Learning attractors in an asynchronous, stochastic electronic neural network

P. Del Giudice[1], S. Fusi[2] , D. Badoni[3], V. Dante[1] and D.J. Amit[4]

[1] Istituto Superiore di Sanità, Physics Lab and

INFN Sezione Sanitá, Viale Regina Elena, 299, 00161, Roma

[2] INFN Sezione Roma 1, Universitá "La Sapienza", Pza Aldo Moro 2, Roma

[3] INFN Sezione Roma 2, Universitá di Tor Vergata,

Viale della Ricerca Scientifica 1, 00133, Roma

[4] Dipartimento di Fisica, Università "La Sapienza",

INFN Sezione Roma 1, Pza Aldo Moro 2, Roma and

Racah Institute, Hebrew University of Jerusalem

March 16, 1998

Abstract

`LANN27` is an electronic device implementing in discrete electronics a fully connected (full feedback) network of 27 neurons and 351 plastic synapses with stochastic Hebbian learning. Both neurons and synapses are dynamic elements, with two time constants – fast for neurons and slow for synapses. Learning, synaptic dynamics, is analog and is driven in a Hebbian way by neural activities. Long term memorization takes place on a discrete set of synaptic efficacies and is affected in a stochastic manner. The intense feedback between the non-linear neural elements, via the learned synaptic structure, creates in an organic way a set of attractors for the collective retrieval dynamics of the neural system, akin to Hebbian learned reverberations. The resulting structure of the attractors is a record of the large scale statistics in the uncontrolled, incoming flow of stimuli. As the statistics in the stimulus flow changes significantly, the attractors slowly follow it and the network behaves as a palimpsest – old is gradually replaced by new. Moreover, the slow learning creates attractors which render the network a prototype extractor: entire clouds of stimuli, noisy versions of a prototype, used in training, all retrieve the attractor corresponding to the prototype upon retrieval.

Here we describe the process of studying the collective dynamics of the network, before, during and following learning, which is rendered complex by the richness of the possible stimulus streams and the large dimensionality of the space of states of the network. We propose sampling techniques and modes of representation for the outcome.

1 Introduction

The electronic neural network (`LANN27`) is on the one hand an implementation of Hebb's idea about learning as an unsupervised process that forms a synaptic structure able to maintain *reverberations* upon future stimulation. In the words of Hebb [14]:

> Let us assume that the persistence or repetition of a reverberatory activity (or "trace") tends to induce lasting cellular changes that add to its stability... When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.[14]
>
> It seems that *short-term* memory may be a *reverberation* in the closed loops of the cell assembly and between cell assemblies, whereas *long term* memory is more structural, a *lasting* change of synaptic connections. [p. 110, emphases in the original]

On the other hand it is an implementation of some lessons emerging from the neurophysiological study of the inferotemporal cortex of performing monkeys[18, 19, 2]. Those lessons are: 1. That reverberations (attractors) exist[18, 23, 5]; 2. That the formation of reverberations is a very slow process of learning; 3. That such learning and dynamics are not directly related to the task the monkey is performing, and in this way embed quite automatically the context of the learning process into the internal representations (reverberations, *working memory*).

We have therefore constructed a system with neural-like elements connected by synapses whose dynamics is unsupervised and depends exclusively on the activities of the two neurons they connect. At the same time we have found that to maintain large analog depth in the synapses for very long periods, in the absence of neural activity, is very difficult, which would make it also biologically rather implausible. Hence, on long time scales the implemented synapses have only three values (see below). Given synapses with a small discrete number of stable states, one runs into a severe problem of memory capacity, of order log(the number of neurons) [4]. Such a constraint appears very general either in material implementations of neural models or in plausible biological modeling. The solution implemented is stochastic transitions between the discrete, stable synaptic states, with low transition probabilities. See e.g. [8, 11]. On the other hand, stochastic learning is a natural way to affect learning in recurrent networks of spiking neurons.

To summarize, the network is a first attempt to study empirically the computational implications of an implementation of the Hebbian paradigm with neurons and synapses with plausible dynamical characteristics. What we monitor is the process of attractors formation in an electronic recurrent neural network (`LANN27`), of 27 neurons and 351 synapses, which implements neural dynamics against a background of plastic synapses, whose dynamics is driven in turn by the neural activities. The synaptic dynamics (learning) is stochastic and is studied under various "environmental" conditions, i.e. different streams of input stimuli. The methodological approach is explained and motivated in [8], where a detailed description of the single elements (neurons and synapses) of the network is given. Here we recapitulate in a synthetical form the electronic schemes of a neuron (Fig. 1) and of a synapse (Fig. 2).

In this paper we give a detailed account of the characterization of the system's dynamics and of the learning process. The extensive tests performed to this end provide both insight into the implications of stochastic learning for real, finite size, noisy networks and methodological suggestions for the next generation networks of spiking neurons implemented in VLSI chips, which are now under development.

2 The `LANN27` network

Relevant features of `LANN27` are:

1. Neurons are analog and are implemented with high gain amplifiers (see Fig. 1). Synapses have 3 stable states, preserved indefinitely by a stochastic "refresh" mechanism. On short time scales relative to the intrinsic time constant of the device implementing the synapse (the capacitor `C` in Fig. 2), the synapse integrates a Hebbian source term induced by the stimulus. This analog charging of the synapse can in turn result in stochastic synaptic transitions from one stable synaptic state to another, with probabilities that depend on a noise source associated with each synapse, and on the strength and duration of the stimulus.

2. Transition probabilities associated with the above stochastic discretization mechanism are small, to allow the network to maintain a significant memory span along the temporal flux of stimuli (avoiding the log catastrophe mentioned in the Introduction) [11] (see Section 3.2).

3. The network undergoes a double, unsupervised dynamics for neurons and synapses. Learning and retrieval are not logically separated; they are distinguished only by the nature of the stimuli. Stimuli presented very briefly or with small amplitude do not provoke synaptic modification and provoke retrieval. Longer, stronger stimuli cause learning. The neurons are the 'fast' dynamical variables, while the typical time scales for synaptic changes are much longer (See e.g. Section 3.1).

4. The Hebbian self-organization of the (symmetrical) synapses produces an attractor structure in the network's state space. A learnt stimulus (attractor) acts as the *prototype* of a class, the latter being the set of all input stimuli that lead the network to that attractor under the neural dynamics. All the stimuli belonging to the same class define the *basin of attraction* of their prototype. Learning is the process of slow synaptic modification that translates the statistics of the stream of input stimuli into the attractor structure.

5. The temporal and spatial statistics of the flux of stimuli is unconstrained. Stimuli may appear in the flow in any order, any number

of times, with or without noise, with arbitrary inter-stimulus intervals and each for an arbitrary length of time.

6. The electronic implementation is completely analog and asynchronous. The choices made in the implementation are an interplay between biological plausibility and hardware constraints. See also the Discussion.

2.1 Neuronal and synaptic dynamics

2.1.1 Neural dynamics

Neuron number i is at time t in state $s_i(t)$. If at time t the synaptic values are $J_{ij}(t)$, then the recurrent input h_i to neuron i, from the network's feedback synapses, evolves according to

$$\tau_h \dot{h}_i(t) = -h_i(t) + \sum_{j \neq i} J_{ij}(t) s_i(t) \tag{1}$$

and a transfer function ϕ determines the state of activation of the neuron:

$$s_i(t) = \phi\left(h_i(t) + H_i(t)\right)$$

where $H_i(t)$ is the external stimulus afferent on the network at time t.

The circuit that implements this effective dynamics is schematically described in Fig. 1.

The neuronal time constant τ_h is chosen much shorter (by a factor of about 10^2-10^3) than the synaptic one. This allows us to ignore the precise moment at which the synaptic values are evaluated in Eq. 1. The neurons are analog and are implemented with operational amplifiers. At high gain the transfer function ϕ tends to the sign function and the neuronal variable can be approximated by a binary variable as in the Hopfield model. This is the case for the tests to be described.

2.1.2 Synaptic dynamics

Fig. 2 sketches the electronic design of the synapse.

