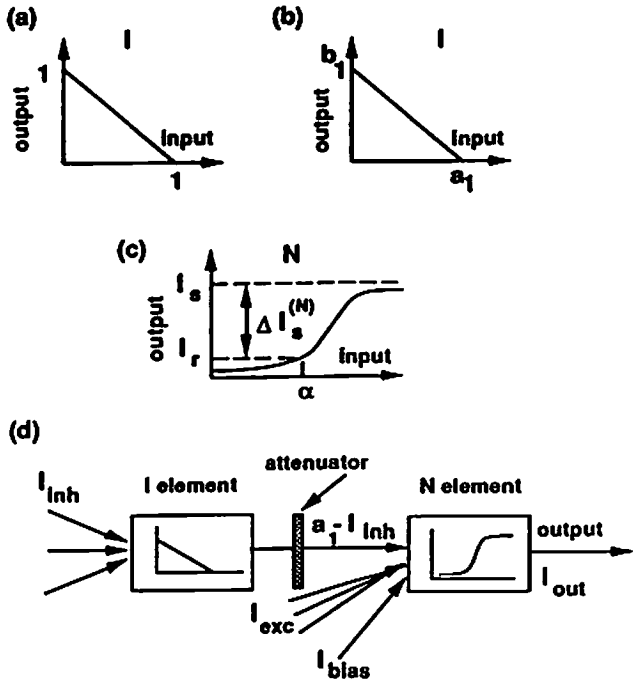


Figure 3: The ION model: (a) normalized inhibitory (I) element, (b) unnormalized I element, (c) nonlinear (N) element, and (d) the ION structure.

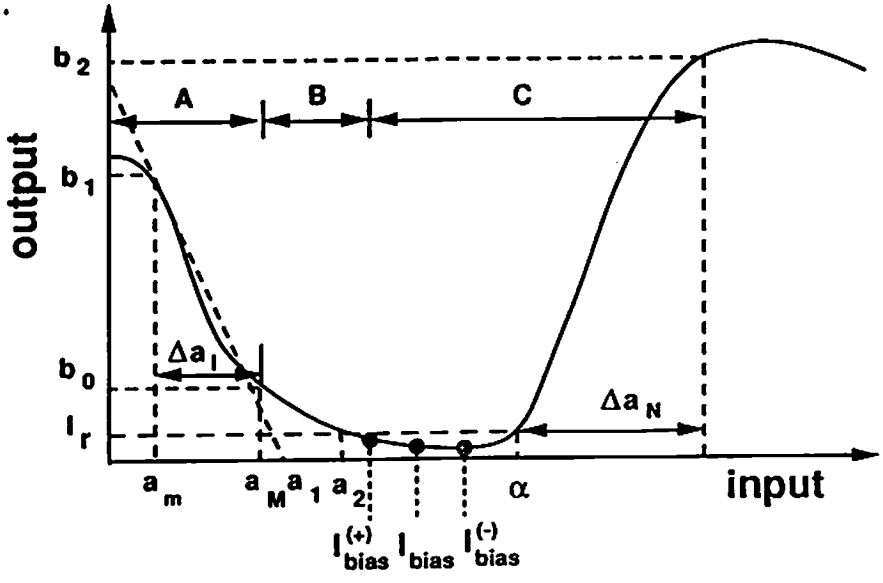


Figure 4: Homogeneous case example: typical characteristic of a Hughes liquid crystal light valve, serving as both I and N elements. Regions A and C are used for the I and N element respectively. Region B serves to separate the I element from the N element.

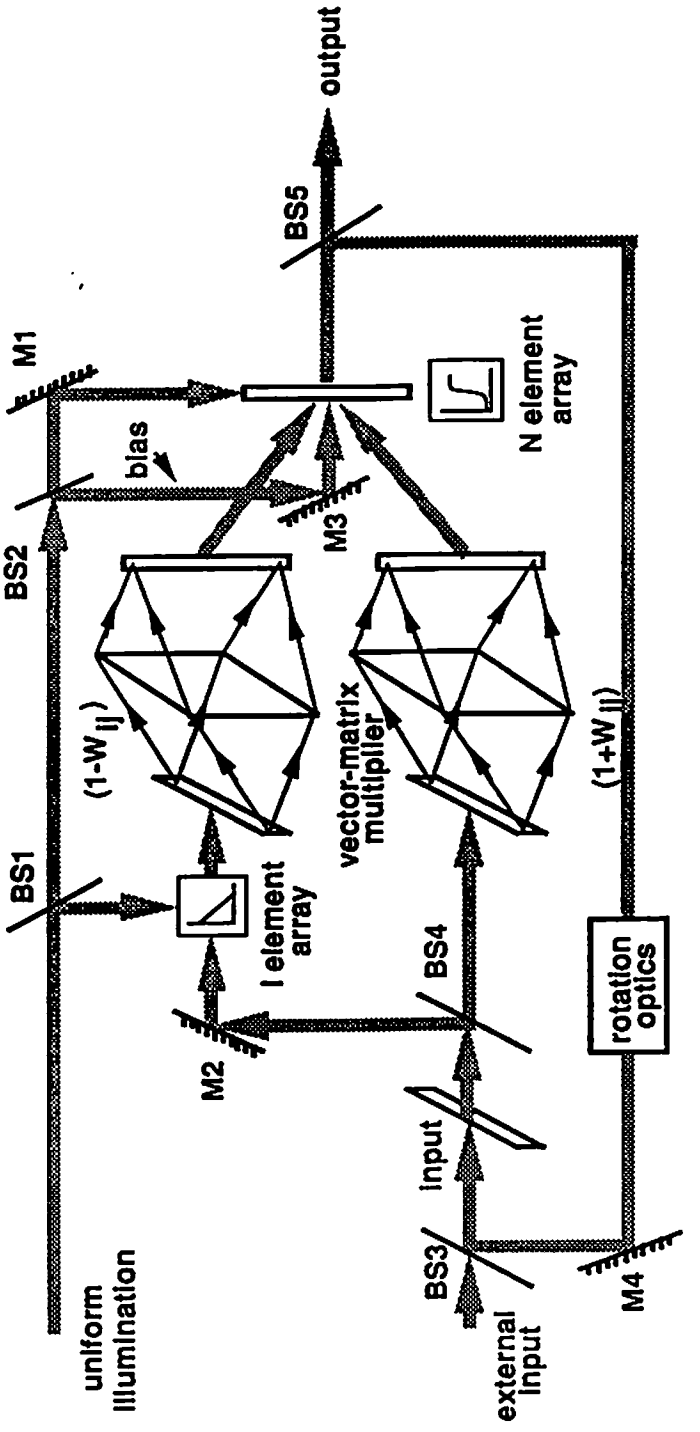
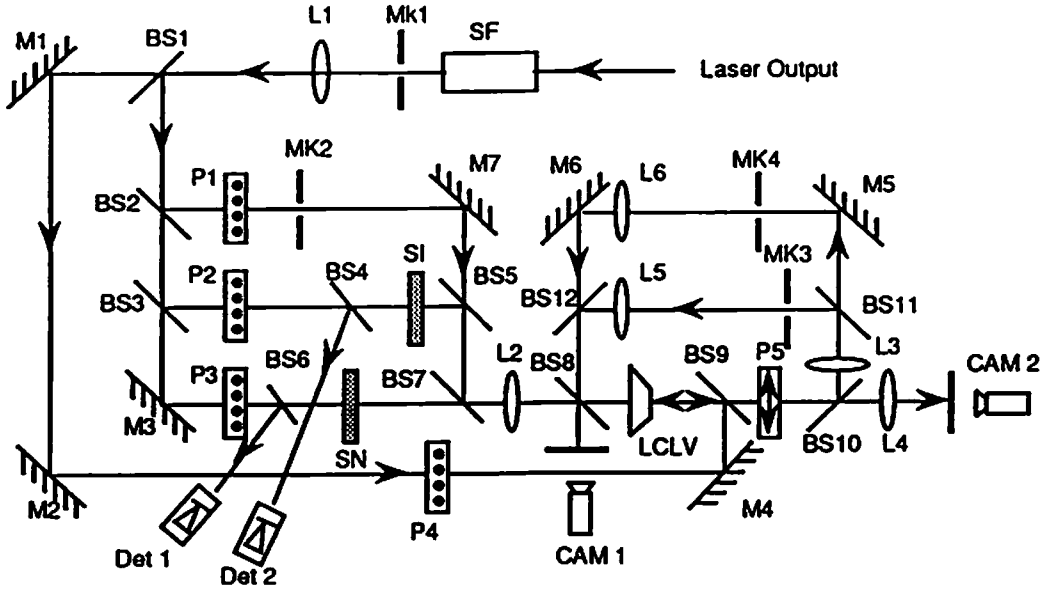


