# Analysis and Verification of an Analog VLSI Incremental Outer-Product Learning System

Gert Cauwenberghs, *Student Member, IEEE,* Charles F. Neugebauer, *Student Member, IEEE,* and Amnon Yariv, *Fellow, IEEE*

*Abstract*—**An architecture is described for the microelectronic implementation of arbitrary outer-product learning rules in analog floating-gate CMOS matrix–vector multiplier networks. The weights are stored permanently on floating gates and are updated under uniform UV illumination with a general incremental analog four-quadrant outer-product learning scheme, performed locally on-chip by a single transistor per matrix element on average. From the mechanism of floating gate relaxation under UV radiation, we derive the learning parameters and their dependence on the illumination level and circuit parameters. It is shown that the weight increments consist of two parts: one term contains the outer product of two externally applied learning vectors; the other part represents a uniform weight decay, with time constant originating from the floating gate relaxation. We address the implementation of supervised and unsupervised learning algorithms with emphasis on the delta rule. Experimental results from a simple implementation of the delta rule on an 8×7 linear network are included.**

## I. Introduction

In an effort to implement parallel algorithms for information and signal processing [1]–[3] more efficiently and directly, there has been a trend toward more distributed and densely interconnected hardware architectures that process information in a highly parallel fashion. In an analog hardware environment the high accuracy found in digital implementations is traded for the simplicity and interconnectivity of their analog equivalents, which for the same silicon area allow for sufficient redundancy at the system level to compensate loss of accuracy at the process level. Analog neural hardware [4], [5] belongs to this category, but applications are also found in the areas of signal processing and process control for linear and nonlinear problems. The task of the connectivity between elements reduces to the function of analog matrix–vector multiplication. Also, in many cases the task of adjusting the connectivity pattern to optimize computation can be formulated into some kind of local incremental learning rule for the connection strengths. Specifically, an important class of supervised and unsupervised learning tasks satisfies learning rules of the incremental outer-product type [1], [2], where the specified change of a connection strength is proportional to the product of two terms originating from the two elements it connects. This class covers Hebbian learning, the delta rule, and error back-propagation, or the generalized delta rule [1].

A variety of CMOS analog four-quadrant matrix–vector multipliers have been developed for different accuracy needs and design constraints [6]. Fully analog implementations require some form of local on-chip analog storage to set the connection matrix, either dynamically on capacitors using a periodic refresh scheme [6]–[8] or permanently on floating gates [9]–[11]. The dynamical volatile approach offers the advantage of modularity, allowing fast external reprogramming of the connections as it is needed, whereas the nonvolatile approach is preferable for applications requiring a fixed connection matrix that only occasionally needs to be modified or reprogrammed. So far, in both categories, the main effort has been on the programmability [6], [7], [9], [10] rather than adaptivity of the connections. When the computation task is not *a priori* defined or the programmed connection strengths suffer from distortions caused by offsets and process variations induced at the fabrication stage, a recursive adaptation scheme, i.e., updating connections as long as discrepancies persist, is advisable. For this purpose, integrated parallel adaptation architectures are far more adequate than external, often serial, programming schemes. Few integrated parallel adaptive architectures have been suggested so far [8]. In the system outlined here [12], [13], which belongs to the category of nonvolatile networks, both functions of analog matrix–vector multiplication and parallel adaptation of the connection matrix are combined in a densely integrated CMOS architecture. Floating gates are employed for the permanent storage of the connections. Illumination of the circuit with uniform ultraviolet light activates adaptation of the weights. Any incremental outer-product adaptation rule can in principle be implemented on this system, by supplying the appropriate learning vectors which form the outer-product increments to the network. In addition, as will be shown below, during adaptation the weights are subject to a uniform decay.

In what follows, we will describe the architecture, examine the individual and collective functional properties of its constituents, derive a model of the learning system and extract its parameters, discuss the implementation of standard learning rules, and verify the learning capability of the system on a small linear network. To offer a better understanding of the mechanism of weight adaptation and its impact on the learning characteristics, the fundamentals of the floating gate relaxation under UV illumination are introduced first.

Manuscript received July 24, 1991; revised November 7, 1991. This work was supported by the Air Force Office of Scientific Research, the Defense Advanced Research Projects Agency, and the U.S. Army CECOM Center for Signal Warfare.

G. Cauwenberghs is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125.

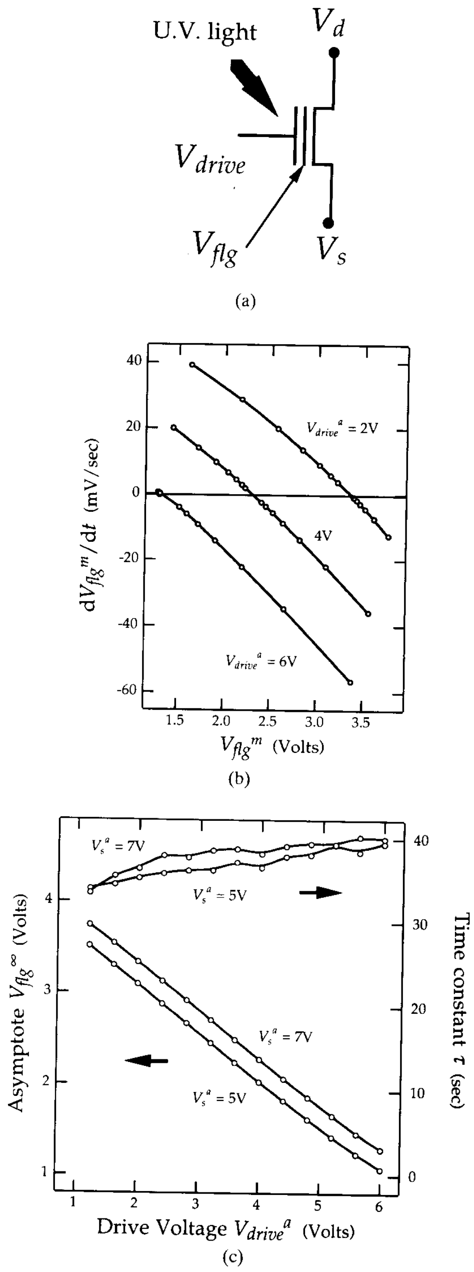
C. F. Neugebauer and A. Yariv are with the Department of Applied Physics, California Institute of Technology, Pasadena, CA 91125.

IEEE Log Number 9105809.

1045–9227/92$03.00 © 1992 IEEE

## II. Floating Gate Relaxation Under UV Radiation

In contrast to the use of ultraviolet radiation in digital nonvolatile memory circuits [14], recent applications [11], [15], [16] employ UV light to adapt analog circuitry. The analog voltage on a floating gate is not directly accessible for measurement or computation but its value can be sensed or made available indirectly through the transconductance of the transistor it belongs to. Voltage increments on floating gates are induced by charge transport through the surrounding oxide, roughly proportional to both the intensity of the incident UV light and the electric field in the oxide. Therefore, under uniform UV radiation the sign and amplitude of such voltage increments are controlled by the voltages of electrodes in the neighborhood of the floating gate, relative to the floating gate voltage [16]. Besides this UV-controlled oxide conductance, the capacitive coupling between the floating gate and the electrodes resulting from the oxide dielectric properties contributes to the dynamic aspects of the adaptation process.



(a)

(b)

(c)

Fig. 1. (a) Floating gate nMOS transistor. (b) Relaxation dynamics. (c) Adaptation characteristics.

