Nanoelectronic Neuromorphic Networks (CrossNets): New Results

Özgür Türel — Ozgur.Turel@stonybrook.edu
Jung Hoon Lee — jlee@grad.physics.sunysb.edu
Xiaolong Ma — xiama@ic.sunysb.edu
Konstantin K. Likharev — klikharev@notes.cc.sunysb.edu

Stony Brook University, Stony Brook, NY, 11794-3800, U.S.A.

***Abstract* -- Our group is developing neuromorphic network architectures for future hybrid semiconductor/nanowire/ molecular ("CMOL") circuits. Estimates show that such networks ("CrossNets") may eventually overcome the cerebral cortex in areal density, operating at much higher speed, at acceptable power consumption. In this report, we demonstrate that CrossNets based on simple (two-terminal) molecular devices can be configured to reproduce the behavior of any known neural network, either feedforward or recurrent, using a synaptic weight import procedure. Two other training methods including the global reinforcement (that may enable CrossNets to perform more intelligent tasks) are also described in brief.**

I. Introduction

Recent spectacular advances in molecular electronics (see, e.g., Ref. 1) give every hope that the electronic industry will transfer relatively soon - in 10 to 20 years - from a purely semiconductor-transistor (CMOS) technology to hybrid ("CMOL") integrated circuits [2, 3]. Such circuits would combine a layer of advanced CMOS devices with two mutually perpendicular arrays of parallel nanowires (that may be formed, e.g., by nanoimprinting). Specially designed functional molecules would self-assemble, from solution, on each crossing of nanowires of the two layers. Estimates show [3] that CMOL circuits may feature unprecedented density (up to 10^{12} active devices per cm^2), at acceptable fabrication costs.

However, feasible molecular devices have limited functionality (e.g., low voltage gain) and can hardly ever be assembled with 100% yield. This is why the CMOL technology seems more suitable for the implementation of such defect-tolerant circuits as embedded and stand-alone memories and neuromorphic networks than Boolean logic. We have proposed [3-5] a family of neuromorphic circuits, called Distributed Crossbar Networks ("CrossNets"), whose topology is uniquely suitable for CMOL implementation – see Fig. 1. CrossNet synapses are based on simple molecular devices (latching switches, see Fig. 2 [4]), while neural cell bodies ("somas") may be implemented in the CMOS subsystem that physically would underlie the molecular device level. The advantages of the CrossNet architectures include arbitrary cell connectivity in quasi-2D CMOL circuits, and very high possible operation speed - intercell latency of the order of 10 ns - at acceptable power consumption below 100 W/cm^2 [3].

