

Robust classification of correlated patterns with a neuromorphic VLSI network of spiking neurons

Srinjoy Mitra[†], Giacomo Indiveri[†] and Stefano Fusi^{†∇}

[†]Institute of Neuroinformatics, UNI-ETH, Zurich

[∇]Center for Theoretical Neuroscience, Columbia University, New York

Email: [srinjoy|giacomo|fusi]@ini.phys.ethz.ch

Abstract— We demonstrate robust classification of correlated patterns of mean firing rates, using a VLSI network of spiking neurons and spike-driven plastic synapses. The synapses have bistable weights over long time-scales and the transitions from one stable state to the other are driven by the pre and post-synaptic spiking activity. Learning is supervised by a teacher signal which provides an extra current to the output neurons during the training phase. This current steers the activity of the neurons toward the desired value, and the synaptic weights are modified only if the current generated by the plastic synapses does not match the one provided by the teacher signal. If the neuron's response matches the desired output, the synaptic updates are blocked. Such a feature allows the neurons to classify spatial patterns of mean firing rates, even when they have significant correlations. If synaptic updates are stochastic, as in the case of random Poisson input spike trains, the classification performance can be further improved by combining the outcome of multiple neurons together.

I. INTRODUCTION

Memory is a fundamental component of all learning mechanisms which lead to the classification of different stimuli. It is widely believed that memory is stored in the synaptic weights of neural networks. However in real physical systems, either biological or electronic, synapses cannot store the values of their weights on long time scales with arbitrary precision and reliably. In real physical neural systems, synaptic weights are restricted to vary in a limited range and long term modifications cannot be arbitrarily small (*i.e.* they have limited resolution). Such systems cannot preserve memories for long time [1], [2], as their weights are continuously modified by ongoing activity and by the storage of new memories. Specifically, memory lifetimes are inversely proportional to the fraction of the synapses that are modified, and do not depend much on the ratio between the range in which the synaptic weights can vary and the minimal long term synaptic modifications [2]. Such a fraction can be strongly reduced if the synaptic modifications are consolidated only with a small probability, at the price of slowing down the learning process [1]. In the case of supervised learning the number of modified synapses can be further reduced. For example memory lifetimes can be greatly extended if the synapses are modified only when necessary, *i.e.* only when the response of the output neurons differs from the desired one. This is the same principle used in classical perceptron learning, and not

only it extends memory lifetimes, but it also allows to store memories whose representations are highly correlated.

It has been shown that networks of spiking neurons that use this learning strategy can classify complex patterns of spike trains ranging from stimuli generated by auditory/visual sensors to images of handwritten digits from the MNIST database [3]. Other examples of spike-driven synaptic models that focus on spike-timing dependent plasticity (such as the ones proposed in [10], [11]), do not consider the problem of memory preservation in case of realistic bounded synapses.

Here we show how it is possible to classify both random and highly correlated patterns of mean firing rates using a VLSI implementation of the plasticity mechanism described in [3] despite the inhomogeneities present in the chip.

II. SPIKE-DRIVEN PLASTICITY IN VLSI

The VLSI device used in this work comprises a network of 16 neurons, and 2048 dynamic synapses, implemented using full custom hybrid analog/digital circuits.

The full chip, fabricated using a standard $0.35\mu\text{m}$ CMOS technology, occupies an area of 6.1mm^2 . For each neuron there are 4 non-plastic excitatory synaptic circuits, that exhibit biologically plausible temporal dynamics [7], 4 inhibitory ones, and 120 excitatory ones with the additional circuits that implement the spike-driven learning algorithm (see black, grey and white blocks respectively in Fig. 3(a)).

Input and output spikes are transmitted using an asynchronous digital bus. Each spike is represented as an *Address-Event*, where the address encodes either the source neuron (while transmitting spikes) or the target synapse (while receiving). Networks of arbitrary topologies can be configured by using an Address-Event Representation infrastructure [8] and routing output spikes back into input synapses of the same chip, to additional instances of the same chip, or to other AER devices. Alternative VLSI implementations of the same spike-based learning algorithm have been recently proposed [9]. They have additional flexibility for configuring the synaptic matrix inside the chip but lack the temporal dynamics aspect of synaptic transmission.

