Unsupervised Learning of Perceptual Feature Combinations

Minija Tamosiunaite[1,2*], **Christian Tetzlaff**[3,4], **Florentin Wörgötter**[1]

***For correspondence:** minija.tamosiunaite@vdu.lt (MT)

[1]Department for Computational Neuroscience, Third Physics Institute, University of Göttingen, Germany; [2]Vytautas Magnus University, Faculty of Informatics, Kaunas, Lithuania; [3]Group of Computational Synaptic Physiology, Department for Neuro- and Sensory Physiology, University Medical Center Göttingen, Germany; [4]Campus Institute Data Science, Göttingen, Germany

Abstract In many situations, it is behaviorally relevant for an animal to respond to co-occurrences of perceptual, possibly polymodal features, while these features alone may have no behavioral relevance. Thus, it is crucial for animals to learn such feature combinations in spite of the fact that they may occur with variable intensity and frequency. Here, we present a novel unsupervised learning mechanism that is largely independent of these contingencies and allows neurons in a network to achieve specificity for different feature combinations. Furthermore, we show that other existing advanced learning rules cannot satisfactorily form such multi-feature representations. Neurons with this property may, thus, serve as an initial stage for the processing of ecologically relevant real world situations for an animal.

Introduction

Coincident events or features can be highly relevant for animals and humans, and recognizing feature combinations may make all the difference between danger and safety. The red color of a mushroom paired with white surface dots as compared to a red one with a plain surface makes the difference between the poisonous Amanita muscaria (toadstool) and the eatable Amanita caesarea. While humans learn such feature combinations usually by supervision, animals often do so via trial and error. For example, rats and other animals perform scouting and probing of novel food sources until found to be safe. Repeated exposure to sensor-perceivable feature combinations in conjunction with no negative effects will them lead to the conclusion that it should be safe to eat this.

A central problem that arises here is that features will not only occur in combination but also on their own. This will happen with different individual- as well as coincidence-occurrence frequencies and, in addition, usually also with variable intensity. Thus, in order to learn the meaning of combined features, the nervous system must learn this without being distracted by these different types of variability. In terms of a neural network, this means that the neurons therein need to learn this in a convergent manner. Hence, neural plasticity must come to a standstill as soon as a feature combination has been detected where ongoing plasticity may lead to unwanted responses to single features alone.

Such an ecologically driven stopping of learning is a non-trivial problem for unsupervised learning, though. For example, the growth of a synaptic weight ω is for conventional Hebbian learning defined by $d\omega/dt = \mu \mathbf{u} g(y)$, with $\mathbf{u}$ the input vector of the neuron, $g(y)$ a function of its output y and $\mu << 1$ the learning rate. This definition, however, leads to infinite weight growth. Many stabilization methods and/or augmentations of the original Hebbian learning rule have been suggested to

prevent this, for example the covariance rule (***Sejnowski, 1977***), subtractive normalization methods (***Miller and MacKay, 1994***), the Bienenstock, Cooper Munroe rule (BCM, ***Bienenstock et al., 1982***), Oja's rule (***Oja, 1982***), and several more. More recently, learning rules had been introduced which combine Hebbian weight growth with a homeostatic, balancing term, called synaptic scaling (***Turrigiano et al., 1998***; ***London and Segev, 2001***; ***Turrigiano and Nelson, 2004***), for achieving convergence to a target activity (***Tetzlaff et al., 2011***). However, below we will show that these methods cannot solve the above problem.

As a consequence the issue of how to control weight development in a way that a neuron will reliably code for a feature combination remains unresolved. Here we suggest a rather simple solution to this. When growing weights in a network, combination-selective responses can be achieved by gradually dropping the learning rate to zero (simulated annealing) as soon as the neuron's activity is getting 'large enough', which happens earlier for combined than for individual stimuli. Simulated annealing has become a textbook method in reinforcement learning (RL), for example for step-size reduction (***Sutton and Barto, 1998***) or for reducing exploration rates (***Eberhart et al., 2001***) as well as in deep-RL (***Mnih et al., 2015***), but it is also widely used in supervised learning (***Nakamura et al., 2021***; ***Huang et al., 2017***; ***Liu et al., 2023***) as well as in different variants of Hebbian learning (***Xu et al., 1992***; ***Hyvärinen and Oja, 1998***; ***Nessler et al., 2008***; ***Krotov and Hopfield, 2019***), the latter being most closely related to the investigations in this study. However, in the context of Hebbian learning, simulated annealing is usually applied as an additional mechanism to ensure a stable convergence of weights, but never as the main factor for activity stabilization on its own.

Central to our approach is that the principle of using the neuron's output as the determining variable for the annealing leads to the advantageous property that neurons in a network will indeed develop specificity for different input or feature combinations. This differs from mere spike-coincidence detection because — as discussed above — the learning of input combination specificity needs to be independent (within reason) of the intensity of the input, represented by its occurrence frequency and its amplitude (or input firing rate). Amplitude invariance can to some degree be achieved using network-intrinsic normalization (e.g. ***Heeger, 1992***) by which differently strong activity, e.g. from external sensory features that converges onto a cell, will still lead to similar, albeit not identical, responses. These could then serve as the normalized inputs to the learning neuron. Below we will however show that conventional learning rules remain rather sensitive to even small amplitude variations that would remain after normalization such that input combination specificity cannot be learned in a robust manner. Another aspect that leads to problems is that learning needs repetitions. However, the brain has little or no influence on the occurrence frequency of any external stimulus or stimulus combination. Hence, to reliably learn input combination specificity the system must tolerate quite some variability in the occurrence frequencies of the different inputs as well as concerning their coincidences.

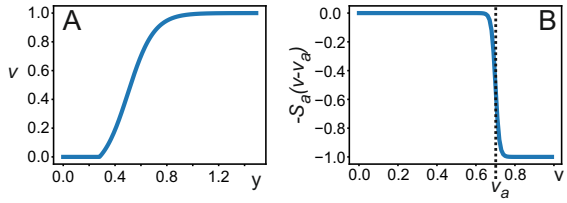

Figure 1. Neural activation- (A) & annealing function (B).

We start this investigation in the first part of the paper by analysing a simple case of a neuron with only two inputs and compare a set of the most common learning rules to show that — different from the other rules — the annealing-rule reliably allows detecting input coincidences in spite of amplitude and frequency variations of the input signals. This way the neuron acts as a (frequency and amplitude independent) AND operator. This will then be extended to a recurrent network and we demonstrate that all possible variants of multi-input ANDs are present in this network. We will finally discuss possible biological mechanisms that might support this function and also other issues concerning the learning of input coincidences.

Models

In this section we will first describe our neuron model, then the learning rule that we are proposing. Afterwards we briefly specify the traditional learning rules to which we are comparing the newly proposed method. Finally, we describe a setup, where we have embedded this rule in a recurrent neural network.

Neuron model

To obtain the neuronal response, first we calculate the weighted sum of the inputs:

$$y = \boldsymbol{\omega}^T \mathbf{u}, \tag{1}$$

where $\mathbf{u} = (u_1, ..., u_n)^T$ are inputs, $\boldsymbol{\omega} = (\omega_1, ..., \omega_n)^T$ are weights, and n is the number of inputs. In analogy to real neurons, we will call y the *membrane potential*. We will first analyze the simplest neuron that can detect co-incidences with $n = 2$, but will increase the number of inputs in the later-shown recurrent network example. To calculate the actual neuronal response v, called *spike rate or rate*, we apply a nonlinear function, where coefficients were set to obtain the response characteristic shown in Figure 1 A. This represents a sigmoidal function with a threshold $y_T \approx 0.281$, beneath which the firing rate will be zero:

$$v = f_s(y) = \begin{cases} \frac{1}{0.9}\left(\frac{1}{1+e^{-10(y-0.5)}} - 0.1\right) & \text{, if } \geq 0, \\ 0 & \text{, otherwise.} \end{cases} \tag{2}$$

Hebbian Learning Rules

We investigate different Hebbian learning rules given by:

$$\frac{d\boldsymbol{\omega}}{dt} = \mu(t)\boldsymbol{u}g(y), \tag{3}$$

with **u** the input vector of the neuron, $g(y)$ the the influence of the neuron's output on the learning and $\mu(t) << 1$ the learning rate, which will change over time due to annealing (see below). We define different characteristics for g as follows:

