American Journal of Intelligent Systems 2014, 4(5): 159-195
DOI: 10.5923/j.ajis.20140405.01

SHARP (Systolic Hebb - Agnostic Resonance - Perceptron): A Bio-Inspired Spiking Neural Network Model that can be simulated Very Efficiently on Von Neumann Architectures

Luca Marchese

Genova, Italy

Abstract This paper describes a bio-inspired spiking neural network that is proposed as a model of a cortical area network and is tailored to be the brick of a modular framework for building self-organizing neurocognitive networks. This study originated in engineering research on the development of a cortical processor that is efficiently implementable with common digital devices. Thus, the model is presented as a hypothetical candidate for emulating a cortical area network of the cerebral cortex in the human brain. The neuron models are biologically inspired but not biologically plausible.

Keywords Sparse Distributed Representation, Local Cortical Area Network, Cortical Minicolumn, Cortical Macrocolumn, ALs, Hebb, Winner Takes All, Pattern Recognition, Insect Olfactory System, Spiking Neuron, Rank Order Coding, Agnostic Resonance, Restricted Coulomb Energy, Radial Basis Function, Deep Learning, Probabilistic Neural Network, Evidence Accumulation, Pulsed Neural Network, STDP, Cognitive systems, Autonomous Machine Learning, Global Workspace, Concept Cells, Neurocognitive Networks

1. Introduction

In this paper, we propose a spiking neural network model that is inspired by the Drosophila olfactory system and that has both supervised and unsupervised learning capabilities. The philosophy of this research is that everything in the model should be biologically inspired but not necessarily biologically plausible on the basis of current knowledge of the human brain. In the first part of the paper, we analyze the architecture of the network and the behaviors of the neuron typologies that are involved. In the same section, we explain in detail the learning algorithm and some important characteristics of the network.

This study originated in engineering research on the development of a cortical processor that is efficiently implementable with common digital devices. Thus, the model is presented as a hypothetical candidate for emulating a cortical area network of the cerebral cortex in the human brain.

The second part of the paper analyzes the proposed model as an element of more complex systems. Furthermore, the network is analyzed as a brick of a modular framework for building neurocognitive networks.

The third part of the paper is dedicated to the software simulation and the tests. An optimized software simulation demonstrates how this model is very efficiently implemented on Von Neumann computers. In this context, a simulation of the model is analyzed on the XOR problem. Some tests using artificial databases are explained. Then, we make an analysis of the algorithm efficiency on Von Newman computers with considerations about trends in current technology. Finally, there is a brief description of a proposal for the development of a cortical processor that utilizes low-cost commercial digital devices.

1.1. Drosophila Olfactory System

Insects provide a reference point for the study of basic cognitive processes because they have much simpler brains with respect to higher animals but have extremely efficient and adaptive capabilities for reacting and making decisions in the face of complex environmental situations. The enhanced tools that are currently available in insect neurophysiology have made it possible to explore neural signal processing in specific parts of the insect brain. The insect brain areas that are responsible for complex behaviors, such as attention, are the Mushroom Bodies (MBs) and the Lateral Horns (LHs), which constitute principally the olfactory learning system (Fig.1).

* Corresponding author:
luca.marchese@synaptics.org (Luca Marchese)
Published online at http://journal.sapub.org/ajis

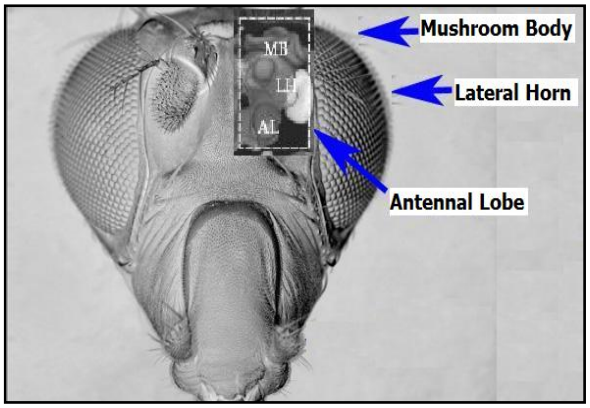

Figure 1. Simulated X-ray view of the Mushroom Body, Lateral Horn and Antennal Lobe, which are parts of the Drosophila olfactory system

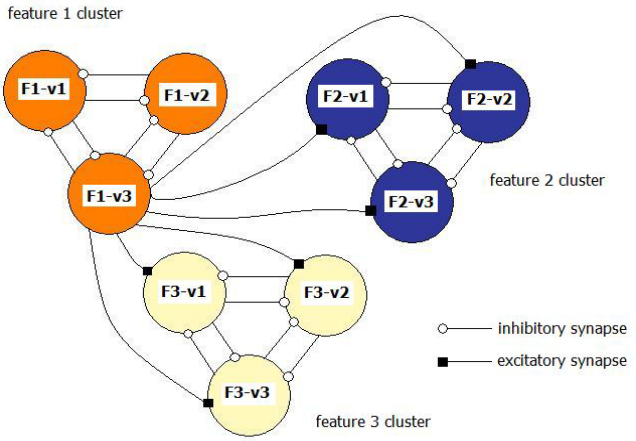

Figure 2. Simplified interconnection scheme of neurons in the Drosophila olfactory system

A neural structure called Antennal Lobes (ALs) generates the spatio-temporal olfactory information that is processed in spatial patterns. The spatio-temporal coding in LHs has been investigated in Nowotny, Rabinovich, Huerta, and Abarbanel [1], where a model for the codification of spatio-temporal patterns into spatial patterns has been presented. ALs is the first layer of the olfactory system in which inputs are decomposed into features, which are represented by feature-specific clusters of neurons that are interconnected in a locally competitive topology [2]. This competitive topology allows for the generation of odor-related patterns of excited and inhibited glomeruli. These patterns will be sent to higher brain structures. The activity of each AL neuron codifies a specific value of a specific feature, while a cluster of neurons represents the feature itself. Thus, any cluster of neurons represents a feature, while any neuron in that cluster represents the value that is assumed by that feature. Neurons in the same cluster are connected together through inhibitory synapses that ensure a WTA (Winner Takes All) behavior inside the same feature cluster. This arrangement means that only one value for a specific feature will be active, and such a behavior can be described as a "discrete quantization" of the input value that originates from the sensor. Neurons in different clusters are linked with plastic synapses that are reinforced when neurons fire together according to a Hebbian rule (Fig.2). These Hebbian synapses between clusters of different features allow the creation of reference prototypes. Recently, Arena, Patanè, Termini [3] have presented an interesting neural network model that was inspired by the drosophila olfactory system. Their model is built around a cluster of Izhikevich spiking neurons [4] that are organized in a layered lattice.

Our model has a similar but not equal topology and different neuron models. Furthermore, we have designed more complex synapses, which are tailored for engineering applications.

1.2. Agnostic Resonance

The first part of the acronym "SHARP" stands for "SystolicHebb" and refers to the learning rule applied to synapses on resonating neurons in a sequential flow. The second part stands for "Agnostic Resonance Perceptron" and refers to the perceptron-like multilayer architecture, where the capability of the neurons in each layer to resonate to a specific value of a feature is "agnostic" because it is not related to previously learned patterns.

We have only to remember the meaning of the word "agnostic".

Thomas Henry Huxley, an English biologist, coined the word "agnostic" in 1860.

Agnosticism can be defined in various ways and is sometimes used to indicate doubt or a skeptical approach to questions. Since Huxley coined the term, many other thinkers have written extensively about agnosticism, and its meaning now has many nuances that are related to religion and philosophy.

We are interested only in the etymology of the word from ancient Greek: Agnostic (Greek: ἀ- a-, without + γνῶσις gnōsis, knowledge). Then, we define "agnostic resonance" as a resonance that is not generated by knowledge.

In the ART (Adaptive Resonance Theory) paradigm, a neuron resonates in response to an input pattern that is similar to its prototype. In our model, this type of knowledge-based resonance does not exist at the level of the neuron but only at the level of the network because the neurons are related to a specific value in a specific feature. In this context, we can speak about "agnostic resonance" because the neurons do not resonate according to the whole pattern (knowledge) but only to a fixed value of a single feature.

The network can be viewed as an ensemble of clusters or in a more classical fashion as a multilayer network in which any layer is related to a single feature. Stimuli are connected directly to each layer.

Any neuron of any layer is connected to any neuron in the subsequent layers with a weighted synapse. Any neuron in any layer is connected to the category layer through weighted synapses.

The category layer is organized in the WTA mode (Fig.4).

In this architecture, the features have been "ordered" in such a way that the neurons that are related to feature f(n) send spikes to the neurons that are related to the features f(n+1), f(n+2), …, up to f(m) (m is the maximum number of features), while the feature f(m) does not send back spikes to the feature f(m-1), and so on. Feed-forward spikes work as "enabling" signals for the subsequent feature neurons. We have defined this behavior as "systolic", which refers to blood flow: this term has been used in past years to define a class of parallel computers in which all of the processors were connected in a sequential chain: systolic systems for computer calculus were defined by professor H.T Kung (Carnegie-Mellon University) in his paper "Why Systolic Architectures?" [32]. A neuron that is associated with the feature f(n) must receive a spike from one neuron in the n-1 previous feature layers to be "enabled" to fire. If a neuron is "enabled" and it is resonating with the input value, it fires, sending a spike to the subsequent layers. The systolic flow and the agnostic behaviors are strictly correlated with the time domain, and the feed-forward spikes in such an architecture carry "evidence accumulation".

In this network, we define a type of neuron that we have called "Integrate, Resonate and Fire" (IRF). This neuron reaches an enabling threshold, integrating spikes that come from resonating neurons in the previous features. Any layer that is associated with a feature is organized into a WTA modality.

The network can be realized with another type of Integrate and Fire (IF) neuron that is composed of two integration sections, which work in Rank Order Coding for the feature input signal and in flat integration for the enabling spike inputs. We have called this neuron Hybrid Integrate and Fire (HIF). The only difference between the two types of neuron is the input decoding modality.

The category neurons are connected to each neuron in each feature layer. We use two different activation functions for the learning activity (synapse update) and for the recognition activity (firing neuron) because the synapse update is proportional to the only resonation level.

The feature neurons are defined as follows:

1) IRF (Integrate - Resonate - Fire):

In this neuron model, the input value is represented by the frequency of the spikes, and any neuron that is associated with a specified feature is resonating at a specific frequency. Due to the undesired resonation of the neurons with near resonating frequency values, all of the neurons that are associated with the same feature are connected among themselves through inhibitory synapses, to obtain a WTA behavior. Any neuron must be triggered by a sequence of spikes that originates from the back-layers, to emit a spike.

SUA (Synapses Update Activation):

$$O_{jnm}^{IRF(U)} = \left(\left|f - f_{Rn}\right| + \varpi\right)^{-1} \times \varepsilon_1 \tag{1}$$

SFA (Spike Fire Activation):

$$\begin{aligned} O_{jmn}^{IRF(F)} &= \left(\zeta^{-1} \times \sum_{0}^{m-1} \zeta_{mn}^{IRF} \times \rho_{mn'} > m\right) \times \\ &\times \left(\left|f - f_{Rn}\right| + \varpi\right) \times \varepsilon_1^{-1} - \left(\sum_{n'=0, n' \neq n}^{N} \zeta_{jn'm}^{WTA}\right) \times \varepsilon_2 \\ &if\left(O^{IRF(F)} > threshold\right)\_then\_(neuron\_fires) \end{aligned} \tag{2}$$

And transforming the logical condition:

$$\begin{aligned} &O_{jmn}^{IRF(F)} = \sum_{0}^{m-1} \zeta_{m}^{IRF} \times \rho_{mn'} + \left(\left|f - f_{Rn}\right| + \varpi\right)^{-1} \times \varepsilon_1 - \\ &- \left(\sum_{n'=0, n' \neq n}^{N} \zeta_{jn'm}^{WTA}\right) \times \varepsilon_2 \\ &if\left(O^{IRF(F)} > threshold\right)\_then\_(neuron\_fires) \\ &j = category\_index \\ &m = feature\_index \\ &n = feature\_value\_index \\ &f = input\_spike\_rate \\ &f_{Rn} = resonation\_rate \\ &\rho = synapse\_weight \\ &\varpi = const \\ &\zeta = action\_potential \\ &\varepsilon_1, \varepsilon_2 = const \end{aligned} \tag{3}$$

2) HIF (Hybrid Integrate – Fire)

This neuron is a Rank Order Integrate and Fire (S. Thorpe); however, it must be enabled to fire by the sequence of spikes that arrive from the back layers. The integration of these spikes together with the weighted integration of the rank-ordered spikes (which represent the feature value) enable the neuron to fire. By adjusting the firing threshold, it is possible to set the generalization when the WTA mechanism is disabled.

SUA (Synapse Update Activation):

$$O_{jmn}^{HIF(U)} = f\left(\Theta\right) \tag{4}$$

SFA (Spike Fire Activation):

$$\begin{aligned} O_{jmn}^{HIF(F)} &= \left(\zeta^{-1} \sum_{0}^{m-1} \zeta_{m}^{HIF} \times \rho_{mn'} > m\right) \\ &\times f\left(\Theta\right) - \left(\sum_{n'=0, n' \neq n}^{N} \zeta_{jn'm}^{WTA}\right) \times \varepsilon_2 \\ &if\left(O^{HIF(F)} > threshold\right)\_then\_(neuron\_fires) \end{aligned} \tag{5}$$

and transforming the logical condition:

$$O_{jmn}^{HIF(F)} = \sum_{0}^{m-1} \zeta_{m'}^{HIF} \times \rho_{mn'} + \varepsilon_1 \times \left(1 + e^{-(f_n(\Theta))}\right)^{-1} - \left(\sum_{n'=0, n' \neq n}^{N} \zeta_{jn'm}^{WTA} \right) \times \varepsilon_2$$

$$if\,(O^{HIF(F)} > threshold)\_then\_(neuron\_fires)$$
$$j = category\_index$$
$$m = feature\_index$$
$$n = feature\_value\_index$$
$$\rho = synapse\_weigth$$
$$\varepsilon_1, \varepsilon_2 = const$$
$$\Theta = rank\_order$$
$$\zeta = action\_potential \quad (6)$$

In the following part of the paper, we use only the IRF model. Furthermore, we introduce in the IRF a recognition control parameter that enables the SHARP neural network to respond to patterns that do not generate a full sequence of systolic spikes. In other words, we want to add to the neural network a generalization capability that is not related to the LSUP (Fig.3) distance that is computed in any feature cluster. The way to add this new behavior is to introduce a new threshold into the SFA in such a way that a feature cluster neuron can be activated by a limited number of spikes. Thus, (3) is updated as follows:

$$if\,(O^{IRF(F)} > \delta\_threshold\ )\_then\_(neuron\_fires) \quad (7)$$

$$\delta\_thresold = threshold - (\delta \times \zeta)$$
$$\delta = absent\_spikes\_number \quad (8)$$

The category cluster neurons' (Figs.5 and 6) activation is defined as

$$O_{jy}^{CC(F)} = \left(\sum_{m=0}^{M} \zeta_m^{IRF} \times \kappa_{yn} \right)$$

$$if\,(O^{CC(F)} > threshold)\_then\_(neuron\_fires)$$
$$\zeta = action\_potential$$
$$\kappa_{yn} \in \chi_{jm} \in \psi = synapse\_weigth \quad (9)$$
$$m = feature\_index$$
$$n = feature\_value\_index$$
$$j = category\_index$$
$$y = cluster\_neuron\_index$$

The category neurons' (Figs.4 and 5) output is defined as

$$O_j^{C(F)} = \sum_{y=0}^{Y} \zeta_{jy}^{CC} - \varepsilon \times \sum_{j'=0, j' \neq j}^{J} \zeta_{j'}^{C}$$
$$y = category\_cluster\_neuron\_index \quad (10)$$
$$j = category\_neuron\_index$$

This formula contains the WTA mechanism between the category neurons (Fig.4). Only one neuron in the category cluster is active, and thus, it is equivalent to consider the output value of the active node.

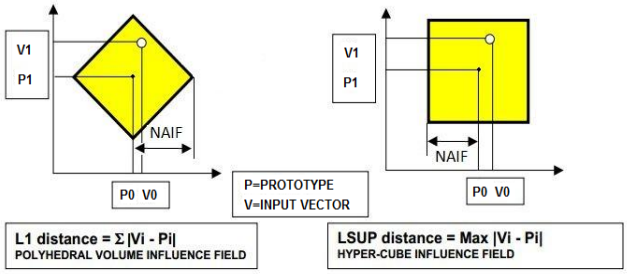

Figure 3. The mathematical concepts of the L1 distance and LSUP distance

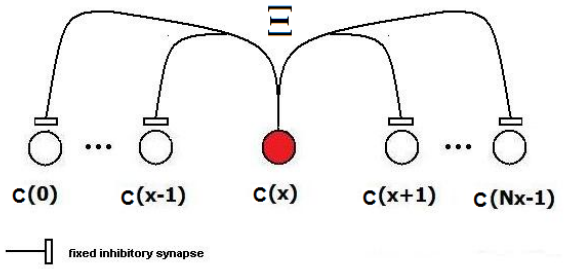

Figure 4. Category neurons are connected among themselves with inhibitory synapses in the same way as neurons inside feature clusters. These inhibitory connections ensure a WTA behavior in the category layer

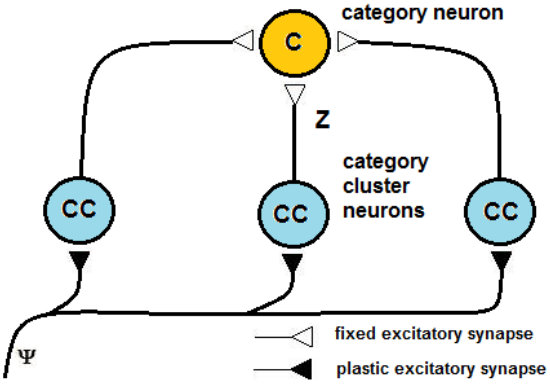

Figure 5. Category cluster neurons work in a daisy chain that is tailored to manage a handover of learning activity, to limit the number of patterns that are learned by a single neuron and thus limit the crosstalk effect

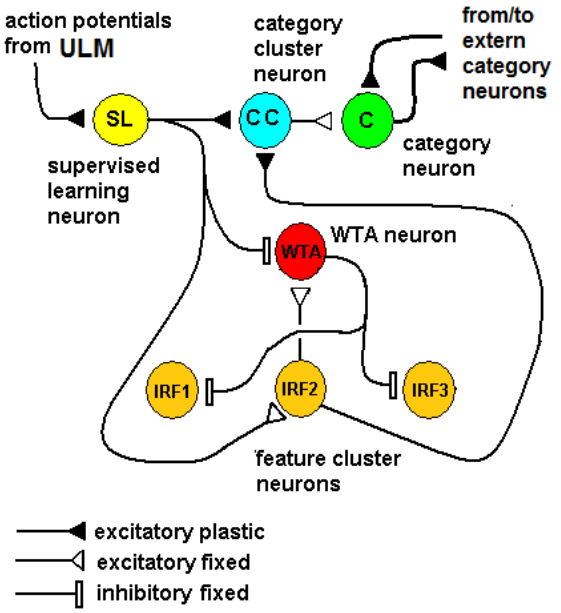

Figure 6. Inside a cluster that is associated with a feature, neurons (that are associated with specific values of such a feature) are fully interconnected through inhibitory neurons (WTA neuron) that are in turn inhibited by the action potential from the category neurons. This mechanism disables the WTA when the category neuron is activated by external action potentials (supervised learning), permitting the nearest neighborhood neurons to fire and STDP to be applied to their synapses. This behavior makes the network able to generalize

2. UNLEASH: Unified Learning by Evidence Accumulation through Systolic Hebb

The learning algorithm that is presented in this paper updates the synapses among the IRF neurons that are associated with specific values of different features when these neurons resonate. The synapses are updated with direct proportionality to the resonation level.

The synapse that connects the resonating neurons with the active category neuron is updated by following the same rule. The learning steps are as follows:

1. In any feature layer, the input frequency makes one neuron maximally resonating and activating. Nearest neighborhood neurons in the same layer resonate and activate also, with a strength that decreases by the distance.
2. The synapses between the spiking neurons in the feature cluster 'n' and all of the spiking neurons in the following feature clusters (n' > n) are updated following the STDP rule. These synapses assume a value that is proportional to the minimum level of resonance between the connected neurons.
3. The synapses between the spiking neurons in the feature clusters and the spiking neuron in the category cluster (activated by action potentials from the SL neuron as in Fig.6) are updated. They assume a value that is proportional to the resonance with the input signal. If the active neuron in the category cluster reaches a number of active afferent synapses that is greater than a specific threshold, it loses the capability to learn and hands over this capability to the next neuron in the cluster chain.

