

A Gradient Learning Rule for the Tempotron

Robert Urbanczik

urbanczik@pyl.unibe.ch

Walter Senn

senn@pyl.unibe.ch

Department of Physiology, University of Bern, 3012-Bern, Switzerland

We introduce a new supervised learning rule for the tempotron task: the binary classification of input spike trains by an integrate-and-fire neuron that encodes its decision by firing or not firing. The rule is based on the gradient of a cost function, is found to have enhanced performance, and does not rely on a specific reset mechanism in the integrate-and-fire neuron.

1 Introduction ---

The problem of learning spatiotemporal spike patterns in networks of integrate-and-fire neurons has been intensively studied in the past few years (Seung, 2003; Fiete & Seung, 2006; Florian, 2007). In contrast to these general-purpose algorithms derived in the framework of stochastic reinforcement learning, a more specialized scenario involving only a single spiking neuron, the tempotron, has recently been considered (Gütig & Sompolinsky, 2006). Due to its specific reset mechanism, the tempotron can emit at most one postsynaptic spike per observation period, and the problem of learning a prescribed input-output behavior hence naturally reduces to that of a binary classification (spike–no spike) of the incoming presynaptic spike trains. A supervised learning rule for such classification tasks was presented in Gütig and Sompolinsky (2006) and found to converge much more quickly than general-purpose approaches such as Seung (2003).

The scope of this note is to introduce a new learning rule for the tempotron task that implements gradient descent in an appropriate cost function, converges more quickly and reliably than the original tempotron rule, and does not rely on a specific reset mechanism.

2 The Tempotron ---

We shall denote by $w = (w_j)$ the N -dimensional weight vector of the tempotron and by $X = \{X_j\}$ its input spike train, where each X_j is a finite set

of the spike times in afferent j . The leaky integration of the postsynaptic currents results in

$$v_w(t, X) = u_{\text{rest}} + w \cdot \text{PSP}(t, X), \quad (2.1)$$

where u_{rest} is the resting potential and $\text{PSP}(t, X)$ is the vector of postsynaptic potentials with components

$$\text{PSP}_j(t, X) = \sum_{s \in X_j} \epsilon(t - s).$$

The PSP kernel ϵ is assumed continuous with $\epsilon(t) = 0$ for $t < 0$. If the dependence on the input spike train is obvious, we shall abbreviate $v_w(t, X)$ as $v_w(t)$.

The tempotron spikes as soon as $v_w(t)$ reaches a threshold value that may be taken to equal zero by redefining u_{rest} . So fixing an observation period $0 \leq t < T$ and assuming that there is a postsynaptic spike in this period, we have for the spike time t_w^s ,

$$t_w^s = \min \{t \mid v_w(t, X) = 0, 0 \leq t < T\}.$$

If there is no spike, we set $t_w^s = T$. The tempotron discards any input spike arriving after its spike time (input shunting), so it sees only an effective presynaptic spike train Z_w with components

$$Z_{w,j} = \{t \in X_j \mid 0 \leq t \leq t_w^s\}.$$

The resulting membrane potential u_w is then obtained by plugging this effective spike train into equation 2.1:

$$u_w(t) = v_w(t, Z_w).$$

In the sequel, the focus is on cases where precise output spike timing is not important, only whether the neuron spikes at all. We hence set $y_w = 1$ if there is a postsynaptic spike: $t_w^s < T$, and $y_w = -1$ if $t_w^s = T$. The target behavior is encoded in a binary variable $z = \pm 1$, and the goal of learning is to find a vector of synaptic strengths w such that $y_w = z$ for the given input spike train X . Ultimately, of course, multiple associations should be learned, that is, we wish to find a single vector w such that for P different presynaptic spike trains X^μ ($\mu = 1, \dots, P$) and associated targets z^μ , the neuron emits a spike just if $z^\mu = 1$. But the fact that multiple associations are being learned is not made explicit in the notation, since we use the standard procedure of applying a learning rule for a single stimulus-response pair to multiple pairings: presenting the different pairs (X^μ, z^μ) one by one (in a

fixed or random order) and applying the single pair learning rule on each presentation.