A poly-silicon floating gate of an MOS transistor with a nearby second-poly drive electrode is depicted schematically in Fig. 1(a). The drive electrode serves a dual purpose. First, under UV adaptation, the drive voltage controls the direction and size of the voltage increments on the floating gate. Second, in the measurement mode with the UV disabled, the voltage on the drive electrode defines a reference for the floating gate voltage arising from the capacitive coupling between the drive electrode and the floating gate. In terms of the learning system these two situations would correspond to the adaptation phase and the network computation stage, respectively. The two phases cannot be combined at the same time; an arbitrary voltage on the drive electrode perturbs a measurement of the floating gate voltage by capacitive coupling. In order to uniquely define the value of the floating gate voltage, the drive electrode voltage in the measurement phase needs to be kept at a fixed reference value. A recursive learning scheme requires, along the adaptation process, periodic sampling of the current state of the network in the measurement mode to update the learning vectors for the next adaptation iteration. Therefore, the quantity of interest here is the floating gate voltage in the measurement reference mode and its incremental changes under a time-varying drive electrode voltage in the adaptation phase. To characterize the interaction between the floating gate, the drive electrode, and the substrate, a linear resistive and capacitive network model for the oxide surrounding the gate is assumed, with conductance values proportional to the UV illumination intensity. As such, it can be shown that the evolution of the floating gate voltage in the measurement mode, $V_{\rm flg}{}^{m}$, satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}V_{\rm flg}{}^{m} = -\frac{1}{\tau}\left(V_{\rm flg}{}^{m} - V_{\rm flg}{}^{\infty}\right). \tag{1}$$

The UV relaxation time constant, $\tau$, results from the parallel oxide conductance and capacitance $RC$ constant, and the asymptotic programming voltage, $V_{\rm flg}{}^{\infty}$, depends on the drive electrode voltage according to

$$V_{\rm flg}{}^{\infty}(t) = \lambda_{\rm drive}{}^{a}\ V_{\rm drive}{}^{a}(t) + \lambda_{\rm drive}{}^{m}\ V_{\rm drive}{}^{m}{}_{0} \tag{2}$$

with $V_{\rm drive}{}^{a}(t)$ the drive voltage on the electrode during adaptation, and $V_{\rm drive}{}^{m}{}_{0}$ the fixed reference electrode voltage in the measurement phase. The coefficient $\lambda_{\rm drive}{}^{a}$, describing the impact of the drive voltage under UV illumination on the asymptote, indicates a measure for the efficacy of the drive electrode adapting the voltage on the floating gate. The quantity $\lambda_{\rm drive}{}^{m}$ represents the electrode–gate capacitive coupling. The derivation of (1) and (2) from a formal analysis of the floating gate relaxation has been omitted here, as it is not of direct interest. A complete analysis, investigating the dependence of $\tau$, $\lambda_{\text{drive}}{}^{a}$ and $\lambda_{\text{drive}}{}^{m}$ on geometry and process parameters, will be given in a forthcoming paper. For completeness, it suffices to mention that, whereas for the capacitive coupling the inequality $0 \leq \lambda_{\text{drive}}{}^{m} \leq 1$ applies, the drive efficacy $\lambda_{\text{drive}}{}^{a}$ can take any value between $-1$ and 1, depending on the relative strength of the capacitive and conductive coupling between the drive electrode and the floating gate. The time constant, $\tau$, exhibits an inverse proportionality with the UV intensity; the asymptote coefficients $\lambda_{\text{drive}}{}^{a}$ and $\lambda_{\text{drive}}{}^{m}$, on the contrary, show no significant dependence on the illumination strength. Similarly, the geometry of the layout tends to affect mainly the time constant, and the asymptote to a lesser extent. Typical observed values for $\lambda_{\text{drive}}{}^{a}$ and $\lambda_{\text{drive}}{}^{m}$ in our design are $-0.5$ and $0.8$ respectively, and with an UV light intensity of about 100 mW/cm$^2$ obtained from a commercial 9 W EPROM eraser source, a time constant below 50 s is achieved. Measurements on the floating gate relaxation, in Fig. 1(b), illustrate the first-order linear dynamic behavior. Floating gate adaptation rates above 30 mV/s are obtained for a 2 V drive voltage on the electrode. The measured characteristics of the asymptote (2) and time constant, $\tau$, are given in Fig. 1(c).

## III. Circuit Architecture and Operation

In densely interconnected integrated circuits a dominant fraction of the area, of order $N^2$ for $N$ elements, is covered by the connections while auxiliary circuitry at the boundaries and interfaces, of order $N$, occupies relatively little space if organized properly, even if the size per individual element is significant. With this in mind, the compact design of the connection circuitry is of the highest priority, which is asserted by reducing the functions performed at the connection site to a strict minimum. The main cell here [12], implementing a connection between two elements, Fig. 2(a), consists of only two transistors, still enough to provide the basic multiplication and learning functions on the lowest level. By properly cascading cells with matching common input, output, and learning vector component lines into a two-dimensional arrangement, Fig. 2(b), the connection matrix is constructed. The compactness of the layout is evident from Fig. 2(c), showing an $8 \times 8$ array with a cell size of $30\,\mu$m $\times 30\,\mu$m fabricated in a $2\,\mu$m CMOS process. The interface circuitry at the boundaries of the array, which supplies the inputs, extracts and amplifies the outputs, and applies the learning vectors, is provided externally but will be integrated on chip in a future implementation. The complexity or architecture of this circuitry is not the issue here; rather we will address the function of the main cell and the array of cells.

### A. Matrix-Vector Multiplication

To obtain linear four-quadrant matrix–vector multiplication

$$O_i = \sum_j W_{ij} I_j \tag{3}$$

with inputs $I_j$, weights $W_{ij}$, and outputs $O_i$, we follow an approach found in [6] using MOS transistors biased above threshold operating in the triode region. By virtually grounding the $I_i^{\text{out}}$ output current line and suppling the input voltage $V_j^{\text{in}}$, the floating gate transistor $T_1$ in Fig. 2(a) injects a current

$$I_i^{\text{out}} = k\left(\left(V_{ij}^{\text{flg}} - V_{\text{th}}\right) V_j^{\text{in}} - {V_j^{\text{in}}}^2/2\right) \tag{4}$$

into that output line, with $V_{\text{th}}$ the transistor threshold voltage. Expression (4) holds for gate voltages well above the threshold, and indicates two-quadrant operation only. To allow both polarities for the weights, a reference current $I_{\text{ref}}^{\text{out}}$, obtained from an identical transistor with same drain voltage $V_j^{\text{in}}$ but with gate voltage $V_{\text{ref}\ j}^{\text{flg}}$, is subtracted from (4). This operation also cancels out the quadratic nonlinearity in (4), as it is common for both currents. External circuitry then converts the differential output currents $I_i^{\text{out}} - I_{\text{ref}}^{\text{out}}$ collected from the whole array into the output voltage vector $O_i$, effectively yielding the weights