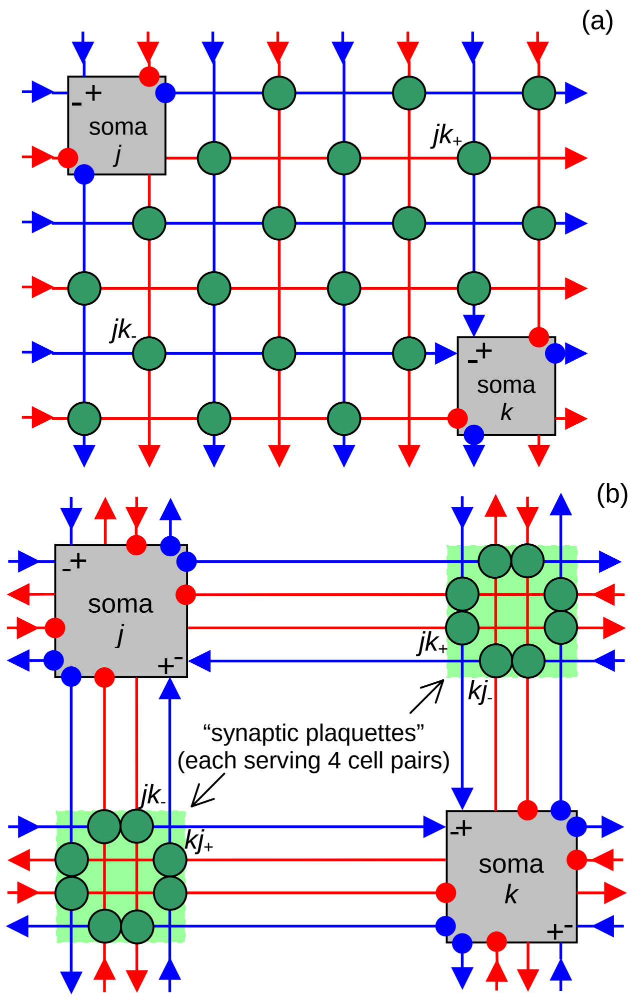

Fig. 1. Generic structure of (a) feedforward and (b) recurrent CrossNets [3-6]. Red lines show "axonic", and blue lines "dendritic" nanowires. Gray squares are interfaces between nanowires and CMOS-based cell bodies (somas). Signs show the dendrite input polarities. Green circles denote molecular latching switches forming synapses. (For clarity, panel (b) shows only the switches belonging to the same "synaptic plaquettes" as those connecting cells *j* and *k*.) Bold red and blue points are open-circuit terminations of the nanowires, that do not allow somas to interact in bypass of synapses. Besides recurrency, CrossNet species may differ by the way the somas are spread over the synaptic field, e.g., in diagonal rows ("FlossBar", see Fig. 8 below) or on a square lattice slightly inclined relatively the nanowire/synaptic lattice ("InBar", see Fig. 6 below).

However, the task of training CMOL CrossNets to perform functions typical for artificial neural networks faces several challenges:

- These networks use continuous (analog) signals, but the synaptic weights are binary, if only one switch for synapse is used. (In the CrossNets shown in Fig. 1, any pair of cells is connected by two switches each way, so that the net synaptic weight may take any of three values, say –1, 0, and +1.)

- The only way to reach for any particular switch in order to change its state is through the voltage applied between the two corresponding nanowires. Since each of these wires is also connected to many other switches, special caution is necessary to avoid undesirable "disturb" effects.

- Processes of turning single-electron latches on and off are statistical rather than dynamical, so that the applied voltage $V = V_a - V_d$ can only control probability rates $\Gamma_{\uparrow\downarrow}$ of these random events.

Fortunately, these rates are very strong functions of V, close to the simple Arrhenius law

$$\Gamma_{\uparrow\downarrow} = \Gamma_0 \exp\left\{ \pm \frac{e(V-S)}{k_B T} \right\}; \qquad (1)$$

here T is the effective temperature. (The shift parameter S depends on the switch design, and may be changed by applying voltage to a special global back-gate electrode.) Since voltage V may easily be made much higher than k_BT/e (e.g., ~500 mV vs. ~30 mV, respectively), the degree of randomness ("fuzziness") of switching may be restricted if necessary. For example, if Γ_0 is very low (so that $\Gamma_0 t \ll 1$, where t is the characteristic time of operation), Eq. (1) ensures that the latch turns on as soon as V exceeds the effective threshold voltage $V_+ = S + (k_BT/e)\ \ln(1/\Gamma_0 t)$, and turns off at $V < V_- = S - (k_BT/e)\ \ln(1/\Gamma_0 t)$.

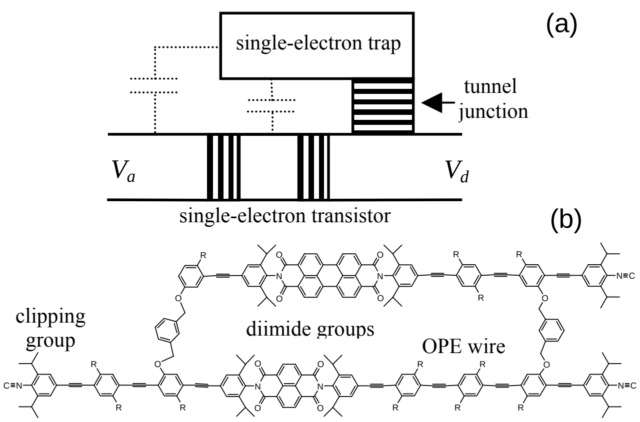

Fig. 2. (a) Schematics and (b) possible molecular implementation of the two-terminal single-electron latching switch. If voltage $V = V_a - V_d$ applied between the axonic and dendritic nanowires approaches certain threshold value $V_+ > 0$, an additional electron is inserted into the single-electron trap and lifts the Coulomb blockade of the single-electron transistor, thus connecting the nanowires. The switch state may be reset (e.g., wires disconnected) by applying a lower voltage $V < V_- < V_+$.

Until a few months ago, our work had been focused on CrossNets with *three-terminal* latching switches. In particular, we have shown [6] how CrossNets of a specific ("InBar") variety, based on such switches, can be used as Hopfield networks, e.g., for recognition of corrupted images.