The shunting of the inputs by a postsynaptic spike leads to a soft reset of the membrane potential $u_w(t)$, and the learning rule suggested by Gütig and Sompolinsky (2006) focuses exclusively on the time t_w^m when the maximal value of the membrane potential is achieved. Based on the vector $\text{PSP}(t_w^m, Z_w)$ of postsynaptic potentials (PSPs) at this point in time, the synaptic strengths are updated by $w \leftarrow w + \eta \Delta w$ at the end of the observation period, that is, at time T . Here η is a positive learning rate, and

$$\Delta w = (z - y_w) \text{PSP}(t_w^m, Z_w). \quad (2.2)$$

Gütig and Sompolinsky (2006) claim that this corresponds to gradient descent in the cost function

$$c_w = (z - y_w) u_w(t_w^m)$$

and invoke the chain rule to argue that the dependence of c_w and u_w on t_w^m is negligible due to the smooth reset.

The problem with this is that $u_w(t_w^m)$ is not a continuous function of the synaptic strengths, let alone a differentiable one to which the chain rule could be applied. Due to input shunting, u_w depends on the effective presynaptic spike train Z_w , and since Z_w can take on only a discrete set of values for a given stimulus, it is a discontinuous function of w (if there are presynaptic spikes in more than just a single afferent). As shown by example in Figure 1, this can lead to a discontinuous increase of c_w along the trajectory arising from the learning rule 2.2 for a set of initial conditions with nonvanishing measure.

In the example, the tempotron eventually converges to the desired state, and, indeed, this will always happen when rule 2.2 is used to learn a single stimulus-response pair. For the $z = -1$ case, when the goal is not to spike, this can be seen by observing that a Lyapunov function of the dynamics is $L_w = \int_0^T v_w(t) \Theta(v_w(t)) dt$, where Θ is the Heaviside step function and $v_w(t) = v_w(t, X)$ is the leaky current integral (see equation 2.1). In fact, each nonvanishing update by equation 2.2 decreases a synaptic weight and also decreases at some instance the leaky current integral above threshold. Although the tempotron update is hill descending for L_w , it does not descend along the gradient of this Lyapunov function. But hill descent for a single stimulus-response pair does not imply that the trajectory remains hill descending for a combination of stimulus-response pairs even when considering the small learning rate limit. So the convergence argument does not extend to the general case of learning multiple stimulus-response pairs.

It is nevertheless interesting that the Lyapunov function L_w is a functional of the leaky current integral $v_w(t)$ and not of the more directly observable