To define mathematically the behaviors that are described in the above steps, we must define mathematically the synaptic structure of the network. The matrix of lateral inhibitory connections between IRF neurons is obtained through a WTA neuron that is associated with any IRF neuron (Fig.6):

$$\Gamma = \begin{bmatrix} \gamma_{11}=0 & \gamma_{12}=\nu & \dots & \gamma_{1n}=\nu \\ \gamma_{21}=\nu & \gamma_{22}=0 & \dots & \gamma_{2n}=\nu \\ \dots & \dots & \dots & \dots \\ \gamma_{n1}=\nu & \gamma_{n2}=\nu & \dots & \gamma_{nn}=0 \end{bmatrix} \quad (11)$$

$$\ni \forall (n \neq n')\gamma_{nn'} = \nu$$

$$\ni \forall (n = n')\gamma_{nn'} = 0$$

Similarly, the matrix of inhibitory connections between category neurons (Fig.4) is the following:

$$\Xi = \begin{bmatrix} \xi_{11}=0 & \xi_{12}=\nu & \dots & \xi_{1n}=\nu \\ \xi_{21}=\nu & \xi_{22}=0 & \dots & \xi_{2n}=\nu \\ \dots & \dots & \dots & \dots \\ \xi_{n1}=\nu & \xi_{n2}=\nu & \dots & \xi_{nn}=0 \end{bmatrix} \quad (12)$$

The matrix of forward excitatory connections (Fig.8a, Fig. 8b and Fig. 9) is defined as follows:

$$\phi = \begin{bmatrix} 0 & \varphi_{12} & \dots & \varphi_{1(m-1)} & \varphi_{1m} \\ \varphi_{21} & 0 & \dots & \varphi_{2(m-1)} & \varphi_{2m} \\ \dots & \dots & 0 & \dots & \dots \\ \varphi_{(m-1)1} & \varphi_{(m-1)2} & \dots & 0 & \varphi_{(m-1)m} \\ \varphi_{m1} & \varphi_{m2} & \dots & \varphi_{m(m-1)} & 0 \end{bmatrix} \quad (13)$$

$$\ni \forall (m = m')\varphi_{mm'} = 0$$

$$m = feature\_index$$

$$\varphi = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2n} \\ \dots & \dots & \dots & \dots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nn} \end{bmatrix} \quad (14)$$

$$n = feature\_value\_index$$

$$\omega = \begin{bmatrix} \rho_1 & \rho_2 & \dots & \rho_j \end{bmatrix} \quad (15)$$

$$j = category\_index$$

The resulting matrix of the forward excitatory connections is

$$\phi = \begin{bmatrix} 0 & \begin{bmatrix} \omega_{11} & \left[\rho_1 \quad \rho_2 \quad \dots \quad \rho_j\right]_{12} & \dots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2n} \\ \dots & \dots & \dots & \dots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nn} \end{bmatrix}_{12} & \dots & \varphi_{1(m-1)} \quad \varphi_{1m} \\ \varphi_{21} & 0 & \dots & \varphi_{2(m-1)} \quad \varphi_{2m} \\ \dots & \dots & 0 & \dots \quad \dots \\ \varphi_{(m-1)1} & \varphi_{(m-1)2} & \dots & 0 \quad \varphi_{(m-1)m} \\ \varphi_{m1} & \varphi_{m2} & \dots & \varphi_{m(m-1)} \quad 0 \end{bmatrix} \tag{16}$$

The bottom-up connections from the feature clusters and category nodes are defined as

$$\psi = \begin{bmatrix} \chi_{11} & \chi_{12} & \dots & \chi_{1m} \\ \chi_{21} & \chi_{22} & \dots & \chi_{2m} \\ \dots & \dots & \dots & \dots \\ \chi_{j1} & \chi_{j2} & \dots & \chi_{jm} \end{bmatrix} \tag{17}$$

$$m = feature\_index$$

$$j = category\_index$$

$$\chi = \begin{bmatrix} \kappa_1 & \kappa_2 & \dots & \kappa_n \end{bmatrix} \tag{18}$$

$$n = feature\_value\_index$$

The resuming matrix of the category node connections is:

$$\psi = \begin{bmatrix} \chi_{11} & \left[\kappa_1 \quad \kappa_2 \quad \dots \quad \kappa_n\right]_{12} & \dots & \chi_{1m} \\ \chi_{21} & \chi_{22} & \dots & \chi_{2m} \\ \dots & \dots & \dots & \dots \\ \chi_{j1} & \chi_{j2} & \dots & \chi_{jm} \end{bmatrix} \tag{19}$$

To limit the crosstalk effect, we have introduced a category cluster that behaves as a connections distributor (Fig.5). If a node in the category cluster has reached the threshold of activated synapses, it forwards the learning capability to the subsequent neuron in the same category cluster. Thus, the matrix of category connections is changed as follows:

$$\chi = \begin{bmatrix} \kappa_{11} & \kappa_{12} & \dots & \kappa_{1n} \\ \kappa_{21} & \kappa_{22} & \dots & \kappa_{2n} \\ \dots & \dots & \dots & \dots \\ \kappa_{y1} & \kappa_{y2} & \dots & \kappa_{yn} \end{bmatrix} \tag{20}$$

$$y = neuron\_index$$

The new resuming matrix of the category node connections is

$$\psi = \begin{bmatrix} \chi_{11} & \begin{bmatrix} \kappa_{11} & \kappa_{12} & \dots & \kappa_{1n} \\ \kappa_{21} & \kappa_{22} & \dots & \kappa_{2n} \\ \dots & \dots & \dots & \dots \\ \kappa_{y1} & \kappa_{y2} & \dots & \kappa_{yn} \end{bmatrix}_{12} & \dots & \chi_{1m} \\ \chi_{21} & \chi_{22} & \dots & \chi_{2m} \\ \dots & \dots & \dots & \dots \\ \chi_{j1} & \chi_{j2} & \dots & \chi_{jm} \end{bmatrix} \tag{21}$$

Before any learning activity, there is only one active neuron in any category cluster. This neuron has *n x m* synaptic inputs and receives a signal from any neuron of any feature cluster in the network. All of the synapses are not active (weight=0). The first learning activity that involves this category activates the synapses that are correlated with the features and the values that are indexed by the input pattern. When the neuron reaches a certain number of activated synapses, it loses the capability to learn new patterns and "delegates" a new neuron for future learning activities. We have labeled this activity as "learning transfer" (Fig.5).

There is another important matrix of synapses that plays an important role in the learning process. For any category of neuron, there is a Supervised Learning Neuron (SL) that is connected, as described in Fig.6. The role of this neuron is to receive an external input and select the cluster that should learn the current input pattern. Furthermore, this neuron inhibits the WTA neuron, disabling the WTA behaviour in the feature clusters.

The top-down connections from the SL neurons to the feature cluster neurons are specular to the bottom-up connections and not plastic. They are defined as

$$\psi^{\Downarrow} = \begin{bmatrix} \chi^{\Downarrow}_{11} & \chi^{\Downarrow}_{12} & \dots & \chi^{\Downarrow}_{1m} \\ \chi^{\Downarrow}_{21} & \chi^{\Downarrow}_{22} & \dots & \chi^{\Downarrow}_{2m} \\ \dots & \dots & \dots & \dots \\ \chi^{\Downarrow}_{j1} & \chi^{\Downarrow}_{j2} & \dots & \chi^{\Downarrow}_{jm} \end{bmatrix} \tag{22}$$

$$m = feature\_index$$

$$j = category\_index$$

$$\chi^{\Downarrow} = \left[\kappa^{\Downarrow}_{1} \quad \kappa^{\Downarrow}_{2} \quad \dots \quad \kappa^{\Downarrow}_{n} \right]$$
$$n = feature\_value\_index \tag{23}$$

The following is the summary matrix of the synapses from the SL neurons to the IRF neurons in the feature clusters. A functional overview of the SHARP neural network is shown in Fig.7 and its representation as a Local Cortical Area Network is shown in Fig. 10.

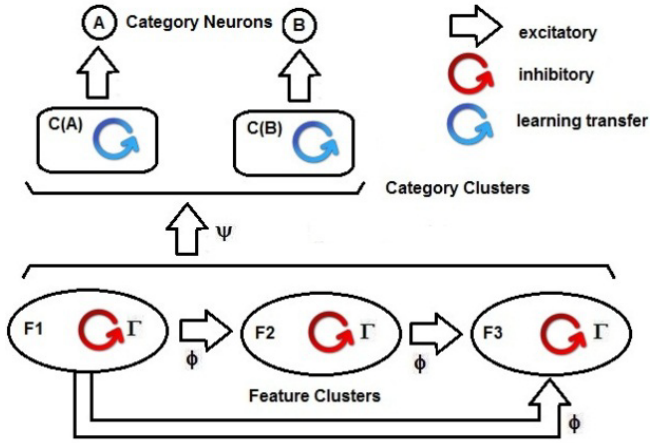

Figure 7. Functional view of the SHARP network. In the lowest side, the clusters that are related to features (F1, F2, F3) are shown, and their forward excitatory interconnections form any F<n> to all of the following F<n+x>. As is indicated, the ensemble of these excitatory connections is the matrix Φ. The red loop arrows inside the clusters indicate the inhibitory connections that are inside any cluster. They constitute the matrix Γ. The matrix ψ of connections between feature clusters and category clusters is indicated by a single green arrow. Category clusters (C_a and C_b) have internal "learning transfer" interconnections that are explained in Fig. 5

$$\psi^{\Downarrow} = \begin{bmatrix} \chi^{\Downarrow}_{11} & \left[\kappa_1 \quad \kappa_2 \quad \dots \quad \kappa_n\right]_{12} & \dots & \chi^{\Downarrow}_{1m} \\ \chi^{\Downarrow}_{21} & \chi^{\Downarrow}_{22} & \dots & \chi^{\Downarrow}_{2m} \\ \dots & \dots & \dots & \dots \\ \chi^{\Downarrow}_{j1} & \chi^{\Downarrow}_{j2} & \dots & \chi^{\Downarrow}_{jm} \end{bmatrix} \tag{24}$$

$$\forall x \in N(\kappa_x = 1) \tag{25}$$

These synapses are normally activated for the purpose of enabling the selection of the appropriate cluster and the inhibition of the WTA behavior when supervised learning is requested. We will show shortly that we must account for their contribution to explain the learning process in terms of STDP (Spike Timing Dependent Plasticity). Now, for simplification, we explain the learning algorithm using the classical Hebb rule and considering only the internal activation value of the neurons (SUA).

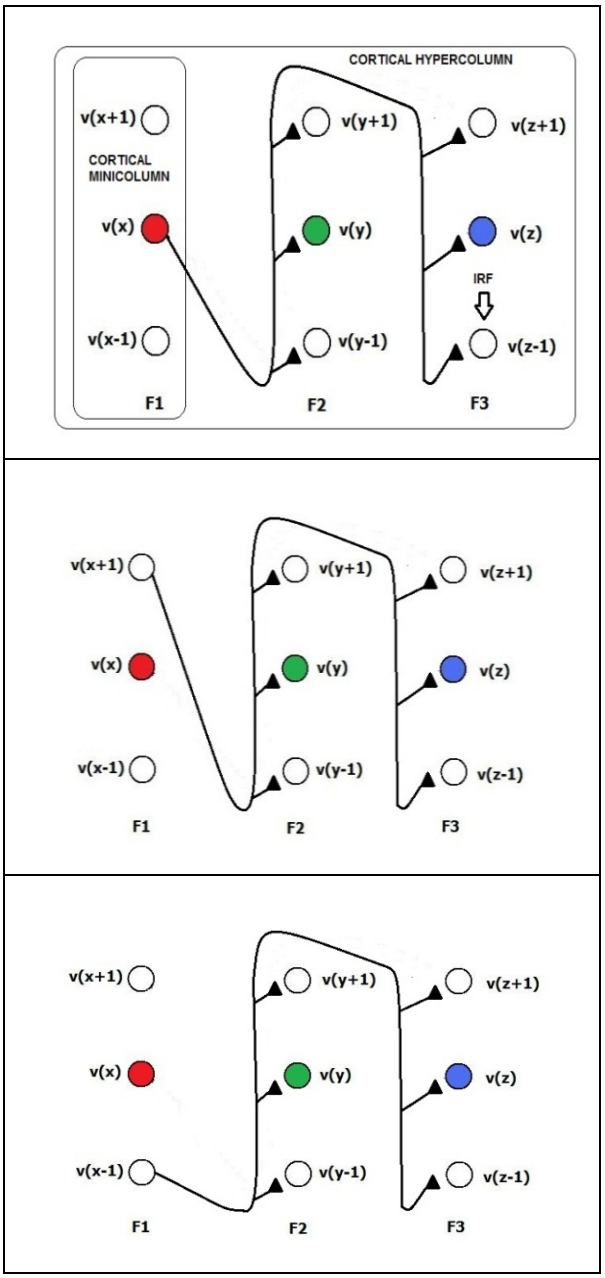

Figure 8a. The excitatory synapses activated from the first layer to the subsequent layers. In the considered case, a pattern that is composed of F1=v(x), F2=v(y) and F3=v(z) is learned. The neurons v(x-1), v(x+1), v(y-1), v(y+1), v(z-1) and v(z+1) are the neighborhood neurons of the primary resonating neuron. They are responsible for the generalization capability of the neural network. The size of the neighborhood area is dependent on the activation (SFA) formula and the firing threshold. In the picture, we have considered a neighborhood of size 1 for simplicity of representation. The activity of the neighborhood is disabled by the intra-cluster inhibitory activity, which is in turn disabled by the inhibitory activity from category neurons for a while when they are activated by action potentials from external activities (supervised learning). At the top of the picture, there is a reference to the possible mapping of the proposed architecture to the cortical hyper-column and mini-column

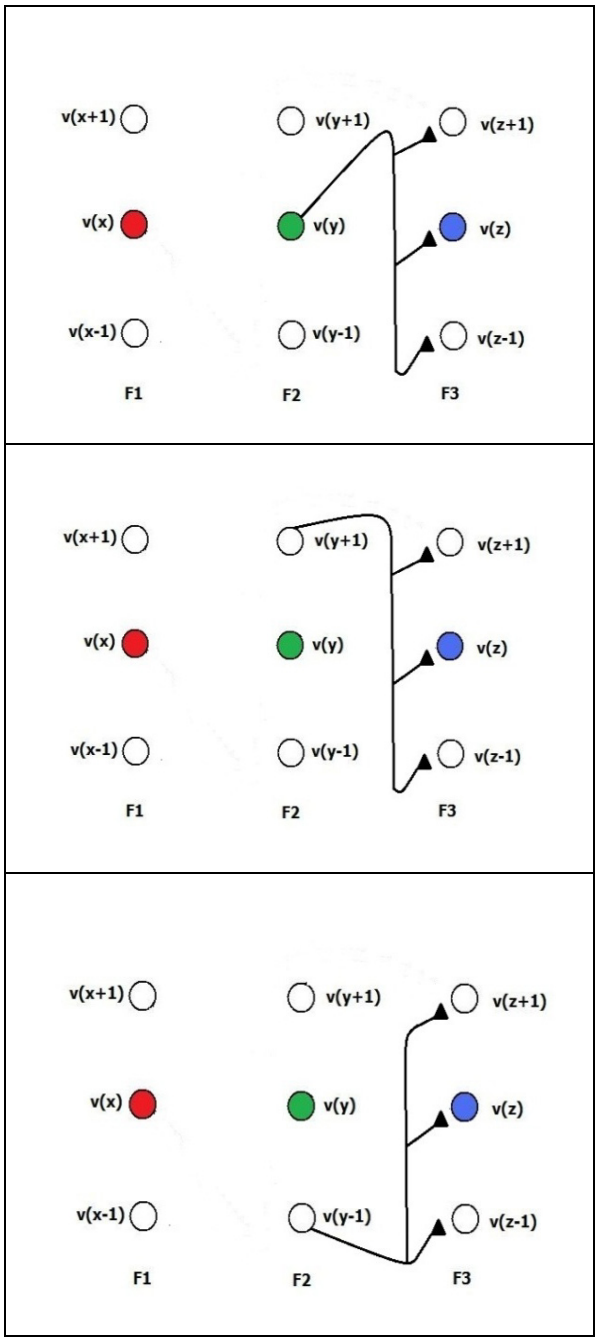

Figure 8b. The excitatory synapses, activated from the second layer to the subsequent layer. In the considered case, a pattern that is composed of F1=v(x), F2=v(y) and F3=v(z) is learned. The neurons v(x-1), v(x+1), v(y-1), v(y+1), v(z-1) and v(z+1) are the neighborhood neurons of the primary resonating neuron. They are responsible for the generalization capability of the neural network. The size of the neighborhood area is dependent on the activation (SFA) formula and the firing threshold. In the picture, we have considered a neighborhood of size 1 for simplicity of representation. The activity of the neighborhood is disabled by the intra-cluster inhibitory activity, which is in turn disabled by the inhibitory activity from the category neurons for a while when they are activated by action potentials from external activities (supervised learning)

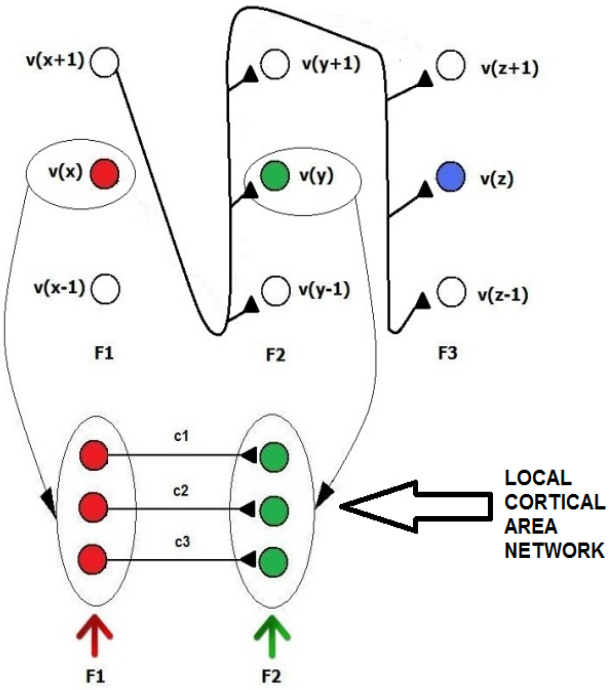

Figure 9. Any neuron in any cluster that is related to a single feature has a clone for any category. The input synapses are replicated, while the axons that are associated with different categories are independent. This ensemble of clone modules should represent a cortical area network

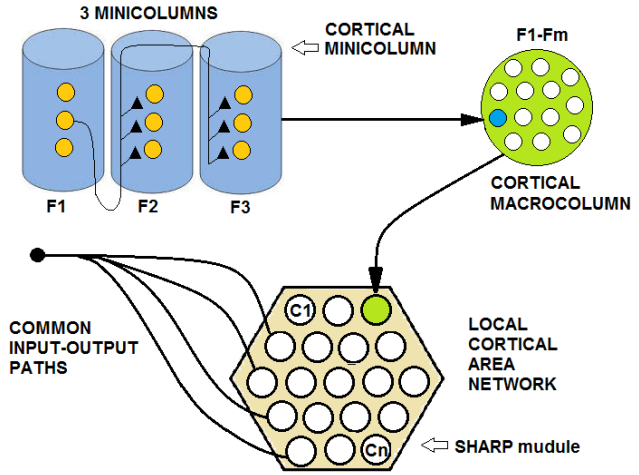

Figure 10. The dense short-range interconnection of a set of macrocolumns in a local area of the cerebral cortex, together with common input and output pathways, constitutes a local cortical area network. A SHARP module should represent a local cortical area network. Any feature cluster is a minicolumn, and the ensemble of all of the feature clusters is a macrocolumn. A macrocolumn identifies a part of a distributed representation of a category or the category itself in the higher level of the hierarchy

At step 2, we apply the Hebb rule:

$$
\begin{aligned}
&\because \left(O_{jnm}^{IRF(U)} \geq \vartheta\right) \wedge \left(O_{j'n'm'}^{IRF(U)} \geq \vartheta\right) \wedge \left(c = j\right) \\
&\therefore \left(\rho_j \in \omega_{nn'} \in \varphi_{mm'} \in \phi\right) = \rho_j \oplus \varepsilon \times \left(\Delta + \left(O_{jnm}^{IRF(U)}\right) \otimes \left(O_{j'n'm'}^{IRF(U)}\right)\right) \\
&j = category\_index \\
&n = feature\_value\_index \\
&m = feature\_index \\
&O = activation\_value \\
&\vartheta = threshold \\
&c = externally\_activated\_category \\
&\varepsilon = learning\_const \\
&\Delta = systolic\_consensus\_offset \\
&\ni \varepsilon \times \Delta \div 2 > \rho_{MAX} \\
&\rho = updated\_synapse \\
&\omega = categories\_matrix \\
&\varphi = values\_matrix \\
&\phi = features\_matrix \\
&< index > / < index' >= pre / post\_synaptic \\
&\otimes = fuzzy\_AND\_operator \\
&\ni \because (a \geq b) \therefore (a \otimes b = b), \because (a < b) \therefore (a \otimes b = a) \\
&\oplus = fuzzy\_OR\_operator \\
&\ni \because (a \geq b) \therefore (a \oplus b = a), \because (a < b) \therefore (a \oplus b = b)
\end{aligned} \tag{26}
$$

The external category is equal to the category index if the category neuron that is associated with the category index is activated by an external signal that is over a threshold. We rewrite (26) as

$$
\begin{aligned}
&\because \left(O_{jnm}^{IRF(U)} \geq \vartheta\right) \wedge \left(O_{j'n'm'}^{IRF(U)} \geq \vartheta\right) \wedge \left(O_{j}^{C(U)} \geq \vartheta\right) \\
&\therefore \left(\rho_j \in \omega_{nn'} \in \varphi_{mm'} \in \phi\right) \\
&= \rho_j \oplus \varepsilon \times \left(\Delta + \left(O_{jnm}^{IRF(U)}\right) \otimes \left(O_{j'n'm'}^{IRF(U)}\right)\right)
\end{aligned} \tag{27}
$$

It is important to note that the weight of a synapse is updated with direct proportionality to the minimum between the activations of the pre and post-synaptic neurons. The hypothesis is that the amount of charge that exceeds the firing threshold plays an important role in the STDP process.

