Real-Time Classification of Complex Patterns Using Spike-Based Learning in Neuromorphic VLSI

Srinjoy Mitra, *Member, IEEE*, Stefano Fusi, and Giacomo Indiveri, *Senior Member, IEEE*

***Abstract*—Real-time classification of patterns of spike trains is a difficult computational problem that both natural and artificial networks of spiking neurons are confronted with. The solution to this problem not only could contribute to understanding the fundamental mechanisms of computation used in the biological brain, but could also lead to efficient hardware implementations of a wide range of applications ranging from autonomous sensory-motor systems to brain-machine interfaces. Here we demonstrate real-time classification of complex patterns of mean firing rates, using a VLSI network of spiking neurons and dynamic synapses which implement a robust spike-driven plasticity mechanism. The learning rule implemented is a supervised one: a teacher signal provides the output neuron with an extra input spike-train during training, in parallel to the spike-trains that represent the input pattern. The teacher signal simply indicates if the neuron should respond to the input pattern with a high rate or with a low one. The learning mechanism modifies the synaptic weights only as long as the current generated by all the stimulated plastic synapses does not match the output desired by the teacher, as in the perceptron learning rule. We describe the implementation of this learning mechanism and present experimental data that demonstrate how the VLSI neural network can learn to classify patterns of neural activities, also in the case in which they are highly correlated.**

***Index Terms*—Classification, learning, neuromorphic VLSI, silicon neuron, silicon synapse, spike-based plasticity, synaptic dynamics.**

I. Introduction

THE process of classification of patterns performed by neural networks is usually the result of a training procedure, during which the synaptic strengths between neurons are modified according to a learning prescription. During training the network has to create, modify and preserve memories of the representations of the learnt classes. In particular the memory elements (the synaptic weights), should be modified to learn from experience and create new memories (*memory encoding*), and at the same time the old memories should be protected from being overwritten by new ones (*memory preservation*).

Here we describe a VLSI system based on a network of integrate-and-fire (I&F) neurons and plastic synapses, which can learn to classify complex patterns, efficiently solving both the memory encoding and the memory preservation problems. Within this context, the term complex patterns refers to sets of spike trains with "high" or "low" mean firing rates, or with graded mean firing rates (i.e., which can assume more than just two values), and that can be correlated or uncorrelated in space. Specifically, we demonstrate how the system proposed can robustly classify patterns of mean firing rates, also in the case in which there are strong correlations between them.

The VLSI circuits operate in real-time, and have biologically plausible time constants (e.g., of the order of tens of milliseconds), so that they are well matched to sensory signals and are inherently synchronized with the real world events. Therefore, our hardware implementation could be efficiently utilized for a wide range of sensory-motor applications such as on-line learning in autonomous robotics, or as a real-time spike-based computational module in brain-machine interfaces.

A. Memory Encoding

In a network of spiking neurons, synaptic modifications are driven by the spikes emitted by the pre-synaptic neuron and by the activity of the post-synaptic cell. A natural prescription for modifying the synapses could be based on the relative timing of the pre- and post-synaptic spikes. Spike timing dependent plasticity (STDP) is one possible mechanism motivated by experimental [1] and by theoretical studies [2], [3]. These mechanisms have important regulatory properties [4] and they can create memories of temporal patterns of spikes (*e.g.,* see [5], [6]). However, STDP in its simplest form is not suitable for learning patterns of mean firing rates [4], as it is usually too sensitive to the specific temporal pattern of spikes and it can hardly generalize to trains of spikes sharing the same mean firing rate. Recently a new model of stochastic spike-driven synaptic plasticity has been proposed [7] that can encode patterns of mean firing rates, and captures the rich phenomenology observed in neurophysiological experiments on synaptic plasticity, including STDP protocols. In this model synaptic weights are modified upon the arrival of pre-synaptic spikes, according to the state of the post-synaptic neuron's membrane potential, and according to its recent spiking activity. If the pre-synaptic spike occurs when the membrane potential is close to its spiking threshold, the synaptic weight is potentiated. If the input spike occurs when the membrane potential is close to its resting potential, then the synaptic weight is decreased. All synaptic weight

Manuscript received March 12, 2008; revised July 02, 2008. First published December 02, 2008; current version published November 28, 2008. This work was supported in part by the Swiss National Science Foundation under Grant PP00A1 06556, in part by the ETH G. TH0201 7404, and the EU grants ALAVLSI (IST-2001- 3 8099), and DAISY (FP6-2005-015803). This paper was recommended by R. Etienne-Cummings.

S. Mitra and G. Indiveri are with the Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich 8057, Switzerland (e-mail: giacomo@ethz.ch).

S. Fusi is with Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich 8057, Switzerland, and also with the Center for Theoretical Neuroscience, Columbia University College of Physicians and Surgeons, New York City, NY 10027 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBCAS.2008.2005781

updates however are blocked if the post-synaptic neuron's recent firing activity is very high or very low. It has been shown that networks of spiking neurons that use this synaptic plasticity model can learn to classify correlated patterns of spike trains ranging from stimuli generated by auditory/vision sensors to images of handwritten digits from the reduced MNIST database [7].

B. Memory Preservation

Immediately after a memory is created, it is most vivid and it is possible to retrieve it if enough synapses have been modified in the proper way. When new memories are stored old memories can be overwritten, eventually making memory retrieval impossible. The forgetting process for physical synapses is typically exponentially fast [8]–[10]. Such a decay of the mnemonic trace is a consequence of the fact that physical synapses are restricted to vary in a limited range and they cannot be modified with an arbitrarily large precision. As a consequence the synapses tend to reach an equilibrium distribution very rapidly, at a rate which depends on the extent by which the synapses are modified. A large number of synaptic modifications implies fast learning, but also fast forgetting. By slowing down the learning process (i.e., by modifying a reduced number of synapses), the memory lifetime can be greatly increased [8], [9]. Extending the range in which the synapses vary, or the number of discrete states that they have to traverse to go from the lower to the upper bound, does not improve the memory performance considerably [10]. These two last considerations induced us to build synapses which are bistable, i.e., they have the minimal number of stable states, and that make transitions between one state to the other stochastically, with small probability, to slow down the learning process. Moreover we further extended the memory lifetime by modifying the synapses only when necessary, i.e., when the input pattern weighted by the plastic synapses would not generate the output desired by the supervisor. The probabilistic nature of the plasticity mechanism proposed is obtained automatically, as a direct consequence of using spiking neurons with irregular activity. Indeed, if the trains of spikes emitted by the pre-synaptic neurons have a Poisson distribution, and the transition between stable states occur only after a certain number of spike-driven events accumulate, then the changes in the synaptic weight are stochastic [11].

C. Hardware Implementation

We implemented a VLSI network of spiking neurons with plastic bistable synapses using full custom hybrid analog/digital neuromorphic circuits. The VLSI device receives its input spikes and transmits its output spikes to the outside world using asynchronous digital circuits. Each time a spike is emitted by a neuron, its address is encoded as a binary word representing an address-event, and written on a digital-bus. In this address-event representation (AER) input and output signals are real–time digital events that carry analog information in their temporal relationships (inter–spike intervals). Recently an increasing collection of spike-based computing chips have been developed within the AER framework [12]–[14]. This device, thanks to its spike-based plasticity mechanisms, can be used in distributed multichip AER systems as a general purpose learning and classification module. VLSI implementations of spike-based learning systems have been previously proposed [15]–[21], but they either lack the combined memory encoding and memory preservation features of the spike-based plasticity mechanism implemented in this device, or cannot cope with highly correlated patterns as efficiently as the system described here. A recent alternative VLSI implementation of the same plasticity mechanism described in this paper has been proposed in [22]. However, the circuits in that device are significantly larger than the ones used here, as they comprise additional digital modules for configuring the synaptic matrix inside the chip. In addition the synapses used in [22] lack the biologically realistic temporal dynamics present on the implementation proposed here.