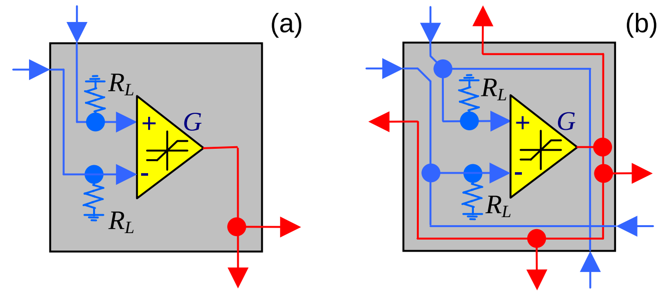

Fig. 3. Cell bodies (somas) of (a) feedforward and (b) recurrent CrossNets, within the simplest fire-rate model, in the operation mode (cf. Fig. 4 below). Low input resistances R_L are used to reduce V_d well below V_a, thus preventing the undesirable anti-Hebbian readjustment of synaptic weights.

The goal of this communication is to report new results that we believe make the whole CrossNet concept much more feasible. In particular, we have shown that the same functionality may be obtained with *two-terminal* switches (Fig. 2) which are much simpler for the chemically-directed self-assembly than their three-terminal counterparts.

One possible approach to settling these switches to form the desirable synaptic weights is described in Sec. II.A. In Sec. II.B we illustrate the application of this strategy to the Hopfield-mode operation of recurrent networks of the InBar variety. (In the same section we demonstrate that at least in this operation mode the CrossNets may be highly defect-tolerant.) The application of the same approach to feedforward CrossNets with FlossBar topology is discussed in Sec. II.C. The bottom line of Sec. II is that CrossNets may be taught to perform, on very high speed and for very large input vector size, any function that artificial neural networks have ever been used for.

In Sec. III we describe possible alternative techniques of CrossNet training, that avoid the use of external precursor networks. We speculate that one of these techniques ("global reinforcement") may also be used for more complex information processing tasks.

II. Synaptic Weight Import

A. General Approach

The first CrossNet teaching procedure that allows to overcome the problems listed in the Introduction, may be used if the necessary synaptic weights w_{jk} can be calculated outside of the CrossNet. In this case the CMOS-level wiring ("microwires") may be used to import the weight information, through the corresponding somas j and k, and then axonic and synaptic nanowires, to set latching switches to proper states.

For this operation, all nanowires are first deactivated and all latching switches are reset to their off state. This may be done, e.g., by raising shift S well above k_BT/e for a short time by applying a short pulse to the global back gate. For the sake of notation simplicity, we will assume that $S = 0$ during all the following operations. In particular, in this case $V_- = -V_+ = -V_t$, where $V_t \equiv (k_BT/e)\ln(1/\Gamma_0 t)$.

Now the somatic circuitry of selected cells is reconfigured as shown in Fig. 4 to apply "write enable" negative voltages with amplitude $V_0 \approx (2/3)V_t$ to their dendritic wires, and "data" voltages $V_a = \pm V_0$, with the sign corresponding to the desirable w_{jk}, to axonic wires. The "fully selected" latching switches, connecting the activated nanowires of opposite polarity, experience the net applied voltage $V = V_a - V_d \approx +(4/3)V_t$ (i.e. beyond the threshold V_t) and turn on, while the "half-selected" devices connected to only one of the wires retain the initial off state. (The latter is also true for the operation mode, in which the activation function of somatic amplifiers (Fig. 3) limits the axonic voltages at a level well below V_t.)

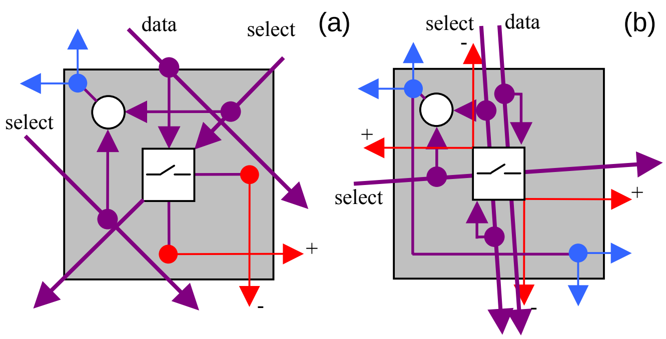

Fig. 4. Somas of (a) feedforward and (b) recurrent CrossNets in the weight import mode. Violet lines show CMOS-level "microwires" carrying select and data signals from the external tutor. Circles are threshold circuits that apply negative "write enable" voltages ($-V_0$) to dendritic nanowires, if both select signals are on. The key circuit lets the data signal be applied (with the polarities indicated on the picture) to axonic nanowires, if the select signal (carried by the microwire parallel to the data line) exceeds the nominal value by ~50%. More explanations are given in Sec. II.B and II.C below.