Figure 5: Conceptual optical architecture for implementing an Amari/Hopfield network using the complementary ION model, shown for the case of 1-D arrays of neurons.

Figure 5: Conceptual optical architecture for implementing Amari/Hopfield network using the complementary ION model, shown for the case of 1-D arrays of neurons.



SF:	spatial filter	Det 1-2:	power meters
L1-6:	lenses	CAM 1-2:	CCD cameras
M1-6:	mirrors	MK 1-4:	masks
P1-5:	polarizers	SI:	I element input
BS1-12:	beam splitters	SN:	N element input

Figure 6: Experimental setup of the ION test circuit.

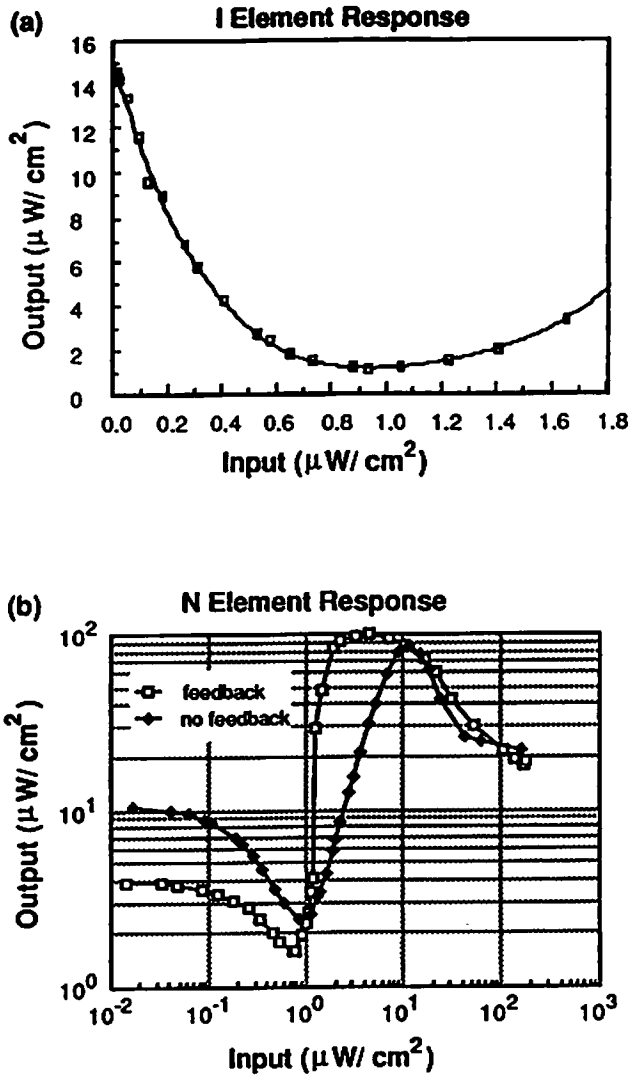


Figure 7: LCLV characteristics of the I and N element in the test circuit for V=5.0 volts, f=1.5 Khz. The vertical axis is the intensity measured at the LCLV output, when in the system of Fig. 6 with a laser power of 200 mW. The I element is fairly linear within 50% of its operation range. The self feedback of the N element, (b), is necessary to satisfy the ION requirement for this particular device.

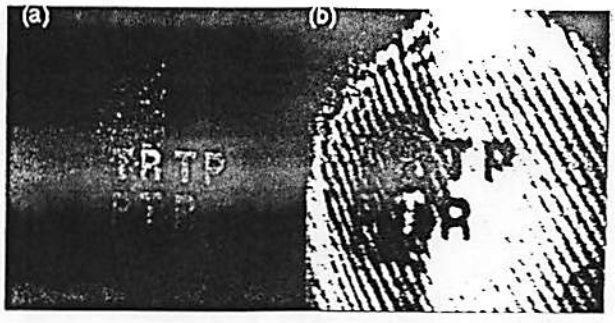


Figure 8: (a)-(b) Results of binary subtraction with (a) input patterns: N inputs (left) and I inputs (right), at LCLV input; and (b) outputs: N element outputs, the subtraction result (left), and I element outputs (right). The ideal result is the residual leg of "R" (top right) and full "T" (bottom right) of the N element output.