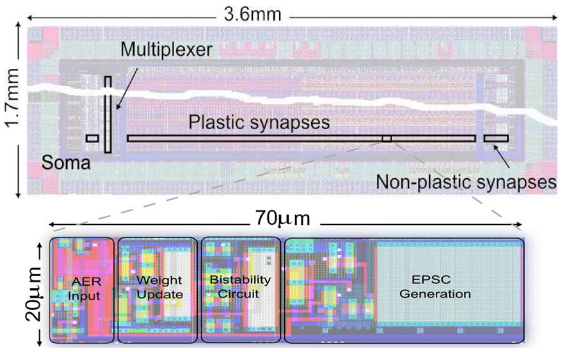


Fig. 1. Layout of a prototype chip comprising an array of 16 I&F neurons connected to rows of 128 AER synapses (120 with learning capabilities, and 8 with fixed excitatory or inhibitory settings). An on-chip multiplexer allows the user to select how many rows of synapses/neuron to configure. A zoomed version of the plastic synapse, with all its functional blocks is shown in the bottom part of the figure.

II. Implementation of the Spike-Based Plasticity Mechanism

The VLSI chip used to carry out the classification experiments is shown in the top part of Fig. 1. It is a prototype chip with 16 low-power integrate-and-fire (I&F) neurons [16] and 2048 dynamic synapses (128 per neuron). It was fabricated using a standard 0.35 μm CMOS technology, and occupies an area of 6.1 mm^2. Given its parallel architecture, processing time does not increase with size, and larger networks can be fabricated by simply using more silicon real-estate. With network sizes of this order, the AER bandwidth required to operate these types of systems is not a bottleneck. For example, in worst-case scenarios (with all input and output neurons firing at a maximum firing rate of 200 Hz) this device would receive approximately 400 k address-events per second, and transmit 3.125 k events/s. The total AER bandwidth required would therefore be less than 10% of the maximum AER bandwidth available, with current AER interfacing boards (5 M events/s) [23]. In typical operating conditions, the bandwidth used is actually less than 0.2% of the maximum available.

For each output neuron on the VLSI chip, we divided the synapse circuits into four different functional blocks: 4 excitatory synapses with fixed (externally adjustable) weights, 4 inhibitory synapses, and two sets of 60 synapses with local

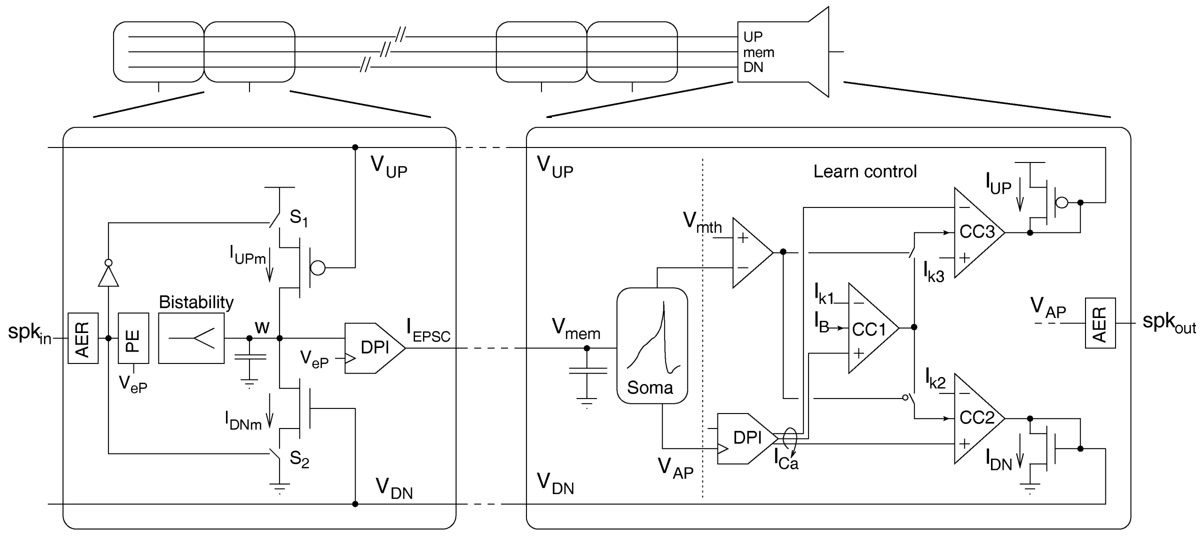


Fig. 2. Schematic diagram of a neuron and its synapses. The synaptic weight node w can be modified only with the occurrence of a pre-synaptic input spike ($spk_{\rm in}$). This turns on switches $S1$, $S2$ and updates w depending on the values of $V_{\rm UP}$ and $V_{\rm DN}$. In parallel, the bistability circuit slowly drives the node w toward either of its two stable states. The DPI circuit produces an excitatory post-synaptic current (I_{EPSC}) at each pre-synaptic spike, with an amplitude that depends on the synaptic weight w. The neuron comprises a low-power I&F soma, an integrator, a voltage comparator and three current comparators(CC). Winner-take-all (WTA) circuits are used as CCs that set the output to be either equal to the bias current I_B, or zero. The voltage comparator enables either $CC2$ or $CC3$, depending on the value of $V_{\rm mem}$ with respect to $V_{\rm mth}$. Another instance of a DPI is used to integrate the soma's action potentials V_{AP}, and its output current I_{Ca} is copied to the three CCs. The signal V_{AP} is also used to generate digital output spikes ($spk_{\rm out}$). The digital input ($spk_{\rm in}$) and output spikes ($spk_{\rm out}$) are both produced by *AER* interfacing circuits.

learning circuits (excitatory plastic synapses). The layout of a plastic excitatory synapse is shown in the bottom part of Fig. 1.

In our experiments we used each individual silicon neuron as a binary classifier, that separates input patterns into two categories. During training, the patterns to be classified are presented to the plastic synapses, in parallel with the teacher signal that represents the desired (high/low) firing rate. The post-synaptic neuron responds with an activity that is proportional to its net input current, generated by the input pattern weighted by the learned synaptic efficacies, and by the teacher signal. Synapses are updated at each pre-synaptic spike by an upward or a downward jump depending on the state of the neuron's membrane potential. The jumps accumulate and eventually lead to a transition in the synaptic weight state. The transitions result in long term changes in synaptic weights, and indicate that the input patterns have been learned. If, during training the neuron's response becomes aligned to the teacher signal (i.e., it is either a high or a low mean firing rate), then the synaptic jumps are blocked. This indicates that enough synapses have learned their correct states and recruitment of further synapses is not necessary for that particular pattern.

III. VLSI Learning Circuits

The learning circuits are responsible for locally updating the synaptic weights to learn to classify input patterns. When an address-event is received by a specific plastic synapse, its learning circuits update the synaptic weight w according to a spike-driven learning rule recently proposed in [7]. The synapse then produces an excitatory post-synaptic current (EPSC) with an amplitude proportional to its weight, and an exponential profile over time which can be set to last from microseconds to several hundreds of milliseconds [24]. The currents of all synapses afferent to the target neuron are summed into the neuron's membrane capacitance node. Eventually the I&F neuron's membrane potential exceeds a threshold and the soma circuit generates an output spike. As prescribed by the theoretical model [7], the magnitude of weight change Δw is determined by the state of the post-synaptic neuron's membrane potential and by the neuron's mean firing rate, at the arrival of the input pre-synaptic spike. The Δw weight updates are performed locally at each synapse, by the *pre-synaptic weight update modules*, depending on two global eligibility traces computed by the neuron's *post-synaptic weight control module*. In Fig. 2 we show a schematic diagram of a neuron with its synapses, and the circuit details of both modules.