In step 3, the synapses between the resonating neurons and the category cluster neuron are updated.

$$
\kappa'_{yn} = \left(\varepsilon \times \left(\Delta + \left(\left|f - f_{Rn}\right| + \varpi\right)^{-1}\right)\right) \oplus \kappa_{yn} \tag{28}
$$

$$
\begin{aligned}
&\because \left(O_{jnm}^{IRF(U)} \geq \vartheta\right) \wedge \left(O_{jy}^{CC(U)} \geq \vartheta\right) \\
&\therefore \left(\kappa_n \in \chi_{jm} \in \psi\right) = \kappa'_{yn} \\
&O_{jy}^{CC(U)} = neuron\_\_activation \\
&\varepsilon = const \\
&\Delta = systolic\_consensus\_offset \\
&\ni \varepsilon \times \Delta \div 2 > \rho_{MAX} \\
&\kappa_{yn} \in \chi_{jm} \in \psi = synapse\_\_weigth \\
&\kappa_{yn}' = updated\_\_synapse\_\_weight \\
&\oplus = fuzzy\_OR\_operator \\
&\ni \because (a \geq b) \therefore (a \oplus b = a), \because (a < b) \therefore (a \oplus b = b) \\
&j = category\_\_index \\
&y = category\_\_cluster\_\_neuron\_\_index \\
&n = feature\_\_value\_\_index \\
&m = feature\_\_index
\end{aligned} \tag{29}
$$

The learning algorithm builds a synaptic structure that carries evidence accumulation in the recognition phase. The Hebb rule is applied between resonating neurons in different feature clusters. The axons from the feature cluster neurons are directed forward to the next feature clusters. Their role is to buildup a consensus (evidence accumulation) that is mandatory to the recognition process. To reformulate(26) and (27) using STDP (Spike Timing Dependent Plasticity),we introduce an additional behavior to the SL (Supervised Learning) neurons that replaces the action potentials from the back-layer IRF neurons and raises the membrane potential of the resonating IRF neurons to the firing threshold. In this context, the SUA is meaningful only for the computation of the synapse update.

SFA must be reformulated to account for the effect of the action potentials from the SL neurons when the effect of the spikes coming from the previous feature clusters is not sufficient to raise the activation up to the firing threshold due to the low weights of the synapses. Thus, (2) is redefined as follows:

$$O_{jmn}^{IRF(F)} = \left(\zeta^{-1} \times \sum_{t0}^{t1} \zeta^{SL} \geq m \right) \times \left(\left| f - f_{Rn} \right| + \varpi \right)^{-1} \times \varepsilon \quad (30)$$

Accordingly, equation (3) becomes

$$O_{jmn}^{IRF(F)} = \left(\sum_{t0}^{t1} \zeta^{SL} \right) \times \varepsilon_1 + \left(\left| f - f_{Rn} \right| + \varpi \right)^{-1} \times \varepsilon_2 \quad (31)$$
$$t1 - t0 = STF(systolic\_time\_frame)$$

In (31), it can be noted that the (weighted) effect of the action potentials that arrive from the previous feature cluster neurons have been replaced by the (not weighted) effect of a train of action potentials from the SL neuron that has been activated by an external signal (supervised learning). There is no longer the effect of the WTA inhibitory synapses, which is disabled by the same action potentials from the SL. We can rewrite (26) using STDP:

$$\begin{aligned} &\because \left(t_{j'n'm'}^{IRF} - t_{jnm}^{IRF} < STF \right) \wedge \left(\left| t_{jnm}^{IRF} - t_{jy}^{CC} \right| < STF \right) \\ &\therefore \left(\rho_j \in \omega_{nn'} \in \varphi_{mm'} \in \phi \right) \\ &\quad = \rho_j \oplus \varepsilon \times \left(\Delta + \left(O_{jnm}^{IRF(U)} \right) \otimes \left(O_{j'n'm'}^{IRF(U)} \right) \right) \\ &t_{jnm}^{IRF} = presynaptic\_spike\_time \\ &t_{j'n'm'}^{IRF} = postsynaptic\_spike\_time \\ &t_{jy}^{CC} = category\_cluster\_neuron\_spike\_time \\ &STF = systolic\_time\_frame \\ &\varepsilon = learning\_const \end{aligned} \quad (32)$$

Normally, the STDP rule says that the synapse is reinforced if the postsynaptic neuron fires after the presynaptic neuron, within a certain delay. This rule produces inferences about the fact that there is already a dependency of the postsynaptic neuron activation on the action potential that is generated by the presynaptic neuron. The synapse is updated to reinforce the dependency.

In such a context, it is difficult to formulate supervised learning. In this model, the top-down connections that are represented by the matrix $\psi^{\Downarrow}$ (excitatory fixed)enable supervised learning. They transport the spikes from the SL neuron, which are activated by spikes that are generated outside of the network, to neurons in the feature clusters that are associated with the category, enabling them to fire and disabling the WTA behavior.

The synapses between resonating neurons in different clusters can be updated because these neurons fire as a result of the contribution of action potentials from the SL. When these synapses become sufficiently conductive, unsupervised reinforcement learning through STDP can occur without the contribution of the SL neuron (Unified Learning).

Spike timing in the systolic structure of the clusters reflects the sequence of the features. The spike of the category neuron could occur before or after the spike of the postsynaptic IRF neuron but within a range that is limited by the STF (Systolic Time Frame).

STF is the total time that is required for the complete propagation of spikes through all of the feature clusters.

Thus, (29) is reformulated as follows:

$$\because \left(\left| t_{jmn}^{IRF} - t_{jy}^{CC} \right| < STF \right) \therefore \left(\kappa_{yn} \in \chi_{jm} \in \psi \right) = \kappa'_{yn} \quad (33)$$

The SL to CC neurons matrix is composed of plastic synapses:

$$\Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1N} \\ \lambda_{21} & \lambda_{22} & \dots & \lambda_{2N} \\ \dots & \dots & \dots & \dots \\ \lambda_{M1} & \lambda_{M2} & \dots & \lambda_{MN} \end{bmatrix} \quad (34)$$
$$N = SL\_neurons\_number$$
$$M = CC\_neurons\_number$$

M and N grow according to need. These synapses are plastic and have only one chance to change their status from ON to OFF.

During the learning period of a CC neuron, many afferent synapses that transport signals from IRF neurons are activated. In the very last phase of this period, a second CC neuron is committed with the synapses into the status of OFF. The synapse is updated as follows:

$$\begin{aligned} &\lambda = \left(1 + e^{-\alpha(AST - AS)} \right)^{-1} \\ &AST_{IRF} = activated\_synaspes\_threshold \\ &AS_{IRF} = activated\_synapses \end{aligned} \quad (35)$$

These synapses switch almost digitally between an ON status and an OFF status, disabling the capability of learning new patterns through STDP as applied to the synapses

between the CC and IRF neurons.

When the active CC neuron reaches the activated synapses' threshold, a new CC neuron is ready to inherit the capability to learn new patterns. There is a short transition time in which two CC neurons can learn the same patterns.

The matrix of fixed inhibitory synapses from the SL neurons to the WTA neurons replicates the structure of the matrix $\psi^{\Downarrow}$:

$$\mathrm{H}^{\Downarrow}=\begin{bmatrix}\theta^{\Downarrow}_{11} & \theta^{\Downarrow}_{12} & \dots & \theta^{\Downarrow}_{1m}\\ \theta^{\Downarrow}_{21} & \theta^{\Downarrow}_{22} & \dots & \theta^{\Downarrow}_{2m}\\ \dots & \dots & \dots & \dots\\ \theta^{\Downarrow}_{j1} & \theta^{\Downarrow}_{j2} & \dots & \theta^{\Downarrow}_{jm}\end{bmatrix} \tag{36}$$

$$m = feature\_index$$

$$j = category\_index$$

$$\theta^{\Downarrow}=\begin{bmatrix}\pi^{\Downarrow}_{1} & \pi^{\Downarrow}_{2} & \dots & \pi^{\Downarrow}_{n}\end{bmatrix} \tag{37}$$

$$n = feature\_value\_index$$

The summary matrix of the synapses from the SL neurons to the WTA neurons is

$$\mathrm{H}^{\Downarrow}=\begin{bmatrix}\theta^{\Downarrow}_{11} & \left[\pi_1 \;\; \pi_2 \;\; \dots \;\; \pi_n\right]_{12} & \dots & \theta^{\Downarrow}_{1m}\\ \theta^{\Downarrow}_{21} & \theta^{\Downarrow}_{22} & \dots & \theta^{\Downarrow}_{2m}\\ \dots & \dots & \dots & \dots\\ \theta^{\Downarrow}_{j1} & \theta^{\Downarrow}_{j2} & \dots & \theta^{\Downarrow}_{jm}\end{bmatrix} \tag{38}$$

$$\forall x \in N(\pi_x = 1) \tag{39}$$

The synapses between the CC neurons and C neurons are

$$\mathrm{Z}=\begin{bmatrix}\sigma_{11} & \sigma_{21} & \dots & \sigma_{m1}\\ \sigma_{12} & \sigma_{22} & \dots & \sigma_{12}\\ \dots & \dots & \dots & \dots\\ \sigma_{1n} & \sigma_{12} & \dots & \sigma_{mn}\end{bmatrix} \tag{40}$$

$$n = category\_neurons$$

$$m = categories$$

2.1. Neural Phase-Locked-Loop Circuit

There is an important issue that is related to the behavior of the neural network that is presented in this paper. In the definition of the learning algorithm that is based on STDP, we have used the STF parameter as the time that is required to complete a full sequence of spikes through the feature clusters. We have also defined the SDL as the basic time that is required for a spike to pass through a cluster-to-cluster synapse. The SDL is roughly a fraction of STF:

$$SDL \approx STF \div n$$

$$n = number\_of\_features$$

The IF neurons that are used in SHARP do not leak membrane potential as in the more biologically plausible LIF (Leaky Integrate and Fire) model and must be reset at the right time if the neuron is not firing. In the next chapter, a detailed explanation of the neuron models will clarify this concept. The reset of the neurons that remain in a transition state is warranted by a PLL (Phase Locked Loop) circuit that locks on the phase of the action potentials from the neurons of the first feature cluster.

The neural PLL circuit is composed of a chain of neurons, which reflects the systolic sequence of the feature clusters. The last neuron in the chain produces an inhibitory activity on all of the neurons in the network.

This retroaction makes the neural network working as an oscillator with the STF as a period. In this framework, we define local cortical area networks as neural oscillators that behave asynchronously within their own network.

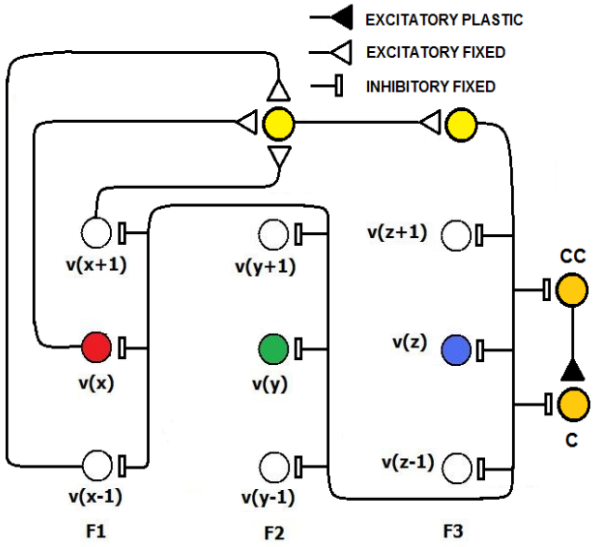

Figure 11. The N-PLL circuit inhibitory activity is triggered by the action potentials that are emitted by the IRF neurons in the first feature cluster. The neurons are IF neurons that have a threshold that can be exceeded with a single spike. Any synapse between PLL neurons adds an SDL delay to the propagation of the signal

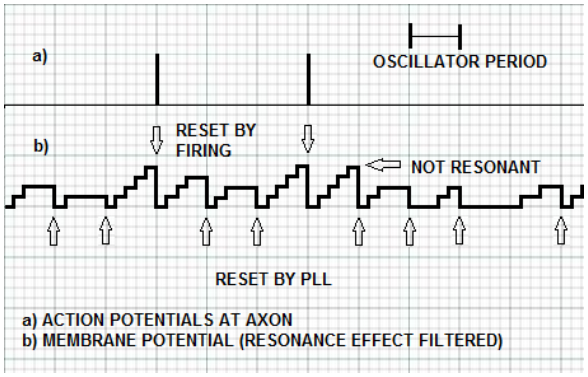

Figure 12. The membrane potential of an IRF neuron that is filtered from the effect of resonance with the input stimulus. A complete firing consensus is reached three times, but only two of those times are synchronized with resonance. When the neuron fires, the membrane potential is automatically reset. When the neuron does not fire, the inhibitory spike from the N-PLL neurons provides the reset of the membrane potential

The synapses in the NPLL circuit are excitatory and inhibitory but fixed. The learning process does not affect at all the behavior of the circuit (Figs.11 and 12). Next, we must define the matrix of synapses that are involved in the NPLL circuit.

The matrix of connections between the NPLL neurons is defined as

$$\begin{aligned} &\mathrm{I} = \left[\imath_{m=0} \quad \imath_{m=1} \quad \ldots \quad \imath_{m=M-1} \quad \imath_{m=M} \right] \ni \forall m(\imath_m = 1) \\ &m = feature\_index \\ &M = feature\_number \end{aligned} \tag{41}$$

The matrix of connections between the NPLL neurons and the IRF neurons is

$$\Sigma = \begin{bmatrix} \mathrm{B}_{11} & \mathrm{B}_{12} & \ldots & \mathrm{B}_{1m} \\ \mathrm{B}_{21} & \mathrm{B}_{22} & \ldots & \mathrm{B}_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ \mathrm{B}_{j1} & \mathrm{B}_{j2} & \ldots & \mathrm{B}_{jm} \end{bmatrix} \tag{42}$$

$$\begin{aligned} &m = feature\_index \\ &j = category\_index \end{aligned}$$

$$\begin{aligned} &\mathrm{B}_{jm} = \left[\beta_1 \quad \beta_2 \quad \ldots \quad \beta_n \right] \\ &n = feature\_value\_index \\ &\ni \forall n(\beta_n = 1) \end{aligned} \tag{43}$$

3. Neuron Models

There are five types of neurons that are involved in the SHARP model:

1. **IRF** (in the feature clusters)
2. **WTA** (in the feature clusters)
3. **C** (the category neurons)
4. **SL** (supervised learning)
5. **CC** (in the category clusters)

In this part of the paper, there is a detailed description of the behavior of any neuron type that is involved in the presented neural network.

3.1. IRF (Integrator-Resonator)

This neuron model is a combination of the integrate-and-fire model and the resonate-and-fire model. In the IRF neuron, the firing threshold can be reached only through the integration of spikes from different dendrites combined with the resonation of a pair of spikes, often referred to as a "*doublet",* on a single dendrite. The resonating part of the neuron behaves in such a way that each pulse alone cannot evoke an action potential, but a pair of spikes can, provided that they have appropriate timing [5][6]. In a classical resonate-and-fire neuron, the appropriate timing of such a doublet can push the internal potential beyond the threshold. In the IRF, the second pulse can do the same only if the internal potential has already reached a certain value due to the integration of the spikes that come from other dendrites. If the neuron reaches the required "offset" of the potential, it can exceed the firing threshold only if the inter-spike interval of the doublet is infinitesimal or if such an interval is equal or very near the eigenperiod that is typical of the neuron [5]. The eigenperiod that is typical of the IRF neuron is strictly correlated with the discrete "feature value" that is associated with the neuron.

$$\left\{ \begin{aligned} &O^{IRF}_{jmn(t1)} = \int_{t0}^{t1} \left(\sum_{0}^{m-1} \zeta^{IRF}_{m} \times \rho_{mn'} - \left(\sum_{n'=0, n' \neq n}^{N} \zeta^{WTA}_{jn'm} \right) \times \varepsilon_1 \right) dt + \Omega \\ &O^{IRF}_{jmn(t1)} = \int_{t0}^{t1} \left(\zeta^{SL} \times \varepsilon_2 \right) dt + \left(\left| f - f_{Rn} \right| + \varpi \right)^{-1} \times \varepsilon_3 \\ &if \left(O^{IRF}_{jmn(t1)} > threshold \right) \_ then \_ \left(P^{IRF}_{jmn(t1)} = \zeta^{IRF}; O^{IRF}_{jmn(t1)} = reset \right) \end{aligned} \right\}$$

$$\begin{aligned} &O = membrane\_potential \\ &P = axon\_potential \\ &\zeta^{IRF} = IRF\_action\_potential \\ &\zeta^{WTA} = WTA\_action\_potential \\ &\zeta^{SL} = SL\_action\_potential \\ &n = feature\_value\_index \\ &m = feature\_index \\ &j = category\_index \\ &\varepsilon_{1,2,3}, \varpi, \Omega = const \\ &t1 - t0 = STF(systolic\_time\_frame) \end{aligned} \tag{44}$$

Fig.14 shows the IF neuron behavior in a state machine. Fig.13 shows the IRF neuron behavior and the four possible situations that are generated by resonation and the "systolic consensus" to fire, which consists of the ensemble of spikes from previous feature clusters. To make the picture readable, the delay between the spikes has been relaxed. In the picture, it appears that the systolic spike train arrives before the spikes of the feature input: this picture shows a simplified representation; in reality, the integration and the resonation are concurrent processes. The spike intervals of the doublets from different inputs are not serial, and the network does not require a total processing time that is equal to the sum of all of the doublets' durations.

Doublets at any feature cluster can occur at the same time, while systolic spikes are mandatorily sequential but have an infinitesimal inter-spike delay that we have called the Systolic Dependency Latency (SDL).

IRF is a mathematical neuron model that is the fusion of two neurons that are found in nature: a resonator and an integrator.

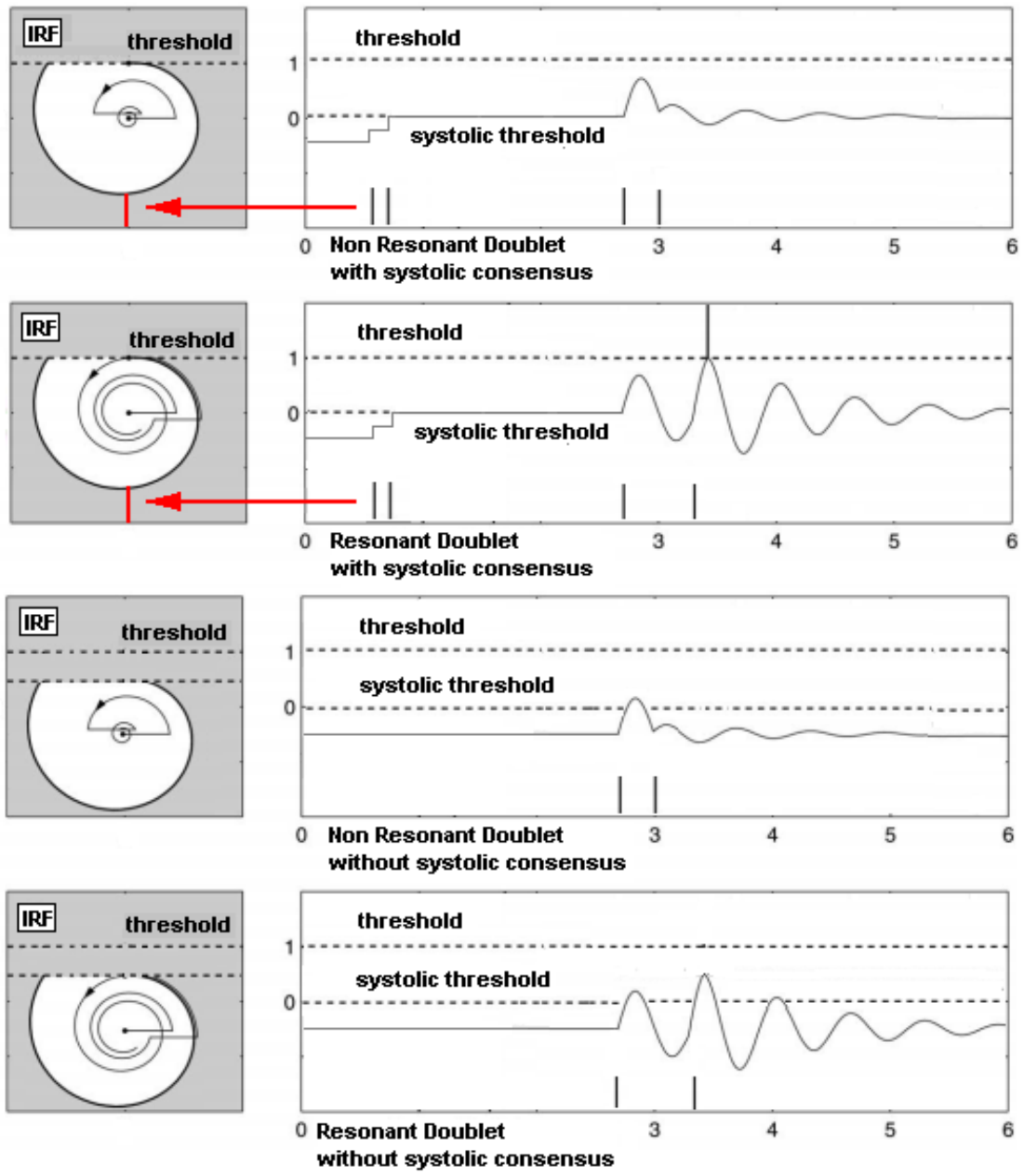

Figure 13. IRF neuron behavior is driven by the systolic spikes that come from previous feature clusters and by the spikes whose temporal interval represents the value of the feature. The picture shows the four possible cases, of which only one can generate an action potential. This example refers to the third feature cluster in the systolic chain (two systolic spikes). The interval between the spikes has been relaxed to make a clearer picture