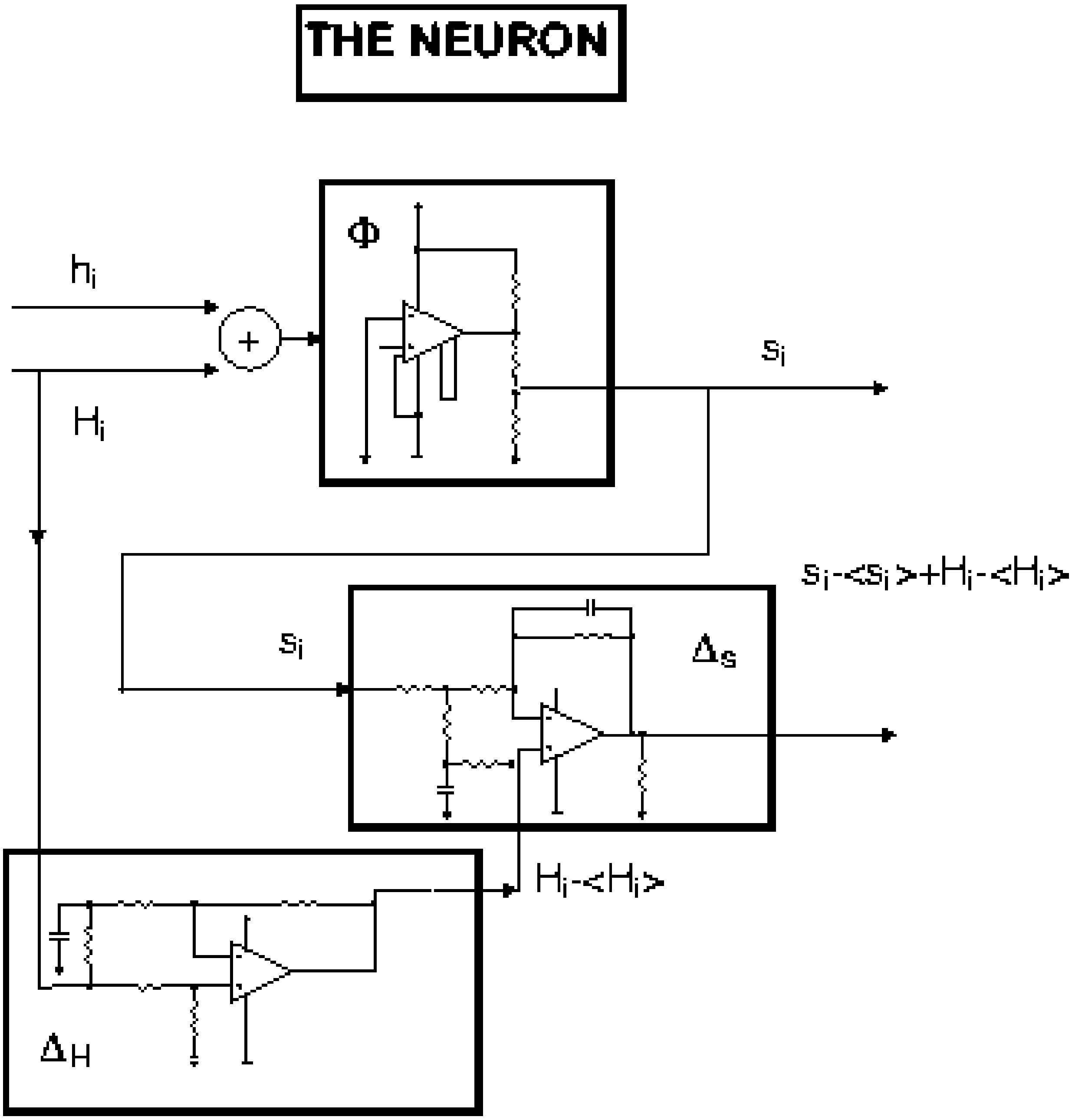

Figure 1: The electronic scheme of the neuron. Inputs to the neuron are: h_i (the recurrent input from all other neurons) and H_i (the external input). Their sum is input to the transfer function box Φ (Eq. 1), a high gain amplifier, whose output is the neuron's state of activation s_i. This in turn is fed both into the synaptic part (see Fig. 2) for the calculation of h_i (Eq. 1) and into the box Δ_s for the calculation of the terms composing the Hebbian source B_{ij} (Eq. 3). The box Δ_H performs the average $H - \langle H \rangle$, which, together with s_i, enters the Δ_s box to produce $H - \langle H \rangle + s - \langle s \rangle$ (Eq. 4).

The synaptic efficacy J_{ij} is associated with the voltage across the capacitor `C`. Its dynamics involves the integration of a stimulus dependent, Hebbian source. In absence of the source, the synaptic value will maintain its analog efficacy on a time of order τ_c. On the other hand, the integrated value determines, through a stochastic mechanism, the settling of J_{ij} on one (of 3) stable state, which is preserved indefinitely in the absence of stimuli. The time evolution of J_{ij} is described by:

$$\begin{aligned}\tau_c \dot{J}_{ij}(t) &= -J_{ij}(t) + B_{ij}(t) + \\ &\quad J_c \Theta\left(J_{ij}(t) - J_{0+} - w_{ij}(t)\right) - J_c \Theta\left(-J_{ij}(t) - J_{0-} + w_{ij}(t)\right)\end{aligned} \tag{2}$$

The first two terms on the right hand side describe the deterministic, analog charging of the capacitor `C` (with time constant τ_c) driven by the term $B_{ij}(t)$. $B_{ij}(t)$ represents the learning source which is stimulus specific:

$$\begin{aligned}B_{ij}(t) &= \\ &\alpha\left[s_i(t) - \langle s_i\rangle_{\tau'} + H_i(t) - \langle H_i\rangle_{\tau''}\right]\left[s_j(t) - \langle s_j\rangle_{\tau'} + H_j(t) - \langle H_j\rangle_{\tau''}\right]\end{aligned} \tag{3}$$

where the angular brackets $\langle ...\rangle_\tau$ denote the temporal mean in a window of width τ. The boxes Δ_s and Δ_H in Fig. 1 schematically describe the circuits which perform the computation of these averages. $B_{ij}(t)$ is a Hebbian learning term, since it is proportional to the product of the activities of pre- and post-synaptic neurons. H_i accounts for the enhanced activity related to an incoming stimulus. The terms $\langle s\rangle_{\tau'}$ prevent learning from the persistence in an attractor state for a time longer than τ'. The analogous terms $\langle H\rangle_{\tau''}$ capture the adaptation process taking place when a stimulus persists for a long time ($\tau'' \gg \tau'$). These terms compensate for the fact that neurons in the implemented network operate most of the time at saturation levels of the gain function.

The last two terms in Eq. 3 represent the stochastic (refresh) mechanism that stabilizes the synaptic efficacy at one of the values J_c, 0 and $-J_c$ and preserves them indefinitely in the absence of stimuli (`REFRESH`-box in Fig. 2). The step function Θ of Eq. 3 is 1 when the argument is positive and 0 otherwise. J_{0+} and J_{0-} are mean thresholds for the stochastic mechanism. While in principle symmetry considerations suggest the choice $J_{0+} = J_{0-}$, it

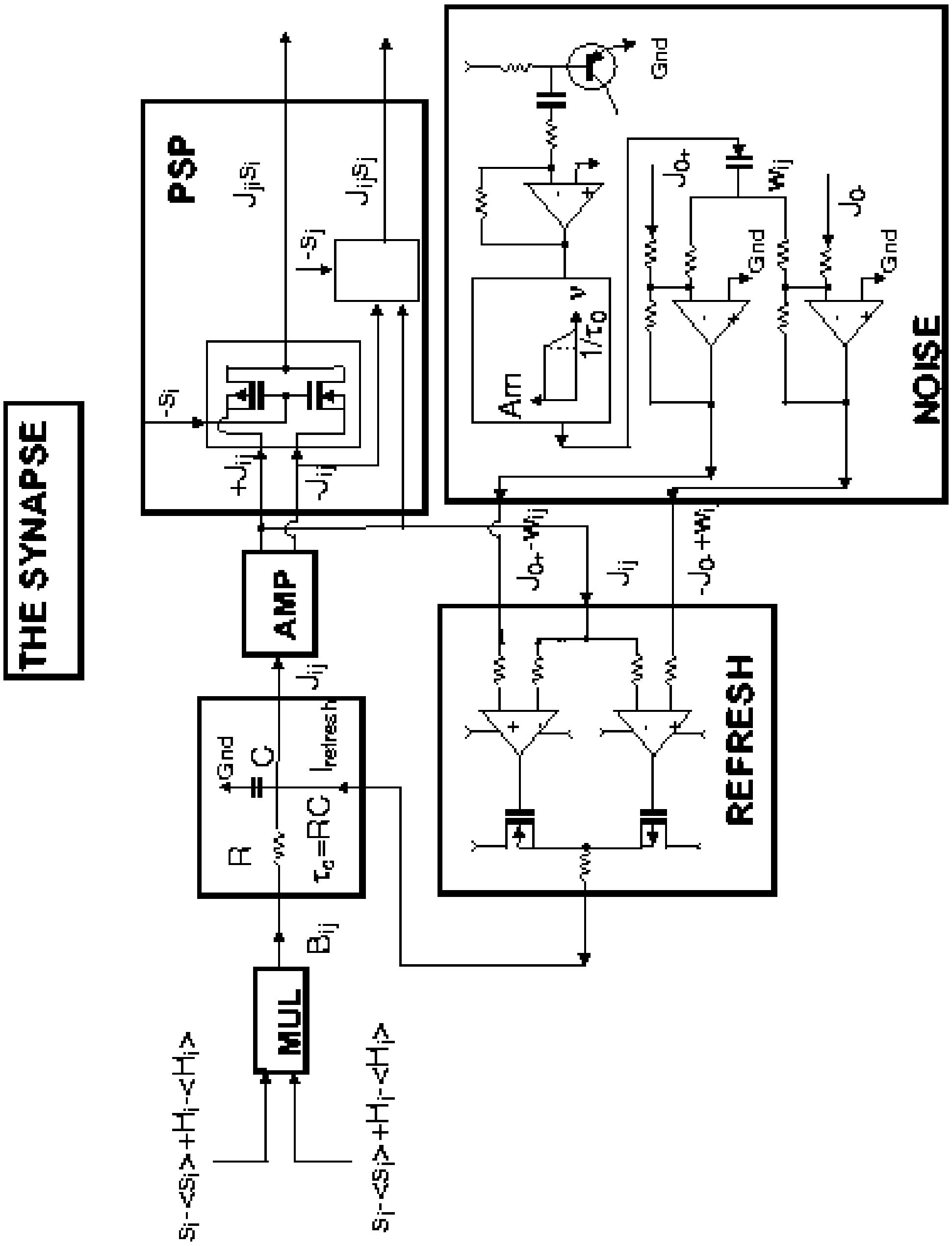

Figure 2: The electronic scheme of the synapse. From the left: the two learning terms (Eq. 4) are multiplied to obtain the Hebbian source B_{ij}. B_{ij} charges the synaptic capacitor `C`; the voltage across `C` is the analog synaptic efficacy J_{ij}. Long term memory is preserved by the stochastic refresh mechanism (box `REFRESH`); the box `NOISE` produces the two noisy thresholds ($J_{0+} - w_{ij}$ and $-J_{0-} + w_{ij}$) appearing in Eq. 3. The noise w_{ij} is generated by an inversely polarized junction. The filter appearing in the figure controls the frequency spectrum of the noise (the lower the frequency pass of its fluctuations the smaller the transition probabilities - see [8]). The noisy thresholds enter the `REFRESH` box in which J_{ij} is composed with $J_{0\pm}$, and the refresh current is determined accordingly. The `PSP` box computes two terms: $J_{ij}s_j$ and $J_{ij}s_i$ to be summed as in Eq8 1 to obtain h_i. The electronic scheme is given for only one of the two terms inside `PSP`, the other is analogous. The scheme adopted for the `PSP` box relies on the symmetry $J_{ij} = J_{ji}$ of the synapses.

was found convenient in the implementation to keep them independent. w_{ij} is the fluctuating part of the synaptic refresh threshold. It has zero mean and maximum amplitude Δ ($\Delta = \max |w_{ij}(t)|$) (See box `NOISE` in Fig. 2).

The stochastic learning mechanism is illustrated in Fig. 3 (details are given in [8]). The figure shows a sample time evolution of a synaptic efficacy starting from 0 at time 0, when a stimulus is presented. The stimulus happens to drive the efficacy towards positive values and into the shaded region where the threshold fluctuates. At $t = 4000\mu s$ the stimulus is removed, before the fluctuations bring the threshold below the instantaneous synaptic value; thus the decay term in Eq. 3 drives J towards its initial value: no transitions occurred. After the next stimulus appears at $t = 6000\mu s$, a fluctuation activates the transition mechanism, and J is driven towards $B(t) + J_c$. When the stimulus is removed, J relaxes to the stable value J_c. In this case the stimulus provokes a transition. The same type of mechanism is responsible for the other possible transitions: $J_c \to 0$, $-J_c \to 0$, $0 \to -J_c$. Also 'double transitions' ($J_c \to -J_c$ and $-J_c \to J_c$) are possible. The probability that a transition occurs is determined, for a given stimulus, by the amplitude of H, by the statistics of the fluctuations of the threshold, by the time constant τ_c of the synapse and by the time the stimulus persists, though the effect is cut off for presentations longer than τ''. Double transitions occur with much smaller probability.