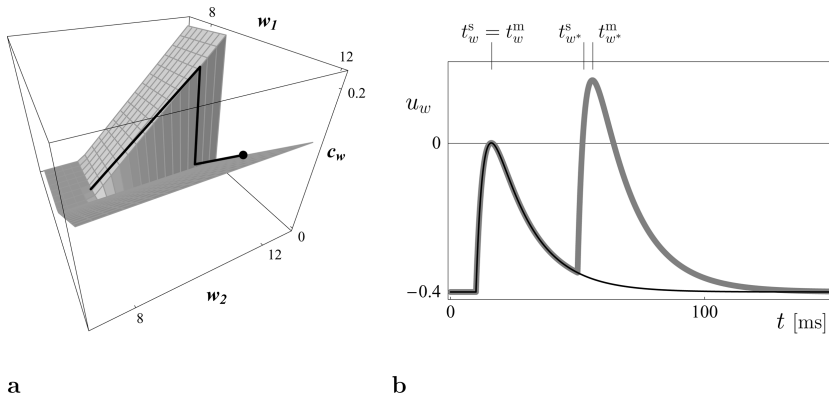


Figure 1: The tempotron update equation 2.2 leading to an upward jump in the cost function. (a) Example cost function and trajectory (black line) of the tempotron rule for a neuron with two afferents and synaptic strengths w_1, w_2 . We assume that the goal is not to emit an output spike ($z = -1$) but that there is a presynaptic spike in the first afferent and, a little later, a further presynaptic spike in the second afferent. In both afferents, initial synaptic strength (marked by the filled black circle) is large enough for a single presynaptic spike to elicit an output spike. But the spike in the first afferent arrives first, causes a postsynaptic spike, and input shunting discards the second input spike. Hence, the tempotron rule initially decreases w_1 only, with a commensurate decrease in cost. This continues up to the critical point where the membrane potential just touches the threshold. As w_1 decreases infinitesimally beyond this point, the neuron ceases to fire from the first input; this shunts the input shunting, the second input becomes visible, and the neuron fires anyway—but at a later point in time, causing the sudden increase in cost. (b) Time course of the membrane potential at the critical point in *a*. The first afferent receives a presynaptic spike at $t = 10$ and the second one at $t = 50$. (Throughout this note, time is measured in ms.) The thin black curve shows the membrane potential for $w = (w_1, w_2)$ where $w_2 = 12$ and $w_1 \approx 8.97$, the critical value just sufficient for a postsynaptic spike from the first input (details of the PSP kernel are given in appendix B). As the neuron spikes, the second input is discarded, and the membrane potential decays rapidly. The spike is caused by just touching the threshold 0, so we have $t_w^s = t_w^m$, $u_w(t_w^m) = 0$ and thus also $c_w = 0$. Since the goal is not to spike, the update changes the synaptic strengths to $w^* = (w_1 - \delta, w_2)$, where the precise value of $\delta > 0$ depends on the learning rate. But this prevents the input at $t = 10$ from causing a postsynaptic spike even for an infinitesimally small δ , and the input to the second afferent at $t = 50$ becomes apparent, yielding the membrane potential shown by the gray line (for a small $\delta > 0$). The value of w_2 is large, and so a postsynaptic spike is emitted despite the update. Now $u_{w^*}(t_{w^*}^m) > 0$, and the update has not only not prevented the postsynaptic spike but also has led to an increase in cost from $c_w = 0$ to $c_{w^*} = u_{w^*}(t_{w^*}^m)$.

membrane potential $u_w(t)$. Since the shunting reset discards inputs, the membrane potential encodes only incomplete information about the learning task, whereas any Lyapunov function provides a measure of how far away we are from task fulfillment. So it seems unlikely that the tempotron update has a Lyapunov function that is a functional of just $u_w(t)$.

3 Alternative Learning Rules for the Tempotron Task

It has been suggested that the above difficulties can be avoided by simply applying equation 2.2 to the leaky current integral instead of the membrane potential (H. Sompolinsky & R. Gütig, personal communication to W. Senn, June 2007). So one would consider the time t_w^M when $v_w(t)$ achieves a global maximum and replace equation 2.2 by $\Delta w = (z - y_w)\text{PSP}(t_w^M, X)$, hoping that this corresponds to gradient descent in $C_w = (z - y_w)v_w(t_w^M)$. The problem with this is that C_w , while continuous, is not differentiable for values of w where the leaky current integral has multiple global maxima. While such kinks in the cost function sometimes are known not to cause problems (e.g., for the classical perceptron learning rule), the situation is different since the trajectory may stick to the kink for extended periods of time. Consider the case where for $z = -1$, there are two suprathreshold local maxima. While generically just one of them will be the global one, this means that the modified update will consider the global maximum only, until the weight vector has changed to the point where the two maxima are (nearly) equal. At this point, the update will start to jump between the two maxima, not yielding gradient descent when there is a jump, and this continues until the neuron ceases to spike. Of course, decreasing the learning rate makes the trajectory stick ever more closely to the kink.

If the neuron emits an erroneous output spike, the value of $v_w(t)$ must be reduced below threshold at all points in time. Hence, for $y_w = 1$, instead of considering just a single time point, we suggest the following integral-based cost function:

$$d_w^{(1)} = \gamma(1 - z) \int_0^T \sqrt{v_w(t)} \Theta(v_w(t)) dt. \quad (3.1)$$

Here, the positive parameter γ will allow us to balance the cost for $y_w = 1$ with the cost for $y_w = -1$. As above, Θ is the Heaviside step function restricting the integration to suprathreshold values. Thus, as for the perceptron learning rule, the cost becomes zero at the decision boundary, that is, when the switch from $y_w = 1$ to $y_w = -1$ occurs. Differentiating with respect to w yields

$$-\nabla_w d_w^{(1)} = -\frac{1}{2}\gamma(1 - z) \int_0^T \frac{\Theta(v_w(t))}{\sqrt{v_w(t)}} \text{PSP}(t, X) dt. \quad (3.2)$$

Note that no second term containing the derivative of the theta function arises since the prefactor $\sqrt{v_w(t)}$ is zero for $v_w(t) = 0$. At the decision boundary, similar to the perceptron rule, the gradient is nonzero thanks to the divergent $1/\sqrt{v_w(t)}$ term but finite. The divergence requires a little attention during simulations, and the computation of such integrals with fixed step size methods is discussed in appendix A.