The spike-driven learning mechanism implemented on our chip acts on the weight of each plastic synapse. The synaptic weight, stored as a voltage across a capacitor on the chip, is updated each time a the synapse receives an input spike. Upon

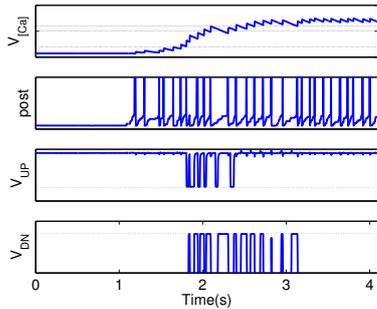


Fig. 1. Stop-learning mechanism control signals (measured data). The top trace ($V_{[Ca]}$) represents the post-synaptic neuron’s integrated spiking activity. The *post* trace represents the neuron’s membrane potential. The lower two traces V_{UP} and V_{DN} are the eligibility traces: they switch to their active state (dotted line) if the post-synaptic neuron fires at intermediate rates and are inactive if it fires at too high or too low rates.

the arrival of a pre-synaptic spike, the weight can be increased with an upward jump, if the post-synaptic depolarization is above some threshold, or decreased with a downward jump otherwise. Over long time scales a bistability mechanism drives the synaptic weight either to its maximum value, if the weight is above a threshold, or to its minimum, otherwise. Therefore the synapse makes a transition to a stable state only if enough jumps accumulate during the stimulation. Furthermore, if input spike trains have Poisson distributions, the synaptic transitions become stochastic, and their transition probability can be directly controlled by modulating the input mean frequencies. Potentiation tends to occur when both pre and post-synaptic neurons are firing at elevated rates, whereas depression tends to occur when the pre-synaptic neuron has high activity and the post-synaptic neuron has a low one (*e.g.* typical spontaneous activity levels).

All synaptic modifications are blocked on the entire dendritic tree if the activity of the post-synaptic neuron is either too high or too low. This stop-learning mechanism prevents the synapses from being updated when the total synaptic current generated by the plastic synapses matches the input generated by the supervisor and it implements the perceptron principle. Indeed the post-synaptic activity is maximal when both the plastic input and the supervisor are highly active and it is minimal when they are both inactive. This is an indication that the post-synaptic neuron would perform correctly also in the absence of the supervisor input, and hence there is no need to modify the synapses. To evaluate the mean post-synaptic activity and decide when to stop learning a current-domain low-pass filter integrates the neuron’s output spikes. The signal $V_{[Ca]}$ of Fig. 1 is a function of this integral. When $V_{[Ca]}$ assumes intermediate values the two eligibility traces V_{UP} and V_{DN} are activated. If a pre-synaptic spike stimulates a synapse while one of these two traces is active, the synaptic weight undergoes a corresponding up or down jump. The circuits that determine whether to activate the eligibility traces or not are based on a series of current-mode winner-take-all circuits, that compare the neuron’s mean output frequency with fixed thresholds [5].

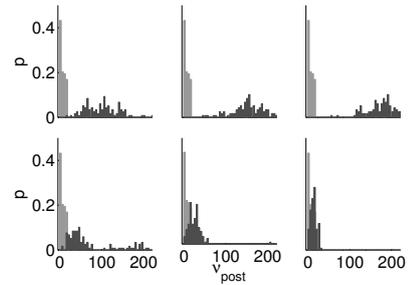


Fig. 2. Time evolution of synaptic weights as training progresses. The gray bars represent the neuron’s output frequency histogram in response to different patterns before training. Black bars represent the neuron’s output frequency histogram during training. The top row shows the evolution when a high teacher signal is used for training, while the bottom row shows the case for a low frequency teacher signal. Synaptic weights stop changing when the output frequency is too high or too low.

III. CLASSIFICATION OF RANDOM BINARY PATTERNS

We stimulated 60 plastic synapses with binary input patterns, represented as vectors of Poisson distributed spike trains with only two (high or low) possible values of mean frequencies (30 or 2Hz), and a 0.5 probability of having a high or low value.