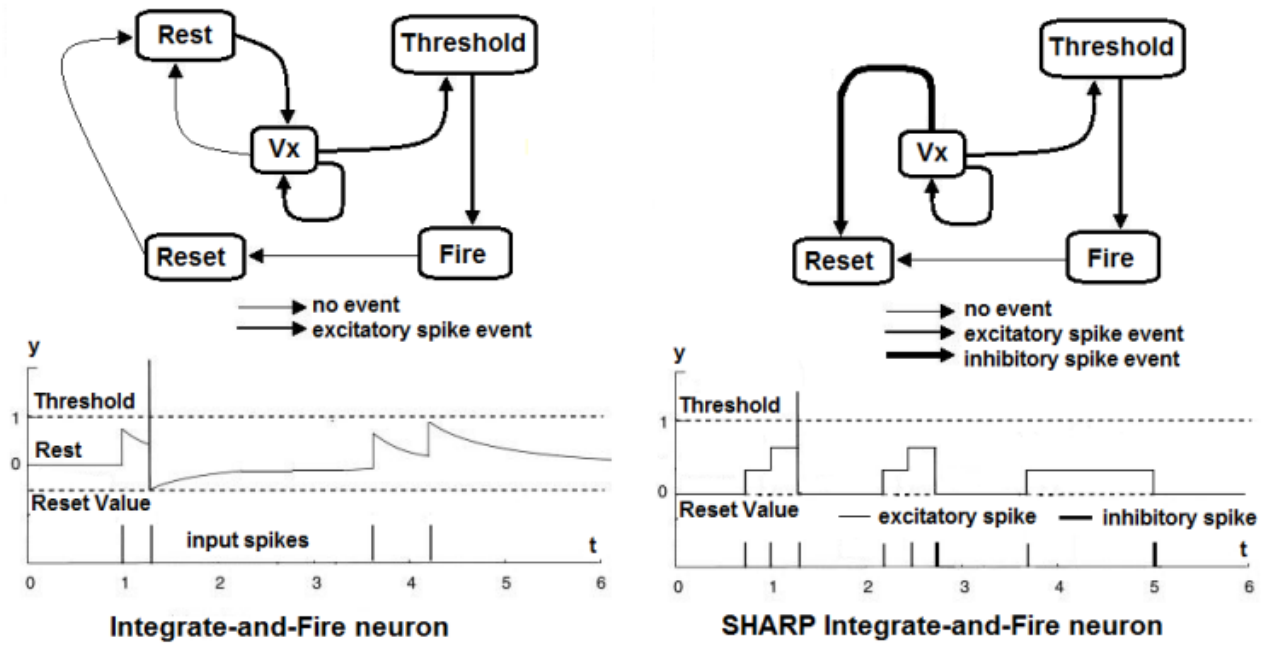

Figure 14. On the left, the more biologically plausible Integrate and Fire neuron behavior. On the right, the simplified Integrate and Fire neuron that is used in SHARP. In the states machine, Vx represents any value that is between Rest/Reset and Threshold

3.2. WTA (Integrator)

This neuron performs the WTA behavior in the feature clusters, and there is one WTA neuron for any IRF neuron. They provide the WTA behavior only when there is not supervised learning. A WTA neuron is excited by the associated IRF neuron, and it is depressed by the activity of the SL neuron in the same category; it works like an integrate-and-fire neuron, but the SL signal is inhibitory.

$$\left\{\begin{array}{l} O^{WTA}_{jmn(t1)} = \int_{t0}^{t1}\left(\zeta^{IRF}_{jmn} - \zeta^{SL}_{jmn}\right)dt \\ if\left(O^{WTA}_{jmn(t1)} > thresh\right)\_then\_\left(P^{WTA}_{jmn(t1)} = \zeta^{WTA}; O^{WTA}_{jmn(t1)} = reset\right) \end{array}\right\}$$
$$O = membrane\_potential$$
$$P = axon\_potential \tag{45}$$
$$\zeta^{WTA} = WTA\_action\_potential$$
$$\zeta^{IRF} = IRF\_action\_potential$$
$$\zeta^{SL} = SL\_action\_potential$$
$$t1 - t0 = systolic\_timeframe$$

3.3. C (Integrator)

This type of neuron is a classical integrator.

Category neurons are activated by the whole sequence of spikes from feature clusters or by the spike train from an SL neuron during supervised learning. In both cases, the spikes pass through the CC neuron that is currently active. The threshold is set in such a way that the membrane potential can reach such a value only when the neuron receives a spike from any feature cluster. If the SL neuron is firing, then the membrane potential can reach the threshold as a result of this contribution. There is an inhibitory activity of all of the other Category neurons that ensures the WTA behavior.

$$\left\{\begin{array}{l} O^{C}_{j(t1)} = \int_{t0}^{t1}\left(\sum_{i=0}^{I}\zeta^{CC}_{jy} - \varepsilon \times \sum_{j'=0, j'\neq j}^{J}\zeta^{C}_{j'}\right)dt \\ if\left(O^{C}_{j(t1)} > thresh\right)\_then\_\left(P^{C}_{j(t1)} = \zeta^{C}; O^{C}_{j(t1)} = reset\right) \end{array}\right\}$$
$$O = membrane\_potential$$
$$P = axon\_potential$$
$$\zeta^{CC} = CC\_action\_potential$$
$$\zeta^{C} = C\_action\_potential$$
$$t1 - t0 = systolic\_timeframe \quad (46)$$

3.4. SL (Integrator)

Supervised Learning neurons receive action potentials from external streams, which provide a supervised learning activity. They send action potential sequences to the IRF neurons in the feature clusters that are associated with the same category. These action potentials replace the spikes that arrive from the back-clusters and enable the resonating neurons to fire, making STDP applicable. SL neurons activate the C neuron, enabling it to fire and making STDP applicable also between IRF resonating neurons and the active CC neuron. The SL neurons deactivate the WTA neurons. In this way, the neighborhood IRF neurons fire, and STDP can be applied, which enables us to obtain the LSUP distance generalization.

The action potentials that are generated by the SL neurons are in the form of bursts. The bursts replace the single spikes that come from the IRF neurons. Thus, the number of spikes in the burst is related to the feature cluster.

$$\left\{\begin{array}{l} O^{SL}_{j(t1)}\int_{t0}^{t1}\left(\sum_{n=0}^{N}\zeta^{C}_{n}\right)dt \\ if\left(O^{SL}_{j(t1)} > thresh\right)\_then\_\left(P^{SL}_{j(t1)} = \zeta^{SL}(burst); O^{SL}_{j(t1)} = reset\right) \end{array}\right\}$$
$$O = membrane\_potential$$
$$P = axon\_potential$$
$$\zeta^{C}_{n} = external\_action\_potential$$
$$n = external\_stream\_index \quad (47)$$

3.5. CC (Weighted-Selective-Integrator)

Category Cluster neurons limit the crosstalk between new and old learned patterns. These neurons have plastic synapses that are connected with axons of IRF neurons. They "share" learning activity with the other CC neurons in the same cluster. CC neurons work in a "daisy chain": when neuron ***n*** reaches the threshold of the activated synapses, it loses the capability to learn and transfers this capability to neuron ***n+1***, and so on.

This neuron has dendrites that are connected to all of the IRF neurons in any feature cluster but has a characteristic threshold of "maximum density" of the activated synapses.

It is possible to see in the formula that the contribution of the spikes from the IRF neurons is weighted. The characteristic behavior of this neuron is concentrated in the role of synapse λ, which enables or disables the influence of the SL neuron depending on the amount of past plastic activity of the synapses. If the density of the activated synapses (k>0) exceeds the threshold, then the action potential from the SL neuron cannot trigger learning. Here, λ is the excitatory synapse between the SL neuron and the CC neuron, and its plastic activity is almost on-off; it is driven by an integrative process of the plastic activity of the synapses that are connected to the IRF neurons.

$$\left\{\begin{array}{l} O_{jy(t1)}^{CC} = \int_{t0}^{t1}\left(\sum_{0}^{M}\left(\zeta_{mn}^{IRF}\times\kappa_{yn}\right)+\lambda_y\times\zeta^{SL}\right)dt \\ if\left(O_{jy(t1)}^{CC} > thresh\right)\_then\_\left(P_{jy(t1)t}^{CC}=\zeta^{CC};O_{jy(t1)}^{CC}=reset\right) \end{array}\right\}$$
$$O = membrane\_potential$$
$$P = axon\_potential$$
$$\zeta^{IRF} = IRF\_action\_potential$$
$$\zeta^{SL} = SL\_action\_potential$$
$$n = feature\_value\_index$$
$$m = feature\_index$$
$$\kappa_{yn}\in\chi_{jm}\in\psi = synapse\_weigth$$
$$\lambda = synapse\_weigth$$
$$\because (y=0)\vee((AST_y > AS_y)\wedge(AST_{(y-1)} = AS_{(y-1)}))$$
$$\therefore \lambda_y = 1(otherwise\_\lambda = 0)$$
$$AST = activated\_synaspes\_threshold$$
$$AS = activated\_synapses$$
$$y = cluster\_active\_neuron\_index$$
(48)

The logical condition refers to the "learning transfer" behavior: the cluster neuron can be activated if the threshold that is related to the number of activated synapses is not reached. In such a case, the activation and learning activity are transferred to the subsequent neuron in the chain. In the logical condition, AS is a value that is proportional to the number of input synapses that are activated, and AST is the threshold that triggers the transfer to the subsequent neuron in the chain.

4. Pattern Recognition: Generalization through LSUP and LNUM

The behavior of the network, when no category neuron is activated by an external signal, is a pattern recognition activity.

When a pattern is received as input, one neuron in any layer resonates and activates. In this phase, a neuron reaches a full activation state only if it has received a spike from a single neuron in any previous layer. Spikes from previous layers arrive to a resonating neuron if the synapses between the resonating neurons in any layer have been activated in the learning phase. This circumstance means that the input pattern or a similar pattern (inside the maximum LSUP distance) has been previously learned. The recognition is positive if any resonating neuron receives an "enabling" signal (a spike) from any previous layer and if at least one category neuron has conductive synapses with all of the activating neurons. The recognition strength can be evaluated in two different ways:

- the first way is to evaluate the minimal synapse strength with a fuzzy-AND operator
- the second way is to evaluate the average of the synapse strengths

More than one category neuron can be excited. WTA connections in the category layer guarantee that only one category neuron can be activated.

Computation of the recognition strength with the fuzzy-AND operator:

$$S_j = \left|\otimes\right|_{m=0}^{M}\kappa\in\chi_{jm} \tag{49}$$

Computation of the recognition strength, averaging the synapse values:

$$S_j = \left(\sum_{m=0}^{M}\kappa\in\chi_{jm}\right)\times M^{-1}$$
$$S = recognit\_strength$$
$$\kappa = synapse\_weight$$
$$m = feature\_index$$
$$M = features\_number$$
$$j = category\_index$$
$$\left|\otimes\right|_x^y = vector\_fuzzy\_AND\_operator$$
$$\ni (\because \alpha_1\le\alpha_2)\therefore\left(\left|\otimes\right|_{i=1}^{2}\alpha_i = \alpha_1\right)$$
(50)

LNUM is an adjustment of the firing thresholds of IRF neurons and CC neurons that enables them to fire if they receive LNUM spikes less than the number that is required.

In this way, a pattern can be recognized if up to LNUM features are extremely noisy (over the maximum LSUP distance). The generalization that is provided on the LSUP distance is determined during the learning phase. The generalization that is provided to the LNUM limit could be adjusted during a recognition phase, although this behavior is useful in an engineering application but is not biologically plausible.

5. Plasticity Driven by Class Specialization

The number of inputs or features is entirely managed by the category nodes, and a new category node that is connected only to a subset of feature clusters can be added. This new category node must have a congruently limited activation threshold. In the same way, new feature clusters and related category nodes can be added. SHARP can grow in the number of features, and new category nodes represent concepts that are specializations of existing categories.

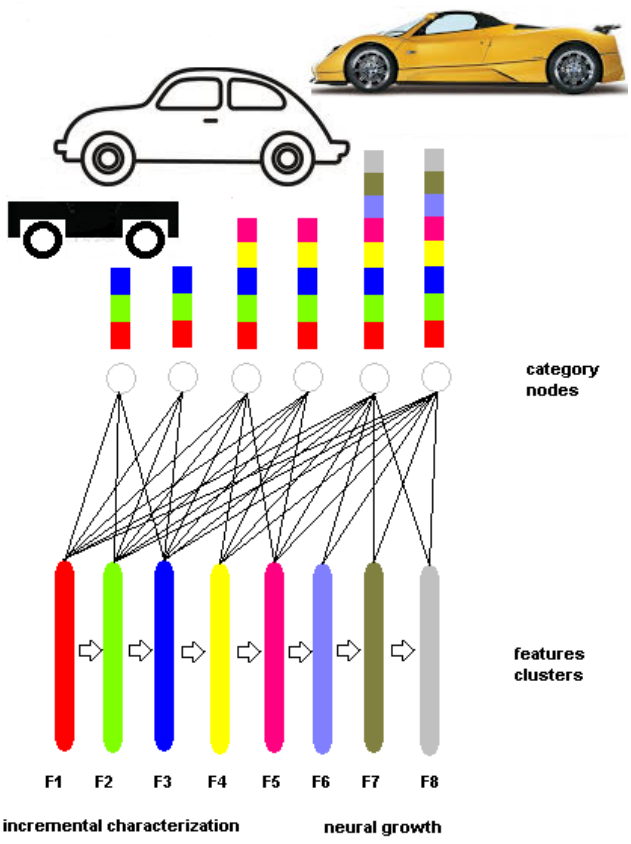

Figure 15. A very simplified example of PCS. The example does not account for the fact that the concept is in reality built by the cooperation and/or hierarchy of multiple streams

The fundamental aspect of this behavior is that plasticity is realized in three ways:

- Synapse updating
- New category nodes
- New features

The third modality is the "plasticity by class specialization" that is shown in Fig.15. This type of plasticity, which is performed with the creation of new neurons, could be triggered in the brain by detecting the crosstalk effect between learned patterns. Such a behavior would be the equivalent of the mismatch condition in the Adaptive Resonance Theory, but the probability of mismatch is forecasted on the grounds of measured crosstalk. This mechanism must be enabled by another process that is capable of detecting the availability of additional discriminant features. We are currently working on the algorithms that could implement these two behaviors.

6. Do Spikes add Entropy in this Architecture? With which Coding Scheme?

SHARP is an NN whose neurons communicate through spikes. We have analyzed neuron models that can manage appropriately spikes that are received at their dendrites, computing the possibility to fire a spike on their axon. We have built such a framework while moved by the desire to mimic biological brain behavior. Can we say that this neural network is a third generation model? To answer this question, we must clarify some concepts.

We normally classify second generation neural networks as those neural paradigms that use analog values to encode information. Neural networks of the third generation use spikes to encode the information that is transmitted between the neurons. Any neural network of the second generation can be transformed from a spiking model by representing analog values for the firing rates of the spikes. Thus, we generally refer to the third generation only for those NNs where the timing of a single spike is meaningful.

A number of studies have investigated the computational efficiency of the NN based on the spiking neurons. Maass demonstrated that a NN that is based on spiking neurons holds more computational power than a NN of the first and second generation. Spiking NNs have been identified as universal approximators because they can simulate NNs of the second generation [7], which are universal approximators themselves. In fact, Maass formulated and demonstrated these theorems using a precise definition of spiking neural networks [8]:

- A finite set V of spiking neurons.
- A set V_in and a set V_out of input and output neurons.
- A set of synapses that cannot be incoming to input neurons
- A weight *w* for each synapse
- A response function $\mathfrak{R}+ \Rightarrow \mathfrak{R}+$
- A threshold function $\mathfrak{R}+ \Rightarrow \mathfrak{R}+$
- A delay $\mathfrak{D}$ for each synapse

In spiking NNs, the rate-versus-timing debate has been framed as a question of whether all of the information about

the stimulus is carried by the firing rate (or by the spike count in some specified time window) or whether the timing of individual spikes within this window also correlates with the variations in the stimulus [9]. This debate cannot have a single answer because spikes can play different roles in different contexts and thus they can encode information in multiple appropriate manners.

The basic consideration is that spikes are, probably, the most appropriate method for transmitting information within biological matter, and we use the term "appropriate" while referring to physical issues and not to information theory. From the point of view of information theory, Maass demonstrated that the spiking neural code can carry more entropy.

Entropy, as defined in thermodynamics and statistical mechanics, is a measure of "variability" or "available information".

We can define, mathematically, entropy as the logarithm of the number of possible states that the system can assume:

$$\begin{aligned} E &= -k\sum_{i=1}^{I} p_i \log(p_i) \\ E &= entropy \\ p_i &= prob\_of\_i^{th} \\ k &= const \\ I &= number\_of\_states \end{aligned} \tag{51}$$

A spike-based information system introduces time as a new variable to encode information. More or less complex integration processes are a baseline for elaborating the information that is carried by spikes. The introduction of time as a new variable increases the entropy that the system can manage if the timing of the single spikes is meaningful but is not encoded by the firing rate. Considering the typical response times to stimuli, it is clear that the firing rate coding scheme does not carry enough information to be a plausible method to encode a world representation in real time.

Independent of the way that sensory neurons use to decode external stimuli with spike trains, we must understand whether it is useful to increase the entropy that is managed by the neurons in the deeper stages of information processing. We know that maximizing the information that is carried by the neural code (i.e, the single spike timing), we can minimize the neural structure (i.e., the neurons, synapses) and manage the same quantity of information. The question is simple: what is better? There is not a Boolean answer, but there are advantages and leaks in both of the solutions. We make a hypothesis with respect to the method that is chosen by nature.

We note that minimizing the information that is carried by the neural code, we can minimize the intelligence of the elementary processing units (i.e., the neurons), and we need more of them to process a certain amount of information and to perform a specific task. On the other hand, when maximizing the information that is carried by the neural code, we need smarter elementary processing units, and fewer units will be required to perform the same task.

At an initial glance, it appears to be better to have a more complex neural code that can reduce the number of neurons that are required to perform a task. The same characteristic that we see as the strength of such a choice is, from another point of view, the main limitation. The more we put intelligence into the elementary processing units, the more we lose in terms of the robustness of the system: a perturbation in the neural code as well as a fault in a processing element become important in the global stream that performs the task. In other words, by putting intelligence into the code and into the elementary processing units, we move away from the principles of connectionism. Fortunately, this dilemma is not Boolean but fuzzy: the correct answer must be sought for, and there could be an optimal point of equilibrium that could be, most likely, different in any component of a complex processing system.

The neurons in the human brain are of multiple types, and they behave differently if assigned to perform different tasks. Specific types of neurons could have different levels of "intelligence" and encode spikes with different schemes.

Much information has been experimentally acquired on sensory neurons because it is easier to explore the neural code that is generated as a response to recordable and measurable external stimuli. In such a case, it is easier to make a hypothesis with respect to the coding scheme having a method for verifying and measuring the accuracy of the reconstruction of the external stimuli: this task is not exactly what the brain is required to accomplish, but it is an efficient way to check the validity of the coding scheme.

The situation is quite different if we analyze the neural code that is involved in neurons that are assigned to elaborate information that is in deeper areas of the processing flow of the brain. Here, it is not possible to make a comparison with external stimuli, and the neural code is a black box from any point of view. In the model that is presented in this paper, we use different and complementary coding schemes inside the same neuron model. The IRF neuron integrates spikes that arrive from different dendrites with different coding schemes. The spikes that arrive from sensory neurons stimulate the internal oscillator, while the spikes that arrive from the IRF neurons in the previous feature clusters are integrated. The integration of the spikes from the previous clusters is spatial and does not contain any reference to the timing except for the latency between the spikes that are in subsequent layers/clusters. The simple presence or absence of a spike is elementary information that is similar to a digital system in which the voltage levels are over or under a certain threshold. In the case of spikes that arrive from sensory neurons, the timing distance of a couple of spikes is sufficient to determine the resonance condition and then the activation.

Until now, SHARP is not a spiking neural network that follows the criterion that the timing of a single spike must be meaningful.

Thus, we introduce a time delay into the synapses that carry spikes from a feature cluster to the next feature clusters.

The time delay can be different for any synapse. Furthermore, the time delay is learned as well as the synapse value.

