USC–SIPI REPORT #227

Incoherent Multiple Imaging for Parallel Optical Interconnection: Applications in Adaptive Neural Computing

by

Douglas J. Wiley

December 1992

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 400
Los Angeles, CA 90089-2564 U.S.A.

Acknowledgments

The author would like to thank Dr. Isaia Glaser for providing the original inspiration for, and guidance of the work, and Professors B. Keith Jenkins and Alexander Sawchuk of the Signal and Image Processing Institute for the support and guidance they have offered over the years. Special thanks also to Dr. Allan Weber and Toy Mayeda for help with the experiments and computer support. This work was supported by the Air Force Office of Scientific Research under the University Research Initiative Program, contract No. F49620-87-C-0007 and grant No. AFOSR-90-0133, and by NTT corporation.

Contents

List of Figures

List of Tables

Abstract

Optical technology has shown commercial success only for long-distance applications and optical disk storage to date, although recent interest has centered on the development of optical technology for shorter-range interconnections and switching. Optical computing and signal processing have traditionally lacked high-quality spatial light modulator devices, but practical applications may be within reach with the newest optoelectronic technologies. Optical interconnection for computer backplane communications represents one short-term goal, but the paradigm of the programmable serial computer may be supplemented by alternative computing technologies such as parallel neural networks. These networks are 'programmed' by a training session according to 'learning rules'.

This work investigates the potential implementation of neural networks using free-space optical interconnection. A refractive optical experimental system was developed that implemented different 'learning rules' specifying changes to the optical interconnections.

Experimental system components included video equipment (liquid crystal television, miniature CRT monitor) to present input signals to the optics. A lenslet array performed simultaneous, multiple imaging of an array of input signals encoded as an array of light intensities (pixels). An analog weighting was optically applied to each input pixel, and a video camera and frame grabber card in a personal computer collected the output, applying a pointwise nonlinearity. The system implemented a completely-interconnected analog-signal, analog-weight neural network.

The analog-signal experiments included neural networks implementing supervised learning algorithms, where an expected output is compared to the actual output (perceptron). Unsupervised learning networks (competitive learning), and two types of (non-learning) maximum-finding networks were also implemented.

Optical hardware requirements and considerations of optoelectronic implementation are addressed.

An extensive characterization of system performance was accompanied by efforts to modify the neural network algorithms to be more tolerant of the inaccuracies of the optical system. Techniques of compensating for error were also explored using simple software algorithms, hardware, and signal encoding choices. The techniques of quantitative performance evaluation developed are applicable to a wide class of optical interconnection and neural network systems.

The experimental system successfully demonstrated learning in a hybrid optical/electronic system, and the experience of building a working system taught valuable lessons about the cumulative effect of many small inaccuracies (particularly crosstalk).

Chapter 1

Introduction

1.1 Optical Interconnection and Neural Networks

Optical interconnection offers the possibility of lower-energy, higher-throughput, higher physical density connections than is possible in all-electronic systems [1, 2]. Neural networks make particularly communication-intensive demands on hardware, and may be well suited to optical interconnection.

Neural networks have radically different characteristics than conventional programmable computers. They offer fault-tolerance and graceful degradation of performance, and form an approach to the use of parallel computer power [3, 4]. Associative memories with recall time that is constant regardless of the number of stored associations can be implemented. The investigation of neural networks is also a tool of the computational neuroscience community [5]. Neural networks are interesting due to the fact that our brains actually work using a neural network-like structure. While a neural network will not likely replace the computers of today, some applications may be well suited to neural network solutions.

Optical interconnection with analog signals may be a particularly good match to neural networks. Incoherent light intensities sum linearly, without interference. Multiplication of an input by an interconnection weight may also be accomplished by a variable-transmission or reflection spatial light modulator. The

fault-tolerance of neural networks to arithmetic inaccuracy may be well suited to optical systems as well.

1.2 Neural Network Introduction

A very common model of neuron unit function possesses a feature of neuron unit response that is a function of the sum of all inputs. These *additive* neuron models may be implemented by using hardware that performs the multiplication of a matrix of values by a vector of values. The matrix values are interpreted as interconnection strengths, and the inner product of a matrix row w_{i*} and the vector x is the input potential p_i to the ith output neuron unit,

$$p_i = \sum_j w_{ij} x_j. \tag{1.1}$$

Optical systems can operate as matrix-vector multipliers, thus implementing one layer of weighted interconnections.

The terms physical neuron unit, unipolar neuron unit, and gate (in the case of digital circuits), will be used synonomously to describe an optical/electronic hardware unit capable of holding and processing a single numerical information value (analog signal). The terms logical neuron unit and bipolar neuron unit both describe two unipolar neuron units operating together to effectively work with bipolar inputs and outputs.

Notation for neural networks will be written as follows. There are $m + 1$ layers of neuron units, numbered 0 through m, the first with N_0 neuron units, the second with N_1 neuron units, and so on. Interconnection matrix $\underline{\underline{w}}^{(n)}$ connects neuron units in the n-1st layer with those in the nth layer. A matrix element $w_{ij}^{(n)}$ connects neuron unit j in the $n-1$st layer to neuron unit i in the nth layer. The propagation rule will in most cases be an inner product, producing a potential $\mathbf{p}^{(n)}$. An activation rule function $\Psi\{\cdot\}$ acts on the potential to produce a neuron unit activation $\mathbf{y}^{(n)}$. Bold-face $\mathbf{\Psi}\{\cdot\}$ denotes the same function, but acting separately on each element of a vector input and producing the corresponding element of a vector output. Input $\mathbf{x}^{(0)}$ takes the place of $\mathbf{y}^{(0)}$ in the initial layer of input neuron units.

The potential can be expressed in componentwise or matrix-vector product form:

$$\mathbf{p}_j^{(n)} = \sum_{i=1}^{N_n} w_{ji}^{(n)} \mathbf{y}_i^{(n-1)}, \qquad \mathbf{p}^{(n)} = \underline{\underline{w}}^{(n)} \mathbf{y}^{(n-1)}, \qquad \mathbf{y}^{(n)} = \Psi\{\mathbf{p}^{(n)}\} = \Psi\{\underline{\underline{w}}^{(n)} \mathbf{y}^{(n-1)}\} \tag{1.2}$$

The notation is illustrated in figure 1.1.

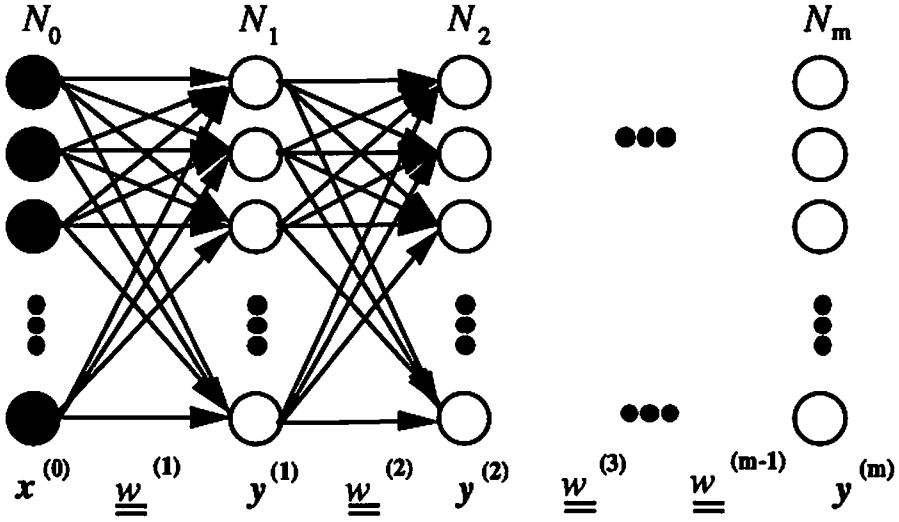


Figure 1.1: Generalized multilayer neural network. Input neuron units are dark disks, other neuron units white circles.

Almost all of the network models that were implemented optically in this work fall into the class of networks with *outer-product* weight update [6]. The weight update value Δw_{ij} is bipolar, and the new weight is the sum of the old weight and the update value, $w_{ij}^{(m+1)} = w_{ij}^{(m)} + \Delta w_{ij}^{(m)}$. The weight update is proportional to the outer product of input $\mathbf{x}$ with an output error value δ_i generated at each output neuron unit: $\Delta w_{ij} = \alpha \delta_i x_j$. The output error is in general a function of the output and potential to that neuron unit, and of a correct output value t_i (in *supervised* learning), $\delta_i = g(p_i, y_i, t_i)$. This class of neural models could potentially employ an optical implementation of the outer product for weight update, combined with local, parallel electronic calculation of output error δ_i.

1.3 Neural Networks as Classifiers

A primary goal of neural network and connectionist learning research is to employ large arrays of highly-interconnected, relatively simple and low-accuracy processors for tasks difficult to implement on serial computers. This approach may offer an effective application of analog optical interconnection.

Many learning algorithms apply to a system performing a *classifier* function [7]. A 'training set' of inputs, representative of all foreseeable inputs, is partitioned into classes. The network accepts an input and (possibly iterates until) one output element (neuron unit) attains a maximal value. The identity of the maximal neuron unit indicates the class of the input as judged by the network.

This classification of input patterns into disjoint sets (*classes*) provides one way of constructing associative memories. After a possible thresholding, the maximal neuron unit can activate an output pattern stored in a successive layer, thereby implementing an autoassociative memory (recalling a reduced-noise version of the input), or heteroassociative memory (recalling a pattern different from the input).

Some of the motivation to use neural networks results from the ability of classifiers to show *generalizability*, defined as performing a useful classification of a pattern that is not in the training set (although we did not specifically analyze these properties experimentally). This ability is described as noise-correction in an associative memory application.

The most common paradigm for training a classifier involves successive presentation of a sequence of input patterns, with an adjustment of the interconnection weights after each presentation. Supervised classifiers have *converged* when each one of the training patterns can be classified correctly. A random order of presentation of the training patterns is usually suggested in the neural network literature. This is intended to minimize the effects of order of the training set. Presenting the same fixed sequence of patterns allows simultaneous training and testing of convergence, for classifier training algorithms that do not change interconnections upon correct classification of a training input. Such a fixed order of training set presentation was used in our experiments. The effects of order were also intentionally examined by repeating the experiments with several different fixed sequences

of the same training pattern. The cycle time of the experimental system precluded large enough training periods required for effectively random order of presentation.

An additional possibility is to sequentially present permutations of the training set, randomly varying the permutation used for each complete presentation. This would allow simultaneous training and testing of convergence, while incorporating a degree of randomness to counteract order-dependent artifacts.

A distinction must now be made between supervised and unsupervised classifiers. Upon learning, unsupervised classifiers form a partition of training set based only on initial conditions (initial weights and parameters such as 'learning rate'). A network can develop classes based partially on the initial conditions. Some applications may employ unsupervised learning for 'feature discovery' whereby random initial conditions are used in the development of classes. These classes are based on differences in the structure of the input patterns. Supervised classifiers, in contrast, attempt to develop a classification conforming to an arbitrarily-specified partition of the training set. Supervised classifier training will generally succeed (converge) only if the classes are chosen to be within the classification capability of the network. Unfortunately, determination of convergence of unsupervised classifiers relies on heuristic judgements in many cases. In both cases, the training period ceases upon convergence or failure of convergence of the network.

1.4 Review of Optical Neural Networks

Several research groups have described optical neural networks that are based on non-holographic interconnection. The system presented herein is a fully dynamic version of film-based lenslet array processor (LAP) systems previously demonstrated by Glaser *et al.* [8-12]. These systems did not implement neural networks *per se*, but demonstrated the matrix-vector multiplication that forms the basis for many neural networks. The coefficients of the matrix hold the synaptic interconnection weights.

The majority of non-holographic optical matrix vector-multipliers can be broadly classified into systems with 1-D input and output arrays, and systems with 2-D input and output arrays. The interconnection weights are stored in a planar

device in both cases. This contrasts with holographic interconnection schemes that store weights in a 3-D medium. Farhat, Psaltis, Prata and Paek laid the groundwork for much subsequent work in optical neural networks work with their implementation of a Hopfield network [13, 14]. Their experimental system used a 1-D LED array for optical interconnection inputs, a 1-D photodetector array for interconnection outputs, a 2-D photographic film interconnection maskfor synaptic weight storage, and electronic subtraction, thresholding and feedback.

Systems that use a non-holographic physically 2-D interconnection weight mask to form optical neural networks often use lenslet arrays. These systems are of two types: *direct* (Fig. 2.1) and *backprojection* (Fig. 2.2) lenslet array processors (this distinction will be discussed in detail later). Direct LAP systems with film-based input, output, and interconnection weight masks have been demonstrated by Glaser *et al.* [8-12]. Miyahara and Farhat demonstrated a 2-D input and output array. This direct system with fixed film interconnection weights, but spatial light modulator (SLM)-based input. It implemented an outer-product associative memory for a radar target-recognition application [15]. Wiley, Glaser *et al.* constructed a video-equipment based direct LAP fully dynamic optical neural network [16]. Yu *et al.* [17, 18] also developed a lenslet array system using video equipment to perform input to and output from the optical system, controlled by a PC. Holographic lenslet arrays and binary optics lenslet arrays complicate this classification. Shin *et al.* describe a neural network that uses holographic lenslet arrays for a quadratic associative memory system [19, 20]. Another diffractive optical system employing phase gratings for fanout [21] has performed simulated annealing at high speed with custom, dedicated optoelectronics.

The lenslet array system described here is a refined version of the system described previously [16, 23]. Incoherent white light illumination is combined with video equipment SLMs and a CCD video camera, forming a completely dynamic system, all controlled by a personal computer (PC).

1.5 Organization

Chapter 1 introduces background on optical technology for interconnection, describes neural networks in general. A literature survey of optical neural networks follows, with a description of each chapter of the dissertation. An itemized list of contributions of the work concludes the chapter.

Chapter 2 describes the lenslet array processor (LAP) architecture in general, detailing the two main types of LAPs (direct and backprojection), and considers possible partially or completely bidirectional systems that combine the features ofthe two main types. Linear algebraic operation of the LAP is outlined, as well as bipolar versions of theses operations. Inner and outer products can be performed, as well as vector sums. A description of communication, computation and control tasks for specific networks is presented. A technique of weight loading and readout that requires no external connections to the weighting units is described.

Chapter 3 describes the experimental system and the approach taken to its development. Hardware and software compensation measures were used to improve the video device performance. Binary neural and digital experiments that employed fixed interconnections are presented.

Chapter 4 summarizes the adaptive neural models implemented optically, multiclass perceptron and competitive learning, and slight variations designed to compensate for the inaccuracy of the optical system.

Chapter 5 concerns performance characterization of the LAP system. A performance figure in terms of operations per second is described, and a parametric model of performance is detailed. Results of system characterization based on the performance model are presented.

Chapter 6 offers conclusions and discussion. It considers the applicability of the work to a more general class of optical systems, summarizes the results of algorithmic variations as a response to optical hardware inaccuracy, and offers discussion on the interplay of characterization and modeling. A high-level design for an optoelectronic neural network is described along with its motivation in terms of neural algorithm hardware requirements.

1.6 Contributions

This work has explored the optical implementation of neural networks on the lenslet array processor architecture. Development of a working, fully dynamic experimental system allowed experimentation with neural algorithm modifications. The experimental system was extensively characterized, and a mathematical model providing the structure for performance evaluation was developed. A more detailed breakdown of the contributions of the work follows.

- Initial steps were taken toward a general methodology of implementation of various (ideally arbitrary) neural network models using lenslet array or holographic interconnection of integrated optoelectronic neuron units.
- Characterization of a variety of commercial devices to obtain useable SLMs for the experimental system.
- Careful study of device response of SLMs used in the experimental LAP system: comparison of acceptable and actual response in terms of uniformity, contrast ratio, and speed.
- Study and partial implementation of compensation techniques for non-idealities of system components.
- Experimental implementation of two schemes for implementation of bipolar analog signals and weights on the LAP system.
- Successful experimental demonstration of digital curcuits and unipolar binary neural networks with up to 16 gates (or neuron units).
- Demonstration of partially-optical learning using perceptron and competitive learning models.
- Analysis of communication requirements and control issues dictated by specific arbitrary neural models, and implications for optoelectronic implementation.
- Development of detailed system-level characterization techniques and an accompanying mathematical model.

- Study and characterization of the most significant system non-idealities for purposes of simulating expected experimental system responses.
- Preliminary description of scaled-up and integrated future LAP systems, and study of some of the limitations to scaleup of the LAP system due to device limitations, imperfect imaging and alignment.

Chapter 2

Lenslet Array Processors

A lenslet array processor is an optical system that performs multiple, simultaneous imaging of a plane of input elements to a plane of weighting elements using a planar array of closely-spaced lenses. The word processor refers to additional components as well, such as active devices (sources, Spatial Light Modulators, and detectors) and additional optics. The taxonomy of Glaser [8-12] refers to two basic types of LAPs; direct and backprojection. Both direct and backprojection LAPs use essentially the same optical system, but use the optical axis in opposite directions. We will describe two basic algebraic operations that can be performed on the direct LAP; matrix-vector multiplication and outer products. The backprojection LAP can also perform matrix-vector multiplication. The use of incoherent light for either LAP makes these operations entirely unipolar. Extensions to bipolar operation will be described later.

2.1 LAP Arithmetic

Both LAP types can implement a general 2-D discrete linear transformation (matrix-vector multiplication). The former operation is followed by a point nonlinearity to incorporate neuron unit or logic gate functionality. The linear transformation performed can be written as

$$F_{l,m} = \sum_{j=1}^{N} \sum_{k=1}^{N} f_{j,k}\, K(j,k;l,m) \quad \text{for } l = 1,\ldots,N,\ m = 1,\ldots,N, \tag{2.1}$$

in which $K(j,k;l,m)$ is the weight (binary or analog valued) connecting input element $f_{j,k}$ with output element $F_{l,m}$. This equation describes N^2 inner products, each of the input array f with one of the subarrays $K(*,*;l,m)$ ($*$ indicates all elements). Each subarray specifies the strengths of connections of all input elements to an output element. Outer product operation results from constraining the N^4 weights to N^2 values and neglecting the summation. The previous linear transformation then provides the product of each input value with each of the N^2 weight values, resulting in N^4 output elements;

$$F_{j,k;l,m} = f_{j,k}\, K_{l,m} \qquad \text{for } j,k,l,m = 1,\ldots,N \tag{2.2}$$

where the N^4 weights are constrained to take on N^2 different values $K_{l,m}$; specifically $K(j,k;l,m) = K_{l,m}$ for all $j,k = 1,\ldots,N$.

The linear transformation operation is used to implement neural networks or digital circuits. A local nonlinear operation performs the neuron unit activation or logic gate function P on each output element $F_{l,m}$. A feedback step then takes the result array at time t as the input array at time $t+1$,

$$f_{l,m}(t+1) = P\{F_{l,m}(t)\} \quad \text{for } l = 1,\ldots,N\ ,\ m = 1,\ldots,N. \tag{2.3}$$

These operations are thus summarized as

$$f_{l,m}(t+1) = P\left\{\sum_{j=1}^{N}\sum_{k=1}^{N} f_{j,k}(t)\, K(j,k;l,m)\right\} \tag{2.4}$$

$$\text{for } l = 1,\ldots,N\,,\ m = 1,\ldots,N.$$

In general the weight mask K is also a function of time, to provide for adaptation in neural networks or reconfiguration of digital circuits.

Input and output data for a neural network implemented on the LAP is organized into N-element by N-element 2-D spatial arrays. The interconnection weight array format is similar but consists of N^2 by N^2 data elements. This N^4 space-bandwidth product requirement is the limiting factor in scaling N to very large values.

2.1.1 Direct LAP

The direct LAP is described in Fig. 2.1, in which part (a) shows a conceptual optical system for its implementation. Figure 2.1(b) is a functional diagram of the system without depicting the lenslet array. There are three planes representing arrays of input, output and (transmissive) weight elements. (In an actual implementation, SLMs are used for input and weights, and detector arrays are used for output.) Plane (B) is always the interconnection weight plane (an SLM). Between plane (A) and (B) lies the lenslet array (L). The direct LAP (Fig. 2.1) operates as follows. The lenslet array performs replication of the N by N elements of plane (A) (implemented by an SLM) N^2 times onto plane (B) to form an N by N array of images, (each containing N by N elements) onto (B). Each copy of the input in this N^2 by N^2 element array lines up with a portion of the N^2 by N^2-element weight array at (B). Each of these will be called a weight submask; $K(*,*;l,m)$ is the (l,m)-th submask. (We note that if $K(j,k;l,m)$ is a kernel of a 2-D linear transformation, each $K(*,*;l,m)$ is one basis function of the transformation). The images of the input array replica are multiplied optically by the transmittance of the corresponding submask and imaged to plane (C). All light corresponding to the transmission of each submask of (B) is then summed (by a spatially integrating detector) to form the (l,m)-th output element $F_{l,m}$ at (C).

2.1.2 Backprojection LAP

The backprojection LAP is described in Fig. 2.2. Optical systems are reversible, so the same optics can image (C) onto (B). The lenslet array performs a superposition summation fan-in instead of a replication fan-out. The input array is now at (C), while the output array (detector) is at (A). The input array at (C) is imaged onto the N^4 transmissive interconnection weight array at (B) so that one weight submask is illuminated exactly by just one input element. Each input element of (C) is sufficiently uniform so that it provides uniform illumination of the entire corresponding weight submask region of (B). The light through each submask is simultaneously imaged onto the entire output array at (A) by the lenslet array, so that the light from each submask *element* is imaged onto just one of the output

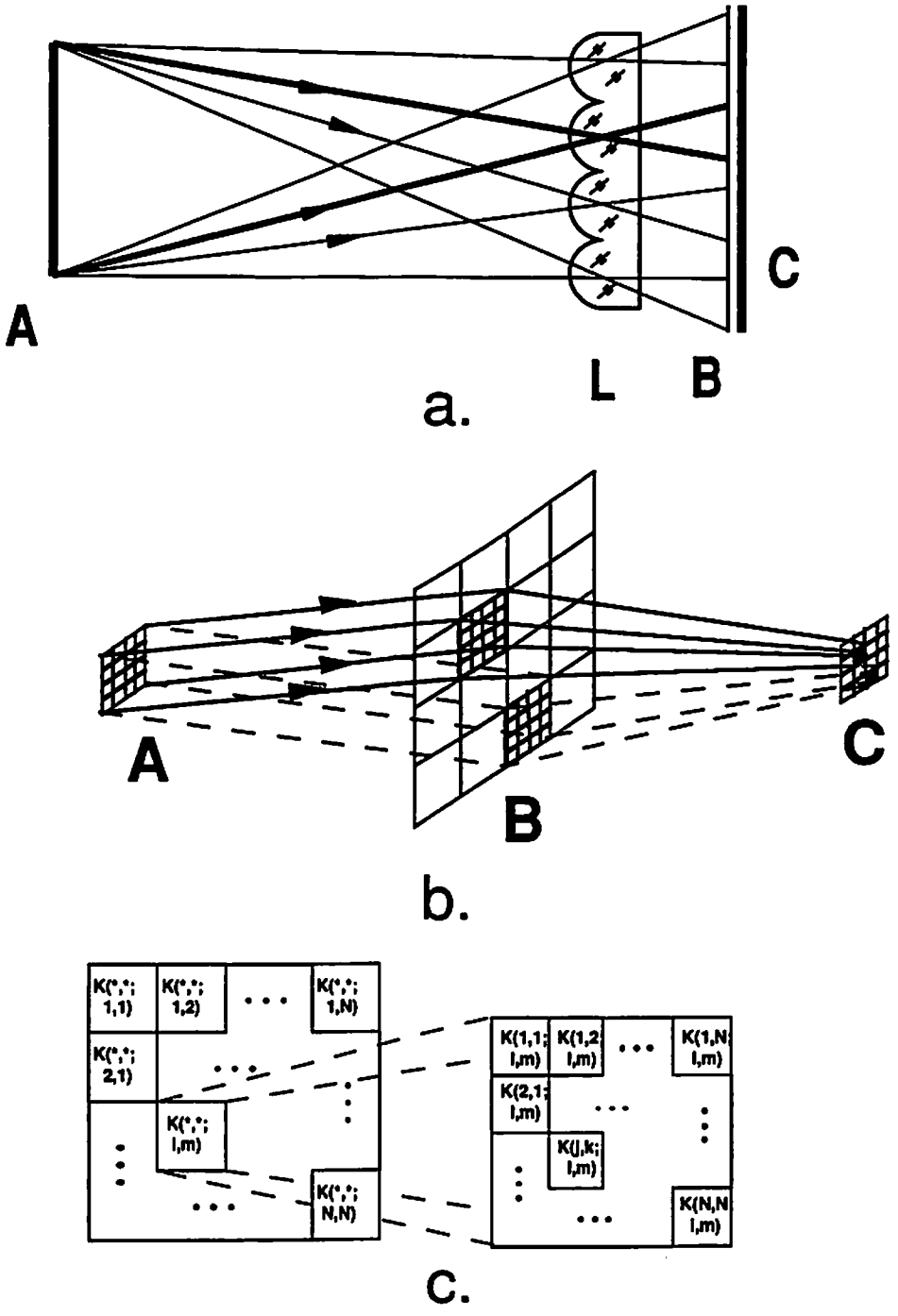


Figure 2.1: A direct lenslet array processor (LAP): (a) physical layout, in which A is the input plane, L is the lenslet array, B is the interconnection weight array and C is the output plane; (b) logical arrangement, in which A, B and C are defined as above; (c) a close-up look at the arrangement of the data of the weight array (B above).