In a network with a finite set of stable synaptic values, the probability that each synapse has to make a transition to one of these states, under the influence of an incoming stimulus, sets the average rate at which the information about past stimuli fades away as an effect of the new ones. Consequently the network exhibits a *palimpsest* property [20, 21], and does not undergo the kind of blackout catastrophe that occurs, for example, in the Hopfield or Willshaw models (see e.g. [6, 24, 22, 1]).

For such a learning mechanism, if the probabilities of synaptic transitions are high, each stimulus changes significantly the synaptic configuration, destroying most of the memory the network had about the past[1]. Small transition probabilities allow for a longer memory span[11].

As the transition probabilities are lowered, the synaptic matrix can keep

[1]In sharp contrast with the Hopfield model in which the number of possible synaptic states grows with the number of patterns presented.

track of a longer series of incoming stimuli [2]: the smaller the average number of synapses that change their state as an effect of a stimulus, the smaller the chance each stimulus has to overwrite previously induced transitions. On the other extreme, if transitions are frequent, the causal dependence of the synaptic structure on old stimuli is rapidly broken, as an effect of successive changes of synapses affected by successive stimuli. This qualitative picture can be made quantitative and precise (see [4, 11]).

The price paid is that with such low transition probabilities multiple presentations of a pattern are needed for the network to develop a corresponding attractor, as seems to be the case for inferotemporal cortex[18]. So, even if a given pattern is in the memory span, its attractor could have not yet been formed, and it could be impossible to retrieve it.

Moreover, the overlaps among the patterns in the memory span introduce larger and larger interference effects, and eventually destroy the ability of the network to retrieve these patterns, setting its limit of capacity.

3 Tools for testing `LANN27`

`LANN27` is conceived to be able to dynamically convert an arbitrary flux of stimuli into a set of attractors in state space. This set of attractors varies on long time scales in response to changes in the statistics of the environment – the flux of arriving stimuli. The arbitrariness in the stimulus stream, mentioned in the Introduction, refers both to the statistical structure of the set of stimuli, e.g. their correlations, as well as to the temporal organization of the stream. In practice, the flexibility in the choice of the input flow of stimuli was somewhat constrained by the limited resources of the network.

Testing the `LANN27` requires:

- hardware and software tools for communication with the network;
- characterization of the probability distributions of transitions as functions of the relevant parameters;
- protocols producing a variety of input streams of stimuli;

[2]A set of patterns (neural configurations) is given, from which stimuli are chosen. The terms "pattern" and "stimulus" are used interchangeably in what follows.

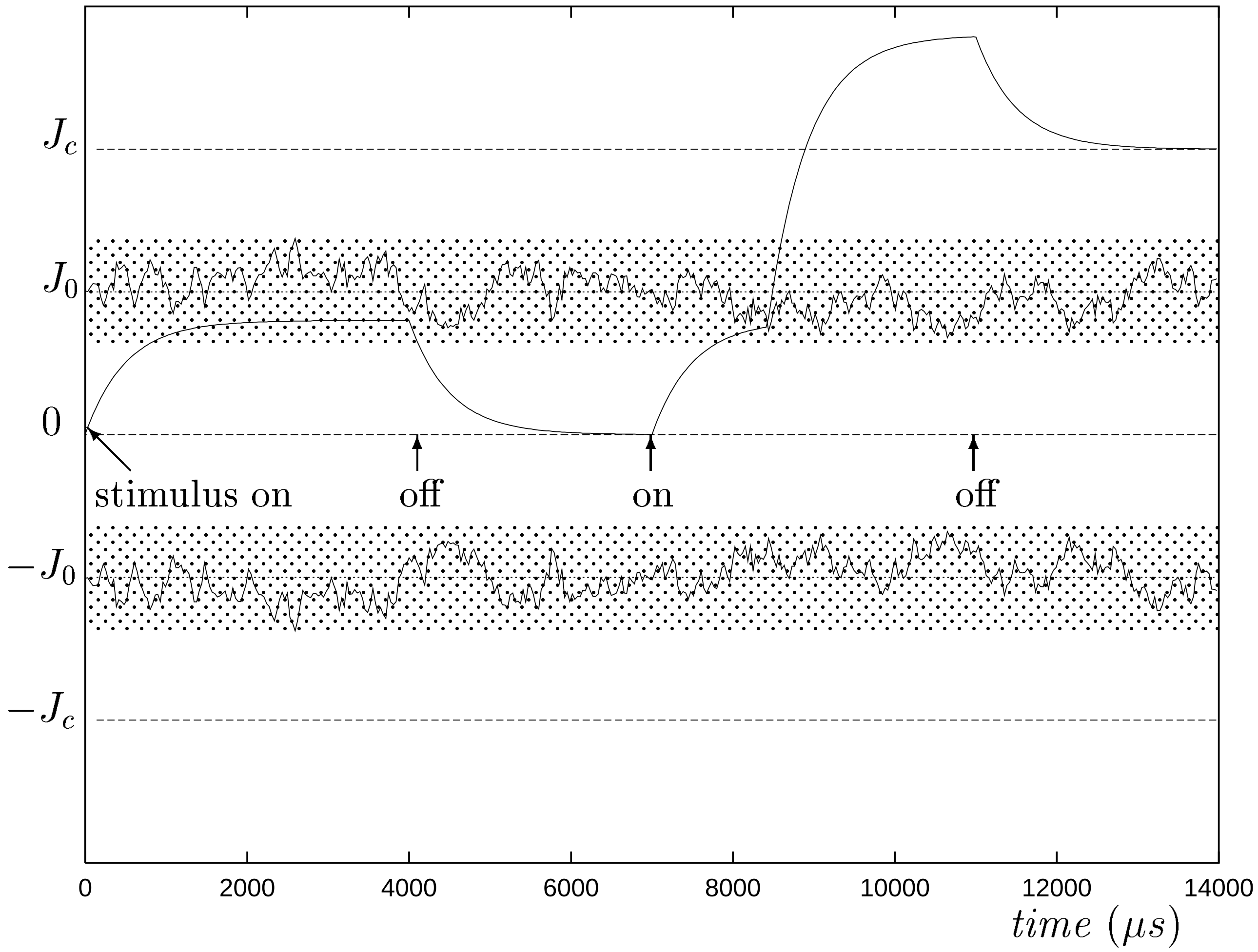

Figure 3: Example of the stochastic learning mechanism. $(-J_c, 0, J_c)$ are the synaptic stable efficacies; $(-J_0, J_0)$ are noisy thresholds fluctuating in the shaded strip. If the analog value of the synapse (solid curve), driven by the stimulus, does not reach the threshold, it decays to the previous stable state (0) upon removal of the stimulus, while it is clipped to J_c if the random threshold happens to go below it. An analogous mechanism governs the other transitions.

- characterization of the collective behaviour of the neurons in the network as shaped by the learning process:
  - a compact dynamical description of the development of attractors and their basins of attraction;
  - a suitable definition of observables to assess the limit of memory capacity of the network;
  - a compact description of the distribution of attractors in the state space of the network.

3.1 Communication and parameter tuning

The input-output of `LANN27` is managed by a PC via a custom-made programmable interface connected to the serial port. The interface (based on a MICRO440E controller) performs digital to analog conversion for communication from the PC to the `LANN27`, and handles the communication protocol, timing and addressing.

Input information to be sent to the `LANN27` includes:

- values for tunable hardware parameters: the magnitude of the positive and negative mean values of the synaptic thresholds (J_{0+} and J_{0-})
- characteristics of the input stimuli: the analog intensity of the external stimuli (H in Eq. 1; $|H|$ is equal for all neurons of a given stimulus)
- length of stimulus interval (T_p)
- the binary ($\pm H$) pattern encoding each stimulus.

The choice of the values for the parameters was guided by the following considerations:

- the values for J_{0+} and J_{0-} were chosen so that the probability distributions over the various synaptic transitions be as similar as possible. Hardware inhomogeneities cause both a spread in the probabilities of a given transition among synapses, and differences among probability distributions for different transitions.

- the amplitude of the external stimulus H is chosen so that:
  1. the Hebbian source drive J inside the shaded region in Fig. 3;
  2. the neuronal dynamics, during the presentation of a stimulus, be dominated by the external stimulus, relative to the recurrent (feedback) activity h_i. The actual value of H is about 2.5 times maximal recurrent activity.
- the stimulus presentation time $T_p = kt_0$, where $t_0 = 72\mu s$ (determined by the clock on the interface) and k is an integer.

Information to be extracted from the `LANN27`:

- the activities of the neurons are extracted by acquiring the pattern of their signs. Recall that in our tests neurons work essentially at saturation, so the pattern of their signs provides a virtually complete information on their state. All values are read simultaneously into a buffer, and then sequentially transmitted to the serial port.
- the synaptic configuration: the 351 synapses are read sequentially after all of them have settled to their stable values.

3.2 Synaptic transition probabilities

The synaptic dynamics is monitored in terms of the transition probabilities between stable synaptic efficacies, as mentioned in Section 2.1, rather than by the detailed synaptic dynamics described by Eq. 3 [3]. Their dependence on the relevant tunable parameters (the presentation time T_p and the intensity $|H|$ of the input stimulus) was studied to arrive at a choice of a working set.

To obtain an estimate of the transition probabilities a long series ($\mathcal{O}(10^4)$) of random uncorrelated stimuli was presented to the network. The relative frequencies of transitions were recorded for each synapse, i.e. the ratio of the number of transitions of each type that occurred, to the number of cases in which the particular type of transition was allowed by the source term.

[3] There are, however, situations in which the analog transients of the synaptic dynamics play an important role: such are, for example, learning schemes in which patterns are presented to the network in a fixed order, and the structure of attractors can be affected by the temporal correlations in the sequence of stimuli [10] (such study is in preparation).