The matrix of time delays that are associated with forward excitatory connections is, from (13), defined as follows:

$$\Pi = \begin{bmatrix} 0 & \mathrm{Y}_{12} & \cdots & \mathrm{Y}_{1(m-1)} & \mathrm{Y}_{1m} \\ \mathrm{Y}_{21} & 0 & \cdots & \mathrm{Y}_{2(m-1)} & \mathrm{Y}_{2m} \\ \cdots & \cdots & 0 & \cdots & \cdots \\ \mathrm{Y}_{(m-1)1} & \mathrm{Y}_{(m-1)2} & \cdots & 0 & \mathrm{Y}_{(m-1)m} \\ \mathrm{Y}_{m1} & \mathrm{Y}_{m2} & \cdots & \mathrm{Y}_{m(m-1)} & 0 \end{bmatrix} \quad (52)$$

$$\ni \forall (m = m')\mathrm{Y}_{mm'} = 0$$

$$m = feature\_index$$

$$\mathrm{Y} = \begin{bmatrix} \mathrm{T}_{11} & \mathrm{T}_{12} & \cdots & \mathrm{T}_{1n} \\ \mathrm{T}_{21} & \mathrm{T}_{22} & \cdots & \mathrm{T}_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \mathrm{T}_{n1} & \mathrm{T}_{n2} & \cdots & \mathrm{T}_{nn} \end{bmatrix} \quad (53)$$

$$n = feature\_value\_index$$

$$\mathrm{T} = \begin{bmatrix} \tau_1 & \tau_2 & \cdots & \tau_j \end{bmatrix} \quad (54)$$

$$j = category\_index$$

The following is the resuming matrix of time delays that is associated with forward excitatory connections.

$$\Pi = \begin{bmatrix} 0 & \begin{bmatrix} \mathrm{T}_{11} & \begin{bmatrix} \tau_1 & \tau_2 & \cdots & \tau_j \end{bmatrix}_{12} & \cdots & \mathrm{T}_{1n} \\ \mathrm{T}_{21} & \mathrm{T}_{22} & \cdots & \mathrm{T}_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \mathrm{T}_{n1} & \mathrm{T}_{n2} & \cdots & \mathrm{T}_{nn} \end{bmatrix}_{12} & \cdots & \mathrm{Y}_{1(m-1)} & \mathrm{Y}_{1m} \\ \mathrm{Y}_{21} & 0 & \cdots & \mathrm{Y}_{2(m-1)} & \mathrm{Y}_{2m} \\ \cdots & \cdots & 0 & \cdots & \cdots \\ \mathrm{Y}_{(m-1)1} & \mathrm{Y}_{(m-1)2} & \cdots & 0 & \mathrm{Y}_{(m-1)m} \\ \mathrm{Y}_{m1} & \mathrm{Y}_{m2} & \cdots & \mathrm{Y}_{m(m-1)} & 0 \end{bmatrix} \quad (55)$$

In the new behavior of the network, neurons in the feature clusters emit a spike when the activation threshold is exceeded, but the spike acts on the postsynaptic neurons with different time delays. Time delays that are associated with synapses are learned with the STDP learning rule. From (32),we build formula (56), which has a new time interval threshold:

$$\because \left(t_{j'n'm'}^{IRF} - t_{jnm}^{IRF} < STF \times \eta\right) \wedge \left(\left|t_{j'n'm'}^{IRF} - t_{jy}^{CC}\right| < STF\right)$$

$$\therefore \left(\rho_j \in \omega_{nn'} \in \varphi_{mm'} \in \phi\right) = \rho_j \oplus \left(\varepsilon \times \left(\Delta + \eta^{-1} \times \left(O_{jnm}^{SUA} \otimes O_{j'n'm'}^{SUA}\right)\right)\right)$$

$$t_{jnm}^{IRF} = presynaptic\_spike\_time$$

$$t_{j'n'm'}^{IRF} = postsynaptic\_spike\_time$$

$$t_{jy}^{CC} = category\_neuron\_spike\_time \quad (56)$$

$$STF = systolic\_time\_frame$$

$$\eta = delay\_factor$$

$$\varepsilon = learning\_const$$

$$\Delta = systolic\_consensus\_offset$$

$$\rho = updated\_synapse$$

The STDP formula for updating the time delay of the synapse is

$$\begin{aligned}&\because \left(t^{IRF}_{j'n'm'} - t^{IRF}_{jnm} < STF \times \eta\right) \wedge \left(\left|t^{IRF}_{j'n'm'} - t^{CC}_{jy}\right| < STF \times \eta\right)\\ &\therefore \left(\tau_j \in \mathrm{T}_{nn'} \in \mathrm{Y}_{mm'} \in \Pi\right) = \tau_j \otimes \left(t^{S}_{j'n'm'} - t^{S}_{jnm}\right)\\ &\otimes = fuzzy\_AND\_operator\\ &\ni \because (a \geq b) \therefore (a \otimes b = b), \because (a < b) \therefore (a \otimes b = a)\end{aligned} \quad (57)$$

The synapse takes a delay that is equal to the time interval between the last spike that is emitted by the presynaptic neuron and the spike that is emitted by the postsynaptic neuron, provided that this interval is within a certain threshold.

A matrix of delays for the connections between feature cluster neurons and category cluster neurons is the following (from (21)):

$$\Omega = \begin{bmatrix} \mathrm{A}_{11} & \begin{bmatrix} \tau_{11} & \tau_{12} & \dots & \tau_{1n} \\ \tau_{21} & \tau_{22} & \dots & \tau_{2n} \\ \dots & \dots & \dots & \dots \\ \tau_{y1} & \tau_{y2} & \dots & \tau_{yn} \end{bmatrix}_{12} & \dots & \mathrm{A}_{1m} \\ \mathrm{A}_{21} & \mathrm{A}_{22} & \dots & \mathrm{A}_{2m} \\ \dots & \dots & \dots & \dots \\ \mathrm{A}_{j1} & \mathrm{A}_{j2} & \dots & \mathrm{A}_{jm} \end{bmatrix} \quad (58)$$

Updating (28) and (33) to manage the delays in multiple STFs:

$$\kappa'_{yn} = \left(\varepsilon \times \left(\Delta + \eta^{-1} \times \left(\left|f - f_{Rn}\right| + \varpi\right)^{-1}\right)\right) \oplus \kappa_{yn} \quad (59)$$

$$\begin{aligned}&\because \left(\left|t^{IRF}_{jmn} - t^{CC}_{jy}\right| < STF \times \eta\right)\\ &\therefore \left(\kappa_{yn} \in \chi_{jm} \in \psi\right) = \kappa'_{yn}\end{aligned} \quad (60)$$

The STDP formula for updating the time delay of the synapses is

$$\begin{aligned}&\because \left(\left|t^{IRF}_{jnm} - t^{CC}_{jy}\right| < STF \times \eta\right)\\ &\therefore \left(\tau_{yn} \in A_{jm} \in \Omega\right) = \tau_{yn} \otimes \left|t^{IRF}_{jnm} - t^{CC}_{jy}\right|\\ &\otimes = fuzzy\_AND\_operator\\ &\ni \because (a \geq b) \therefore (a \otimes b = b), \because (a < b) \therefore (a \otimes b = a)\end{aligned} \quad (61)$$

The addition of the time domain to the network behavior enables the network to complete the identification of a pattern by making inferences on the evolution of the pattern over time.

The new network can reconstruct patterns that have very noisy features, which makes inferences on the history of other features, as shown in Fig.16. We have labeled as "LTIM" this capability to reconstruct the dynamic temporal evolution of the input. The network operates with three types of generalizations: LSUP (feature value), LNUM (pattern incompleteness), and LTIM (temporal evolution).

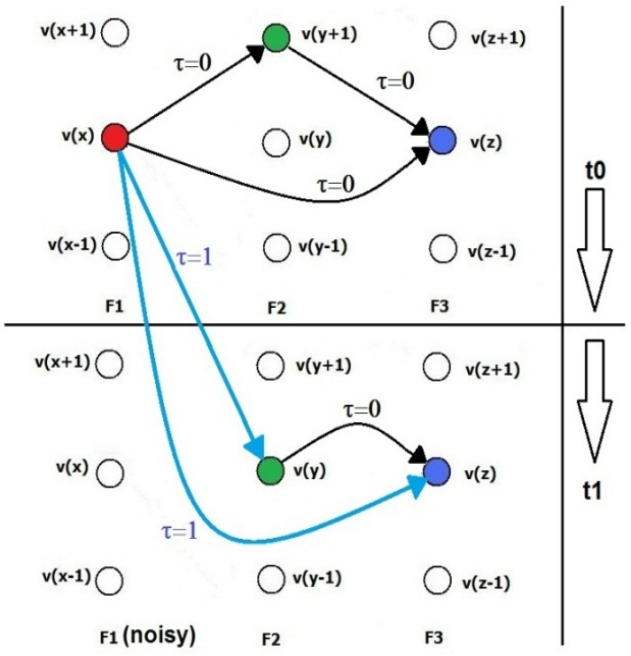

Figure 16. The LTIM generalization capability of SHARP. At the instant t1, the feature F1 is extremely noisy and, assuming that LSUP is set to 0 or saturated, the history of the feature relationships enables the recognition of the pattern as a result of spikes that are transmitted through delayed synapses

The network works in the space and time domain using single spike timing to build information; thus, we argue that SHARP is a spiking neural network although it does not have the requirements that were established by Maass. In a third generation neural network, following the Maass criteria, the neuron model is more biologically plausible andhas an activation function that makes the membrane potential evolve over time. At any time step, the membrane potential of any neuron is updated. When the membrane potential exceeds the threshold, the neuron fires. The time is involved in the activation function and in the synaptic delay. In SHARP, the time domain is involved exclusively in the synaptic delay. Nevertheless, SHARP manages an SDR (Sparse Distributed Representation) of reality in the time and space domains.

7. Autonomous Machine Learning

The learning activity of SHARP is triggered by a spike train to an SL neuron from an external module. In a neurocognitive network, modules that are associated with different modalities should be able to learn autonomously and thereby start a cooperative process.

A method for deciding whether the current stimulus must be learned and associated with some "unlabeled" category node is using a probabilistic method. The network should trigger a learning activity when the same stimulus has been received more frequently than others in a relatively large

timeframe.

We have provided unsupervised learning by adding an external modulethat associatesun supervised learning with STM (Short Term Memory) and supervised learning with LTM (Long Term Memory). The module is a spiking Learning Vector Quantization (LVQ) neural network. Any prototype is connected to an "unlabeled" category neuron and is continuously updated by the input patterns. We have called this module ULM (Unsupervised Learning Module).ULM works as a "working memory" in which the input patterns are memorized for a short period of time. For a frequent input pattern, the LVQ process moves the synapse weights of the nearest prototype in the direction of the pattern. When the input pattern is very close to the prototype, the associated neuron is excited over the threshold and emits a spike.

Fig.17 shows the connections between ULM and SHARP.

The synapse Sx transfers the action potentials from the prototype neuron in ULM to the SL neuron in SHARP.

In this way, the prototype neurons of ULM are randomly connected to the SL neurons, distributing supervised learning of the STM neurons in the ULM to the SL neurons in SHARP. The conductivity of Sx decays (anti-Hebbian) when crossed by action potentials, and Cx is connected to another SL neuron.

In this way, the category neurons of ULM are randomly connected to the SL neurons, distributing supervised learning of the STM prototypes that are eligible to become LTM (Long Term Memory): Fig.18 explains the possible connection from STM to LTM assuming that SHARP is located in the cortex while ULM is located in the hippocampus (Fig. 21). The hypothesis of this model is that there would be an STM module in the hippocampus for any local area network in the cortex.

Thus, when an SL neuron of SHARP is activated by an external action potential, an LTM activity is triggered. Then, the SL neuron and the associated category neuron cannot be further engaged for a learning activity. This model suggests the possibility that the STM could be held in separate streams as a result of uni-modal unsupervised learning and can be promoted to LTM through supervised learning.

The LVQ NN has an input layer and a prototype layer. Any input is connected to any prototype through plastic synapses.

In the proposed model, the input neurons resonate and fire, which converts the level of resonance to a single output spike timing. There is a fixed number of neurons for any feature, which are associated with specific values and resonate at specific frequencies. Thus, the input is expanded from *m* (the number of features) to *nxm* (where n is the number of reference values for any feature).

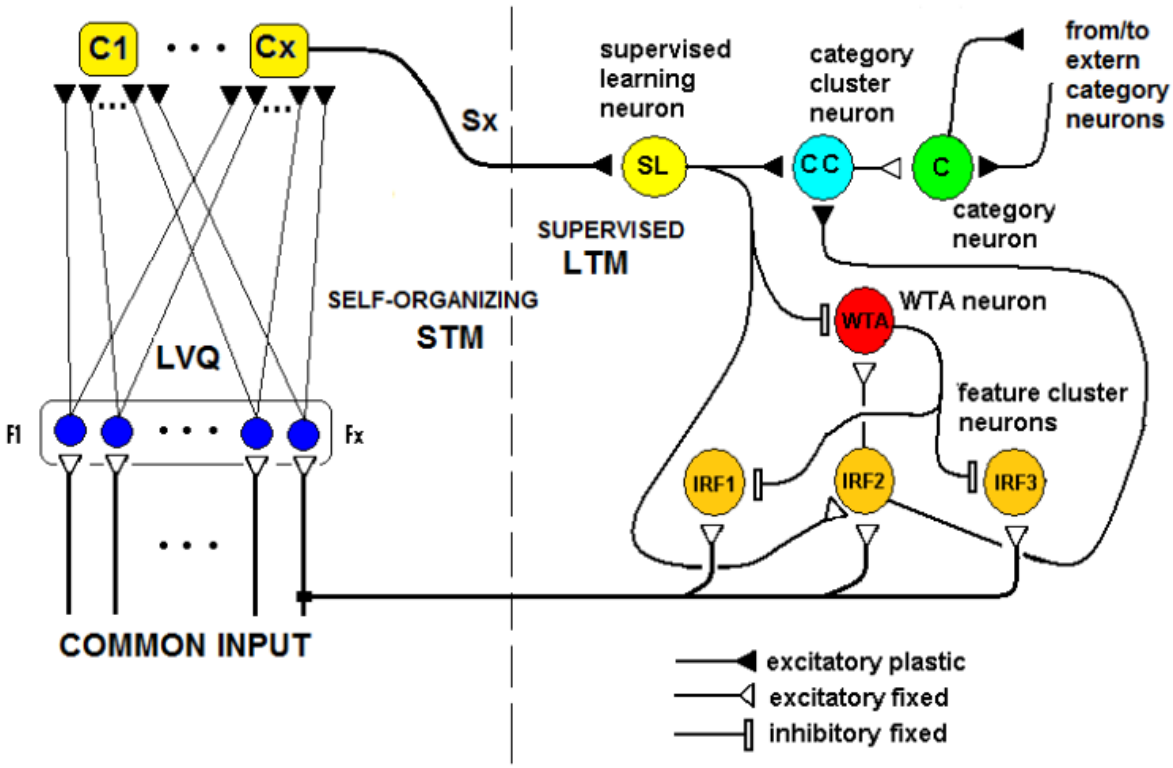

Figure 17. The connection scheme of the "Unsupervised Learning Module" together with the SHARP module. P1, P2, P3 are the prototypes of the ULM that constitute the Short Term Memory. For simplicity, ULM has only three prototypes, and SHARP is shown partially (only the feature Fx-1 cluster with three value neurons IRF1, IRF2, IRF3). Neurons in the prototypes of ULM are connected to the category neurons of ULM with plastic synapses (s1, s2, s3). The strengths of s1, s2 and s3 decay over time but grow when the prototype is resonating with an input pattern. The most important part of this scheme is the sC3 synapse that connects the ULM with the SHARP module. This synapse is plastic and decays when crossed by action potentials

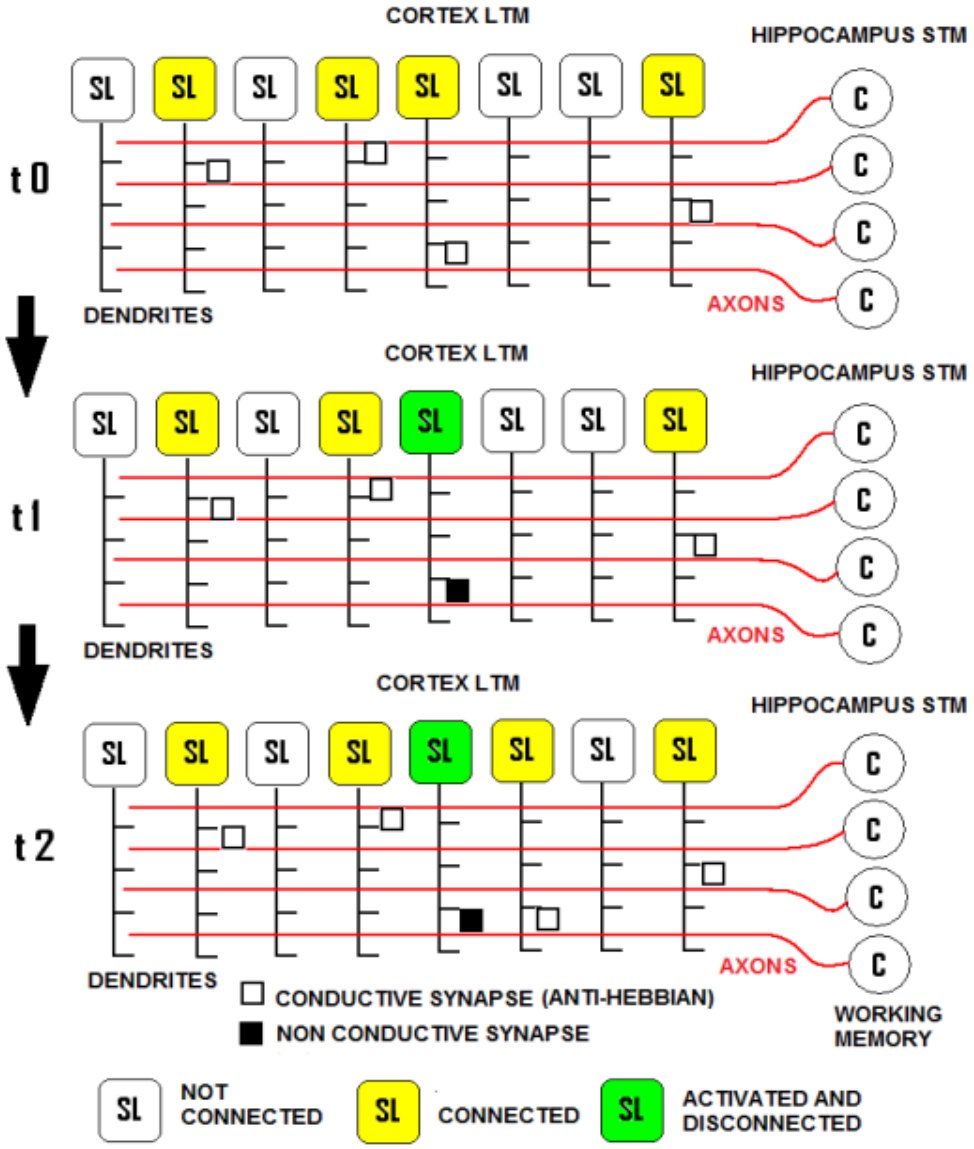

Figure 18. The interconnection matrix between our model of STM or working memory (the ULM module), which works in an unsupervised manner, and our model of LTM (the SHARP module), which waits for an external teaching signal on the SL neurons to activate a category node. Any SL neuron can be connected with only one axon from the working memory, and vice versa. When a neuron of the ULM (C) emits action potentials, the SL neuron that is associated is activated, and the anti-Hebbian synapse is deactivated. This SL neuron loses the possibility of receiving other inputs from the ULM

Inside the ULM module are the following matrixes of synapses. The first matrix is the ensemble of fixed excitatory connections of feature inputs to input neurons:

$$S = \begin{bmatrix} s_{11} & \dots & s_{1M} \\ \dots & \dots & \dots \\ s_{N1} & \dots & s_{NM} \end{bmatrix} \quad (62)$$

$$N = grid\_values$$

$$M = features\_number$$

The second matrix is relative to the Hebbian excitatory connections between input neurons and the excitatory prototype neurons (neurons of type 4 in Figs.19 and 20):

$$P = \begin{bmatrix} p_{11} & \dots & p_{1X} \\ \dots & \dots & \dots \\ p_{I1} & \dots & p_{IX} \end{bmatrix} \quad (63)$$

$$I = inputs\_number$$

$$\ni I = N \times M$$

$$X = prototypes\_number$$

The third matrix is relative to the anti-Hebbian excitatory connections between the input neurons and the inhibitory prototype neurons (neurons of type 3 in Figs.19 and 20):

$$\overline{P} = \begin{bmatrix} \overline{p}_{11} & \cdots & \overline{p}_{1X} \\ \cdots & \cdots & \cdots \\ \overline{p}_{I1} & \cdots & \overline{p}_{IX} \end{bmatrix}$$
$$I = inputs\_number \tag{64}$$
$$\text{э } I = N \times M$$
$$X = prototypes\_number$$