A. Pre-Synaptic Weight-Update Module

This module, shown in the bottom-left part of Fig. 2, comprises five circuit blocks: an AER interface circuit, a Pulse Extender (PE), a local weight update circuit, a bistability weight refresh circuit, and a diff-pair integrator (DPI) synapse [24]. The AER circuit acts as an interface that receives the address-event signals ($spk_{\rm in}$) and activates the switches (MOSFETs) labeled $S1$ and $S2$. The weight update circuit is composed of the two switches $S1$ and $S2$ in series to an n-FET and a p-FET respectively. This circuit increases or decreases the node voltage w by sourcing or sinking charge from the capacitor connected to that node. The polarity of weight change depends on the value of the eligibility traces $V_{\rm UP}$ and $V_{\rm DN}$, broadcast from the post-synaptic weight control module. The bistability circuit is a low slew-rate amplifier in positive feedback configuration. Its negative input is connected to the weight w and its positive input is connected to the threshold bias θ. The amplifier supply rails are set to the voltages that correspond to the two synaptic weights stable states, defined as V_H and V_L for the high and low state respectively. Its bias current, a weak subthreshold current I_r , determines the positive and negative slew rates. The bistability circuit drives the weight toward one of the two stable states obeying the following equation:

$$\frac{dw}{dt} = \begin{cases} +\frac{I_r}{C_w} & \text{if } w > \theta \text{ and } w < V_H \\ -\frac{I_r}{C_w} & \text{if } w < \theta \text{ and } w > V_L \end{cases}. \tag{1}$$

In parallel to this slow drive toward one of the two stable states, every pre-synaptic spike spk_{in} induces one instantaneous jump on w according to the following rule:

$$\Delta w = (I_{\text{UP}m} - I_{\text{DN}m})\frac{\Delta t}{C_w}, \tag{2}$$

where C_w is the capacitance connected to node w, Δt the duration of the input pulse spk_{in}, and $I_{\text{UP}m}$ and $I_{\text{DN}m}$ are copies of I_{UP} and I_{DN} in the post-synaptic weight control module, derived from the global eligibility traces V_{UP} and V_{DN}. In addition to updating the synaptic weight, the input pulse spk_{in} is extended by the PE circuit (V_{eP}) and fed into the synapse DPI module. With every pre-synaptic spike the DPI produces an EPSC (I_{EPSC}) with an amplitude that is proportional to the synaptic weight w, and with exponential temporal dynamics [24].

B. Post-Synaptic Weight Control Module

As well as implementing the leaky I&F soma model, the post-synaptic weight control module in the bottom right part of Fig. 2 produces two global eligibility traces V_{UP} and V_{DN}, that are used to determine the sign and amplitude of the weight change Δw, and that are broadcast to all the synapses belonging to the same neuron. Post-synaptic spikes are integrated by another instance of the DPI circuit, to produce a current I_{Ca} proportional to the neuron's average spiking activity. In parallel, the instantaneous value of the neuron's membrane potential V_{mem} is compared to the constant threshold bias V_{mth}. Three current-mode winner-take-all circuits (CC1, CC2, CC3) [25] compare the I_{Ca} current to the three threshold currents I_{k1}, I_{k2}, and I_{k3}. The values of the final eligibility traces are derived from the output currents I_{UP} and I_{DN}, that depend both on the outcome of the current-mode comparison, and on the state of V_{mem} with respect to V_{mth}. Specifically,

$$I_{\text{UP}} = \begin{cases} I_B, & \text{if } I_{k1} < I_{Ca} < I_{k3} \text{ and } V_{\text{mem}} > V_{\text{mth}} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$I_{\text{DN}} = \begin{cases} I_B, & \text{if } I_{k1} < I_{Ca} < I_{k2} \text{ and } V_{\text{mem}} < V_{\text{mth}} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where I_B is the winner-take-all circuit's bias current. When not null, currents I_{UP} and I_{DN} are complementary, i.e., only one of them is equal to I_B at a time.

In Fig. 3 we show the output of these circuits in response to a Poisson distributed excitatory input to a non-plastic synapse. The top trace $V_{[Ca]}$ represents the gate voltage measured from the MOSFET producing I_{Ca}, while the two bottom traces V_{DN} and V_{UP} represent the gate voltages of the n- and p-FETs producing I_{DN} and I_{UP}, respectively. The second trace from the top (V_{mem}) represents the membrane potential of the post-synaptic neuron. The dashed horizontal line in this plot represents V_{mth}.

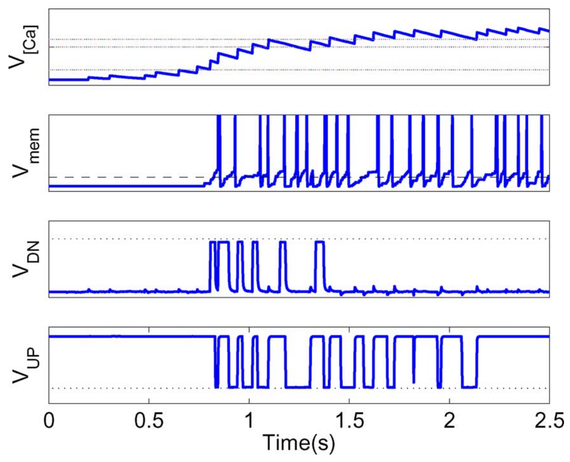


Fig. 3. Experimental data measured from the *post-synaptic module* . The top trace ($V_{[Ca]}$) represents the post-synaptic neuron's integrated spiking activity. The V_{mem} trace represents the neuron's membrane potential. The lower two traces V_{UP} and V_{DN} are the eligibility traces, which can be activated if $V_{[Ca]}$ lies within set intermediate ranges, and are disabled if $V_{[Ca]}$ is either too high or too low. In case of intermediate values of $V_{[Ca]}$, the eligibility traces are actually activated (i.e., are driven toward the dashed lines in the corresponding panels) depending on the value of the post-synaptic neuron's membrane potential.

As the neuron's spikes are integrated, the output current I_{Ca} increases with an exponential profile over time. The voltage $V_{[Ca]}$ changes accordingly, before reaching a steady-state asymptotic value that depends on the mean input frequency [24]. As I_{Ca} becomes larger than the first threshold I_{k1} both I_{UP} and I_{DN} are activated. When I_{Ca} becomes larger than the second threshold I_{k2} the I_{DN} current is deactivated (and V_{DN} changes accordingly). As I_{Ca} becomes larger than the third threshold I_{k3}, also the I_{UP} current is switched off. When not null, the magnitude of I_{UP} or I_{DN} is equal to the winner-take-all subthreshold bias current I_B. Changes in I_{UP} and I_{DN} create voltage swings of a few 100 *m*Vs in V_{UP} and V_{DN}. These small voltage changes do not create appreciable cross-talk among the metal wires that broadcast the signals spanning the entire length of the chip, including the V_{mem} node. This is a better solution for distributing these global control signals, compared to using digital voltages, as in alternative approaches [26].

Except for I_{UP} and I_{DN}, which have a common source, all voltage and current variables that determine the synaptic dynamics can be independently set.

IV. Stochastic Plasticity and Learning Rate

The VLSI spike-based learning circuits are deterministic. Variability is introduced in the system by using Poisson distributed spike trains as input signals. To characterize the stochastic nature of the weight update process, we generated synthetic Poisson distributed spike trains on a workstation and converted them into Address-Events using a dedicated PCI-AER board [12], [27].