We next turn to the situation that the tempotron should spike ($z = 1$) but does not ($y_w = -1$). One option would be to use the tempotron update, equation 2.2, for this case. But in combination with our integrative learning rule for $y_w = 1$, it seems somewhat artificial to base the update on the single point in time when the membrane potential is maximal. Also, in a situation where the global maximum is far from threshold and there is an additional local maximum that is only slightly lower, it may be suboptimal to go just for the global maximum. In this case, it seems preferable to use a weighting function for $\text{PSP}(t, X)$ based on a soft max of the membrane potential rather than the hard max of equation 2.2. On the other hand, when the global maximum is just below threshold, there is little need for taking a local maximum into account.

These considerations lead us to suggest a soft max that adaptively becomes harder as the threshold is approached. Assuming $y_w = -1$, we first introduce the function

$$\Psi_{w,r} = \frac{1}{T} \int_0^T (v_w(t) - r)^{-2} dt, \quad (3.3)$$

where $r \geq 0$ is a regularization parameter and, in terms of this function, define the contribution to the cost as

$$d_{w,r}^{(-1)} = (1 + z)\Psi_{w,r}^{-2/3}.$$

The choice of exponents ensures that for $r = 0$, there is vanishing cost but a finite gradient at the decision boundary. For $r > 0$, the value of $d_{w,r}^{(-1)}$ is nonzero at the decision boundary, and a discontinuity arises since the cost vanishes once the output is correct. The reason for nevertheless considering $r > 0$ becomes apparent on inspection of the explicit formula for the gradient

$$-\nabla_w d_{w,r}^{(1)} = (1 + z) \frac{4\Gamma_w}{3\Psi_w^{5/3}} \quad (3.4)$$

with

$$\Gamma_w = \frac{1}{T} \int_0^T \frac{\text{PSP}(t, X)}{|v_w(t) - r|^3} dt. \quad (3.5)$$

Here, due to the cubic term in the denominator, the weighting factors of the postsynaptic potentials can have a large dynamic range in the case of

$r = 0$. Since an extremely large range is biologically implausible, there is a trade-off between realism and mathematical accuracy made explicit by the parameter r .

In summary, the proposed update of the synaptic strengths is

$$\begin{aligned} w &\leftarrow w - \eta \nabla_w d_w^{(-1)} & \text{for } y_w = -1 \\ w &\leftarrow w - \eta \nabla_w d_{w,r}^{(1)} & \text{for } y_w = 1, \end{aligned} \tag{3.6}$$

and for $r = 0$, this amounts to gradient descent in the total cost function:

$$d_w = \frac{1}{2}(1 - y_w)d_w^{(-1)} + \frac{1}{2}(1 + y_w)d_{w,r}^{(1)}.$$

4 Performance Comparison

We compared the performance of rule 3.6 to the original tempotron rule 2.2 when learning multiple stimulus-response relationships. The results are summarized in Figures 2 and 3. A striking feature of both procedures, seen in Figure 3, is the presence of very large fluctuations in the time needed for learning. This may be due to the fact that in contrast to the perceptron, the space of correct solutions is not convex for the tempotron learning task. Although the fluctuations in learning time are still large, they are much smaller for our gradient-based procedure, which learns more reliably than the original tempotron rule.

Since the tempotron rule is itself heuristic, we have also simulated the following spike-timing-based approximation: if there is no output spike, the tempotron rule 2.2 is used, but in case of an output spike, the right-hand side of equation 2.2 is replaced by $(z - y_w)\text{PSP}(t_w^s, X)$; the postsynaptic potentials at the time of spiking determine the update. Since the difference between t_w^s and t_w^m is small, this yields a performance very similar to the original tempotron rule as shown in Figure 2 (see the star). But the modified rule does not rely on input shunting, since spike time is observable for any model of an integrate-and-fire neuron.

5 Discussion

We have presented a new learning rule for the tempotron task that has two advantages compared to the original proposal: it is based on the gradient of a cost function, and we found enhanced performance (as compared to the original tempotron learning rule).¹ But how plausible are these rules as biological models?

¹The convergence of the original tempotron rule can be accelerated by including a momentum term (Gütig & Sompolinsky, 2006), but this is also expected to be the case for our rule.

