Review of Stability Properties of Neural Plasticity Rules for Implementation on Memristive Neuromorphic Hardware

Zlatko Vasilkoski, Heather Ames, Ben Chandler, Anatoli Gorchetchnikov, Jasmin Léveillé, Gennady Livitz, Ennio Mingolla and Massimiliano Versace

***Abstract*— In the foreseeable future, synergistic advances in high-density memristive memory, scalable and massively parallel hardware, and neural network research will enable modelers to design large-scale, adaptive neural systems to support complex behaviors in virtual and robotic agents. A large variety of learning rules have been proposed in the literature to explain how neural activity shapes synaptic connections to support adaptive behavior. A generalized parametrizable form for many of these rules is proposed in a satellite paper in this volume [1]. Implementation of these rules in hardware raises a concern about the stability of memories created by these rules when the learning proceeds continuously and affects the performance in a network controlling freely-behaving agents. This paper can serve as a reference document as it summarizes in a concise way using a uniform notation the stability properties of the rules that are covered by the general form in [1].**

I. Introduction

RECENT advances in computational power and memory capacity lead to the realization of large-scale artificial neural systems subserving perception, cognition, and learning on biological temporal and spatial scales. One example of such technologies is under development by Hewlett-Packard in collaboration with Boston University in the DARPA-sponsored "Systems of Neuromorphic Adaptive Plastic Scalable Electronics" (SyNAPSE) initiative. SyNAPSE seeks hardware solutions that reduce power consumption by electronic "synapses" in order to achieve memory density of 10^{15} bits per square centimeter. One approach is based on memristive devices. The memristor, initially theorized by Chua [2] and later discovered by HP Labs [3] is a nanoscale element that has a unique property of "remembering" the past history of its stimulation in its resistive state and does not require power to maintain its memory. These properties make memristors ideal candidates to implement dense, low-power synapses supporting large-scale neural models.

As the design of a dense memristive memory is still work in progress Hewlett-Packard in collaboration with Boston University developed a software package *Cog Ex Machina* [4] that includes a hardware abstraction layer and decouples neuronal modeling from hardware implementations of these models. This allows to prepare large scale models, test them on currently available hardware, for example on a cluster of Graphic Processing Units (GPUs), and run them on memristor-based hardware as soon as it will become available.

One of the goals of SyNAPSE project is to enable the creation of whole-brain system to support real-time behavior in virtual and robotic agents. The MoNETA (Modular Neural Exploring Traveling Agent; [5]) project of the Neuromorphics Lab at Boston University is an example of such system. In the process of MoNETA development it became clear that the current state of our knowledge about the brain does not allow to decide a-priori which of the many learning rules that were designed in the past decades will be necessary for the complete system. A proposed solution was to design a general form of a learning rule that can be parametrized to implement many individual rules of several important classes [1]. This work would not be complete without a concise review and summary of the properties of all the rules that can be implemented within this general form. The purpose of this paper is to provide such a summary.

A hardware-based autonomous agent has to operate under multiple constraints. It is hard to predict in dynamic environments when the learning is necessary and when it might become detrimental to performance, so the successful implementation should have learning constantly enabled, but operating using the set of rules that does not lead the resulting synaptic weight matrix towards saturation at either zeros or maximal weight values, the rules that preserve usable dynamic range. Most of the rules presented below and in [1] have been extensively analyzed previously either during their introduction or in subsequent publications. Nevertheless, we consider it important to have a concise summary of these analyses in a single source and with consistent notation, and this is the goal of this paper. Section II presents the three classes of rules analyzed, Section III presents the stability analysis and usability recommendations, and Section IV concludes this work.

II. Learning Rules

We analyze three basic classes of learning rules: variations of basic Hebb rule with different decay gatings, variable threshold-based rules that maintain the time-average of the

This work was partially funded by the DARPA SyNAPSE program, contract HR0011-09-3-0001. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense. M.V., E.M., H.A., and B.C. were also supported in part by CELEST, a National Science Foundation Science of Learning Center (NSF SBE-0354378 and NSF OMA-0835976).

A.G., M.V., H.A., B.C., J.L., G.L., and E.M. are with the Department of Cognitive and Neural Systems and Center of Excellence for Learning in Education, Science, and Technology, Boston University, 677 Beacon Street, Boston, MA 02215, USA (e-mail: {anatoli, versace, starfly, bchandle, jasminl, glivitz, ennio}@cns.bu.edu).

Z.V. is with Harvard Medical School, Harvard University, Cambridge, MA, USA (e-mail: zlatko_vasilkoski@hms.harvard.edu).

TABLE I

HEBB RULE DERIVATIVES USED IN FIRING RATE-BASED MODELS, WHERE α IS A DECAY RATE.