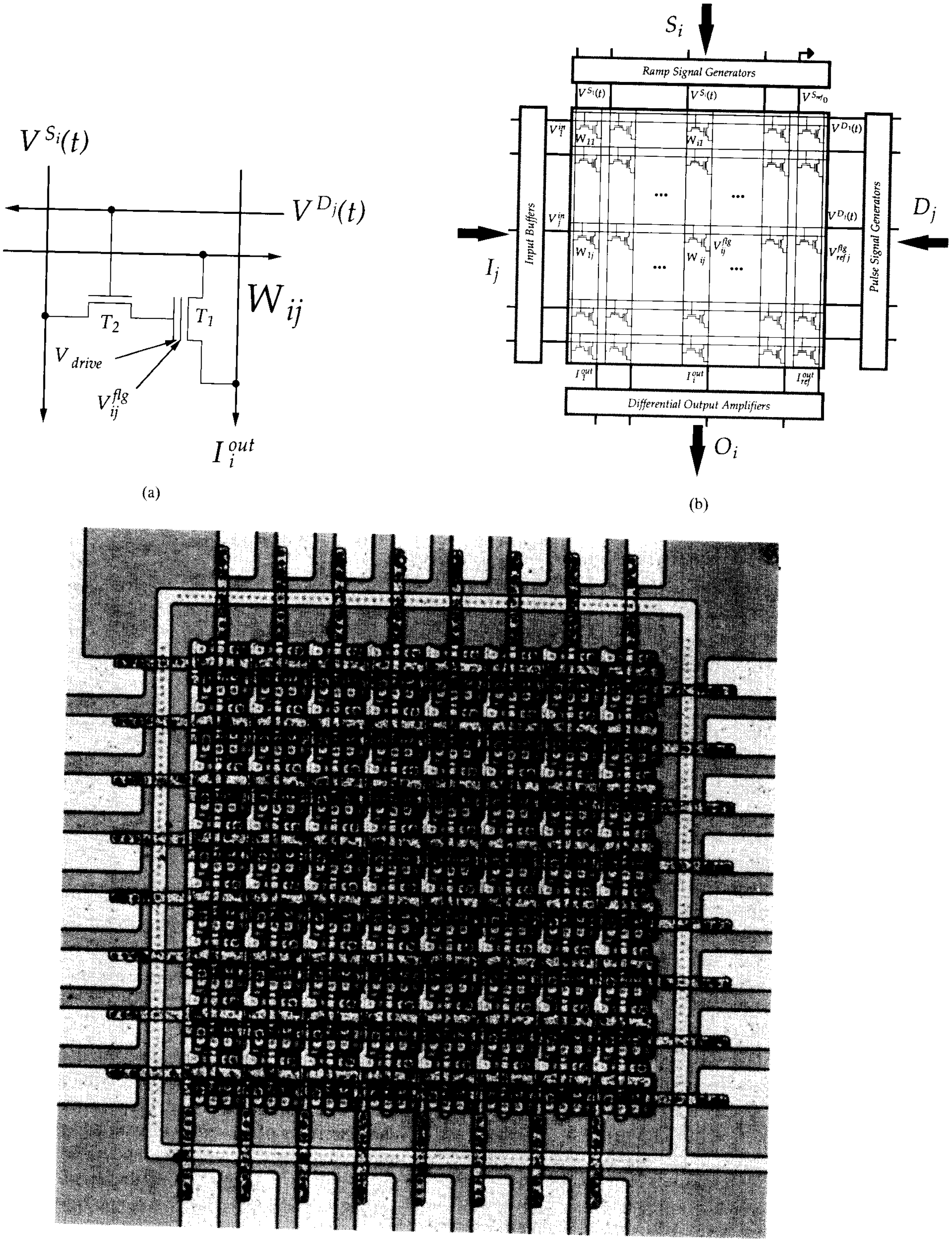
$$W_{ij} = R\ k\left(V_{ij}^{\text{flg}} - V_{\text{ref}\ j}^{\text{flg}}\right) \tag{5}$$

with $R$ the transresistance of the output amplifiers. Expression (5) assumes output amplifiers with input impedances sufficiently small to ensure proper virtual grounding of the output and reference current lines. Contrary to [6], the reference current $I_{\text{ref}}^{\text{out}}$ has only been generated once, common for all output currents, in a separate output row of the connection matrix to reduce the layout area. Thus, on average, each analog multiply accumulate operation occupies only one transistor in the array. Nevertheless, a linearity region for the multiplier ranging ±400 mV in both the drain input voltages and the differential floating gate weights with a total harmonic distortion (THD) below 2%, has been demonstrated at modest current levels in the $\pm 5\,\mu$A range [13]. In the present implementation, the transresistance output amplifiers are constructed externally with discrete components, limiting the time response currently to the ms range. We recently developed an integrated configuration, presently under investigation, with a time response below $5\,\mu$s and with an amplifier input impedance of $5\,\Omega$, supporting large multiplier arrays with several hundred inputs.

### B. Analog Incremental Outer-Product Learning

Analog four-quadrant incremental outer-product adaptation of the weights in the learning phase of the network is achieved by controlling the floating gate relaxation under UV illumination and loading the learning product onto the drive electrode. The product voltage is generated by a single transistor, $T_2$, using a dynamic sampling technique. Transistor $T_2$, connected to the drive electrode and the learning drive signal lines as shown in Fig. 2(a), samples the drive signal $V^{S_i}(t)$ onto the drive electrode each time the other signal, $V^{D_j}(t)$, goes high. To generate the learning product in this way a special encoding scheme, Fig. 3(a), is devised [12] to construct the drive signals according to the learning vector components. One component is encoded in the slope $S_i$ of $V^{S_i}(t)$, a periodic ramp voltage signal. The time displacement $D_j$ of a pulsed voltage signal $V^{D_j}(t)$ relative to $V^{S_i}(t)$ represents the other component. As such, for a reasonably small pulse duty cycle, the sampled voltage follows the desired four-quadrant learning product

$$V_{\text{drive}}{}^{a}{}_{ij} = V_{\text{drive}}{}^{a}{}_{0} + S_i\ D_j \tag{6}$$



(a)

(b)

(c)

Fig. 2. Circuit architecture: (a) main cell; (b) network; (c) chip micrograph.

with an offset $V_{\text{drive}}{}^{a}{}_{0}$ given by the ramp signal dc voltage component. The parasitic capacitance between the drive electrode and the substrate, inherent in the layout, is sufficiently large to reduce switching noise and clock feedthrough to acceptable levels, while small enough to avoid slowing down of the sampling dynamics in response to the narrow pulse width. To reduce the drift of the sampled voltage on the electrode between consecutive pulses, which is intensified

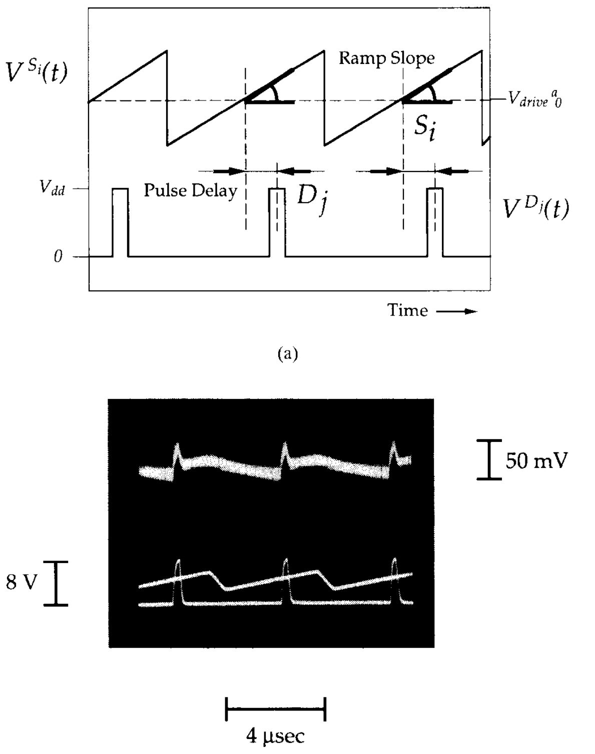


Fig. 3. (a) Learning drive signals. (b) Signal (lower trace) and drive electrode (upper trace) waveforms.

by the UV photocurrents generated in the $T_2$ reverse diode junctions, a fairly high pulse repetition rate, above 100 kHz, is needed. An oscillogram of a typical sampled drive electrode voltage waveform, with corresponding ramp and pulse signals, is shown in Fig. 3(b). In the computation phase of the network, the ramp and pulse drive signals are disabled, and the bias voltage $V_{\text{drive}}{}^{m}{}_{0}$, uniform for the entire array, is supplied to the drive electrodes of all cells. This is simply achieved with the same transistors $T_2$ by driving all pulse signal lines $V^{D_j}(t)$ steadily high while applying $V_{\text{drive}}{}^{m}{}_{0}$ to all ramp drive lines.