First (Eq. 4), there is a rule, which we call *membrane Hebb rule*:

$$g(y) = y. \tag{4}$$

Secondly, there is a rule that relies on the actual rate v and, hence, considers the output transform (Eq. 2), called *rate Hebb rule*:

$$g(y) = v. \tag{5}$$

Note that in this case, learning will only start if the membrane potential y is larger than ≈ 0.281, which the point where v begins to exceed zero (Fig. 1 A). This creates a bootstrapping problem if learning starts with small initial weights. To avoid this problem, in some cases we have replaced this rule with a rule that is largely output independent, called *Heavi Hebb rule*, given by:

$$g(y) = H(y - \eta) \tag{6}$$

with H being the Heaviside function:

$$H(y - \eta) = \begin{cases} 1 & \text{, if } y - \eta > 0, \\ 0 & \text{, otherwise.} \end{cases} \tag{7}$$

This learning rule augments traditional Hebbian learning by the assumption that weight change will *not* depend on the actual activation of the neuron. Instead learning will start as soon as the membrane potential y exceeds a threshold η and then only depends on the incoming input(s). Analysis of the experimental literature shows (see Discussion) that this type of learning may indeed play a strong role. Note also that, when using small η, this method avoids the bootstrapping problem

of infinitely slow learning when starting a system with near-zero weights. Furthermore, the rule in Eq. 6 removes the problem of the exponential run-away that occurs in conventional Hebbian learning due to the fact that the neuronal output coupled with the learning-equation creates a positive feedback loop. Below, we will also show that this rule works best for the here-investigated task of coincidence detection. For simplicity, we used $\eta = 0$ but results will not change much as long as one uses reasonably small values for η.

Here we investigate only the first and the third rule (membrane and Heavi), because the second case, which uses essentially a sigmoid function as output transform, can be tuned to either approximate the one or the other.

Learning Rate Annealing

Central to our method, however is that the spike rate v guides the annealing of the learning rate $\mu(t)$, where we start annealing, as soon as the neuron has reached "high enough" outputs v. The annealing equation is as follows:

$$\frac{d\mu}{dt} = -\nu S_a(v - v_a)\mu, \tag{8}$$

where v_a is the annealing threshold and $S_a(x)$ is another sigmoidal function:

$$S_a(x) = \frac{1}{1 + e^{-\beta x}}, \tag{9}$$

where we used $\beta = 100$ to obtain a steep step-like transition (see Figure 1 B). However, the method will work in a similar way with several times bigger or smaller β. Learning starts at $t = 0$ with $\mu(0) = \mu_0$. The sigmoidal function leads to the following effect: at the time when v exceeds v_a, the annealing rate abruptly increases. The value for v_a is expected to be around or higher than the inflection point of v. We have investigated $v_a \in [0.45, 0.95]$. Note that, if annealing happens too early, the neuron's differentiation capability remains low, as its activation function non-linearity will not play any role.

Reference Models

We compared our method to the classical BCM (***Bienenstock et al., 1982***) and Oja (***Oja, 1982***) rules as well as to a newer approach called synaptic scaling (***Tetzlaff et al., 2011***).

Note that in the following we use x for annotating the neuron's output, where we use either $x = y$ (linear case) or $x = v$ (non-linear case).

For the BCM rule we use the basic equations and parameterizations from (***Toyoizumi et al., 2014***), as given in equations 1 and 2 therein:

$$\tau_\omega \frac{d\boldsymbol{\omega}}{dt} = \boldsymbol{u}x(x - \theta), \tag{10}$$

where $\boldsymbol{u}$ are inputs, τ_ω defines the learning rate, and θ is defined as follows:

$$\tau_\theta \frac{d\theta}{dt} = -\theta + x\frac{x}{x_0}. \tag{11}$$

Accordingly, x_0 provides the target value for output x. Furthermore, τ_θ is significantly larger than τ_ω. Values used in experiments are given in the corresponding figure legends.

For the Oja rule, we use the standard formulation from (***Oja, 1982***):

$$\frac{d\boldsymbol{\omega}}{dt} = \mu x(\boldsymbol{u} - x\boldsymbol{\omega}). \tag{12}$$

For synaptic scaling, we use the following equation taken from (***Tetzlaff et al., 2011***):

$$\frac{d\boldsymbol{\omega}}{dt} = \mu x\boldsymbol{u} + \mu\nu(x_0 - x)\boldsymbol{\omega}^2, \tag{13}$$

where $\nu < 1$ and parameter x_0 determine the value at which the output is stabilizing (for concrete values see figure legends).

Experimental Settings

In the first part of the results section, we are investigating our system on a neuron with only two inputs with varying input amplitudes, frequencies and the standard deviation of these parameters as well as the coincidence rates between both inputs. This is extended by detailed statistics how our system behaves for different annealing-rates ν and thresholds υ_a. Finally, we compare the results from the system with annealing to results obtained from a set of the most common learning rules under similar conditions.

In the second part of the results sections, we employed the annealing learning rule for generation of *all possible* coincident combinations of N external inputs. For that we create a randomly connected network of M neurons with sparse connectivity of c connections per neuron on average. We use for connectivity a Gaussian distribution with standard deviation of $c/5$. However we are limiting this to a minimum of at least one connection onto each neuron. We also impose a limit on the maximally allowed connections, where for $c = 2$ this amounts to allowing connection numbers in the interval $[1, 3]$. We analyse the cases $M = 200$ and $M = 1000$, with $c = 2$ and $c = 10$ (allowed interval $[1, 19]$). In addition to those connections, 15% of randomly selected neurons are supplied with one connection each from randomly chosen external input neurons. We analyzed cases of $N = 3$ and $N = 5$ external inputs. Inputs can take two values: 0 or 0.5. For this part of the study we did not vary input amplitudes or frequencies. The goal of this part of the study is to show that such a system can self-organize into creating output neurons that respond to *any* possible combination of active inputs. Hence, one such neuron will then respond if a certain subset of k inputs is active at the same time (AND operation) and not respond if any one of these k inputs is not present. In this case the remaining $n - k$ inputs will not be able to drive this output neuron whatsoever. We were considering that a neuron is signaling for a certain input combination in case its activity is above a "classification" threshold for this combination, but below threshold for any other combination. We analyzed a set of thresholds from $\upsilon = 0.5$ to 0.9 in steps of 0.1.

We measured how many neurons, which are signaling a possible combination, appear within a network by calculating statistics from 100 trials to generate and train a network. For this, we varied the network connection matrix trial by trial. Also, neurons in the network were generated with an annealing threshold drawn from a uniform distribution in $[0.75, 0.95]$, which also was re-generated for each trial. Then, we present results as percentage of neurons in the network that represent a certain combination. Hence, 2% means that there were 4 neurons representing that combination in a neural network with $M = 200$ neurons and 20 neurons representing that combination in the $M = 1000$ network.

Example code for the different experiments will be made publicly available after journal-acceptance of this article.

Results

First we analyze the properties of the annealing learning rules as compared to the reference methods (BCM, Oja, Synaptic Scaling) for a neuron that has only two inputs. This analysis is started by an analytical calculation that compares membrane Hebb with Heavi Hebb after which we show simulation results for a wide variety of cases that cannot be captured by analytics. The central finding here is that only the annealing learning rules allow for reliable separation without having to re-tune neuron parameters individually in reaction to potentially quickly changing input situations, which is impossible for any real neuron.

This part is followed by a study of recurrently connected networks with more than two inputs, where we ask how reliably such a network could detect various types of coincidences.

Separation Properties

In the following we analyze how well does the Heavi Hebb rule, as compared to the membrane Hebb rule, separate the resulting output spike rates (coincidence case) relative to the individual