Fig. 4 shows the outcome of two training experiments that point out the stochastic nature of the weight update mechanism. Fig. 4(a) shows an experiment that did not produce a transition in the synaptic weight stable state, while Fig. 4(b) shows an equivalent experiment in which the weight made a transition. The bottom panels in both subfigures show the input spike

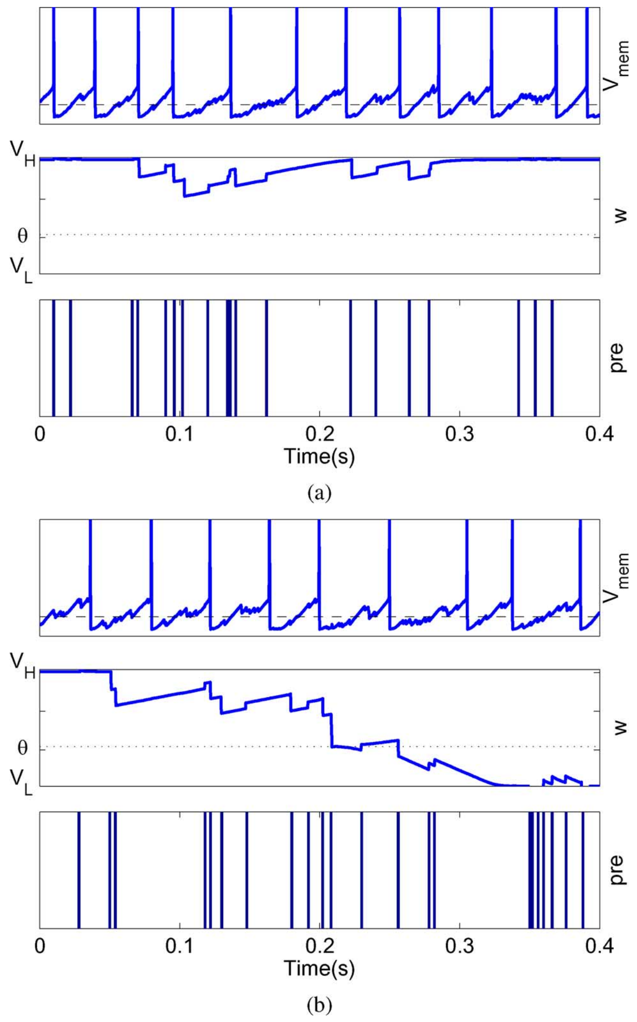


Fig. 4. Stochastic transitions in synaptic states. In both figures the non-plastic synapse is stimulated with Poisson distributed spikes, that makes the post-synaptic neuron to fire at an average rate of 30 Hz (V_{mem}). The pre-synaptic input (pre) is stimulated with Poisson distributed spike trains with a mean firing rate of 60 Hz. (a) The updates in the synaptic weight (w) did not produce an LTD transition during the 400 ms stimulus presentation. (b) The updates in the synaptic weight produced an LTD transition that remains consolidated. V_H and V_L represent the potentiated and depressed levels respectively. w denotes the synaptic weight, and θ the bistability threshold.

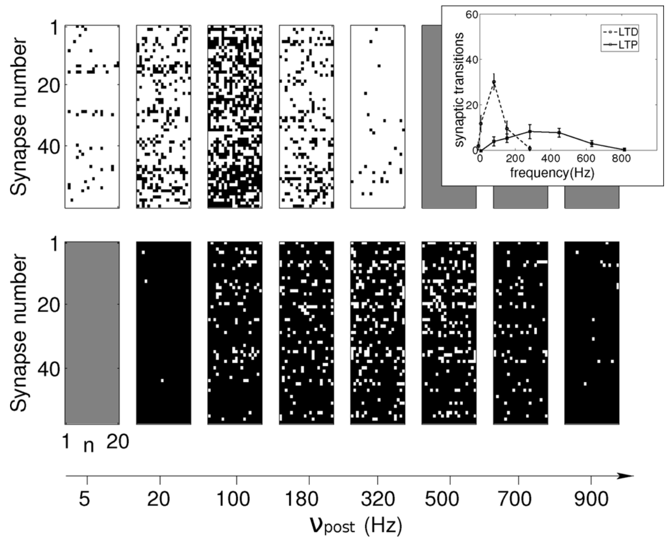


Fig. 5. LTD and LTP transitions of 60 synapses measured over 20 trials, for different values of post-synaptic frequency ν_{post}. Each subplot has the number of trials (n) on the x axis and the synapse address on the y-axis. White pixels represent a high synaptic state and black pixels a low one. The subplots in the top row represent LTD transitions, starting from an initial condition in which all synapses were initialized to the high state. The bottom row represents LTP transitions, starting from an initial condition on which all synapses were reset to a low state. The transitions in both cases are stochastic and depend on the frequency of the post-synaptic neuron (see lower ν_{post} axis.) Gray subplots indicate that no data was taken. The inset in the top right corner shows the average number of synaptic transitions and standard deviation for both LTP and LTD conditions, as a function of the post-synaptic frequency.

trains sent to the plastic synapse (pre). In both cases the mean firing rate was set to 60 Hz. The top panels show the membrane potential of the post-synaptic neuron (V_{mem}), firing at an average firing rate of 30 Hz (in both cases). The middle panels show the changes in the synaptic weight voltage. In one case the weight does not cross the bistability threshold θ and maintains its previous stable state, while in the other it switches state. As evident from the measurements, even if the mean firing rates of pre- and post-synaptic neurons are the same, the nature of the weight transition is stochastic. The transition probability can be tuned by changing the pre- and post-synaptic neuron's mean firing rates, by modifying the value of the bistability threshold θ, by changing the height of the synaptic weight jump (set by I_B), and/or by modulating the threshold bias of post-synaptic neuron V_{mth} (see Fig. 2).

Synaptic weight transitions from low to high values are denoted as long-term potentiation (LTP) events, while transitions from high to low values are denoted as long-term depression (LTD). One essential feature of this learning rule, implemented in the post-synaptic weight control module, is the non-monotonicity of the LTP/LTD probabilities as a function of the mean post-synaptic firing rate ν_{post} [7]. Such non-monotonicity, arises from the conjunction of the conditions expressed in (3) and (4) with the occurrence of pre-synaptic spikes, (equivalent to a Hebbian potentiation or depotentiation signal). The fact that the learning stops when ν_{post} is either very high ($I_{Ca} > I_{k2}, I_{k3}$) or very low ($I_{Ca} < I_{k1}$), prevents the synaptic weight from saturating. As the stop-learning occurs when the learned synaptic weights lead to a correct classification, this mechanism implements a Perceptron learning rule. In Fig. 5 we show experimental results where we measured the LTP and LTD transitions of 60 synapses over 20 trials, for different values of ν_{post}. The abscissa of each panel represents the trial number, while the ordinate represents the synapse address. White pixels represent a high weight, and black pixels a low weight. To test for LTD transitions (top row) we initialized the plastic synapses to a potentiated state and plotted a black pixel if the final state of the weight was low after 400 ms of stimulation. Similarly, the bottom row shows the LTP transitions, measured after initializing all synapses to a depressed state. Synapses can be initialized to a high or low state at the beginning of each training session by temporarily changing the threshold θ of the bistability circuit in the pre-synaptic weight-update module to Gnd or V_{dd} respectively. The location of the LTP/LTD transitions in Fig. 5 changes from trial to trial due to the stochastic nature of the plasticity mechanism, while their density changes with ν_{post}. The inset in the top right corner of Fig. 5 shows the

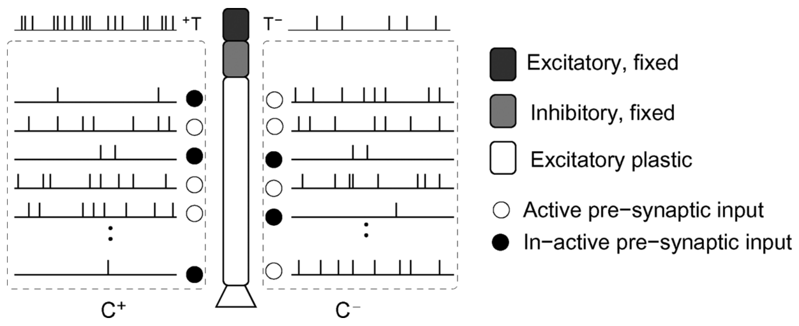


Fig. 6. Training patterns: two examples of training patterns are shown on the left and right sides of a neuron symbol. Poisson spike trains of high (30 Hz) or low (2 Hz) frequency are randomly assigned to each synapse, and represented as white or black circles respectively. These binary vectors (represented by white/black circles) are assigned to the C^+ or C^- class arbitrarily. During training, C^+ patterns are presented together with a T^+ (teacher) signal, while C^- patterns are presented in parallel to a T^- spike train. Training of C^+ and C^- class patterns are interleaved in a random order. When training the same pattern over multiple iterations, new realisations of Poisson spike trains with the same distribution of high/low mean firing rates are generated.

average number of synaptic transitions across trials, for both LTD and LTP conditions, as a function of $\nu_{\rm post}$.