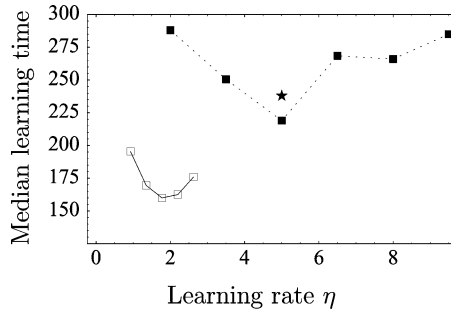


Figure 2: Tempotron learning (equation 2.2, filled squares) compared to our gradient-based learning rule (equation 3.6, empty squares) for a neuron with 100 afferents on sets of 190 random stimulus-response pairs, as detailed in appendix B. *Learning time* refers to the number of cycles through the set of patterns needed for successful learning, with the median estimated based on 500 training sets. To balance the two contributions in the cost function of our proposed rule, we used $\gamma = 0.2$. For this choice of γ , roughly the same number of errors are committed on $z = 1$ patterns as on $z = -1$ patterns during learning. The value $r = 0.05u_{rest}$ was used for the regularization parameter. The point marked by the star gives the result obtained when one modifies the tempotron rule by using t_w^s instead of t_w^m in equation 2.2 if there is a postsynaptic spike.

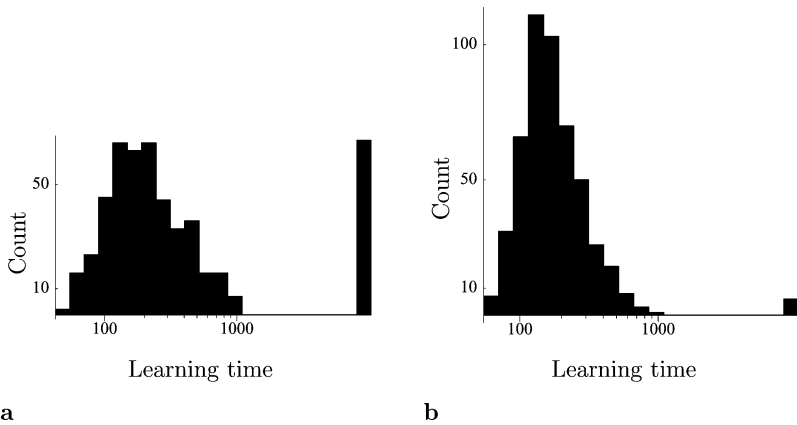


Figure 3: Distribution of learning times for the tempotron rule (a) and the proposed gradient procedure (b) for the problem considered in Figure 2. The best learning rate from Figure 2 is used for each procedure. The scale of the x -axis is logarithmic, and the right-most bar in each histogram shows the number of trials for which successful learning was not observed within 1000 sweeps through the training set. Note the number of trials where we did not observe convergence of the tempotron rule.

Locating the maximum of the membrane potential as required by the original tempotron rule is a temporally global operation. To implement the rule, each synapse would have to keep track of a state variable a representing the maximal value of u_w encountered so far, and for remembering the value its individual postsynaptic potential then had, a second variable b is needed. Whenever the current membrane potential exceeds the value of a , a gating circuitry is required to update a and b with the current values.

Since this seems complex, Gütig and Sompolinsky (2006) introduced a convolution-based rule as a biologically plausible approximation. They suggest that each synapse j computes

$$v_j = \int_0^T u_w(t) \text{PSP}_j(t, Z_w) dt. \quad (5.1)$$

At time T , synapses are then updated by using a nonlinear operation on the above integral,

$$w_i \leftarrow w_i + \eta(z - y_w) \Theta(v_j - \vartheta),$$

where ϑ is an appropriately chosen threshold. Compared to the maximum-based rule, storage capacity is now reduced by a factor of nearly two (Gütig & Sompolinsky, 2006). Further, for a task where the number of patterns was equal to the number of afferents, a roughly fivefold increase in learning times was observed.

The patterns used to obtain these results had the rather special property that there was exactly one input spike per afferent in each of the stimuli. In this case, a large value of v_j means that the membrane potential was high at the time of the single presynaptic spike in afferent j . Due to this temporal correlation, the synapse has a strong influence on the spike-no spike decision, and it makes sense that it is selected for modification via the comparison of v_j to the threshold ϑ . But this reasoning no longer holds when the number of presynaptic spikes per afferent fluctuates. Then the value of v_j is just as much influenced by the number of input spikes as by the temporal correlation and comparing to a fixed threshold is thus inappropriate. Using an adjustable threshold results in a more complex procedure since each synapse needs an additional state variable to count the presynaptic spikes.