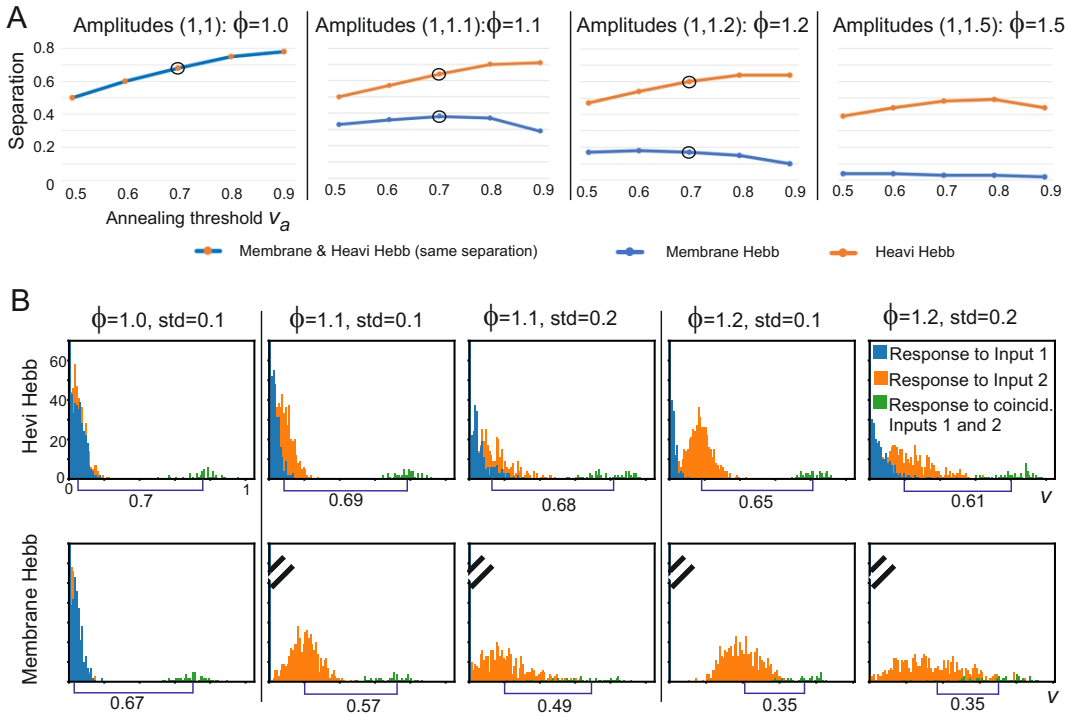

Figure 2. A) Separation properties calculated analytically B) Histograms of numerical results for Heavi Hebb and membrane Hebb. In both cases, input presentation frequencies are equal. In B) input coincidence is 10% everywhere; input amplitudes are drawn from a Gaussian distribution where the mean for the blue distributions was normalized to 1.0 and for the orange ones to ϕ; standard deviation are indicated above the plots. Annealing parameters are $v_a = 0.7$, $\nu = 0.2$. Initial weights are $\boldsymbol{\omega}(0) = [0.001, 0.001]^T$ and initial learning rate $\mu_0 = 0.0005$; Euler integration with step $dt = 1$. Disks mark those plot with a direct correspondence to the respective data points in A. Tilted lines are truncation marks for blue histograms.

rates obtained from only one input. We can here obtain analytical arguments under the assumption of independent constant inputs in the limit of few coincidences only (where the latter constraint is needed for the membrane Hebb rule only). Then, we also complement these analytical considerations by some simulations that allow relaxing the above constraints.

Hence, we assume two constant inputs, u_1 and ϕu_1 with $\phi > 1$. For the case of output-independent learning (Heavi Hebb rule) one can calculate weight growths over time as

$$\omega_1(t) = \mu_0 u_1 t + \omega_0 \tag{14}$$

$$\omega_2(t) = \mu_0 \phi u_1 t + \omega_0 \tag{15}$$

where μ_0 is the learning rate before annealing and ω_0 the start weight. Accordingly, the membrane potentials are:

$$y_1(t) = \mu_0 u_1^2 t + \omega_0 u_1 \tag{16}$$

$$y_2(t) = \mu_0 \phi^2 u_1^2 t + \omega_0 \phi u_1. \tag{17}$$

For the membrane Hebb rule we get for the weights:

$$\omega_1(t) = \omega_0 e^{\mu_0 u_1^2 t} \tag{18}$$

$$\omega_2(t) = \omega_0 e^{\mu_0 \phi^2 u_1^2 t}, \tag{19}$$

and the membrane potentials are given by:

$$y_1(t) = \omega_0 u_1 e^{\mu_0 u_1^2 t} \tag{20}$$

$$y_2(t) = \omega_0 \phi u_1 e^{\mu_0 \phi^2 u_1^2 t}. \tag{21}$$

If we allow for (rare) coincidences between the two inputs then the membrane potential becomes $y_1 + y_2$ and the neuron's output will be $v_{1+2} = f_s(y_1 + y_2)$ (see Eq. 2). Due to the definitions of y_1 and y_2 the following conjecture holds: $v_{1+2} > v_2 > v_1$. As a consequence v_{1+2} will eventually hit the annealing threshold v_a at time t_a. If we now assume instantaneous annealing, then all weight growth will stop and we can ask which values will the individual outputs v_1 and v_2 have reached? This way we can assess the separation between the coincidence-driven output (which is then at v_a) and the other two outputs. To be able to call such a neuron an AND-operator a clear separation is needed and here we are only concerned with v_2, which is anyhow larger than v_1. Hence, we calculate for different parameters u_1, μ_0, ϕ and v_a how big the separation $s(t_a)$ between $v_{1+2}(t_a) = v_a$ and $v_2(t_a)$ is as $s(t_a) = v_a - v_2(t_a)$. This last step has to be calculated numerically as the resulting terms cannot any longer by analytically solved. Figure 2 A shows the results. Note, that μ_0 has no influence on the separation, it only determines how early/late the annealing threshold will be reached. The figure shows that only for identical amplitudes the separation between the coincidence case and the individual input case will be the same for membrane- and Heavi Hebb. For all other situations, the Heavi Hebb rule leads to a far better separation. Furthermore, note that separation is largely independent of the annealing threshold, which adds to the robustness of the annealing approach.

In panel B we show how the Heavi- versus membrane Hebb rules behave when using inputs with a Gaussian distribution in amplitudes and the same presentation frequencies for both inputs. Coincidence rate was 10%. Responses to the individual inputs are shown in orange and blue whereas the coincident case in green. The results are consistent with the analytical analysis in panels A except for a slight increase in separation values due to more balanced weight growth in the simulation, because of the 10% coincidences, where the analytics could only be calculated for the limit case of 0%. The Heavi Hebb rule leads to a much stronger separation. Numbers at the bottom show the distance between the mean values of the orange and green distributions, where separability entirely ceases towards the right for membrane Hebb. Furthermore, note that membrane Hebb shows the expected exponential run-away property for the stronger (orange) distributions and the blue ones do not develop any firing rate v above zero for unequal amplitudes. Using a rate Hebb rule, would mitigate these effects as soon as the membrane potential to rate transformation approaches the Heaviside property.

Heavi Hebb rule: Neuron output analysis

In the following we focus on the Heavi Hebb rule, which provides a better separation as shown above. In Figure 3 we present histograms of neuron outputs for different input combinations the neuron is receiving. Input amplitudes are drawn from Gaussian distribution and are characterized by mean and standard deviation (see first column in Figure 3 A). We use mean amplitudes of 1, 1.2 and 1.5. Standard deviation std=0.1 is used for Figure 3; however we also show cases with std=0.2 (i.e., higher input variance) in Figure 2 above.

In addition to the amplitude distribution, inputs are characterized by presentation frequency. Note that this frequency is *not* a firing frequency; instead, this represents the "presentation frequency" and can be understood as how often stimuli are delivered to the neuron by the external world. In part A of Figure 3, we show results in case both inputs are presented with the same frequency, while in part B results are shown in which the first stimulus is twice more frequent.

Another important input parameter is how frequently two inputs coincide at the neuron. We consider 50, 30 and 10% coincidence. When presentation frequency of the two inputs differ, we calculate the percentage of coincidence from the input with smaller presentation frequency.

In Figure 3 we show the input distributions of the neuron (left column) and the neuron output in case of coincidence in green, while the response to single inputs are blue and orange. All neuron outputs v are limited to the interval $[0, 1]$, due to the non-linear response curve (see Figure 1).