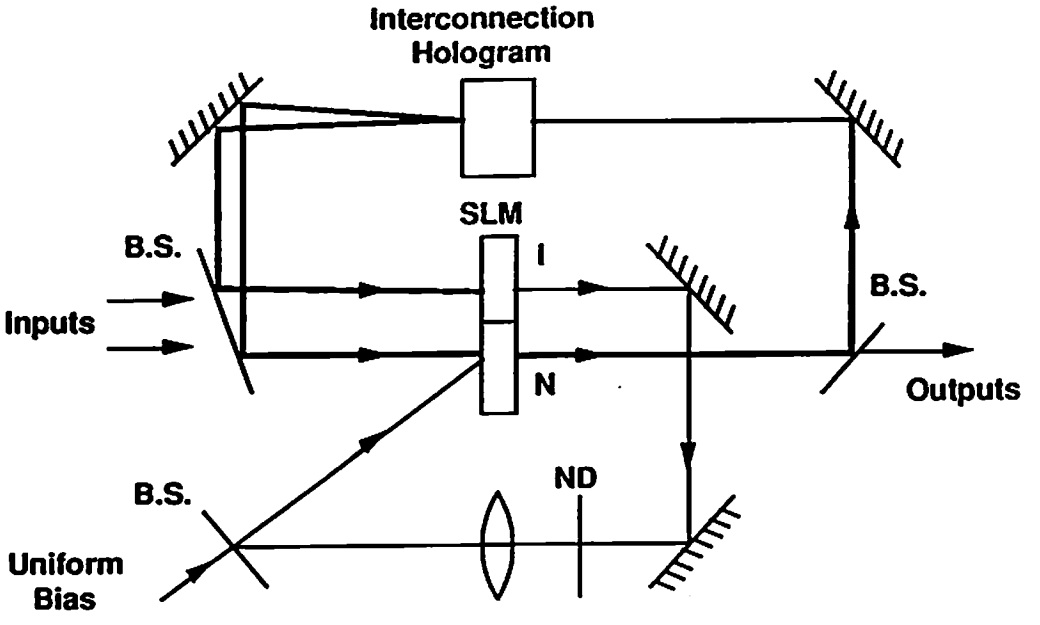


Figure 9: Single layer feedback net using a single spatial light modulator to implement both I and N elements.

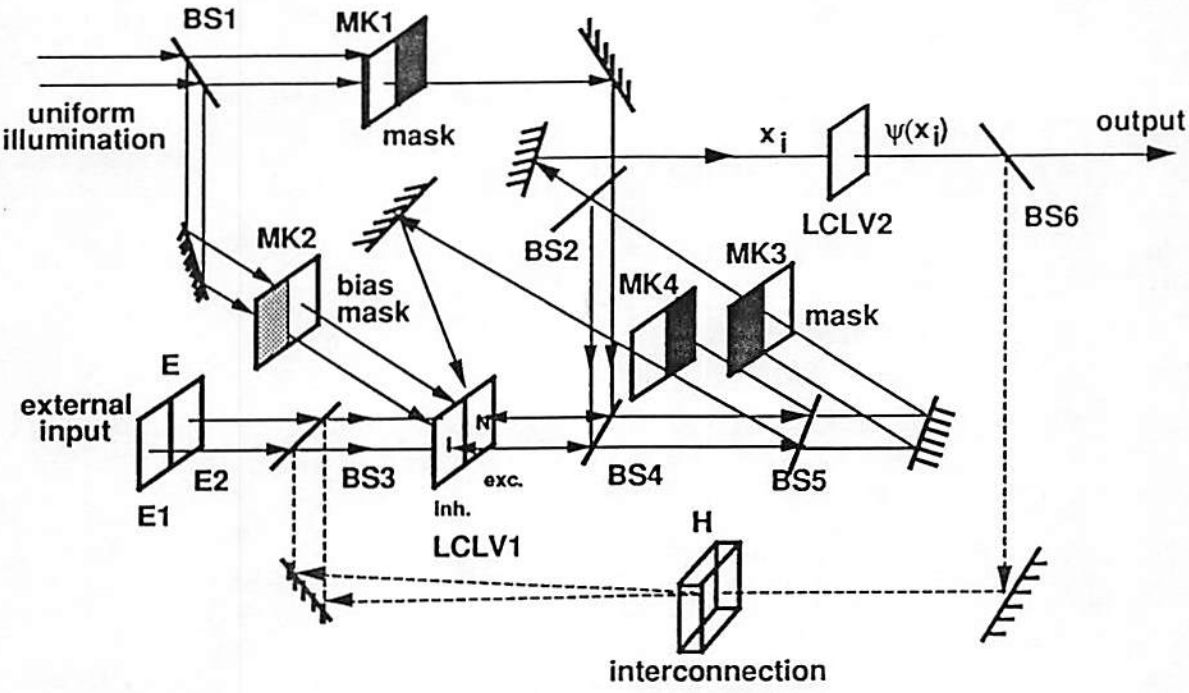


Figure 10: Conceptual implementation of a neural network based on Grossberg's mass action type neurons, using the ION model and LCLVs. MK: mask; BS: beamsplitter. Solid lines for beams depict the mass action neuron implementation; dotted lines depict addition of interconnections to implement a complete neural network.