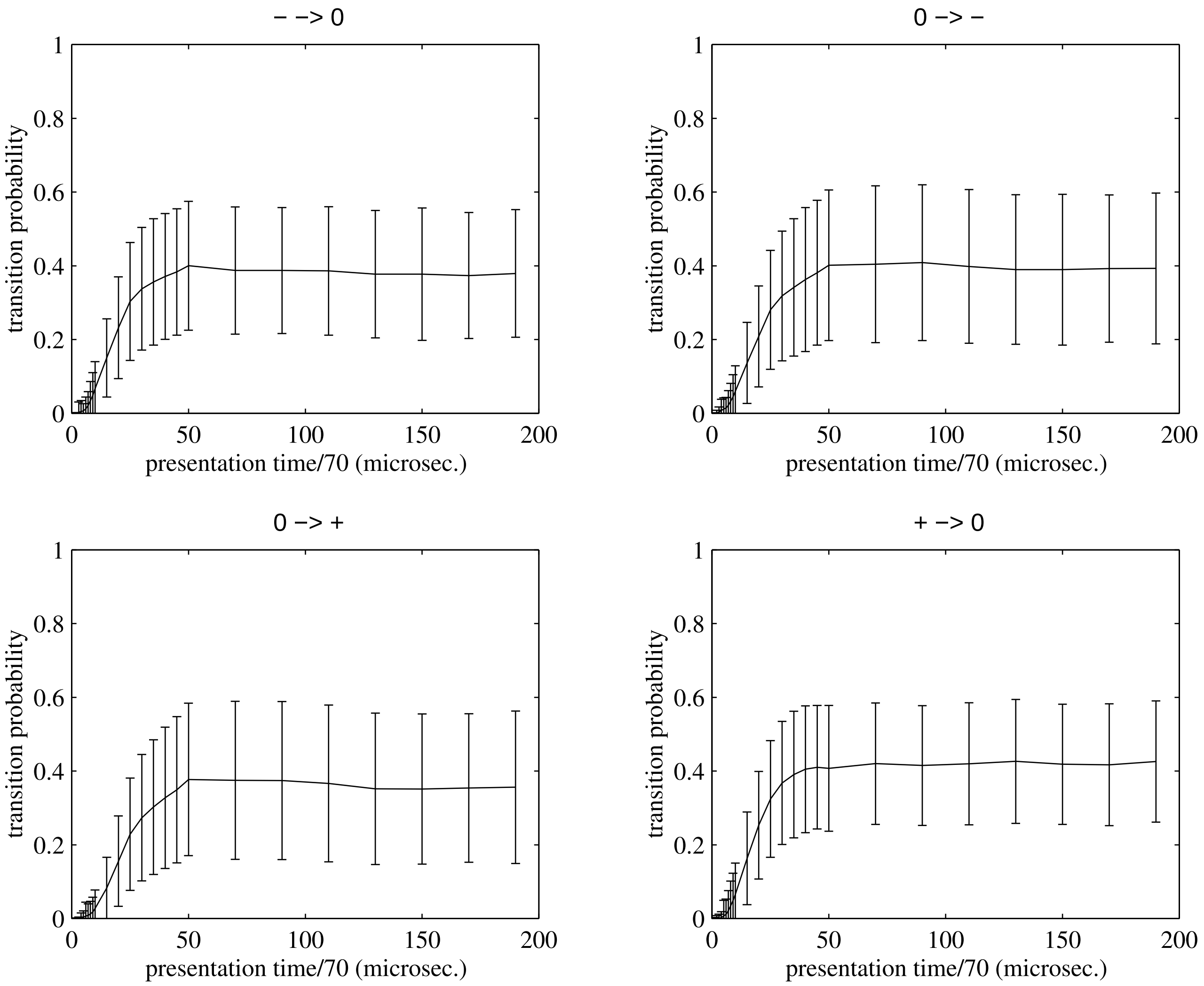

Figure 4: The four transition probabilities $0 \to J_c$, $0 \to -J_c$,$J_c \to 0$,$-J_c \to 0$ vs presentation time T_p. The error bars are standard deviations among the synapses. On top of each plot we indicate the corresponding transition ("–" standing for $-J_c$ and "+" for J_c). Double transitions ($J_c \to -J_c$ and $-J_c \to J_c$) are not shown in the figure. The gradual flattening of the curves reflects the effect of $H- < H >$ in Eq. 3.

Fig. 4 presents the probability distributions obtained for the four possible transitions ($0 \to J_c$, $J_c \to 0$, $-J_c \to 0$, $0 \to -J_c$), as a function of the presentation time.

Fig. 5 presents the dependance on T_p of the average overall transition probability of the synapses with its standard deviation in the synaptic population (error bars). as well as the fraction of synapses that never made a transition. The value $T_p = 10t_0$ ($k = 10$) has all synapses participate in the learning process, while keeping the estimated transition probabilities as small as possible. 10 synapses were excluded because hardware instabilities made them unreliable.

3.3 Input streams of stimuli

To investigate the learning process of `LANN27`, p prototypes (p strings of 27 ± 1's), are generated at random, subject to the constraint that each pair should differ by not less than 12 neuronal states. Three 'protocols' of input patterns sequences were adopted:

- *incremental protocol*: In a first stage only the first prototype is repeatedly presented for learning (here and in the following this means $k = 10$, $T_p = 720 \mu s$). In the second phase the first two prototypes are presented in random permutations, in the third phase three prototypes and so on. The asymptotic regime of this protocol, in which all p prototypes keep being presented, corresponds to a stationary environment.

- *palimpsest protocol*: In a first phase the network is repeatedly presented two prototypes in random permutations for learning. In the next phase only the second and the third prototypes are presented for learning (the first prototype is no longer presented).

- *generalization protocol*: Around each prototype, a class of randomly degraded versions of the prototype is generated. The network is repeatedly presented, from the beginning, a random permutation of degraded versions of all p prototypes.

The three protocols are meant to illustrate how learning is affected by the temporal structure in the stream of input stimuli.

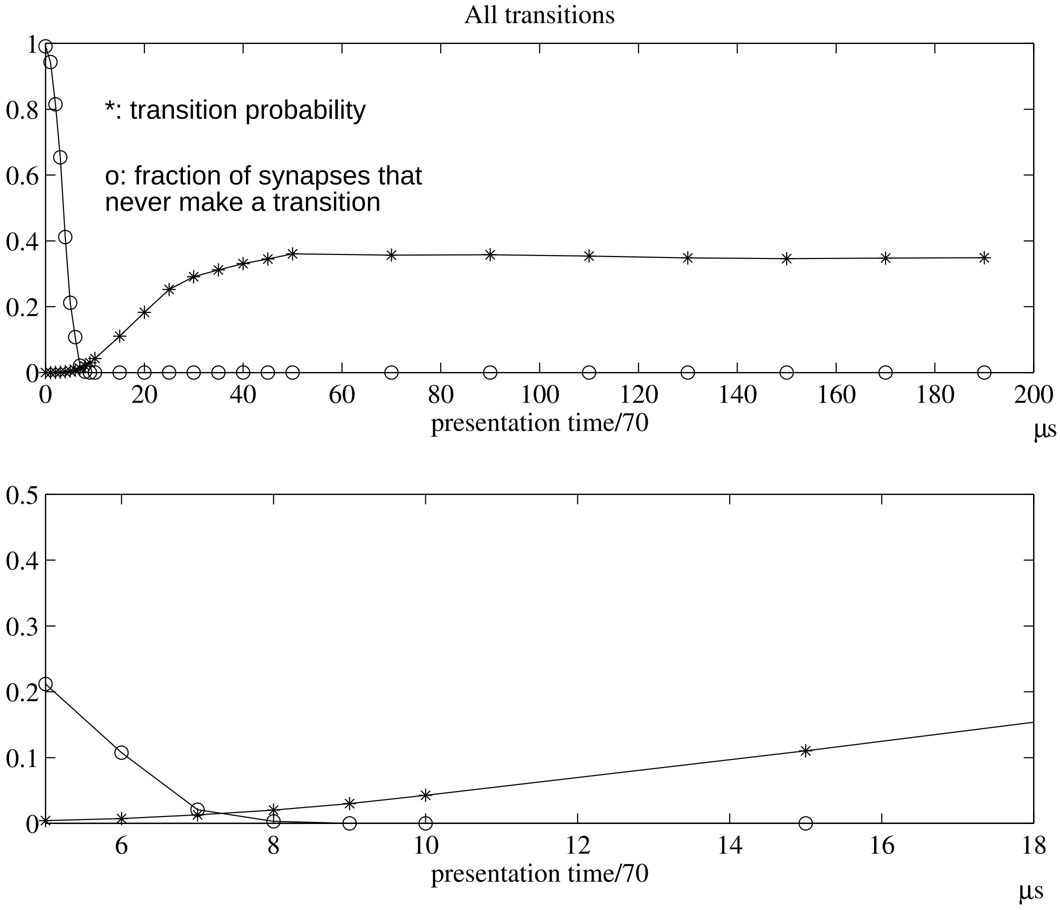

Figure 5: Overall transition probabilities vs presentation time. Top: the curve marked by stars is the average overall transition probability vs T_p at fixed stimulus strength $|H|$; the curve marked by circles is the fraction of synapses that never made a transition vs T_p. Bottom: zoom into the upper picture – the region in which the fraction of synapses that never make a transition goes to zero. For $k = 10$ ($T_p \sim 700\mu s$) all synapses participate in the learning process, while the average transition probability is about 5%. This value of T_p provides the desired compromise.

The *incremental* protocol is meant to expose the effect of crossing the limit of capacity on the attractors structure. Besides, this protocol illustrates the gradual structuring of attractors, as the input from the environment gets richer.

The *palimpsest* protocol is used to explore the dynamics of 'forgetting' in the network. It exposes how the network adapts to changes in the temporal flux of stimuli, forgetting those which are not seen for a long time, under the pressure of new incoming stimuli. This means that previously formed attractors, corresponding to those stimuli, have their basins of attraction gradually shrunk and they finally disappear. The 'palimpsest' property also allows the network to make room for new information, discarding those no longer present in the environment.

The *generalization* protocol has been designed in order to expose how learning brings about a *prototype extraction*, when incoming stimuli are chosen from different sets of patterns which are highly correlated inside each set. Learning in a 'realistic' environment envisages a flux of stimuli in which classes are dynamically defined on the basis of the correlations inside groups of stimuli. Prototype extraction should result as a recognition of these correlations.

Several general issues related to learning in the above conditions are discussed in [8].

The above protocols imply learning experiments in 'controlled' environments, in which only the stimuli belonging to classes to be memorized are repeatedly presented to the network, in fluxes with various statistics. According to our basic expectations, the function of the network and its ability to structure its synaptic matrix according to statistically dominant information appearing in the incoming flux, must be robust to the appearance of random stimuli in the stream of stimuli. We then checked that `LANN27` has this property by producing fluxes of stimuli in which repetitions of prototypes are intermixed with random stimuli. Attractors form well for p=1 with 2 random patterns presented for every presentation of the stimulus to be learned. Also for p=2 and one random pattern for every presentation of the two patterns leads to the formation of good attractors. The vicinity of the limit of capacity (see next Section) makes more elaborate checks difficult.