B. Recurrent Inbar as a Hopfield Network

Let us illustrate this training strategy on a simple example of a recurrent CrossNet working as a Hopfield network. Earlier we have already shown [6] that CrossBars with InBar topology may operate in this mode very effectively, if the synaptic weights follow the usual "clipped Hebbian rule" [7]

$$w_{jk} = \text{sgn} \sum_{p=1}^{P} \xi_j^{(p)} \xi_k^{(p)}, \qquad (2)$$

where $\xi_j^{(p)}$ is the j-th pixel on the p-th pattern. More specifically, the network capacity P_{max} at 99% fidelity is close to $0.5M$, where M is the connectivity parameter [5, 6]. (In recurrent CrossNets, each cell is directly connected to $4M$ other neurons.)

More recently, we have studied the defect tolerance of this operation mode, using both the (approximate) analytical theory and numerical modeling. For example, Fig. 5 shows results for a 3744-neuron InBar with $M = 25$. It is remarkable how resilient the network may be, if the number of stored patterns P is not too close to $P_{max} \sim 10$. For example, for P as high as 4 (i.e., $\sim P_{max}/2$), the network functions reasonably well (with 99% fidelity) even in the case when approximately 85% (!) of switches are bad.

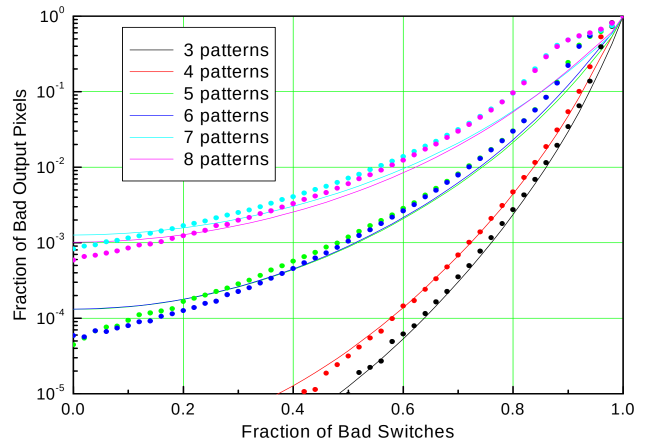

Fig. 5. Defect tolerance of the recurrent InBar with connectivity parameter $M = 25$, operating in the Hopfield mode. Lines show the results of an approximate analytical theory, while dots those of a numerical experiment.

Figure 6 shows the procedure of importing the synaptic weights (2) into the recurrent InBar (or rather just one of its $4MN^{1/2} \gg 1$ steps, where M is InBar's connectivity parameter and N is the total number of cells in InBar matrix).

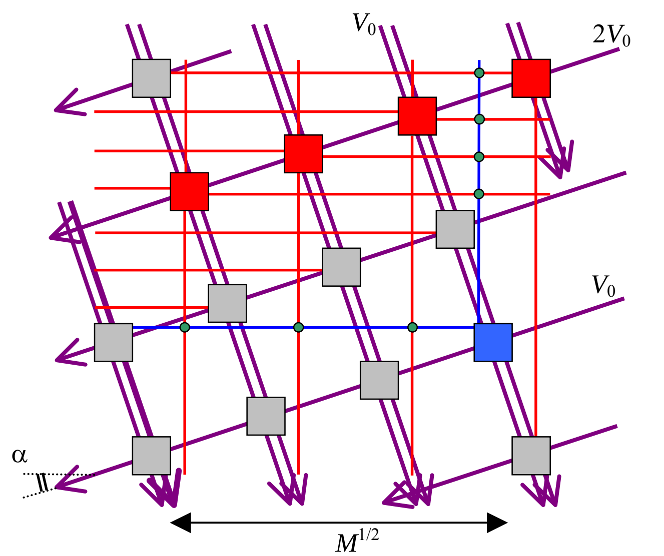

Fig. 6. Teaching the recurrent InBar to operate as a Hopfield network. The somas selected for axon activation are painted red, while those with activated dendrites as shown in blue. Only the non-zero values of the select signals are indicated. For the sake of clarity, the figure shows only the molecular devices being switched on at this particular weight import step, and only the nanowires activated at this step. The shown incline angle $\alpha = \arctan(M^{1/2})$ corresponds to a relatively small value $M = 9$ of the connectivity parameter; for important applications M should be much higher, i.e. α much smaller.