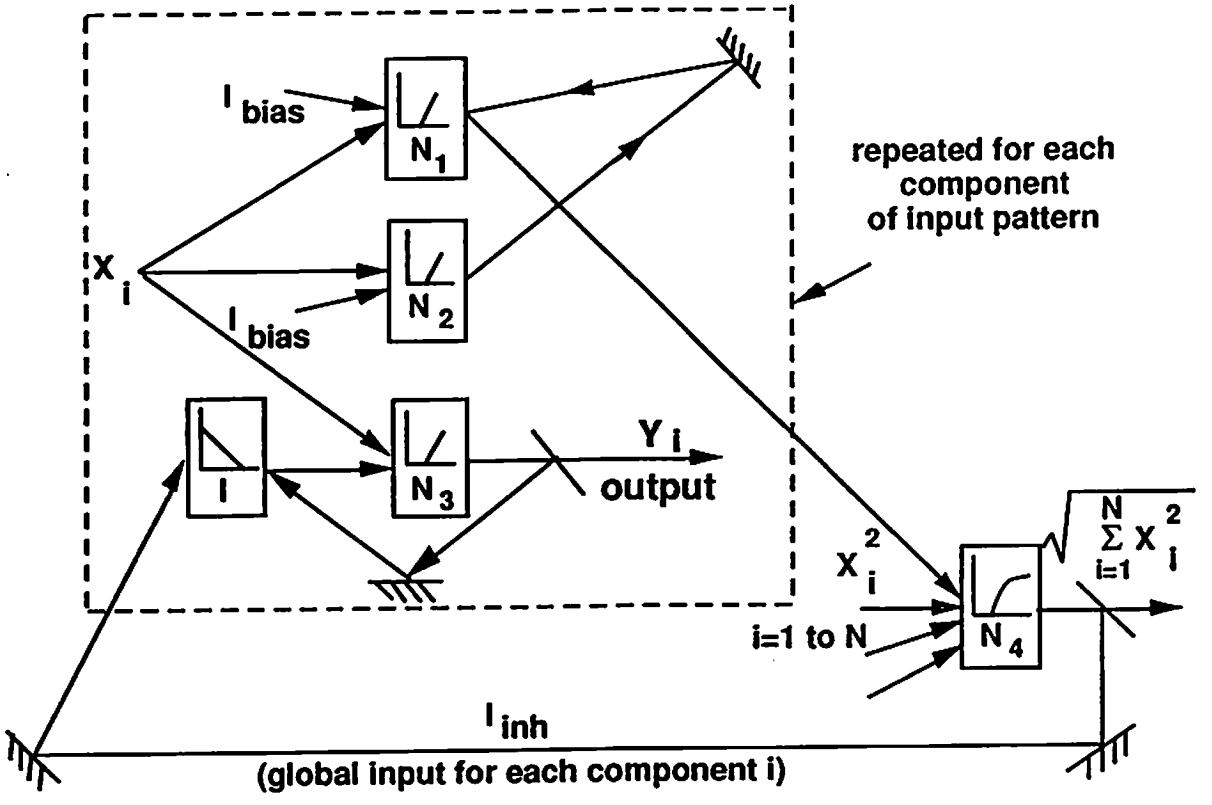


Figure 11: ION structure for implementation of input normalization based on mass action neurons.

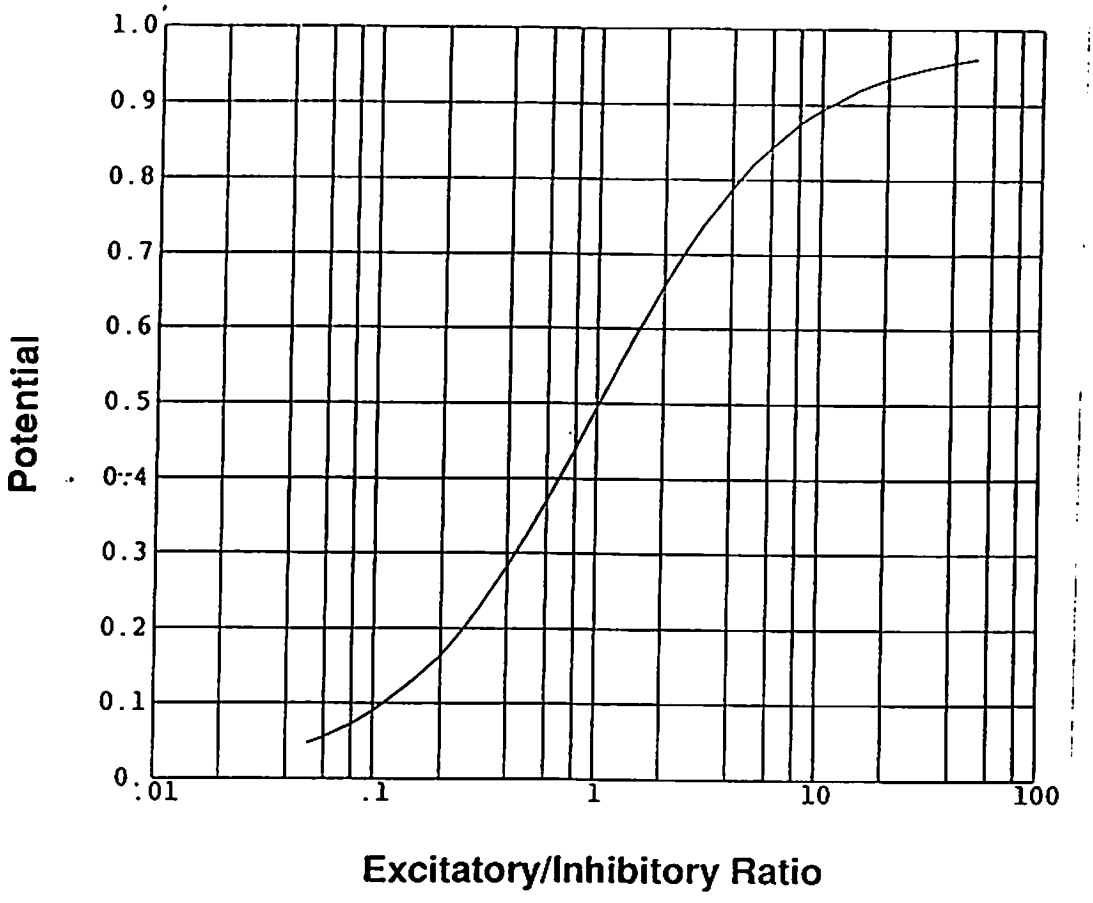


Figure 12: Potential output of a mass action neuron vs. excitatory/inhibitory input ratio for the case of very small leakage ($A = 0.01$).