During training the neuron receives an additional Poisson spike train (teacher signal) to one of its non-plastic excitatory synapses. Depending on the desired output frequency (high or low) the teacher signal has a mean frequency of 250Hz or 20Hz. We trained a single neuron to classify random input patterns using either a teacher high signal, or a teacher low one. We tested the state of the synapses after a few training sessions, by presenting the input pattern without the teacher signal, and measuring the neuron’s mean output frequency. In Fig. 2, we show the evolution of the synaptic weights as training progresses. The top row shows the neuron’s output frequency histogram for a high teacher signal while the bottom row shows the histogram for a low teacher signal. The light gray bars show the histogram of the neuron’s response when tested with different random patterns before training. The dark bars represent the frequency histogram measured during testing, as the training progresses. When trained with a high teacher signal synaptic plasticity pushes the output frequency to higher values (top row). However, the output frequency does not increase in an unbounded manner as the stop-learning mechanism takes over. In the bottom row, the same patterns produce low output frequencies when the neuron is trained with the low teacher signal.

To quantify the chip’s classification ability, we generated several random input patterns of mean firing rates and randomly assigned them to a C^+ class (associated to a T^+ teacher spike train of 250Hz mean rate) or to a C^- class (associated to a T^- teacher signal of 20Hz). Fig. 3 shows a set of two patterns where black circles represent a high input frequency (30Hz), and white circles represent a low input frequency (2Hz).

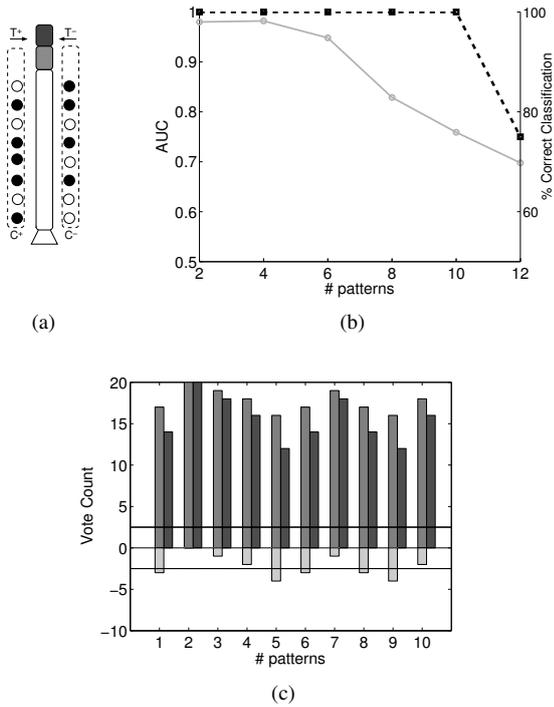


Fig. 3. (a) Silicon neuron representation with two examples of binary input patterns (to the left and to the right), labeled as C^+ and C^- class, with corresponding T^+ and T^- teacher signals. (b) Area Under ROC curve (AUC) for classification experiments of multiple input patterns with one single neuron (solid line), and classification performance of a pool of 20 neurons obtained using a majority rule decision (dashed line). (c) Individual classifier results, and majority rule decision outcome for classification of 10 different patterns. Dark and light gray bars represent the vote count of correct (positive) and incorrect (negative) classification outcomes. Black bars represent the sum of vote counts. Negative black bars (not present) would represent misclassification error, while black bars within ± 2 would represent “unclassified” decisions.

Training sessions with C^+ and C^- patterns were done in random order for many iterations with new Poisson distributions created each time. At the end of training, the neuron was tested to see if it correctly memorized the patterns. If the training phase was successful the neuron distinguished between patterns belonging to the C^+ or C^- classes by firing with a high or low frequency respectively. To decide for class C^+ or class C^- in the testing phase, it is sufficient to see if the neuron’s output firing rate is above or below a set threshold.

In order to determine the optimal classification threshold we used a discrimination analysis based on the Receiver Operating Characteristics (ROC [6]). Figure 3(b) shows performance of a single neurons classifier (solid line) and for a pool of neurons taken together (dashed line) as a function of the number of patterns being memorized. The data points on the solid line represent the Area Under the ROC Curve (AUC), which is considered a standard performance metric for classifiers with graded outputs. A magnitude of 1 denotes 100% correct classification where as 0.5 denotes classification performance down to chance level. The data points on the dashed line represent the results obtained by combining the outcome of

multiple output neurons using a binary decision mechanism. The right hand axis in Fig. 3(b) denotes the percentage correct classification performed by the pool of neurons.

