Reinforcement learning in populations of spiking neurons

Robert Urbanczik, Walter Senn

Department of Physiology, University of Bern, Bühlplatz 5, CH-3012 Bern, Switzerland

Abstract

Population coding is widely regarded as a key mechanism for achieving reliable behavioral responses in the face of neuronal variability. But in standard reinforcement learning a flip-side becomes apparent. Learning slows down with increasing population size since the global reinforcement becomes less and less related to the performance of any single neuron. We show that, in contrast, learning speeds up with increasing population size if feedback about the population response modulates synaptic plasticity in addition to global reinforcement. The two feedback signals (reinforcement and population-response signal) can be encoded by ambient neurotransmitter concentrations which vary slowly, yielding a fully online plasticity rule where the learning of a stimulus is interleaved with the processing of the subsequent one. The assumption of a single additional feedback mechanism therefore reconciles biological plausibility with efficient learning.

Key words: Reinforcement learning, population coding, spiking neuron

1 Introduction

It borders on a truism that information processing in the brain is highly distributed in order to achieve reliable behavioral responses despite neuronal variability. Consequently the role of neuronal populations in encoding sensory stimuli and the associated problem of decoding the population activity has been intensively studied (see [1,2] for reviews). For investigating reward based learning, a common strategy has been to use aggregate descriptions of population activity, where the population is in effect treated as a single rate

Email addresses: urbanczik@pyl.unibe.ch (Robert Urbanczik), senn@pyl.unibe.ch (Walter Senn).

based unit [3]. But the resulting plasticity rules involving pre- and postsynaptic population rates are difficult to interprete at the level of a single spiking neuron. On the other hand, most models of reinforcement learning which explicitly deal with spiking neurons have focused just on single neurons or small neuronal assemblies [4–6].

The reason for these discrepancies becomes apparent when one considers standard approaches to reinforcement learning where a single global reward signal assesses the response of an entire neural network. Applied to a population, where the read-out is by design relatively invariant to the behavior of any single neuron, the reward signal evaluating the population response cannot reliably assign credit at the level of a single neuron or even a single synapse. In human terms, the standard reinforcement approach is analogous to having a class of students write an exam and being informed by the teacher on the next day only whether the majority of the class has passed or failed whereas the individual scores are kept secret. That this leads to slow learning is highlighted by the otherwise biologically plausible simulations reported in [7]. There, a large network of integrate and fire neurons was trained to associate a single stimulus with one of just two responses. To achieve an 80% probability of a correct response, more than 100 presentations of the stimulus where required and performance did not improve with training extended beyond this point. In contrast to this, behavioral results indicate that reinforcement learning can be reliable and fast. Macaque monkeys, for instance, correctly associate one of four complex visual scenes with one of four targets after a total of just 12 presentations on average [8].

To some extent the population learning problem can be sidestepped if plasticity is confined to the read-out with the population neurons themselves just serving as fixed feature detectors. This seems adequate for basic sensory-motor integration tasks when stimuli can be described by a few features such as spatial location or angle. Further, in this case, the topographically organized lateral connectivity observed in sensory areas can provide an effective way of damping neuronal response variability. But for learning complex stimuli a prohibitively large population size is necessary if only the read-out is plastic. In particular, for feature neurons with Gaussian tuning curves the flexibility of read-out learning is severely compromised unless the population size increases exponentially with the stimulus dimension [1]. It thus seems unlikely that plasticity in higher cortical areas is confined to a population read-out.

Instead of a broadcasting a single reward signal, learning procedures in artificial intelligence (such as the back-propagation algorithm) use an involved machinery to compute individualized feedback signals for each neuron [9,10]. Our objective is to point out that for learning in a population of spiking neurons there is a large and fertile middle ground between such complex and biologically unrealistic procedures and the standard reinforcement approach.