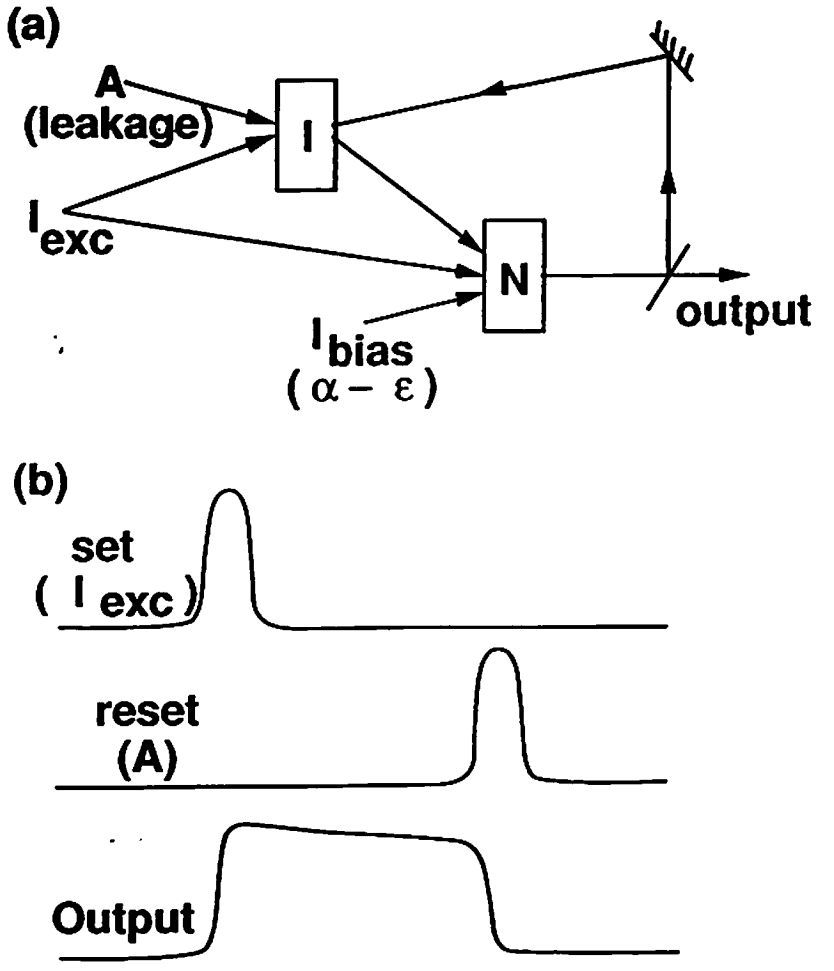


Figure 13: ION implementation of a sample and hold network showing (a) the ION circuit, and (b) the operation timing diagram.

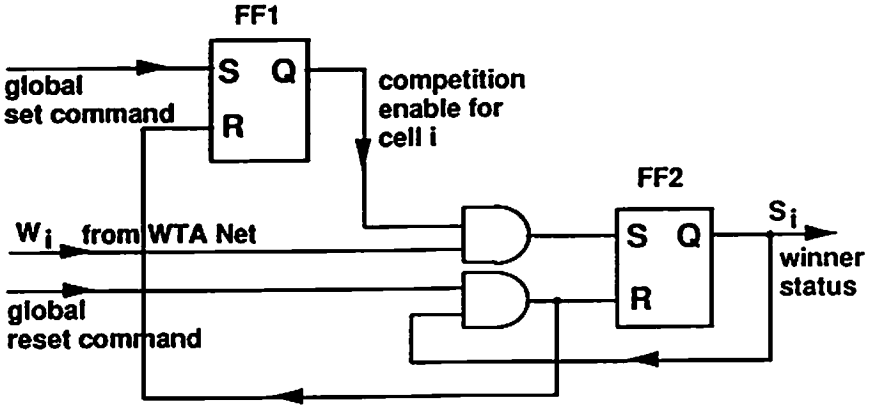


Figure 14: Block diagram of a selective attention network based on ION. One of these networks is used for each grandmother cell in the winner-take-all (WTA) net.

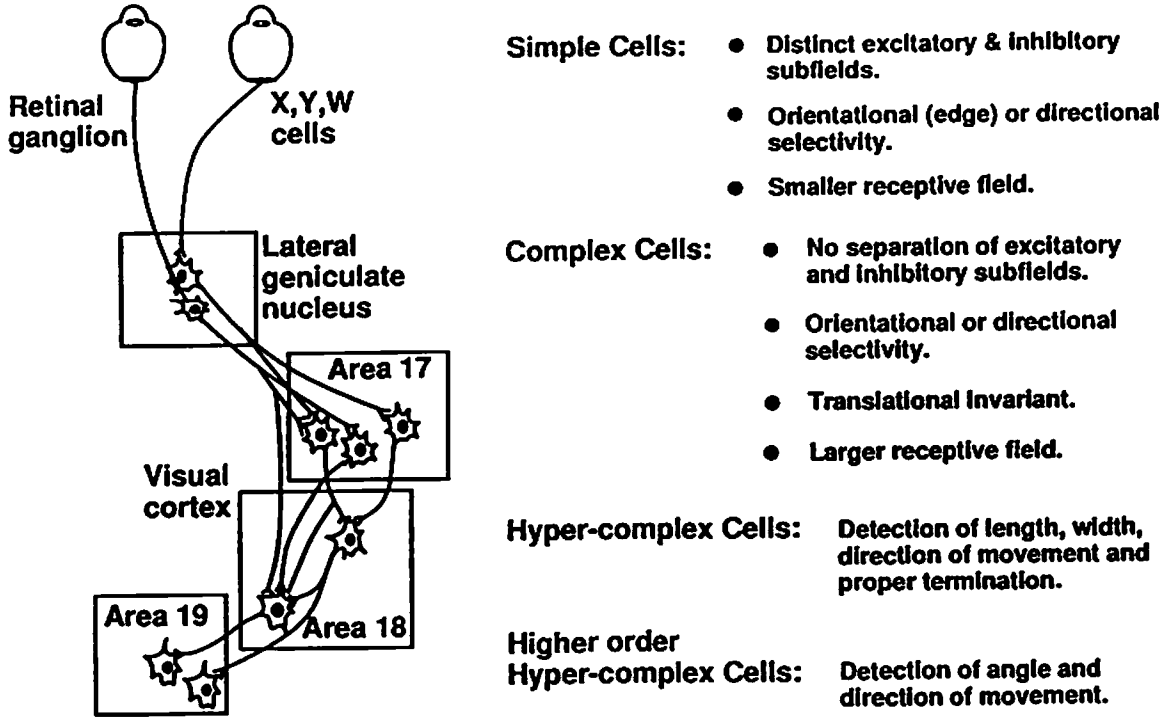


Figure 15: Cell types and features in the visual cortex of cat [Hubel62].