As expected, the output for coincidence (green) is always the highest. We can also observe that the gap between the blue or orange histograms and the green histogram is in almost all cases quite big. Furthermore, this gap "sits at the same location" such that a unique discrimination threshold v_d

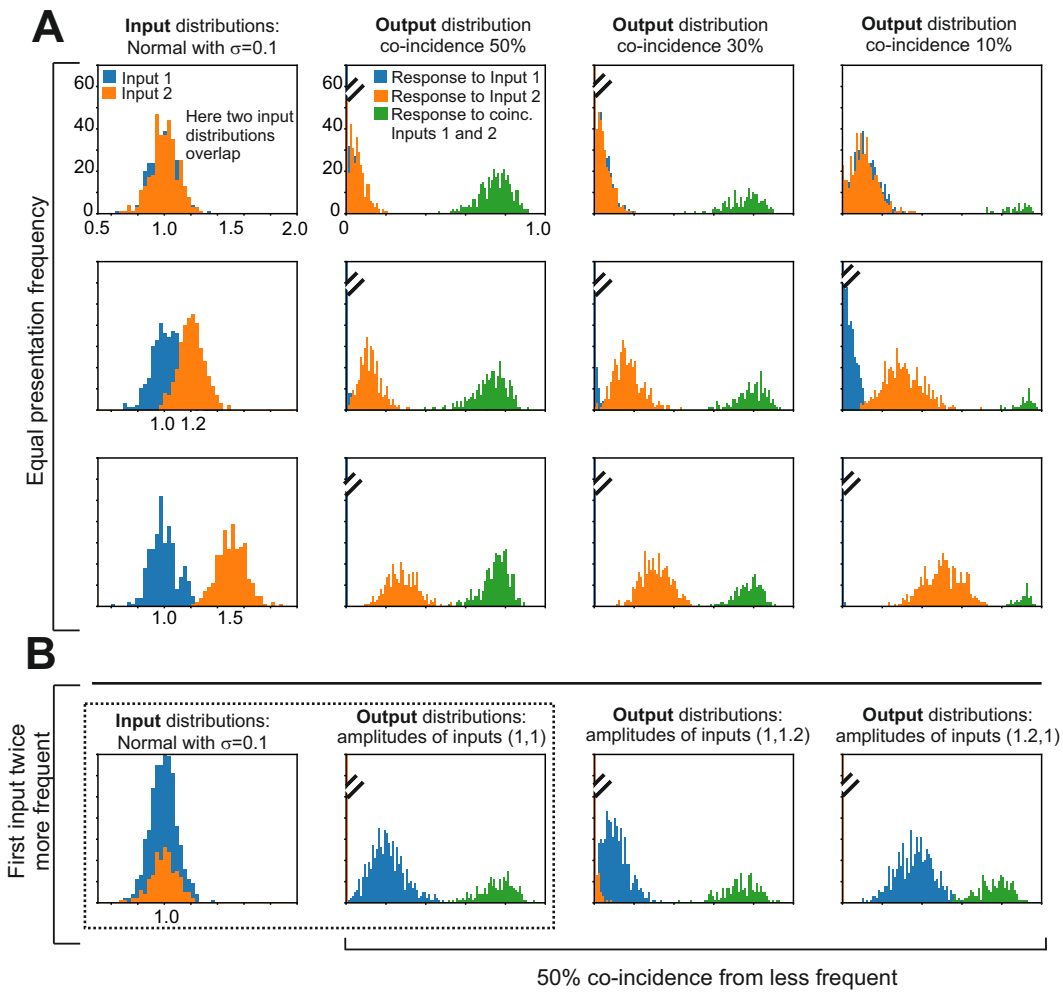

Figure 3. Histograms of responses at the output of the neuron. A: Equal presentation frequency; B: Different presentation frequency. Parameters: $\upsilon_a = 0.7$, $\nu = 0.1$, std=0.1. Mean amplitudes of the inputs are indicated in the first column. Initial weights are $\boldsymbol{\omega}(0) = [0.001, 0.001]^T$ and initial learning rate is $\mu_0 = 0.0005$; Euler integration with step $dt = 1$. For other parameters: see plot inscriptions. Response histograms in case of amplitude difference (blue or yellow) are grouping very close to zero, where we truncate the zero bin to optimize for visibility (see truncation marks).

could be defined to differentiate coincident from non-coincident responses (e.g. $\upsilon_d = 0.6$). These properties are, thus, largely independent of input amplitudes, frequencies, and percentages of coincidence. Thus, only due to these invariances such a neuron can indeed be called "input coincidence detector" (AND operation-like). Below we will show that other learning rules do not achieve this, but first we quantify the robustness of these properties.

In Figure 4 we show how the separation of coincidence vs no coincidence varies with different annealing parameters, where we vary the annealing onset threshold and the annealing speed υ_a and ν (Eq. (8)). We show the classification error as well as a contrast measure between coincidence vs no coincidence. Classification threshold is kept at $\upsilon = 0.5$. First, in the top left corner in Figure 4 A we present an error plot in parameter space in case both inputs have the same presentation frequency and both amplitudes are equal: mean=1, std=0.1, with 50% coincidence. This is the most favorable case from all shown and one can see that the error is zero (or very small) in a very big region of the parameter space (white and light colored patch in the middle of the plot). This patch slightly decreases when amplitudes differ (to the right), or coincidence percentage decreases (down). The errors “above-left” the white patch are false positives, while for “bottom-right”

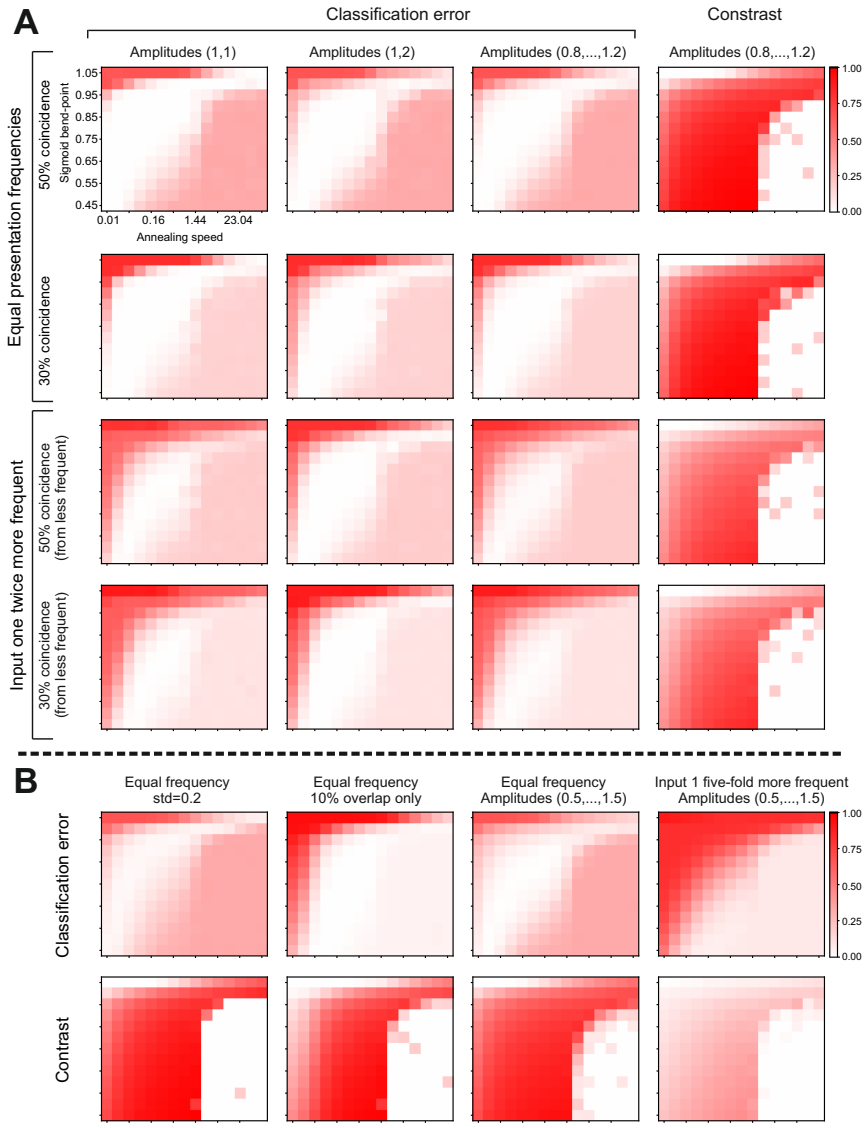

Figure 4. Method performance in respect to parameter variations. Parameters are annealing onset threshold and annealing rate. A: for variable amplitude, coincidence and presentation frequency, B: extreme cases: bigger variance, smaller coincidence, bigger amplitude difference, bigger frequency difference. When not indicated otherwise, amplitudes are equal, coincidence is 30%. Contrast for a single trial is given as $(a_{pos} - a_{neg})/(a_{pos} + a_{neg})$, where a_{pos} is the average amplitude for response to coincidence and a_{neg} is the average amplitude of responses to singular inputs. Average over 20 trials is indicated. Initial weights are $\boldsymbol{\omega}(0) = [0.001, 0.001]^T$ and initial learning rate $\mu_0 = 0.0005$; Euler integration with step $dt = 1$.

they are false negatives.

In the third column the same type of representation is shown, but for an amplitude mix, where the first input average amplitude is always at one, while the second input average amplitude is drawn from a set {0.8, 0.9, 1.0, 1.1 and 1.2} (uniform probability), std=0.1 everywhere. Also in this case the error is zero in a big patch of the parameter space.