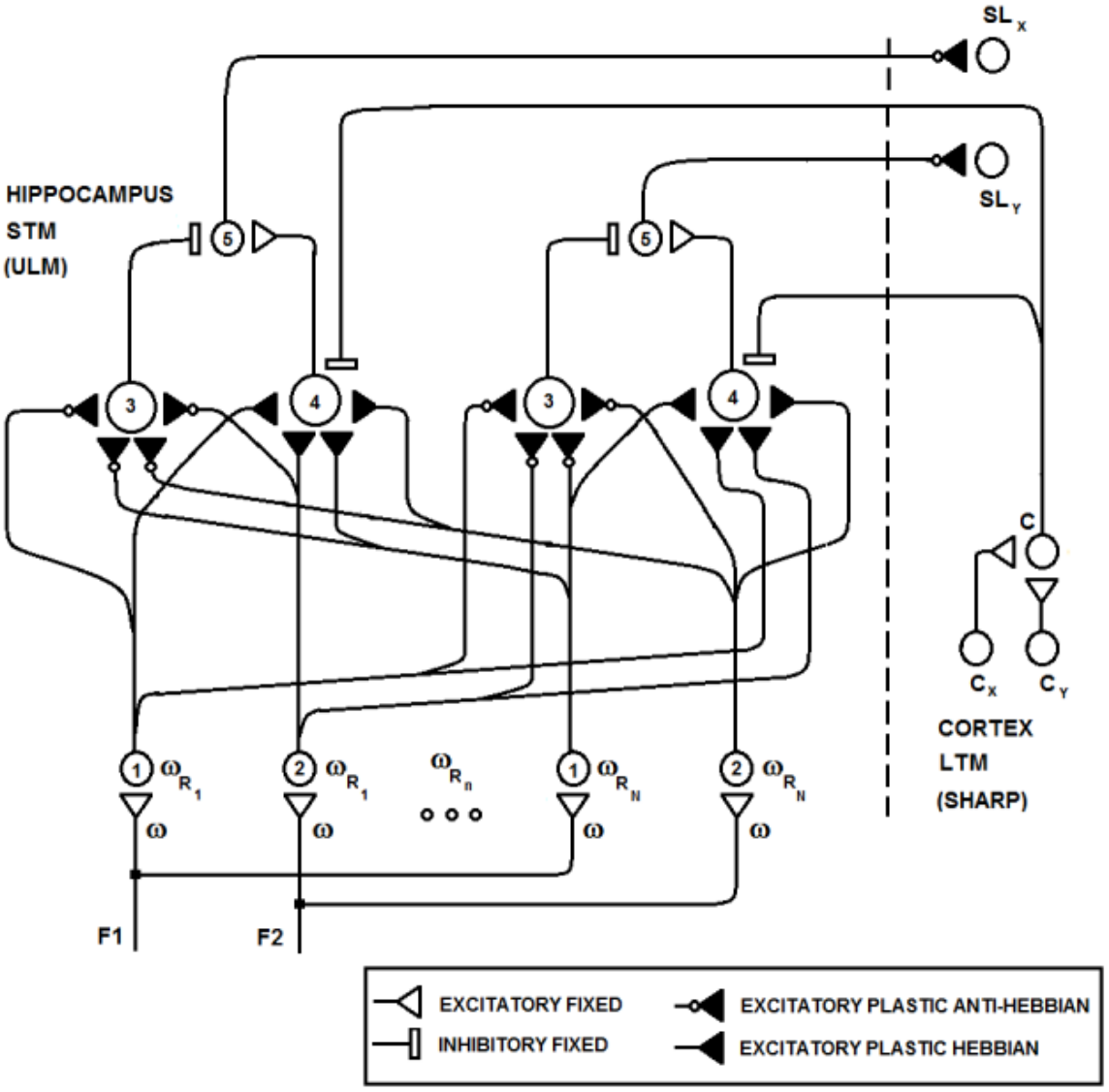

Figure 19. Details of the neural circuit in the ULM with two features (F1, F2) and N resonation values in the input grid (neuron types 1, 2). Neurons of types 3 and 4 are Rank Order Integrate and Fire with anti-Hebbian and Hebbian synapses, respectively. Neurons of type 5 fire when the excitatory activity of neurons of type 4 exceeds the inhibitory activity of neurons of type 3. This circumstance occurs when the same pattern is presented very frequently and, thus, a training signal is sent to the cortex (SL neurons). Category neurons in the cortex inhibit neurons of type 4, and thus, a pattern that is recognized at the LTM level cannot be learned at the STM level

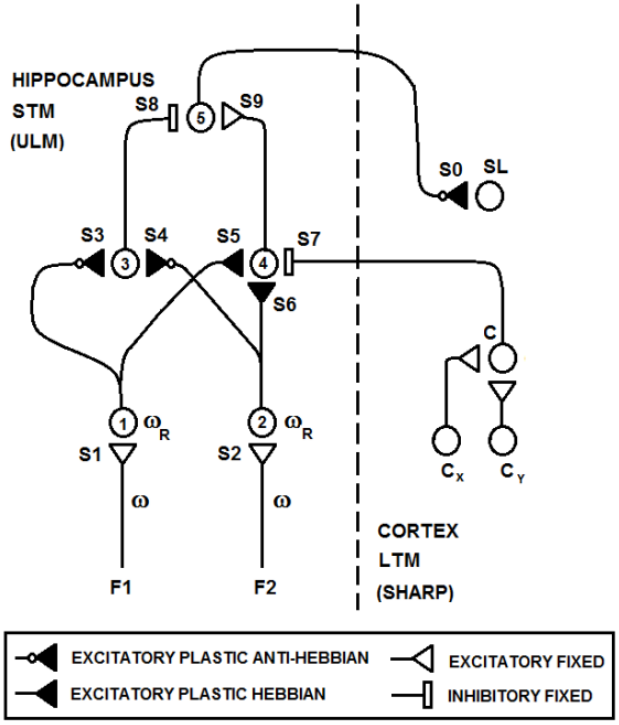

Figure 20. A simplified view of the circuit shown in Figure 19. Only the neurons that are related to one resonation value are shown. It can be seen that there is a pair of neurons (3, 4) for any prototype, and any input neuron connects to any neuron 3 with anti-Hebbian synapses and to any neuron 4 with Hebbian synapses. The circuit would be the same considering one feature and two resonation values. Neuron C in the cortex fires when an input pattern is recognized (one category of neuron fires)

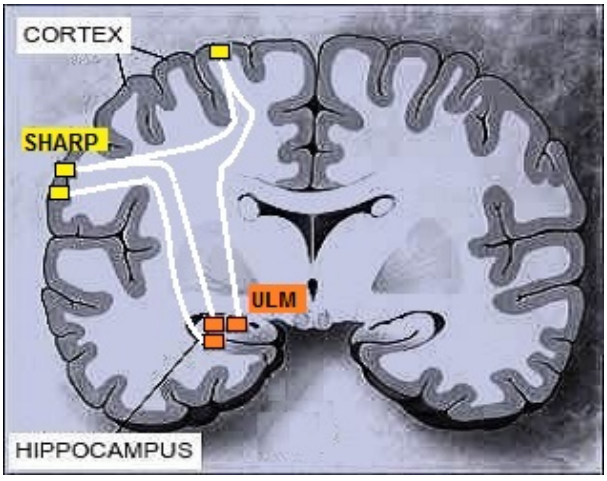

Figure 21. The hypothetical position in the brain of the SHARP modules (local cortical area networks) and the associated ULM modules. Local cortical area networks should interconnect with one another in the cortex layers or possibly through white matter. Short Term Memory modules in the Hippocampus should connect to local cortical area networks in the cortex through white matter

The fixed excitatory connections between the excitatory prototype neurons and the final prototype neurons are defined as follows:

$$G = \begin{bmatrix} g_1 & \dots & g_X \end{bmatrix}$$
$$X = prototypes\_number \tag{65}$$

The fixed inhibitory connections between the inhibitory prototype neurons and the final prototype neurons are defined as follows:

$$\overline{G} = \begin{bmatrix} \overline{g}_1 & \dots & \overline{g}_X \end{bmatrix}$$
$$X = prototypes\_number \tag{66}$$

From the SHARP counterpart, there are fixed inhibitory synapses that act on the excitatory prototype neurons:

$$Q = \begin{bmatrix} q_1 & \dots & q_X \end{bmatrix} \tag{67}$$

The learning algorithm acts only on $P, \overline{P}$ and is given as follows:

$$p_{ix} = p_{ix} + \varepsilon \times (t_x - t_{ix})$$
$$\varepsilon = const$$
$$t_x = post\_synaptic\_spike\_time$$
$$t_{ix} = pre\_synaptic\_spike\_time \tag{68}$$

and

$$\overline{p}_{ix} = \overline{p}_{ix} - \varepsilon \times (t_x - t_{ix})$$
$$\varepsilon = const$$
$$t_x = post\_synaptic\_spike\_time$$
$$t_{ix} = pre\_synaptic\_spike\_time \tag{69}$$

The neurons of types 1 and 2 (Figs.19 and 20) have, in fact, the same behavior because they must resonate to a specific frequency (a doublet) and emit a spike that has an associated time delay. These neurons have been differentiated only because they are related to different features:

$$O(t) = f((\omega - \omega_R), t)$$
$$if\,(O > threshold)\_then\_neuron\_fires\_after\_\tau$$
$$O = membrane\_potential$$
$$\omega = input\_frequency$$
$$\omega_R = resonant\_frequency$$
$$\tau = spike\_time\_delay \tag{70}$$

The neurons of types 3 and 4 are Rank Order Integrate and Fire:

$$
\begin{aligned}
&O = \sum_{n=1}^{N} w_n \times (\varepsilon \times \zeta_n)^{(n-1)} \\
&if\,(O > threshold)\_then\_neuron\_fires \\
&O = membrane\_potential \\
&\varepsilon = const \\
&\zeta = action\_potential \\
&\ni\,(\varepsilon \times \zeta_n) < 1.0 \\
&n = input\_number
\end{aligned}
\tag{71}
$$

Only one specific sequence of spikes raises the membrane potential to exceed the threshold [10].

The exponent (n-1) makes the sensitivity of the cell decrease by a specific factor that depends on the constant that is in the formula.

Neurons of type 5 (Figs.19 and 20) are simple Integrate and Fire neurons.

8. Deep Learning in Hierarchy and Composite Complex Structures

It is universally recognized that the brain is intrinsically parallel but is also a hierarchical structure. Hierarchy is a fundamental concept in any complex behavior; thus, we must explore how the proposed model could work in a hierarchical structure. This exploration is a fundamental analysis for the purpose of measuring how efficiently the elementary behavior can be a "brick" of a more complex system and thus, in our case, of the brain.

It is possible to reduce the LSUP maximum distance in the learning activity, thus providing higher specialization of the concepts that are in lower levels of the hierarchy. As an example, "vehicle" would be a parent class of "car" and "truck".

In the SHARP simulation on the computer, the processing speed is independent of the number of examples learned. The number of synapses is, instead, directly proportional to the number of categories that are required for the classification. In a software implementation, the larger the number of classes is, the larger the memory that is required to store the synapses, and the execution time is linearly affected. Thus, building a hierarchy of SHARP modules by clustering categories (or classes) is a way of reducing drastically the memory that is required for the synapses.

Hierarchy is not the only way that multiple SHARP networks can be interconnected to produce more complex neural networks and behaviors. For example, multiple SHARP networks can be connected to have the same global input pattern but different identification tokens of the associated category. In this way, the bottleneck of the synapses that are associated with the categories can be easily solved:

$$
\begin{aligned}
&Tcn = Ncn^{Nn} \\
&Tcn = tot\_cat\_number \\
&Ncn = net\_cat\_number \\
&Nn = net\_number
\end{aligned}
\tag{72}
$$

Fig.22 shows an example of multiple SHARP modules that are connected in a SCC (Symbolic Category Composition) mode. As can be seen, input is shared among all of the networks, while the associated category is composed trivially as an ensemble of configurations. In this example, it is possible to manage 19^4=130321 categories, but in a serial computer, the execution time is only four times greater (the sequential execution of the four SHARP networks).

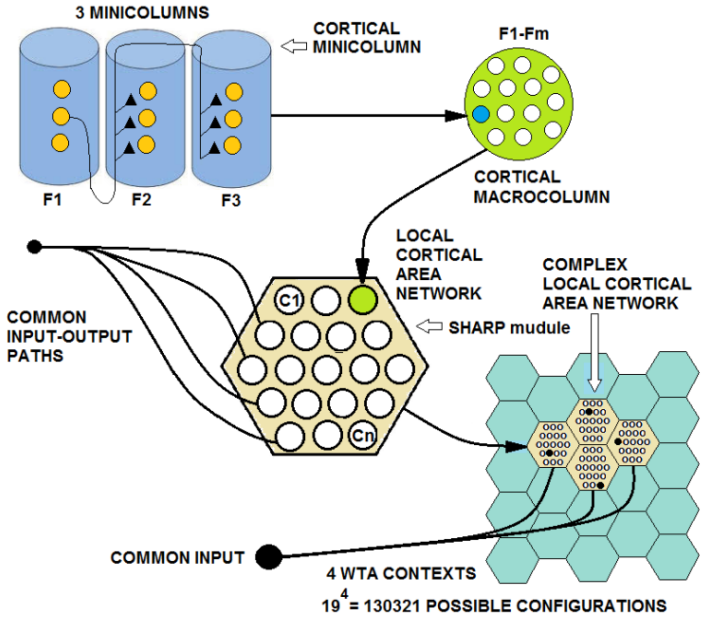

Figure 22. Multiple SHARP networks are connected to the same inputs, and learn only a token of the distributed category representations. This configuration can be considered to be a complex cortical area network. A total of 4 modules that are composed of 19 hyper-columns can represent 19^4 configurations

9. Hierarchical Concepts Associative Nodes in a Self-Organizing Neurocognitive Network

Multiple parallel streams in the brain communicate among themselves during an information processing flow. These streams are cognition processes or sensorial processes that communicate among themselves at different levels, merging and integrating information through inter-stream channels. We have attempted to analyze how SHARP could be one type of the bricks that are components of a complex neurocognitive network. The approach that we propose in this paper is well known in literature [13] and we only have inserted the SHARP modules in such a framework adapting some concepts to the neural network model. We have defined HiCANs (Hierarchical Concept Associative Nodes), which should behave as channels for inter-stream communication. The principal idea is that HiCAN(s) work as nodes where knowledge of a specific stream can be associated with knowledge of another stream.

We have defined a framework in which a neurocognitive network grows around simple learning elements that each process a single stream. We have analyzed here how the proposed neural network can be the basic learning element of the system while not excluding that other different behavioral elements could be included. Category nodes are directly connected to HiCAN(s) because a category node can be activated by another category node of another stream, as shown in Fig.23.

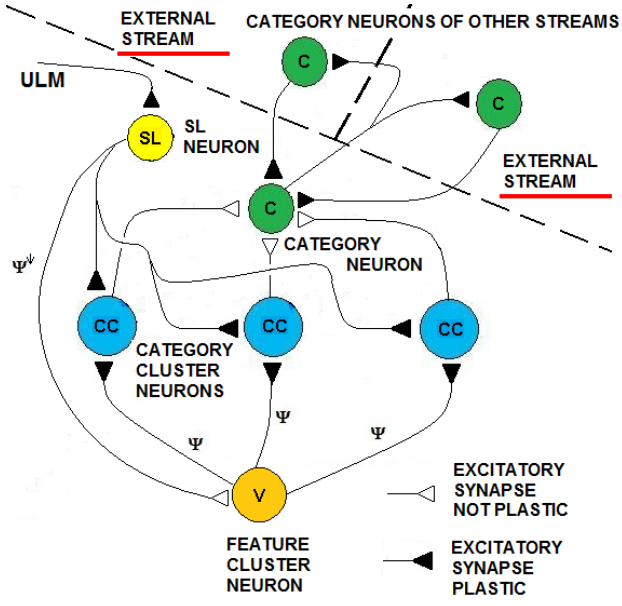

Figure 23. Two Concept Associative Nodes. These are inter-stream interconnections between category neurons of different streams

The framework of the proposed neurocognitive network follows these criteria:

- The existence of multiple information streams carried by learning modules (local cortical area networks)
- Learning modules should be able to grow on the "need" by committing new neurons that can manage additional features
- The modules should be able to classify input patterns by activating category neurons
- The modules should be able to learn, without supervision, frequently recurring patterns and to associate them with agnostic category neurons that are created based on "need" through external ULMs
- HiCAN is not a neuron but instead is a network of neurons that are strictly correlated with a concept. There is a high probability that if one neuron fires, all neurons in the network fire
- The streams can be organized into multiple hierarchies, but streams at any level of a hierarchy can communicate through HiCANs with streams at any level of another hierarchy
- The streams can be sensory-processing or associative-processing
- HiCANs can exist at different levels of the hierarchy because they should be representing concepts at different levels of abstraction (example: "blond girl" and "Marilyn Monroe")

The attractor state of a neurocognitive network is a state of coordination in which interconnected local area networks coordinate their activities within milliseconds [11, 12]. According to this hypothesis, local area networks in a neurocognitive network set local spatiotemporal constraints on one another through long-range interconnections and, thus, self-organize into large-scale patterns of coordinated activity [13]. HICANs could be elements of such behavior because they impose associations to spatiotemporal patterns between local area networks.

Furthermore, the HiCAN that is described in this paper recalls some recent discoveries about neurons that have been called "concept cells" by neuroscientists who have found them in the medial temporal lobe (MTL) region of the human brain. These cells have highly selective responses to complex stimuli [14-23].

The "concept cells" are neurons that were found to be responsive to pictures of specific persons or specific objects.

Quian Quiroga et al. [18] estimate that approximately 40% of the responsive units in MTL behaviour can respond to complex categories (concepts), such as a specific person or object.

In their view, the existence of cells that respond to a single individual or category (category cells) is compatible with the thinking that there are cells that encode aspects of meaning of a specific stimulus. A HiCAN connects different streams that are related to different modalities (i.e., images, sound). The role of HiCAN is to complete a complex concept, such as a person or an object, through the association of category nodes that are "descriptors" of such a concept in a specific modality. The existence of a HiCAN that represents "John" does not mean that there is a category neuron in the stream "voice recognition" that represents John. That category

could be shared with other persons at least up to a certain degree of the features details (see 1.8 Plasticity by Class Specialization).To find a more complete match between the "concept cells" and the HiCANs, we must consider more recent experiments, which are reported in Quian Quiroga, Kraskov, Koch and Fried [19]. These investigators found that single MTL neurons can encode information about the same percept that can arise in different modalities, such as visual, textual and sound.

This experiment has been realized through the use of implanted microelectrodes, from which the researchers recorded from 750 MTL units. They found 79 responsive units, and 17 of them have the above-described triple invariance behavior (visual, textual and sound). Furthermore, the most important experiment has shown the activation of a concept cell by consciously thinking about the associated object. These experiments show that there is an obvious connection between the "thinking" about an image and the firing rate of the corresponding concept cell [23].

Christ of Koch wrote the following: "Every time you encounter a particular person or object, a similar pattern of spiking neurons is generated in higher-order cortical regions. The networks in the medial temporal lobe recognize such repeating patterns and dedicate specific neurons to them. You have concept neurons that encode family members, pets, friends, co-workers, the politicians you watch on TV, your laptop, that painting you adore. Conversely, you do not have concept cells for things that you rarely encounter, such as the barista who just handed you a non fat chai latte tea."

About the question of whether concept cells can be considered to be grandmother cells [24-26, 18] responded: "Although these cells bear some similarities to "grandmother cells', several arguments make this interpretation unlikely. First, it is implausible that there is one and only one cell responding to a person or concept because the probability of finding this cell, out of a few hundred million neurons in the MTL, would be very small."

Consistent with this statement, the definition of HiCAN is "a network of neurons" that fire together to represent a concept. These neurons that concur in a sparse representation of a concept have been called "sister cells" [14] and are not mandatorily in contiguous locations of the brain.

All of the aspects of this scenario match correctly the behavior of the HiCAN that we have applied in our framework for building cognitive systems based on SHARP neural network modules. A fundamental question about this sparse representation of a concept in multiple "concept cells" is whether it is possible to infer the existence of the concept by verifying only the activation of a single cell of such a network of concept cells. The opinion of the scientists who worked on these experiments is that there is no need to check the activation of all of the other concept cells because the results of the experiments showed a consistent combination of selectivity and invariance [18] of the single cell behavior in response to a specific conceptual stimulus. A consistent combination of selectivity and invariance is, in our opinion, the proof that single "concept cells" are part of a network (HiCAN) that is strongly correlated with the concept. Thus, we agree that there is no need to check the activation of all of the other concept cells. Nevertheless, there is only a high probability that such a network is univocally excited by such a concept.

There is another aspect that we would like to underline about the similarities of the framework that is proposed in this paper and the recent experimental results on "concept cells" [14-22]: in their experiments, the researchers found concept cells in different MTL regions, such as in the left and right sides of the hippocampus or in the left side of the parahippocampal cortex and in the amygdala; however, the highest degree of invariance (across modalities) was found in the hippocampus and entorhinal cortex. This fact suggests that concept cells exist at different levels of abstraction (due to different degrees of invariance) and are probably located near the stream to which they are associated (far regions in the brain). Different levels of abstraction and the concurrence of multiple streams are two key features of the framework that was developed for HiCANs. We have called this framework "MOdular Scalability AImed to Cognition" (MOSAIC) because new cognitive streams (modules) can be added to the system that connects them through HiCAN(s) to existing modules that are "experience streams" and could work as teachers (Fig. 23, Fig.24a, Fig. 24b, Fig.24c).

These nodes could also be considered to be a special case of the "global workspace model" [27-29]. This theory views the human brain as being organized into a network of specialized automatic processors that provide for sensation, reasoning, motor control, language and other functions. This theory assumes that there is a global workspace that is widely distributed throughout the brain (especially in the cerebral cortex), which shares its content with the specialized processors.

In this framework, the specialized processors compete to gain access to this global workspace, to send and receive globally available information.

The HiCAN framework operates as a generic global workspace model that behaves in a cooperative and synergic context instead of in a competitive manner. The information is shared selectively between streams that are contextually correlated during any specific cognitive effort. HiCANs behave as nets of signals that carry information at different levels of the hierarchy.

The global workspace model has been recognized as a prominent approach to modeling the neural correlates of consciousness. According to this theory, an increase in the globally distributed activity and the inter-communication between the regions of the cerebral cortex is well documented during conscious mental effort. More specifically, information in the memory reaches consciousness when the amount of activity that represents that information crosses a threshold [30]. The global workspace has been explored with simulations that are based on IDAs (Intelligent Distributed Agents) [31]. IDAs are not a connectionist system, and they are based on artificial distributed agents ("codelets") that are realized as

independent and concurrent Java processes that are executed on a JVM (Java Virtual Machine), which runs on a conventional computer.