Name	Equation
Classic Hebb	$\dot{w_i} = \eta x_i y$
Hebb + passive decay	$\dot{w_i} = \eta x_i y - \alpha w_i$
Pre-synaptically gated decay (outstar) [7]	$\dot{w_i} = \eta x_i y - \alpha x_i w_i$
Post-synaptically gated decay (instar) [7]	$\dot{w_i} = \eta x_i y - \alpha y w_i$
Oja [8]	$\dot{w_i} = \eta x_i y - \alpha y^2 w_i$
Dual OR [9]	$\dot{w_i} = \eta x_i y - \alpha(x_i + y) w_i$
Dual AND	$\dot{w_i} = \eta x_i y - \alpha x_i y w_i$

TABLE II

THRESHOLD-BASED RULE VARIATIONS, WHERE ϵ IS THE RATE OF THRESHOLD DYNAMICS AND σ' IS A DERIVATIVE OF THE SIGMOIDAL ACTIVATION FUNCTION IN [12].

Name	Equation 1	Equation 2
Covariance 1	$\dot{w_i} = \eta x_i (y - \theta_y)$	$\dot{\theta}_y = \epsilon(y - \theta_y)$
Covariance 2	$\dot{w_i} = \eta (x_i - \theta_{x_i}) y$	$\dot{\theta}_{x_i} = \epsilon(x_i - \theta_{x_i})$
Original BCM [11]	$\dot{w_i} = \eta(y - \theta_y) x_i y - -\alpha w_i$	$\dot{\theta}_y = \frac{y}{\epsilon} - \theta_y$
iBCM [12]	$\dot{w_i} = \eta(y - \theta_y) x_i y \sigma'(y)$	$\dot{\theta}_y = \epsilon(y^2 - \theta_y)$
lBCM [13]	$\dot{w_i} = \eta(y - \theta_y) x_i \frac{y}{\theta_y}$	$\dot{\theta}_y = \epsilon(y^2 - \theta_y)$
Textbook BCM [10]	$\dot{w_i} = \eta(y - \theta_y) x_i y$	$\dot{\theta}_y = \epsilon(y^2 - \theta_y)$

activation, and explicit temporal trace-based rules. The general assumptions for this analysis are the linear neuron assumption

$$y = \sum_{i=1}^{m} w_i x_i = X^T W \quad (1)$$

where y is the postsynaptic cell activity, $x_i \geq 0$ is activity of the presynaptic cell i , and w_i is the synaptic weight between these two cells.

A. Hebb-based Rules

In Hebbs rule [6] there is a direct dependency between synaptic plasticity and repeated and persistent stimulation of a postsynaptic cell by a presynaptic cell, which is usually expressed as a weight change being proportional to a correlation between the presynaptic and the postsynaptic activities. All the rules discussed in this section are based on this fundamental principle and are widely applied in continuous firing rate dynamical models expressed in differential form. A general form for Hebbian-based rules with gated decay is:

$$\dot{w_i} = \eta x_i y - f(x_i, y) w_i \quad (2)$$

where η is the learning rate, and $f()$ is a decay gating function. In this equation the first term is the Hebbian or correlation term and the second term is the decay term that helps to normalize the weights and sets the equilibrium limits for the weight values. Table I provides variations of this rule analyzed in this paper.

B. Threshold Based Rules

The primary difference between variable threshold based rules and Hebbian rules is the introduction of a threshold in considering the influence of cell activities on learning. While the Hebbian correlation term leads to an increase in weight whenever both presynaptic and postsynaptic activities are positive, this is not always the case for threshold-based rules. These rules are set up so that if the activity of the cell is higher than a certain threshold then the weight increases, but if it is lower than the threshold then the weight decreases. Effectively, the Hebbian weight change is proportional to $x_i y$, while threshold-based weight change is proportional to $x_i(y - \theta_y)$ or $(x_i - \theta_{x_i})y$. This class of learning rules include covariance rules (e.g. [10]) and multiple variations of the BCM rule (e.g. [11]–[13]). These rules commonly include a second equation governing the dynamics of the variable threshold $\theta_{x,y}$:

$$\dot{\theta}_{x_i,y} = g(\theta_{x_i,y}, x_i, y) \quad (3)$$

Learning rules based on such thresholds are then formulated as a system of two equations, one with faster dynamics (threshold in equation (3)), and another with slower dynamics (synaptic weight change). Table II presents the summary of the variable threshold rules analyzed here.

C. Temporal Trace Based Rules

This class of learning rules includes rules that require the maintenance of an explicit temporal trace to calculate the synaptic weight change. These rules appear in the reinforcement learning literature, with the Temporal Difference (TD) rule [14] being a classic example:

$$\Delta w_i = \eta x_i(t-1)(r(t) - y(t-1) + \gamma y(t)) \quad (4)$$

Under conditions of $\Delta t \rightarrow 0$ and $\gamma = 0$ this rule is reduced to the Rescorla-Wagner learning equation [15]:

$$\dot{w_i} = \eta(r - y) x_i \quad (5)$$

In both cases r is the actual reward and y is a predicted reward signaled by a postsynaptic cell. Since in both of these rules the true reward is compared with the prediction of rewards, these rules can also be considered as cases of a delta rule.