At each step, a specific set of "select" voltages applied to microwires allows to "activate" (i.e., apply voltages $\pm V_0$ to):

- axons of all cells of one quasi-horizontal row of the InBar (shown in red in Fig. 6), and

- two dendrites of each $2M^{1/2}$–th cell of another row, separated by vertical distance less or equal to $2M^{1/2}$ from the "axonic" row (Fig. 6 shows, in blue, just one of these cells).

The sign of axonic voltages is controlled by "data" microwires and follow the rule (2); if such voltage is positive, the net voltage V applied to the switch between the axonic selected wire and the selected dendrite exceeds V_t, turning the latch on. Returning to Fig. 2b we see that in the operation mode the selected cells are only becoming connected via one latch, so the corresponding synaptic weight w_{jk} = +1 as required by Eq. (2). In the case of the opposite sign of the data, only the device on the horizontal dendrite is turned on, providing for w_{jk} = -1 in the operation mode. Thus the CrossNet will operate in the Hopfield mode absolutely similar to the network based on three-terminal switches [6], with only ~30% loss of capacity in comparison with continuous-weight Hopfield networks [7].

C. Feedforward FlossBar as a Multilayered Perceptron

It is well known that practical application of Hopfield networks is rather limited. Many more applications (say, for pattern classification) have been developed for perceptrons with one or more hidden layers [7]. Some CrossNet species, e.g., feedforward FlossBars [5], are directly suitable for the use as perceptrons. Figure 7 shows how externally-calculated synaptic weights may be imported into this network. As shown in Fig. 4a, the necessary reconfiguration of the somatic cell may be even simpler than for the recurrent network.

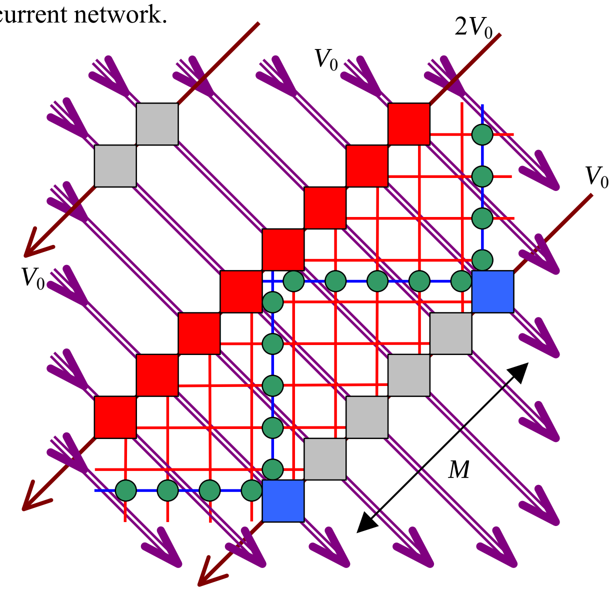

Fig. 7. Importing synaptic weights into the feedforward FlossBar. In this figure, the connectivity parameter M equals 4. For notation, see Fig. 6.

At each time step, the external "select" signals activate:

- both dendrites of each M-th cell of one row of somas, and

- both axons of each cell of the previous row.

As a result, importing all synaptic weights of one layer takes M steps.

Unfortunately, the information loss at synapse clipping may affect the performance of feedforward networks as pattern classifiers more seriously than the Hopfield networks. For example, Fig. 8 shows the results of our calculations of the average error of a simple perceptron, as well as perceptrons with one to three hidden layers, induced by synapse clipping, i.e. rounding of the initially continuous weight to the closest of L quantization levels. Figure 8b shows that the error depends on the effective gain g of the somatic cell. (At $g \approx g_t = 1/\sqrt{M}$ the system is close to linear, and the error rate does not depend much on the number of layers.) One can see that for two-latch synapses (L = 3) the error may be above 20%, unacceptable for most applications. At the same time, an increase of the number of levels L to, say, 33 makes the clipping-induced errors negligible (one to two percent).