For this, we present gradient based learning schemes where synaptic plasticity is modulated not just by reward feedback but also by a single additional feedback signal encoding the population response. Since the two signals change on similar time scales, reward and population feedback can be provided by similar mechanisms, e.g. ambient neurotransmitter concentrations. Applied to different neuronal coding strategies such as firing rate or latency, the scheme results in slightly different learning rules, suggesting that there ought to be a match between coding and plasticity. With regard to performance, the key finding is that the additional modulation by the population response dramatically changes the scaling properties of population learning. Instead of becoming slower and slower with increasing population size, as for standard reinforcement procedures, learning now speeds up when the population size is increased.

2 Results

2.1 Coding in a population

Behavioral decisions are hardly controlled by the postsynaptic output of a single neuron. For robustness, and in the case of mean firing rates codes also for speed, one needs to consider a population of N neurons, each encoding more or less the same information about the stimulus. We shall denote by w^ν ($\nu = 1, \ldots, N$) the synaptic vector of the ν-th neuron and assume that its presynaptic input is a spike pattern X^ν. So the neurons in the population need not have exactly the same input, but we assume that the X^ν are highly correlated. This allows for variations in the connectivity to the input layer as shown in Fig. 1.

Different neurons will produce different postsynaptic spike trains Y^ν and aggregating these into a population response must be based on how neurons encode information. Since different encoding strategies are likely to exist, we adopt a general approach by assuming a scoring function $c(Y^\nu)$, which assigns a numerical value to any spike train. Aggregating the neural responses then amounts to simply adding the scores. Possible choices for the scoring function are counting the number of spikes (a pure firing rate code) or the time elapsed between the start of stimulus and the first spike. While we shall consider different scoring functions, initially, and for most of the paper, we shall assume the following spike/no-spike code: $c(Y^\nu) = 1$ if the neuron produces one or more postsynaptic spikes, otherwise $c(Y^\nu) = -1$. This is also the coding assumed in Fig. 1. We shall say that the population response determining behavioral decisions is 1, if the majority of neurons has $c(Y^\nu) = 1$ (otherwise the population response is -1).

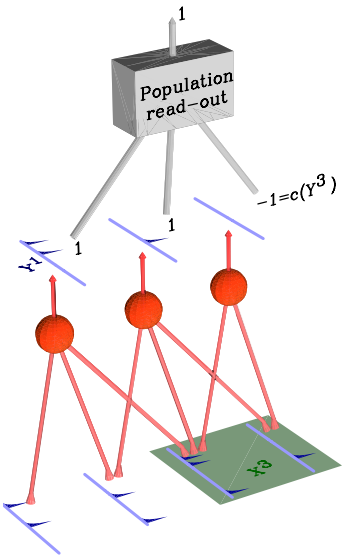

Fig. 1. Sketch of a population of spiking neurons and the corresponding population read-out. Each neuron is connected to only a part of the input layer, neuron 3, for instance, just responds to the spike pattern X^3 underlayed in green. The majority of the neuronal decisions determines the population read-out. The implementation of this read-out is considered in the Discussion. Learning modifies the synaptic strengths, driving the individual neurons (red) to achieve a correct population response.

2.2 Learning in the single neuron

To learn from trial and error different responses to a given stimulus must be explored and, for this, randomness in the neuronal activities provides a convenient mechanism. In numerical simulations we shall assume as specific mechanism the fluctuating spike threshold provided by the escape noise model (Methods) but our main findings do not depend on the details of the reinforcement learning procedure at the single neuron level.

We assume that plasticity can be understood as updating the vector of synaptic strengths w in the neuron to modulate the log-likelihood $\mathcal{L}_w(Y_t|X)$ of producing the postsynaptic spike train Y_t upto time t in response to an input spike pattern X. (We have dropped the neuron index ν, since we deal with only a single neuron in this section). In the simplest learning scenario, the i-th synapse compute its eligibility

$$e_i(t) = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial}{\partial w_i}\mathcal{L}_w(Y_t|X)$$

and updates the synaptic strength by $\dot{w}_i = -\eta e_i(t)$ in case of an erroneous response. This decreases the odds of repeating the same mistake again. The parameter η, controlling the magnitude of the update, is called the learning

rate. For the escape noise neuron, $e_i(t)$ depends on pre- and postsynaptic spike timing and on the value of the membrane potential at time t (see Methods and [11]).