Földiák [16], [17] proposed a modified Hebbian rule where learning is proportional to the product of the presynaptic activity and the trace of the postsynaptic activity, with the postsynaptically-gated decay also being based on the trace:

$$\Delta w_i = \eta \theta(t)(x_i(t) - w_i(t)) \quad (6)$$

where

$$\theta(t) = (1 - \epsilon)\theta(t-1) + \epsilon y(t) \quad (7)$$

With $\Delta t \rightarrow 0$ the latter equation reduces to the differential form $\dot{\theta}_y = \epsilon(y - \theta_y)$ and the rule shows some similarity to the threshold based rules, although the weight update is dependent on the trace of activation rather then on thresholded

TABLE III

TEMPORAL TRACE RULE VARIATIONS, WHERE ϵ IS THE RATE OF THRESHOLD DYNAMICS AND γ IS AN INFLUENCE OF FUTURE REWARDS [14].

Name	Equation 1
TD [14]	$\Delta w_i = \eta x_i(t-1)(r(t) - y(t-1) + \gamma y(t))$
Földiák [16], [17]	$\dot{w_i} = \eta\theta_y(x_i - w_i)$ $\dot{\theta}_y = \epsilon(y - \theta_y)$

activation. The trace of the postsynaptic activity has the role of keeping a running average of the "categorization" cell to sample changes in the presynaptic cells, mirroring the fact that that the features to be learned are relatively stable in the environment.

Table III presents the summary of the variable threshold rules analyzed here.

III. STABILITY ANALYSIS

A. Hebb-based Rules

The left side of the first order system of differential equations encountered in Table I, written in a matrix form as $\dot{W} = AW$, expresses the change (growth or decay) of the synaptic weights while the matrix A on the right hand side gives the direction (sign) of this change. For stability it is necessary that the eigenvalues of A are non-positive. Thus, one way of analyzing stability of this system is by finding the eigenvalues λ_i of the matrix A and then analyzing the circumstances that lead to non-positive eigenvalues. Another, more general way of stability analysis can be done by analyzing the system of first order equations such as $\dot{W} = F(W)$ at the fixed points by setting $\dot{W} = 0$, and then check the stability by analyzing the change of the function at the fixed point.

1. $\dot{w_i} = \eta x_i y$. This rule is well known to be unstable; the following analysis is provided for completeness. The matrix form of this learning rule is $\dot{W} = \eta X y = \eta X X^T W = \eta C W$. Here $C = XX^T$ is a covariance matrix, which is positive semidefinite and thus has all positive eigenvalues. The synaptic weights W are exponentially diverging for any values of the column vector of pre-synaptic inputs. The growth of W is determined by the principle component (the eigenvector that corresponds to the largest eigenvalue) of the covariance matrix $C = XX^T$. Thus this rule extracts the first principle component of the input data, but the norm ($||W||^2$) grows without bounds. The usual mechanism of external normalization to correct this issue is not applicable for the low power hardware implementation because it requires the information collection from all synapses and thus extensive wiring and corresponding losses. The above analysis assumes a positive learning rate η. A learning rule with $\eta < 0$ is usually referred to anti-Hebbian because it weakens the synapse if pre- and post-synaptic neurons are active simultaneously, a behavior that is contrary to that postulated by Hebb. In this case the weighs decay to 0, which is of no practical interest.

2. $\dot{w_i} = \eta x_i y - \alpha w_i$. The matrix form of this learning rule is $\dot{W} = \eta X y - \alpha I W = (\eta X X^T - \alpha I) W = (\eta C - \alpha I) W$. As in the previous case the stability analysis is based on the sign of the eigenvalues of matrix $A = \eta C - \alpha I$. The single nonzero eigenvalue of this matrix is $\lambda = x_1^2 + ... + x_m^2$. For this learning rule the following three distinct cases need to be considered:

a) Synaptic weights W are exponentially converging to zero for $\lambda = \sum x_i^2 < \frac{\alpha}{\eta}$.
b) Synaptic weights are exponentially diverging for $\lambda = \sum x_i^2 > \frac{\alpha}{\eta}$.
c) Synaptic weights are stable for $\lambda = \sum x_i^2 = \frac{\alpha}{\eta}$ and as they are dominated by the largest eigenvalue and its largest eigenvector V, converging to $VV^T W_0$.

The last case could have been interesting if its condition was easy to achieve, but in the majority of dynamic neural networks even if they have intrinsic or explicit normalization they adjust activations and not the squares of activations. As a result the usability of this rule in autonomous agents is questionable.

3. $\dot{w_i} = \eta x_i y - \alpha x_i w_i$. In this learning rule, a change of synaptic weights can only occur if the pre-synaptic neuron is active. The direction of the change is determined by the activity of the post-synaptic neuron. For a positive learning rate η the synapse is strengthened if the post-synaptic cell is reasonably active comparing to the current value of the weight ($\eta y > \alpha w_i$) otherwise it is weakened. For a negative η, the correlation term has a negative sign and the learning rule gives rise to anti-Hebbian plasticity, which has an interesting (albeit useless) stabilizing effect on the post-synaptic firing rate. If the pre-synaptic firing rates are kept constant, the post-synaptic firing rate will finally converge to zero.