V. Classification of Random Binary Vectors

In these experiments we trained the network to classify binary vectors of 60 dimensions (i.e., sent to 60 plastic synapses), using an additional teacher signal sent to the neuron's non-plastic excitatory synapse.

Fig. 6 shows the stimulation protocol used: the training patterns consist of binary vectors of Poisson distributed spike trains, with either a high mean firing rate (30 Hz), or a low one (2 Hz). The vectors are created in a random fashion with 50% high rates (white circles), and 50% low rates (black circles). These patterns are assigned to either a C^+ or a C^- class. During *training* , when a C^+ pattern is presented to the plastic synapses, a T^+ teacher signal consisting of a Poisson distributed spike train with mean frequency of 250 Hz, is presented to one of the neuron's non-plastic excitatory synapses. Similarly, for C^- patterns a T^- signal, with mean rate of 20 Hz, is used. For each new training session, we generate new Poisson spike trains keeping the same pre-defined arrangement of binary high/low mean frequencies. During *testing* , we disable the learning mechanism (e.g., by setting $I_B = 0$ A), present the input pattern without the teacher signal, and evaluate the network's response. In both training and testing paradigms the duration of spike train is 400 ms.

Prior to performing the classification experiments with multiple patterns we characterized the system's performance with control experiments using a single binary pattern. Specifically, to show how the synaptic weights evolve in the learning process we carried out the following experiment: we initialized the synaptic weights by training the neuron with 10 random binary vectors as both class C^+ and class C^-, alternatively. We then generated a new "single" pattern, and measured the neuron's response to it, with the weights fixed in their initial state. Next we trained the neuron with this pattern as a C^+ class, interrupting the training session to test the neuron (i.e., to measure its response in absence of the teacher signal) for three consecutive times. At the end of the C^+ class training session, we assigned this single pattern to the $C-$ class, and re-trained the synapses, again interrupting the training session and measuring intermediate results for three times. We repeated this experiment with 50 different binary vectors and plotted its outcome in Fig. 7. The light gray bars in all panels represent the neuron's output frequency ($\nu_{\rm post}$) histogram in the initial (random) conditions. Given that in the current VLSI implementation the down jumps in the synaptic weight are larger than the up jumps (e.g., see Fig. 4) the random initialization phase biases the distribution of firing rates towards the C^- class. The plots in the figure's top row show how the distribution of the output frequencies gradually increases as the training potentiates more synapses (dark gray bars). Similarly, the plots in the bottom row show how the distribution of firing rates gradually decreases (reflecting decreases in synaptic weights) as the neuron is trained with the same single pattern assigned to $C-$ class. The fact that the output frequency distributions do not increase beyond approximately 200 Hz, and do not decrease completely to zero is a direct consequence of the circuit's stop-learning mechanism.

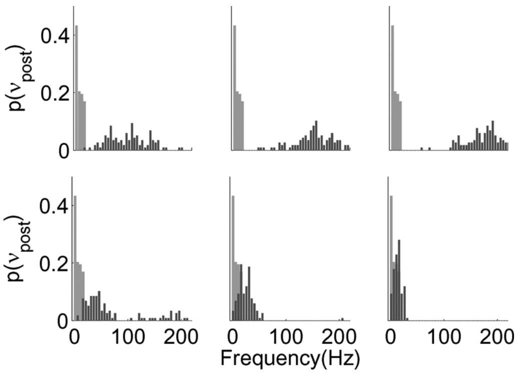


Fig. 7. Probability of output frequency as a neuron's training progresses. The gray bars, in all plots, represent the neuron's response to the input patterns in its initial conditions (before training). The black bars in the top row represent the neuron's output frequency as training of C^+ class patterns progresses. The bottom row shows the data as training of C^- class patterns progresses. Synaptic weights stop changing when the output frequency is too high or too low.

The two rightmost plots of Fig. 7 show that the output frequencies obtained from training C^+ and C^- classes are well separated, once training is completed. These results show that the device can be robustly used to perform classification of patterns of mean firing rates.

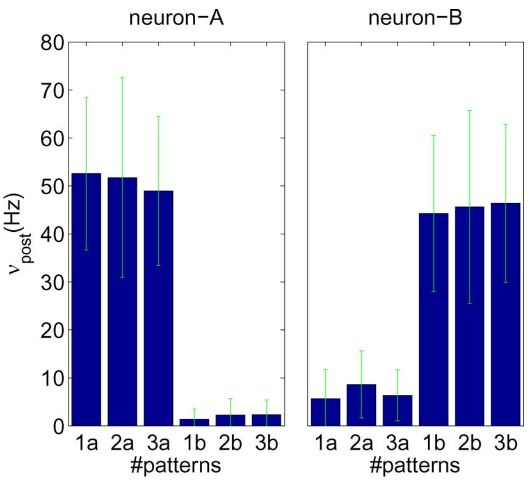


Fig. 8. Average output frequency of two neurons, after having been trained to classify six 6 input patterns belonging to two different classes. In the left plot, patterns 1a,2a, and 3a were assigned to the C^+ class and patterns 1b, 2b, and 3b to the C^- class, while training neuron-A. In the right plot the class assignments were swapped, while training neuron-B.

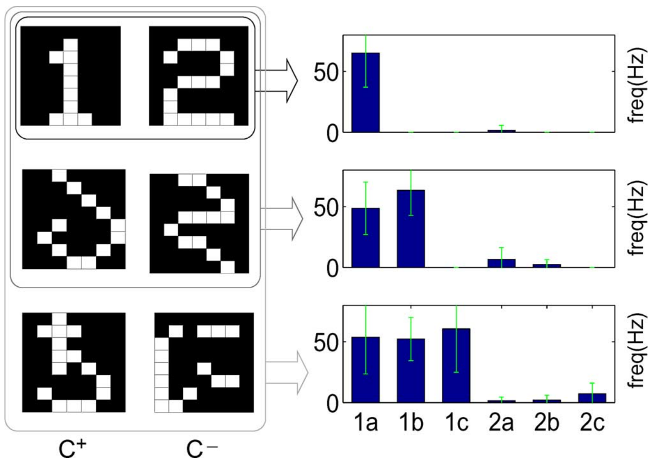


Fig. 9. Pattern recognition of 2-D binary images. The data is converted into a 1-D vector of Poisson mean firing rates (30 Hz for white pixels, 2 Hz for black pixels). Numbers 1 and 2 in three different languages are assigned to C^+ and C^- classes respectively. The plots in the right panel show classification results when one, two, or three members of each class are trained together (top to bottom). Untrained patterns in the top two rows are not tested.

To characterize the system's real classification performance, we carried out the following experiment with "multiple" (six) patterns: we trained one neuron (labeled as *neuron-A*) with three random binary vectors (*1a*, *2a*, and *3a*) assigned to the C^+ class, and with other three random vectors (*1b*, *2b*, and *3b*) assigned to the C^- class. We interleaved the training of the six patterns in random order. In parallel, we trained a different neuron (labeled *neuron-B*) with the same six patterns, but swapping the class assignments. After training, both neurons were tested with the same six patterns. To measure the mean performance and its variance, we repeated the entire procedure thirty times, creating new sets of patterns each time. In Fig. 8 we plot the average output frequencies of the two neurons in response to the six input patterns, during the testing phase. As expected, in the left plot, *neuron-A* fires at a higher frequency for all the patterns trained as C^+ class, and does not produce a significant response to the C^- class patterns. Similarly in the right plot, *neuron-B* responds with a high firing rate to the very same patterns that were assigned to the C^- class for *neuron-A*, and with a low rate for the other set of patterns. Due to the stochastic nature of the spike trains used, and the mismatch effects present in the VLSI circuits, the two neurons do not have identical performance figures. However, in this experiment a single threshold frequency is enough to correctly categorize the six patterns in two distinct classes from the neurons response.