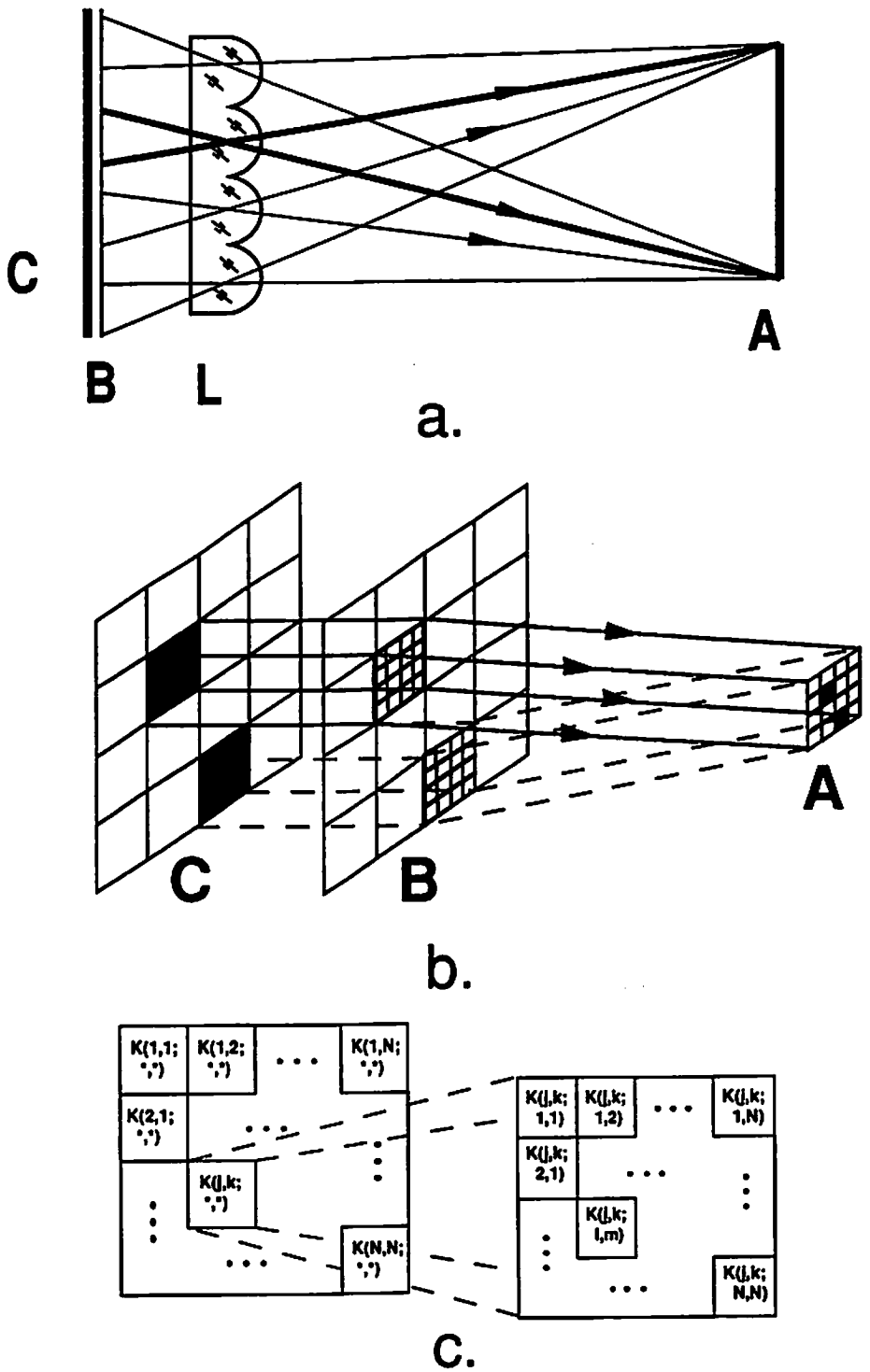


Figure 2.2: A backprojection lenslet array processor: (a) physical layout, in which A is the output plane, L is the lenslet array, B is the interconnection weight array and C is the input plane (note similarity to direct LAP (Fig. 2.1(a))); (b) logical arrangement in which A, B and C are defined as above; (c) a close-up look at the arrangement of the mask (B above) (note differences from direct LAP (Fig. 2.1(c))).

array elements. Note that we can exchange the positions of planes (C) and (B) in the backprojection LAP with no alteration of function as long as the input plane can perform a transmissive multiplication with the values of the interconnection plane [18].

2.1.3 Outer Products on the LAP

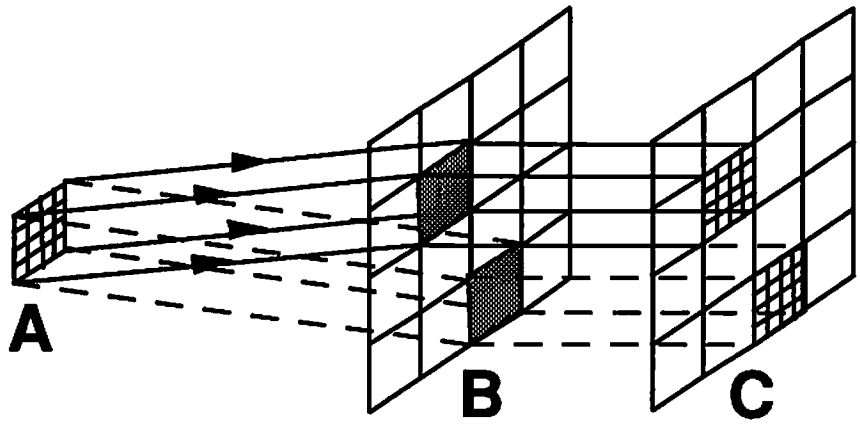


Figure 2.3: Logical arrangement of the outer product operation using the direct LAP, in which A is the input plane and B is the interconnection weight plane. The weights within each interconnection subarray $K(*,*;l,m)$ of plane B have identical values. No summation is performed at the output plane C. (The physical layout is identical to the direct LAP).

Figure 2.3 describes the operation of outer product formation on the direct LAP, where the N^2 output elements at (C) (of Fig. 2.1(b)) are replaced by N^4 output elements (implemented by N^4 detectors). The term *weight submask* will be used here to denote N by N adjacent weights in plane (B) of Fig. 2.1(b) and Fig. 2.2(b). Each weight submask in plane (B) has a uniform value for all N^2 elements. One operand of the outer product is placed on the input array, the other on the submask array.

The inherent superposition of all weight images of the input of the backprojection LAP precludes the outer product operation, unless the lenslet array is discarded entirely. The transpose of the direct outer product then occurs immediately behind the weight plane.

We note however, that the backprojection and direct LAPs are theoretically equivalent in hardware complexity. A detailed analysis of LAP diffraction limits, alignment and geometrical error tolerances is given in Ref. [11].

2.1.4 Bidirectional LAP

The class of adaptive neural networks with feedforward connections and outer-product weight update in general demand a connection from the output back to the weights. This is not provided optically in either version of the LAP. A parallel optical connection would be preferable to an electronic one.

The physical adjacency of the weight and output planes in the direct LAP makes parallel electronic output-to-weight connection feasible (subject to packaging concerns). An output-to-weight connection on the backprojection LAP would involve a bidirectional plane-to-plane imaging path. It is unclear if a bidirectional imaging is advantageous compared to two unidirectional imagings. Beamsplitters and some form of encoding could wavelength or polarization-multiplex a reversible path. Alternatively, designating two adjacent channels to two unidirectional components of the path may make more sense to optoelectronic fabricators.

Multilayer networks trained using the *backpropagation* rule [3] call for fully bidirectional interconnection layers, possibly implemented by a completely bidirectional LAP that operated as a direct LAP during forward propagation of an input, and as a backprojection LAP during backpropagation of output error. Such a network can be analyzed in terms of communication, computation and control requirements, as is done in Section 2.1.6.

2.1.5 Neural Networks and Hardware Requirements

Let us restrict the discussion to learning rules that form weight update values Δw_{ij} by multiplying the error of the output δ_i by the input x_j (yielding an outer-product weight update). Examples of learning rules of this class include Hebbian, least mean squares (LMS), Widrow-Hoff, and backpropagation learning. Thus we consider neural network models in which an additive neuron unit sums the values received from all weighted connections to that neuron unit. Error values δ_i are

found by applying a learning-rule-specific function g which in general depends on neuron unit potential, output and desired output. The constant α is a learning rate parameter. In LMS learning, for example, $\delta_i = \Psi'(p_i)(t_i - y_i)$ [3], where Ψ' is the derivative of Ψ with respect to its argument. Note that in these networks, the communication and computation requirements of the weight update process use only existing interconnections (yielding what is called a local learning rule), perhaps in the reverse (backward) direction. No additional connections are needed. Figure 2.4 depicts one arrangement of this class of neural networks, where the

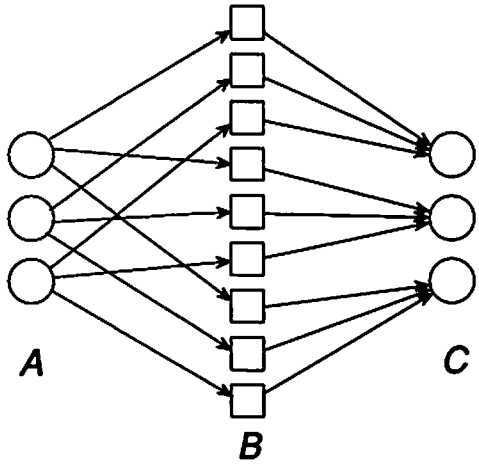


Figure 2.4: Fully-interconnected single-layer neural network with weighting units in plane B depicted as squares, neuron units as circles. Inputs are introduced at A, outputs obtained at C.

weights providing fan-in to a neuron unit are physically adjacent. There are four arithmetic/storage (computation) tasks, and three communication patterns during weight adaptation of a single interconnection layer network.

2.1.6 Communication, Computation and Control Requirements

The four arithmetic/storage tasks are as follows. A single incoming signal from plane A must be multiplied by a stored weight value in the weight pixels in plane B, and sent on to output neuron unit pixels in plane C. Each output neuron unit pixel must sum many such weighted signals. Weight update then involves

multiplying each weighted signal by an error signal sent against the arrows of Fig. 1 (C to B), and subsequently adding that weight update value to the stored weight. The simplification of linear arrays is diagramed, but depiction of planar arrays of weight pixels with either linear or planar arrays of neuron unit pixels is intended.

The three patterns of communication in this single interconnection layer network are fan-out of input from plane A to plane B, summing fan-in (B to C), and fan-out of error (C to B). The last two patterns use identical connections, but in opposite directions. In these networks, if weights providing fan-in to an output neuron unit are located together with that neuron unit in a 'fan-in superpixel', the bidirectional connection is local to that superpixel and the connections between superpixels are unidirectional. On the other hand, grouping weights that provide fan-out from a neuron unit pixel (in a single-interconnection layer network) yields locally unidirectional and globally bidirectional communication. These results hold even in a 'crossed finger' crossbar layout [22] if locality is re-defined to be along a line segment instead of within a planar patch.

Multiple interconnection layers (corresponding to a cascade of single-layer networks such as in Fig. 2.4) can be 'programmed' by the backpropagation learning rule. The communication advantage of global unidirectionality of fan-in weight superpixels then applies only to the first layer. Additionally, a fourth pattern of communication and a fifth arithmetic/storage task emerges. The error arriving at plane B must be multiplied by the stored weight, and be sent backward, along existing connections, from plane B to plane A.

Control requirements include orchestration of the input/update cycle; presenting inputs in random or other sequential order, waiting for the propagation delay time of the system before weight update. The input, weight and output units must all receive a global clocking signal in order to synchronize their operations correctly. Convergence of the training session must also be detected, or an unsuccessful training session must be abandoned after a predetermined time limit.

2.1.7 Operation with No External Weight Connections

Given that a fully interconnected LAP neural network has N^2 inputs and outputs, but N^4 weights, it may be desirable to avoid electronic interconnections to each weight. It is possible to load and read out the weights using the existing forward

connections and an output-to-weight connection. Outer product weight update rules require this back connection already. A penalty is paid in time, however. N^2 forward connection steps must be sequentially taken to load all N^4 weights. Multiple-layer networks have often-unpredictable and lengthy training times that will limit practicality more than the N^2 loading/readout time.

Assuming an initial global clearing of weights to zero and the outer-product weight update mechanism described, we have two choices of ways load weights. The desired fan-in weights to a neuron unit x_i are placed on the input, and a pattern with only one nonzero value is sent as output error signal δ_i (Fig. 2.5(a)). Cycling through all patterns will then load the fan-in weights to each output neuron unit. Alternatively, one input x_j can be nonzero, with the desired fan-out weights of that input neuron unit sent back to the weights mechanism as δ_i (Fig. 2.5(b)).

Readout of the weights is easily accomplished by a sequential presentation of the one nonzero pattern set and reading out the fan-in weights of that input neuron unit (Fig. 2.5(c)). Alternatively, we can use the backward direction of a fully bidirectional network and one nonzero output error signal δ_i to read out N^2 weights one at a time (fan-in weights to that output neuron unit) (Fig. 2.5(d)).

2.2 Matrix Arithmetic on the LAP

Linear transformations over m-dimensional finite arrays, containing $N \times N \times \cdots \times N$ elements and an N^{2m} element kernel, are equivalent to simple matrix-vector multiplications of an $1 \times N^m$ element vector with an $N^m \times N^m$ element matrix. The choice of representation lies entirely as a matter of convenience: available hardware and the natural dimensionality of the input and output signals. For the LAP, $m = 2$.

2.2.1 Unipolar Arithmetic

Inner products can be performed on both types of LAP system. Vector sums can be obtained from inner products by a transformation on the data before the same inner products. Outer products can be obtained by using the direct LAP,

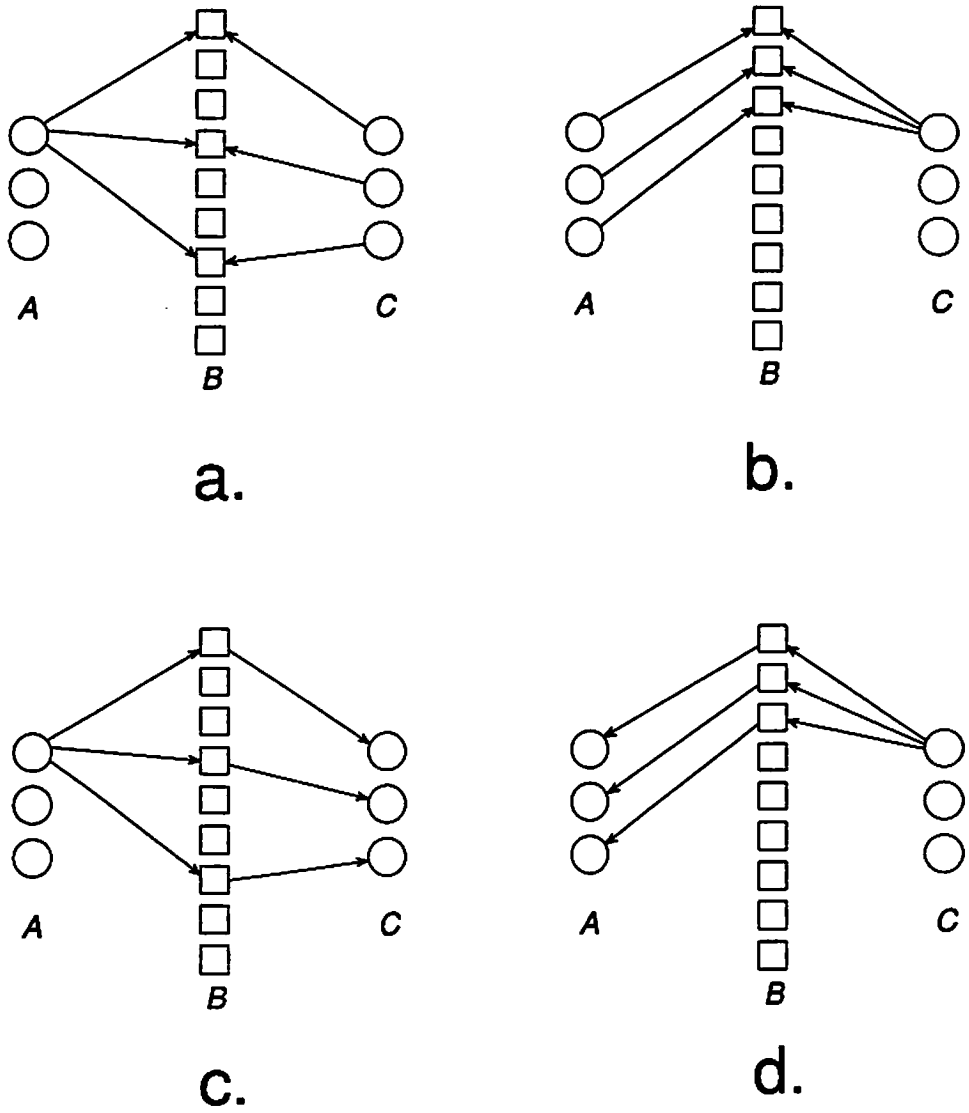


Figure 2.5: Weight loading and readout using existing connections of the network and outer-product weight update (arrows show a non-zero signal); (a) shows loading of fan-in weights to a single output neuron unit with a nonzero error signal, (b) shows loading of fan-out weights to a single input neuron unit with non-zero input value, (c) shows readout of fan-out weights, (d) readout of fan-in weights.

constraining the N by N weights in each submask to be identical in value, and replacing the N by N array of detectors with an N^2 by N^2 array of detectors. Since we use incoherent light, these operations have a unipolar (non-negative real) form (and two methods of bipolar implementation will be discussed for each).

A vector sum may be mapped into a matrix-vector multiplication as follows to calculate a sum of vectors $\sum_{k=1}^{n} \mathbf{a}^{(k)}$. Form a new set of vectors $\mathbf{b}^{(j)}$ such that $b_i^{(j)} = a_j^{(i)}$, then

$$\hat{\mathbf{1}}^T \mathbf{b}^{(j)} = \sum_{k=1}^{n} b_k^{(j)} = \sum_{k=1}^{n} a_j^{(k)}. \tag{2.5}$$

Thus

$$\mathbf{c} = \sum_{k=1}^{n} \mathbf{a}^{(k)} \quad \text{where} \quad c_j = \sum_{k=1}^{n} a_j^{(k)} = \hat{\mathbf{1}}^T \mathbf{b}^{(j)}. \tag{2.6}$$

Although this mapping is valid for either bipolar or unipolar matrix-vector multiplication, consider the case of unipolar implementation upon an incoherent matrix-vector multiplier such as the direct LAP. The mapping then corresponds to taking the 'folded transpose' of a weight mask with one **a** vector upon each submask. By placing $\hat{\mathbf{1}}$ on the LAP input and one **b** upon each submask, we can perform the sum of **a** vectors on the LAP. Thus $N^2 = n$ terms in the summation of Eq. (2.6) can be computed in one step by properly encoding the vectors **a** onto the mask.

Implementation of vector summation upon the backprojection LAP requires use of the 'folded transpose' of the weight array used for the direct LAP (as stated for the more general case previously). Placement of one **a** upon each submask, with $\hat{\mathbf{1}}$ as input gives the same vector sum output array **c**.

Unipolar outer products of two N^2 length vectors (laid out as N by N 2-D arrays spatially) can also be performed on the direct LAP, leading to an N^2 by N^2 element result array. Let **c** and **d** be vectors of this form. Place **c** upon the input to the direct LAP. Now set every weight in the (l, m)-th interconnection submask to the value of the (l, m)-th element of **d**. Capture of the resulting output image as an N^2 by N^2 array then contains the product of every element of **c** by every element of **d** exactly once, i.e. the outer product $\mathbf{cd}^T$ (in which **c** and **d** are column vectors) but represented physically in folded format (Fig. 2.3).

2.2.2 Bipolar LAP Matrix Arithmetic

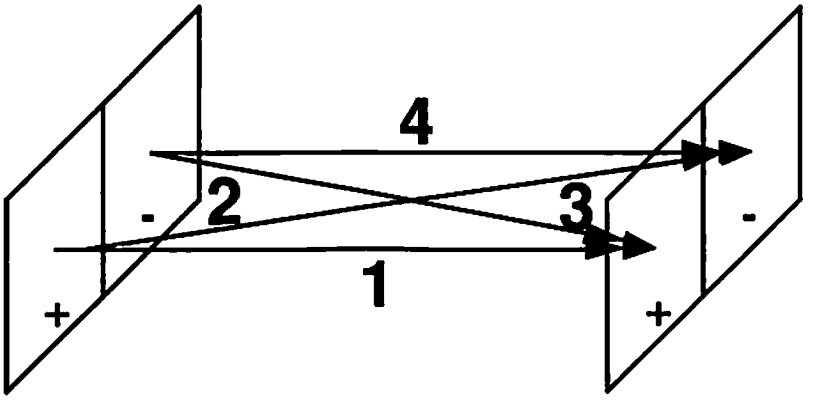
Bipolar operations can be accomplished by several coding schemes [8, 12]. Two of the methods are time multiplexing and space coding. Time multiplexing is accomplished by using a time sequence of positively and negatively interpreted signals, while space coding designates negative quantities by certain spatial positions in the data arrays. Space coding introduces specific dependencies accross the input, output and interconnection weight arrays, while time multiplexing requires intermediate storage and time-sequencing mechanisms (or some form of phase-lock detection). The discussion will concentrate on space coding. Specific additional operations introduced by bipolar space coding will be detailed for the three matrix arithmetic operations. Electronic subtraction is assumed for both methods. General bipolar multiplication can be represented by 4 distinct cases (Fig. 2.1).

case \ sign	Input	Weight	Output
1	+	+	+
2	+	-	-
3	-	+	-
4	-	-	+

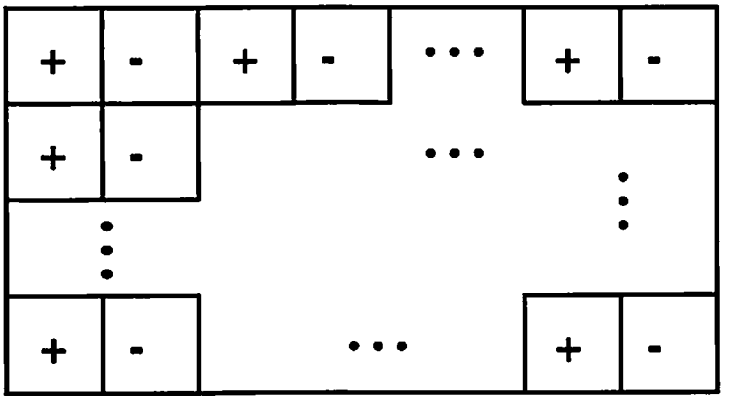
Table 2.1: The 4 cases of bipolar multiplication sign where Input×Weight=Output (Input, Weight and Output are bipolar).