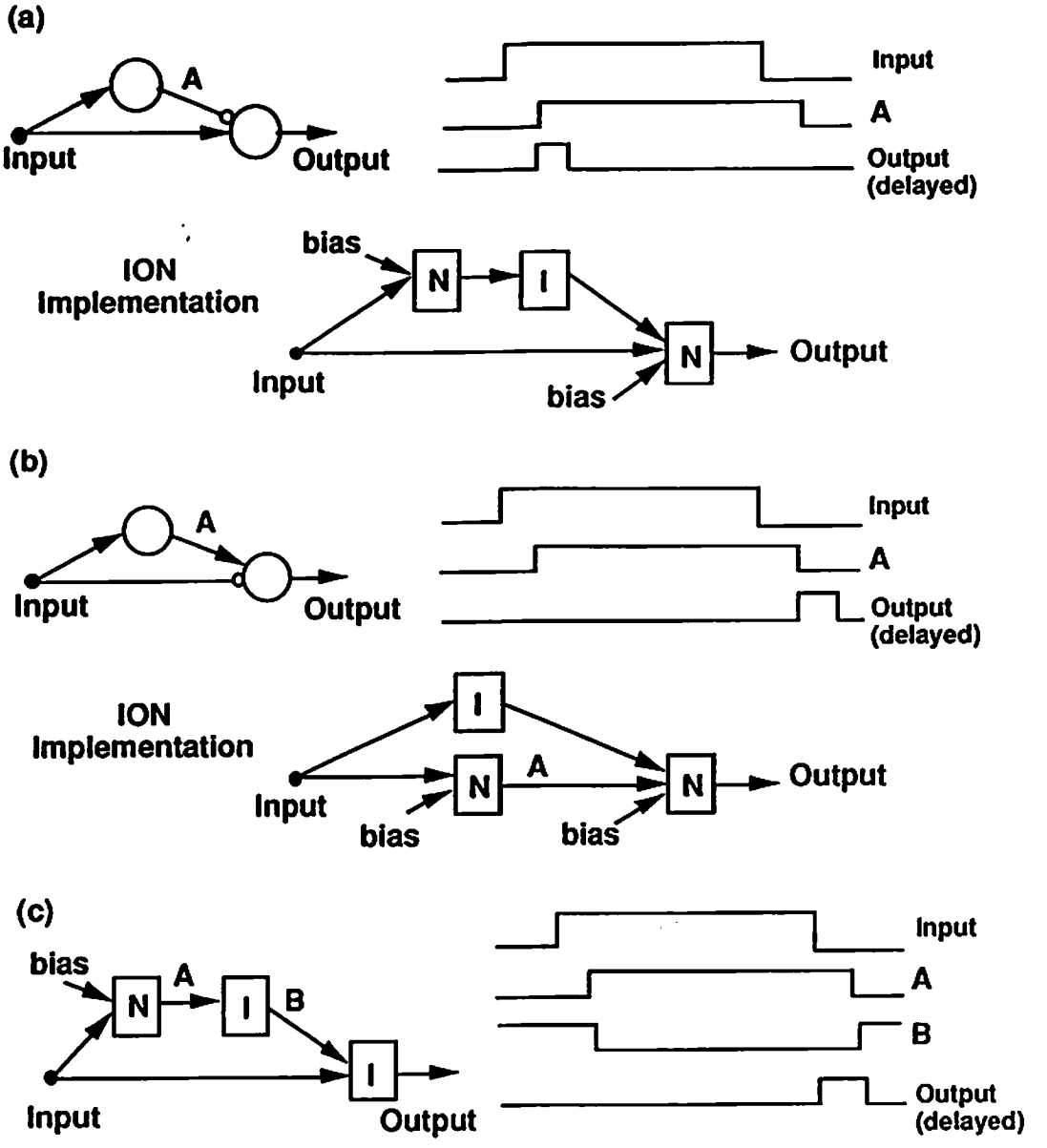


Figure 16: Implementation of transient cells and their temporal responses. The shown waveforms are for the binary case. (a) on-transient cell, (b) off-transient cell, and (c) another implementation of off-transient cell. The temporal responses shown assume each N element has one time unit of delay and each I element has negligible time delay.

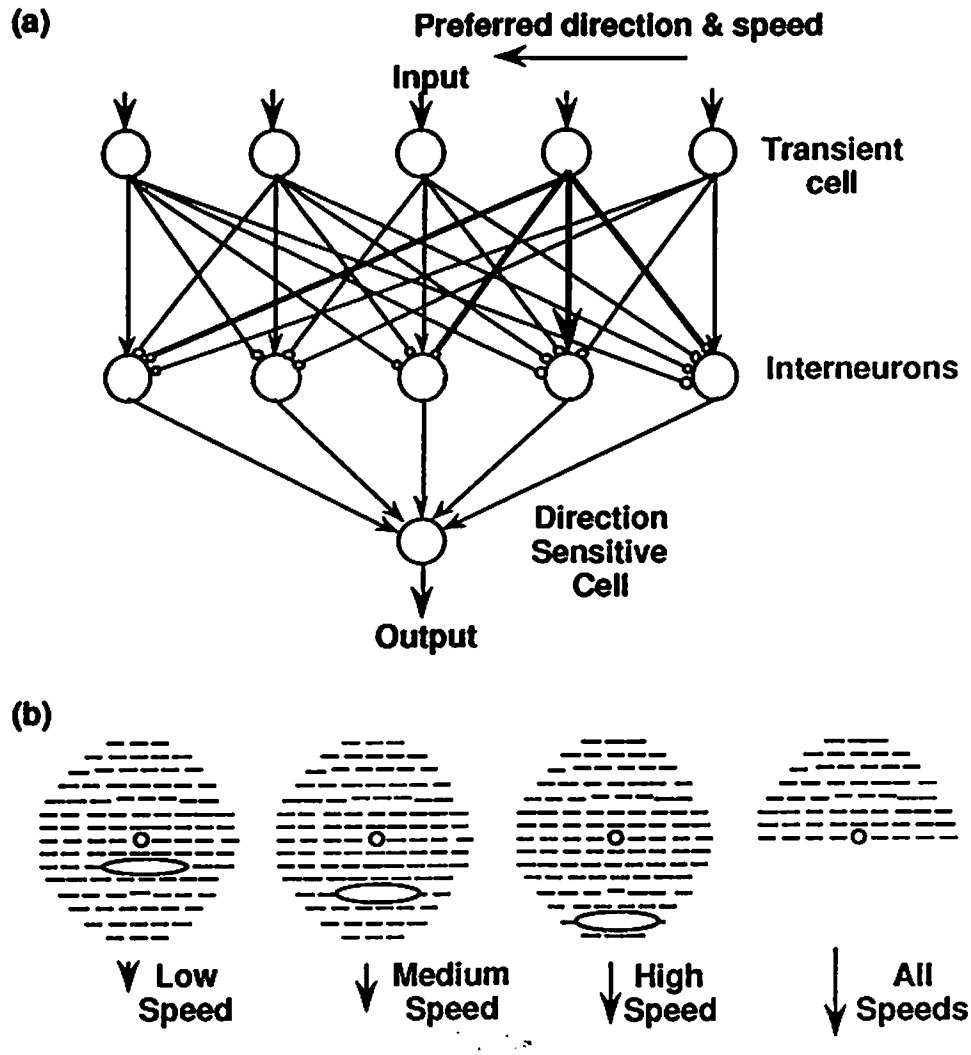


Figure 17: A neural network for direction sensitive and speed sensitive motion cells based on lateral inhibition. (a) A 1-D example, and (b) examples of 2-D interconnection patterns of the interneurons for various speed preference. The degree of directional selectivity can be changed by altering the width of the non-inhibitory regions.