Combining the outer-product electrode drive voltages in (6) and the UV activated relaxation of the floating gates according to (1) and (2), the adaptation of the connection matrix satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}V^{\text{flg}}{}_{ij} = -\frac{1}{\tau}\left(V^{\text{flg}}{}_{ij} - V^{\text{flg}}{}_{\text{B}} - \lambda_{\text{drive}}{}^{a}\; S_i\; D_j\right). \qquad (7)$$

The term $V^{\text{flg}}{}_{\text{B}}$ defines a uniform bias for the gate voltages, set by the drive electrode bias voltages $V_{\text{drive}}{}^{m}{}_{0}$ and $V_{\text{drive}}{}^{a}{}_{0}$. This gate voltage bias is used to drive the transistors sufficiently above threshold for proper linear operation of the matrix–vector multiplier but has no effect on the implemented weights because of the differential output current configuration. With a zero-slope ramp signal $V^{S_{\text{ref}}}(t) \equiv V_{\text{drive}}{}^{a}{}_{0}$ applied on the reference output row, from (5) the weights obey

$$\frac{\mathrm{d}}{\mathrm{d}t}W_{ij} = -\frac{1}{\tau}\left(W_{ij} - R\;k\;\lambda_{\text{drive}}{}^{a}\; S_i\; D_j\right). \qquad (8)$$

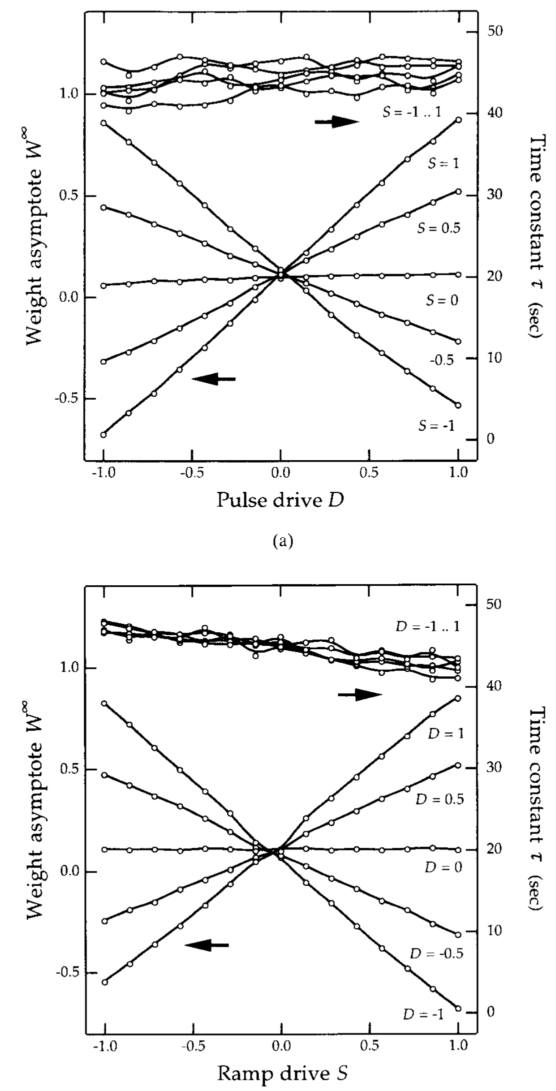


Fig. 4. Outer-product adaptation characteristics. The indicated pulse and ramp drives, $D$ and $S$, are relative to full scale (see text).

In addition to the desired outer-product weight adaptation, a uniform weight decay applies with time constant $\tau$ from the UV relaxation behavior. Parts (a) and (b) of Fig. 4 show characteristics of the time constant and the asymptotic weight under stationary pulse delay and ramp slope signal drives, measured on a single cell of the array. The indicated drive values $S_i$ and $D_i$ are relative to the full scales, 1.6 V/$\mu$s and 2 $\mu$s respectively.

## IV. Learning Scheme and Parameters

Whereas (8) describes the continuous evolution of the weights under a varying learning excitation $S_i(t) \times D_j(t)$, a learning session on the network only allows for discrete-time updates of the learning vectors $S_i$ and $D_j$. Indeed, as argued in Section II, the outputs of the network, required to determine the learning vectors, are unavailable under adaptation. Learning on the network is organized with an iterative scheme, alternating periodically between the computation and adaptation mode. During each iteration, the array adapts under a steady excitation $S_i$ and $D_j$ for a fixed time interval $\Delta t$ much smaller than the relaxation time constant $\tau$. Afterwards, during a short interrupt in the computation mode, a new input vector is supplied and the learning vectors $S_i$ and $D_j$ are updated for the next iteration according to the input vector and the settled output vector as specified by the learning rule. With this learning scheme, the discrete change in the weights in the $\Delta t$ interval between two consecutive iterations $(k)$ and $(k+1)$ can be expressed from (8) in the form

$$\Delta W_{ij}^{(k+1)} = W_{ij}^{(k+1)} - W_{ij}^{(k)} = -\alpha\, W_{ij}^{(k)} + \eta\, S_i^{(k)}\, D_j^{(k)} \quad (9)$$

with two learning parameters: $\alpha$, the uniform weight decay rate, and $\eta$, the incremental outer-product learning rate. Expressing the learning vector components $S_i$ and $D_j$ in dimensionless units relative to full scale $-1 \leq S_i, D_j \leq 1$, rather than in their physical units as slopes and time delays of the drive signals applied to the array, the obtained parameters read

$$\alpha = 1 - \exp(-\Delta t \,/\, \tau) \;\approx \Delta t \,/\, \tau \quad (10)$$

$$\eta = \alpha\; R\; k\; \Delta V_{\text{drive}}\; |\lambda_{\text{drive}}{}^{a}| \quad (11)$$

with $\Delta V_{\text{drive}}$ the nominal $(S_i \times D_j = 1)$ electrode drive voltage excitation. The absolute sign in (11) covers the case of a negative drive efficacy $\lambda_{\text{drive}}{}^{a}$, causing a negative learning rate $\eta$ in (9). Conceptually, in this situation a positive $\eta$ is enforced, simply by reversing the polarity of one of the two learning vectors, e.g. by inverting the slopes of the ramp signals.

Regardless of the particular learning task to be implemented on the physical system (9), the presence of the weight decay implies a limit on the effectiveness of the training samples applied sequentially to the network in the adaptation process. The weight decay favors learning excitations $S_i \times D_j$ that occurred within the last few $\tau$ time intervals; training samples presented before have relatively little effect. Though this may be useful, particularly for applications requiring a fast adaptive response under dynamically changing conditions, a sufficient number of training samples need to be provided to the network within one time constant $\tau$ interval, in order to project the underlying characteristics of the training data onto the network. This requires the condition $\tau \gg \Delta t$, met by adjusting either the adaptation time step $\Delta t$ or the UV illumination strength. In the $8 \times 7$ network used for the learning experiments, for a learning time step of 0.6 s, the time constant has been extended to 240 s by a fivefold reduction of the UV intensity.

From (10) and (11), the change in $\tau$ relative to $\Delta t$ affects both learning parameters $\eta$ and $\alpha$ to the same extent. In order to control the relative strength of these parameters, i.e., rescaling the weight decay and outer-product learning increments independently, the other device and circuit parameters in (11) need to be evoked. In practice, the transresistance $R$ of the output amplifiers and the voltage output range $\Delta V_{\text{drive}}$ of the ramp signal drivers are suitable for this purpose. Whereas some learning applications require a certain weight decay $\alpha < \eta$ to improve performance, as will be addressed in the next section, others may not. However, the limited adjustment range of the relative decay in the physical system excludes the case $\alpha \ll \eta$. To investigate the impact of a finite decay parameter, the critical value $\alpha \sim \eta$ has been established on the network for the learning experiments. With $R = 330$ kΩ and $\Delta V_{\text{drive}} = 1.5$ V, a weight decay rate $\alpha = 2.5 \times 10^{-3}$ and a learning rate $\eta = 7.0 \times 10^{-3}$ are obtained. Fig. 5(a) illustrates the incremental weight adaptation characteristics for a single connection, under steady learning components $S_i$ and $D_j$. To verify the dynamic response of the learning system under a fluctuating learning product $S_i \times D_j$, as in a realistic learning situation, the evolution of the weights $W_{ij}$ under a random excitation sequence $S_i$ and $D_j$ has been observed. From a linear spectral analysis of the recorded data, the impulse response of the learning system, that is, the relaxation of the weight for a unit impulse excitation $S_i \times D_j$, is derived in Fig. 5(b). As expected from (9), an exponential decay profile is observed, with a sharp transition at zero time, confirming the instantaneous response of the weight adaptation to fluctuations in the learning excitations.