In the fourth column the average contrast is shown. The areas where contrast is big, but error is non-zero (down-middle of the plot) represent cases where the outputs are well differentiated, but the coincidence cases have not fully traversed the value of 0.5, where our classification threshold for calculating the error is set. The error-free patch decreases when inputs have different presentation frequencies (rows 3-4), however the “good” parameter areas still are not singular.

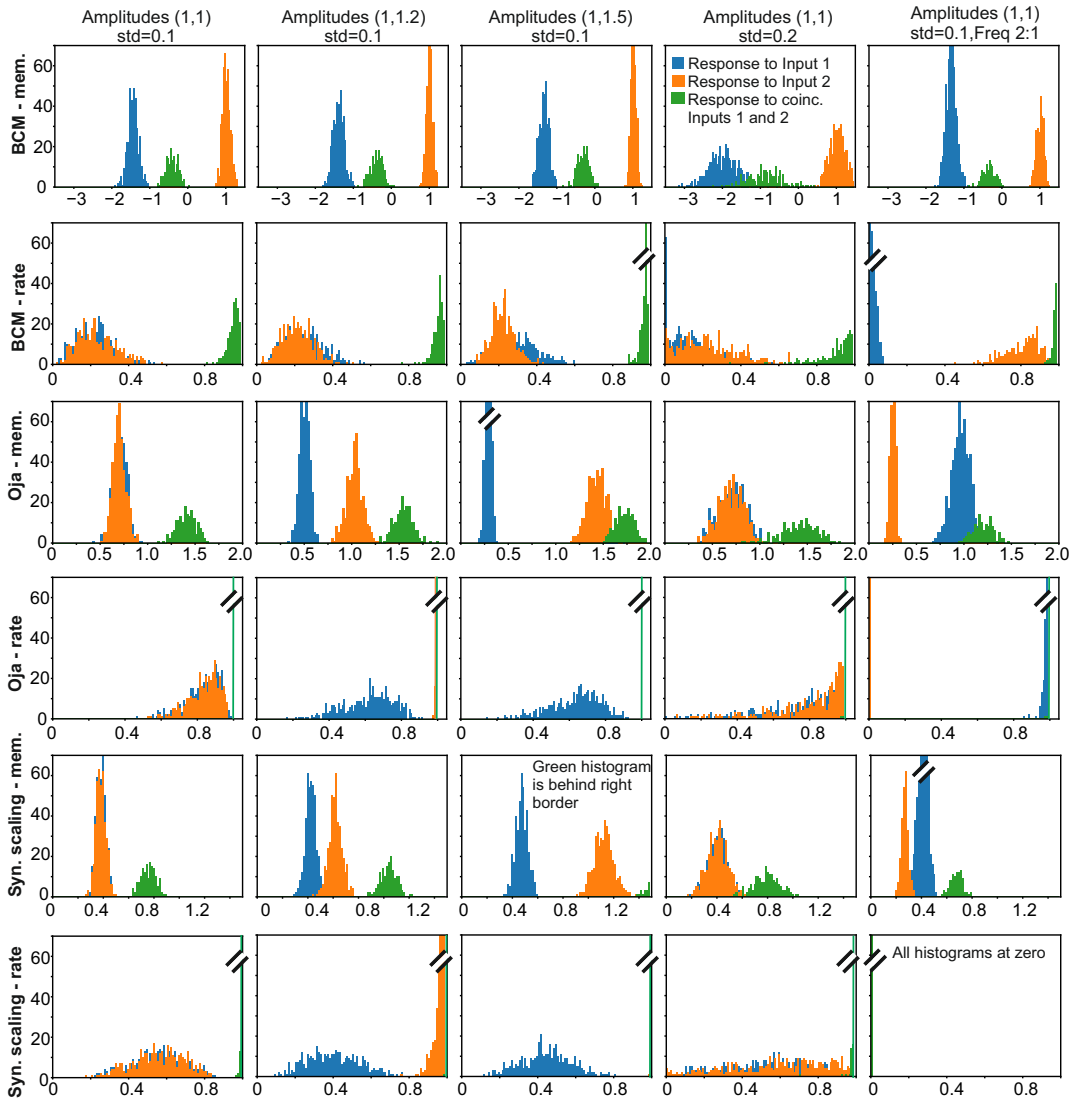

Figure 5. Results for the reference methods: BCM, Oja and synaptic scaling using linear (y, marked *mem*) and non-linear (v, *rate*) output terms in the learning. Input coincidence is 30% everywhere. Amplitudes and STD shown above each column. For all linear versions $\boldsymbol{\omega}(0) = [0.001, 0.001]^T$, for non-linear: $\boldsymbol{\omega}(0) = [0.2, 0.2]^T$. Other parameters: BCM linear: $v_0 = 1$, $\tau_\omega = 0.01$, $\tau_\theta/\tau_\omega = 10$, $\theta(0) = 0.2$; BCM nonlinear, different from above: $v_0 = 0.3$; Oja & synaptic scaling (linear & non-linear): $\mu = 0.001$; Synaptic Scaling: linear: $v_0 = -200$, $\nu = 0.01$, non-linear: $v_0 = -1.35$, $\nu = 0.7$.

In Figure 4 B we present various less favorable cases to investigate the limits of the rule: higher input variance (std=0.2), small coincidence (just 10%), wide amplitude range in the interval [0.5,1.5], as well the case when one input is five times less frequent but with 30% coincidence. Except for the last, five times less frequent case, we always get a parameter region where errors are zero. In the case where one input is five times less frequent (right-most plot) we can get a low classification error in some parameter range, but not zero. Note, that in this case the coincidence percentage is very small, too, as we calculate the 30%-percentage from the less frequent input. Thus, this case is, indeed, very unfavorable.

Comparison to reference methods

In Figure 5 we show results obtained with the three reference methods where we use either the membrane potential y or the rate output v to drive learning. Results of this investigation are char-

acteristic for the problems that these methods have with the task of input coincidence detection. Also in some cases, quite some parameter tuning was needed to get any reasonable result. Note also, that all reference rules heavily depend on the output, which is assumed to be monotonically increasing; thus they are not compatible with Heaviside regimes we used in our rule.

The most prominent property of the membrane-BCM method is that the coincidence output is not the highest: outputs to single inputs reside on both sides of the green coincidence histogram. Note, that the original membrane-BMC method results in negative weights. In the paper of Toyoizumi et al (***Toyoizumi et al., 2014***) an augmented BCM equation is provided to avoid this, but using this will not alter the above observation. When using the rate to drive learning with BCM, the situation improves, but differences in presentation frequency lead to non-separatable distributions (compare to Figure 3 B where the corresponding situation is shown for the Heavi Heb method). However, in general the rate-BCM performs second best in our task, which may be due to the fact the BCM does indeed also implement some sort of "simulated annealing", where an increase of the neurons activity will shift the BCM threshold accordingly (see Discussion section for details).

For membrane-Oja there is no unique separation threshold to be found for the difference cases and it depends on the stimulus situation. Furthermore, some distributions overlap and cannot be separated at all. For rate-Oja, we could not find any parameter setting that produces reasonable results and plots shown here were the best we could produce. Note, that Oja in existing literature is normally used in linear regimes (i.e., only as membrane-Oja).

The same is true for rate-scaling (last row), where membrane scaling works better and nicely separates all distributions, albeit always with different separation thresholds.

Hence, none of the reference methods can reliably assure that (1) there is a large gap between the single input distributions and the distributions of coincident inputs and (2) that this gap does not shift when using (within reason) different input situations allowing for a unique separation threshold. As mentioned above, only due these two properties a neuron can operate as an "input coincidence detector", which cannot reliably be achieved with theses methods.