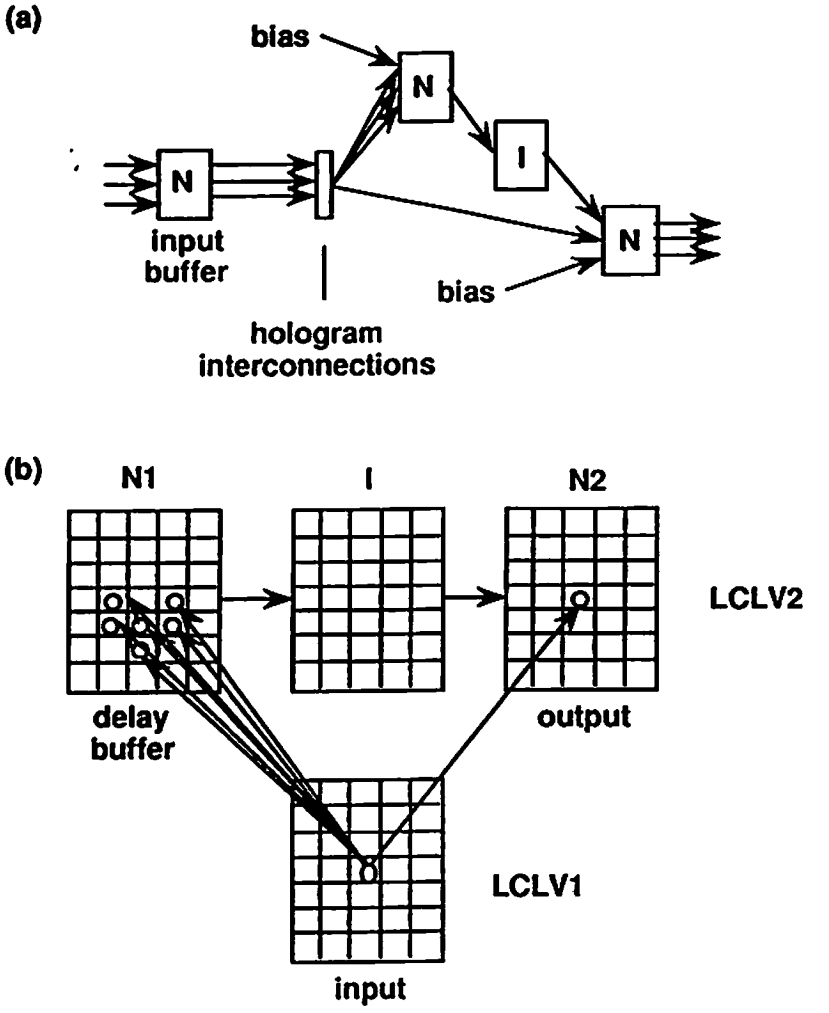


Figure 18: Optical implementation of direction and speed sensitive cells based on lateral inhibition, showing (a) ION implementation, and (b) physical layout and interconnection point spread function. The N1 region serves as delay interneurons, and the other regions (I and N2) serve as complete neurons. This implementation suppresses downward and horizontal movement.

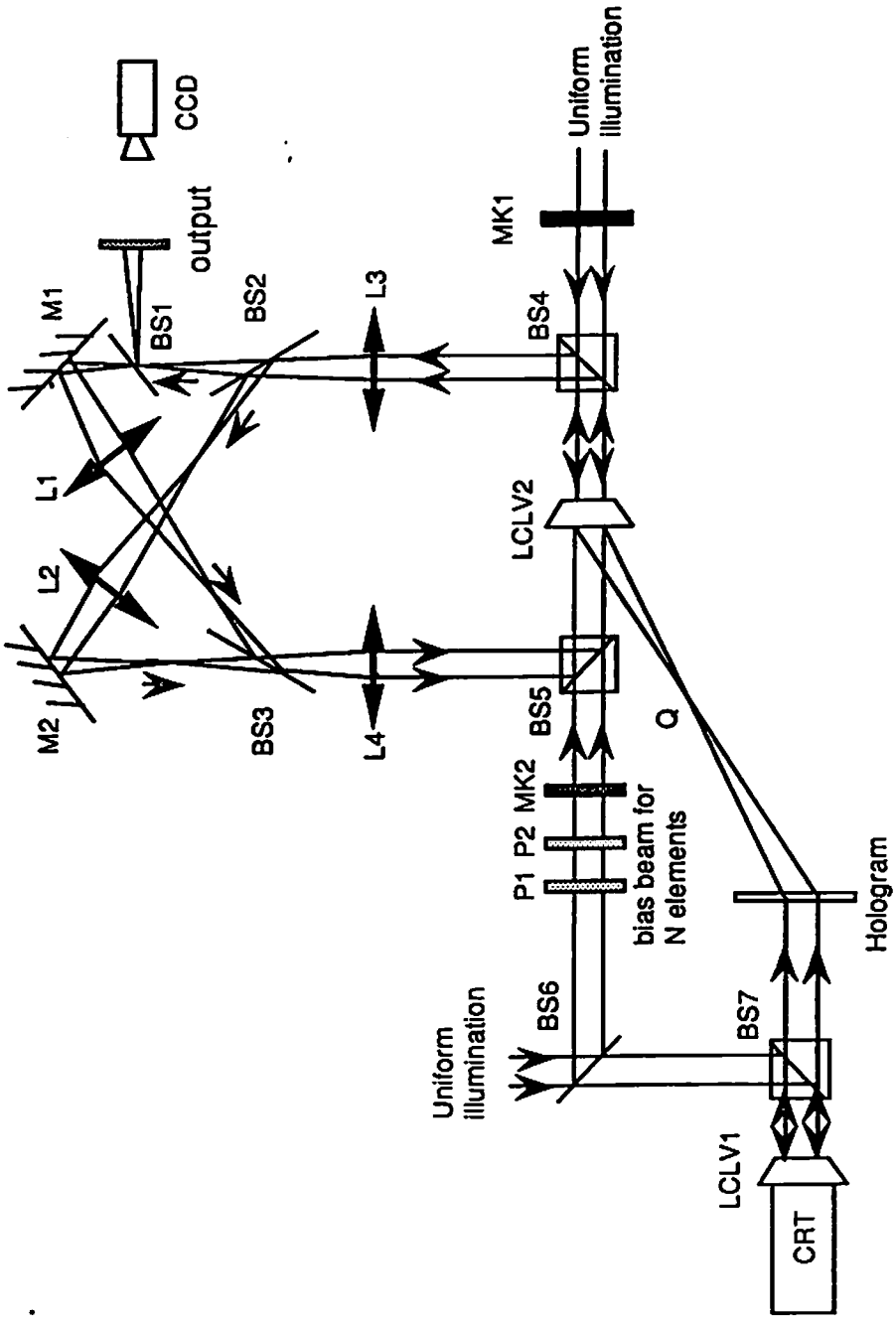


Fig. 19 Optical setup for the implementation of visual cortex operations.