A time multiplexed system performs one optical cycle for each of the 4 cases of bipolar multiplication. This allows use of N^2 elements in four temporal cycles.

In space coding bipolar operation, all 4 cases are performed in one optical cycle using a pair of physical (non-negative real) elements for one 'logically bipolar element', obtaining $N^2/2$ bipolar elements from N^2 physical ones (see Fig. 2.6(a)). We may thus designate alternating elements of the input and output array horizontally as positive and negative (Fig. 2.6(b)). We define positive and negative



a.



b.

Figure 2.6: Bipolar arithmetic is accomplished by using two physical elements for input and output each for one signed element. The four physical weights that make up a signed logical weight are used two at a time for each case of sign of the weight (with the other two set to zero; (a) shows the 4 physical weights; each arrow connects the input pair on the right with an output pair on the left. Paths 1 and 3 are used for a positive weight, 2 and 4 for a negative weight. (b) shows the resulting halving of elements as they appear on the input and output planes.

indices as follows: $p_z^+ = 2z - 1$ and $p_z^- = 2z$. A signed logical input element $f_{j,s}^{(L)}$ consists of two unipolar input elements f_{j,p_s^+}, f_{j,p_s^-}, such that $f_{j,s}^{(L)} = f_{j,p_s^+} - f_{j,p_s^-}$. This element is *normalized* if $(f_{j,p_s^+})(f_{j,p_s^-}) = 0$. The input array is then indexed by (j,s), $j = 1,\ldots,N$, $s = 1,\ldots,N/2$. A signed logical output element $F_{l,r}^{(L)}$ is defined similarly for $l = 1,\ldots,N$, $r = 1,\ldots,N/2$, $F_{j,r}^{(L)} = F_{j,p_r^+} - F_{j,p_r^-}$. Now signed interconnection weights can be identified. Define a signed logical weight $W^{(L)}(j,s;l,r)$ to be the difference of two positive quantities

$$W^{(L)}(j,s;l,r) = W^+(j,s;l,r) - W^-(j,s;l,r). \tag{2.7}$$

We find that the four physical weights are related to those of Eq. (2.7) by:

$$K(j,p_s^+;l,p_r^+) = K(j,p_s^-;l,p_r^-) = W^+(j,s;l,r) \tag{2.8}$$

and

$$K(j,p_s^+;l,p_r^-) = K(j,p_s^-;l,p_r^+) = W^-(j,s;l,r). \tag{2.9}$$

While Eqs. (2.7), (2.8), and (2.9) hold also when the input or the interconnection mask weight values are unnormalized, normalization results in best use of the available dynamic range.

In both LAP types, the four weights of a signed logical weight (each corresponding to one arrow in Fig. 2.6(a)) occur in two adjacent pairs, in two different but adjacent submasks. For both types of LAP, once weights are written and input is presented, one optical cycle performs all four cases of bipolar multiplication in one time step. The result will generally come out unnormalized. It is useful to re-normalize at this point before feedback to the input device in iterative processing.

Derivation of the bipolar inner product involves substitution of the following into the definition of the signed logical output element:

$$F_{l,p_r^+} = \sum_{j=1}^{N} \sum_{s=1}^{N/2} f_{j,p_s^+} K(j,p_s^+;l,p_r^+) + f_{j,p_s^-} K(j,p_s^-;l,p_r^+) \tag{2.10}$$

$$F_{l,p_r^-} = \sum_{j=1}^{N} \sum_{s=1}^{N/2} f_{j,p_s^+} K(j,p_s^+;l,p_r^-) + f_{j,p_s^-} K(j,p_s^-;l,p_r^-) \tag{2.11}$$

so formation of an output element involves

$$F_{l,r}^{(L)} = F_{l,p_r^+} - F_{l,p_r^-} = \sum_{j=1}^{N} \sum_{s=1}^{N/2} K^{(L)}(j,s;l,r) f_{j,s}^{(L)}. \tag{2.12}$$

Normalized space-coded bipolar inner products involve the following operations beyond those needed for unipolar ones. The sign of input values determines the non-zero unipolar input element of the pair. Likewise, the sign of each weight value determines the two non-zero weights of the four controlling the connection of an input element pair and an output pair. Normalization of the output element array then requires taking the difference of the two components. In practice, one might compare the magnitude of the positive and negative sums and compute the absolute value of the magnitude difference.

Space-coded bipolar *outer* products involve a step of redundancy reduction in addition to normalization. The two inputs to the computation are space-coded to the input and interconnection weight array as in the unipolar outer product. Each input is a vector of length $N^2/2$. Outer product operation then yields an array of intermediate results, each consisting of four unipolar components. These are combined to obtain one bipolar value;

$$F_{j,s;l,r}^{(L)} = F_{j,p_s^+;l,p_r^+} + F_{j,p_s^-;l,p_r^+} - F_{j,p_s^+;l,p_r^-} - F_{j,p_s^-;l,p_r^-}. \tag{2.13}$$

In both types of LAP, the two positive result locations are $F_{j,p_s^+;l,p_r^+}$ and $F_{j,p_s^-;l,p_r^+}$ and the two negative result locations are $F_{j,p_s^+;l,p_r^-}$ and $F_{j,p_s^-;l,p_r^-}$. These four components will occur in adjacent *groups* of N by N elements corresponding to copies of the input (weighted by one submask) but will not occur in adjacent *elements.*

After this redundancy reduction and normalization, we obtain $N^4/4$ signed elements from the outer product of two $N^2/2$ length bipolar vectors as expected.

Chapter 3

Experimental System and Binary-Signal Experiments

This chapter will describe an experimental LAP system and some of the ensuing design and implementation issues.

3.1 The Experimental System

Figure 3.1 shows the optical portion of the experimental direct LAP system. The main components of the system are the spatial light modulators (SLMs) for the input and interconnection weights, and the detector of the optical output of the system. The input SLM is a commercial liquid crystal television (LCTV). The interconnection weight SLM is an optically-addressed liquid crystal light valve (LCLV), which was controlled by a small CRT. The LCLV operates in reflection mode. The output detector was a video camera. The entire optical system functions as follows: the input SLM (LCTV) impresses the system input onto the light from an incoherent white light source S (incoherent lamp). The lenslet array, LA, creates multiple images of the input pattern, which goes through lens L1, and passes through the polarizing beamsplitter PB to the LCLV . On the other, 'write' (right in the figure) side of the LCLV, a CRT displays the entire interconnection weight array pattern, which is imaged by lens L3 and controls the reflectivity of the 'read' (left in figure) side of the LCLV. The multiple images of the input patterns

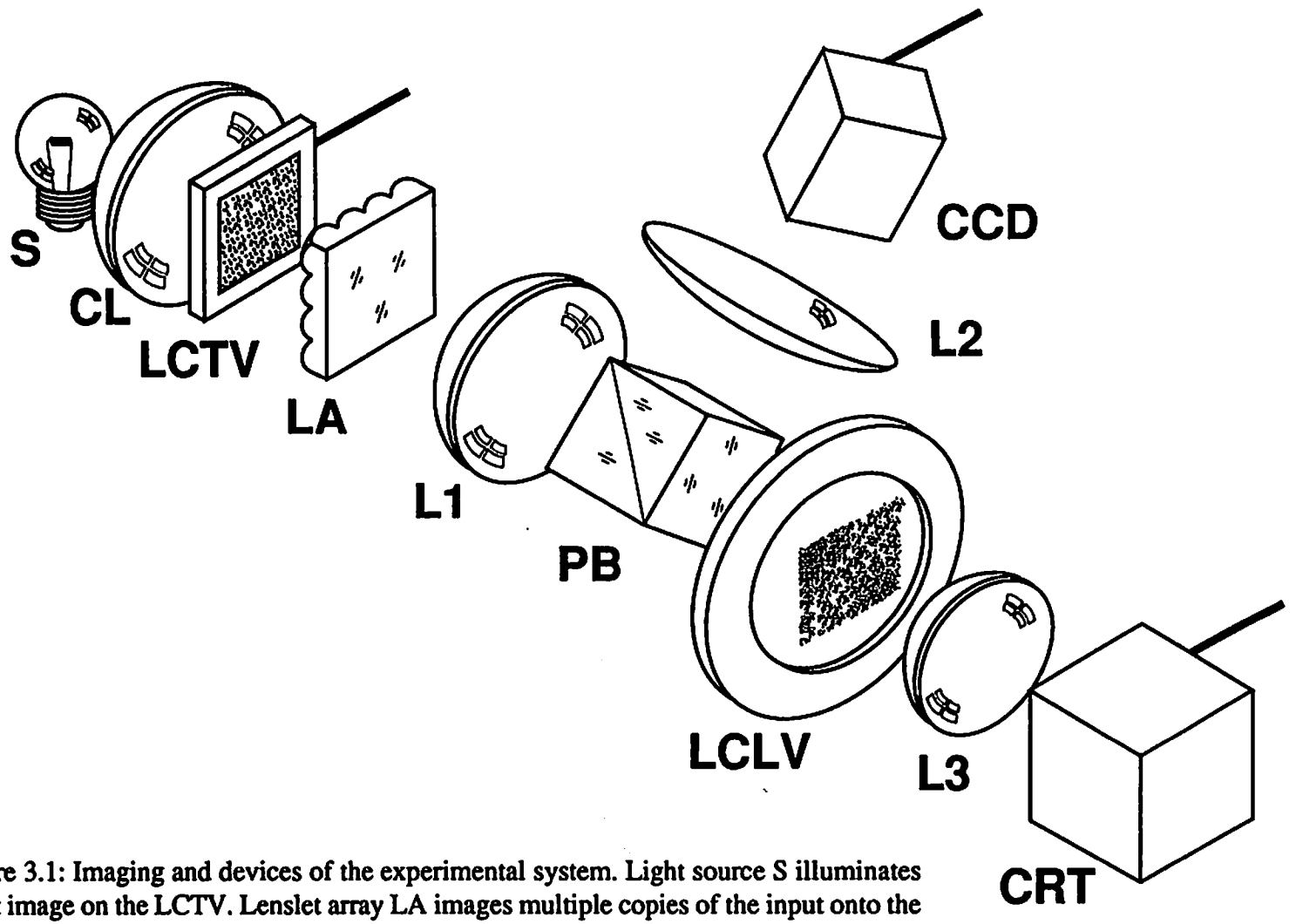


Figure 3.1: Imaging and devices of the experimental system. Light source S illuminates input image on the LCTV. Lenslet array LA images multiple copies of the input onto the reflective SLM LCLV, which then are collected on the CCD camera. External polarizers P1, P2 (not shown) increase extinction ratio of the polarizing beamsplitter PB in white light.

formed by the lenslet array and relayed by lens L1, are re-imaged on the 'read' side of the LCLV, multiplied by the image of the weight pattern, and reflected back at the beam splitter PB. This new image is now reflected by the beamsplitter and imaged by lens L2 into the CCD camera.

An 'ideal' LAP would have the mask sit directly in the image plane of the lenslets, creating a compact and rigid system. Unfortunately, the use of a reflective LCLV forced us to have a beamsplitter in the system, and relay optics to get enough space for it, as shown in Fig. 3.1.

The electronic and control part of the experimental direct LAP system is depicted in Fig. 3.2. Two video "frame grab" cards (FG1 and FG2) inside a personal

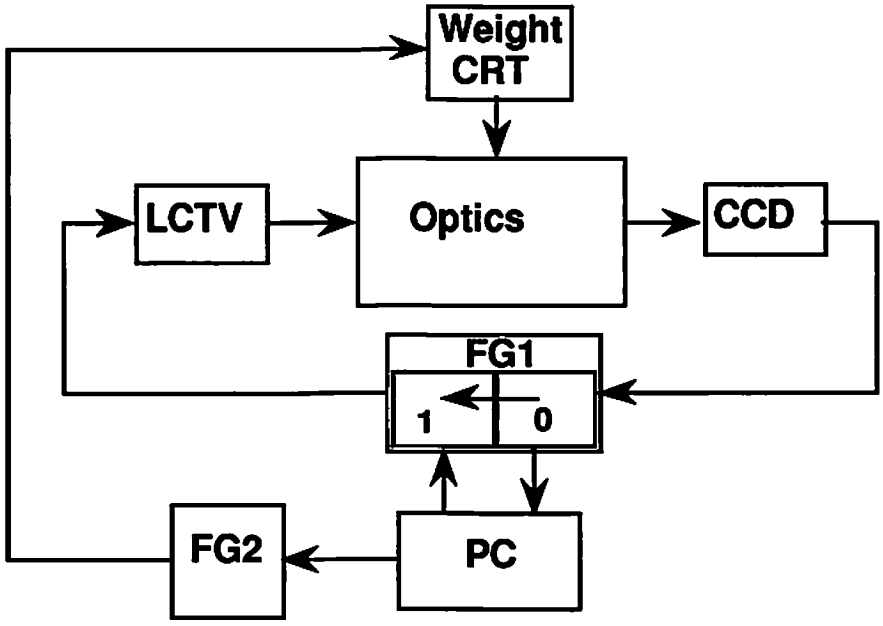


Figure 3.2: Electronic operations of the experimental system. Frame grabbers perform collection and generation of input, output FG1 and interconnection FG2 images.

computer (an AT clone) perform output of interconnection weight array and input images to the SLMs (LCTV and CRT), and collect the image from the CCD camera. Thresholding and summation operations occur on the video cards under computer control (although future systems could use area detectors or optical superposition for summation).

As shown in Fig. 3.2, operation of the system is performed as follows:

1. Update or initialize the LCTV with the image containing input values of the N^2 input elements, by placing image on buffer 1 of frame grabber card FG1.

2. Update or initialize the C.R.T. with the image containing all N^4 interconnection weights, by placing image on the second frame grabber FG2.

3. Optical subsystem performs the interconnection and multiplication operations.

4. Capture result image from the CCD camera into buffer 0 of FG1.

5. Perform summation or averaging step within buffer 0 of FG1 over each of the N^2 submask images that will contain the output neuron unit result. This step could be done optically in future systems.

6. Pass result image from buffer 0 of FG1 through the point nonlinearity lookup table into buffer 1 of FG1, updating the input values of the N^2 output elements.

7. Return to step 2 if iteration is not completed, or read out final result.

3.2 Characterization, Compensation and Optics Specifics

Building a working experimental system introduces many errors and inaccuracies. Consider inaccuracies of two groups of components: active devices and passive optical components. Computer techniques were employed to simulate the results of more perfect components that a larger-scale development effort could provide. Computer-optimized multi-element lens design and manufacture would allow development of passive optics with better performance than that of the experimental system. The SLMs that will be available in the near future should also work more effectively than the video devices we used for the experiments.

Compensation of imperfection was accomplished by a methodology of testing, correction, and re-testing. We endeavored to restrict compensations to those that would likely be obviated by using higher quality components. Additionally, it

was desirable to get a start on the more demanding requirements of future analog operation.

Video devices using standard National Television System Committee (NTSC) signals were chosen. Device linearity, contrast ratio, dynamic range, sensitivity or transmission (reflection) efficiency and isolation of sub-portions of the device were important issues. Experiments found the optimum setting of critical controls, measured Automatic Gain Control (AGC) tendencies, and measured the success of anti-AGC compensations.

The passive optical components contribute errors in the form of nonuniform illumination, stray light, vignetting, and cumulative effects of abberations. Non-uniform light distribution will also be caused by a non-perfect illumination system. Experiments were conducted studying extinction ratio of LCLV-'reading' reflective optics in white light, and real-time statistics were used during mechanical alignment of the light source for optimum illumination uniformity.

Overall equalization of weighted interconnections resulting in the equivalent of a fixed film mask was also used. This procedure was iterative, and corrected imperfections of both the active devices and passive components.

3.2.1 Passive Components and Illumination

Imaging and illumination within the system also suffered from vignetting and nonuniformity imposed by imperfections in the optics. While the use of polychromatic incoherent illumination provides freedom of coherent artifacts like speckle and diffraction pattern noise, we had to be more careful with chromatic aberrations and with the shallower depth-of-focus of optics with diffuse light. Polychromatic optical systems impose different demands on optical components and systems.

The computer program that controls the system as whole must sample the correct regions of the camera image to gather the output of the system. This mechanism of gathering output was generalized to admit a system construction and diagnostic tool displaying running statistics of the values of the different output elements. Critical alignment and adjustment of the optics of the system can be done while watching a variety of system diagnostics.

The light source we used was an incandescent 100-watt quartz Tungsten-Halogen bulb with condenser optics. We used a "hot mirror" to eliminate infrared

light that is not well modulated by the SLMs and causes excessive chromatic aberration.

The measured extinction ratio of the cube polarizing beamsplitter in polychromatic diffuse light (using also the built-in polarizer of the LCTV panel) was found inadequate (about 10 to 1); augmentation of the polarizing beamsplitter with two additional glass-sandwiched sheet polarizers increased the extinction ratio of the beamsplitter by an order of magnitude, to more than 100 to 1 at the expense of light efficiency. The LCLV operates by rotation of polarization, so the extinction ratio of the beamsplitter sets a limit on the dynamic range of the LCLV. The additional polarizers restored the assembly to acceptable performance.

We used a plastic molded lenslet array with lenslets of focal length of about 4 mm, and pitch of 1 mm [24]. The small size allows acceptable performance in white light with a lenslet of singlet design [11]. While the images produced by the lenslets exhibit some barrel distortion and other image defects, they are adequate for our application.

Vignetting and other effects cause loss of intensity in the corners of the transmitted images and unequal illumination between images. This occurs when imaging the LCTV by the lenslet array. This fall-off can be minimized by putting the LCLV father away from the lenslet array, which in turn increases the spacing between the individual images at the weight SLM. This results in increased space-bandwidth product (which must be delivered by the CRT+LCLV system), for a given N.

Photographic lenses, both similar photographic quality 50mm f/1.4, were used for the CCD imaging and CRT imaging. (L2 and L3 in Fig. 3.1). The relay lens was actually a pair of inexpensive doublets (L1 in Fig. 3.1). The use of a high quality lens for this role would have undoubtedly alleviated some of the problems we experienced.

The influence of aberrations in limiting the number of interconnections has been described by Sakano et al. [30]. The scale of our system was such that aberration was not a major problem, although both the weights and detector positions (in their respective planes) were perturbed from a rectangular grid to

line up with the input-copy array images. The signal spot size was also large relative to the resolution limit of the lenslet array.

We also briefly experimented with other LAP optical configurations. Putting the beamsplitter ahead of the lenslet array (between S and LA in Fig. 3.1) facilitates a method of optical summation, but resulted in excessive flare and low contrast. Using optical instead of electronic summation was done in earlier systems [9] by defocusing, but the aberrations contributed by the relay optics in our system (L1 in Fig. 3.1) made that method unreliable.

3.2.2 Active Devices

Our choice of using consumer-grade video components made individual correction of each device necessary; we also used measurements of the characteristics of the system as a whole in evaluating device performance. The two most important experimental tasks were selection of optimum critical control settings and avoidance of AGC tendencies. Most consumer video display devices come with AGC or a DC restore circuit that cannot be turned off. Special compensation, discussed below, was used; note that this kind of compensation would not be required for an SLM designed specifically for optical computing.

The response of the video camera was characterized using a calibrated step tablet. This video camera and a standard laboratory optical power meter measured the transmission (or reflection) responses of the two SLM devices as a function of the numerical gray level input to the computer video card that controls it. As noted above, an LCTV panel was used for the input signal while a combination of an optically driven LCLV and a CRT were used for the interconnection mask. Each of these had one or more critical adjustment controls; a brightness adjustment for the LCTV, and brightness and contrast adjustments for the CRT. Figure 3.3 shows this data as a family of response curves for our LCTV. Similarly, the CRT is represented by a set of 2-D graphs on a range of the settings, as shown in Fig. 3.4. We then used former calibration curves for the LCLV to obtain the resulting response of the CRT+LCTV combination.

A discussion of each device in turn follows. Each was calibrated and characterized. Maximum contrast ratio was the most important characteristic since

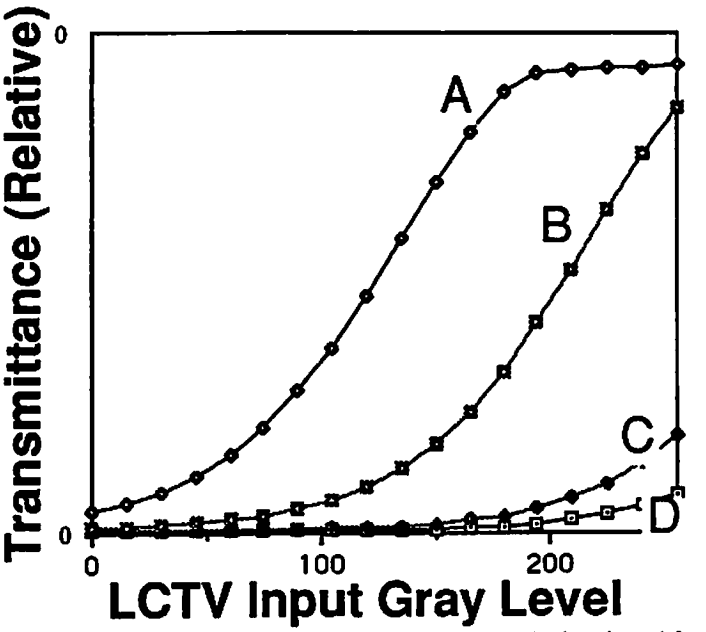


Figure 3.3: Transmission intensity characteristics of the liquid-crystal TV input device. The contrast control was calibrated for selection of the highest dynamic range and linearity region. Each curve refers to a different setting of the controls.