3.4 Collective behaviour of `LANN27`

3.4.1 Memory capacity

The variety of tests that can be performed is limited by the small size of the network, which implies low memory capacity. To put the discussion of attractor memory capacity in context, we recall the situation for the Hopfield model[15, 1], a paradigmatic example of attractor network. In the limit of an infinite network, as the number of stored patterns p goes above p_c $(= 0.138N)$ there is an abrupt change in the dynamics of the system: above p_c there are no attractors in the vicinity of the memories embedded in the synaptic matrix. The network undergoes a transition to complete "blackout". When the number of neurons (N) is finite, the situation is more complex. For a number of memories just below the limit of capacity, a broad distribution of attractors appears in response to presentations of the memorized patterns. As N is increased, keeping p/N constant, this distribution becomes sharply peaked on the chosen memory. Just above p_c, for low N, the situation is not much different, but the distribution of attractors develops very differently as N increases for fixed p/N, moving towards the blackout condition mentioned above. Thus, for low N, crossing α_c produces a fairly smooth change in the dynamical behaviour of the network, which renders the definition of capacity for small networks quite subtle. (see e.g. [1] Chapter 6, Section 5)

In the present network the number of neurons cannot be varied to detect the memory capacity. We introduced a different criterion to decide whether the network is above or below its limit of capacity: given a learnt pattern ξ, and the attractor s^ξ that the network developed for it, if the patterns which are nearest neighbours (at Hamming distance 1) to ξ, when used as stimuli, make the network relax to the same s^ξ, we will say that the network is below the limit of capacity. The attractor s^ξ can (and in fact sometimes does) differ from ξ. The only requirement is that the attractor coincides for each memory and its nearest neighbours (besides, of course, requiring that for ξ and $\xi' \neq \xi$ one has $s^\xi \neq s^{\xi'}$). The analysis is illustrated in Fig. 6.

Each histogram in the figure is constructed as follows: for a network with p learnt prototypes ($p = 1 \ldots 4$) 200 different sets of p prototypes were generated; for each set the prototypes were presented 1500 times in random permutations, for long enough to induce synaptic modifications (learning session). Following each learning session, the network was presented briefly (for

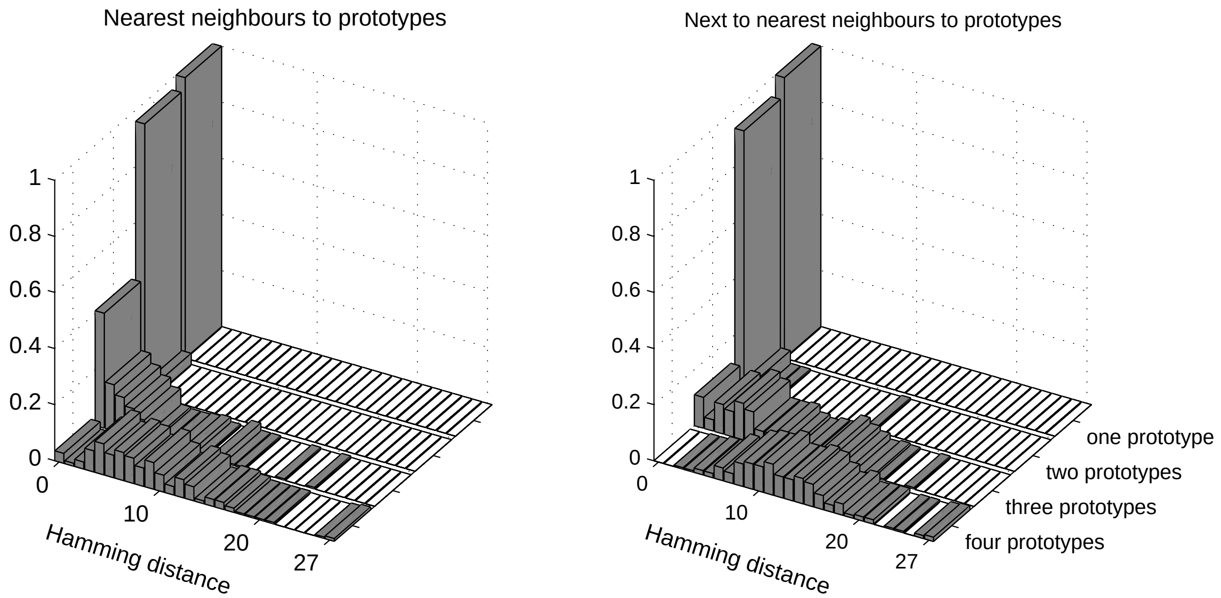

Figure 6: The distribution of Hamming distances of target attractors for stimuli in a sphere (left: Hamming radius 1, right: Hamming radius 2) surrounding the prototypes in state space, for networks with p learnt prototypes.

retrieval) with every stimulus in the sphere of given Hamming radius about each one of the learnt prototypes. The pattern retrieved by the network for each presentation is recorded, and the maximum Hamming distance is measured between the attractor retrieved for the prototype's nearest neighbours and the attractor retrieved when the prototype itself is presented. This is repeated for the p prototypes of each set, and the histogram of these maximum distances for each p is drawn.

The left part of the figure indeed shows that for $p \leq 2$ the prototype and its nearest neighbours share the same attractor for almost all cases. $p = 3$ is 'borderline', in that the above coincidence still holds for about 40% of the cases; for $p = 4$ there is virtually no relation between the prototype's attractor and the attractors of its nearest neighbours. The right part of the figure illustrates the same analysis for the next to nearest neighbours of the prototypes. It is seen that already for $p = 3$ the Hamming distances between the attractors of the next to nearest neighbours and those of the prototypes have a large spread.

It is apparent from the histograms that $p = 3$ is the value of the limit of capacity according to our definition. Given the finite size of the network, fluctuations in its behaviour are possible also below the limit of capacity. An example of this is one case (out of 200) corresponding to the bar at distance 13 in the right histogram for 2 prototypes; one stimulus at distance 2 from one prototype was attracted by an attractor of another prototype.

The following phenomenological description of the structuring of the state space emerges: for low p we have essentially p attractors, which in most cases coincide with the prototypes. As p approaches the limit of capacity, the state space gets increasingly crowded with attractors, which tend to group around the prototypes; each prototype and its neighbours still share the same attractor. Above $p = 3$ the attractors tend to spread over the whole state space.

3.4.2 Development of the basins of attraction

The network is expected to reflect the statistics of the flow of incoming stimuli by dynamically changing the landscape of its state space. At a given time, attractor states will correspond to the statistics of the classes of stimuli dominating the input flow in the recent past. A subsequent stimulus, if presented for a sufficiently long time, may affect the landscape: it may strengthen an existing attractor if it is similar enough to the prototype of one of the learnt classes, or it may cause some of the attractors to slightly move, or new attractors to start forming, depending on the degree of 'novelty' it brings. Such a stimulus provokes "learning", during its presentation, and when removed leads to recall when the network relaxes to one of its available attractors. By contrast, an input stimulus which is presented for a very short time, will not affect the structure of the state space, and will cause the network to relax to one of the available attractors. It leads only to "retrieval". So learning and retrieval regimes are only distinguished by the presentation time of each stimulus. In testing the behaviour of `LANN27`, we found it convenient to group stimuli in the input flow in suitable sequences of 'learning' and 'retrieval' phases, according to the presentation time.

Even for a relatively small network like `LANN27`, it is difficult to monitor the learning process sculpting the attractors and the basins around them. The huge number of possible network states renders the sampling of the space of states difficult. Besides, the high dimensionality of the state space precludes representations of its landscape that preserve topological and metric properties. We propose a graphical approach to capture and describe the development of attractors during the learning process by taking a sequence of snapshots of the state space at different learning stages. This description, for each snapshot, projects the 27-dimensional space of network states onto a set of planes, each corresponding to one pair of prototypes. In Fig. 7 the logical scheme of this graphical representation is depicted, and the procedure is explained below.

At fixed intervals during each learning protocol we perform a retrieval phase: at each Hamming distance, for each prototype, we generate sample patterns to be used as stimuli for the network. They are presented to the network for the minimal presentation time (about $72\mu s$). We explicitly checked that, for this presentation time, the synaptic configuration remains essentially unchanged, by comparing all the synapses before and after each retrieval phase. On average, only about 1% change their state after $\mathcal{O}(10^4)$ presentations. Given the size of the pattern space, it is not explored exhaustively. This space is sampled by assigning a number of patterns, for each of the possible Hamming distances from 0 to 27, to be generated and presented for retrieval to the network. This number is chosen to reflect the combinatorial dependence of the number of possible patterns on their distance from the prototype. For each pattern presented for retrieval, the neuronal configuration to which the network relaxes is recorded.

For each pair of prototypes, we project the state space onto a plane, to expose the structure of the basins of attraction, in the following manner: Two of the prototypes are assigned two points separated by a distance equal to the Hamming distance between them. Each network configuration is assigned a point in the plane whose euclidean distances to the two points representing the prototypes is equal to the corresponding Hamming distance between that configuration and the two prototypes.

Two sequences of 28 circles each are drawn, each sequence centered around one prototype. All the patterns whose representative point lie on a circle of radius r are at fixed Hamming distance r from the prototype represented by the center of the circle.[4] Each intersection of a circle centered around one prototype (of radius r_1) and a circle centered around the other (of radius r_2), represents the set of all network states at Hamming distances r_1 and r_2 from the two prototypes. To each such point corresponds a large number of network states. Around each intersection point we draw a small diamond shaped region, whose sides are circular arcs halfway between each of the intersecting circles and the neighbouring circles. The representative intersection point is therefore at the center of the corresponding diamond. Note that the patterns form a discrete set (the vertices of the 27-dimensional hypercube) hence the points corresponding to the entire set of patterns are well separated in the plane, allowing the introduction of the diamonds. Moreover, there are diamonds which do not correspond to any possible state of the network. These forbidden regions are black in the picture.

Each diamond is assigned a colour to represent the relative number of stimuli corresponding to this point which flowed to the attractors near the two prototypes or elsewhere. This is done by associating to each prototype one of the fundamental colour components (red, green, blue), and assigning to each diamond a colour mixture reflecting the relative frequency with which, when receiving an input pattern corresponding to the diamond, the network relaxed to each one of the attractors. For example, the presence of a bright red diamond means that the vast majority of the corresponding input patterns caused the network to end up in the 'red attractor', while a dim

[4]Note that in this representation the equivalence of Euclidean distances between points on the plane and the Hamming distance between the corresponding patterns holds in general only for distances between the patterns and each of the two prototypes.

green diamond indicates that many times the network relaxed into a state which is far from all the three attractors, and fell into the green one in the other cases.

This colour assignment, illustrated in Fig. 7, encodes in a quantitative way information related to the basins of attraction. In the example given in the figure, red is associated with prototype $\mathtt{P}_1$, blue with $\mathtt{P}_2$ and green with $\mathtt{P}_3$. The pictures describing the projections of the other two pairs of prototypes are constructed in the same way. It is seen in this particular example, that, when a pattern is presented to the network at a distance from $\mathtt{P}_1$ up to half of the course between $\mathtt{P}_1$ and $\mathtt{P}_2$ the network converges to $\mathtt{P}_1$ in most cases, as can be seen in the red-dominated region around prototype $\mathtt{P}_1$, and the same is true for prototype $\mathtt{P}_2$. The region in the picture with a dominating green component describes situations in which the network converges to the third prototype, mostly in the central region where patterns are essentially orthogonal to both prototypes $\mathtt{P}_1$ and $\mathtt{P}_2$. Red or blue-dominated remote areas refer to initial configurations for which the network converges to an "anti-attractor". That means that it converges to a configuration obtained from the attractor corresponding to a prototype by inverting the states of all the neurons. The appearance of such 'anti-attractor' states is as an artifact of the symmetry of the equations governing the network dynamics under the inversion of the states of all neurons.