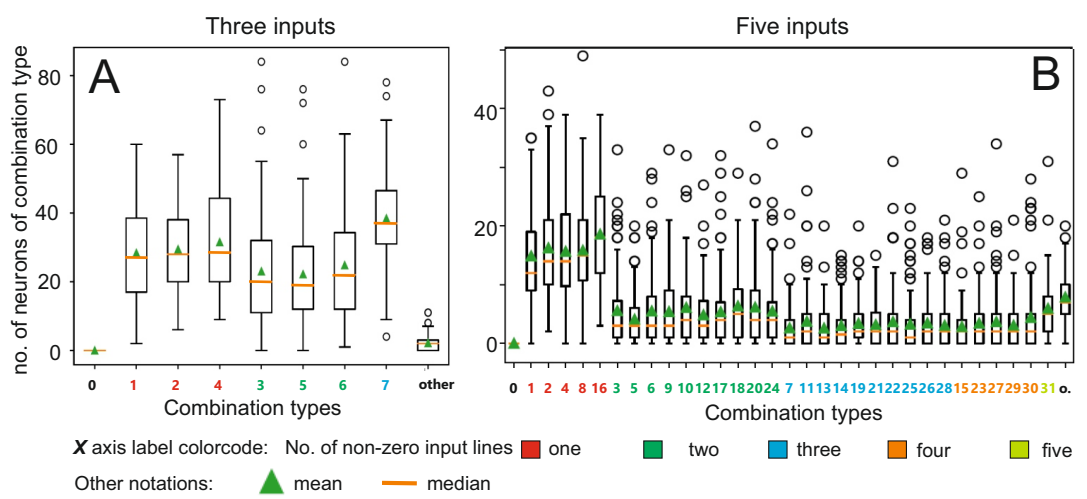

Figure 6. Box plots for different combinations of inputs for the cases of $N = 3$ and $N = 5$ inputs. Combinations are aligned in ascending order of active inputs, with color code indicating the number of inputs, see legend at the bottom. Combinations are indicated by decimal numbers corresponding to binary set notation (e.g. “3” means the set: 00011, where the two last inputs are active). “o” means other, where this denotes occurrences of cells signaling several different combinations. The size of the neural network is $M = 200$, average connectivity $c = 2$, annealing parameters are: annealing rate $\nu = 0.3$, where the annealing threshold υ_a for each neuron individually is drawn from a uniform distribution [0.75,0.95]. Initial weights are chosen from Gaussian distribution with mean 0.001 and std 0.0002. Initial learning rate $\mu = 0.0005$. Euler integration with $dt = 1$. Median, mean and standard deviation are shown on the basis of 100 trials.

Recurrent network: all input combinations

First, in Figure 6 we provide a box plot for the number of different combinations obtained for $N = 3$ or $N = 5$ external inputs. Statistics are shown for 100 randomly generated networks. For this we count, after learning, how many neurons respond, for example, to an input combination of "x11xx". Such a neuron, shown with green index "12" in the panel B, thus, requires inputs 2 and 3 (encoded as "1") to be active, where the other inputs may or may not be present (encoded as "x"), but they will not be able to drive this neuron on their own. One can see that for $N = 3$ the number of cells representing different combinations is essentially uniformly distributed, while for $N = 5$ the number of neurons representing single inputs is higher than the rest. As expected standard deviations are high but, in spite of this, for any of the possible combinations there are always at least a few cells that represent them.

It is here important to note that this network does not produce an excess of neurons that respond to the condition "other" (about 7 aut of 200 cells do this in the 5-input case, panel B). "Other" means that a neuron would code, for example, for "x1x1x" as well as for "11xxx" and possibly for even more different combinations. If self-organization were driven by a pure random process a very strong excess of such neurons would be expected, which is not the case here. In the next figures this is analyzed in more detail. Hence, our networks, indeed, self-organize into a set of input-combination selective neurons.

In Figure 7 we analyze how the proportion of different combinations change with varying decision threshold (A,C) and for the same decision threshold but in different network architectures

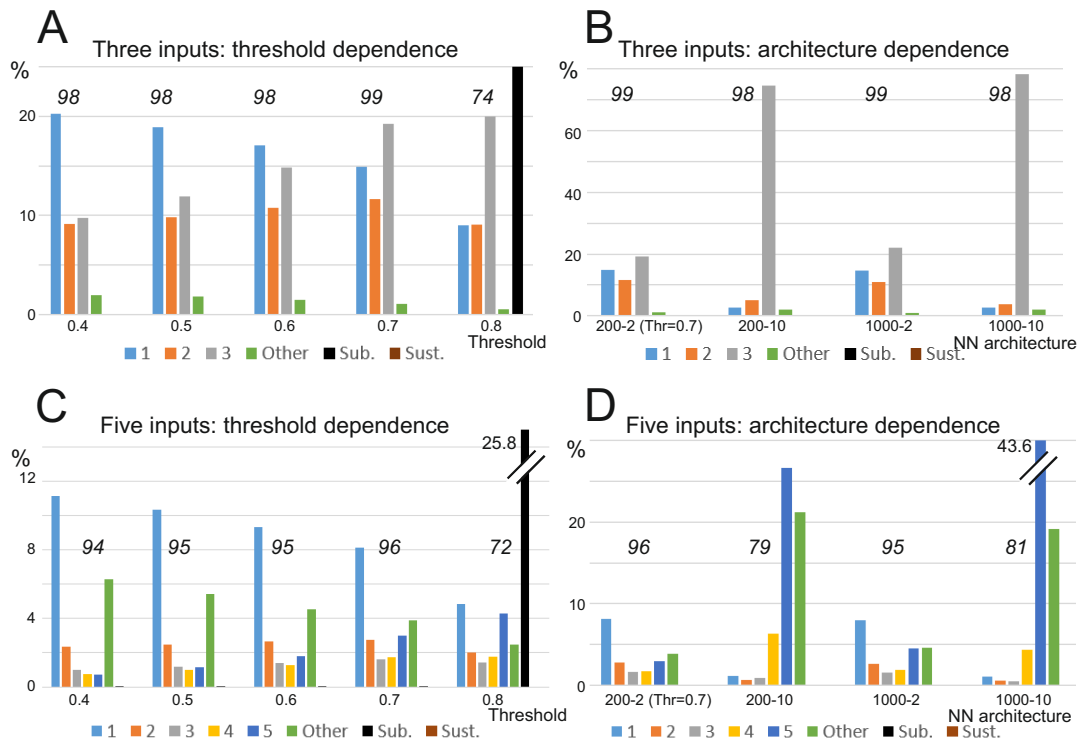

Figure 7. Different combination distribution based on decision threshold and neural network architecture. A and B: three input case; C and D: five input case. Numbers 1 to 5 indicate combinations responsive to corresponding number of inputs; “Other” represent cells signaling more than one combination (see text for explanation), “Sub.” denotes sub-threshold cases, while “Sust.” denotes sustained activity, which does not subside after switching off the inputs, which does not happen here (but in the baseline, see next Figure). Neural network (NN) architecture notation: “No of neurons”-“connectivity” (M-c). Numbers above column groups denote percentage of combination-selective neurons (vs. “Other” and “Sub.” neurons). Initial settings and learning parameters as in Fig. 6.

(B,D). We quantify how many neurons are – on average – selective for *any* input combination. To achieve this, we first sum the number of neurons that represent the same *type* of input combination: single input, two inputs and so on. Then we divide this sum by the number of type-identical combinations. For example, for the $N = 5$-case there are 5 single, 10 double, 10 triple, 5 quadruple and 1 quintuple combination existing. Hence, percentage plots in figure 7 do not sum up to 100. However, to also be able to show the strong difference between combination-selective versus non-selective ("other" + "sub-threshold" + "sustained") neurons, we provide their total percentage of the combination-selective neurons, too (numbers in italics). Standard deviations are of the same order of magnitude, as shown in the box-plot above and omitted here to make the diagrams better readable.

Decision threshold dependence is analyzed in parts A and C for $N = 3$ and $N = 5$, respectively. All combinations are represented with an increasing prevalence of more-complex combinations for higher threshold values. Single cases are over-represented, where this over-representation decreases with increasing decision threshold.

The green column in the histograms show “Other” cases, which counts all the cells in the network that are active with two or more combinations, where one is not the subset of the other. This number is not substantial when connectivity is low $c = 2$ and is relatively high if we use connectivity $c = 10$ (see part D). If the decision threshold becomes too high (see $\upsilon_a = 0.8$ in A and C), sub-threshold cases emerge (black column). These are cases where the neuron may "fire" but never reaches annealing threshold. None of the networks that we trained this way showed sustained activity, which is a type of activity that persists after the inputs have been switched off (but see next).