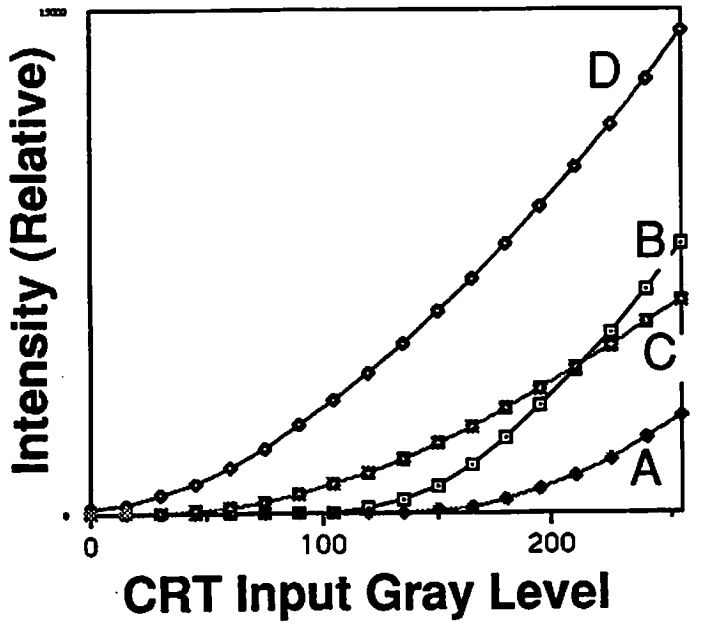


Figure 3.4: Emission intensity of the CRT interconnection weight device. Brightness and contrast controls must both be set for the region of highest linearity and dynamic range. Each curve refers to a different setting of the controls.

our illumination intensity was sufficient. A set of critical adjustment parameters can then be found leading to maximum contrast ratio (with adequate transmission/reflection/efficiency). Dynamic range requirements on each device are found by examining the process of multiplication. Resolution of the product of $m+1$ levels $0,\ldots,m$ of two terms requires distinguishing m^2+1 levels $0,\ldots,m^2$. This range requirement is multiplied by the number of summations performed in each LAP inner product; k products of m^2+1 levels have a range requiring $k \times m^2+1$ resolvable levels $0,\ldots,k \times m^2$. This requires a contrast ratio of $k \times m^2$ as a rule of thumb. Fan-in of k binary levels requires resolving $k+1$ levels $0,\ldots,k$. One of our experiments had a fan-in of 5 (the parity-checking network).

The LCTV we used was taken from a consumer-type hand-held color videocassette recorder, the screen measures 5 by approximately 7 cm [25]. The black-and-white consumer hand-held LCTVs that we tested had unacceptably low contrast ratio. Although the display we used has color capability, this feature was not used. Contrast ratios of over 100 to 1 were measured.

An AGC-like effect was noticed with the LCTV. When a small portion of the screen was set at level zero (black), its transmission came out lower than when the whole screen was set at value zero. We used a small-enough portion (25% area) of the screen, leaving the rest off (white), and masked off the white boundary with black card.

Several interference filters, 40 and 80 nm wide at measured center wavelengths, as well as glass filters were tested in an attempt to improve the contrast ratio of the LCTV. We found that contrast was improved insignificantly while light transmission became very low.

CRTs have a high contrast ratio and are conveniently self-emissive, but are subject to both brightness and geometrical uniformity problems. We experimented with several different 5 inch diagonal gray screen monochrome CRTs [26]. It was found that 'balancing' the sum of gray levels on a scan line substantially increased the isolation of sub-portions of the screen, to acceptable levels. This was accomplished by drawing the complement image of each value patch in an area on the side that was masked off by black card. Using an optical meter, near the surface of the screen, contrast ratios better than 1000 to 1 were measured.

The LCLV was the only component that was designated as "optical processing" quality [27]. Problems of consumer-type video did not come up with this device. An optimal bias voltage setting is found by seeking maximum contrast ratio (this voltage varies from device to device).

The CCD camera [28] has an AGC-disabling switch which we used. We used a video card lookup table to threshold above the noise and dark current level and improve performance. Also since our system averages the values of several (10 or more) pixels to gather one signal point, noise is decreased to approximately one or two quantization levels (out of 256). The camera was tested using a light box and calibrated step tablets. The residual influence of the background was found to be less than 3 (out of 256) quantization levels.

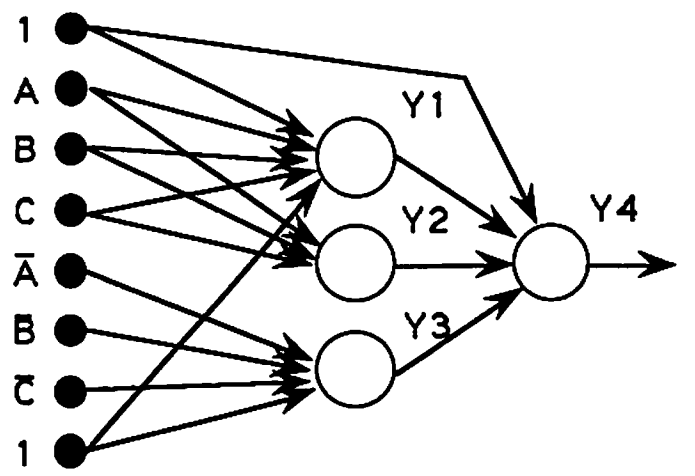
Summation of the fan-in to an output element was performed by summing a region of pixels on the video frame grab card that receives the camera signal. As noted before, two frame grabber cards were used. [29].

3.2.3 Overall Compensation

An electronic equivalent of a passive optical mask (to improve uniformity) was used. To calibrate this mask, we used an automated iterative computer procedure that measures the effect of each individual weight. The program tested for compensation factors for each of the N^4 weights. All weights start out as the maximum value, 255. The stronger connections are then weaked progressively until higher overall uniformity is obtained. We note that once the compensated values are found, they are kept in the computer and can be recalled instantly. To allow diagnosis and analysis, the computer and video card were used to generate test pattern sequences at a low rate of 15 or 30 second intervals.

3.3 Binary Neural and Digital Experiments

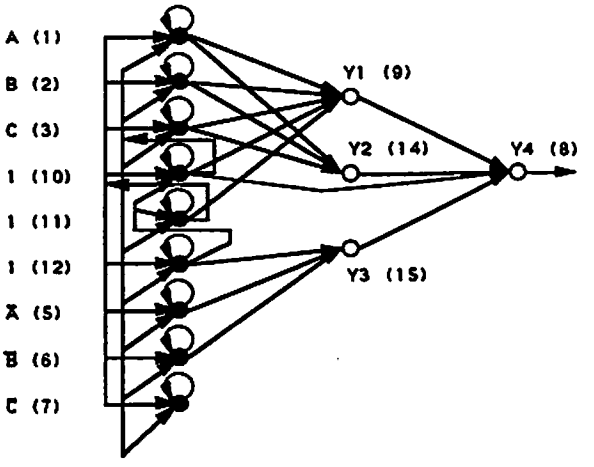
The experiments reported here used binary interconnnection weights and binary input values. The weights were fixed, not adapted during a training session as in experiments described later. Figure 3.5 shows a unipolar neural network that computes odd parity for 3 bit inputs (ABC). Figure 3.5(a) is a schematic of the

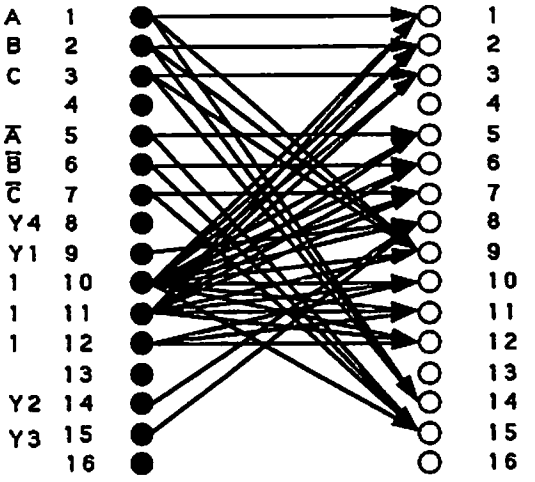


A	B	C	Y4
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

a.

b.





c.

d.

Figure 3.5: Parity checking network for 3 bits. (a) depicts the desired network. (b) is the truth table, (c) shows the network re-mapped to function with the input as the initial state, and (d) is the actual network implemented, re-mapped to one layer.

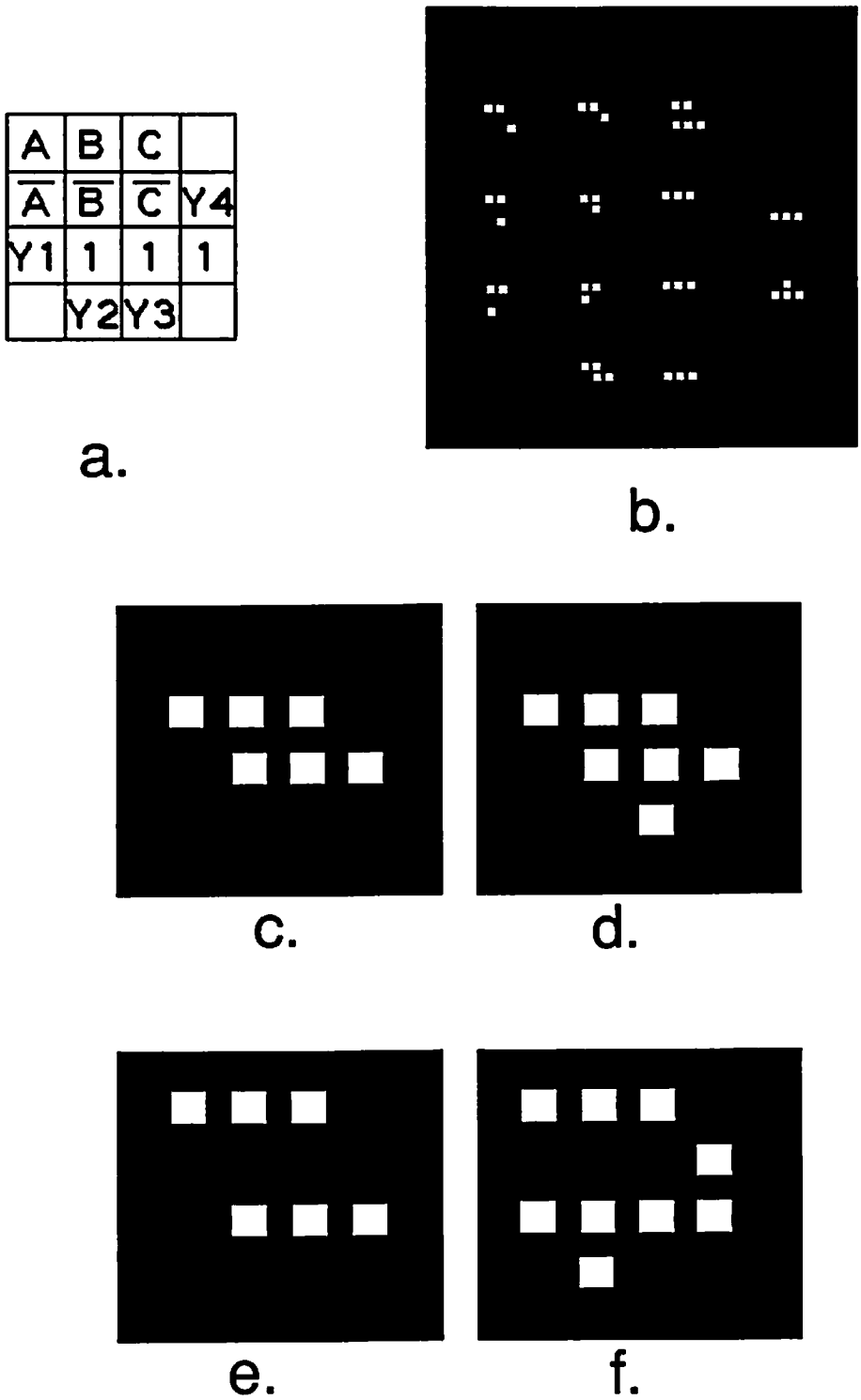
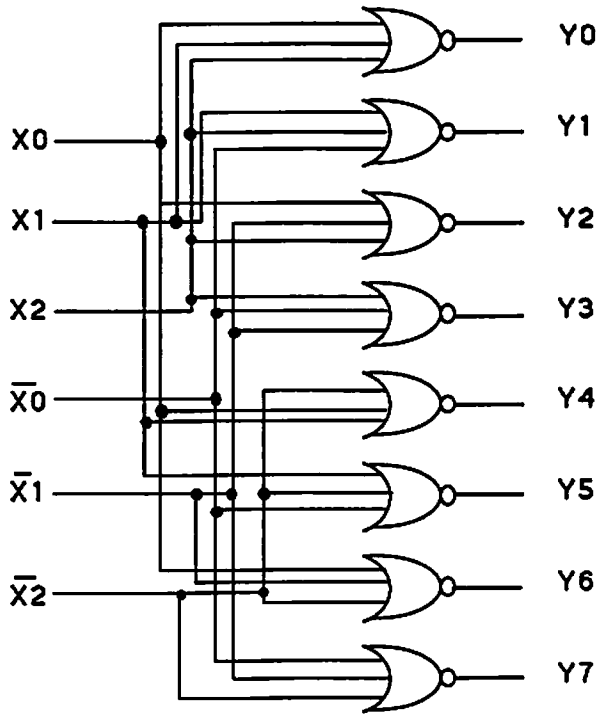


Figure 3.6: An experimental demonstration of the network of the former Fig. 3.5; (a) is the physical layout of the input plane, (b) is the interconnection mask, and (c) through (f) show examples of results from the network.

network, and Fig 3.5(b) is its truth table. A uniform hard-clip threshold of 2.5 was used for all neuron units excluding the input neuron units. All weights have value 1. The network is re-drawn twice to show the mapping to hardware. In Fig. 3.5(c), it is configured to have uniform threshold 2.5 for all neuron units, including input neuron units. Input layer lateral and self-feedback connections have been added to hold the input values constant from one cycle to the next. Thus, each input neuron unit that corresponds to a variable receives two constant 1 inputs. Each constant 1 neuron unit must also have three constant one inputs. A threshold of 2.5 then leaves all input neuron units unchanged through multiple cycles. In Fig. 3.5(d), the uniform threshold network has been re-drawn as a single layer network, as implemented in hardware. Not shown are feedback connections from each output neuron unit to the corresponding input neuron unit. The numbers in Fig. 3.5(d) indicate assignment of neuron units to hardware neuron units.

The details of the optical implementation of the network are depicted in Fig. 3.6. Figure 3.6(a) show the layout of neuron units on the 2-D input array. All neuron units, including input neuron units, are thresholded in the experimental network (Fig. 3.5(d). Figure 3.6(c) through (f) shows experimental results of the neural network implemented on the direct LAP. Two sample inputs (as displayed on the LCTV) are shown on the left ((c): ABC=000, (e): ABC=111) and corresponding outputs (as captured on the camera) are shown on the right ((d): Y4=0, (f): Y4=1). Fig. 3.6(b) shows the reflected and inverted interconnection weight array pattern (as displayed on the CRT), in which a bright pixel indicates an interconnection weight of one. Each individual image is reflected about the diagonal (from upper left to lower right) and the input image is reflected about the anti-diagonal within each image (from upper right to lower left).

In a second example (briefly described previously [16]) a digital logic circuit consisting of a 3-to-8 decoder was implemented (Figs. 3.7 and 3.8). The truth table for the decoder appears as Fig. 3.7(b) The circuit is composed entirely of 3-input NOR gates. The circuit has a single interconnection layer. Figure 3.8(a) shows the layout of gates; Fig. 3.8(b) depicts the (reflected and inverted) interconnection weight array, while Fig. 3.8(c) through (f) show two input/output pairs. Sample inputs are shown on the right, ((c): X2,X1,X0= 001, (e): X2,X1,X0= 000) and



a.

X2	X1	X0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

b.

Figure 3.7: 3-to-8 decoder circuit. (a) depicts the digital logic diagram. (b) is the truth table.

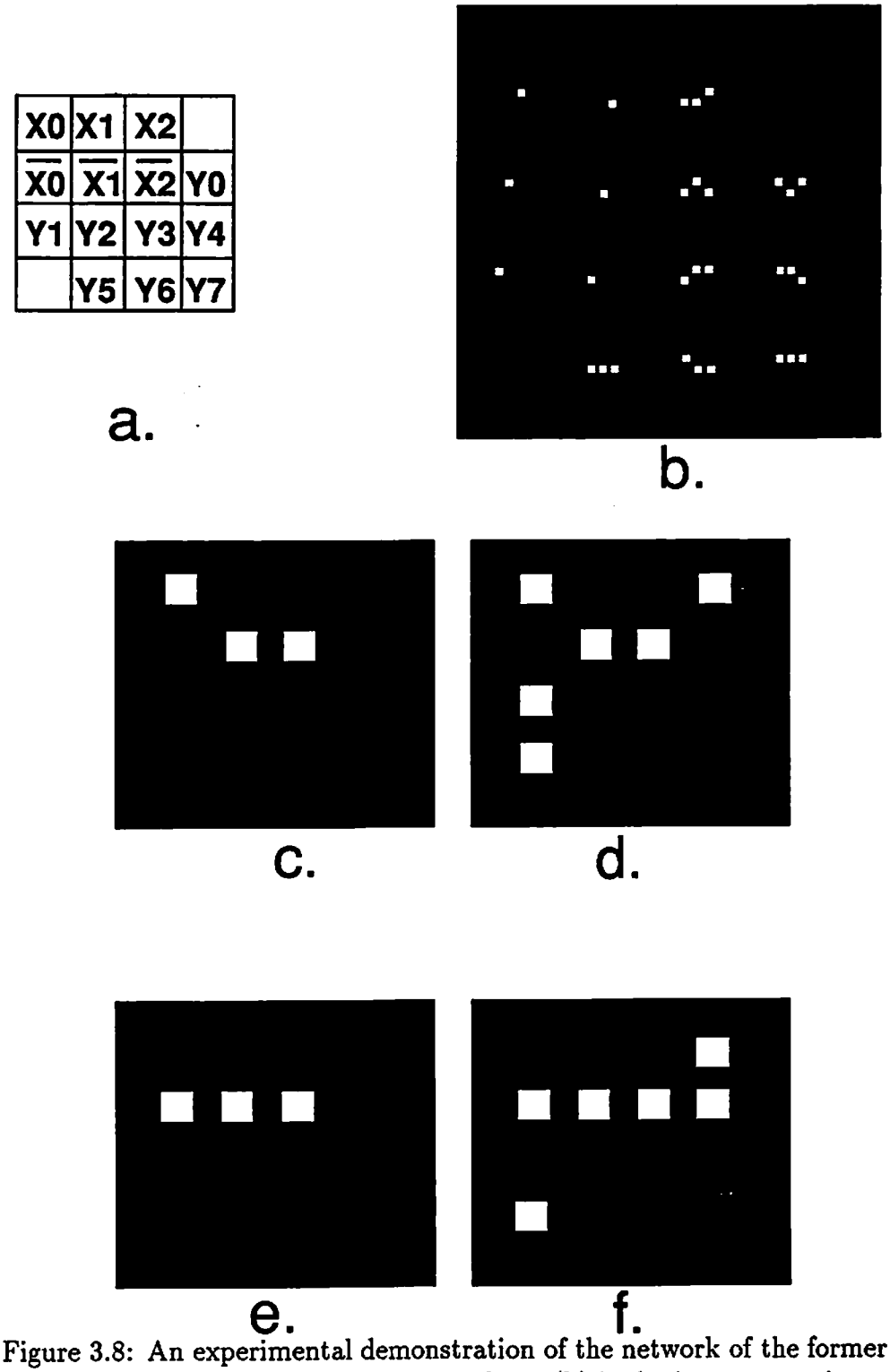


Figure 3.8: An experimental demonstration of the network of the former Fig. 3.8. (a) is the physical layout of the input plane, (b) is the interconnection mask, and (c) through (f) show examples of results from the network.

corresponding outputs are shown on the left ((d): Y1 is the result, (f): Y0 is the result). Only two cases are shown, but all possible cases were tested and worked correctly.

Chapter 4

Adaptive Neural Networks and Analog-Signal Experiments

The capabilites of the experimental system were explored in experiments of successively increasing difficulty. Initial experiments (already described) involved fixed unipolar analog weights and binary inputs. Later experiments employed bipolar arithmetic, adaptive weights and analog inputs. The two most challenging network types implemented were the multiclass perceptron [7] and competitive learning [3]. The perceptron is an archetypal supervised classifier, while competitive learning is unsupervised. Two maximum-finding networks were also implemented, one employing an alternative neuron unit function, the 'leaky integrator'.

4.1 Perceptron

The perceptron learning algorithm is one of the best-studied supervised classifiers. A convergence proof exists (for an ideal 2-class perceptron) guaranteeing successful classification of any two classes with the property of linear separability. Two linearly separable classes can be geometrically separated by a hyperplane in a space of the same dimensionality as the inputs (known an the 'feature space' of the classifier). Generalizations of the basic two-class case exist known as multiclass perceptrons. The basic operating mechanism of the perceptron is to take the inner product of the normal vector to the separating hyperplane and an input. The sign

of the inner product is the resulting class. The multiclass perceptron involves one 'linear discriminant function' per class, computed by an inner product of a separation hyperplane normal vector with the input. The maximum inner product result indicates the classification result [7].

The standard multiclass perceptron learning algorithm proceeds as follows. The input patterns are presented along with desired output. If the desired output does not match the actual, the input pattern is added to the input weights of the desired maximum neuron unit (Eq. 4.1), and the input pattern is also subtracted from the actual maximal neuron unit (Eq. 4.2). Let y_i be the inner product potential (matrix-vector multiplication result element) at step k with y_m the maximally-valued potential, let the kth input pattern $x^{(k)}$ be in class p. Let learning rate α be between zero and unity. Weight update is nonzero at step k only if $m \neq p$, where two sets of fan-in weights are altered for the mth and pth neuron units;

$$w_{pi}^{(k+1)} = w_{pi}^{(k)} + \alpha x_i^{(k)} \tag{4.1}$$

$$w_{mi}^{(k+1)} = w_{mi}^{(k)} - \alpha x_i^{(k)}. \tag{4.2}$$

Initial weights are theoretically unimportant, but in practice influenced the success or failure of classification of the fixed-dynamic range optical system.

Experiments were also conducted employing a modified multiclass perceptron algorithm. It uses a margin requirement on the maximality of the maximum output neuron unit. This increases the robustness of the classification, making misclassification less likely under perturbations of the system's arithmetic accuracy. The margin can be chosen to exceed the average or maximum time-variation observed in an experimental system's output.

The modified multiclass algorithm is a generalization of the two-class margin perceptron [7], which introduces a third possible case of training patterns presentation result. The correct output neuron unit may be maximal, but by an amount less than the margin. The previously stated algorithm does exactly nothing in this case, since the actual and desired maximum neuron unit are the *same* neuron unit. Two variations were explored to introduce a weight correcting step in this third case. In the first variation, the input pattern is multiplied by the learning rate

and added to the maximal neuron unit's fan-in weights only (Eq. 4.4 alone). The second variation subtracted the scaled input pattern from the secondary maximum as well (Eq. 4.4,4.5),

$$y_m = \max_i y_i, \qquad y_s = \max_{i,i\neq m} y_i. \tag{4.3}$$

If $m = p$ and $y_m - y_s < b$ (case 3),

$$w_{pi}^{(k+1)} = w_{pi}^{(k)} + \alpha x_i^{(k)} \tag{4.4}$$

$$w_{mi}^{(k+1)} = w_{mi}^{(k)} - \alpha x_i^{(k)} \tag{4.5}$$

where the previously stated steps are taken in the first two cases; correct classification $m = p$ and $y_m - y_s \geq b$ (no action), and in the case of incorrect classification $m \neq p$, (Eqs. 4.1,4.2). Small margins were used, 5 or 10 out of 255, so that the requirements on dynamic range and uniformity did not become much more severe. All optical experiments reported here used the first variation, but unsuccessful experiments on larger data sets were tried using the second variation. No significant improvement in classification capability was noticed with the second variation.

A scaling factor was applied to each weight to improve performance, and maximize use of the limited dynamic range of our experimental system. The scaling factor was calculated to set the maximum weight value of all interconnections to the (analog) signal maximum of the experimental system (which was gray level 255), and all other weights were scaled also by the same multiplicative factor. This resulted in loss of small signals below the system noise level, but freed the experimentor from the need to manually scale the inputs so that a weight value was never 'clipped', or lost in saturation. This scaling held constant a maximum measure on the set of all weights. The training inputs were scaled identically with the weights, preserving the mathematical form of the multiclass perceptron.