The matrix form of this learning rule is $\dot{W} = \eta X y - \alpha X W = (\eta C - \alpha diag(X)) W$. The eigenvalue solution gives the two distinct eigenvalues one of which is $\lambda = \eta(x_1 + ... + x_m) - \alpha$, and all the other ones are $-\alpha$. Assuming a normal decay rate $\alpha > 0$, this gives the stability constraints on X which is $\lambda = \sum x_i \leq \frac{\alpha}{\eta}$. For this learning rule the following three distinct cases need to be considered:

a) Synaptic weights W are exponentially converging to zero for $\lambda = \sum x_i < \frac{\alpha}{\eta}$.
b) Synaptic weights are exponentially diverging for $\lambda = \sum x_i > \frac{\alpha}{\eta}$.
c) For $\lambda = \sum x_i = \frac{\alpha}{\eta}$ there is no contribution from the exponent and if y is calculated by equation (1), then $W(t) = W_0$, so the weights do not change.

The fixed point for this rule is $\frac{\eta}{\alpha} y$. It is a scalar and as a result all the weights to this cell converge toward the same value, which is proportional to the activation of the cell y. Note that there is no general meaning in this convergence unless the activation y is set externally and does not depend on presynaptic activity whether in the form of equation (1) or any other form. Also note that this rule was originally proposed in [7] for situations where indeed the postsynaptic activity is set by external means and the presynaptic activity is normalized so that $\lambda = \sum x_i = \frac{\alpha}{\eta}$. This application can be useful in neuromorphic designs, but the developer shall be cautious that the rule is used exactly as intended. For a negative (anti-Hebbian) learning rate $\eta < 0$ as $t \to \infty$, the W and y are always converging to zero.

4. $\dot{w}_i = \eta x_i y - \alpha y w_i$. The matrix form of this learning rule is $\dot{W} = \eta X y - \alpha y W = (\eta C - \alpha y I)W$, where under the linear neuron assumption, $A(t) = \eta C - \alpha y I$ is time dependent. The time dependence is contained in the scalar term y and an analytical solution can be found:

$$W(t) = \frac{\eta}{\alpha} X \frac{1}{1 + e^{-\eta C t\left(\frac{\eta C}{\alpha X^T W_0} - 1\right)}} \quad (8)$$

From this solution it can be seen that as $t \to \infty$ the exponent goes to zero and the synaptic weights thus go to the fixed point $W \to \frac{\eta}{\alpha} X$. The synaptic weights approach this fixed point whenever the postsynaptic neuron is active. In the stationary state, the set of weight values W reflects the pre-synaptic firing pattern X. In other words, the pre-synaptic firing pattern X is stored in the weights W and scaled by a factor $\frac{\eta}{\alpha}$. This learning rule is an important ingredient of competitive unsupervised learning. It is also the most biologically plausible of the whole family as well as very simple. We would recommend to use this rule wherever possible and only try other rules if this one does not produce a desired learning.

5. $\dot{w}_i = \eta x_i y - \alpha y^2 w_i$. Oja's rule converges asymptotically to synaptic weights that are normalized. The normalization implies competition between the synapses that make connections to the same post-synaptic neuron. If some weights grow others must decrease thus redistributing the synaptic efficacy. The matrix form of this equation is $\dot{W} = \eta X y - \alpha y^2 W = (\eta C - \alpha W^T C W I)W$ where the term $W^T C W$ is a scalar and an analytical solution can be found (see [18]):

$$W(t) = \sqrt{\frac{\eta}{\alpha}} \frac{e^{\eta C t} W_0}{\sqrt{||e^{\eta C t} W_0||^2 - ||W_0||^2 + \frac{\eta}{\alpha}}} \quad (9)$$

It can be seen that as $t \to \infty$ weights converge to the eigenvector V of the covariance matrix corresponding to the largest eigenvalue $\lambda = \sum x_i^2$. This principle eigenvector V of the covariance matrix is a unit vector pointing in the direction of the highest variance in the input space. Since $W \to \sqrt{\frac{\eta}{\alpha}} V$, the output of the neuron becomes proportional to the principle component of the input. The Oja rule converges to the state $||W^2|| = \frac{\eta}{\alpha}$. This can be seen by multiplying the matrix equation by W^T, giving $||W^2|| = \eta y^2 - \alpha ||W^2||$. The fixed point analysis shows that it converges to the state $||W^2|| = \frac{\eta}{\alpha}$. Oja's rule is very useful in the initial stages of pattern recognition systems when the inputs can be discriminated well by their first principal component. In this case a two stage network that first uses Oja's rule and then postsynaptically gated decay can achieve code compression and efficient recognition. If the first principal component of the input does not carry the discriminating power, then the application of Oja's rule is not that useful.