## V. Learning Rule Implementations

Incremental outer-product learning rules for supervised and unsupervised learning tasks on linear and nonlinear networks have been developed and extentions on the original formula have been devised to improve specific digital computer implementations. For a two-layer (input–output) feedforward network configuration, an incremental weight update

$$\Delta W_{ij}^{(k+1)} = \eta\; f(O_i^{(k)},\; T_i^{(k)})\; I_j^{(k)} \quad (12)$$

is specified for each training iteration $(k)$ with $I_j$ the training sample inputs, $O_i$ the corresponding outputs of the network connected to the inputs by the weights $W_{ij}$, and, in the case of supervised learning, $T_i$ the target outputs associated with the inputs $I_j$. For a linear matrix–vector multiplier, the outputs simply relate to the inputs as given in (3), but the general form of (12) is still valid if nonlinear activation functions such as sigmoid or radial basis functions are applied to the weighted sums in (3) and if the two interconnected layers are surrounded by other layers embedded into a larger network [1]. As such, the conclusions drawn here for learning in linear networks apply to nonlinear multilayered networks too. The particular expression for $f(O_i,\, T_i)$ defines the learning rule. From (9), any rule that complies with the general expression (12) can be implemented on the analog hardware system by applying $S_i \equiv f(O_i,\, T_i)$ and $D_j \equiv I_j$ to the network, however including the additional weight decay term $-\alpha W_{ij}$. Specific implementations with uniform weight decay (9) are examined in further detail below for both cases of unsupervised and supervised learning.

### *A. Hebbian Rule Unsupervised Learning*

In the case of unsupervised learning where the network is trained freely without target $T_i$ to detect statistical patterns in the input vector, most commonly an incremental weight

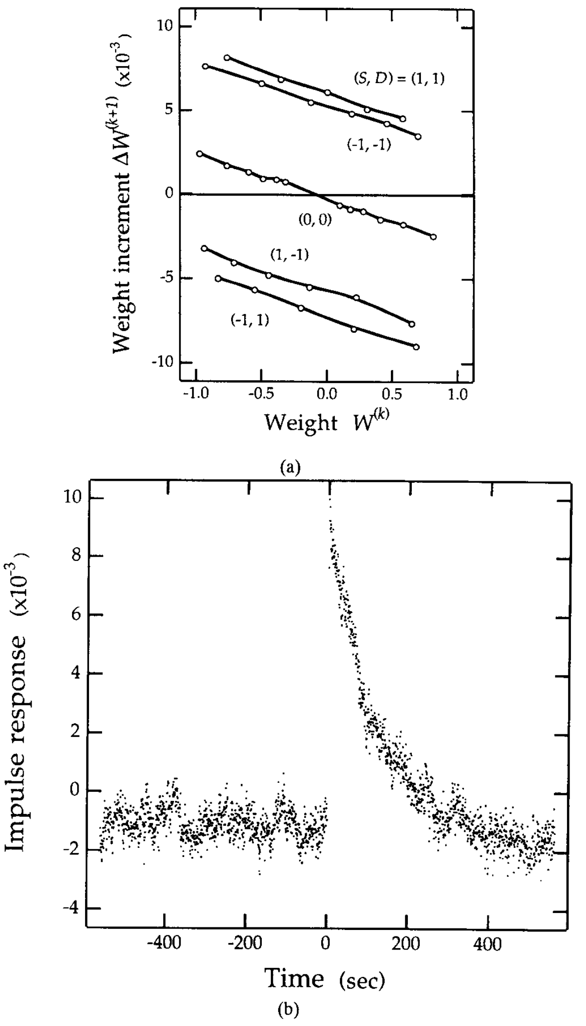


Fig. 5. Outer-product weight increments: (a) relaxation; (b) impulse response.

update rule of the Hebbian type, $f(O_i) \equiv O_i$, is used [2], [17]. Convergence issues necessitate some form of passivation and normalization or clipping of the weight patterns arising spontaneously from the Hebbian activation. A widely used method to incorporate passivation in the learning process consists in including a passive decay $-\alpha W_{ij}$ in the weight update rule [2]. From (9), this passive decay is naturally included in the physical learning system

$$\Delta W_{ij}^{(k+1)} = -\alpha\, W_{ij}^{(k)} + \eta\, O_i^{(k)}\, I_j^{(k)} \qquad (13)$$

for a straight implementation of the Hebb rule $f(O_i) \equiv O_i$. Still, a normalization or saturation mechanism is required to ensure confinement of the weights within the desired range. Instead of relying on the intrinsic saturation of the learning hardware, a functionally more elegant alternative involves dynamically renormalizing the weights along the learning process, by a simple rearrangement of the output circuitry. Removing the transimpedance amplifiers from the output stage of the network, (4) becomes

$$\sum_j k\Big((V_{ij}^{\rm flg} - V_{\rm th})\,(V_j^{\rm in} - V_i^{\rm out}) - (V_j^{\rm in} - V_i^{\rm out})^2/2\Big) = 0 \qquad (14)$$

with $V_i^{\rm out}$ the voltage on the $i$th output line, no longer virtually grounded. Regardless of the voltage range on the floating gates $V_{ij}^{\rm flg}$, the voltage on the output line $V_i^{\rm out}$ now follows a weighted average of the input voltages $V_j^{\rm in}$ depending on the relative strength of the triode conductances $k(V_{ij}^{\rm flg} - V_{\rm th})$. As in the previous output configuration, a differential processing of the outputs $O_i \propto V_i^{\rm out} - V_{\rm ref}^{\rm out}$ provides four-quadrant operation of the connections $W_{ij}$, though with this scheme the cancellation of the quadratic nonlinearities in (14) is only partial. With some algebraic manipulations, omitted here, it can be shown that (14) imposes the constraints $\sum_j W_{ij} = 0$ and $-1/N \leq W_{ij} \leq 1 - 1/N$ on the weights $W_{ij}$, with $N$ the number of input units. Hence the type of weight renormalization obtained this way has subtractive as well as divisive features. A formal analysis of the mentioned renormalization properties along with performance test results of the algorithm on the network will be given elsewhere.