Simulation in a conventional computer program used floating-point arithmetic, without any global scaling of weights. The program initially established linear separability of the training set by attempting the supervised learning procedure. High dimensionality (16-D) of the input space made the task of finding linear

separable sets easy; only one set out of dozens used for simulation was not linearly separable as judged by the floating point simulation.

While the order of presentation is theoretically unimportant, different orders of input can yield different maximum dynamic range of the weights. An experiment on a system with a limited dynamic range may thus converge for one order of input presentation, but not another. Crosstalk between input elements also limited the classification capabilities of the experimental system.

The perceptron algorithm proved to be quite robust in application. Performance of the original and margin variation algorithms was very similar for small margins, where the third case of result appears relatively infrequently. Most of the experiments used a margin of 1, or 5 gray levels. A margin of 1 yielded virtually identical performance to margin 0 (original algorithm), while a margin of 5 resulted in convergence with some data sets that did not converge with margin 0. A margin of 10 or more out of 255 proved to easily prevent convergence in many cases. Computer simulation with accurate digital arithmetic yields correct classification varying only in the number of steps required, for margins that can be accurately represented by digital fixed or floating point variables.

Time multiplexed bipolar binary inputs and bipolar analog interconnection weights were used for the majority of perceptron experiments.

Figure 4.1 shows the input to an experiment where weights successfully classifiying the input were 'learned' starting from zero after 5 complete presentations of the input, with a margin of 5. Four classes with two patterns in each class formed the input data.

Time-multiplexed bipolar weights were used as well in another perceptron experiment, where the initial weights were pre-calculated as the sum of the members of each class. Four translations each of binary-valued letters U and S (Fig. 4.2) were correctly classified into two classes (all Us and all Ss), with a learning rate of 1 and a margin of 1, after 10 complete presentations of the input patterns.

Another experiment with five classes of two patterns each classified all patterns correctly on the first try. The sum of the two patterns in each class again formed the initial weights, with learning rate 1 and margin 1. Figure 4.3 depicts the data set and the five classes.

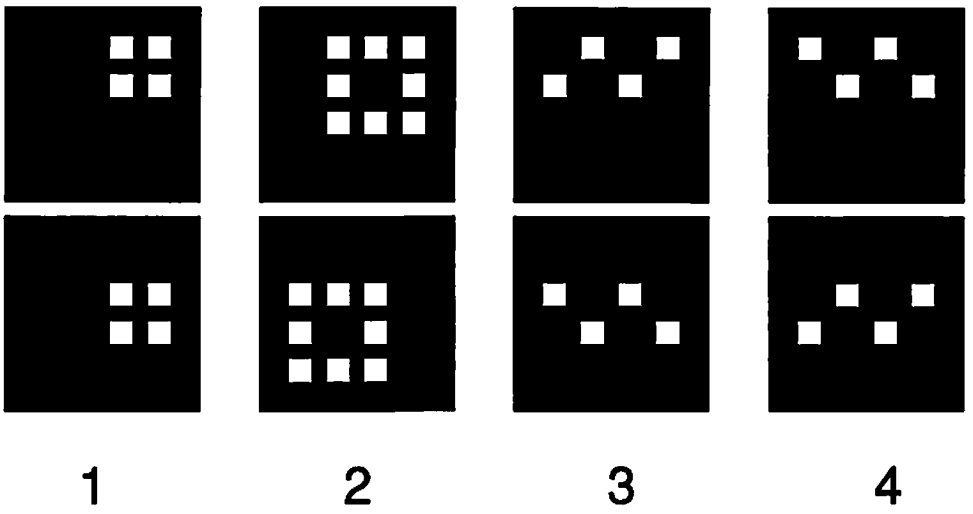


Figure 4.1: Perceptron with 4 classes was successfully implemented on the optical system, two input patterns per class shown, initial weights zero, 7 complete sets of patterns were presented before convergence (successful classification of all patterns).

The margin variation was successful in counteracting the time-variation of output analyzed later in detail in this paper. The training set depicted in Fig. 4.1 was presented in full nine more times after convergence. Although the average range of output was 6 gray levels, all patterns were correctly classified each time, as a result of the training with margin 5.

Convergence (success) of an optically-interconnected network using the perceptron learning rule was thus demonstrated with incoherent light, analog bipolar weights, binary bipolar inputs, and refractive fan-out.

4.2 Competitive Learning

Unsupervised competitive learning mechanisms were first investigated by Rosenblatt [3]. Von der Malsburg and others developed models based on competitive learning. A common yet simple model was chosen for implementation [3]. It features competition between units to 'learn' (adapt input weights), 'learning' by increasing response to an input pattern, and conservation of total input weight of each neuron unit. The input patterns are assumed binary. This model has also

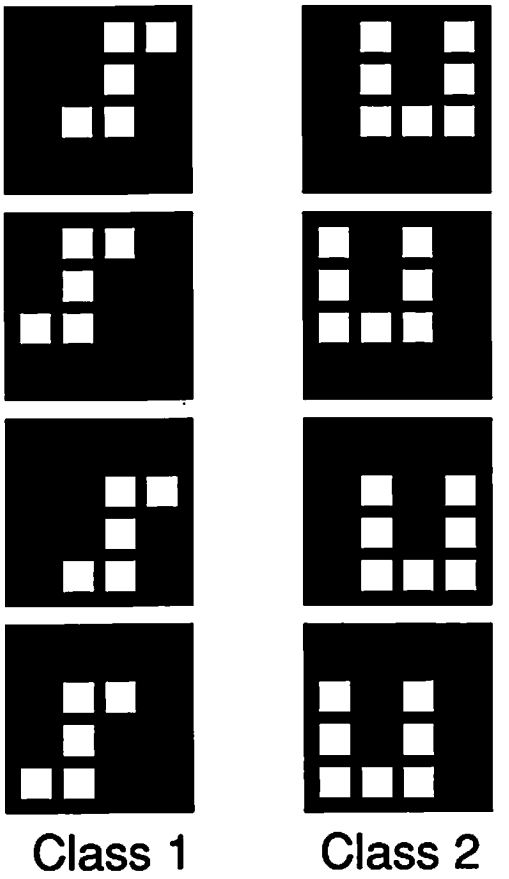


Figure 4.2: Training set for a two-class perceptron experiment involving four translations each of U and S characters. Initial weights used were the sum of the classes, and all patterns were classified correctly after 10 complete presentations of all inputs.

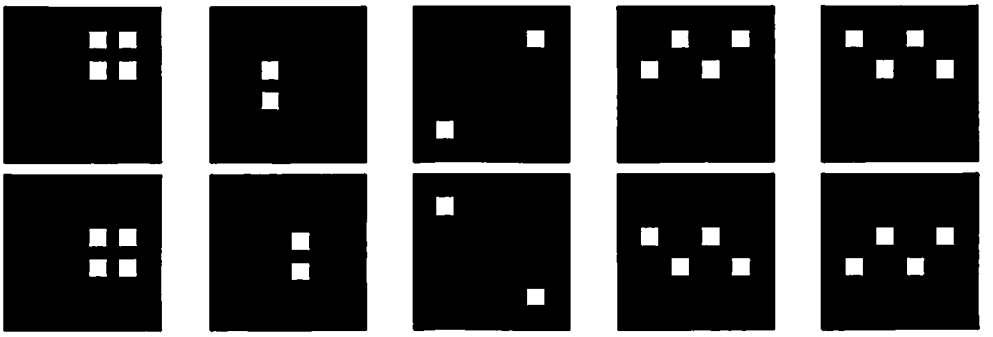


Figure 4.3: Training set for a 5-class perceptron experiment that used zero initial weights, and correctly classified all patterns upon one complete presentation of the training set.

been generalized to allow a neuron unit to have input weights that do not sum to one, and to generalize the input patterns to have analog values. Figure 4.4 depicts a multiple layer competitive learning network. Competition to update weights occurs between units within a dashed box, called a *competitive cluster*. Only single-interconnection layer networks were implemented experimentally, with the entire output layer serving as one competitive cluster. In practice, the maximum valued 'winner' was selected by digital comparison and a serial program loop, avoiding an iterative competition process.

Let n_k be the sum of the elements x_j of the kth input pattern. Let y_m be the maximum output term y_j. Let s_j equal the sum of the input weights to the jth neuron unit, and let g be the learning parameter between zero and unity

$$n_k = \sum_{j=1}^{n} x_j^{(k)}, \quad y_i^{(k)} = \sum_{j=1}^{n} w_{ij}^{(k)} x_j^{(k)}, \quad s_i^{(k)} = \sum_{j=1}^{n} w_{ij}^{(k)} \tag{4.6}$$

$$y_m = \max_i y_i, \quad w_{ij}^{(k+1)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)} \tag{4.7}$$

Input weights of the maximum output neuron unit are then updated as follows

$$\Delta w_{ij}^{(k)} = \begin{cases} 0 & \text{if } i \neq m \\ g s_i^{(k)} \frac{x_j^{(k)}}{n_k} - g w_{ij}^{(k)} & \text{if unit } i = m. \end{cases} \tag{4.8}$$

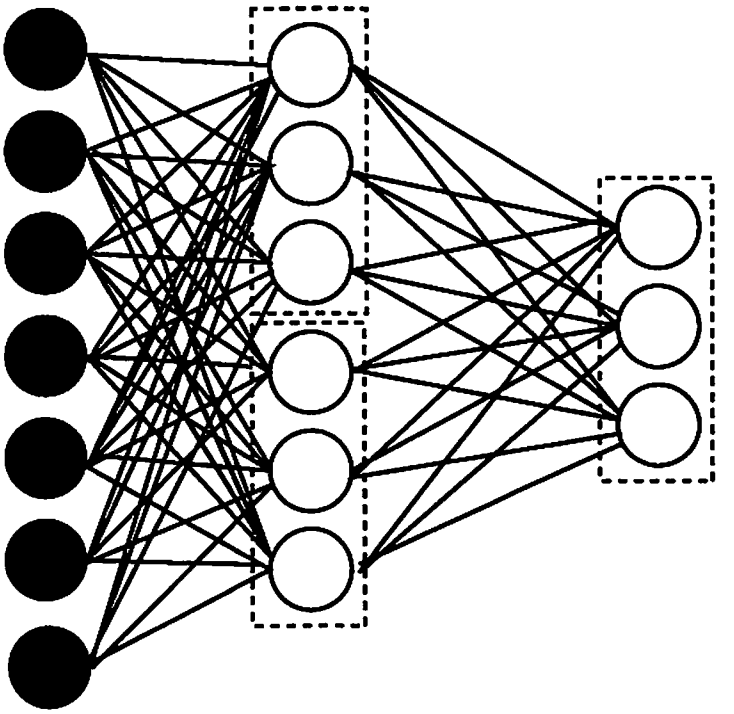


Figure 4.4: General multilayer competitive learning network. Each layer may contain multiple competitive clusters in which only the maximally-valued neuron unit adapts its input weights upon presentation of an input pattern.

A particular feature of the PDP [3] model is to conserve the sum of fan-in weights to an output neuron unit, a *local* normalization. Other work describes a competitive learning model without this normalization. This more clearly emphasizes the fundamental mechanism of calculating a weight update that is proportional to the difference between input and stored weight [31]. An alternative global scaling of maximum weight (as in our perceptron experiments) could then ease requirements on implementation hardware.

Convergence of unsupervised classifiers is not as clear-cut as that of supervised ones. Each training pattern always changes the interconnection weights in unsupervised classifiers, as no criteria of correct classification exists as a motivation to leave weights unchanged. A sequence of decreasing learning rate parameters can be employed, but was not used for our experiments. Instead, the criterion of unchanging classification over a period of several complete presentations of the training set was used. This criteria also guided choices of learning rate α. Large enough α prevented convergence in the competitive learning experiments.

Figure 4.5(a) depicts a classification of an analog-valued input data set resulting from operation of the experimental system. Unipolar analog weights and inputs were used.

A computer simulation was performed that used identical inputs and parameters (Fig. 4.5(b)). The simulation grouped patterns differently, however. Inaccuracy in representing initial weights may account for slightly different classification results than in the computer simulation. Different 'tunings' of the system, with different non-uniformity patterns, also resulted in differing classifications.

Several binary experiments were also performed investigating the effect of different learning rates. A learning rate of 0.15 resulted in identical results for four repeated runs, while a learning rate of 0.3 resulted in three different classifications over five repetitions.

Figure 4.6(a) shows the classification resulting from three of five runs with learning rate 0.3, and from (all) four runs of an otherwise identical experiment with learning rate 0.15. Figure 4.6(b), (c) depict the two other classifications resulting from learning rate 0.3. The parentheses indicate an oscillatory condition in effect at

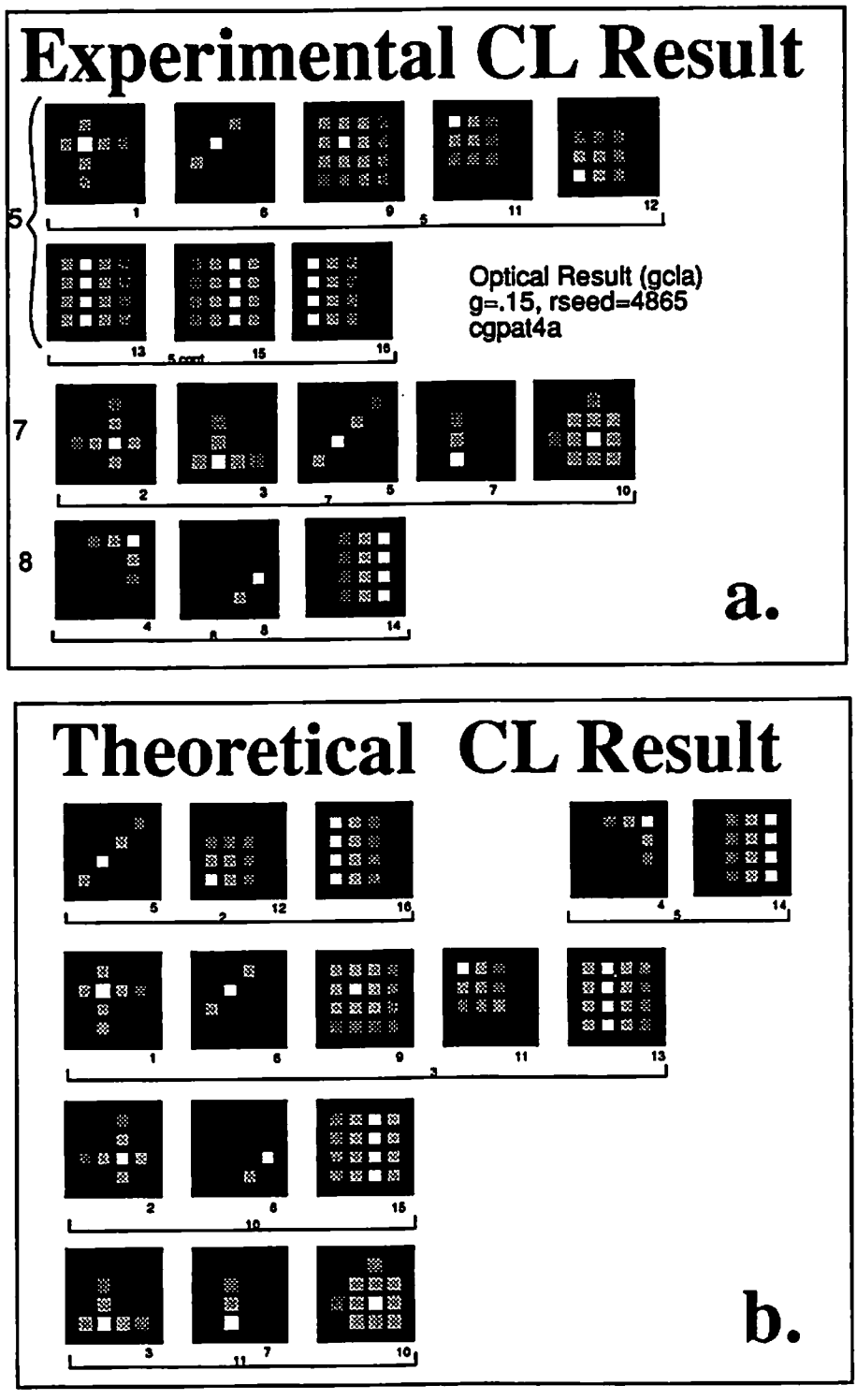


Figure 4.5: Competitive learning experiment with analog input patterns; (a) experimental classification result; a class is shown as the individually-numbered patterns above brackets with the same bracket number; (b) simulation results using identical inputs and parameters as the optical experiment. Inaccuracy of simulation initial weight representation may contribute to differing result.

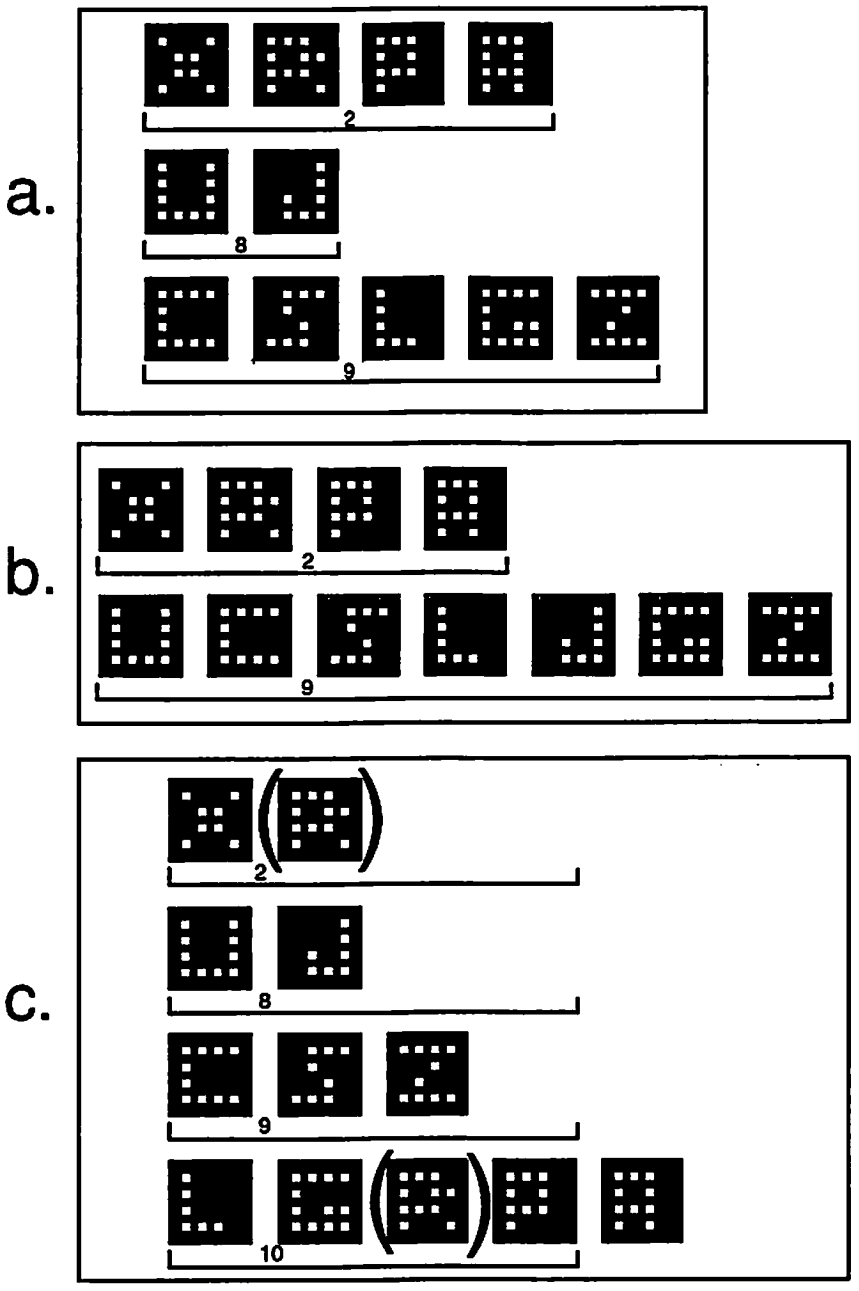


Figure 4.6: Binary optical experimental results showing sensitivity to learning rate with $\alpha = 0.3$, three classifications occurred in five trials; (a) one classification occurred three times, the other two only once ((b), (c)); another experiment resulted in the same classification as in a. over four trials with $\alpha = 0.15$.

termination of iteration. The parenthesized input pattern was alternately classified in two different classes.

Different inital weights changed the resulting classes completely (four sets were used). Different input pattern sequences changed the results of both simulation and optical experiments. Learning rate of 0.3 was used, and three different random seeds were used with a conventional pseudorandom function call to generate the weights. These same weights were then used for both optical and simulation experiments. Figure 4.7(a) shows the three different classifications resulting from LAP optical experiment, while Fig. 4.7(b) depicts the classification of simulation. The learning rate affected repeatability, but also affected sensitivity to presentation order. The analog valued data set was 'shuffled' to create a differently ordered set, where every previously fourth pattern became adjacent. Two other sets were additionally created with still different presentation orders of the same data. Two different learning rates were then tried with this data. All four results were different with learning rate 0.3, but the original and fourth–interleaved sets yielded classifications identical except for the class of one input pattern (the other two sets were not used).

A variation of the competitive learning algorithm was also explored. This variation attempted to reduce the number of output units that never adapted their fan-in weights at all. A variable 'threshold of learning' influenced the choice of 'winner', (the only output neuron unit to adapt its weights) as well as the maximum output neuron unit value. Unfortunately, sensitivity to input pattern order was more pronounced as a result. 'Winning' causes an output neuron unit to have a response that is scaled down by a multiplicative factor $1 - l_i$. This scale-down effect would decrease linearly over a time period corresponding to that required for a complete presentation of the training set (with p patterns)

$$y_m = \max_i (1 - l_i)\, y_i \tag{4.9}$$

$$l_i^{(k+1)} = \begin{cases} d & \text{upon neuron unit } i \text{ 'winning'} \\ l_i^{(k)} - \frac{d}{p-1} & \text{upon neuron unit } i \text{ 'losing'}. \end{cases} \tag{4.10}$$

after first being sharply set to its maximum, $1 - d$. Successive patterns that were previously classified together were not always classified similarly when using this

Simulation Results

Set1

Class	Char
2	S
5	UL
7	XZ
8	RPA
9	J
10	CG

Set2

Class	Char
1	UCLG
2	RPA
4	x
5	J
9	SZ

Set3

Class	Char
1	S
7	UCLG
8	RPA
11	XJZ

Optical Results

Set1

Class	Char
2	SX
10	UCLJ
10c.	GZRP
10c.	A

Set2

Class	Char
2	XRPA
4	UJ
10	CSLG
10c.	Z

Set3

Class	Char
4	S
8	J
9	UCLG
9c.	Z
11	XRPA

Figure 4.7: Binary experimental results showing initial weight dependence. Three different sets of random initial weights yielded three different classifications for (a) simulation; (b) optical experiment.

modification (with our fixed order training set presentation). Figure 4.8 shows the classification results of the modified algorithm with the three initial weight sets, and the same unipolar binary character input data. The very similar patterns *R,P,A* were classified together in *all* unmodified algorithm optical experiments and simulations, using three different sets of initial weights. The modified, enforced

Enforced Learning Results

Set1		Set2		Set3	
Class	Char	Class	Char	Class	Char
2	ULZ	1	RPA	1	SX
4	A	2	S	2	UJ
5	P	3	UJ	5	C
6	CJ	4	X	6	R
8	S	6	L	7	L
9	X	9	G	8	Z
10	G	10	Z	9	A
11	R	11	C	10	G
				11	P

Figure 4.8: Binary optical experimental results using 'enforced learning' variation shows initial weight dependence and increase in number of classes formed (output neuron units that adapted their weights). Three different sets of random initial weights yielded three different classifications as in the unmodified algorithm.

learning algorithm classified all three into different classes in two cases, and put all three in one class in just one case. All three of these input patterns were adjacent in the sequential presentation order. The unmodified algorithm, by contrast, grouped the patterns together in spite of the adjacent ordering. (Random presentation order may reduce this effect.)