6. $\dot{w}_i = \eta x_i y - \alpha(x_i + y) w_i$. Analysis for this learning rule is similar to the learning rules 3 and 4. The matrix form of this equation is $\dot{W} = \eta X y - \alpha X W = (\eta C - \alpha diag(X) - \alpha y [\vec{e}])W$, where $[\vec{e}]$ is the unit vector. Under the linear neuron assumption, the matrix is time dependent. The time dependence of this learning rule is contained in the scalar term y and is very similar to the Instar or postsynaptically gated rule 4 above. The time domain solution of this rule can be found by replacing C in the Instar learning rule by $Z = C - \frac{\alpha}{\eta} diag(X)$, giving the following analytical solution

$$W(t) = \frac{\eta}{\alpha} \frac{Z W_0}{X^T W_0} \frac{1}{1 + e^{-\eta Z t\left(\frac{\eta Z}{\alpha X^T W_0} - 1\right)}} \quad (10)$$

From the above expression it can be seen that as $t \to \infty$, the exponent goes to zero and the synaptic weights thus go to the fixed point $\frac{\eta X y}{\alpha(X+y)}$. From this it can be seen that the synaptic weights approach the equilibrium under constraint on the pre- and post-synaptic signals of $x_i + y \neq 0$ or $x_i \neq -y$. This is usually true as both X and y are non-negative in most of the networks, and in the case they are both zero there is no change to the synaptic weight. This rule has the stable behavior and at the same time the resulting weights reflect the correlation between the presynaptic and postsynaptic firing. Unlike plain Hebbian rule, this correlation is scaled and favors the high amplitude signals during coactivations of the pre- and postsynaptic cells over the low amplitude signals. For example when both signals are 1, the weighs will settle to 0.5, but when both signals are 0.5, the weights will settle only to 0.25. Similar to the presynaptically gated decay rule, this rule also works better if the postsynaptic activity is driven by inputs other than the presynaptic cell.

7. $\dot{w}_i = \eta x_i y - \alpha x_i y w_i$. The matrix form of this equation is $\dot{W} = \eta X y - \alpha X y W = (\eta C - \alpha diag(X) y)W$, and under the linear neuron assumption, the matrix $A = \eta C - \alpha diag(X) y$ is time dependent. The time dependence of this learning rule is contained in the scalar term y and is very similar to the Instar learning rule. The time domain solution of this rule can be found simply by replacing $\beta = \alpha diag(X)$ in equation (8):

$$W(t) = \frac{\eta}{\beta} X \frac{1}{1 + e^{-\eta C t\left(\frac{\eta C}{\beta X^T W_0} - 1\right)}} \quad (11)$$

As $t \to \infty$ the exponent in equation (11) goes to zero and the synaptic weights converge to the fixed point $\frac{\eta}{\alpha}[\vec{e}]$, where $[\vec{e}]$ is the unit vector. At the same time $||W||^2 \to \frac{\eta^2}{\alpha} m$, where m is the number of inputs (the size of X). Thus the weighs will saturate at $\frac{\eta}{\alpha}$ as soon as there will be enough of overlap between the presynaptic and postsynaptic firing. Without such stimulation the weights do not change. This behavior only appears interesting for the cases when one needs to establish a maximal strength coactivation-based link between two cells independently of the levels of cell activities. Similar to the presynaptically gated decay rule, this rule also works better if the postsynaptic activity is driven by inputs other than the presynaptic cell.

B. Threshold Based Rules

Table II contains systems of two differential equation each. One equation describes the change in the synaptic weight W. The other is governing the dynamics of the variable threshold θ. Note that θ has a slower dynamics than activation variables and represents the running average of the activation over time.

Condition 1. The critical condition for stability of these rules is that the moving threshold θ that stabilizes the synaptic weights, must grow more rapidly than y.

Analytically, the stability of this autonomous pair of equations is analyzed at the equilibrium point $(\bar{W}, \bar{\theta})$ and it is determined by the real part of the eigenvalues of the Jacobian matrix $J(\bar{W}, \bar{\theta})$ obtained from the system. For most of the equations in a matrix form it was more convenient to multiply from the left by X^T and transform equation in terms of $\dot{W}$ to equation in terms of $\dot{y}$.

1. Covariance 1. The matrix form of this system of equations after multiplication by X^T is $\dot{y} = \eta||X||(y - \theta_y)$ and $\dot{\theta}_y = \epsilon(y - \theta_y)$. The equilibrium points are $\bar{y} = \theta_y$ and $\bar{\theta}_y = y$. The Jacobian is

$$J(\bar{y}, \bar{\theta}_y) = \begin{pmatrix} \eta||X|| & -\eta||X|| \\ \epsilon & -\epsilon \end{pmatrix} \tag{12}$$

The stability condition as $y \rightarrow \bar{y}$ is $\frac{\eta}{\epsilon}||X|| < 1$. Therefore to preserve stability the learning rate η shall be slower than $\frac{\epsilon}{||X||}$.

2. Covariance 2. The matrix form of this system of equations is $\dot{W} = \eta(X - \theta_x)X^T W$ and $\dot{\theta}_x = \epsilon(X - \theta_x)$. The equilibrium points are $\bar{W} = 0$ and $\bar{\theta}_x = X$. The equilibrium point stability analysis is inconclusive for this system of equations, but from the main equation it can be seen that the rule is stable for bound X.