The sampling strategy may in principle affect the relative strength of the colour components contributing to different regions of the picture. We performed checks using different samplings of the state space, providing evidence that the qualitative picture is essentially unaffected, as long as the combinatorial dependence of the number of patterns on the Hamming distance from the prototypes is reproduced.

As we showed in 3.4.1 the position (and the number) of the attractors can change in time, depending on the value of p. Near the limit of capacity they tend to form fluctuating clouds around the prototypes. Therefore, to characterize the structure of the state space through the above graphical representation, it was decided to keep the pair of reference points in the plane fixed at the prototypes, introducing a 'tolerance' parameter t effectively grouping all attractors inside a sphere of Hamming radius t around the prototypes: given an input pattern, if the distance between the attractor to which the network relaxed, and one of the prototypes $\mathtt{P}_i$ is less than t, it is counted as a convergence to the attractor corresponding to that prototype i.

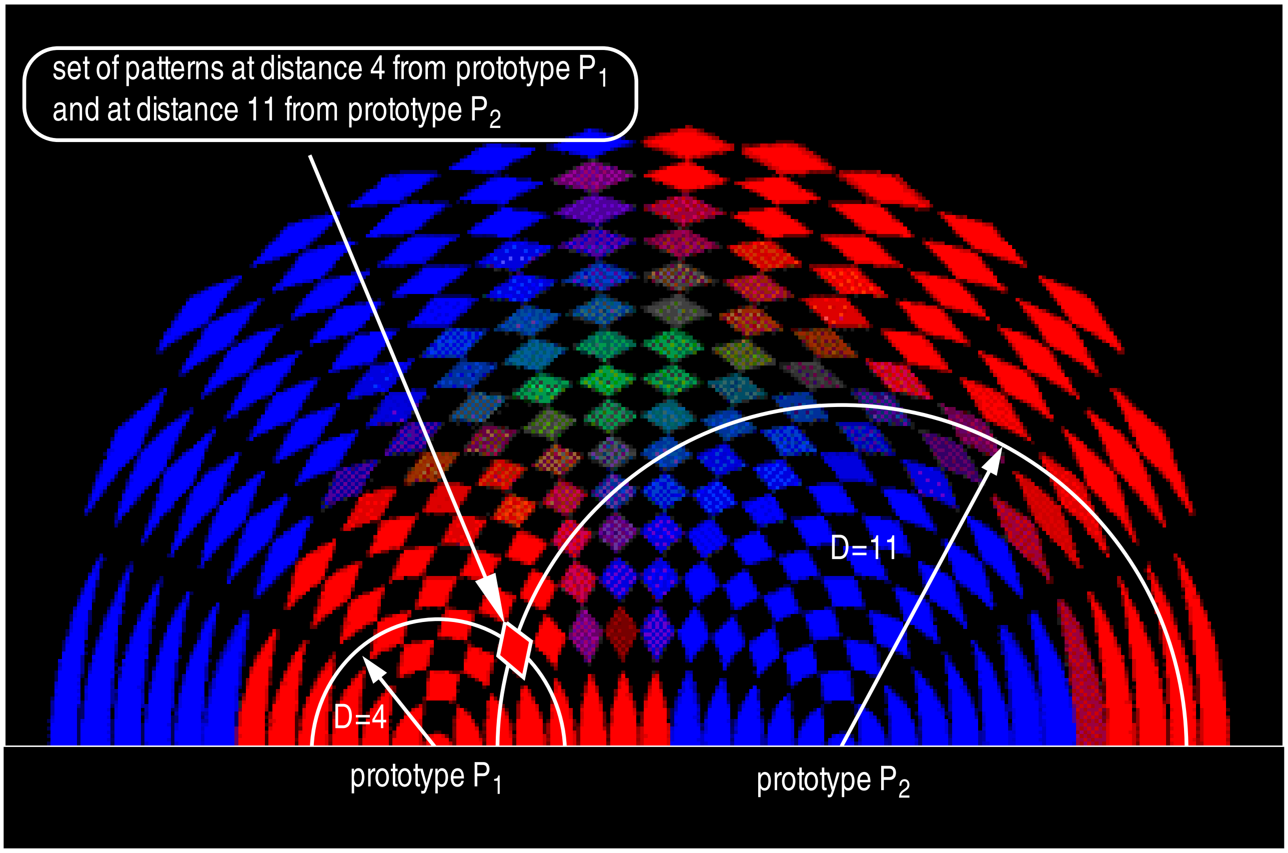

Figure 7: Colour scheme for exploring attractor basins of attraction, by two-dimensional projections of the state space (snapshot). The drawing in white is superposed on an actual sampling of a network that has learnt three prototypes, to highlight the logical scheme. The coloured diamonds crossed by one of the two white circles represent the set of all patterns at fixed Hamming distance from prototypes $\mathtt{P}_1$ or $\mathtt{P}_2$ at its center. For instance, the diamond shaped region drawn in the picture represents the set of patterns at Hamming distance 11 from $\mathtt{P}_1$ and 4 from $\mathtt{P}_2$. Colours are assigned to the diamonds according to the network's response in a retrieval phase, to a large set of stimuli at the two given distances from the two prototypes. For each diamond (that is for each pair of possible Hamming distances from the two prototypes $\mathtt{P}_1$ and $\mathtt{P}_2$) we measure the numbers n_1, n_2 and n_3 of times in which a pattern presented from this region to the network converges to the prototypes $\mathtt{P}_1$, $\mathtt{P}_2$ and $\mathtt{P}_3$, respectively. The number of stimuli for which the network does not converge to any attractor within a fixed tolerance is denoted by z. The box is assigned a colour calculated, in RGB components, as: $\mathtt{R} = n_1/D$, $\mathtt{G} = n_2/D$, $\mathtt{B} = n_3/D$, where $D = n_1 + n_2 + n_3 + z$.

Thus, the graphical representation does not distinguish a situation with only three attractors (each corresponding to one of the prototypes) and situations in which several attractors exist in the tolerance sphere.

Though subjected to the above limitations, the graphical representation in Fig. 7 provides a semi-quantitative description of the basins of attraction, that displays in a readable form their key features, and is valuable in gaining an intuitive understanding of the dynamical process underlying learning.

For $p = 3$, the number of fundamental colours equals the number of prototypes, which makes the colour assignment of the diamonds unambiguous. For $p > 3$, assigning different colours to the prototypes, and mixing them according to the algorithm in Fig. 7 would make the representation degenerate.

An extension to $p > 3$ prototypes can be obtained by replicating the procedure for each pair of prototypes, assigning (with reference to the caption of Fig. 7) the numbers n_1 and n_2 as in the $p = 3$ case, and letting all the other attractors contribute to n_3. Of course this can also be applied to the cases $p \leq 3$, but we preferred to adopt the colour assignment of Fig 7 in order to have the clearest correspondence between prototypes and colours. More compact representations for large p are under study.

Fig. 8 shows a typical evolution of the basins of attraction during learning for the *incremental* (left) and *palimpsest* protocols, with $p = 3$ prototypes. The tolerance parameter is $t = 5$ in both cases. In the incremental protocol, following 5 presentations for learning ($T_p = 720\mu s$) of the 'red' prototype alone, the corresponding attractor starts forming near the prototype and its basin grows larger and larger, eventually covering most of the state space (second row from the top). Then 'red' and 'blue' prototypes (third and fourth row), and later (last two rows) all three prototypes, are presented in random permutations. Learning rearranges the state space to create basins of attraction for new stimuli entering the environment. In the long run the network loses memory of the original bias and the three prototypes have basins of comparable size.

The emerging picture is of a clean and stable asymptotic structure of the state space, even if the network is at its limit of capacity. A parallel inspection of Fig. 6 suggests that at $p = 3$ large basins of attraction may coexist with a number of minor attractors, mostly grouped around each prototype. We will come back to this point later.

We do not show the analogous pictures for the generalization protocol.

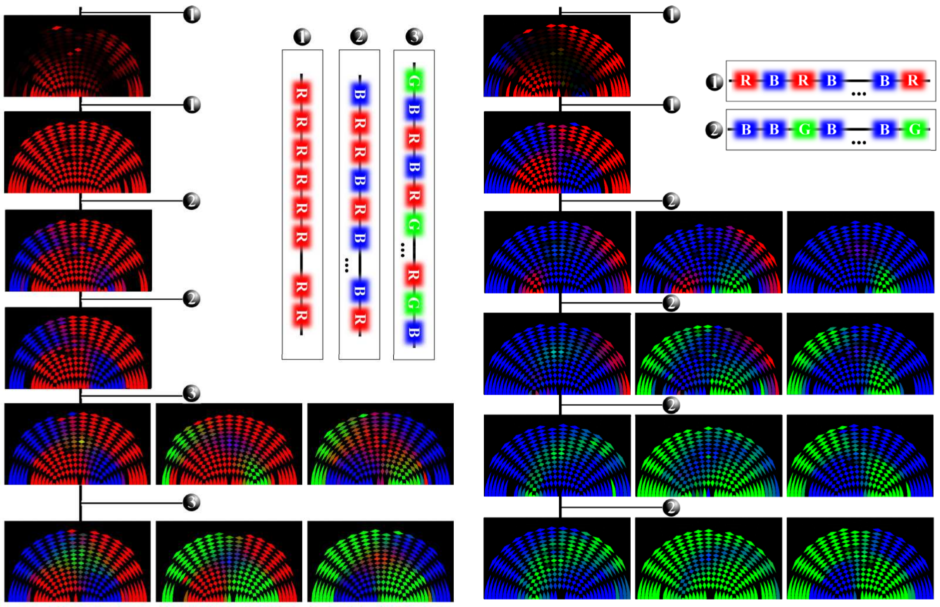

Figure 8: Development of the basins of attraction during learning for the incremental protocol (left) and for the palimpsest protocol (right), $p = 3$. The state space is monitored by performing a retrieval cycle every 5 learning steps, with a total of 30 learning presentations for each of the protocols. Learning proceeds along the vertical direction in the picture, from top to bottom (small spheres in the figure represent the sequence of learning presentations illustrated in the corresponding rectangle).

In this case the basins are 'noisy' versions of the ones observed in the final stage of the incremental protocol, as long as $p \leq 2$. At $p = 3$ the network reaches its limit of capacity, and even a small spread of the stimuli inside each class during learning, destroys the ability of the network to correctly classify them. The 'prototype extraction' ability exhibited by the network for low p is analyzed in the next section.

In the palimpsest protocol, for the first two retrieval cycles only 'red' and 'blue' prototypes are presented 10 times each for learning (Fig. 8, right, two top rows). Then the 'red' prototype is not presented any more, and one more stimulus (the 'green' prototype) enters the environment. One observes, on the right of Fig. 8 (from the third row down) how the 'red' attractor gradually fades away while the novel prototype extends its basin of attraction. Learning made the synaptic structure adapt to the changes in the temporal statistics of the environment: 'old' stored information no longer seen is forgotten, in favour of more and more robust storage of recent stimuli.