The classification performance greatly improves for the multiple output neurons condition, because the synaptic updates are stochastic and independent on different neurons. As a consequence every output neuron can be regarded as a weak classifier and the errors made by each of them are independent. The binary decision mechanism that combines the result of the different output neurons was implemented using a majority rule decision process: each neuron in the pool individually classifies the learned pattern to be in C^+ or C^- and votes for the class chosen. The score is positive (+1) if the vote is correct, and negative (-1) otherwise. The total outcome is computed by counting all the scores and using a majority rule. Figure 3(c) shows the outcome of a pool of 20 neurons for 10 different input patterns. The dark and light gray bars represent the correct and incorrect votes during classification. The black bars represent the net sum. Using this method we can also define an “unclassified” outcome. For example, a pattern can be defined to be unclassified (rather than misclassified) if the difference between the correct and incorrect votes does not exceed 10% of total members in the pool.

To simplify the testing procedure in the experiment of Fig. 3(c), rather than using a pool of 20 different silicon neurons we used a single neuron and repeated the training and testing procedure on the same input patterns multiple times. For each trial the specific realization of the Poisson trains of spikes generating the teacher signal was different, as if we were considering a different output neuron.

IV. CLASSIFICATION OF CORRELATED INPUT PATTERNS

To generate correlated patterns we used a random prototype as a starting point and generated additional patterns by changing only a randomly selected subset of channels. In Fig. 4 four patterns (labeled 1-4) are generated starting from the prototype labeled '0'. Patterns in Fig. 4(a) have 30% correlation, and indeed show a small degree of similarity to the prototype. Patterns in Fig. 4(b), with 90% correlation, have most of their input channels in the same state as the prototype. In the experiments that follow we systematically increased the percentage of correlation, and repeated the experiment for increasing numbers of input patterns, ranging from two to eight.

Figure 5 shows the AUC obtained from ROC analysis for a series of experiments carried out using multiple sets of correlated patterns. The curves show a rapid drop in AUC, indicating low classification performance, when the correlation between the patterns increases above 90%, due to the bistable nature of the synaptic weights. Similar to the result described in Fig. 3(b) there is also a consistent drop in performance with increasing number of patterns to be classified.

To evaluate the effect of the stop-learning mechanism, we compared the performance of the system with the corresponding circuits enabled and disabled. We carried out a classification experiment starting with two completely orthogonal

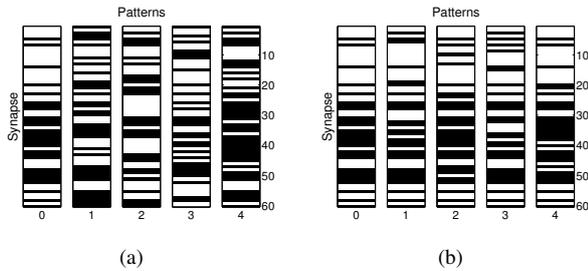


Fig. 4. Four correlated patterns (labeled 1-4) are created from the same randomly generated prototype (labeled 0). (a) Patterns with 30% correlation with the prototype. (b) Patterns with 90% correlation.

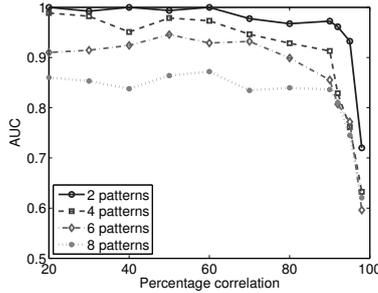


Fig. 5. Area under ROC Curve (AUC) as a function of percentage of correlation, for different sets of patterns.

sets of four patterns. The two patterns assigned to C^+ class consisted of random binary vectors for synapses 1-30 and all zeros for 31-60. Other two patterns belonging to the C^- class were generated by assigning random binary vectors to synapses 31-60, and setting the synapses 1-30 to zero.