Results can be compared to baseline performance, where the weights obtained by our learning procedure are randomly reshuffled (permuted) in between connections in the network, while the general network connectivity pattern (which neuron connects to which other neuron) remains the same. The percentage of different cases is shown in Figure 8, column group “permuted”. Here we see that both for $N = 3$ (A) and $N = 5$ (B), we only have a few percents of neurons responding to single inputs only, but essentially no more-complex combinations emerge (compare to column “learned” drawn on the left). Instead, for baseline most of the cells remain sub-threshold. The question naturally arises whether this is just a scaling effect? Hence, to investigate if we can get more useful above-threshold combinations with bigger weights, we increased all weights in the baseline by 1.5 or be 2 (columns “permuted x 1.5” and “permuted x 2”). Here we get a few more single responses and, as discussed above, also more “Other” responses (numbers in green), but now also sustained activity emerges (brown column in the diagrams) and dominates for “permuted x 2”. Hence, the network activity does not come to rest after stimuli have been removed. Thus, this baseline shows that the learning method, suggested in this study, allows generating in an unsupervised manner neurons selective for *specific* combinations of inputs (low number of random="other"

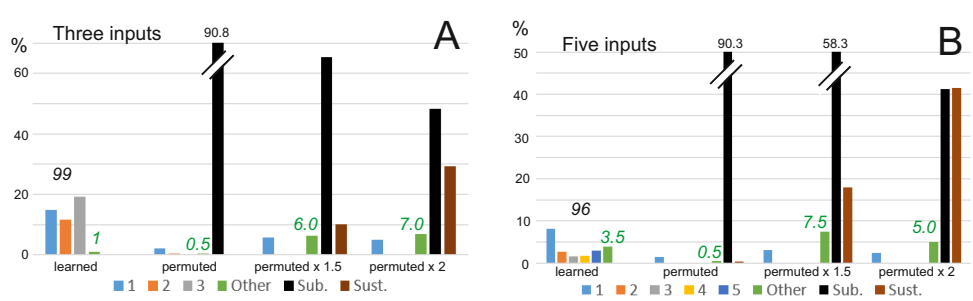

Figure 8. Comparison to baseline. A: Three input case, B: Five input case. The column group “learned” shows performance of our method, $M = 200$, $c = 2$, $\upsilon_a = 0.7$, copied from Figure 7; “permuted” is for the case with learned weights randomly permuted; “permuted x 1.5” and “permuted x 2” for cases with permuted weights multiplied by 1.5 and 2, respectively. Numbers in green refer to the class "other".

combinations) without leading to sustained activation.

Discussion

In this study we have shown how it is possible for neurons to arrive, by unsupervised learning, at a well-defined target activity and, when using multiple inputs, how different neurons in a network become specific for different feature combinations. Interestingly, in those networks all combinations are found. A comparison with a wide variety of other learning rules shows that these properties are not easily attainable in an amplitude and frequency independent manner. The neurons in our system develop this way the property of a (somewhat unconventional) "switched AND-OR operator" (Fig. 9), where certain input lines must be present (AND property) while others are irrelevant (switches remain open for those) such that only the AND-gated inputs drive the output. To achieve this we have introduced two modifications of the traditional Hebb rule, which we will discuss in the following:

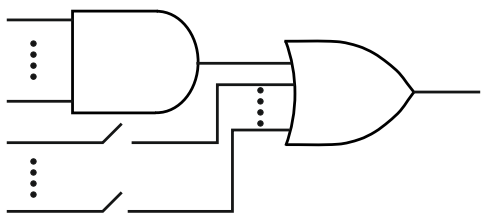

Figure 9. Switched AND-OR operator.

A) We reduced the influence of the neuronal output onto learning to an all-or-none behavior by using the Heaviside function, possibly with a threshold ($\eta > 0$). This way learning will start as soon as the neuronal activity is larger than this threshold but not depend on the actual neuronal activity.

B) We used annealing of the learning rate, as soon as the neuron has reached a certain output level. While annealing is a well-known supplementary technique in many, also unsupervised, approaches (***Xu et al., 1992***; ***Nessler et al., 2008***) we use it as the *main* mechanism to stabilize learning.

All-or-non learning

The use of the Heaviside function for Hebbian learning provides, from a theoretical perspective, several clear advantages because it leads only to linear weight growth (Eq. 6). Different from this, the membrane Hebb rule, which uses the membrane potential to drive learning, leads to exponential weight growth (Eq. 4) and a strong run-away effect of the weights that belong to the stronger inputs (see Fig. 2 B). Furthermore, (especially at dendritic spines) it appears that the post-synaptic depolarization effects, that influence Ca^{++} influx through NMDA channels, which determine LTP, have an all-or-none effect on plasticity. The absolute values for Ca^{++} within the dendrite required for the induction of synaptic plasticity have been estimated as $150–500\ nM$ for LTD and $> 500\ nM$ for LTP (***Cormier et al., 2001***). Furthermore, it has been measured that a single EPSP can raise the Ca^{++}-level to $700\ nM$, where a pairing of post-synaptic depolarization with synaptic stimulation would even drive it up to as much as $12\ \mu M$ (***Sabatini et al., 2002***).

Based on these findings Rackham et al. (***Rackham et al., 2010***) had designed a model of plasticity in spines that predicts that an EPSP resulting from the activation of a single synapse is sufficient to cause a significant Ca^{++} influx through NMDA receptors. This is in line with experimental data (***Bloodgood and Sabatini, 2007***; ***Canepari et al., 2007***; ***Sobczyk and Svoboda, 2007***). As a consequence, it appears that every post-synaptic back-propagating spike or dendritic spike will be enough to lead to substantial Ca^{++} influx to trigger plasticity (at a spine). This argues for a sharp transition of the post-synaptic learning influence, where the use of the Heaviside function would represent a limit case. Sigmoidal transition functions similar to Eq. 2 could be used instead, where results for this study will be little affected if the sigmoid is steep enough (data not shown).

Learning Rate Annealing

In 1998, Bi and Poo (***Bi and Poo, 1998***) had shown that the change in EPSP amplitude is inversely related to the size of the EPSP when employing a plasticity protocol. Hence, large synapses grow less than small synapses. This is potentially a ceiling (saturation) effect of LTP and could, in theoretical terms, indeed be captured by a learning rate annealing mechanism. This, however, points to a core problem: For theoreticians the learning rate is just a single variable and learning rate annealing is essentially just an abstraction of meta-plasticity. Linking this to complex multi-faceted biophysical processes, thus, remains difficult. There is a wealth of literature that suggests that the reduction of LTP, due to meta-plasticity, could rely on effects that influence NMDA receptors (***Huang et al., 1992***; ***Coan et al., 1989***; ***Youssef et al., 2006***; ***Satoshi et al., 1996***; ***Frey et al., 1995***). However, the time course of this might be too fast as these effects seem to decay within about one hour (***Huang et al., 1992***). Stimulus driven annealing ought to be able to act rather on longer time-scales because the animal may only now and then encounter the relevant stimuli. Longer lasting reduction of LTP could be obtained by mechanisms that operate on its later phases (late-LTP) (***Frey and Morris, 1997***; ***Redondo and Morris, 2011***; ***Luboeinski and Tetzlaff, 2021***; ***Lehr et al., 2022***) suspected to be essential for establishing synaptic consolidation. However, any potential role of this mechanisms in meta-plasticity related to annealing effects remains unknown. Nevertheless, it seems conceivable that neurons reduce their 'learning-efforts' by reducing the synthesis of some relevant biochemical components using a saturation-driven kinetics, as soon as the neuron's activity has grown enough, which could be understood as learning rate annealing.

In the theoretical literature, learning rate annealing is a very widely used mechanism applied with different learning rules and for different purposes (***Sandholm and Crites, 1996***; ***Smith, 2002***; ***Nessler et al., 2008***; ***Krotov and Hopfield, 2019***). Notably, the BCM rule also has a mechanism built in that could be understood as annealing. Its threshold θ relies on the time-averaged level of postsynaptic firing. Thus, if firing levels are maintained at a high level, this threshold shifts, making LTP harder to obtain. With quite some tuning, this rule was also able to solve the tasks investigated in this study at least to some degree. However, BCM relies on a squared output term, which – in spite of the threshold-induced damping – makes its behavior for multiple inputs in a network complex and often unstable. Note that, under the physiological condition of $1 > \theta > \mu$, it can be easily shown by equating the Nullclines and using a Taylor expansion, that the characteristic second fix-point of BMC (the first is at the origin) will only be a stable node if u and/or x_0 are big. In network learning this cannot be unequivocally assured, leading to instabilities. Given this complexity, it is unclear how BMC could be modeled in biophysical terms, also in view that saturation-driven kinetic mechanisms that operate of one or more compounds needed for LTP, do not map well to this rule. Synaptic scaling has been suggested as a possible mechanism to achieve this, too. Scaling operates on rather long time scales, slower than learning. The existing scaling-rules, however, do not lead to the same fix-points when different stimulus situations are present.

Comparing to other learning principles

Currently our method leads only to weight growth. It is however, straightforward to complement this with a decay term (forgetting) and/or with a mechanism for LTD. This can, for example, be done by using a sigmoidal function $G(y-\eta), \eta > 0$ with values between -1 and $+1$ instead of the Heaviside function, which will lead to weight reduction for $y - \eta < 0$. We are currently investigating this rule, but this goes beyond the scope of the current study.