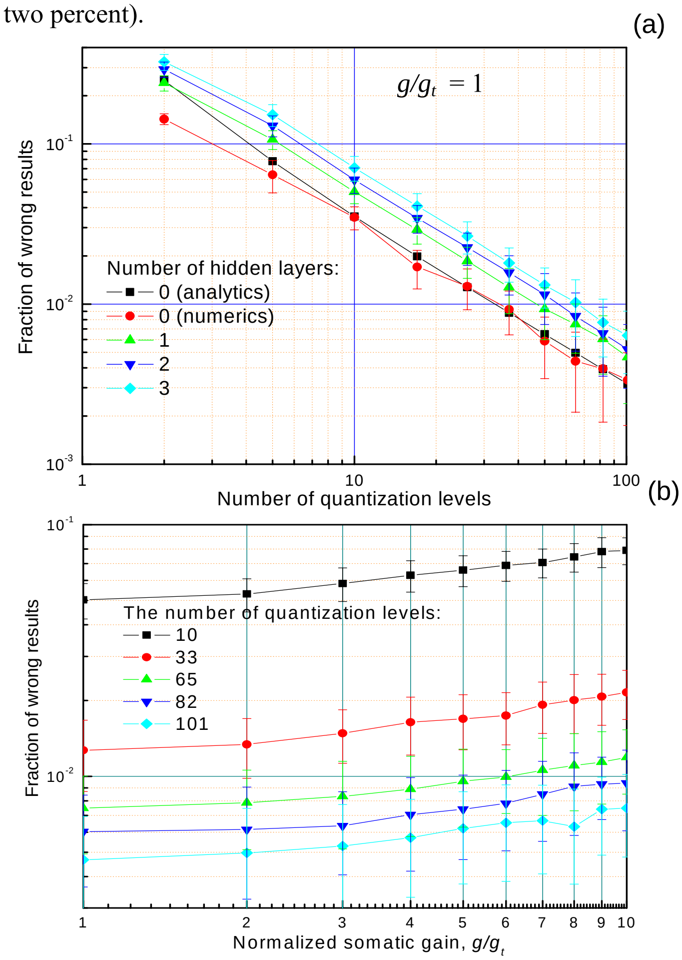

Fig. 8. Output error of few-layer perceptrons, with M = 100 neurons on each layer, induced by synaptic weight rounding to L discrete values, as a function of (a) number of quantization levels L, and (b) effective somatic cell gain $g \equiv GR_L/R$ for one hidden layer. (R is the resistance of the connected latching switch, while G is the linear voltage gain of the somatic amplifier.) Each numerical result was obtained by averaging over 100 random input vectors. Results of an approximate analytical theory for the simple perceptron are also shown; there results do not depend on g.

Such multi-valued synapses, with $L = 2n^2+1$, may be readily implemented by replacing each latching switch shown in Fig. 1 with a square array of $n \times n$ switches (Fig. 9). In the operation mode, all n axonic wires are fed with the same voltage, while the resulting currents flowing into n dendritic wires are just summed up on the load resistance R_L. As a result, the net output (post-synaptic) signal from two arrays is proportional to $w = (l_+ - l_-)/n^2$, where $l_\pm$ are the numbers of latches turned on in each array.

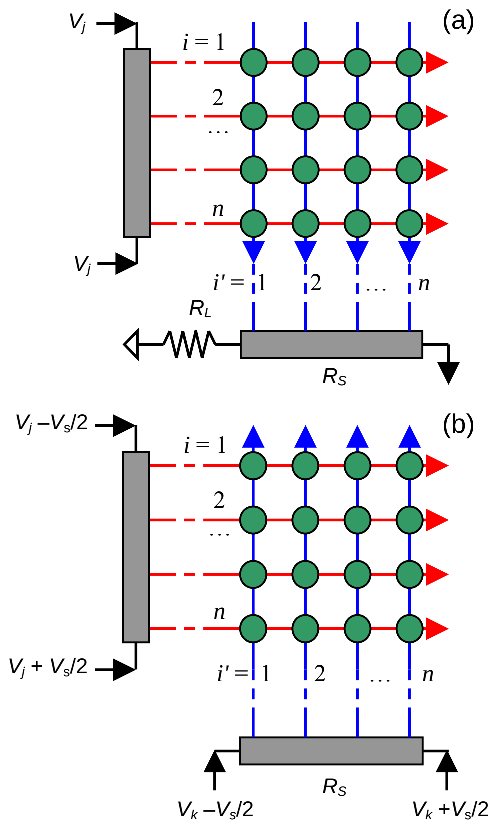

Fig. 9. A half of the synapse for providing $L = 2n^2+1$ discrete levels of the weight in (a) operation and (b) weight import modes. The dark-gray rectangles are resistive metallic strips (with the total resistance $R_S \ll R_L$) serving as soma/nanowire interfaces.