To provide the reader with more insight on the significance of this experiment we used 2-D binary patterns representing digits (see Fig. 9) as input patterns. The 2-D input space is translated to a 1-D binary vector by simple raster scan and Poisson input spike trains are created with a high mean firing rate for white pixels, and a low rate for black pixels, as in the previous experiment. All symbols on the left column of Fig. 9 's left panel represent the digit **1** in different languages, while the symbols in the right panel represent the digit **2** . The average overlap between the patterns representing the digits in the left and right panels of Fig. 9 is 6%. A neuron is first trained with just two patterns (*1a* as class C^+, and *2a* as class C^-) and its response during testing is shown in the top row of Fig. 9 's right panel. The neuron is then trained with four patterns (the patterns in top two rows of the figure's left panel), and testing shows correct classification behavior (see middle row of figure's right panel). Finally, the neuron is trained with all six patterns. The test results in the bottom row of the figure's right panel show that the neuron can successfully *learn* to distinguish the patterns representing the character **1** from those representing the character **2** in three different languages.

In all experiments described until now, we always used 'balanced' stimuli, stimulating on average 50% of the synapses available. However the learning algorithm has been shown to work reliably also in the unbalanced case, provided inhibition is used as well [7]. Furthermore, in the experiments described up to now the number of patterns belonging to C^+ and C^- class were always kept equal. Hence, after training was complete, the neuron had to categorize the set of input patterns in two equal halves. To verify the capability of a neuron to recognize one pattern out of many, we created four random binary patterns (labeled 1 through 4) and trained four neurons (labeled *A* through *D*), with uneven class assignments. For neuron *A*, only pattern 1 was assigned to the C^+ class, and all other patterns were assigned to the C^- class. Similarly, for neuron *B* only pattern 2 was assigned to the C^+ class, and for neurons *C* and *D* only patterns 3 and 4 were assigned to C^+ respectively. After multiple iterations of the training session we tested all four neurons. This was repeated forty times with new sets of four randomly generated patterns, with identical class assignments. We used a fixed threshold frequency (20 Hz) as decision boundary and counted the number of times each neuron's output crossed it, while testing all four patterns. As expected, neuron-*A* crossed the threshold many more times in response to pattern 1, compared to the responses to other patterns, and all other neurons behaved correspondingly. The gray bars in Fig. 10 show the fraction of times (f_T) each neuron crossed the decision boundary. The height of the gray bars for the patterns corresponding to the C^+ class show the fraction of correct

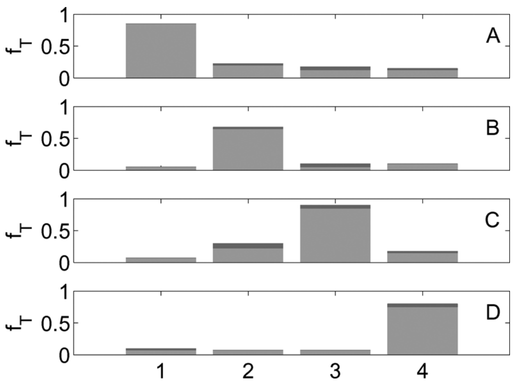


Fig. 10. Four neurons (A-D) are tested with four different random patterns (1–4), each trained with one pattern as C^+ and three other as C^-. Gray bars show the fraction of time the output crossed a fixed frequency threshold while testing. Each neuron crossed the threshold many more times for their corresponding C^+ patterns (like pattern-3 for neuron-C) compared to others. Black bars represent the fraction of time output frequency lies within a narrow band around the threshold, depicting unclassified result.

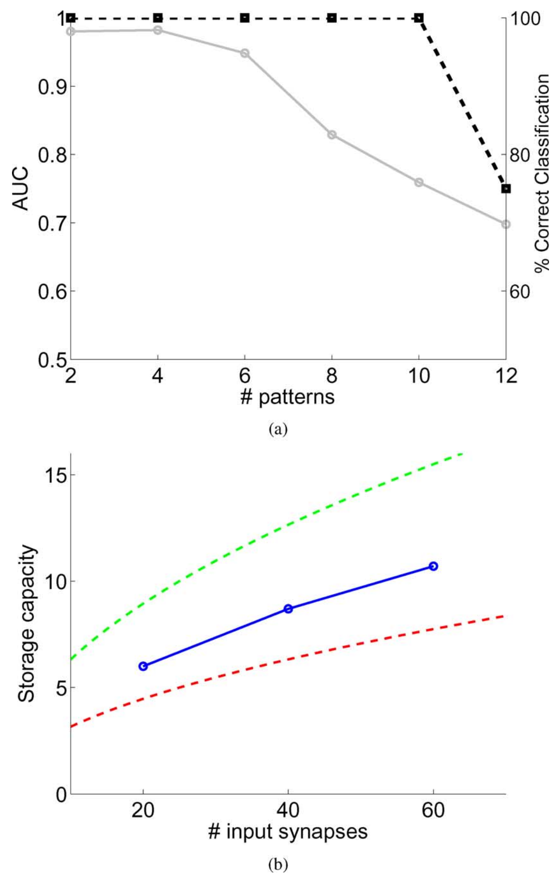


Fig. 11. Classification performance and memory capacity. (a) Area under ROC curve (*AUC*) measured for pattern classification experiments using the output neuron as a single classifier (solid line), and as 20 independent classifiers (dashed line), combined using a majority decision rule (see text for details). (b) Storage capacity of a single neuron (measured for an *AUC* value of 0.75) against the number of plastic synapses used. The solid curve shows the data obtained from chip, while the dashed lines show the theoretical predictions with and without the *stoplearning* condition.

classification, while the gray bars for patterns of the C^- class show the fraction of misclassified results. We also counted the number of times ν_{post} resided within the narrow band between 16 Hz and 20 Hz, and considered that as un-classified output. This means that the output is neither high enough to categorize the pattern as a C^+ class nor low enough for a C^- class. The thin black bars show the fraction of times this happened.

VI. Classification Performance

In order to do a quantitative characterization of the network's classification performance, we used a discrimination analysis based on the Receiver Operating Characteristics (ROC) [28]. An ROC graph is a technique for measuring the sensitivity and specificity of a binary classifier. In a ROC graph the classifier's true positive rate (number of correct classifications) is plotted against its false positive rate (number of mis-classifications) using a sliding threshold for the classifier's decision boundary (e.g., the neuron's output frequency), ranging from 0 to infinity. The area under the ROC curve (AUC) is a measure of the classification performance. While unity area denotes perfect classification, an area of 0.5 indicates that the classifier is performing at chance.

We performed a classification experiment analogous to the one done for the data shown in Fig. 8, but with different sets of input patterns, ranging from two to twelve. A ROC analysis was done on the neuron's output frequency, for each set of input patterns used. Fig. 11(a) shows the AUC values of the single neuron classifier (solid line) as a function of the number of patterns in a set. In addition, instead of using a single neuron as a binary classifier, we used 20 independent classifiers trained with the same sets of patterns. Rather than using 20 different neurons as classifiers, we trained the same neuron multiple times, with different realizations of the Poisson trains for the teacher signal, but same input pattern spike trains [29]. In the testing phase, a majority decision rule was used for the classification result. The dashed line in Fig. 11(a) shows the performance achieved using this method. This is a *boosting* technique [30] that provides clear advantages over the use of a single classifier. The improvement can be attributed to the fact that the synaptic updates are stochastic and independent on the different classifiers. As a consequence every neuron response can be regarded as the outcome of a weak classifier which makes errors independent from the other classifiers. When using multiple VLSI neurons as multiple classifiers, the effects of mismatch and variability in the fabricated circuits can actually result beneficial in this respect.