The modified algorithm *was* successful in increasing the number of output neuron units that 'learned' (*i.e.* number of classes that formed), however. The average number of classes formed in unmodified algorithm optical experiment was 3 (with the same input data and three sets of initial weights), in unmodified simulation was 5, but was 8.3 for enforced learning optical experiments.

The competitive learning experiments explored sensitivity to learning rate, pattern order and initial weights for this unsupervised network. The paradigm of 'feature discovery' describes the application of optical accuracy modifying the mathematical initial conditions and subsequent learning evolution.

4.3 Maximum Networks

Two maximum-finding neural networks were implemented experimentally. These networks use fixed interconnections. The PDP [3] competitive learning model describes use of such a maximum network for each competitive cluster, but the slow clock speed of the experimental system essentially prohibited an experimental combination of the two networks; the maximum networks were thus implemented in isolation.

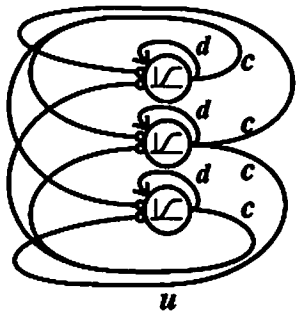


Figure 4.9: Basic lateral inhibition network for simple competitive cluster. A feedback excitation is combined an inhibitory signal received from all other neuron units.

The first maximum-finding network directly simulates 'lateral inhibition', where each neuron unit 'inhibits' every other neuron unit (sends a signal that reduces the other unit's input activation) [4]. Figure 4.9 depicts the network with three neuron units, while experiments used eight and sixteen neuron units. Unipolar inhibitory signals were communicated optically, and weighted by a uniform fixed value (subject to the 'equalization' procedures [23]). This inhibitory input from all other neuron units was differenced with a feedback excitatory input equal to a neuron units' output at a previous time step, where neuron unit output is a

the slope of the function f,

$$f(x) = \frac{1}{1 + e^{-\kappa(2x-1)}} \tag{4.11}$$

where

$$u_i^{(new)} = d \cdot f(u_i) - c \sum_{k \neq i} f(u_k). \tag{4.12}$$

Figure 4.10 depicts f with $\kappa = 7.0$. While being simple, this algorithm suffers from strong sensitivity to input strength. Selection of inhibitory weight c and

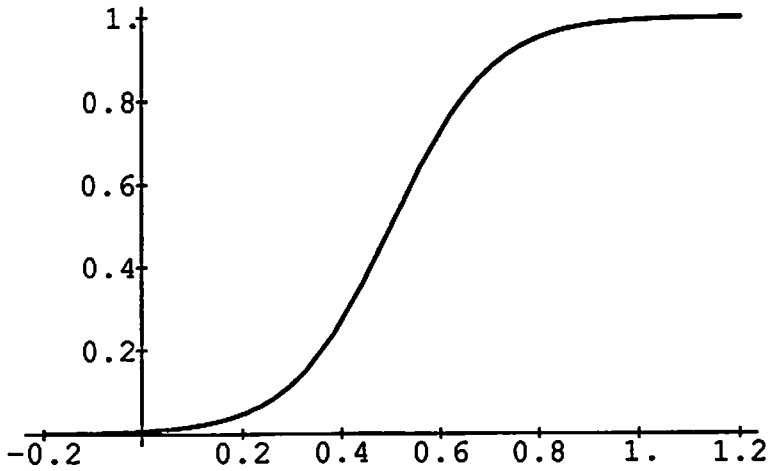


Figure 4.10: Sigmoid nonlinearity function graph, shown normalized to 1. Unipolar unit activation maps to unipolar unit response.

excitatory weight d must be chosen as a function of input data maximum and average value.

One computer simulation experiment using a network of 8 neuron units, and sigmoid constant $\kappa = 7.0$ converged in eight iterations.

The optical implementation of the same network showed similar sensitivity to input strength. A different $c(1)$ and $d(1.5)$ were used, due to a mismatch of overall gain between the optical interconnection and electronic self-feedback term. A successful maximum-finding result was obtained with a 16-element input to a

16-neuron unit network, with the inputs at values 255, 130, 100 respectively, and the rest of the inputs at 15, 20, and 25. Another trial successfully discriminated 195 from 255, a maximum that was maximal by 15%. The sum of the inputs was in this case more significant than the increase in difficulty of the maximum-finding problem. Convergence to the solution was rapid in all cases, if convergence ocurred at all, within ten iterations in all cases.

The second maximum network uses an alternative neuron unit function model, called the *leaky integrator*. Time-dependence is exhibited by this model. Activation of the neuron unit is a function of not only present input potential, but by a history of past input potentials as well (the previous neuron unit models described have activation that is a function only of present input potential). Consider a single leaky integrator neuron unit. The time evolution of the neuron unit output y(t) is given by

$$\tau \frac{d}{dt} y(t) = -y(t) + h + I, \tag{4.13}$$

where τ is a time constant, I is the input potential to the neuron unit, and h is the steady-state rest level (in steady state $y(t) = h + I$).

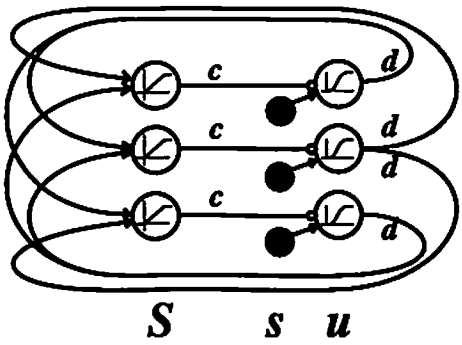


Figure 4.11: Amari-Arbib network implementing the Didday prey selection model, a maximum-finding network. Inhibitor (S) and indicator (u) neuron unit pairs compete; inputs s are continually input as the system equilibrates.

Leaky integrator neuron units are interconnected to implement the Didday prey selection network as proposed by Amari and Arbib [5]. One layer of neuron units ('relative foodness') with values u_i, indicate the result of finding the maximum. Each neuron unit in this layer is paired with an inhibitor neuron unit ('sameness layer') with value S_i. Each indicator neuron unit inhibits all of the inhibitor neuron

units except the one inhibiting itself. Inputs s_i are fed into the indicator neuron units continually as the system stabilizes to a steady-state solution. Figure 4.11 depicts the network with three inputs. Constant c scales the self inhibition term $-cg(S_i)$ where g is a linear limiter function given in Eq. 4.14, while d scales the mutual excitation term $d\sum_{k\neq i} f(u_k)$, where g is a linear limiting function

$$g(x) = \begin{cases} 0 \text{ if } x < 0 \\ x \text{ if } 0 \leq x \leq 1 \\ 1 \text{ if } x > 1. \end{cases} \tag{4.14}$$

The maximum network is thus governed by

$$\tau_u \frac{d}{dt} u_i = -u_i + s_i - cg(S_i) + h_u \tag{4.15}$$

$$\tau_S \frac{d}{dt} S_i = -S_i + d\sum_{k\neq i} f(u_k) + h_S. \tag{4.16}$$

A discrete-time approximation using Euler's method was used for computer simulation and LAP system experimental implementation,

$$u_i^{(new)} = u_i + \Delta u_i = u_i + \frac{\Delta t_u}{\tau_u}(-u_i + s_i - cg(S_i) + h_u) \tag{4.17}$$

$$S_i^{(new)} = S_i + \Delta S_i = S_i + \frac{\Delta t_S}{\tau_S}\left(-S_i + d\sum_{k\neq i} f(u_k) + h_S)\right). \tag{4.18}$$

A drawback of this model is the large number of parameters to be chosen/optimized. The model also exhibits sensitivity to the strength of the input patterns as well.

Simulation revealed that floating-point accuracy permitted tailoring the parameters so that all units but one went to zero. Many parameter settings instead yield a single maximally-valued output unit with the other units' values *not* going to zero. Other settings resulted in two units approaching maximal values as the others reached low, but nonzero steady states. Both simulaltion and optical implementation showed that successive iterations successively increased the ratio of maximum to other outputs. Unfortunately, sensitivity to overall strength of inputs was increased with adjustment of the parameters for increased ability to

discriminate between two closely-valued inputs. Successful results were often obtained after significantly more iterations than in the simpler maximum network, on the order of 100 iterations.

An eight-element input vector was used for a simulation, with input values 0.4, 0.2, and 0.15, and all other inputs 0.1. Time constants h_u and h_S were 0.1 and 0.2 respectively, $\Delta t_u = \Delta t_S = .08$, and interconnection weights $c = 0.8, d = 0.15$. The ratio of maximal neuron unit value to the second-most maximum increased steadily with the number of iterations. A maximum value of 0.33, and second-most maximum of 0.077 was obtained after 115 iterations. The same parameters worked much better with a data set with 0.7, 0.5 maximum and second-most maximum inputs (other inputs as before). A value of 0.58 versus 0.17 was reached after 25 iterations, and 0.63 maximum with otherwise zero floating point values was achieved after 50 total iterations.

Neuron potential depended on the difference between optical input and electronic self-feedback. The optical system thus introduces an additional relative scale factor to be matched by simulation. Correlation of parameters of optical experiment and simulation was in general difficult.

The Didday maximum network optical implementation actually converged much faster than simulation predicted. Rest levels of $h_u = 40, h_S = 25$ were used, weights $c = 0.5, d = 10, \Delta t = 0.1$ and $\kappa = 0.7$ were employed. An input vector with values 255, 250, 100, with other inputs at 25, 21, 10, 15 and 10 reached a distinct maximum of 254, with second-most max 23 (and all other neuron unit values at 2 or 3) in just 7 steps. The maximum networks used fixed analog weights and unipolar analog inputs. The iterative processing nature of theses networks renders them sensitive to overall input pattern strength, as well as other parameters.

Chapter 5

LAP Performance Characterization

5.1 Arithmetic Operations

Performance of the LAP is most naturally expressed in terms of finite-range analog unipolar operations. As described earlier, unipolar inner product operations accomplish $N^2(2N^2-1)$ operations in one optical cycle, while unipolar outer product operations perform N^4 operations (multiplications) per optical cycle.

The total time τ for an optical cycle is the sum of the times required for each component to respond and stabilize. Denote this time in seconds for the following components as: S_i for the input device, S_w for the interconnection weight array SLM, D for the detector array, T for the activation/logic function, W for weight update calculation. At the present state-of-the-art, we may consider the optical propagation time negligible. The sum of these terms adds up to τ:

$$S_i + S_w + D + T + W = \tau \tag{5.1}$$

(if the input and weight SLMs are updated in parallel, $\max(S_i, S_w)$ should replace $S_i + S_w$). The basic performance figure is calculated as operations per optical cycle divided by time for optical cycle,

$$P_{ui} = \frac{N^2(2N^2 - 1)}{\tau} \approx 2\frac{N^4}{\tau} \tag{5.2}$$

operations per unit time for the unipolar inner product, and $P_{uo} = N^4/\tau$ for the unipolar outer product. Unipolar vector sum performance is the same as unipolar inner product, as long as the initial data transformation is added to the cycle time replacing τ with τ_v.

Bipolar operation performance is now similarly calculated, by taking into account that N^2 non-negative terms yield $N^2/2$ bipolar ones. Add the additional operations of bipolar arithmetic into the τ term, calling it τ_b. Then

$$P_{bi} = \frac{N^2(N^2 - 1)}{2\tau_b} \approx \frac{N^4}{2\tau_b} \tag{5.3}$$

bipolar operations per second. Bipolar outer products attain $P_{bo} = N^4/4\tau_b$ bipolar operations per second. Note that these results for space coded bipolar operation are identical to those obtained for time multiplexed bipolar by replacing τ with $4\tau_b$. There are twice the number of bipolar terms of the space coding case, N^2, and 4 time steps must be taken for the complete bipolar array operation.

The term W in Eq. (5.1) expresses the dependence of weight update on the present and past computations of the network, which is specific to the type of neural network used. Computation of weight update depends possibly on inputs, outputs, or external information such as desired outputs. A global or local neighborhood of values of any of these may be required for a single weight update computation. Specialized applications with infrequent updates to one of the SLMs can be modeled by setting S_i or S_w to zero appropriately. Similarly, applications with little or no computation required for weight update can be modeled by setting W to zero.

Earlier works [10, 11] estimated the practical limit to LAP scale-up as being $N \approx 50$ (about 6×10^6 elements in the 2-D weight array). Assuming total cycle

time τ is one microsecond, the rate of unipolar operations would be on the order of 1.25×10^{13} sec^{-1} for the unipolar case, or a Teraops-range bipolar throughput.

The energy efficiency of the LAP is dictated by two factors; physical light loss from photometric inefficiency, and loss from the broadcasting nature of the fan-out of the system. The photometric inefficiency losses are inherent to incoherent, non-diffractive optical interconnection systems. Noise immunity and conceptual simplicity are the benefits obtained from this incoherent, non-diffractive inefficiency. The loss from broadcast operation results from blocking unused light signals at the weight plane. Diffractive systems may offer a way to avoid sending light signals to undesired positions, but these systems also tend to have low efficiency in large-scale applications [32].

5.2 Parametric Performance Model

Characterization of the processor was required throughout the development, especially during modification from unipolar binary to analog bipolar arithmetic operation. Characterization of working systems and devices is a powerful tool in the optoelectronic system development process. Simple mathematical modeling forms the required structure for quantitative measures of performance. A combination of empirical observation and theoretical considerations forms the basis for the modeling. The exact mathematical assumptions used can be to some degree arbitarary; consistent application of the model is more important in comparison of two working prototype systems.

Modeling can also serve as a basis for the prediction of the outcome of repeated calculations, but the cascading-error nature of this process makes significantly more exacting demands on the accuracy of the model than those of performance metric motivation.

5.2.1 Introduction to Parametric Performance Model

The quantitative measures of performance are the parameters of the mathematical model. These parameters can be adjusted to yield ideal, error-free operation. A

different set of parameters model operation of a working optical system. Inaccuracy is categorized into four classes, with consideration to decoupling the influence of the different classes upon each other. Inaccuracy of just one category at a time can be explored using this model.

Define $z_{ij} = w_{ij}x_j$ as an unsummed product of a weight element with an input copy element for convenience. More information about system accuracy is available without the summation step, for purposes of characterization.

The parametric LAP model breaks down performance into four categories. These categories concern nonlinearity of weight-input copy multiplication, time-variation of output terms (z_{ij}), non-uniformity of output terms, and crosstalk between elements in the two-dimensional input, weight and output planesmodifying the expected output terms results. Local crosstalk refers to the degree of signal 'leaking' between adjacent inputs, weights or outputs while global crosstalk refers to this unintentional signal transfer between non-adjacent elements. The approximation of the model deliberately avoids computational-intensive manipulation of extensive data from experimental systems.

Nonlinearity in the multiplication of weight and input-copy terms is modeled as the average-case, to de-couple from space and time variation of the output. The product of this multiplication is in general a nonlinear function of weight and input value. Modeling the multiplication as a product of low-order polynomials is assumed accurate enough for reasonably linear product behavior

$$z_{ij} = \left(\alpha_0 + \alpha_1 w_{ij} + \alpha_2 w_{ij}^2\right)\left(\beta_0 + \beta_1 x_j + \beta_2 x_j^2\right). \tag{5.4}$$

This polynomial representation leads naturally to metrics comparing the magnitude of second order, first order and zero order coefficients, where we can define linearity as the ratio of first to second order coefficients $\alpha_1/\alpha_2, \beta_1/\beta_2$. Bias may be similary defined as the ratio of zero to first order coefficients $\alpha_0/\alpha_1, \beta_0/\beta_1$.

Time-variation of output occurs as the optical system gathers the result of repeated presentation of the same weights and inputs. Averaging over all output terms allows de-coupling from space variation of output. First and second order statistics on the spatial average case offer an immediate characterization of time-variation. Efforts could be made to characterize the statistical variation as

Gaussian or not for device-manufacture efforts, but our use of video equipment made the exact distribution less interesting.

Non-uniformity (or space-variation) of output occurs as a uniform weights and inputs are presented to the system. A several-output result average for each unsummed product is used for this statistical characterization, to de-couple from time-variation of output.

Crosstalk can occur between elements of the input, weight and output (unsummed product) planes. This results in an output term u_{ij} that is a function of not only its intended value z_{ij}, but of other intended products z_{pq}. The following assumptions were made to yield a computationally tractable (linear) model. Unintended signal transfer was assumed to be additive and proportional to intended products by a factor γ_{ijpq},

$$u_{ij} = \sum_{(p,q)} \gamma_{ijpq} z_{pq}. \tag{5.5}$$

Physically adjacent crosstalk is distinguished from non-physically adjacent. Horizontal and vertical physical adjacency is distinguished from diagonal adjacency. Crosstalk effects are also assumed uniform. Define $S_4(z_{ij})$ as the set of unsummed product terms z_{pq} making up the horizontal and vertical neighbors (4-neighbors) of z_{ij}. Define $S_{d8}(z_{ij})$ as the set of terms z_{pq} making up the diagonal neighbors (diagonal 8-neighbors) of z_{ij}, and $S_e(z_{ij})$ as the terms z_{pq} with $p \neq i$ or $q \neq j$ (see Fig. 5.1). The following equation thus expresses the crosstalk contributions as a linear sum. Assuming crosstalk coefficients are uniform over a lenslet allows a matrix equation to be written for each lenslet

$$u_{ij} = az_{ij} + b \sum_{(p,q)\in S_4(z_{ij})} z_{pq} + c \sum_{(p,q)\in S_{d8}(z_{ij})} z_{pq} + d \sum_{(p,q)\in S_e(z_{ij})} z_{pq} \tag{5.6}$$

Given theoretical unsummed product array z_{ij} and actual crosstalk-augmented unsummed product array u_{ij}, we can calculate the matrix $\underline{\underline{S_i}}$ from z_{ij}, and perform a least-squares minimization, and average over lenslets i to form the average

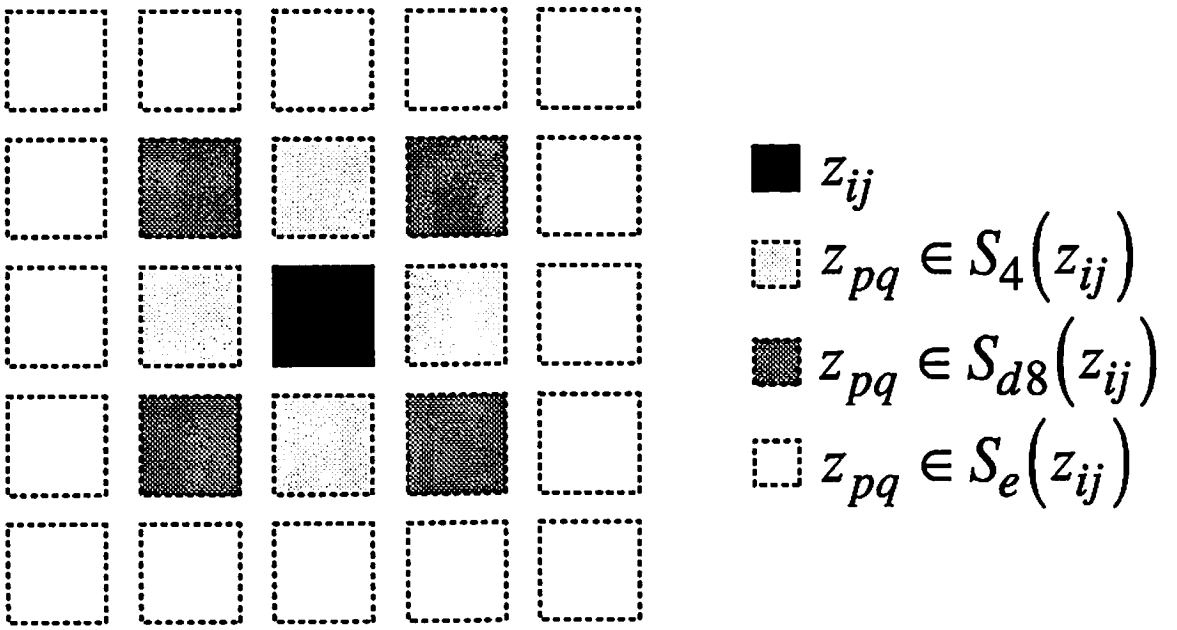


Figure 5.1: Unsummed products elements z_{pq} are either (physically adjacent) 4-neighbors, (S_4), diagonal 8-neighbors (S_{d8}), or non-neighbors (S_e) with respect to a given unsummed product z_{ij}.

minimum-error parameter set.

$$\mathbf{u}_i = \begin{bmatrix} u_{i1} \\ \vdots \\ u_{in} \end{bmatrix} = \underline{\underline{S}}_i \mathbf{a}_i = \begin{bmatrix} z_{i1} & \sum_{S_4(z_{i1})} z_{pq} & \sum_{S_{d8}(z_{i1})} z_{pq} & \sum_{S_e(z_{i1})} z_{pq} \\ \vdots & \vdots & \vdots & \vdots \\ z_{in} & \sum_{S_4(z_{in})} z_{pq} & \sum_{S_{d8}(z_{in})} z_{pq} & \sum_{S_e(z_{in})} z_{pq} \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} \tag{5.7}$$

An alternative technique of estimation of crosstalk parameters works with the set of patterns with only one element at maximum. Each lenslet then defines a set of equations directly solvable for a, b, c and d.

A directly solvable system of 4 equations results for the same such input patterns by using the average of four-neighbors or diagonal eight-neighbors, or non-neighbors. The average over all N^2 imagelets forms a global crosstalk measurement.

$$\begin{aligned} u_{ij} &= az_{ij} \\ \bar{u}_4 &= b\bar{z}_{iq} \quad \text{for } z_{iq} \in S_4(z_{ij}) \\ \bar{u}_{d8} &= c\bar{z}_{iq} \quad \text{for } z_{iq} \in S_{d8}(z_{ij}) \\ \bar{u}_e &= d\bar{z}_{iq} \quad \text{for } z_{iq} \in S_e(z_{ij}) \end{aligned} \tag{5.8}$$

Separation of local from global crosstalk has motivation in the errors of the optical hardware. Local crosstalk can result from low-quality imaging lenses with poor point-spread function or alignment inaccuracies. Global crosstalk can be caused by scattered light and backreflections.

5.3 Performance Modeling and Characterization Results

The choice of a representative set of inputs and interconnection patterns is an exercise in approximation itself. An automated characerization 'suite' was presented and the results recorded. Evaluation of the parameters of the model guided this choice of test. Inputs and interconnection test patterns fall into three groups, designed to measure time-variation and non-uniformity, multiplication nonlinearity, or crosstalk. A repeated identical pattern and value indicates non-uniformity and time-variation. A progression sequence of inputs and weights at sampled intervals reveals nonlinearity, and a repertoire of binary patterns yield data on crosstalk.

Repeated identical patterns were presented ten times in a row, and the strength of each of the 256 weighted interconnections was recorded (unsummed products). Three sets of inputs and interconnections were used. One measured the response to weights and inputs at maximum (255). Another measured the response to weights and inputs equal to zero, and the third used a range of inputs spaced

equally from zero to maximum, and uniform submasks (all fan-in weights to one output neuron unit), each equal to one of the values of the inputs.