3. Original BCM (oBCM). The matrix form of this system of equations after multiplication by X^T is $\dot{y} = \eta||X||(y - \theta_y - \frac{\alpha}{\eta||X||})y$ and $\dot{\theta}_y = \frac{y}{\epsilon} - \theta_y$. The equilibrium points are $\bar{y}_1 = 0$, $\bar{y}_2 = \theta_y + \frac{\alpha}{\eta||X||}$, and $\bar{\theta}_y = \frac{y}{\epsilon}$. To ensure $y \geq 0$ the condition is $\epsilon \geq 1$. The Jacobian for non-trivial second point is

$$J(\bar{y}, \bar{\theta}_y) = \begin{pmatrix} \eta||X||(\theta_y + \frac{\alpha}{\eta||X||}) & -\eta||X||(\theta_y + \frac{\alpha}{\eta||X||}) \\ \frac{1}{\epsilon} & -1 \end{pmatrix} \tag{13}$$

Here $y \rightarrow \frac{\alpha}{\eta||X||(1-\frac{1}{\epsilon})}$. The stability condition is $\alpha > 0$, which is a normal operating regime for this rule. In terms of usability the whole family of BCM rules was designed to explain the formation of stable receptive fields. As these rules push the postsynaptic activity towards the equilibrium value, they shall be used in the networks where the amplitude of the postsynaptic activity does not carry any information.

4. IBCM. The matrix form of this system of equations after multiplication by X^T is $\dot{y} = \eta||X||(y - \theta_y)y\sigma'(y)$ and $\dot{\theta}_y = \epsilon(y^2 - \theta_y)$. The equilibrium points are $\bar{y}_1 = 0$, $\bar{y}_2 = \theta_y$, and $\bar{\theta}_y = y^2$. The Jacobian for non-trivial second point is

$$J(\bar{y}, \bar{\theta}_y) = \begin{pmatrix} \eta||X||\bar{\theta}_y\sigma'(\bar{\theta}_y) & -\eta||X||\bar{\theta}_y\sigma'(\bar{\theta}_y) \\ 2\epsilon\bar{\theta}_y & -\epsilon \end{pmatrix} \tag{14}$$

This system of equations is always stable.

5. LBCM. The matrix form of this system of equations after multiplication by X^T is $\dot{y} = \eta||X||(y - \theta_y)\frac{y}{\theta_y}$ and $\dot{\theta}_y = \epsilon(y^2 - \theta_y)$. The equilibrium points $\bar{y}_1 = 0$, $\bar{y}_2 = \theta_y$, and $\bar{\theta}_y = y^2$. The Jacobian for non-trivial second point is

$$J(\bar{y}, \bar{\theta}_y) = \begin{pmatrix} \eta||X||(2 - \frac{1}{\theta}) & -\eta||X|| \\ 2\epsilon\bar{\theta}_y & -\epsilon \end{pmatrix} \tag{15}$$

The stability condition is $\frac{\eta}{\epsilon}||X|| < 1$.

6. Textbook BCM. The matrix form of this system of equations after multiplication by X^T is $\dot{y} = \eta||X||(y - \theta_y)y$ and $\dot{\theta}_y = \epsilon(y^2 - \theta_y)$. The equilibrium points are $\bar{y}_1 = 0$, $\bar{y}_2 = \theta_y$, and $\bar{\theta}_y = y^2$. The Jacobian for non-trivial second point is

$$J(\bar{y}, \bar{\theta}_y) = \begin{pmatrix} \eta||X||\bar{\theta}_y & -\eta||X||\bar{\theta}_y \\ 2\epsilon\bar{\theta}_y & -\epsilon \end{pmatrix} \tag{16}$$

The stability condition is $\frac{\eta}{\epsilon}||X|| < 1$.

C. Temporal Trace Based Rules

1. Földiák. A similar method of analysis as for threshold rules was applied. The matrix form of this system of equations after multiplication by X^T is $\dot{y} = \eta\theta(||X|| - y)$ and $\dot{\theta}_y = \epsilon(y - \theta)$. The equilibrium points are $\bar{y} = ||X||$, and $\bar{\theta} = y$. The Jacobian is

$$J(\bar{y}, \bar{\theta}) = \begin{pmatrix} -\eta\bar{\theta} & 0 \\ \epsilon & -\epsilon \end{pmatrix} \tag{17}$$

This system of equations is always stable and for any $||X||$, $y \rightarrow ||X||$ (assuming positive rates η and ϵ).

2. Temporal Difference. This rule is given in terms of discrete time steps. For the purpose of stability analysis an extrapolation to the continuous differential equation as $t \rightarrow \infty$ gives $\dot{y} = \eta||X||(r + (\gamma - 1)y)$. The solution of this equation is

$$y(t) = r - (r - y_0)e^{\frac{\eta}{\gamma - 1}||X||t} \tag{18}$$

From this analytical solution it can be seen that the stability of this rule is given by keeping the factor in the exponent non-positive. This is fulfilled for $\eta||X|| \geq 0$ and $\gamma \leq 1$, which are the normal conditions of network operation. Stability of Rescorla-Wagner rules follows from the same equation with $\gamma = 0$ and the same condition on X. Note that the actual values of the weights are not included in the solution, but from equations (1) and (18) $r(t) \leftarrow X^T(t)W(t)$. As a result $w_i(t) \rightarrow \frac{r(t)}{x_i(t)}$. Similar inference can be made for all other rules where the convergence of y is defined in this paper. While Temporal Difference rule is successful in modeling predictive reward, the modification that allows learning the time interval between the conditional stimuli and rewards requires excessive memory storage to represent all possible time intervals. In this respect, the combination of this rule with a system that can efficiently represent time intervals is recommended to reduce the memory requirement.