The other rules investigated in this study can lead to LTD, but none of them could reliably solve our task. Clearly, reinforcement learning and supervised learning would be able to achieve this, though. Both, however, require evaluative feedback in the form of rewards or by use of an error function. Different from this, our method is non-evaluative and essentially performs a process of self-organized stimulus sorting. Any potential ecologically meaningful evaluation could then come on top and, for example, reinforcement learning of a beneficial behavioral policy could make use of the responses of our feature-combination specific neurons.

Conclusions

With the mechanisms employed here we demonstrated that neurons can learn to respond to specific input combinations in an unsupervised manner. While such responses are reminiscent of an AND operation, this can only be achieved if the system reacts in a rather invariant way – and different from a classical AND – to stimuli of different amplitude and occurrence frequency, which is assured by annealing. We believe that this specificity for input combinations may be of ecological relevance for an animal, because it allows learning to respond to sets of inputs that might indicate situations with different – positive or negative – valance, where individual features might be irrelevant.

Acknowledgments

Supported by the German Science Foundation (DFG), grant WO 388/17-1 (F.W.) and grant TE 1172/7-1 (C.T.), as well as by the European Commission H2020, grantno.: 899265 "ADOPD" (F.W.) and the German Federal Ministry of Education and Research (BMBF) grantno. 01IS22093A "KISSKI" (C.T.).

References

Bi GQ, Poo MM. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. Journal of Neuroscience. 1998; 18(24):10464–10472.

Bienenstock EL, Cooper LN, Munro PW. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. Journal of Neuroscience. 1982; 2(1):32–48.

Bloodgood BL, Sabatini BL. Nonlinear regulation of unitary synaptic signals by CaV2. 3 voltage-sensitive calcium channels located in dendritic spines. Neuron. 2007; 53(2):249–260. doi: 10.1016/j.neuron.2006.12.017.

Canepari M, Djurisic M, Zecevic D. Dendritic signals from rat hippocampal CA1 pyramidal neurons during coincident pre-and post-synaptic activity: a combined voltage-and calcium-imaging study. The Journal of Physiology. 2007; 580(2):463–484. doi: 10.1113/jphysiol.2006.125005.

Coan E, Irving A, Collingridge G. Low-frequency activation of the NMDA receptor system can prevent the induction of LTP. Neuroscience Letters. 1989; 105(1-2):205–210.

Cormier R, Greenwood A, Connor J. Bidirectional synaptic plasticity correlated with the magnitude of dendritic calcium transients above a threshold. Journal of Neurophysiology. 2001; 85(1):399–406. doi: 10.1152/jn.2001.85.1.399.

Eberhart RC, Shi Y, Kennedy J. Swarm intelligence. Elsevier; 2001.

Frey U, Schollmeier K, Reymann K, Seidenbecher T. Asymptotic hippocampal long-term potentiation in rats does not preclude additional potentiation at later phases. Neuroscience. 1995; 67(4):799–807.

Frey U, Morris RG. Synaptic tagging and long-term potentiation. Nature. 1997; 385(6616):533–536.

Heeger DJ. Normalization of cell responses in cat striate cortex. Visual Neuroscience. 1992; 9(2):181–197.

Huang G, Li Y, Pleiss G, Liu Z, Hopcroft JE, Weinberger KQ. Snapshot ensembles: Train 1, get m for free. arXiv preprint arXiv:170400109. 2017; doi: 10.48550/arXiv.1704.00109.

Huang YY, Colino A, Selig DK, Malenka RC. The influence of prior synaptic activity on the induction of long-term potentiation. Science. 1992; 255(5045):730–733.

Hyvärinen A, Oja E. Independent component analysis by general nonlinear Hebbian-like learning rules. Signal Processing. 1998; 64(3):301–313.

Krotov D, Hopfield JJ. Unsupervised learning by competing hidden units. Proceedings of the National Academy of Sciences. 2019; 116(16):7723–7731. doi: 10.1073/pnas.1820458116.

Lehr AB, Luboeinski J, Tetzlaff C. Neuromodulator-dependent synaptic tagging and capture retroactively controls neural coding in spiking neural networks. Scientific Reports. 2022; 12(1):17772. doi: 10.1038/s41598-022-22430-7.

Liu C, Huang W, Xu RYD. Implicit bias of deep learning in the large learning rate phase: A data separability perspective. Applied Sciences. 2023; 13(6):3961. doi: 10.3390/app13063961.

London M, Segev I. Synaptic scaling in vitro and in vivo. Nature Neuroscience. 2001; 4(9):853–854. doi: 10.1038/nn0901-853.

Luboeinski J, Tetzlaff C. Memory consolidation and improvement by synaptic tagging and capture in recurrent neural networks. Communications Biology. 2021; 4(1):275. doi: 10.1038/s42003-021-01778-y.

Miller KD, MacKay DJ. The role of constraints in Hebbian learning. Neural Computation. 1994; 6(1):100–126.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. Human-level control through deep reinforcement learning. Nature. 2015; 518(7540):529–533. doi: 10.1038/nature14236.

Nakamura K, Derbel B, Won KJ, Hong BW. Learning-Rate Annealing Methods for Deep Neural Networks. Electronics. 2021; 10(16):2029. doi: 10.3390/electronics10162029.

Nessler B, Pfeiffer M, Maass W. Hebbian learning of Bayes optimal decisions. Advances in Neural Information Processing Systems. 2008; 21. doi: 10.5555/2981780.2981926.

Oja E. Simplified neuron model as a principal component analyzer. Journal of mathematical biology. 1982; 15:267–273.

Rackham O, Tsaneva-Atanasova K, Ganesh A, Mellor J. A Ca2+-based computational model for NMDA receptor-dependent synaptic plasticity at individual post-synaptic spines in the hippocampus. Frontiers in Synaptic Neuroscience. 2010; p. 31. doi: 10.3389/fnsyn.2010.00031.

Redondo RL, Morris RG. Making memories last: the synaptic tagging and capture hypothesis. Nature Reviews Neuroscience. 2011; 12(1):17–30. doi: 10.1038/nrn2963.

Sabatini BL, Oertner TG, Svoboda K. The life cycle of Ca2+ ions in dendritic spines. Neuron. 2002; 33(3):439–452. doi: 10.1016/S0896-6273(02)00573-1.

Sandholm TW, Crites RH. Multiagent reinforcement learning in the iterated prisoner's dilemma. Biosystems. 1996; 37(1-2):147–166.

Satoshi F, Yoichiro K, Masami M, Hidekazu F, Hiroshi S, Kenya K, Kenichi I, Zhixiong C, Hiroshi K. The long-term suppressive effect of prior activation of synaptic inputs by low-frequency stimulation on induction of long-term potentiation in CA1 neurons of guinea pig hippocampal slices. Experimental Brain Research. 1996; 111:305–312.

Sejnowski TJ. Storing covariance with nonlinearly interacting neurons. Journal of Mathematical Biology. 1977; 4(4):303–321.

Smith AJ. Applications of the self-organising map to reinforcement learning. Neural networks. 2002; 15(8-9):1107–1124. doi: 10.1016/S0893-6080(02)00083-7.

Sobczyk A, Svoboda K. Activity-dependent plasticity of the NMDA-receptor fractional Ca2+ current. Neuron. 2007; 53(1):17–24. doi: 10.1016/j.neuron.2006.11.016.

Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press; 1998.

Tetzlaff C, Kolodziejski C, Timme M, Wörgötter F. Synaptic scaling in combination with many generic plasticity mechanisms stabilizes circuit connectivity. Frontiers in Computational Neuroscience. 2011; 5:47. doi: 10.3389/fncom.2011.000.

Toyoizumi T, Kaneko M, Stryker MP, Miller KD. Modeling the dynamic interaction of Hebbian and homeostatic plasticity. Neuron. 2014; 84(2):497–510. doi: 10.1016/j.neuron.2014.09.036.

Turrigiano GG, Leslie KR, Desai NS, Rutherford LC, Nelson SB. Activity-dependent scaling of quantal amplitude in neocortical neurons. Nature. 1998; 391(6670):892–896.

Turrigiano GG, Nelson SB. Homeostatic plasticity in the developing nervous system. Nature Reviews Neuroscience. 2004; 5(2):97–107. doi: 10.1038/nrn1327.

Xu L, Oja E, Suen CY. Modified Hebbian learning for curve and surface fitting. Neural Networks. 1992; 5(3):441–457.

Youssef FF, Addae JI, Stone TW. NMDA-induced preconditioning attenuates synaptic plasticity in the rat hippocampus. Brain Research. 2006; 1073:183–189. doi: 10.1016/j.brainres.2005.12.008.