3.4.3 Prototype extraction

The *generalization protocol* is meant to expose the *prototype extraction* ability of the network as an associative memory. To quantitatively assess this point, we perform a long learning session[5] (1500 presentations) using as stimuli randomly chosen patterns obtained from each of the p prototypes by inverting the state of one neuron, i.e. each of the p classes of input stimuli – $p = 1 \ldots 4$ – is made of all the 27 nearest neighbours to the corresponding prototype). The p prototypes are never used as learning stimuli. After learning, we perform a retrieval phase, presenting briefly to the network the prototypes and all the $27p$ patterns at Hamming distances 1 from the prototypes that were used for learning, Then we record the attractor to which the network relaxes for each of them. The outcomes are classified in two ways: 1. for each prototype we measure the maximum Hamming distance between the prototype and the attractors reached by its nearest neighbours used as stimuli for retrieval. The maximum of these p distances is recorded. 2. for each prototype we measure the maximum distance between the attractors reached by the neighbours of

[5]Though tens of learning presentations are sufficient in general to structure the state space, we choose here to have a very long learning session to ensure that a truly asymptotic regime had been reached.

the prototype as retrieval stimuli and the attractor reached by the prototype, as in Fig. 6. This procedure is repeated for 100 randomly generated sets of p prototypes. The distribution of these 100 maximal distances is given by the histograms in Fig 9.

From Fig. 9 one sees that for $p = 1$, for essentially all nearest neighbours to the prototype the network relaxes to the prototype itself, despite the fact that the prototype had never been seen during learning: *prototype extraction* occurred. This associative ability of the network is still present with $p = 2$. For about 50% of the sets the two prototypes attract all their nearest neighbours; for 33% of the sets there is at least one attractor different from one of the prototypes. At $p = 3$ the terminal attractors are significantly different from the prototypes, becoming essentially uncorrelated with them at $p = 4$. This, in fact, may be construed as yet another criterion for the memory capacity. The left picture in the figure shows that, for low p, when the configuration attracting the prototype does not coincide with the prototype, still that attractor is shared with most of the prototype's nearest neighbours (all of them for $p = 1$).

3.4.4 Distribution of attractors in state space

As complementary information to that provided by the above analysis, we take a closer look at the details of the distribution of attractor positions. It was pointed out in 3.4.2 that the above graphical representation provides a 'coarse grained' description of the attractor structure of state space: one cannot tell from that representation if a large red region corresponds to a single, big attractor (coincident with or near the red prototype), or it hides a several attractors surrounding the prototype. Moreover, the information concerning the attractors far from the prototypes is completely lost.

At given intervals along the learning sequence, a retrieval cycle is performed as in section 3.4.2. For each input pattern we read the state of activity to which the network relaxed. Each fixed point is added to an 'attractor list', and the number M of times each attractor has been visited is recorded. We also record the overlap m_{max} of each attractor with the nearest prototype. The overlap $m^{\mu\nu}$ between the two patterns ξ^{μ} and ξ^{ν} is defined as: $m^{\mu\nu} = \frac{1}{N}\sum_{i=1}^{N} \xi_i^{\mu}\xi_i^{\nu}$. This is related to the Hamming distance $d^{\mu\nu}$: $m^{\mu\nu} = 1 - (2/N)d^{\mu\nu}$.

Finally each attractor is assigned a label of 'stability': for each input

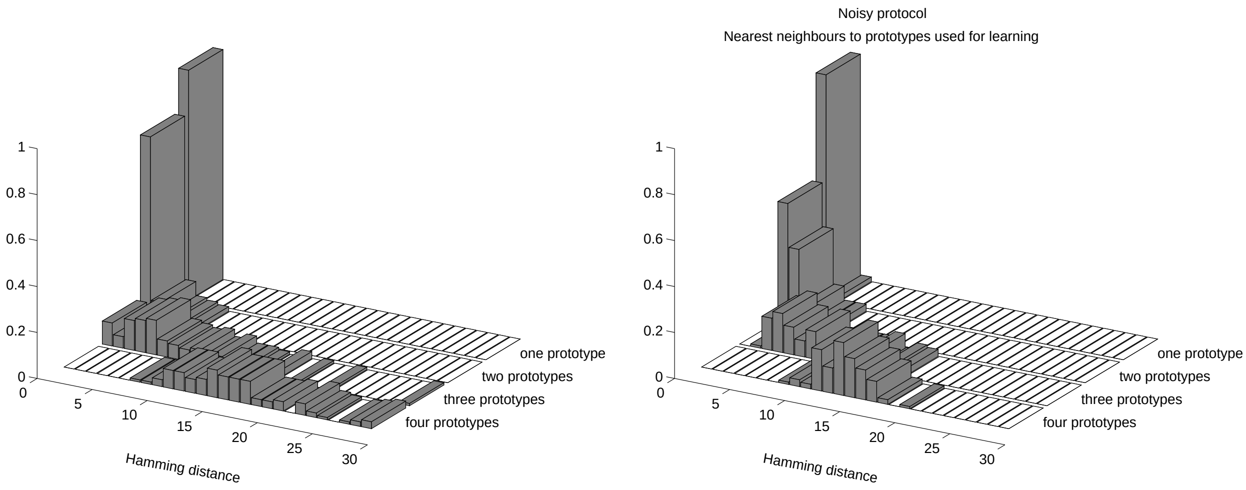

Figure 9: Prototype extraction: Left: the distribution of maximum Hamming distances between target attractors for prototypes' nearest neighbours and the target attractors for the prototypes, following learning with a *generalization* protocol. Right: for the same data, the distribution of maximum Hamming distances between target attractors for prototypes' nearest neighbours and the prototypes.

pattern in the retrieval sequence, the network state is read twice. If for any stimulus leading to the attractor there is a change between the two readings, it is labeled 'unstable'. Otherwise it is 'stable'.

To illustrate the distribution of the attractors in state space, we make the graphical construction described in Fig. 10, taken as an example from an intermediate stage of an incremental learning. We assign each attractor the colour of its nearest prototype. Along the horizontal axis, labeled by the values of m_{max}, we draw a bar of height M for each attractor, colored corresponding to the nearest prototype. Solid if the attractor is 'stable', and dashed if 'unstable'.

Starting from the right, at overlap 1, a large (i.e. frequently visited), stable attractor coincident with the red prototype, a smaller, stable one coincident with the green prototype, then an even smaller attractor whose nearest prototype is the blue one (with overlap ~ 0.8), and so on. The same caveats pertinent to the graphical construction in Fig. 7, regarding the biases possibly induced by the sampling, also apply here, and should be kept in mind when assigning a quantitative value to the plot.

Unstable attractors are particularly frequent during the early stages of learning, when the synaptic structure is essentially uncorrelated with the patterns which are to be learnt, and consequently the input to a neuron is likely to be near zero; in this case, the limited resolution alone of the electronic device implementing the neuron's transfer function can result in fluctuating and unpredictable values at the neuron's output. Furthermore, at $p = 3$ the limit of capacity of the network is crossed, which is an additional source of instability.

In Fig. 11 we show the distribution of attractors for the *incremental*, and *palimpsest* protocols. The four pictures in each figure are snapshots of the distribution of attractors during learning. For the *incremental* protocol (Fig. 11, left), the first picture is taken at the end of the learning phase in which only one prototype is presented for learning, and it shows that in this case the corresponding attractor (together with its anti-attractor) dominates the state space. The next picture shows the final distribution of the two-prototype phase. It is seen that for $p = 2$ the structure of the state space is quite simple, and only two attractors coincident with the two learnt prototypes are present. The mature configuration of the state space for $p = 3$, shown in the third picture, is more complex, and the anticipated multiplicity of attractors shows up. Still, the attractors near the prototypes dominate the

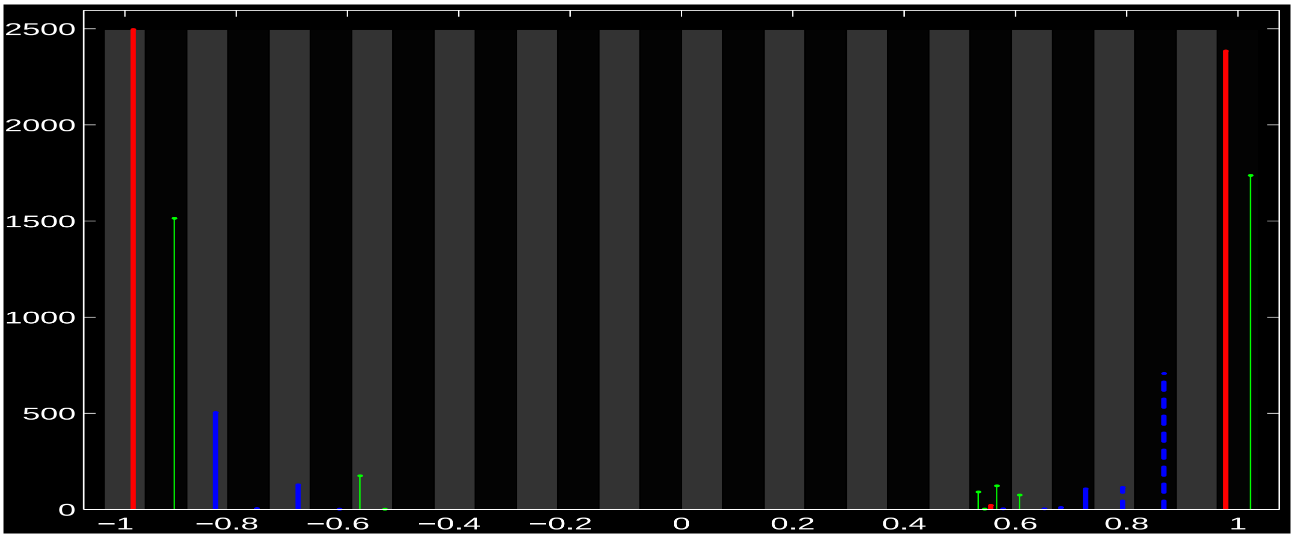

Figure 10: An example of the graphical representation of the distribution of the attractors relative to their nearest prototypes, at a given learning step. The shaded strips of alternate intensity along the overlap axis mark a unit jump in the corresponding Hamming distance. The picture refers to the mature attractor distribution in an incremental protocol, with $p = 3$

state space. The last picture is taken after 1000 further learning presentations of the 3 prototypes. Comparison with the previous picture shows that, as an effect of the saturation of the memory capacity, the attractor distribution keeps fluctuating though the environment is stationary.