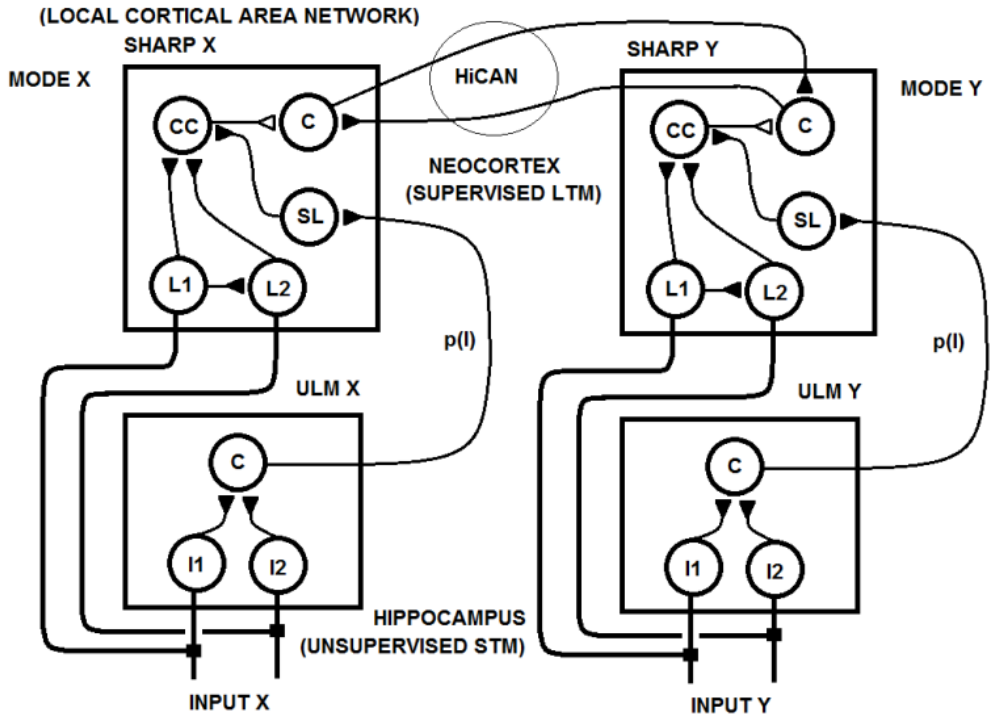

Figure 24a. A multimodal concept network (HiCAN) is created through the STDP between two category neurons in the two cortical area networks

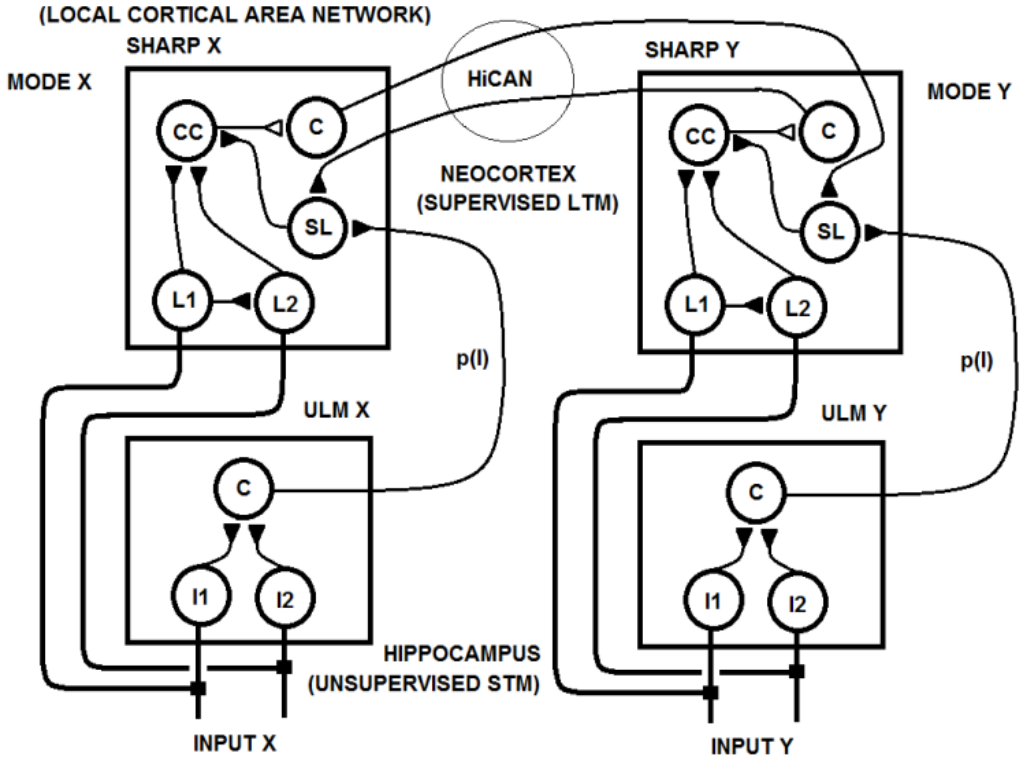

Figure 24b. In the second hypothetical circuit, a multimodal concept network (HiCAN) is created through STDP between the category neurons and the SL neurons in the two cortical area networks

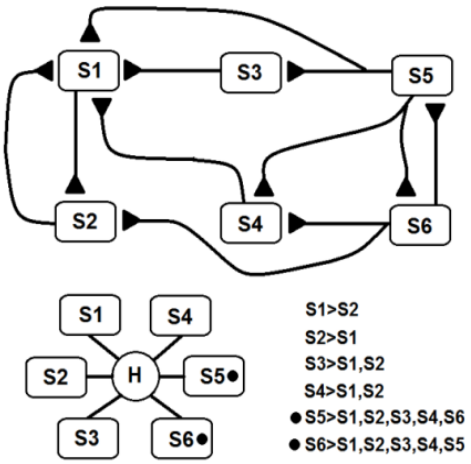

Figure 24c. In the HiCAN that is shown, s1, s2, s3, s4, s5 and s6 are connected streams. In this HiCAN, only S5 and S6 represent a complex concept. The bottom-left simplification is useful for representing complex networks of interconnected streams, such as in Fig. 24c

10. Optimized Software Simulation

In the history of neural networks, pattern recognition problems have been a paramount application target and benchmark.

In recent years, we have observed development of many new paradigms more or less inspired by biological systems. These often have been applied to pattern recognition problems running on Von Neumann architecture computers. Due to the serial nature of these computers, many paradigms suffer low performance for real time applications, whereas others do not show capability to store many new patterns without losing or affecting previous knowledge (plasticity versus stability problem). We have proposed a paradigm inspired by the Drosophila olfactory system that can be simulated very efficiently on Von Neumann architectures.

Following these concepts, synapses between feature neurons have been simulated with a 4-dimensional matrix of bytes while synapses between feature neurons and category neurons have been simulated with a 3-dimensional matrix of bytes. Below we will analyze the operations performed during the learning phase and during the recognition phase.

● In the learning phase, the following conditions are met:

Any synapse in the 4-dimensional byte-matrix connecting two resonating neurons is updated.

Any synapse in the 4-dimensional byte-matrix connecting a resonating neuron with neighborhood activating neurons in different layers is updated. Any synapse in the 3-dimensional matrix of bytes connecting a resonating neuron or a neighborhood activating neuron with a category neuron is updated. The maximum value is assigned to this synapse in the case of a resonating neuron. In the case of a neighborhood neuron, the value assigned to the synapse is inversely proportional to the distance of the neighborhood neuron from the resonating neuron.

● In the recognition phase, the following conditions are met:

Any resonating neuron in any layer checks in the 4-dimensional bit-matrix if it has received a spike from resonating neurons in any previous layer. If this condition is not verified, the process is interrupted and the input pattern is not identified. Any resonating neuron that verifies the reception of spikes from the preceding layers activates category neurons, sending spikes on the weighted routes defined in the 3-dimensional byte matrix. This operation can be performed by averaging the value of the synapse (in byte matrix) with the current activation of the category node or by operating a fuzzy-AND between the same elements.

Considering the type of operations performed during the learning and recognition phases, it is clear that this algorithm does not require any complex mathematical operators; instead, the algorithm requires only simple read and write operations from and to memory. The number of read and write (r/w) operations is not correlated to the number of patterns the neural network has learned but is fixed. Actually, the number of memory r/w operations is proportional to the number of features. Such software implementation requires poor computation capability and fixed, small or almost null parallelism. This algorithm places a large demand on memory space to memorize the huge number of synapses. From the previous considerations, we can assert that this algorithm is "tuned" with the current commercial technology. Currently, microprocessors use limited parallelism with memory sharing, and they can actually still be classified as Von Neumann machines. Furthermore, in the last twenty years, memories have grown at a much faster rate than processor speed (clock frequency), and the current trend is still the same.

We have created three pattern recognition test-benches with different targets. They are based on sets of artificial databases of random patterns generated by a program appositely designed for this test. These artificial tests are necessary to reach the capacity limits of the network, having full control of statistical parameters of the training set.

10.1. XOR Test

The first theoretical test that we have conducted on the network is the XOR (exclusive OR) test to investigate if the network can solve this problem as a MLP with two hidden layers or as an RBF neural network.

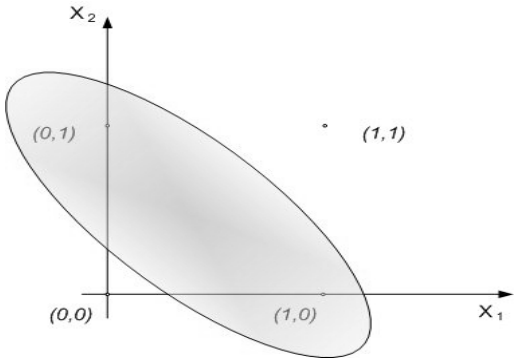

Figure 25. The representation of the XOR problem

Figure 26. The spikes raster related to the XOR problem

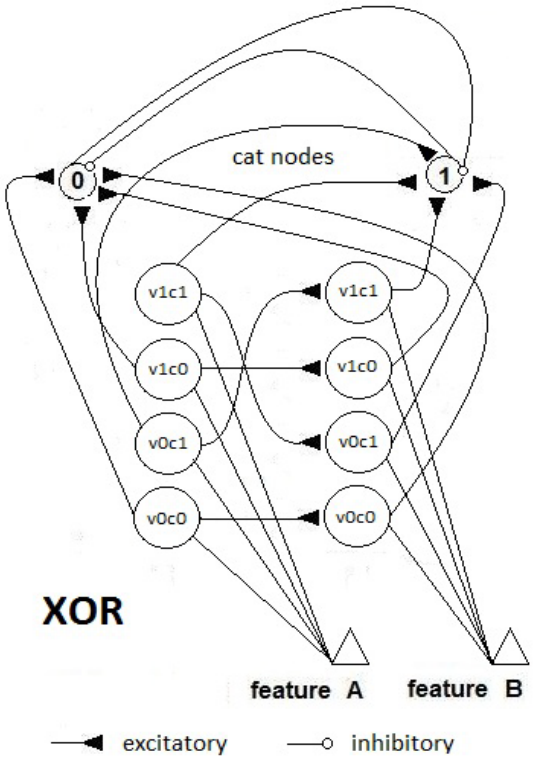

Figure 27. The SHARP neural network after learning the XOR problem. The category clusters have been removed for simplicity. Feature A is the first input of the "gate," and Feature B is the second input. The only valid values for this problem are 0 and 1 (in a digital view), so there are 2x2 neurons in any feature cluster. Only the paths of the forward excitatory synapses activated are shown

This theoretical test is executed to verify the capability of the neural network to perform correctly on problems that are not linearly separable. The typical function used to execute this test is the logic function of exclusive or. As it is well known, the single layer Perceptron cannot learn this function, whereas MLP does. The not linearly separable function of XOR is shown in Fig. 25. SHARP demonstrates its ability to work correctly with the XOR problem. The resulting network is shown in Fig. 27; Fig. 26 represents the related spikes raster. It is important to remark that SHARP, although the last letter of the acronym stands for "Perceptron", does not have anything in common with the classical Perceptron. Thus, the number of layers is not relevant. Actually the number of layers in SHARP is exactly equal to the number of features of the problem (in the Perceptron, this is the number of inputs in the input layer).

10.2. Random Artificial Database Test

In the SHARP test framework, we have created a program that can generate example patterns and related validation patterns (with added noise controlled by a NOISE_LIMIT parameter).

Any component of any vector is pseudo-randomly generated. For any example vector generated, we have a noisy pattern for the validation purpose.

The result is a database of completely random patterns (64 features wide) uniformly distributed to eight classes. Thus, there is no relation between patterns contained in any class, and clusters produced by a Kohonen Map are not distinguishable (Fig. 28).

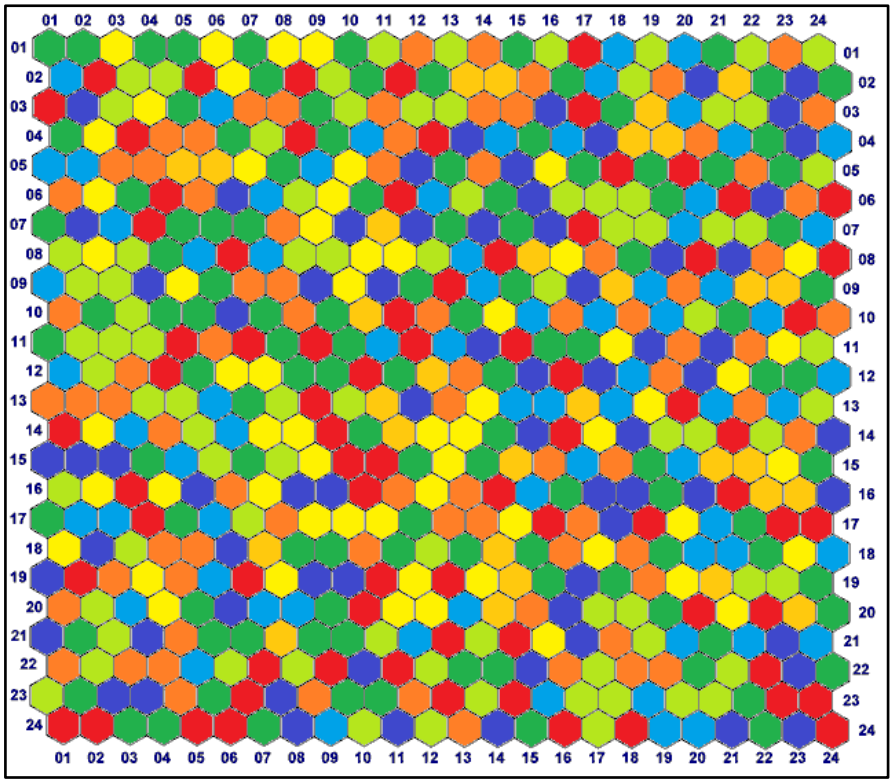

Figure 28. The output obtained by a SOM trained with a subset of points from the random artificial database. The eight colors refer to the eight different classes associated with the original examples. SOM is unable to create class-related clusters, demonstrating the real random nature of the dataset

This test (Table 1) is important to check the crosstalk in the worst condition. The measurability and control of the parameters related to the training and validation databases are essential to validate and measure the capabilities and the limits of the neural network. However, this test is invalid on real-world applications, which are considered mandatory for any new pattern-recognition algorithm.

We have produced the following test conditions:

TEST_1: 5000 training examples + 5000 validation with $\mu = 5$

TEST_2: 5000 training examples + 5000 validation with $\mu = 10$

TEST_3: 5000 training examples + 5000 validation with $\mu = 15$

Here μ is the maximum noise added to any component of the vector

$$v[n] = v[n] + (-\mu < \xi < \mu) \tag{73}$$

The sign of noise is chosen randomly.

TEST_1, TEST_2 and TEST_3 are checking the only L_SUP distance generalization. Other tests (not reported here) have been done considering also the generalization L_NUM, defined as:

$$L\_NUM = \sum_0^n \chi_n$$

$$\begin{cases} \chi_n = 0 \because (\upsilon_n > \nu_n - \mu) \wedge (\upsilon_n < \nu_n + \mu) \\ \chi_n = 1 \because (\upsilon_n \le \nu_n - \mu) \vee (\upsilon_n \ge \nu_n + \mu) \end{cases}$$
$$\nu = training\_vector\_component$$
$$\upsilon = validation\_vector\_component \tag{74}$$
$$m = total\_components\_number$$
$$n = component\_index$$

L_NUM defines the number of features outside the L_SUP limits that can be accepted to consider the pattern matching with the prototype.

10.3. Complex Artificial Database Test

In short, this database differs from the previous one because although patterns are still randomly generated, they are filtered with complex, relational, features-range rules that are associated with specific classes. In other words, these filters accept or reject the randomly generated number depending if it satisfies the rule associated with the current feature and the current class. The rule can be complex, as it can assert multiple ranges whose validity is a function of all or some values assumed by previously generated features.

Table 1. The table shows the results of the comparative test between SHARP and RCE with a training database of 5000 pseudo-randomly generated patterns. The validation database is composed of patterns derived from the learning patterns, adding noise on any component limited by μ ($c - \mu < c' < c + \mu$). ID = correctly identified; UNC = identified with uncertainty; NID = not identified; WID = wrongly identified. TLT = Total Learning Time; TRT = Total Recognition Time; APLT = Averaged Pattern Learning Time; APRT = Averaged Pattern Recognition Time; CPN = Committed Prototypes Number (applicable on RCE). All computed times are related to the effective time required by the learning/recognition algorithm: the overhead related to all of the other operations required (read / write / preprocessing) that are congruent for both the neural paradigms have been excluded. The test has been executed on an Intel® Core™ i5-3320M CPU @ 2.60 GHz with 4 GB RAM

		SHARP Nif=5(μ =5) Nif=10(μ =10)	**RCE (LSUP)** Minif=0	
			Maxif μ	Maxif 90
RECALL	ID	5000	5000	5000
	UNC	0	0	0
	NID	0	0	0
	WID	0	0	0
VALIDATION μ =5 (noise up to 10% on components)	ID	5000	5000	3630
	UNC	0	0	1370
	NID	0	0	0
	0	0	0	0
VALIDATION μ =10 (noise up to 20% on components)	ID	0	0	3388
	UNC	0	0	1611
	NID	0	0	0
	WID	0	0	0
VALIDATION μ =15 (noise up to 30% on components)	ID	4350	5000	3353
	UNC	349	0	1646
	NID	0	0	0
	WID	300	0	0
TLT (1 CYCLE)		52 mS/155 mS	188 mS	160 mS
TRT		1 mS	107 mS	100 mS
APLT		10 µS/30 µS	38 µS	32 µS
APRT		200 nS	20 µS	20 µS
CPN		NA	5000	4669
CYCLES		1(NA)	1	6

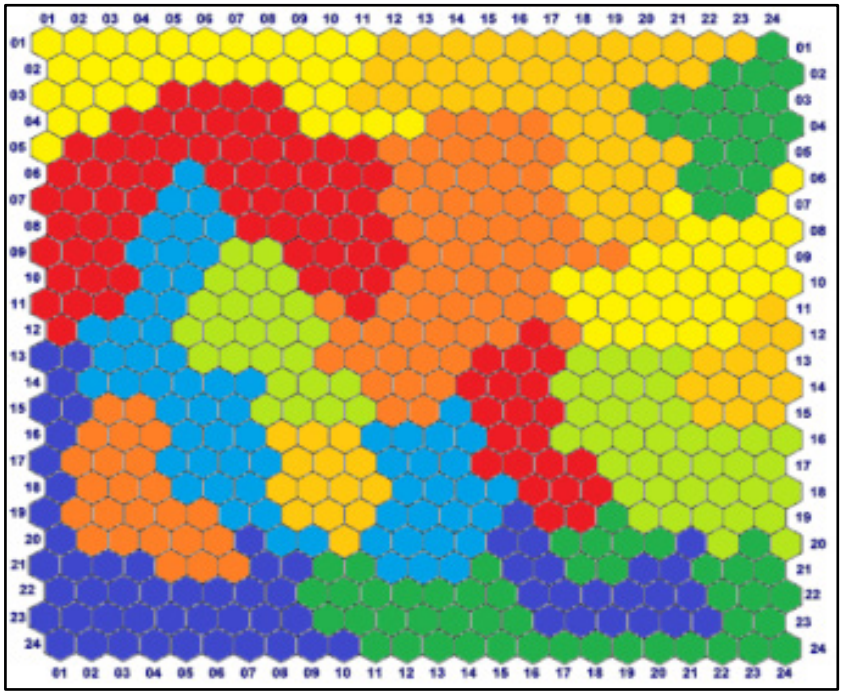

Figure 29. The SOM representation of the complex database generated with the tool. All patterns have been distributed between eight classes with similarities between couples of classes

Table 2. The table shows the results of the comparative test between SHARP and RCE with a training database of 10000 patterns pseudo-randomly generated with constraints rules for any component and category. The validation database is composed of patterns derived from the learning patterns, adding noise on any component limited by μ (c – μ < c' < c + μ). ID = correctly identified; UNC = identified with uncertainty; NID = not identified; WID = wrongly identified. TLT = Total Learning Time; TRT = Total Recognition Time; APLT = Averaged Pattern Learning Time; APRT = Averaged Pattern Recognition Time; CPN = Committed Prototypes Number (applicable on RCE). All computed times are related to the effective time required by the learning/recognition algorithm: the overhead related to all of the other operations required (read / write / preprocessing) that are congruent for both the neural paradigms have been excluded. The test has been executed on an Intel® Core™ i5-3320M CPU @ 2.60 GHz with 4 GB RAM

		SHARP Nif=5(μ=5)Nif=10(μ=10)	**RCE (LSUP)** Minif=0	
			Maxif μ	Maxif 90
RECALL	ID	10000	10000	10000
	UNC	0	0	0
	NID	0	0	0
	WID	0	0	0
VALIDATION μ=5 (noise up to 10% on components)	ID	10000	10000	7267
	UNC	0	0	2732
	NID	0	0	0
	WID	0	0	0
VALIDATION μ=10 (noise up to 20% on components)	ID	10000	10000	6812
	UNC	0	0	3187
	NID	0	0	0
	WID	0	0	0
VALIDATION μ=15 (noise up to 30% on components)	ID	7809	10000	6764
	UNC	1147	0	3235
	NID	0	0	0
	WID	1043	0	0
TLT (1 CYCLE)		67 mS/216 mS	340 mS	320 mS
TRT		2.3 mS	400 mS	390mS
APLT		11 μS/22 mS	34 μS	32 μS
APRT		230 nS	40 μS	39 μS
CPN		NA	10000	9341
CYCLES		1(NA)	1	6

This tool can generate complex databases of patterns associated with multiple classes that can have any arbitrary shape in the space of variables (features). Although the completely random database is the hardest problem for a classifier, a database generated with this system represents a more real classification problem whose complexity can be easily varied (Fig. 29).