5.3.1 Time-Variation and Non-Uniformity Results

IP Time Variation	Avg.	Max	Min
Zero wt.s & inputs	0.556	1.55	0
Range wt.s & inputs	0.944	2.96	0
Max wt.s & inputs	8.28	14.6	1.83

Table 5.1: Results of measurement of standard deviation of the time-variation of output for all N^4 unsummed products for three data sets with both weights and inputs at maximum, an intermediated average value, and zero.

Results for standard deviation of time-variation are shown in Table 5.1, depicting average, maximum and minimum standard deviations over all unsummed products for the cases of both weights and inputs at 0, at 255, and as a range of values with average 64.

UP Time Variation	Avg.	Max	Min
Zero wt.s & inputs	0.122	0.483	0
Range wt.s & inputs	0.669	1.58	0
Max wt.s & inputs	7.74	12.98	3.77

Table 5.2: Results of measurement of standard deviation of the time-variation of output for all N^2 inner products for three data sets with both weights and inputs at maximum, an intermediated average value, and zero.

Statistics on the time-variation of the summed, inner product performance were gathered as well. Cancellation of lower and higher response than average resulted in lower variance than outer-product response (Table 5.2). The time-variation output was principally an artifact of the video equipment used, and may have resulted from variation in the analog-to-digital converter of the frame grabber card in the PC.

The data gathered with the maximum input and weight data set was analyzed differently to give non-uniformity statistics. The variation in response of the different output elements was found for each trial. An individual weight calibration step was performed to equalize the value of all unsummed product outputs, but significant non-uniformity was still observed. The results again show increasing variance with increased signal strength. Inputs and weights set to zero deviated by 0.42, range 1, but 255-valued inputs and weights deviated by 16.6, with range 54 (out of approximately 255).

A delay of .4 seconds was used when the LCTV signal was changed, and a 1.4 second delay was used after changing the signal to the CRT/LCLV combination. This was to allow the SLM components to stabilize, and represents a compromise between stability and speed of the system cycle time. The input and weight SLMs were continually sent the same signal over the ten trial period, but a long-term time response variation was observed, over a period of five seconds. Uniformity increased steadily over the ten trials as average signal level decreased, from a standard deviation of uniformity initial high of 13% to a final low of 8%. The slow response of the LCLV was most likely to blame, since it was operated with a bias voltage and frequency aimed at sensitivity.

5.3.2 Nonlinearity Results

Nonlinearity of average multiplication was modeled as described in Eq. 5.4 (a product of quadratics), but the specifics of our experimental system multiplication response curve made a modification convenient (one that is algebraically equivalent to Eq. 5.4:

$$z_{ij} = a(x_j + bx_j^2)\left(\begin{cases} c(w_{ij} + dw_{ij}^2) & \text{if } w_{ij} \leq 60 \\ ew_{ij} + f & \text{else} \end{cases}\right) + gx_j + hw_{ij}, \qquad (5.9)$$

a piecewise linear and quadratic curve yielded a close approximation to actual response (Fig. 5.2). An additive 'dark' noise proportional to input or weight biased the simulation graphs to be similar to experiment. The following values for parameters a,...,h were used to produce these curves: a=1.475, b=-0.00142875, c=0.000105, d=0.5, e=1.05, f=-12.75, g=13, and h=3.0. The weight multiplication

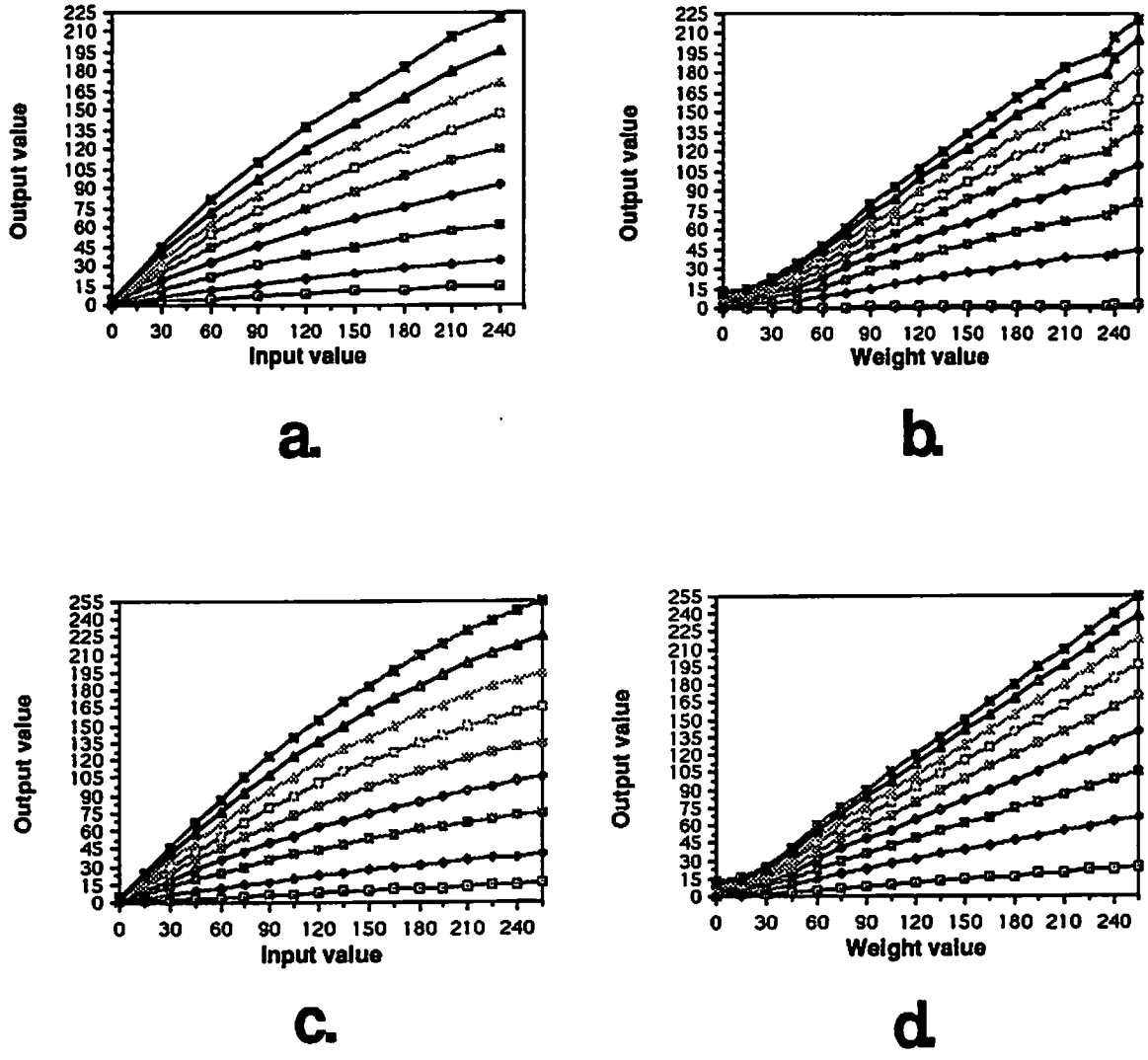


Figure 5.2: Actual system response curves; (a) as a function of input with a given weight parameter; (b) as a function of weight with a given input parameter; (c) parametric model simulation of response simulating (a); (d) parametric simulation response simulating (b).

had significant second-order behavior for $w \leq 60$, where $\beta_1/\beta_2 = 1/0.5$, but was closely modeled with no second-order component by $\beta_1/\beta_2 = 1/0$ for $w > 60$. Input linearity was good, $\alpha_1/\alpha_2 = 1/0.000105$. Bias metrics were not directly available from the exact form used, but the model assumed $\alpha_0 \times \beta_0 = 0$. Many techniques of curve-fitting can supply a minimum-error fit of the coefficients, but a semi-heuristic match sufficed for our purposes.

5.3.3 Other Characterization Results

Additional characterization data was provided by recording the duration of each substep of unipolar and time-multiplexed bipolar operation. A least-squares regression plotting actual product as a function of expected ideal product was performed for both summed and unsummmed products on the range test data. The ideal graph would be a line from the origin at forty-five degrees. Correlation coefficients correlating actual Y_i and ideal X_i multiplication performance for summed and unsummed products were 0.83 and 0.98 respectively, for a linear curve-fitting minimizing the sum of squared error, where correlation coefficient is defined as the customary sample correlation coefficient

$$R = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{(\sum_i (X_i - \bar{X})^2 \sum_i (Y_i - \bar{Y})^2)^{1/2}}. \tag{5.10}$$

Figure 5.3 shows the results for multiple simultaneous inner products of a range of values on input, and a range of weights, each used for all of one ouput neuron unit's fan-in weights. Figure 5.4 shows the same data for unsummed products.

A timing analysis was conducted by embedding calls to the PC clock within the program controlling the experiments. Elapsed time could then be calculated between calls, and the duration of each portion of operation can then be examined individually. Arithmetic operations purely within the contol program executed faster than the hundredths of seconds resolution of the clock. Operations on the frame grabber card took measurably long intervals however; writing all N^4 globally scaled weights, summing the pixels of a sampling region, and writing black between sample regions on the card. An intentional delay was applied to allow the SLM devices to stabilize; 0.4 seconds for the LCTV, and 1.4 seconds

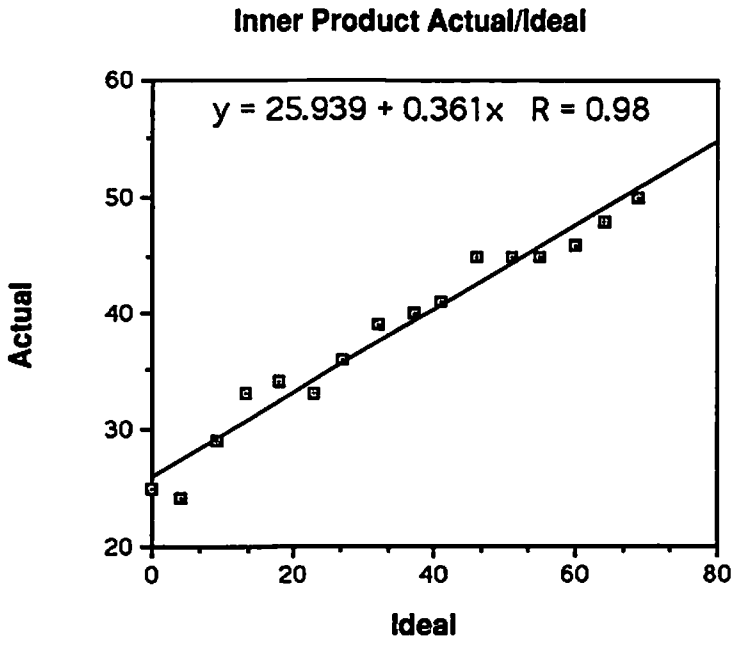


Figure 5.3: Graph of actual system response versus theoretical ideal response (ideally a 45 degree line) for multiple simultaneous inner products of a range of values; correlation coefficients and linear best fit coefficents are shown.

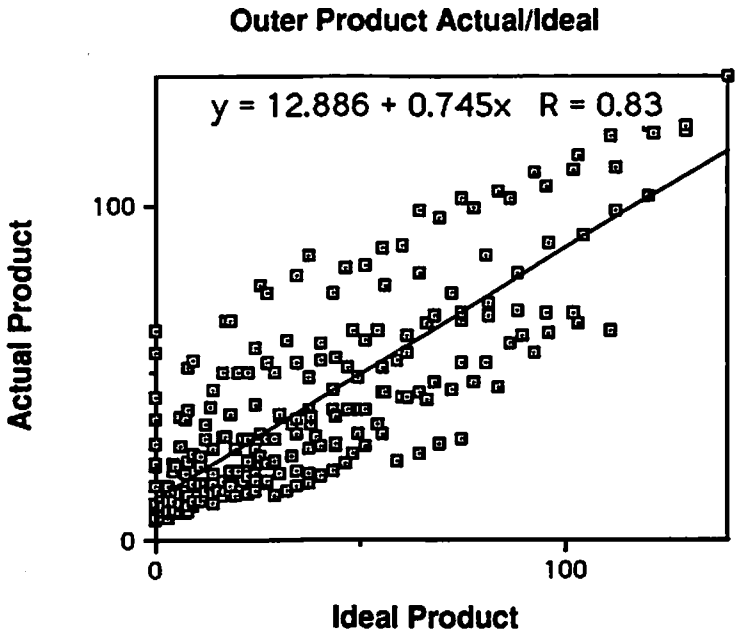


Figure 5.4: Graph of actual system response versus theoretical ideal response for multiple simultaneous outer products of a range of values; correlation coefficients and linear best fit coefficents are shown.

for CRT/LCLV combination SLM. Figures 5.5, 5.6 depict timing results for the unipolar and time-multiplexed bipolar inner product operations.

5.3.4 Crosstalk Estimation Results

Two sets of experiments were performed, as diagrammed in Fig. 5.7. One set employed complete, maximum interconnections with a variety of binary input patterns. The second left all inputs at maximum while varying submask pattern, keeping each submask pattern identical at one time. The binary pattern with one input element active, or one submask element active, is shown.

The simpler crosstalk estimation method was used with the set of patterns with one pixel at maximum only. Solving for b/a, c/a, d/a and taking the average over all N^2 Euclidean-basis vector inputs yielded the following. Global crosstalk was $12.4\% = d/a$ for input patterns, and 10.0% for weight patterns. Input patterns showed $4.6\% = c/a$ 4-neighbor, and $1.2\% = b/a$ diagonal 8-neighbor crosstalk, while weight pattern crosstalk was 0.41 and 0.28 respectively.

The other method of crosstalk estimation (numerical minimization) works with patterns that have more than one element at maximum. Many types of inputs and identical arrays of submask images were used in two sets of experiments as before. Both Cholesky decomposition [33] and singular value decomposition [34] numerical methods were used, and compared to find identical results.

Numerical solution yielded a consistent scale factor a for each set of data. The input pattern sequence was invariant under the number of active binary input elements, while the submask patterns increased non-linearly with the number of zero-value submask elements.

Figure 5.8(a) shows the results of estimation of global crosstalk. N^2 Euclidean-basis vector patterns a, b, c, d's were averaged for the first data point; the 4-off patterns show 1 row on and 1 column on averaged over all 4 cases, 8-on show row pairs, column pairs averaged over all 6 cases; row off, column off over 4 cases, and one input of N^2 off only (15 on) averaged over all $N^2 = 16$ cases. Row and column combinations did not differ significantly in global crosstalk. The worst case was shown to be the one-on pattern for both inputs and weights. Input device global crosstalk remained closer to a constant proportion of total light signal intensity,

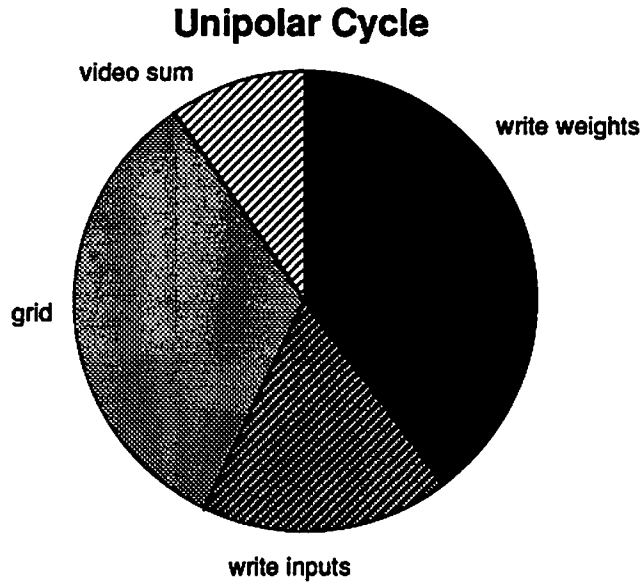


Figure 5.5: Cycle timing breakdown for unipolar optical cycle. Writing weights and inputs to the SLM devices, writing black between sample points of captured output image and summation over all inputs to a neuron unit represented the only parts of the cycle with durations measurable on a hundredths-of-seconds time scale.

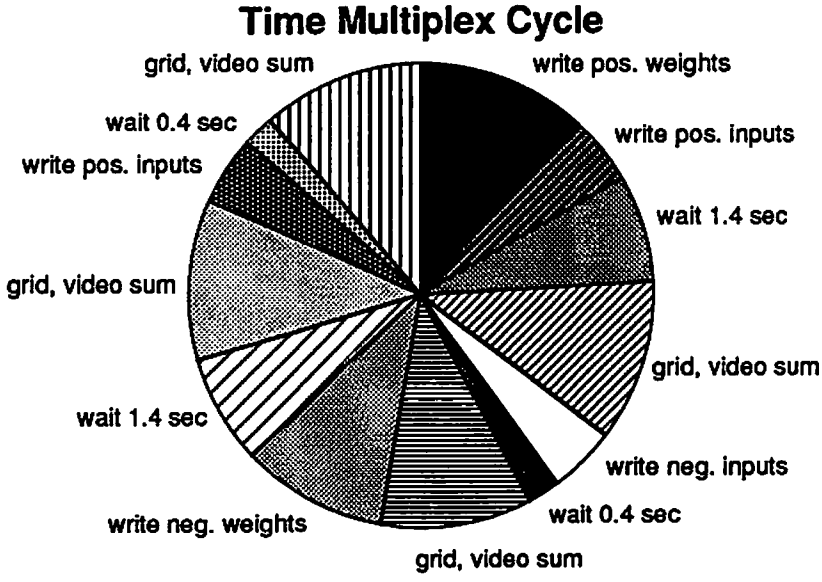


Figure 5.6: Cycle timing breakdown for time-multiplexed bipolar optical cycle. All four combinations of positive and negative inputs and weights must be presented.

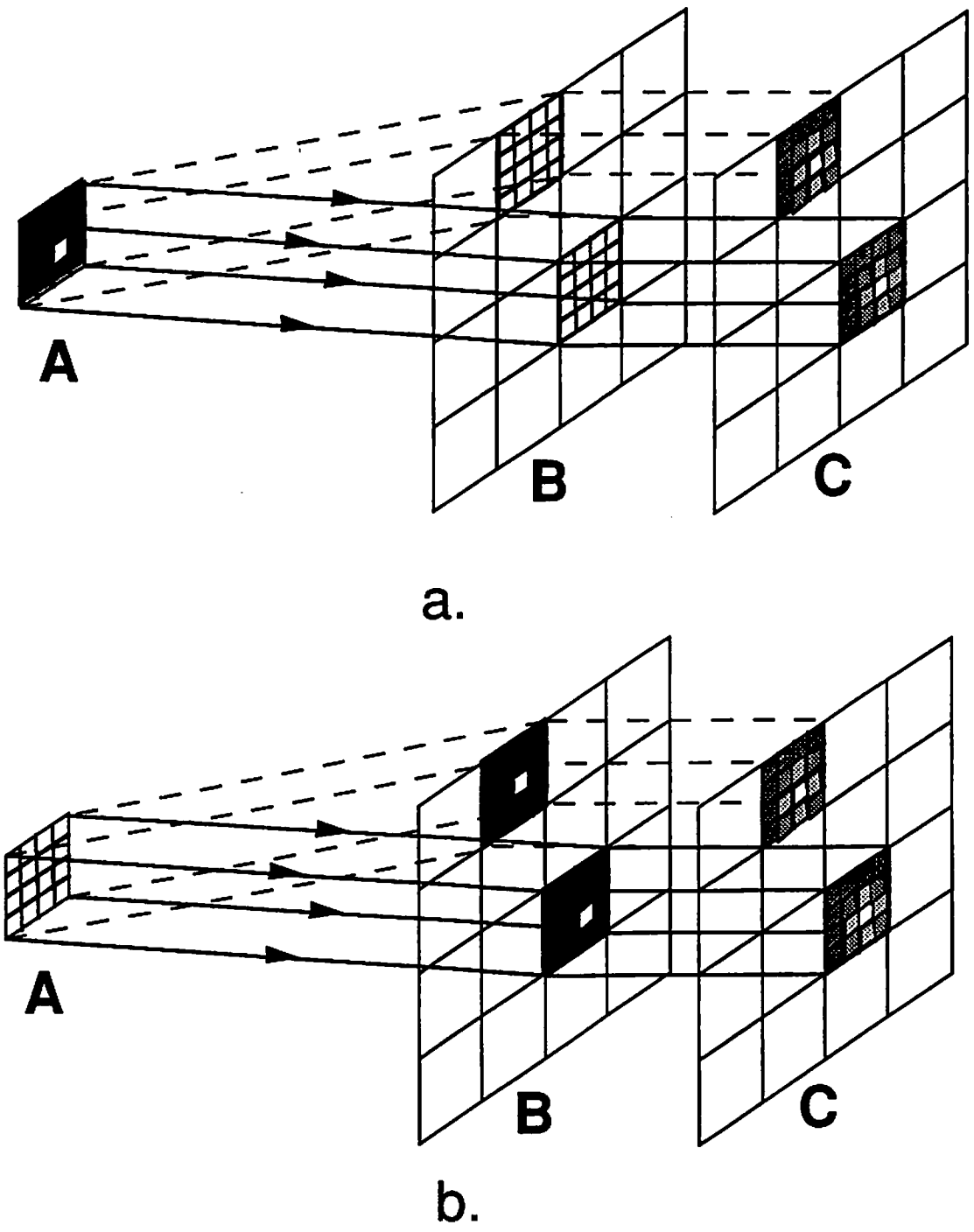


Figure 5.7: Graphic depiction of the effect of local and global crosstalk between input and weight elements; (a) output result with single non-zero input, and all interconnections weights set to maximum; (b) output result with single non-zero fan-in weight to each output neuron unit, and all inputs at maximum. Ideal output image for both would have one fully activated output without a neighborhood of partial activation, and without a global partial activation.