As the number of patterns to classify increases, the classification performance decreases, as expected from theory. This is due to the fact that the overlap among patterns belonging to the C^+ and C^- classes increases with the number of random patterns generated. The stop-learning properties implemented in this chip have a very strong effect on classification of overlapping patterns. We present a detailed comparison between the performance of networks with and without stop-learning mechanisms in [29]. In Fig. 11(b) we plot the storage capacity (solid line), defined as the number of patterns for which the classifier has an AUC value larger than 0.75, versus the number of input synapses (N) used by the classifier. The top and bottom traces show theoretical predictions, derived from [7], with (top dashed

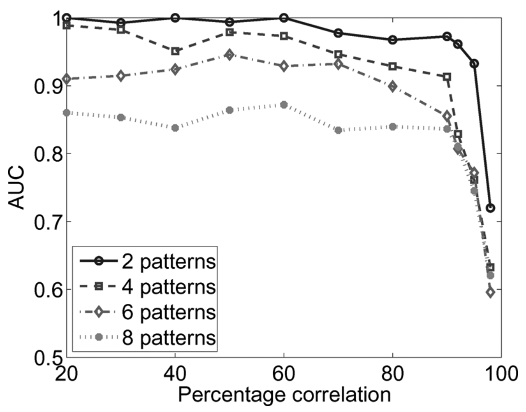


Fig. 12. *AUC* values computed for different sets of patterns (2 to 8), as a function of the percentage of correlation among the patterns. In every experiment half of the patterns are randomly assigned to the C^+ class, and half to the C^- class.

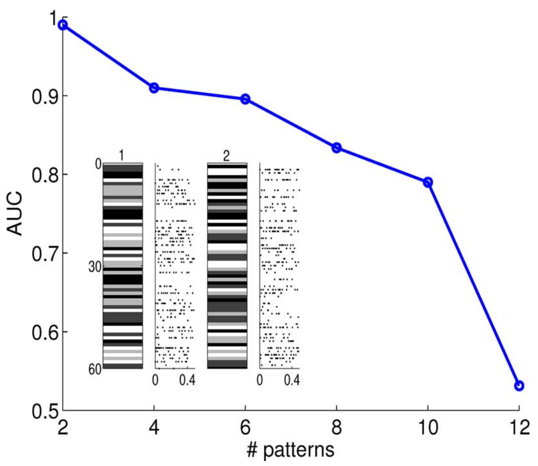


Fig. 13. Classification performance (AUC values) measured in response to graded input patterns. The patterns are formed by spike trains with four possible mean frequency values. Two typical input vectors and example spike raster plots are shown in the figure inset. As predicted by theory, classification performance degrades with increasing numbers of patterns in the input sets.

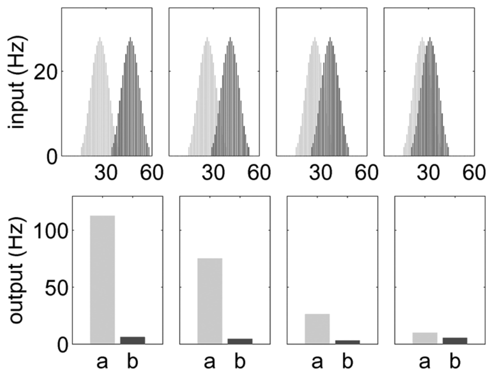


Fig. 14. Classification performance for two graded overlapping input stimuli with Gaussian profiles, *a* (gray) and *b* (black). The top row shows the two input stimuli for increasing areas of overlap. The X axis represents the input synapse address and Y axis its mean input frequency. The neuron is trained to classify *a* as a C^+ class and *b* as C^- in all cases. The bottom row shows neuron's mean output frequency in response to the *a* and *b* patterns during the testing phase.

line) and without (bottom dashed line) the stop-learning condition. As evident from the figure, the neuromorphic circuits do not perform as well as the theoretical learning rule they implement. However they preserve the same scaling properties of networks with stochastic bounded synapses [7], therefore allowing for the construction of VLSI spike-based learning networks with very large storage capacities.

VII. Correlated and Graded Patterns

The important advantage of the stop-learning mechanism implemented in this device lies in its ability to classify *correlated* patterns. To test this claim, we created correlated patterns using a prototype with 60 random binary values (as explained in Fig. 6) as a starting point. We then generated new patterns by changing only a random subset, of a size that depends on the amount of correlation. These patterns were then randomly assigned to either the C^+ or the C^- class. In the experiments that follow we systematically increased the percentage of correlation among the input patterns, and repeated the classification experiment with increasing numbers of input patterns per class, ranging from two to eight [29]. Fig. 12 shows the AUC obtained from the ROC analysis carried out on the outcome of these experiments. The curves show a substantially constant AUC value for patterns with low and intermediate values of correlation, with a rapid drop in AUC (indicating low classification performance) only when the correlation among patterns increases beyond 90%. This remarkable performance derives from both the bistable nature of the synaptic weights and the stochastic nature of the weight updates [9].

In addition to using binary vectors, we performed experiments with *graded* patterns: patterns in which the mean frequencies could be 2, 10, 20, or 30 Hz (as opposed to just 2 or 30 Hz).

Two samples of random graded input patterns are shown in the inset of Fig. 13. Example spike raster plots corresponding to the mean frequencies used are shown next to the patterns. We performed experiments similar to those described in Section VI for classifying random patterns assigned to two classes using a single neuron. We quantified the classifier's performance using the ROC analysis as shown in Fig. 13. The AUC value decreases with the numbers of patterns presented to the classifier during the training phase. The overall trend is similar to the one shown in Fig. 11(a), but the rate at which the AUC value decreases (i.e., the classification performance degrades) is much higher here. This is an expected behavior, as the similarity between input patterns is even higher for graded values, and the correlation between patterns increases as more and more are used in the training set.

To further analyze the classification performance of the device on graded input patterns, we created spatially distributed Gaussian profiles as input vectors. The profiles had standard deviation of 6 synapses and a maximum mean frequency of 30 Hz. In Fig. 14 (top row) we show two such patterns in four panels, with increasing amount of overlap. The first pattern (labeled *a*) is centered around the synapse #25, while the other pattern (labeled *b*) is gradually shifted from synapse #45 (leftmost plot) to synapse #30 (rightmost plot). The pattern *a* was trained as a class C^+ pattern, while pattern *b* was trained as a C^- class pattern. The outcome of the classification experiment, during the test phase, is plotted in the bottom row of Fig. 14.

As expected, the neuron responds with a high firing rate to pattern *a* and with a low one to pattern *b*. But the difference between the high and low mean rates decreases as the overlap between the patterns increases. The classifier manages to maintain a difference in the output firing rates, even for the case in which the patterns overlap almost completely.

VIII. Conclusion

We proposed an efficient VLSI implementation of a spike-based learning algorithm that solves both *memory encoding* and *memory preservation* problems, by using binary synaptic weights that are updated in a stochastic manner. We presented experimental data from the chip describing the detailed behavior of the learning circuits, at the single neuron and synapse level, and presented classification results of patterns of mean firing rates, at the network level. We presented experimental results in which 60 synapses/neuron were used, and we are currently evaluating real-world problems that can take advantage of all the 2048 synapses and 16 neurons available on the chip.

The results reported demonstrate the correct functionality of the spike-based learning circuits for the difficult case of random patterns. For demonstration purposes, we showed how these results could be extended also to binary 2-D images that represent digits written in different alphabets. To characterize the classification performance of the learning circuits in a thorough and quantitative way, we performed ROC analysis on multiple iterations of training and test sessions (in order to obtain significant statistics). In addition we demonstrated how the learning performance could be further improved by using *boosting* techniques. Specifically we showed how performance increased significantly when using a virtual pool of neurons trained on the same input data independently. We showed how the scaling properties of the VLSI system are in accordance with those predicted by theory [9], and used ROC analysis to demonstrate how the classification performance is robust even with highly correlated patterns. Finally, we showed how the same learning circuits can correctly classify more complex patterns that are composed of more than two mean frequencies, and that can have significant amounts of overlap in the spatial distribution of firing rates.