### *B. Delta Rule Supervised Learning*

Most supervised learning algorithms training the network to map the input samples $I_j$ to the target output samples $O_i(I_j) \approx T_i$, such as the popular error back-propagation rule, are based on the delta rule [1], Widrow–Hoff rule [18] or the LMS algorithm [2], [3], [19] with $f(O_i,\, T_i) \equiv T_i - O_i$ for linear networks (3). The algorithm is proven to converge under a sequential and cyclical presentation of the training samples $(I_j^{(k)}, T_i^{(k)})$, yielding asymptotic weights close to optimum in a least square error sense [2], [19]. However, the inclusion of the weight decay in the update rule (9),

$$\Delta W_{ij}^{(k+1)} = -\alpha\, W_{ij}^{(k)} + \eta\, \left(T_i^{(k)} - O_i^{(k)}\right) I_j^{(k)} \qquad (15)$$

introduces a steady bias in the evolution of the weights, pulling their asymptotes toward the origin. To characterize the effect of the decay term in (15) on the learned weights under otherwise ideal learning conditions, we train the network with perfectly linear samples

$$T_i^{(k)} = \sum_j W_{ij}^T I_j^{(k)} \qquad (16)$$

selected from repeatedly generated random inputs $-1 \leq I_j^{(k)} \leq 1$ and from a fixed target weight matrix $W_{ij}^T$. For the analysis under the condition $\eta \ll 1$, the weights effectively adapt according to the statistical average rather than the instantaneous value of the stochastic fluctuations in the outer product. For zero-mean random inputs $\langle I_j^{(k)} \rangle = 0$ with variance $\langle I_p^{(k)} I_q^{(k)} \rangle = \sigma^2\, \delta_{pq}$, the weights on average change as

$$\langle \Delta W_{ij}^{(k+1)} \rangle = -\alpha\, W_{ij}^{(k)} + \eta\, \sigma^2\, \left(W_{ij}^T - W_{ij}^{(k)}\right). \qquad (17)$$

A first observation of (17) concerning the dynamics of convergence reveals a time constant $\left(\alpha + \eta\, \sigma^2\right)^{-1}$ characteristic

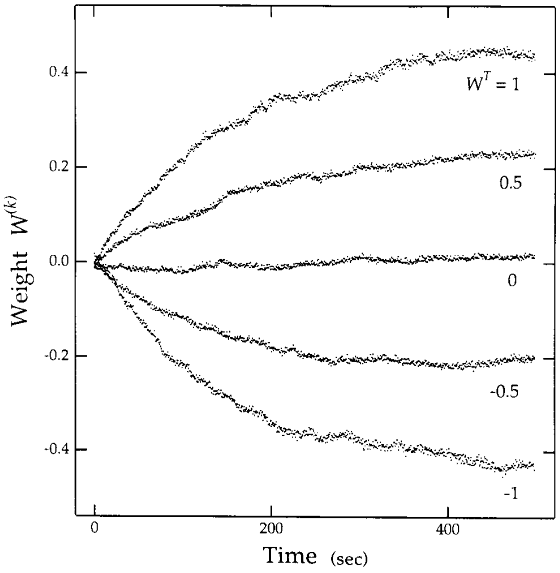


Fig. 6. Single-cell weight dynamics under delta rule training.

of the required learning time, considerably faster than the UV relaxation when $\eta\,\sigma^2 > \alpha$. More importantly, for $k \to \infty$, (17) implies a uniform scale reduction of the asymptotic weights $W_{ij}{}^{\infty}$ with respect to the targets $W_{ij}^T$,

$$W_{ij}{}^{\infty} = \lambda\, W_{ij}^T, \qquad \lambda = \left(1 + \alpha/\eta\sigma^2\right)^{-1}. \tag{18}$$

Only in the absence of weight decay, i.e., $\alpha = 0$, does the algorithm yield the target weights without a reduction in scale. However, the learning task is still effectively accomplished for any value of $\lambda$ under (18) since the uniform scaling of the weights consistently for all connections causes a scaling of the outputs (3) with the same reduction factor $\lambda$ for any input combination $I_j$. In general, depending on the relative strength of the decay $\alpha$ versus learning rate $\eta$ and the input variance $\sigma^2$ of the training samples, we have $0 \leq \lambda \leq 1$, but preferably $\lambda \geq 0.5$ for practical purposes. For the implementation on the network, $\alpha = 2.5 \times 10^{-3}$ and $\eta = 7.0 \times 10^{-3}$, and from a uniform random distribution of the input training samples $-1 \leq I_j^{(k)} \leq 1$, a medium value of $\lambda \approx \frac{1}{2}$ is obtained.

Fig. 6 shows the measured dynamics of weight adaptation under the delta rule (15) starting from zero initial conditions, obtained from a single connection in the array by supplying constantly generated linear samples (16) to the network for different target weights $W_{ij}^T$. The observed learning time constant, around 180 s, and weight asymptote scalar, $\lambda \approx 0.45$, agree with the expected values from the model (17). The high-frequency weight fluctuations in Fig. 6 are only partially due to stochastic contributions of the outer-product increments $\sim O(\eta\sigma^2)$; additive noise induced by the measurement process accounts for the discrepancy.

To verify the parallel learning capability of the learning system, the delta rule algorithm has been tested on the full network of Fig. 2. One output column of the $8 \times 8$ array has been reserved for the reference output $I_{\text{ref}}^{\text{out}}$ while the remaining connection cells support eight input units $I_j$ and seven output units $O_i$. External pulse and ramp signals generated with discrete components supply the learning vectors $D_j$ and $S_i$ to the network. Fig. 7 shows the measured response of the $8 \times 7$ network weight matrix starting from arbitrary initial conditions to the delta rule learning increments (15). The target weight matrix $W_{ij}^T$, used to determine the target outputs $T_i^{(k)}$ in conjunction with the random inputs $I_j^{(k)}$, is given in Table I. Because of the scaling of the weights $W_{ij}^{(k)}$ versus $W_{ij}^T$ and the outputs $O_i^{(k)}$ versus $T_i^{(k)}$, a traditional mean square error measure would be inappropriate for judging the learning performance. Instead, a figure of merit for the correspondence between the network and the target, accounting for a uniform scale factor, is given by

$$\Gamma_{O.T}^{(k)} = \sum_i O_i^{(k)} T_i^{(k)} \;/\; \Big(\sum_i O_i^{(k)2} \sum_i T_i^{(k)2}\Big)^{1/2} \tag{19}$$

for the outputs, and similarly

$$\Gamma_{W.W^T}^{(k)} = \sum_{i,j} W_{ij}^{(k)} W_{ij}^T \;/\; \Big(\sum_{i,j} W_{ij}^{(k)2} \sum_{i,j} W_{ij}^{T2}\Big)^{1/2} \tag{20}$$

for the weights. The correlation formulas (19) and (20) yield a number between $-1$ and 1, with values of 1 and $-1$ for perfect direct and opposite coherence, respectively, and 0 for no correspondence at all. The matrix correlation (20), involving a measurement of the weights, is observed every 50 samples, and the vector correlation (19) is recorded for every sample. Both are plotted in Fig. 7. A correspondence $\Gamma_{W.W^T}^{(k)}$ of 0.93 between adapted and target weights has been achieved after about 600 iterations on a time scale roughly comparable to the UV relaxation time constant $\tau = 240$ s. The weight matrices observed at convergence for two training sessions with different initial conditions are included in Table I. Similar training tests, with other arbitrary target matrices $W_{ij}^T$, have demonstrated proper convergence with correspondence figures $\Gamma_{W.W^T}^{\infty}$ all between 0.90 and 0.95. Whereas the offsets and variance induced by the learning circuitry combined with various noise sources certainly affect the performance, it is anticipated that the main limitation for further improvement is currently imposed by the nonlinearity of the matrix–vector multiplier (3). Weight variations $\Delta W_{ij}$ mount up as high as 4% of full scale over the input range, as indicated in Table I. The nonlinearity of the weights causes the persistance, after convergence, of fluctuations in the output-target vector correlation $\Gamma_{O.T}^{(k)}$, Fig. 7, under the random inputs $I_j^{(k)}$. These fluctuations illustrate the attempts of the learning system to map a linear target function onto a slightly nonlinear network for the entire range of the inputs. It is expected that for nonlinear learning tasks on large-scale multilayered networks with a sufficient number of hidden units, the problem of network inadequacy will be redressed.