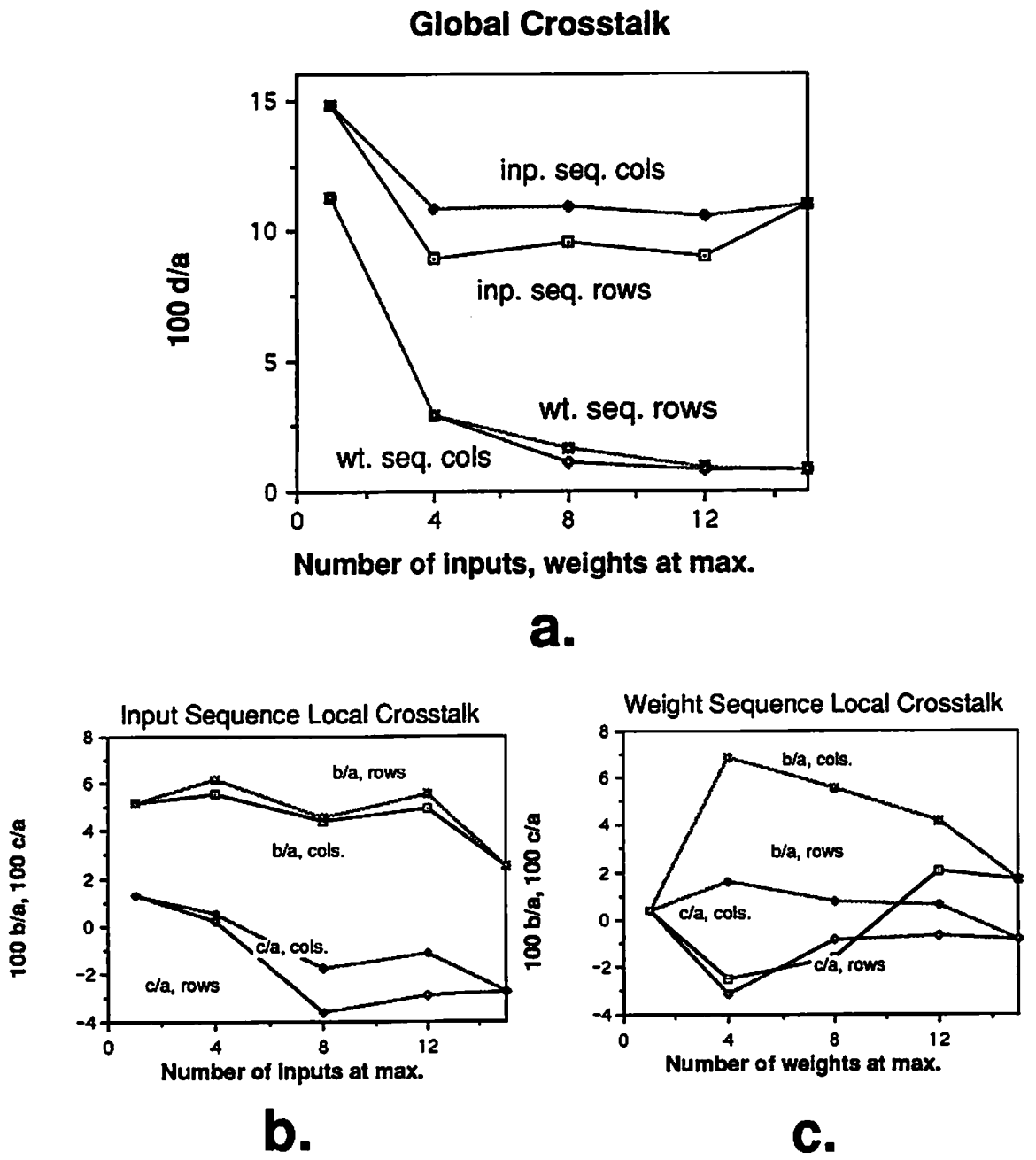


Figure 5.8: (a) Global crosstalk as a function of total incident signal for the input device with all weights at maximum varied for the one-on pattern, while global crosstalk as a function of interconnection weight with all inputs at maximum decreased with the number of inputs at maximum; (b),(c) Local crosstalk as a function of total incident signal for the input device with all weights at maximum was anisotropic with positive horizontal/vertical crosstalk and negative diagonal crosstalk, while crosstalk as a function of interconnection weight with all inputs at maximum was strongest in the horizontal/vertical direction with column patterns, while being essentially zero in diagonal ones. Row patterns yielded slightly negative crosstalk both directions.

while weight device crosstalk decreased monotonically, but nonlinearly with the number of elements at maximum.

Figure 5.8(b),(c) show the results of estimation of local crosstalk. Again, combinations of columns, and rows were examined for 4-neighbor and diagonal 8-neighbor crosstalk. Severe anisotropy was observed from the CRT/LCLV weight SLM, while the LCTV input device yielded an essentially consistently anisotropic response, having similar positive horizontal/vertical crosstalk with both row and column patterns, and negative diagonal crosstalk with both. The weight local crosstalk was most pronounced with horizontal/vertical 4-neighbors,with column patterns. Row patterns tended toward negative crosstalk for both row and column patterns. The physical scan-line operation of the CRT electron gun is inherently more anisotropic than the active matrix TFT LCD screen. (Negative crosstalk can be accounted for by action of the compensation measures described previously.)

Chapter 6

Discussion and Conclusions

6.1 Applicability of Work

The LAP provides a versatile example of an incoherent optical matrix-vector multiplier. Complete interconnection of 2-D arrays is accomplished with refractive free-space optics. Bipolar analog calculation of inner products, outer products and vector sums can be performed in one parallel step. Performance of the overall system must account for the different requirements on reconfiguration speeds of the input and interconnection weight SLMs.

The LAP system forms a general and flexible framework for the implementation of many kinds of neural networks. Bipolar signals and weights involve additional (independent and parallel) tasks that could be implemented in special-purpose hardware. The LAP has potential for the implementation of fully-interconnected high-speed parallel networks of the order of two thousand neuron units.

6.2 Algorithm/Hardware Interplay

The two adaptive neural models (perceptron and competitive learning) implemented possess many common features. This has allowed the experimental implementation of both models on one optical hardware system with minor modifications. The two models both require inner product potential formation and

maximum output element detection. The perceptron in effect performs a 'normalization' by adding and subtracting the input pattern at one update step. Competitive learning employs an explicit normalization of fan-in weight at the expense of additional computations. The perceptron experiments showed limited classification capability (in terms of total number of patterns classified by one interconnection weight set), but successful classification of small pattern sets was successfully demonstrated. Crosstalk of input elements, non-uniformity and time-variation of the response of the experimental system contributed to deviation from theoretical performance of both models. Initial weights proved to have significant influence on resulting classification in competitive learning, and also on the success or failure of the perceptron classifier. In general, experimental implementation allowed investigation, at the algorithmic level, of inaccuracies characteristic of lenslet array systems. Inaccuracies guided specialization of neural models for the purpose of optical/optoelectronic implementation, specifically margin variation perceptron and 'enforced' competitive learning. Many mathematically trivial details make a significant difference to implementation, such as the need for bipolar signal values or normalization of signals to operate with a limited dynamic range.

6.3 Characterization and Modeling Process

An important aspect of the work presented herein was the interplay between characterization, modeling/simulation and development. Characterization guided development, modeling influenced specific aspects of characterization, and the results of characterization both influenced modeling and provided a specific data point for comparison of experimental results with simulation results. This approach is a start on a very general methodology of optical neural network or interconnection system development.

Theoretical and empirical considerations provided inspiration for the mathematical structure motivating quantitative performance metrics. Consistent application of this structure allows evaluation of design decision alternatives. Comparison of general neural network and optical interconnection hardware is possible using slight modifications. The development of the model also establishes the

groundwork for future, more accurate modeling, and eventual use of the model as a predictor of the results of iterated operation of neural algorithmic variants.

The characterization experiments fall into groups designed to measure the model parameters. Most of the characterization experiments used binary interconnection patterns to examine crosstalk and non-uniformity. A more extensive characterization effort could employ a variety of analog interconnection patterns as well. The same linear least-squares estimation as we applied could then be performed using this additional data as well.

Increasing the detail of the model results in a more extensive quantitative metric set, but at a cost of decreasing reliability/data reduction, given a fixed amount of characterization information. Employment of analog-valued interconnection patterns allows more detailed modeling efforts than our binary-valued pattern based estimation.

Optoelectronic system development (and optoelectronic fabrication) is an intricate process with many subtle parameters to be optimized. Characterization and modeling provide information valuable in selection of these parameters.

6.4 Model Refinements and Extensions

The simplifications of the parametric model presented thus far could be relaxed at the cost of increasing computational demands. The required accuracy of simulation varies with the application at hand. In general, replacing the single parameters described with individual arrays of parameters, (one for each physical element) will increase accuracy while increasing storage and computation requirements. These arrays can be individually matched to an actual fabricated system, or use random values generated with a statistically equivalent distribution.

The maximum and minimum interconnection strength of each weight varied from one physical element to another in the experimental system. Arrays of minimum and maximum interconnections strengths would more accurately simulate our experimental system. The multiplication linearity also varied with physical element, providing another opportunity to use an array in simulation.

Local crosstalk values could be modeled for larger neighborhoods of physical elements than the nine-neighborhood used in our work. Five by five, or more, neighborhoods of physical elements could realistically share signal crosstalk. Extension to optical interconnection systems lacking discrete pixels (such as some holographic systems) could be facilitated by defining local crosstalk neighborhood contributions as integrals rather than sums.

Isolating and characterizing the crosstalk in the free-space optical interconnection between two individual SLM/active element planes was not modeled in our simulations, but was explored in characterization experiments. Significantly different characteristics found for the input and weight imaging stages could be explicitly modeled, providing more simulation accuracy at a cost.

6.5 Optoelectronic Implementation Considerations

A particular feature of this investigation has been an interest in practical issues of implementation of neural networks using optics, lenslet arrays in particular. Bulk optical implementation experience has led to the development of a list of considerations foreseen for optoelectronic implementation. Optoelectronic researchers often describe a *smart SLM* composed of *smart pixels*, consisting of optical input devices(detectors), optical output devices (sources or modulators), and digital or analog electronic computation and memory circuits. This smart pixel can be identified with a hardware unit performing the neuron unit function, or weighting unit function. The following section considers the optoelectronic system fabrication implications of the general network implementation requirements developed thus far.

While a multilayer network can be mapped into a single layer network with the same number of neuron units (but more interconnections), it may be impractical to perform this mapping at large scaleup of the network. Such a mapping is appropriate to an implementation of a network with two opposing planes of optoelectronics, which we will call a reflective geometry (input and output on both

sides of a plane). In contrast, a transmissive geometry would allow the cascade of successive neuron unit layers.

Inverted cavity quantum well device technology offers precisely the capability for a transmissive geometry, utilizing a transparent substrate. A cascade of multiple planes of optoelectronics (neuron unit layers) is thus achievable. This scenario avoids the feedback step required upon mapping a multilayer network to a single layer. This feedback step presents little problem if the input and output neuron units are physically adjacent in a reflective mapping. Going one step further, we may associate either all of the fan-in weights of the output neuron unit *or* the fan-out weights of the input neuron unit in that same physical area, leading to a fan-in or a fan-out *superpixel.*

We describe a high level design requiring only reflective devices: reflective modulators, lasers and detectors fabricated on just one side of an opaque substrate, operating outward from the substrate. This choice would be suitable directly implementing a single layer network or a multilayer network mapped to a single layer.

The high-level neural and weight smart pixel model is presented in an attempt to transcend specifics of implementation. We choose to assume planar arrays of superpixels with electronic summation of weighted inputs, and electronic multiplication for outer product weight update. Figure 6.1 depicts a design for a fan-in superpixel of a single interconnection-layer network. Communication local to the superpixel is done electronically (curved bidirectional arrows in Fig. 6.1(a)), while the global fan-out of input signals is done optically in reflection (straight arrows in Fig. 6.1(a)). The input and output neuron unit are packaged with all fan-in weights in a planar patch (Fig. 6.1(b)) within a 2-D array. Arithmetic operations are indicated for formation of output and weight updates given by the LMS rule, where $\delta_i = \psi'(p_i)(t_i - y_i)$ where t_i is the expected output corresponding to the input (Fig. 6.1(c)) [3]. The optical interconnection can be implemented, for example, either by phase grating fan-out at the neuron unit (or reflection) plane, or a single lenslet associated with a superpixel can image all diffusely reflected neuron unit optical outputs (from other superpixels) to the weight array of the

superpixel. Feedback of output to input (shown by a dashed arrow) is required only for iterative single layer, or multiple layer networks.

Extensive semiconductor laser research has led to the development of high-speed sources, and high-bandwidth long-distance interconnections. This technology could yield very high throughput in the shorter-range application of smart pixel neural networks. Unfortunately, fabrication of such systems is a much more extensive and expensive effort than construction of bulk optical implementations of neural networks. Collaboration between systems and device researchers is becoming increasingly important, such as in deciding the *desired* scale of a network, while estimating the yields associated with different scale network designs given a specific materials system.

6.6 Conclusions

An important tool of the investigation was the experimental system. Building a working prototype system gives important insight towards identifying the specific challenges of the optical interconnect technology. Physically adjacent positive and negative channels were intentionally used in space-multiplexed bipolar coding in the interest of uniformity of the *difference* of the two channels while the time-multiplexed bipolar coding uses the same physical connection, providing even more uniformity of the difference of positive and negative signals (still subject to time-variation of output). These are examples of the kind of self-compensation at the algorithmic level that facilitate optical implementation. In contrast, much of the neural literature has investigated algorithms purely mathematically, or with a strong bias to conventional serial computer accuracy and representation.

The experimental optical system allowed investigation of realistic inaccuracies at the neural-algorithmic level. The extent to which optoelectronic systems will share the inaccuracies of the bulk optical LAP system is unclear, but nonuniformity and crosstalk appear to be common problems in many free-space optical interconnect systems, including holographic interconnect systems [35].

The results allowed comparison of theory with experiment. The effects of input crosstalk and nonuniformity were shown to have significant influence on the convergence of the perceptron and maximum networks, limiting the number of

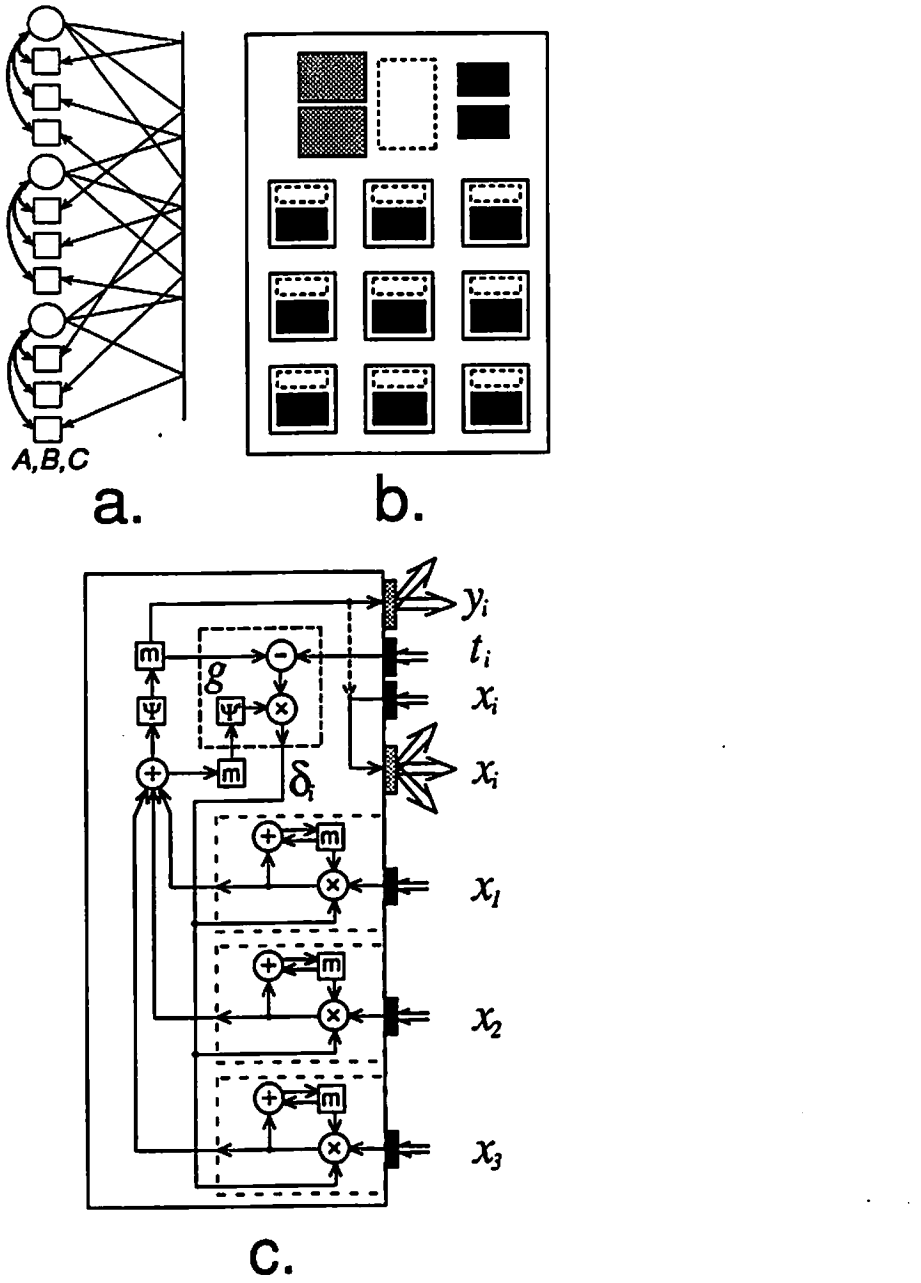


Figure 6.1: Fan-in superpixel; (a) reflective mapping of generalized neural network; input and output neuron units are combined to a single neuron unit with optical input broadcasting (straight lines) and electronic, bidirectional (curved lines) weight-output neuron unit connection; (b) top view of high-level design for an optoelectronic implementation of the superpixel; layout consists of neuron unit circuitry, optical input/output, and an array of weight pixels (gray shading, sources; black, detectors; dashed boxes, circuits; Ψ, neuron unit nonlinearity; m, analog-value memory); (c) side view depicting information flow, processing and storage.

patterns that could be successfuly classified by one set of optical interconnection weights. The stability of the competitve learning algorithm was explored by adapting the optical interconnections, as well as sensitivity to initial weights and pattern presentation order. Both oscillatory and stable results were obtained. Development of neural alogorithmic variants that are tolerant of crosstalk could help increase the size of the classifiied pattern sets obtainable with similar optical systems. Variants described in this paper were shown tolerant of time-variation of output.

Incoherent, intensity-coded multiple imaging for fan-out represents a reasonably near-term choice for free-space optical interconnection in optoelectronic architectures. The system characterization and perfomance modeling techniques described could augment device characterization and performance modeling.

Experimental implementation is thus a valuable tool in assessing the feasibility of LAP incoherent interconnection, and the effect of parallel hardware characteristics on neural network performance.

6.7 Future Directions

Maximum performance demands dedicated, high-speed optoelectronic SLMs/sources/detectors. Free-space optical interconnection must augment parallel electronic computation. Alternative computing technologies will supplement (not replace) the serial programmable computer.

Identification of appropriate roles for optical techniques remains a research area. Increasing attention is given to application-oriented integration of optoelectronic systems. Supplementation of parallel computer power with optical communication presents one established application area ('optical bus'). Another is consideration of neural algorithm requirements versus optoelectronic hardware perfromance. The work presented here utilized a bulk-optical system, and considered a limited set of adaptive neural algorithms. Consideration of multilayer (backpropagation) networks and dedicated (LAP or other) optoelectronic systems remains unexplored.

References

[1] M. R. Feldman, S. C. Esener, C. C. Guest, and S. H. Lee, "Comparison between optical and electrical interconnects based on power and speed considerations," Appl. Opt. **27**, 1742–1751 (1988).

[2] D. A. B. Miller, "Optics for low-energy communication inside digital processors," Opt. Lett. **14**, 146–148, (1989).

[3] D. E. Rumelhart, J. L. McClelland, and the PDP research group, *"Parallel Distributed Processing,"* (MIT Press/Bradford Books, Cambridge, Mass., 1992).

[4] R. P. Lippman, "An introduction to computing with neural nets," IEEE ASSP April 1987, 4–22.

[5] M. A. Arbib, *"The Metaphorical Brain 2 Neural Networks and Beyond,"* (John Wiley & Sons, New York, NY, 1992).

[6] B. K. Jenkins and A. R. Tanguay, "Photonic Implementation of Neural Networks," in *Neural Networks for Signal Processing*, B. Kosko, ed., 287–382, (Prentice-Hall, Englewood Cliffs, New Jersey, 1992).

[7] R. O. Duda, P. E. Hart, *"Pattern Classification and Scene Analysis,"*, (John Wiley & Sons, New York, NY., 1973).

[8] I. Glaser, "Methods for information processing with spatially incoherent light," in *Progress in optics*, E. Wolf, ed., Vol. XXIV, 389–511, (North-Holland, New York, 1987) 1987.

[9] I. Glaser and L. Perelmutter, "Optical interconnections for digital processing: a noncoherent method," Opt. Lett. **11**, 53–55 (1986).

[10] L. Perelmutter and I. Glaser, "Digital incoherent optical interconnections," in *Proc. SPIE* **700**, 215–220 (1986).

[11] I. Glaser, "Lenslet array processors," Appl. Opt. **21**, 1271–1280 (1982).

[12] I. Glaser, "Representing bipolar and complex imagery in noncoherent optics image processing systems: comparison of approaches," Opt. Eng. **20**, 568–573 (1981).

[13] N. Farhat, D. Psaltis, A. Prata, and E. Paek, "Optical implementation of the Hopfield model," Appl. Opt. **24**, 1469–1475 (1985).

[14] N. Farhat and D. Psaltis, "Optical implementation of associative memory based on models of neural networks," in *Optical Signal Processing*, J. Horner, ed., 129–162, (Academic, New York, New York 1987).

[15] S. Miyahara, "Automated radar target recognition based on models of neural nets," PhD thesis, U. Pennsylvania, (1987).

[16] D. J. Wiley, I. Glaser, B. K. Jenkins, and A. A. Sawchuk, "Lenslet array based dynamic optical neural network," in *Conference Record of the 1990 Topical Meeting on Optical Computing*, Kobe, Japan, Proc. SPIE **1359** 245–246 (SPIE, Bellingham, Wash., 1990).

[17] T. Lu, S. Wu, X. Xu, and F. T. S. Yu, "Two-dimensional programmable optical neural network," Appl. Opt. **28**, 4908–4913 (1989)

[18] T. Lu, K. Choi, S. Wu, and F. T. S. Yu, "Optical disk based neural network," Appl. Opt. **28**, 4722–4724 (1989).

[19] S. Shin J. Jang and S. Lee, "Programmable quadratic associative memory using holographic lenslet arrays," Opt. Lett. **16**, 838–840 (1989).

[20] S. Shin, "Programmable optical interconnections using holographic lenslet arrays," In *"Korea-USA Joint Workshop on Optical Neural Networks,"* Seoul, Korea 1990.

[21] N. M. Barnes, A. W. O'Neill and D. Wood, "Rapid, supervised training of a two-layer, opto-electronic neural nework using simulated annealing," Opt. Comm **87** 203–206 (1992).

[22] Y. Nitta, J. Ohta, K. Mitsunaga, S. Tai and K. Kyuma, "Optoelectronic associative memory using and advanced optical neurochip," Appl. Opt. **30** 1328–1330 (1991).

[23] D. J. Wiley, I. Glaser, B. K. Jenkins, and A. A. Sawchuk, "Incoherent dynamic lenslet array processor," submitted to *Appl. Opt.*

[24] Lenslet array MRP-110 from Aeroflex Laboratories, Plainview, New York.

[25] LCTV display from the Sony GV-8 Video Walkman (Tokyo, Japan).

[26] Ikegami PM-580 (Japan).

[27] Hughes twisted nematic LCLV serial number LT2118 (Carlsbad, Calif.).

[28] Hitachi camera model KP-140 (Tokyo, Japan).

[29] The mask (LCLV) frame grab card was a Matrox PIP-1024 (Dorval, Quebec). The input/output card (LCTV and CCD) was a Data Translation DT-2851 (Marlboro, Mass.). Both have a nominal resolution of 512 by 512 (actual resolution is 512 by 480) and 256 (8 bit) gray levels. The Data Translation card has simultaneous input/output capability, and built-in hardware for region summation and graphics, which were found useful for our work. Two different cards were needed in order to avoid conflicts on the PC bus.

[30] T. Sakano, K. Noguchi and T. Matsumoto, "Optical limits for spatial interconnection networks using 2-D optical array devices", Appl. Opt. **29** 1094–1100 (1990).

[31] B. Kosko, *"Neural Networks and Fuzzy Systems,"* (Prentice Hall, Englewood Cliffs, NJ., 1992).

[32] D. Psaltis, D. Brady K. Wagner, "Adaptive optical networks using photorefractive crystals," Appl. Opt. **27** 1752–1759 (1990).

[33] G. W. Stewart, *"Introduction to Matrix Computations,"* (Academic Press, Orlando, Fla., 1973).

[34] W. H. Press, B. P .Flannery, S. A. Teukolsky and W. T. Vetterling, *"Numerical Recipes in C,"* (Cambridge University Press, New York, 1988)

[35] W. Zhang, K. Itoh, J. Tanida and Y. Ichioka, "Parallel distributed processing model with local space-invariant interconnections and its optical architecture," Appl. Opt. **29**, 4790–4797 (1990).