Due to device mismatch inherently present in the VLSI circuits, the properties of the learning circuits can vary across all synapses on the chip. Preliminary measurements indicate that these variations produce standard deviations in the inter-spike-interval (ISI) distributions, and LTP/LTD probabilities of about 20%. Software simulations carried out in [7] demonstrate that with 20% variability artificially introduced in all dynamical parameters of the algorithm the performance degrades smoothly from 5% of misclassified and non-classified patterns to 20%. Indeed the experimental results of Section VI validate these simulations and show that the spike-based learning mechanisms proposed is (by design) robust to such variations. Furthermore, despite mismatch and inhomogeneities, the device tested could perform robust, efficient, and real-time classification of complex patterns of spike trains even with significant correlations. To our knowledge, the classification performance achieved with this system has not yet been reported for any other spike-based learning VLSI chip.

References

[1] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science*, vol. 275, pp. 213–215, 1997.

[2] R. Kempter, W. Gerstner, and J. L. van Hemmen, "Hebbian learning and spiking neurons," *Phys. Rev. E*, vol. 59, no. 4, pp. 4498–4514, 1999.

[3] L. F. Abbott and S. Song, "Asymmetric hebbian learning, spike timing and neural response variability," *Advances Neural Inf. Process. Syst.*, vol. 11, pp. 69–75, 1999.

[4] L. Abbott and S. Nelson, "Synaptic plasticity: Taming the beast," *Nature Neurosci.*, vol. 3, pp. 1178–1183, Nov. 2000.

[5] R. Gütig and H. Sompolinsky, "The tempotron: A neuron that learns spike timing–based decisions," *Nature Neurosci.*, vol. 9, pp. 420–428, 2006.

[6] R. Legenstein, C. Näger, and W. Maass, "What can a neuron learn with spike-timing-dependent plasticity?," *Neural Comput.*, vol. 17, no. 11, pp. 2337–2382, 2005.

[7] J. Brader, W. Senn, and S. Fusi, "Learning real world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Comput.*, vol. 19, pp. 2881–2912, 2007.

[8] D. J. Amit and S. Fusi, "Dynamic learning in neural networks with material synapses," *Neural Comput.*, vol. 6, p. 957, 1994.

[9] S. Fusi, "Hebbian spike-driven synaptic plasticity for learning patterns of mean firing rates," *Biolog. Cybern.*, vol. 87, pp. 459–470, 2002.

[10] S. Fusi and L. F. Abbott, "Limits on the memory storage capacity of bounded synapses," *Nature Neurosci.*, vol. 10, pp. 485–493, 2007.

[11] S. Fusi, M. Annunziato, D. Badoni, A. Salamon, and D. J. Amit, "Spike-driven synaptic plasticity: Theory, simulation, VLSI implementation," *Neural Comput.*, vol. 12, pp. 2227–58, 2000.

[12] E. Chicca, A. M. Whatley, V. Dante, P. Lichtsteiner, T. Delbrück, P. Del Giudice, R. J. Douglas, and G. Indiveri, "A multichip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 5, no. 54, pp. 981–993, 2007.

[13] R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jimenez, and B. Linares-Barranco, "A neuromorphic cortical-layer microchip for spike-based event processing vision systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 12, pp. 2548–2566, Dec. 2006.

[14] T. Y. W. Choi, P. A. Merolla, J. V. Arthur, K. A. Boahen, and B. E. Shi, "Neuromorphic implementation of orientation hypercolumns," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 6, pp. 1049–1060, 2005.

[15] P. Häfliger, "Adaptive WTA with an analog VLSI neuromorphic learning chip," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 551–572, Mar. 2007.

[16] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike–timing dependent plasticity," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 211–221, Jan. 2006.

[17] E. Chicca, D. Badoni, V. Dante, M. D'Andreagiovanni, G. Salina, S. Fusi, and P. Del Giudice, "A VLSI recurrent network of integrate–and–fire neurons connected by plastic synapses with long term memory," *IEEE Trans. Neural Netw.*, vol. 14, no. 9, pp. 1297–1307, Sep. 2003.

[18] U. Mallik, R. J. Vogelstein, E. Culurciello, R. Etienne-Cummings, and G. Cauwenberghs, "A real-time spike-domain sensory information processing system," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, vol. 3, pp. 1919–1922.

[19] H. Riis and P. Hafliger, "Spike based learning with weak multilevel static memory," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2004, pp. 393–396.

[20] A. B.-i. Petit and A. F. Murray, "Synchrony detection and amplification by silicon neurons with STDP synapses," *IEEE Trans. Neural Netw.*, vol. 15, pp. 1296–1304, Sep. 2004.

[21] F. Schürmann, K. Meier, and J. Schemmel, "Edge of chaos computation in mixed-mode VLSI – A hard liquid," in *Advances in Neural Information Processing Systems*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, vol. 17, pp. 1201–1208.

[22] D. Badoni, M. Giulioni, V. Dante, and P. Del Giudice, "An aVLSI recurrent network of spiking neurons with reconfigurable and plastic synapses," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 1227–1230.

[23] D. Fasnacht, A. Whatley, and G. Indiveri, "A serial communication infrastructure for multichip address event system," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2008, pp. 648–651.

[24] C. Bartolozzi and G. Indiveri, "Synaptic dynamics in analog VLSI," *Neural Comput.*, vol. 19, no. 10, pp. 2581–2603, Oct. 2007.

[25] J. Lazzaro, S. Ryckebusch, M. Mahowald, and C. Mead, "Winner-take-all networks of $O(n)$ complexity," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, vol. 2, pp. 703–711.

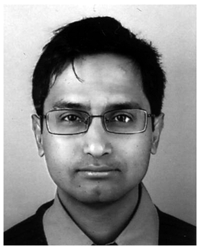
[26] S. Mitra, S. Fusi, and G. Indiveri, "A VLSI spike-driven dynamic synapse which learns only when necessary," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 2777–2780.

[27] V. Dante, P. Del Giudice, and A. M. Whatley, "PCI-AER – hardware and software for interfacing to address-event based neuromorphic systems," *The Neuromorphic Eng.*, vol. 2, no. 1, pp. 5–6, 2005.

[28] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, no. 26, pp. 861–874, 2006.

[29] S. Mitra, G. Indiveri, and S. Fusi, "Robust classification of correlated patterns with a neuromorphic VLSI network of spiking neurons," in *IEEE Proc. BioCAS07*, 2007, pp. 87–90.

[30] R. Polikar, "Essemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, 2006.



Srinjoy Mitra received the B.Sc. and B.Tech degrees from Jadavpur University, Calcutta, India, and the M.Tech, degree in microelectronics from the Indian Institute of Technology, Bombay, India, in 2003. He is currently working toward the Ph.D. degree at the Institute of Neuroinformatics (UNI-ETH), Zurich, Switzerland.

He briefly worked in microelectronic industry as an analog designer. His main research interests are neuromorphic circuits for spike based computation and analog VLSI design for bio-engineering applications.

Stefano Fusi received a Laurea degree in physics in 1992. He then worked for National Institute for Nuclear Physics (INFN) until 1999, when he received the Ph.D. degree in physics from the Hebrew University, Jerusalem.

He moved to the Institute of Physiology in Bern where he stayed until 2005. In 2003–2004 he was a Visiting Scientist at Brandeis University, Waltham, MA. Since 2005, he has been an Assistant Professor at the Institute of Neuroinformatics, ETH, Zurich, and at the Center for Theoretical Neuroscience, Columbia University, NY.

Giacomo Indiveri graduated in electrical engineering from the University of Genoa, Italy, in 1992. He received the ETH Habilitation in neuromorphic engineering, in 2006.

He is a Privat Dozent at the Institute of Neuroinformatics of the Swiss Federal Institute (ETH) and the University of Zurich. He worked as a Postdoctoral fellow with Prof. C. Koch, at Caltech from 1994 to 1996. In 1996 he moved to the Institute of Neuroinformatics, UZH-ETH, Zurich, Switzerland. He has been working on the design of biologically inspired neuromorphic systems, spike-based plasticity, neuromorphic cognitive architectures, and artificial (asynchronous, distributed) neural VLSI systems for over a decade.