Fig. 11 (right) shows the evolution of the attractor distribution for the *palimpsest* protocol which exposes, coherently with the results of section 3.4.2, the dynamics of 'forgetting' in the network: it is seen that, starting from a disordered structure of the state space (first frame), two stable attractors are formed, corresponding to the red and blue prototypes which are learnt in the first phase (second frame). Then, as the red prototype stops being presented and the green one enters the incoming flux, the red attractor is destroyed in favour of the green one (third and fourth frames).

4 Discussion

The network implemented, studied and described in this report is both naive and small. Yet it has been a source of a large number of lessons concerning the implementation, the testing and the representation. Despite all the limitations it is, we believe, the first instance of a fully asynchronous analog device that learns, gradually, in a stochastic unsupervised manner, from a free stream of stimuli and is able to maintain its learned, collective functionality across indefinite intervals of inactivity.

The first limitation is the low memory capacity of the network. It is due both to the low number of neurons and to the coding chosen for the memorized states (50% of neurons active). Yet the actual capacity, related to a rather complex learning dynamics, is captured by theory[11]. Moreover, the expectation of 'palimpsest' behaviour can be verified and is confirmed by the tests. Again, despite the low memory capacity the network is able to create classifying attractors and to extract prototypes, due the gradual learning process for which each stimulus modifies only slightly the synaptic 'engram'.

In general, if the network is dominated by repeatedly presented patterns, similar to each other, over a long time span, the corresponding attractor grows larger and larger, eventually covering the whole space with its basin of attraction. This also determines an 'inertia' in the process of learning new stimuli that enter the flow: new stimuli initially tend to be interpreted

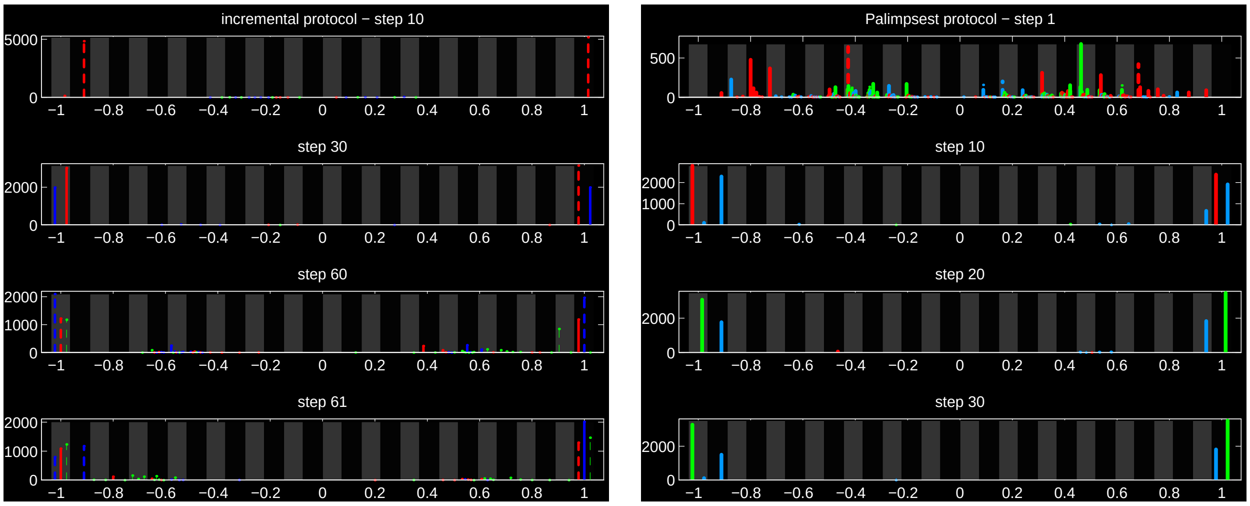

Figure 11: Distribution of attractors for the incremental (left column) and palimpsest (right) protocols. The incremental learning protocol ($p = 3$) is as follows: first prototype presented for 10 steps (we call 'one step' the sequence of learning presentations intervening between two retrieval cycles) then first and second prototypes for 20 steps and finally the three of them for 30 steps. A final retrieval cycle is performed after 1000 further presentations. Palimpsest protocol: distribution of attractors for the three prototypes. The learning protocol is as follows: first and second prototypes presented for 10 steps, then first prototype is not presented any more and the second the third classes are presented for 20 steps.

as ‘exceptions’ to the old ones, making the network change its assumptions about the size of the class of patterns being learnt and moving slightly the attractor, instead of forming a new one. When, upon repeated alternating presentations of old and new stimuli, new attractors form, they have to work hard stealing space from the the basins of old ones. and for a long time their basins stay smaller than those of the old one if patterns belonging to the old class keep being presented. Input flows that are statistically richer determine in general smaller basins of attraction, as an effect of the competition among the different classes (particularly in our case, in which the latter are essentially orthogonal).

The dynamical behaviour exhibited by the `LANN27` is in qualitative agreement with simulations of networks with similar structure and learning dynamics ([8, 12]). What is lacking in those simulations is the detailed distribution of synaptic transition probabilities as is determined by hardware inhomogeneities in our device.

There are many reasons for moving away from the type of implementation described here. What is worth keeping is the general feature of organic learning of attractors from uncontrolled flows of stimuli and the formation of a robust neural dynamics by the learning process. Another lesson to carry along is the difficulty of the testing process, due to the complexity of the developing space of states as well as that of the presentation of the outcome.

A number of features of the present implementation must change either because of implementational considerations or because of computational considerations. The list and the motivations are many and long. We discuss some briefly, as examples, leaving the longer descriptions to future reports.

1. The size of the network implies that any implementation of a larger network must be in VLSI. The natural neural unit in VLSI is the Mead linear integrator[17, 7], rather than the RC integrator.

2. The level of power consumption of the present network is absurd. The solution is spiking neurons and current generators with transistors working in a sub-threshold regime [16].

3. The size of the noise generators on every synapse is larger than the entire neuron. This is unacceptable and unnatural. The unacceptability is related to the fact that as the number of neurons increases the number of synapses increases even faster and the space occupied by the

noisy synapses becomes enormous. It is unnatural because nature has clearly found a way to make synapses so much smaller than neurons. Spikes emitted at random are a natural solution to this problem as well. With spiking neurons one has a reliable, distributed low frequency noise generator.

4. To exploit this noise source, synaptic dynamics should be driven by a folding of pre-synaptic spikes and post-synaptic depolarization, in a given time window. The fluctuations of the number of spikes in the window provide the noise in the synaptic transitions [7].

5. Neurons must be separated into excitatory and inhibitory neurons, which is necessarily non-symmetric. This is Dale's rule, but it is essential for maintaining a stable, low rate, spontaneous activity [3]. That in turn serves as the repository of the distributed noise.

6. Spiking neurons provide a solution also for the problem of the differentiation between three different states: stimulus on, selective activity in attractor and spontaneous activity. This flexibility is resolved at the level of spike rates: in the presence of stimulus some neurons have really high rates; in an attractor some neurons have low-high rates and most neurons have very low (spontaneous) rates, as observed in infero-temporal cortex, for example. This provides a solution to several computational-cognitive problems:

   - It eliminates the need for the term $H - \langle H \rangle$ in the learning source Eq. (4). The spike rates of the neurons are analog variables, whose magnitude describes each of the three dynamical states;
   - It allows a simple separation between learning and retrieval;
   - It allows for a natural distinction between recognized and unfamiliar stimuli presented for retrieval. In the latter case the system relaxes to a state of spontaneous activity, all neurons at very low rates.

Such an implementation is already under development and will be reported elsewhere.

References

[1] D.J. Amit, "Modeling brain function", Cambridge University Press, 1989

[2] D.J. Amit The Hebbian paradigm reintegrated: local reverberations as internal representations, Behavioural and Brain Sciences, 18 617-657 (1995)

[3] D.J. Amit, N. Brunel, "Global spontaneous activity and local structured (learned) delay activity in cortex", *Cerebral Cortex*, **7**, 2, 1996

[4] D.J. Amit and S. Fusi "Learning in neural networks with material synapses", *Neural Computation* **6** 957 (1994)

[5] D.J. Amit, S. Fusi, V. Yakovlev A paradigmatic working memory (attractor) cell in IT cortex , Neural Computation, 9 1101 (1997)

[6] D.J. Amit, H. Gutfreund and H. Sompolinsky, 1987 Statistical mechanics of neural networks near saturation, *Annals of Physics*, **173**, 30

[7] M. Annunziato: Hardware implementation of an attractor neural network with integrate-and-fire neurons and stochastic learning, thesis (in Italian), unpublished (1995)

[8] D. Badoni, S. Bertazzoni, S. Buglioni, G. Salina, D.J. Amit and S. Fusi "Electronic implementation of an analogue neural network with stochastic transitions", *Network*, **6** 125 (1995)

[9] Bliss TVP and Collingridge GL 1993 A synaptic model of memory: long-term potentiation in the hippocampus, *Nature* **361** 31

[10] Brunel N. "Hebbian learning of context in recurrent neural networks, Neural Computation", **8** (1996)

[11] N. Brunel, F. Carusi and S. Fusi "Slow stochastic Hebbian learning of classes of stimuli", submitted

[12] Del Giudice P., Fusi S., "Simulations of a learning attractor neural network", unpublished

[13] S. Fusi, M. Mattia, "Modeling networks with VLSI (linear) integrate-and-fire neurons", submitted

[14] Hebb DO 1949 *The Organization of Behavior* (Wiley, NY)

[15] Hopfield J. 1982 "Neural networks and physical systems with emergent computational abilities", Proc. Natl. Acad. Sci. USA, **79** 2554

[16] Lazzaro J. P. 1992, "Low-power silicon spiking neurons and axons". IEEE International Symposium on Circuits and Systems San Diego, CA.

[17] C. Mead, 1989. "Analog VLSI and Neural System" (Reading, MA: Addison-Wesley)

[18] Miyashita Y and Chang HS 1988 Neuronal correlate of pictorial short-term memory in the primate temporal cortex, *Nature*, **331** 68

[19] Miyashita Y 1988 Neuronal correlate of visual associative long-term memory in the primate temporal cortex, *Nature* **335** 817

[20] Nadal JP, Toulouse G, Changeux JP and Dehaene S, 1986 Networks of formal neurons and memory palimpsests *Europhys. Lett.* **1** 535

[21] Parisi G, 1986 A memory which forgets *J. Phys. A* **19** L617

[22] Sompolinsky H., The theory of neural networks: The Hebb rule and beyond, in L. van Hemmen and I. Morgenstern eds. *Heidelberg Colloquium on Glassy Dynamics* (Springer-Verlag, Heidelberg, 1987)

[23] Williams GV and Goldman-Rakic PS 1995, Modulation of memory fields by dopamine D1 receptors in prefrontal cortex, *Nature* **376** 572

[24] Willshaw D, Buneman O P and Longuet-Higgins H 1969 Non-holographic associative memory, *Nature*, **222** 960