In order to fix the desirable value of $l_\pm$ in each array during the weight import mode, both vertical and horizontal wires are fed with graded voltages: $V_i = V_0 + V_s \times (i/n - ½)$, where $1 \leq i \leq n$ is the nanowire number. (Such voltage gradients may be readily generated, e.g., by simple resistive strips serving as contacts for axonic and dendritic nanowires – see Fig. 9.) This creates a gradient of the net voltage applied to switches, and hence a domain of switches being turned on. The boundary of this domain, defined by the equation $V \equiv V_a - V_d = V_t$, and hence the total number l of latches turned on, depends on both average axonic and dendritic V_0. (The former voltage, carrying information on the desired synaptic weight, is now continuous.) With close values of V_s for both dendrites and axons, the boundary is inclined by ~45° to the array edges, providing for a smoother $l(V_0)$ dependence. A simple analysis shows that the maximum possible value of the spread V_s equals $V_t/3$.

We have to acknowledge that the FlossBar structure is somewhat inconvenient for the CMOS subsystem design. InBar geometry is more convenient, because all CMOS somatic cells may have similar layout. In this context, we are currently exploring properties of feedforward InBars as pattern classifiers. An additional possible advantage of these networks may be higher fidelity at relatively low values of L, but high values of the connectivity parameter M.

III. Other Possible Training Methods

To summarize the previous section, CrossNets may perform, with very small loss of fidelity, the functions of Hopfield networks, multilayer perceptrons, and very probably any other fire-rate-model neural network (either feedforward or recurrent), provided that the synaptic weights have been calculated externally.

For some cases, for example the Hopfield network with clipped weights, such calculation does not present any problem – see Eq. (2). However, in some cases (e.g., multilayered perceptrons for nonlinear classification problems) the weight calculation may only be performed by training a precursor network with continuous weights, which is homomorphic to the CrossNet. Since we are speaking about very large networks, such training may take long time if performed on the usual sequential computers,. For some applications with limited input vector size (say, handwritten character recognition), such long training may be quite acceptable. However, in other cases (e.g., recognition of large-size patterns, such as detailed optical images) the precursor network training may require impracticable computer resources.

A. Error Backpropagation

A possible resolution of this problem is the direct training of CrossNets with multi-level (quasi-continuous) synapses (Fig. 9) by error backpropagation. We have developed a method for such training, that requires to double the number of nanowires and latching switches connecting each pair of somas (Fig. 10).

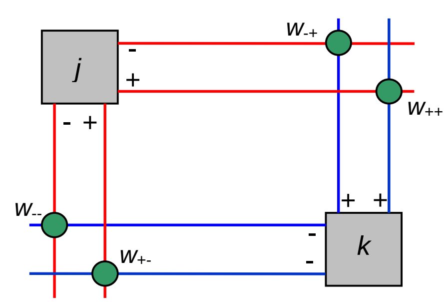

Fig. 10. Cell coupling in a feedforward CrossNet with the double number of nanowires and switches (cf. Fig. 1a). Each green circle denotes either a single molecular device or a multi-device array (Fig. 9). The net synaptic weight $w = w_{++} + w_{--} - w_{+-} - w_{-+}$.

The network training process is time-multiplexed into the following three steps that are repeated periodically:

(i) Somas are configured as shown in Fig. 3a, and the network provides the usual feedforward signal propagation.

(ii) Each somatic amplifier is reconfigured to provide linear amplification of the error signal with voltage gain proportional to $g'(U)$, where $V = g(U)$ is the activation function at the signal propagation stage, and U is the final value of the dendritic voltage at the first step. In contrast with the first stage, the amplifier is now fed with *axonic* voltage and applies its output signal to *dendritic* nanowires, so that the system performs the backpropagation of error signals ε (as usual, initiated by the tutor [7] at the system output).

(iii) The developed signals V and ε are applied, through the axonic and dendritic nanowires, respectively, to synapses with the same signs as shown in Fig. 10. (This requires temporary storage of V during the second step of the process; the storage may be achieved by using just one additional capacitor per soma.) Using Eq. (1) for switching rates (with $S = 0$), it is straightforward to show that if the fraction of latches turned on is not too large, the change of the average net synaptic weight obeys the equation

$$d\langle w_{jk} \rangle / dt = 4\Gamma_0 \sinh(eV_j / k_B T) \sinh(e\varepsilon_k / k_B T) \qquad (3)$$

that reduces to the canonical law of the backpropagation method [7] for relatively low values of V and ε. (The deviations of Eq. (3) from this law at larger signal values may actually be beneficial for the faster convergence of training iterations.)