To implement our procedure, synapses must keep track of three state variables to compute the integrals in equations 3.2, 3.3, and 3.5. The integrals, equations 3.2 and 3.5, are convolutions similar to equation 5.1, but the weighting factors are not functions of the membrane potential but of the leaky current integral. While formally well defined, in the context of input shunting, it is artificial to base learning on the leaky current integral, since it is hard to argue that $v_w(t)$ is even an observable quantity if this reset mechanism is used. But our procedure applies just as well to models

that do not discard input spikes to reset the membrane potential (Herz, Gollisch, Machens, & Jaeger, 2006). Then it is more natural to assume, as we do, an integration of the postsynaptic current that ignores the reset. From a biological perspective, this may even seem preferable to assuming input shunting.

In our rule, the weights of the postsynaptic potentials are algebraic functions of v_w , and some evaluations of algebraic functions are also needed in computing the update. In itself, this seems innocuous, since such computations can be easily implemented by chemical reaction systems. For instance, assuming mass action kinetics, steady-state concentrations of reactants have an algebraic relationship where the exponents are determined by the stoichiometry of the reactions. However, the ideal weighting functions for our rule have singularities. For the update occurring when the neuron does not spike, equation 3.4, we have explicitly addressed this by introducing a regularization parameter. We have not done this for the update resulting from an erroneous spike, equation 3.2, because there the singularity is more notational than real since the integral itself stays finite. For instance, the numerical procedure we describe in appendix A does not encounter a singularity in this case.

To present numerical results, we had to choose specific weighting functions, but this does not mean that these are the only possible, let alone the best, choices. As an example, consider the case of an erroneous spike where in the cost based on $\int_0^T f(v_w(t)) \Theta(v_w(t)) dt$, we use the specific choice $f(x) = \sqrt{x}$. But the key point is that f have an infinite tangent at $x = 0$ so that the gradient does not vanish at the decision boundary; $f(x) = \sqrt{x}$ is just a function with this property that seemed convenient.

The class of possible weighting functions increases considerably if one assumes that learning does not stop immediately when the neuron finds the desired spike–no spike behavior but continues until this behavior is robustly achieved. For this, one would assume a margin parameter $\kappa > 0$, and if the goal is to spike, stop learning only once the maximum of v_w exceeds κ (and not just when it exceeds zero). Likewise, if the goal is not to spike, learning stops only when the maximum of v_w is smaller than $-\kappa$. In the latter case, one can then use the cost $L_w^{(\kappa)} = \gamma(1 - z) \int_0^T v_w(t) \Theta(v_w(t) + \kappa) dt$. This yields a simple expression for the gradient, $\nabla_w L_w^{(\kappa)} = \gamma(1 - z) \int_0^T \Theta(v_w(t) + \kappa) \text{PSP}(t, X) dt$. For the case that the neuron does not spike when it should, using a margin makes it possible to use a weighting function with a much milder divergence in equation 3.3.

Using a margin is attractive from a biological perspective because the learned behavior then has some robustness against computational inaccuracies. It is also mathematically appealing since, just as in case of the perceptron (Anlauf & Biehl, 1989), one can then use a cost that is not just itself a smooth function but also has a smooth gradient. In contrast to the bias term in the perceptron, the resting potential in the tempotron is not

adapted. Hence, introducing a margin reduces the storage capacity and thus makes it difficult to compare performance to the original tempotron rule. This is why we have not focused on using a margin here.

To us, the biologically most problematic aspect of the above learning rules is that they are strictly episodic. In the original tempotron rule, the neuron needs to be told when a stimulus presentation begins, both to reset the shunting and to initialize the calculation of the maximum. On one hand, this implies an additional feedback signal to each synapse. On the other hand, there are situations where determining stimulus duration (i.e., temporal segmentation) is itself an important aspect of the learning problem. In these cases, it is inconvenient if the plasticity rule assumes that the segmentation problem has already been solved. While we have used an episodic formulation too, our update for the case of an erroneous spike is readily transformed into an online form. Instead of integrating over a fixed time period in equation 3.2, one just low-pass-filters the integrand, thus replacing the hard time window with a soft one. Unfortunately, this does not work so well for the rule we use when there is no spike. The ratio of two integrals is needed to compute the update, equation 3.4, and replacing the integrations by low-pass filters would amount to using approximations in a potentially highly nonlinear operation (division). Hence, deriving an efficient online algorithm for the tempotron task remains an important open problem.²