As in the case of the completely random database, we have patterns composed of 64 features, and any feature is in a 0-127 range mapped on one byte. The noisy patterns that are used for validation are generated in the same way as the learning patterns. To make this last measurement, we have trained both networks with the same increasing subsets of the examples contained in the entire database.

RCE increases the number of prototypes to learn new patterns, which leads to an increase in execution time of the recognition phase because prototypes must be sequentially compared with the input pattern in a Von Neumann machine. In contrast, SHARP creates a number of prototypes equal to the number of classes that must be managed resulting in recognition time that is stable and independent of the quantity of learned patterns (Table 2).

$$
\begin{aligned}
&step\_1: \\
&v = rand() \\
&step\_2(simple): \\
&\because (\lambda_k^{cn} < v < \Lambda_k^{cn}) \therefore v_k^m = v \\
&step\_2(back\_correlated): \\
&\because (\lambda_k^{cn} < v < \Lambda_k^{cn}) \wedge (\lambda_{k-x}^{cn'} < v_{k-x}^m < \Lambda_{k-x}^{cn'}) \\
&\therefore v_k^m = v \\
&v = pseudo\_random\_value \\
&\lambda = lower\_\lim \\
&\Lambda = higher\_\lim \\
&c = class \\
&m = pattern\_index \\
&k = component\_index \\
&n = range\_index
\end{aligned}
\tag{75}
$$

In the simple step 2, the value assigned to a specific component indexed by *k* must be simply inside the range indexed by n associated with class *c*.

In the back-correlated step 2, the value assigned to a specific component, indexed by *k,* must be inside the range, indexed by *n,* associated with class *c* when the value assigned to the component indexed by *k-x* is inside the range indexed by *n'* associated with class *c.*

10.4. Circle in the Square Test

This test, originally proposed by DARPA to check the capability of neural networks, seems to be a "toy problem"; however, it is still a valid method to discover lacks and limits of neural networks in pattern recognition tasks. This problem requires a system to identify which points of a square lie inside and which points lie outside a circle whose area is half that of the square. The intrinsic bi-dimensionality of the problem offers an easily interpretable image of results that would be hidden in a more complex problem.

We have compared SHARP with a RCE network. Sometimes, this test can discover conceptual lacks of a pattern recognition algorithm that could not be discovered with more complex tasks. The RCE neural network creates prototypes with large NIF when they are far from edges with a different class (fig. 31). The process subsequent to a "mismatch" condition reduces the NIF of existing prototypes. The NIF of a new prototype is limited by the distance of the nearest prototype owned by a different class. Thus, the number of prototypes grows as a function of the precision requested to recognize the correct class around the points located in the circumference. Few prototypes can roughly recognize points inside and outside the circle.

The SHARP neural network, because of its "agnostic resonance", does not have a mismatch mechanism or a NIF property as intended in RCE. When a pattern is learned, some synapses between any layer and all of the following layers are activated. For any layer representing a feature, a certain number of neurons around the most resonating neuron are involved in the process of synaptic update. The dimension of the region related to synaptic update around the most resonating neuron, is equivalent to NIF in L_SUP mode for an RCE neural network. The SHARP model has prototypes distributed on the entire set of feature layers as a chain of synapses activated by a "systolic Hebb" learning process. These "systolic prototypes" can be overlapped (Fig. 30), so an input pattern can activate more than one category. The synapses that interconnect subsequent feature clusters and synapses that connect the feature layers with the category clusters are analogically activated as a function of the distance from the most resonating neuron. In this way, the contributions to activation of a category neuron is weighted, and conflicting recognitions almost always have different strengths. The WTA mechanism in the category layer substitutes at run-time the "mismatch process" of the RCE, resolving confliction recognitions.

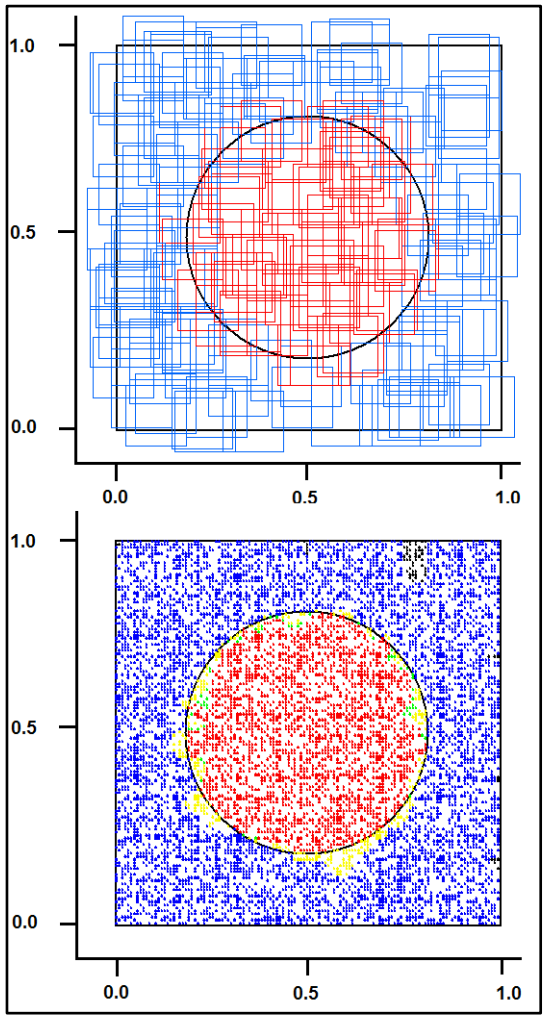

Figure 30. The result of "Circle in the square test" with SHARP. On the upside: entangled prototypes (200 examples learned). On the downside: 10,000 points recognized (yellow/green = uncertainty/error, black(except contour) = not identified)

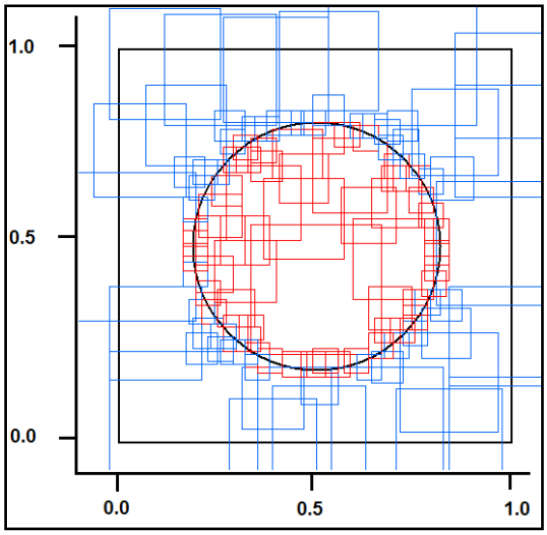

Figure 31. The prototypes for the "Circle in the square" test of a RCE neural network in L_SUP mode

11. Technological Considerations

SHARP can perform one shot learning and recognition by simply addressing synaptic values stored in memory. In this paragraph, we analyze the performance of the network compared with a RCE-like algorithm in a pattern recognition problem.

The main problem of RCE and RBF networks is that their algorithms must compare the input pattern with all of the prototypes stored in their database. This operation can be very time consuming when the number of prototypes is huge and the algorithm is executed in a serial computer. In this case, a real-time performance requires the use of SIMD processors. Such processors can compute, in a parallel way, the Euclidean distance or L1 distance between the input pattern and many prototypes. The number of prototypes that can be analyzed at the same time depends on the processor type and its scalability, as most of them can be connected in daisy chain. There are also neural chips implementing neural networks in a RBF-like manner, which can be connected in a daisy chain. Most chips of this type are laboratory units whereas the *Cognimem CM1K* is a commercial version that embeds 1000 prototypes and allows many chips to be connected in a daisy chain.

We have proposed an algorithm that can be executed in real time in a serial computer completing the recognition in a time that is independent of the number of prototypes stored in the network (Fig. 32). Furthermore, the software simulation algorithm requires only sequential readings of synaptic values from memory in lieu of complex computations. The biggest demand of resources is only related to the memory required to store synaptic values. The number of "systolic synapses" between sequential feature clusters grows rapidly (on a factorial basis) with the number of features managed by the network. The synapses between feature clusters and category clusters grow linearly with the number of features and the number of classes.

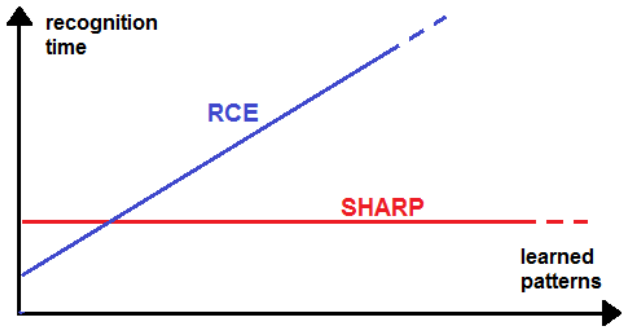

Figure 32. The recognition time required by SHARP and that required by an RCE in a Von Neumann computer

The current size of RAM and FLASH memories enables the implementation of huge neural networks based on the SHARP architecture. This architecture seems to be well aligned with technology trends of the last 20-30 years. Memory size in the last 20 years has increased almost 1000 times faster than processor clock speed, and it appears that this trend will continue for the next several years (Fig. 33). This means that memory based algorithms (i.e., using LUT) are advantageous over high computation algorithms.

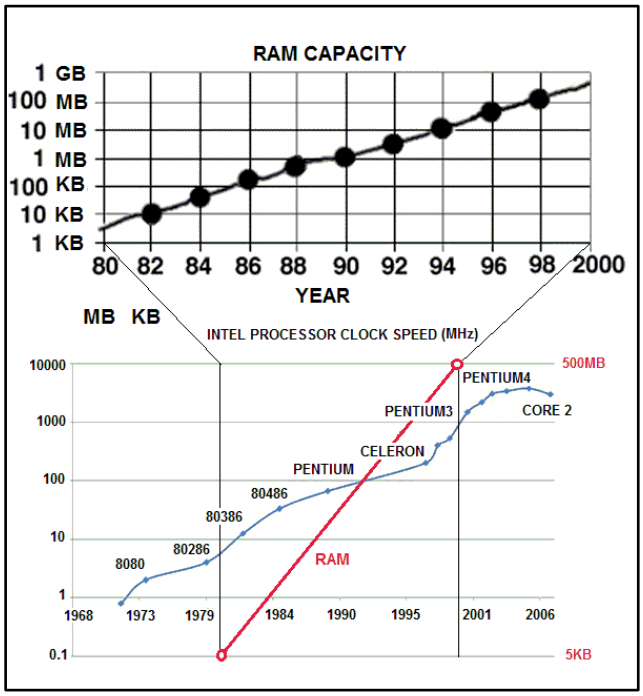

Figure 33. The growth of clock frequency and the growth of ram capacity in the last twenty years. The growth of memory has been 1000 times the growth of clock frequency

The resources in terms of memory space required to implement SHARP (Table 3) on a conventional processor can be computed using the following formulas:

$$D(\Phi) = C^2 \times V^2 \times F^2 \tag{76}$$

$$D(\Gamma) = V^2 \times F \tag{77}$$

$$D(\Psi) = C \times V \times F \tag{78}$$

$$D(\Psi^{\downarrow}) = C \times V \times F \tag{79}$$

$$\begin{aligned} N &= C \times (V \times F \times 2 + 2 + L) \\ C &= categories \\ F &= features \\ V &= values \\ N &= neurons \\ L &= category\_cluster\_neurons \end{aligned} \tag{80}$$

11.1. Parallel Hardware Implementation

SHARP can be efficiently implemented on FPGA (Field Programmable Gate Array) or multiple instances of a very simple microcontroller together with distributed fast FRAM (Ferromagnetic Random Access Memory). We have designed, at a higher level, a hardware platform intended to maximize parallelization of the recognition process. We

have one FRAM bank, for any couple of features, which stores synaptic connections between couples of layers. Actually, this FRAM stores synaptic connections between the neurons associated to particular features and neurons associated to other features.

The project has been optimized to demonstrate that by implementing SHARP in hardware, it is possible to obtain fully parallel huge neural networks with very limited digital electronic resources.

The principal constraint in parallelizing the process is the distribution of memories dedicated to a specific feature or group of features. The management of memories could be implemented in a single large FPGA or one small FPGA (or CPLD) / uP for any memory block.

The other solution is using a single, very simple uP or CPLD for any FRAM (or FLASH) block managing one feature or a small group of features. The result is a scalable multi-modules system that can be adapted to the pattern dimensionality.

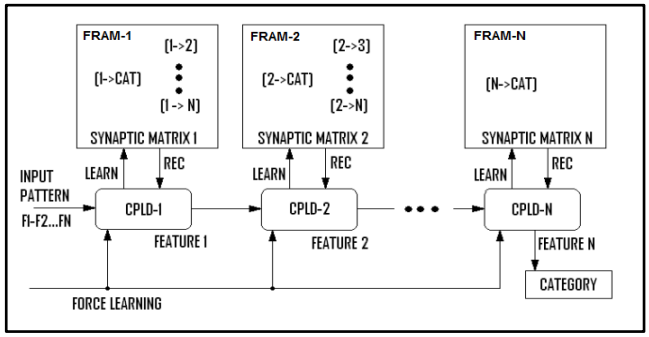

Figure 34. The hardware implementation with stages composed of a CPLD and a flash memory. In this picture, each stage manages one feature; thus, the parallelism and the speed performance are maximized

The architecture is shown in Fig. 34, where it is possible to distinguish the distribution of operations and synapses management between processing elements. In this system, the features are fed serially in the systolic chain of the modules composed of one CPLD and one FRAM memory. This type of hardware implementation is proposed as a low-cost, scalable solution for a cortical processor.

Table 3. The matrices of synapses needed for a simplified implementation of SHARP on a conventional computer. It also shows the typology of synapse and the type size required to memorize one element of the matrix. Fixed (not plastic) synapses can be represented by a single constant value for any matrix

SYNAPSE MATRIX	TYPE	SIZE
Φ	Excitatory / Plastic / Feed-Forward	Byte
Γ	Inhibitory / Fixed / Intra-Feature	Byte
Ψ	Excitatory / Plastic / Bottom-Up	Byte
Ψ↓	Excitatory / Fixed / Top-Bottom	Byte

12. Conclusions and Future Work

We have presented a novel, biology-inspired neural network model that integrates the firing of neurons.

We have analyzed the rough biological plausibility of the described paradigm and its spiking neurons models. Then, we have analyzed the issues related to its digital software implementation. Our target has been to find an algorithm respecting the architectural and behavioral characteristics of the proposed paradigm, but that could be very efficiently executed in a typical Von Newman machine. This model is strongly based on the use of memory, but it requires low or almost null computational power. The algorithm is executed in real time on Von Newman processors with performance independent of the number of stored prototypes. In this paper, we have analyzed how the proposed algorithm can be considered a "brick" of a more complex hierarchical structure, considering many other potential analogies with biological systems.

We have compared the performance of this algorithm with the classical RCE and RBF. We have also analyzed the project of a possible implementation of such algorithm on commercial hardware to maximize the recognition speed. Currently, we are working on algorithms that could automate the PCS. Furthermore, we plan to work on cognitive systems built on the MOSAIC framework with SHARP modules. The design of a scalable, digital, asynchronous chip working as a "cortical processor" will be the second target of future research.

REFERENCES

[1] Nowotny T, Huerta R, Abarbanel HD, Rabinovich MI. Self-organization in the olfactory system: one shot odor recognition in insects. BiolCybern. 93(6):436-46, 2005. (2003).

[2] Sachse, S. and C. G. Galizia (2002). "Role of Inhibition for Temporal and Spatial Odor Representation in Olfactory Output Neurons: A Calcium Imaging Study." J Neurophysiol 87(2): 1106-1117.

[3] Paolo Arena, Luca Patané, Pietro Savio Termini (2012): Learning expectation in insects: A recurrent spiking neural model for spatio-temporal representation. Neural Networks 32: 35-45 (2012).

[4] Izhikevich E.M. (2003) Simple Model of Spiking Neurons. IEEE Transactions on Neural Networks, 14:1569- 1572.

[5] Izhikevich E.M. (2001) Resonate-and-Fire Neurons. Neural Networks, 14:883-894.

[6] *Llinas RR* 1988 - The intrinsic electrophysiological properties of mammalian neurons: insights into central nervous system functionDec 23;242(4886):1654-1664, Science— id: 9930, year: 1988, vol: 242, page: 1654, stat: Journal Article.

[7] W. Maass. (1996) On the computational power of noisy spiking neurons. InAdvances in Neural Information Processing Systems, D. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, volume 8, pages 211-217. MIT Press (Cambridge), 1996.

[8] W. Maass. (1997) Networks of spiking neurons: the third

generation of neural network models. Neural Networks, 10:1659-1671, 1997.

[9] Fred Rieke, David Warland, Rob de Ruyter van Steveninck, William Bialek (1999) -Spikes: Exploring the Neural Code - Bradford Book.

[10] J. Gautrais, S. Thorpe (1998) Rate coding versus temporal order coding: a theoretical approachBiosystems 48 (1), 57-65.

[11] Bressler S.L. (1995) Large-scale cortical networks andcognition. Brain Res Rev 20:288-304.

[12] Bressler S.L. (2002) Understanding cognition through large-scale cortical networks. Curr Dir Psychol Sci 11:58-61.

[13] Bressler S.L., Tognoli E. (2006) Operational principles of neurocognitive networks. Int J Psychophysiol 60:139-148.

[14] Cerf et al (2010), On-line, voluntary control of human temporal lobe neurons. Nature, October 28, 2010. doi:10.1038/nature09510.

[15] Gelbard-Sagiv et al. 2008Internally Generated Reactivation of Single Neurons in Human Hippocampus During Free Recall-Published Online September 4 2008Science-3-October-2008: Vol. 322 no. 5898 pp. 96-101 - DOI: 10.1126/science.1164685.

[16] Kreiman, Koch, Fried (2000) - Category-specific visual responses of single neurons in the human medial temporal lobe - Nature Neuroscience3, 946 - 953 (2000) - doi:10.1038/ 78868.

[17] Quian Quiroga, R., Reddy, L., Kreiman, G., Koch, C. & Fried, I. (2005).Invariant visual representation by single neurons in the human brain. Nature, 435:1102–1107.

[18] Quian Quiroga, R., Kreiman, G., Koch, C. & Fried, I. (2008). Sparse but not "Grandmother-cell" coding in the medial temporal lobe. Trends in Cognitive Science, 12, 3, 87–94.

[19] Quian Quiroga, R., Kraskov, A., Koch, C., & Fried, I. (2009). Explicit Encoding of Multimodal Percepts by Single Neurons in the Human Brain.*Current Biology*, 19, 1308–1313.

[20] Quian Quiroga, R. & Kreiman, G. (2010a). Measuring sparseness in the brain: Comment on Bowers (2009). Psychological Review, 117, 1, 291–297.

[21] Quian Quiroga, R. & Kreiman, G. (2010b). Postscript: About Grandmother Cells and Jennifer Aniston Neurons. Psychological Review, 117, 1, 297–299.

[22] Viskontas, I., Quian Quiroga, R. & Fried, I. (2009). Human medial temporal lobe neurons respond preferentially to personally relevant images. Proceedings of the National Academy Sciences, 106, 50, 21329-21334.

[23] Asim Roy (2012) Discovery of Concept Cells in the Human Brain – CouldIt Change Our Science? –Natural Intelligence Vol.1 Issue.1 INNS magazine.

[24] Barlow, H. (1972). Single units and sensation: A neuron doctrine for perceptual psychology. Perception, 1, 371–394.

[25] Barlow, H. (1995). The neuron doctrine in perception. In The cognitive neurosciences, M. Gazzaniga ed., 415–436. MIT Press, Cambridge, MA.

[26] Gross, C. (2002). Genealogy of the grandmother cell. The Neuroscientist, 8, 512–518.

[27] Baars, Bernard J. (1988), A Cognitive Theory of Consciousness (Cambridge, MA: Cambridge University Press).

[28] Baars, Bernard J.(1997), In the Theater of Consciousness(New York, NY: Oxford University Press).

[29] Baars, Bernard J. (2002) The conscious access hypothesis: Origins and recent evidence. Trends in Cognitive Sciences, 6 (1), 47-52.

[30] J.A. Reggia – Neural Networks, 2013 – Elsevier The rise of machine consciousness: Studying consciousness with computational models.

[31] Baars, B. J., & Franklin, S. An architectural model of conscious and unconscious brain functions: Global Workspace Theory and IDA. Neural Networks (2007), doi:10.1016/j.neunet.2007.09.013.

[32] H.T Kung. Why Systolic Architectures? January 1982 37 c3 1982 IEEE.