## VI. Discussion and Conclusion

We have described and demonstrated the operation of a compact, fully parallel on-chip learning system in analog CMOS integrated technology for training large-scale matrix–vector

TABLE I
TARGET AND LEARNED WEIGHT MATRICES OF THE 8 × 7 NETWORK FOR TWO DELTA RULE TRAINING SESSIONS (INDICATED ERROR VALUES REFER TO NONLINEARITIES OF THE WEIGHTS)

| Weights $W_{ij}$ | | j = 1 | | j = 2 | | j = 3 | | j = 4 | | j = 5 | | j = 6 | | j = 7 | | j = 8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| i = 1 | **Target:** | **-0.031** | | **-0.198** | | **0.512** | | **0.211** | | **-0.170** | | **0.206** | | **-0.462** | | **0.311** | |
| | *Sess. A:* | *-0.02* | *± 0.00* | *-0.05* | *± 0.01* | *0.27* | *± 0.03* | *0.13* | *± 0.00* | *-0.07* | *± 0.00* | *0.05* | *± 0.01* | *-0.19* | *± 0.02* | *0.16* | *± 0.03* |
| | *Sess. B:* | *-0.02* | *± 0.02* | *-0.03* | *± 0.03* | *0.36* | *± 0.00* | *0.15* | *± 0.03* | *-0.07* | *± 0.04* | *0.07* | *± 0.02* | *-0.21* | *± 0.02* | *0.14* | *± 0.01* |
| i = 2 | **T:** | **0.139** | | **0.323** | | **0.183** | | **0.245** | | **0.222** | | **-0.357** | | **-0.508** | | **-0.271** | |
| | *A:* | *0.04* | *± 0.01* | *0.07* | *± 0.01* | *0.09* | *± 0.01* | *0.09* | *± 0.02* | *0.06* | *± 0.02* | *-0.10* | *± 0.01* | *-0.16* | *± 0.02* | *-0.06* | *± 0.02* |
| | *B:* | *0.07* | *± 0.02* | *0.07* | *± 0.02* | *0.08* | *± 0.02* | *0.11* | *± 0.03* | *0.06* | *± 0.04* | *-0.07* | *± 0.02* | *-0.17* | *± 0.03* | *-0.07* | *± 0.03* |
| i = 3 | **T:** | **0.087** | | **0.259** | | **-0.070** | | **-0.429** | | **0.186** | | **0.458** | | **0.045** | | **0.150** | |
| | *A:* | *0.01* | *± 0.03* | *0.07* | *± 0.01* | *-0.02* | *± 0.02* | *-0.15* | *± 0.03* | *-0.03* | *± 0.02* | *0.18* | *± 0.03* | *0.08* | *± 0.02* | *0.07* | *± 0.03* |
| | *B:* | *0.06* | *± 0.01* | *0.10* | *± 0.00* | *-0.03* | *± 0.00* | *-0.13* | *± 0.02* | *-0.05* | *± 0.01* | *0.21* | *± 0.01* | *0.09* | *± 0.00* | *0.12* | *± 0.01* |
| i = 4 | **T:** | **-0.333** | | **0.297** | | **-0.533** | | **-0.309** | | **0.382** | | **-0.350** | | **-0.111** | | **-0.044** | |
| | *A:* | *-0.14* | *± 0.01* | *0.10* | *± 0.00* | *-0.11* | *± 0.01* | *-0.07* | *± 0.00* | *0.15* | *± 0.01* | *-0.09* | *± 0.00* | *-0.05* | *± 0.01* | *0.02* | *± 0.00* |
| | *B:* | *-0.10* | *± 0.03* | *0.13* | *± 0.02* | *-0.12* | *± 0.01* | *-0.03* | *± 0.02* | *0.15* | *± 0.02* | *-0.08* | *± 0.02* | *-0.05* | *± 0.02* | *0.06* | *± 0.02* |
| i = 5 | **T:** | **-0.222** | | **-0.486** | | **-0.488** | | **-0.295** | | **0.161** | | **-0.350** | | **0.365** | | **-0.259** | |
| | *A:* | *-0.08* | *± 0.03* | *-0.21* | *± 0.02* | *-0.09* | *± 0.01* | *-0.04* | *± 0.03* | *0.07* | *± 0.02* | *-0.07* | *± 0.03* | *0.12* | *± 0.03* | *-0.08* | *± 0.02* |
| | *B:* | *-0.05* | *± 0.04* | *-0.17* | *± 0.03* | *-0.10* | *± 0.02* | *-0.02* | *± 0.03* | *0.08* | *± 0.03* | *-0.08* | *± 0.03* | *0.15* | *± 0.03* | *-0.06* | *± 0.03* |
| i = 6 | **T:** | **0.471** | | **0.338** | | **0.220** | | **0.469** | | **0.232** | | **-0.470** | | **-0.119** | | **-0.384** | |
| | *A:* | *0.17* | *± 0.00* | *0.09* | *± 0.00* | *0.11* | *± 0.00* | *0.17* | *± 0.01* | *0.09* | *± 0.01* | *-0.13* | *± 0.01* | *-0.01* | *± 0.00* | *-0.10* | *± 0.00* |
| | *B:* | *0.17* | *± 0.01* | *0.08* | *± 0.00* | *0.12* | *± 0.00* | *0.19* | *± 0.01* | *0.12* | *± 0.01* | *-0.11* | *± 0.01* | *-0.03* | *± 0.01* | *-0.10* | *± 0.00* |
| i = 7 | **T:** | **-0.555** | | **-0.212** | | **0.588** | | **0.427** | | **0.509** | | **0.077** | | **0.107** | | **0.177** | |
| | *A:* | *-0.19* | *± 0.00* | *-0.08* | *± 0.01* | *0.26* | *± 0.03* | *0.13* | *± 0.01* | *0.14* | *± 0.00* | *0.07* | *± 0.01* | *0.03* | *± 0.01* | *0.11* | *± 0.01* |
| | *B:* | *-0.17* | *± 0.02* | *-0.11* | *± 0.01* | *0.28* | *± 0.00* | *0.19* | *± 0.03* | *0.16* | *± 0.03* | *0.06* | *± 0.02* | *0.02* | *± 0.02* | *0.09* | *± 0.03* |

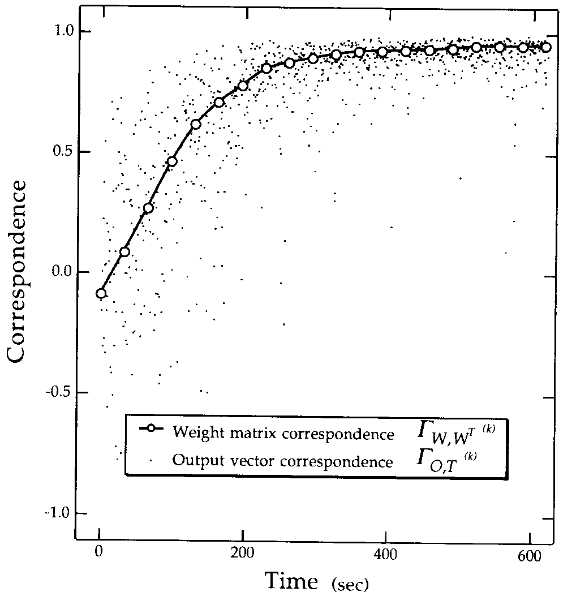


Fig. 7. Learning dynamics of the 8 × 7 network under the delta rule.

multiplier networks. Because of its parallel interface, the system supports learning applications on multilayered densely interconnected artificial neural networks using modules of matrix–vector multipliers to provide the connections between different layers of neurons. Rather than relying on sequential programming of the connections, the adaptation of the weight matrix is performed by parallel increments specified by an arbitrary outer-product iterative learning algorithm. Besides the enhancement in learning speed, the parallel on-chip learning configuration offers the additional advantage of a reduced sensitivity of the weights to process variations and offsets induced in the fabrication stage, granted the scale of the network provides enough degrees of freedom to adjust for these errors by the learning algorithm.