Additional patterns with increasing overlap were generated following an analogous procedure: the random binary vectors were assigned to overlapping subsets of synapses (e.g. 1-33 and 27-60 for 10% overlap) The inset of Fig. 6 shows an example of four patterns with 20% overlap (see grey dashed box). Due to the random nature of the binary vectors, the number of correlated synapses is usually less than the overlap percentage. When the overlap is set to 100% this experiment is equivalent to that of random uncorrelated patterns described in Sec. III. Conditions with little or no overlap between patterns were classified properly (high AUC values) even with the stop-learning mechanism disabled (see the squares in Fig. 6). However, the effect of the stop-learning mechanism becomes evident for high values of overlap (see the circles in Fig. 6).

V. CONCLUSIONS

We presented robust classification results of correlated patterns of mean firing rates, using a VLSI implementation of a recently proposed spike-driven stop-learning plasticity mechanism. Our results demonstrate the correct functionality of the spike-based learning circuits and confirm the theoretical predictions about the scaling properties of the network [3]. Despite the inhomogeneities present in the VLSI device the

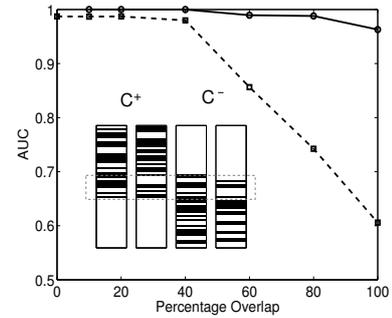


Fig. 6. Classification performance with (circles) and without (squares) the stop-learning mechanism enabled. Examples of C^+ and C^- patterns with 20% overlap are shown in the figure inset (the overlapping region is within the dashed box).

device tested could perform robust and real-time classification of spike trains. It is therefore an ideal computational block for learning tasks in adaptive neuromorphic sensory-motor systems and brain-machine interfaces. We are currently applying the chip presented in this paper to classification tasks on real-world problems (such as auditory signal classification) using real-time spike data obtained from AER sensory devices.

ACKNOWLEDGMENT

This work was supported by the Swiss National Science Foundation grant no. PP00A106556, the ETH grant no. TH02017404, and by the EU grants ALAVLSI (IST-2001-38099) and DAISY (FP6-2005-015803).

REFERENCES

- [1] D. J. Amit and S. Fusi, "Dynamic learning in neural networks with material synapses," *Neural Computation*, vol. 6, p. 957, 1994.
- [2] S. Fusi and L. F. Abbott, "Limits on the memory storage capacity of bounded synapses," *Nature Neuroscience*, vol. 10, pp. 485–493, 2007.
- [3] J. Brader, W. Senn, and S. Fusi, "Learning real world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Computation*, 2007, (In press).
- [4] G. Indiveri and S. Fusi, "Spike-based learning in VLSI networks of integrate-and-fire neurons," in *Proc. IEEE International Symposium on Circuits and Systems, ISCAS 2007*, 2007, pp. 3371–3374.
- [5] S. Mitra, G. Indiveri, and S. Fusi, "Learning to classify complex patterns using a VLSI network of spiking neurons," in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge (MA): MIT Press, 2008, (In Press).
- [6] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, no. 26, pp. 861–874, 2006.
- [7] C. Bartolozzi and G. Indiveri, "Synaptic dynamics in analog VLSI," *Neural Computation*, vol. 19, pp. 2581–2603, Oct 2007.
- [8] E. Chicca, A. M. Whatley, V. Dante, P. Lichtsteiner, T. Delbrück, P. Del Giudice, R. J. Douglas, and G. Indiveri, "A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity," *IEEE Transactions on Circuits and Systems I, Regular Papers*, vol. 54, pp. 981–993, 2007.
- [9] D. Badoni, M. Giulioni, V. Dante, and P. Del Giudice, "An aVLSI recurrent network of spiking neurons with reconfigurable and plastic synapses," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, IEEE. IEEE, May 2006, pp. 1227–1230.
- [10] R. Güttig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions," *Nature Neuroscience*, vol. 9, pp. 420–428, 2006.
- [11] R. Legenstein, C. Näger, and W. Maass, "What can a neuron learn with spike-timing-dependent plasticity?" *Neural Computation*, vol. 17, no. 11, pp. 2337–2382, 2005.