Figure 19: Optical setup for the implementation of visual cortex operations.

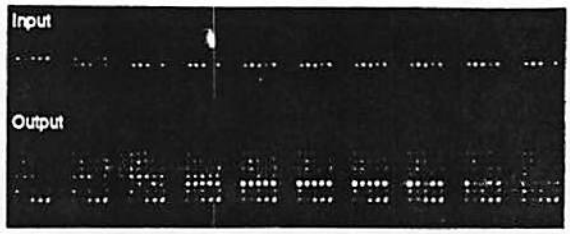


Figure 20: Experimental result of off-transient cells responding to a horizontal moving line input.

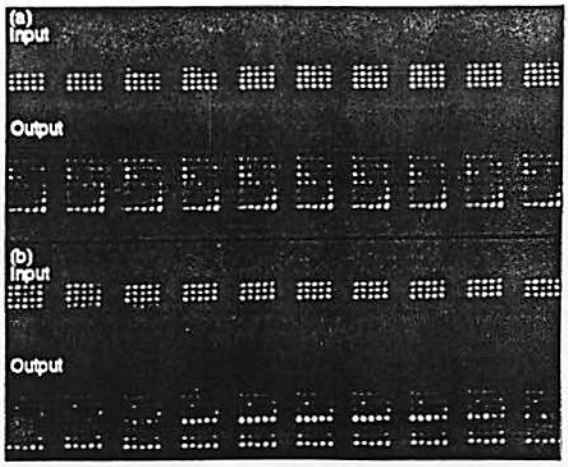


Figure 21: Time sequences of frames showing experimental responses of off-transient cells corresponding to (a) on transient and (b) off transient inputs in the fourth row (of the output). The input is shown spatially inverted relative to the output.

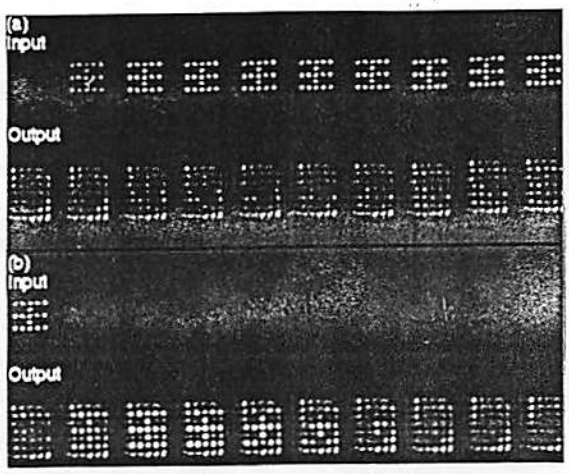


Figure 22: A single time sequence showing responses of off-transient cells to a Chinese character input during and after (a) on transient and (b) off transient.

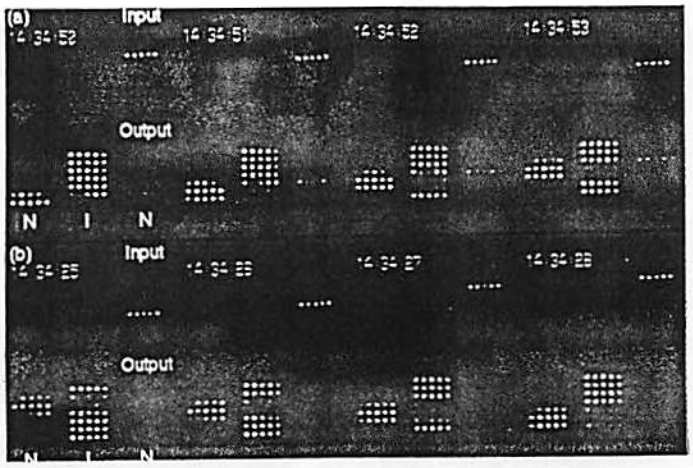


Figure 23: Time sequence showing responses of direction sensitive cells to (a) an upward moving bar, and (b) a downward moving bar. System output is the right-most array of each three-tuple.

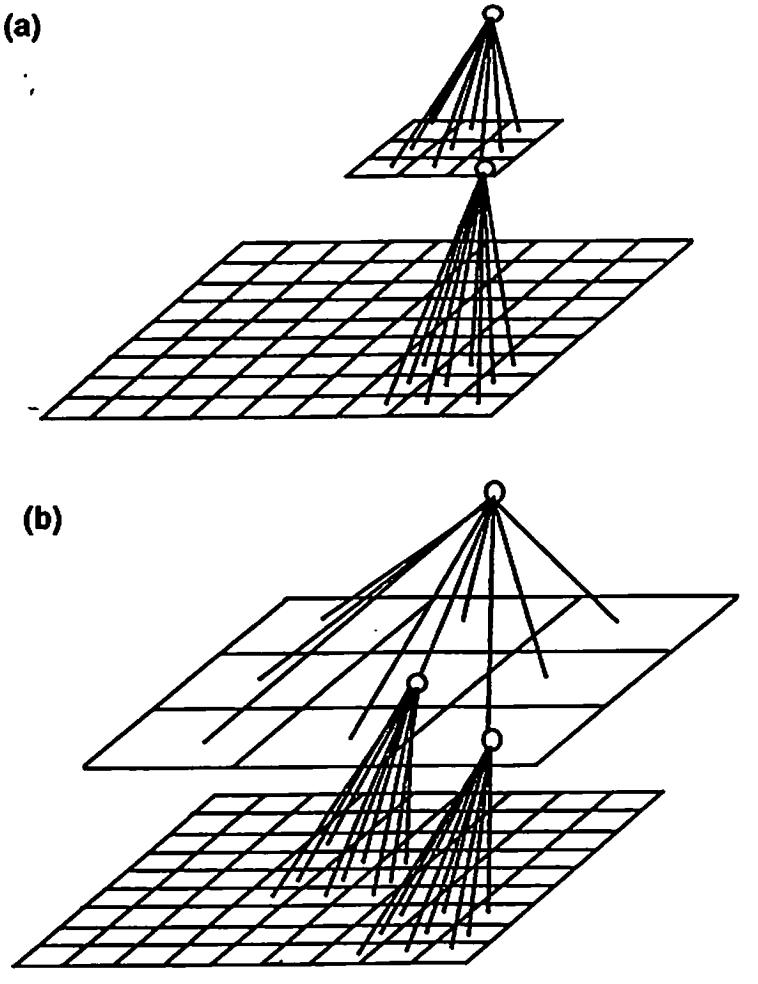


Figure 24: Pyramid structures for a massively connected visual cortex implementation: (a) space variant, and (b) space invariant connections for the neurons in the same layer.

List of Figures