On the microelectronic level, each connection between two input and output units occupies a single cell in a two-dimensional arrangement. To reduce the area-intensive connectivity and adaptation functions performed locally in each cell, a higher complexity has been allowed for the interface circuitry at the boundaries of the connection array, buffering the inputs, amplifying the outputs and providing the signals for the generation of the outer-product increments. For densely interconnected networks, most of the area is covered by the array of connections where a simple design and compact layout of the connection cell leads to substantial savings in the total silicon area. Presently, with a cell size of 30 $\mu$m × 30 $\mu$m in 2 $\mu$m technology, the integration of a 256 × 256 array of adaptive connections on a 1 cm$^2$ die is feasible. Under such conditions, with an input voltage range of ±200 mV, proper biasing of the triode transistors $T_1$ easily permits a peak power dissipation below 1 $\mu$W per cell, thus never exceeding 100 mW for the entire array. Despite the favorable integration density and power consumption levels, the true parallel computation power of the network is currently restricted by the pin-out limitations on its inputs and outputs. For multilayered networks, however, the I/O bottleneck applies to the input and output layers only, not to the internal layers; thus the pin-out problem becomes less of a factor as the relative proportion of hidden units increases.

The weight increments in the learning process, induced under UV illumination, consist of two contributions: a uniform passive decay and an active outer-product increment. An adjustment of the UV intensity or the adaptation time step affects both terms to the same extent, hence controlling the speed of the learning. The relative proportions of both terms are adjustable as well, though bounded by the limits of the hardware. With properly adjusted learning parameters, learning time constants below 50 s are possible under a modest $100\ \mathrm{mW/cm^2}$ UV illumination level. While the active outer-product contribution to the weight increments directly relates to the learning task, as it is specified by the learning rule, the decay term may prove useful in passivating weight fluctuations and offering the flexibility of fast changes according to dynamically changing conditions. For supervised learning applications, however, a direct implementation of the delta rule in the presence of weight decay causes a uniform scale reduction in the outputs with respect to the targets. Results from a delta rule experiment conducted on the $8 \times 7$ network demonstrate the learning capability of the system, well within the accuracy limits imposed by the network nonidealities.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to R. Benson, B. Gupta, D. Kerns, C. Mead, A. Moore, V. Pedroni, and others for helpful discussions and/or critical reading of the manuscript. The CMOS chip was fabricated through the MOSIS service.

## REFERENCES

[1] D. E. Rumelhart and J. L. McCelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge MA: M.I.T. Press, 1986.

[2] T. Kohonen, *Self-Organisation and Associative Memory*. Berlin: Springer-Verlag, 1984.

[3] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[4] H. P. Graf and L. D. Jackel, "Analog electronic neural network circuits," *IEEE Circuits and Devices Mag.*, vol. 5, pp. 44–49, 1989.

[5] Y. Tsividis and S. Satyanarayana, "Analogue circuits for variable-synapse electronic neural networks," *Electron. Lett.*, vol. 23, pp. 1313–1314, 1987.

[6] F. J. Kub, K. K. Moon, I. A. Mack, and F. M. Long, "Programmable analog vector-matrix multipliers," *IEEE J. Solid-State Circuits*, vol. 25, pp. 207–214, 1990.

[7] D. B. Schwartz, R. E. Howard, and W. E. Hubbard, "A programmable analog neural network chip," *IEEE J. Solid-State Circuits*, vol. 24, pp. 313–319, 1989.

[8] B. Hochet, V. Peiris, S. Abdot, and M. J. Declercq, "Implementation of a learning Kohonen neuron based on a new multilevel storage technique," *IEEE J. Solid-State Circuits*, vol. 26, pp. 262–267, 1991.

[9] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 floating gate synapses," in *Proc. Int. Joint Conf. Neural Networks* (Washington DC), 1989, pp. 191–196.

[10] B. W. Lee, B. J. Sheu, and H. Yang, "Analog floating-gate synapses for general-purpose VLSI neural computation," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 654–658, 1991.

[11] E. Vittoz, H. Oguey, M. A. Maher, O. Nys, E. Dijkstra, and M. Chevroulet, "Analog storage of adjustable synaptic weights," in *VLSI Design of Neural Networks*. Norwell MA: Kluwer Academic, 1991, pp. 47–63.

[12] C. F. Neugebauer, A. Agranat, and A. Yariv, U.S. Patent Application filed, California Inst. Technology, Pasadena, CA, CIT-2062, 1990.

[13] G. Cauwenberghs, C. F. Neugebauer, and A. Yariv, "An adaptive CMOS matrix-vector multiplier for large scale analog hardware neural network applications," in *Proc. Int. Joint Conf. Neural Networks* (Seattle WA), vol. I, 1991, pp. 507–511.

[14] L. A. Glasser, "A U.V. write-enabled PROM," in *1985 Chapel-Hill Conf. VLSI*, 1985, pp. 61–65.

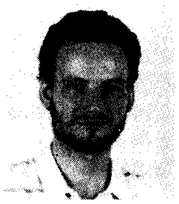
[15] C. A. Mead, "Adaptive retina," in *Analog VLSI Implementation of Neural Systems*. Norwell, MA: Kluwer Academic, 1989, pp. 239–246.

[16] D. A. Kerns, J. E. Tanner, M. A. Sivilotti, and J. Luo, "CMOS UV-writable non-volatile analog storage," in *Proc. Advanced Research in VLSI Int. Conf.* (Santa Cruz, CA), 1991.

[17] D. O. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.

[18] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *IRE WESCON Convention Record*, part 4, 1960, pp. 96–104.

[19] T. Kohonen, "An adaptive associate memory principle," *IEEE Trans. Comput.*, vol. C-23, pp. 444–445, 1974.



**Gert Cauwenberghs** (S'89) was born in Belgium on July 30, 1965. He received the engineer degree in applied physics from Vrije Universiteit Brussel, Belgium, in 1988 and the M.S. degree in electrical engineering from the California Institute of Technology, Pasadena, in 1989, where he is currently working toward the Ph.D. degree in electrical engineering. His research interests include analog VLSI for signal and information processing, hardware adaptation mechanisms, learning in artificial neural networks, and analog storage in VLSI.

**Charles F. Neugebauer** (S'90) received the bachelor's degree in electrical engineering and physics from the California Institute of Technology, Pasadena, where he is currently completing doctoral studies in the Department of Applied Physics.

Mr. Neugebauer is a member of Tau Beta Pi and Sigma Xi and holds two U.S. patents, with several applications pending. His research interests include analog VLSI neural systems and CCD signal processing. He is the recipient of a National Science Foundation Creativity in Engineering Award and an AT&T Doctoral Fellowship.

**Amnon Yariv** (S'56–M'59–F'70), a native of Israel, obtained the B.S. (1954), M.S. (1956), and Ph.D. (1958) degrees in electrical engineering from the University of California at Berkeley.

He joined the Bell Telephone Laboratories, Murray Hill, NJ, in 1959, during the early stages of the laser effort. He came to the California Institute of Technology, Pasadena, in 1964 as an Associate Professor of Electrical Engineering, becoming a Professor in 1966. In 1980 he became the Thomas G. Myers Professor of Electrical Engineering and Applied Physics. He took part, with various coworkers, in discovering a number of early solid-state laser systems, in proposing and demonstrating the field of semiconductor integrated optics, in inventing the semiconductor distributed feedback laser, and in pioneering the field of phase conjugate optics. His current research interests center on nonlinear optics, semiconductor lasers, and integrated optics. He is a founder and chairman of the board of the ORTEL Corporation and of the Accuwave Corporation.

Dr. Yariv has published widely in the laser and optics fields and has written a number of basic texts in quantum electronics, optics, and quantum mechanics. He is a member of the Americal Physical Society, Phi Beta Kappa, the American Academies of Arts and Sciences, and the National Academies of Engineering and Science and is a fellow of the Optical Society of America. He was the recipient of the 1980 Quantum Electronics Award of the IEEE, the 1985 University of Pennsylvania Pender Award, and the 1986 Optical Society of America Ives Medal.