TABLE IV

STABILITY OF BASIC HEBB RULE DERIVATIVES.
$C^* = W_0^T XX^T XX^T W_0, C = XX^T$, m IS A SIZE OF VECTOR X.

Name	Equation	Behavior	Conditions
Classic Hebb	$\dot{w}_i = \eta x_i y$	$W \to \infty$ $\|\|W\|\|^2 \to \infty$	any X $\eta > 0$
Hebb + passive decay	$\dot{w}_i = \eta x_i y-$ $-\alpha w_i$	$W \to \infty$ $\|\|W\|\|^2 \to \infty$ $W \to 0$ $\|\|W\|\|^2 \to 0$ $W \to \frac{\alpha}{\eta} C W_0$ $\|\|W\|\|^2 \to const$	$\sum x_i^2 > \frac{\alpha}{\eta}$ $\sum x_i^2 < \frac{\alpha}{\eta}$ $\sum x_i^2 = \frac{\alpha}{\eta}$
Pre-synaptically gated decay (outstar) [7]	$\dot{w}_i = \eta x_i y-$ $-\alpha x_i w_i$	$W \to \infty$ $\|\|W\|\|^2 \to \infty$ $W \to 0$ $\|\|W\|\|^2 \to 0$ $W \to \bar{W}_0$	$\sum x_i > \frac{\alpha}{\eta}$ $\sum x_i < \frac{\alpha}{\eta}$ $\sum x_i = \frac{\alpha}{\eta}$
Post-synaptically gated decay (instar) [7]	$\dot{w}_i = \eta x_i y-$ $-\alpha y w_i$	$W \to \frac{\eta}{\alpha} X$ $\|\|W\|\|^2 \to$ $(\frac{\eta}{\alpha})^2 C^*$	any X
Oja [8]	$\dot{w}_i = \eta x_i y-$ $-\alpha y^2 w_i$	$W \to \sqrt{\frac{\eta}{\alpha}} V$ $\|\|W\|\|^2 \to \frac{\eta}{\alpha}$	any X
Dual OR [9]	$\dot{w}_i = \eta x_i y-$ $-\alpha(x_i + y) w_i$	$W \to \frac{\eta}{\alpha} \frac{Xy}{X+y}$	$\sum x_i > \frac{\alpha}{\eta}$
Dual AND	$\dot{w}_i = \eta x_i y-$ $-\alpha x_i y w_i$	$W \to \frac{\eta}{\alpha} [\vec{e}]$ $\|\|W\|\|^2 \to$ $(\frac{\eta}{\alpha})^2 m$	any X

TABLE V

STABILITY OF THRESHOLD-BASED RULES VARIATIONS.

Name	Equations	Behavior	Condition
Covariance 1	$\dot{w}_i = \eta x_i (y - \theta_y)$ $\dot{\theta}_y = \epsilon(y - \theta_y)$		$\frac{\eta}{\epsilon} \|\|X\|\| < 1$
Covariance 2	$\dot{w}_i = \eta(x_i - \theta_{x_i}) y$ $\dot{\theta}_{x_i} = \epsilon(x_i - \theta_{x_i})$		
Original BCM	$\dot{w}_i = \eta(y - \theta_y) x_i y-$ $-\alpha w_i$		$\alpha > 0$
iBCM [12]	$\dot{w}_i = \eta(y - \theta_y) x_i y \sigma'(y)$ $\dot{\theta}_y = \epsilon(y^2 - \theta_y)$	$y \to 1$	Always
lBCM [13]	$\dot{w}_i = \eta(y - \theta_y) x_i \frac{y}{\theta_y}$ $\dot{\theta}_y = \epsilon(y^2 - \theta_y)$	$y \to 1$	$\frac{\eta}{\epsilon} \|\|X\|\| < 1$
Textbook BCM [10]	$\dot{w}_i = \eta(y - \theta_y) x_i y$ $\dot{\theta}_y = \epsilon(y^2 - \theta_y)$	$y \to 1$	$\frac{\eta}{\epsilon} \|\|X\|\| < 1$

TABLE VI

STABILITY OF TEMPORAL TRACE-BASED RULES VARIATIONS.

Name	Equation 1	Behavior	Conditions
TD [14]	$\Delta w_i = \eta x_i (t-1) \times$ $\times (r(t) - y(t-1) +$ $+ \gamma y(t))$	$y \to r$ $w_i(t) \to \frac{r(t)}{x_i(t)}$	any X
Földiák [16], [17]	$\dot{w}_i = \eta \theta_y (x_i - w_i)$ $\dot{\theta}_y = \epsilon(y - \theta_y)$	$y \to \|\|X\|\|$	any X

TABLE VII

RECOMMENDATIONS FOR USAGE.