Appendix A: Numerical Integration Procedure

In calculating the update for the proposed learning rule, we encounter integrals of the form

$$\int_0^T \phi(f(t))g(t) dt,$$

where f and g are well-behaved functions but ϕ has singular points. In Euler's method, the numeric integration would be based on the approximation

$$\int_{\tau-\Delta\tau}^{\tau} \phi(f(t))g(t) dt \approx \Delta\tau g(\tau)\phi(f(\tau)),$$

²In this context, it is interesting to compare our update to the one for increasing spiking probability in an escape noise neuron (Pfister, Toyozumi, Barber, & Gerstner, 2006) when the neuron does not spike. The episodic update for this is $Z^{-1} \int_0^T \exp(\beta v_w(t)) \text{PSP}(t, X) dt$ with $Z = 1$, and since $Z = 1$, the update can easily be transformed to an online form. But if we want to benefit from a properly normalized soft-max update as done in the rule proposed here, we would have to set $Z = \int_0^T \exp(\beta v_w(t)) dt$, and then it is again difficult to find an online version due to the divisive nonlinearity.

where $\Delta\tau$ is a fixed small step size. In this case, this is unsafe when $f(\tau)$ is close to a singular point of ϕ . For efficiency, we do not want to adopt an adaptive step size and hence use the following safer approximation based on a linear interpolation of the well-behaved function f :

$$\int_{\tau-\Delta\tau}^{\tau} \phi(f(t))g(t) dt \approx \Delta\tau g(\tau) \int_0^1 \phi((1-s)f(\tau-\Delta\tau) + sf(\tau)) ds.$$

This is possible since the integral on the right-hand side can be solved analytically in all cases of interest to us.

As an example consider the update 3.2 where $\phi(x) = \Theta(x)/\sqrt{x}$. Assuming that we encounter a singular point of ϕ in the interval $(\tau - \Delta\tau, \tau)$ because $f(\tau - \Delta\tau) = -a < 0$ but $f(\tau) = b > 0$ the above formula yields

$$\int_{\tau-\Delta\tau}^{\tau} \phi(f(t))g(t) dt \approx \Delta\tau g(\tau) \frac{2\sqrt{b}}{a+b}.$$

Appendix B: Simulation Details

For $t > 0$, the PSP kernel we use is given by

$$\epsilon(t) = \frac{1}{\tau_m - \tau_s} (e^{-t/\tau_m} - e^{-t/\tau_s})$$

with $\tau_m = 15$ and $\tau_s = 3$ (time measured in ms). The value of the resting potential is $u_{\text{rest}} = -0.4$.

The observation period in the simulations described in section 5 is $T = 300$, integrated with a step size of $\delta t = 0.1$. For each of the random stimuli used, each afferent receives a random number of zero, one, two, or three presynaptic spikes at randomly chosen times within the observation period. Initially the weights were set to the constant value $w_i = 55/N$, resulting in an output spike for roughly half of the patterns prior to learning.

Acknowledgments

We acknowledge fruitful discussions with Wulfram Gerstner and Jean-Pascal Pfister and valuable comments of Robert Gütig and Haim Sompolsky.

References

Anlauf, J., & Biehl, M. (1989). The AdaTron: An adaptive perceptron algorithm. *Europhysics Letters*, 10, 687–692.

- Fiete, I., & Seung, H. (2006). Gradient learning in spiking neural networks by dynamic perturbation of conductances. *Phys. Rev. Letts.*, *97*, 048104.
- Florian, R. (2007). Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural Computation*, *19*, 1468–1502.
- Gütig, R., & Sompolinsky, H. (2006). The tempotron: A neuron that learns spike timing-based decision. *Nature Neuroscience*, *9*, 420–428.
- Herz, A., Gollisch, T., Machens, C., & Jaeger, D. (2006). Modeling single-neuron dynamics and computations: A balance of detail and abstraction. *Science*, *314*, 80–85.
- Pfister, J., Toyozumi, T., Barber, D., & Gerstner, W. (2006). Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural Computation*, *18*, 1318–1348.
- Seung, H. (2003). Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, *40*, 1063–1073.

Received September 7, 2007; accepted June 2, 2008.