For InBars with relatively large values of the connectivity parameter M this method may work for just four molecular switches per synapse. (We are currently checking this hypothesis.) However, for FlossBar perceptrons the "digital noise" created by the synaptic weight discreteness is too high (Fig. 8), and a latch array (Fig. 9) should be used for each of four sub-synapses shown in Fig. 10.

B. Global Reinforcement

A possible alternative way to train CrossBars without external precursor was suggested by our group earlier [5, 6] for binary, three-terminal devices. Upon slight modification, this method may also work for CrossNets with more realistic two-terminal devices.

The idea of this approach is based on the fact of chaotic excitation of recurrent CrossBars with asymmetric signs of dendritic signals (Figs. 1, 10), at sufficiently large effective gain of somatic cells ($g > g_t \approx M^{-1/2}$) [3-6]. One may say that in this regime the system walks randomly though the multi-dimensional phase space of all possible values of V_j. Now, let input signals be inserted into some of the cells, and outputs picked up from a smaller subset of cells. As soon as the system acquires, by chance, a state providing a satisfactory output signal vector, the external tutor commands all somatic cells to apply, for a brief time, the amplified values of input voltages U back to dendritic wires. The result is described by Eq. (3) with the replacement of ε_k for U_k, so that the pulse provides a Hebbian change of $\langle w_{jk} \rangle$, increasing the probability of the system's eventual return to this "good" point of the phase space. The extremely high potential speed of CrossBars make this method look much more promising for these circuits than for neural networks implemented on serial computers. However, at this stage it is still not clear whether the chaos-based procedure may be fast enough for efficient training.

Our plans are to perform extensive testing and quantitative characterization of both methods described in this section in near future.

IV. CONCLUSIONS

The main conclusion of this work is that the import of externally calculated synaptic weight values (Sec. II) may allow CrossNets to perform, at much higher speed, essentially any function ever implemented with an artificial neural network. Moreover, the high speed may allow to use global reinforcement (Sec. III) to train CrossNets to perform all these, and possibly more complex signal processing tasks.

There is a hope that in a more distant future the very high possible integration scale of CMOL CrossNets ($\sim 10^{11}$ neural cells, with $\sim 10^4$ synapses each, on a $\sim 30 \times 30$ cm^2 silicon wafer) will allow the development of hierarchical CrossNet-based systems [3] capable of performing tasks comparable with those typical for the mammal's cerebral cortex.

ACKNOWLEDGMENTS

The molecular design shown in Fig. 2b belongs to Prof. Andreas Mayr (SBU/Chemistry). Useful discussions with P. Adams, J. Barhen, S. M. Sherman, and V. Protopopescu are gratefully acknowledged. The work was supported in part by DOE and NSF.

REFERENCES

[1] J. R. Heath and M. A. Ratner, "Molecular electronics", Phys. Today, vol. 57, pp. 43-49, May 2003.

[2] K. Likharev, "Electronics below 10 nm", in *Nano and Giga Challenges in Microelectronics*, Amsterdam: Elsevier, 2003, pp. 27-68.

[3] K. Likharev, A. Mayr, I. Muckra, and Ö. Türel, "CrossNets: High-performance neuromorphic architectures for CMOL circuits", in *Molecular Electronics III* (*Annals of New York Acad. Sci*., vol. 1006, 2003), pp. 146-163.

[4] S. Fölling, Ö. Türel and K. K. Likharev, "Single-electron latching switches as nanoscale synapses", in *Proceedings of the 2001 International Joint Conference on Neural Networks*, Washington, DC, pp. 216-221.

[5] Ö. Türel and K. Likharev, "CrossNets: Possible neuromorphic networks based on nanoscale components", *Int. J. of Circuit Theory and Applications*, vol. 31, pp. 37-53, Jan. 2003.

[6] Ö. Türel, I. Muckra and K. Likharev, "Possible nanoelectronic implementation of neuromorphic networks", in *Proceedings of the 2003 International Joint Conference on Neural Networks,* Portland, OR, pp. 365-370.

[7] See, e.g., J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Cambridge, MA: Perseus, 1991.