Name	Recommendation
Classic Hebb	Not recommended
Hebb + passive decay	Not recommended
Pre-synaptically gated decay (outstar) [7]	Recommended only when postsynaptic activity is set **not** through the synapses that use this rule for learning
Post-synaptically gated decay (instar) [7]	Recommended to learn patterns of presynaptic signals in any systems that require input recognition
Oja [8]	Recommended for code compression when PCA on the input is benefitial
Dual OR [9]	Recommended to detect correlations between high values of pre- and postsynaptic activity
Dual AND	Only recommended for coincidence detection where signal amplitudes do not matter
Covariance Rules	No suggestions
BCM family: [10]–[13]	Recommended for feature detection where the output is in binary present/absent terms
TD [14]	Reinforcement learning: without learning of reward timings, or in combination with an efficient time representation
Földiák [16], [17]	No suggestions aside from those proposed in the original presentation by the author

IV. CONCLUSIONS

This paper summarizes a mathematical characterization of the behavior of several learning rules that will be targeted in future evolution of large-scale memristive neural systems based on *Cog Ex Machina* framework [4] through the applications of a general form learning rule [1]. In particular, the analysis focuses on three classes of rules conforming with the general form: basic Hebb rule derivatives, threshold-based rules, and explicit temporal trace-based rules. Tables IV, V and VI summarize the analyzed behavior of the synaptic weights W for each rule, and the conditions that should be imposed on the learning rule parameters and presynaptic activity x, to obtain the listed behavior.

Table VII summarizes our current recommendations clarifying what are the reasonable applications of the analyzed rules based on the results of stability analysis, equilibrium weight values, and original suggestions by the respective rule designers. These suggestions are by no means complete but we hope that they can serve as a starting point for system designers. Future work will further characterize the usability of these rules by taking into account the properties of the resulting weight matrices in whole-brain simulations like MoNETA project [5], abilities of each rule in developing stable and meaningful receptive fields, and performance criteria such as computational cost and memory efficiency.

REFERENCES

[1] A. Gorchetchnikov, M. Versace, H. M. Ames, B. Chandler, J. Léveillé, G. Livitz, E. Mingolla, G. Snider, R. Amerson, D. Carter, H. Abdalla, and M. S. Qureshi, "Review and unification of learning framework in *cog ex machina* platform for memristive neuromorphic hardware," in *Proceedings of the International Joint Conference on Neural Networks*, 2011, this volume.

[2] L. Chua and S. M. Kang, "Memristive devices and systems," *Proceedings of IEEE*, vol. 64, no. 2, pp. 388–398, 1976.

[3] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

[4] G. Snider, R. Amerson, D. Carter, H. Abdalla, M. S. Qureshi, J. Léveillé, M. Versace, H. M. Ames, S. Patrick, B. Chandler, A. Gorchetchnikov, and E. Mingolla, "From synapses to circuitry: Using memristive memory to explore the electronic brain," *IEEE Computer*, vol. 11, pp. 37–44, 2011.

[5] M. Versace and B. Chandler, “MoNETA: A mind made from memristors,” *IEEE Spectrum*, vol. 47, no. 12, pp. 30–37, 2010.

[6] D. O. Hebb, *The Organization of Behavior*. New York, NY: Wiley, 1949.

[7] S. Grossberg, “Adaptive pattern classification and universal recoding i: Parallel development and coding of neural feature detectors,” *Biol Cybern*, vol. 23, pp. 121–134, 1976.

[8] E. Oja, “A simplified neuron model as a principal component analyzer,” *J Math Biol*, vol. 15, pp. 267–273, 1982.

[9] S. Grossberg, S. Hwang, and E. Mingolla, “Thalamocortical dynamics of the McCollough effect: Boundary-surface alignment through perceptual learning,” *Vision Research*, vol. 42, pp. 1259–1286, 2002.

[10] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: MIT Press, 2001.

[11] E. L. Bienenstock, L. Cooper, and P. Munro, “Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex,” *J Neurosci*, vol. 2, no. 1, pp. 31–48, 1982.

[12] N. Intrator and L. N. Cooper, “Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability conditions,” *Neural Netw*, vol. 5, pp. 3–17, 1992.

[13] C. Law and L. Cooper, “Formation of receptive fields according to the BCM theory in realistic visual environments,” in *Proc Natl Acad Sci USA*, vol. 91, 1994, pp. 7797–7801.

[14] R. Sutton and A. Barto, *Reinforcement Learning*. Cambridge, MA: MIT Press, 1998.

[15] R. A. Rescorla and A. R. Wagner, “A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement,” in *Classical Conditioning II*, A. H. Black and W. F. Prokasy, Eds. New York, NY: Appleton-Century-Crofts, 1972, pp. 64–99.

[16] P. Földiák, “Forming sparse representations by local anti-Hebbian learning,” *Biol Cybern*, vol. 64, pp. 165–170, 1990.

[17] ——, “Learning invariance from transformation sequences,” *Neural Comput*, vol. 3, pp. 194–200, 1991.

[18] J. L. Wyatt and I. M. Elfadel, “Time-domain solutions of Ojas equations,” *Neural Comput*, vol. 7, no. 5, pp. 915–922, 1995.