To learn based on $e_i(t)$, however, requires that information about reward is provided instantaneously at each point in time. In a realistic setting, success or failure only become apparent once the entire stimulus encoded by the spike trains in X has been processed. Mathematically, the most convenient way to deal with this is to integrate $e_i(t)$ over the stimulus duration and to use this for updating the synapses after the stimulus has been presented. To calculate this so called eligibility trace, however, each synapse would need to know when the stimulus starts and ends in order to initiate and terminate the integration. Since this requires intricate feedback mechanisms, we replace the hard time window of the integral by a soft time window and obtain the eligibility trace as a running mean. This is achieved by having each synapse low pass filter $e_i(t)$, computing its eligibility trace E_i as

$$\tau_{\mathrm{M}}\dot{E}_i = e_i(t) - E_i\,, \tag{1}$$

where the time constant τ_{M} controls the memory length of the synapse. The time constant should be roughly matched to the duration of the stimuli and we assume $\tau_{\mathrm{M}} = 500\,\mathrm{ms}$ in all simulations.

Based on the above eligibility trace , we will eventually present a fully on-line theory of learning where no explicit information about stimulus onset and termination is needed at the level of the synapse. But, for the sake of clarity, we will initially assume that synaptic changes occur only at times T when a stimulus ends (e.g. via $\Delta w_i = -\eta E_i(T)$ if there is an error).

2.3 Episodic learning in a population

In this section we shall assume that immediately after a stimulus presentation has ended, feedback becomes available and synaptic updates occur only then. To learn a fixed number of prescribed stimulus-response associations, these learning episodes are repeated with a different stimulus-response pair used in each episode. The goal of learning of course is to obtain the correct population response for each of the associations.

In the standard reinforcement learning approach one assumes a critic producing a reinforcement signal R which indicates whether the population response is correct ($R = 1$) or not ($R = -1$). For a stimulus presentation ending at time T, the synaptic update occurring then is

$$\Delta w_i^\nu = \eta(R - R_{\mathrm{base}})E_i^\nu(T)\,, \tag{2}$$

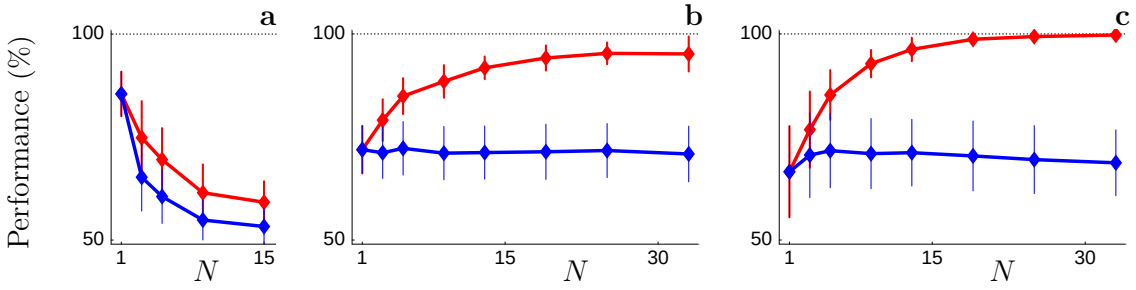

Fig. 2. The performance (percentage of correct responses) after a fixed number of learning episodes as function of the population size N. The red curves show the performance of the population read-out, the blue ones show the average performance of the single neurons. **a**: Learning based just on global reward, Eq. (2). **b**: Learning with individual reward for each neuron, Eq. (3). **c**: Attenuated learning with individual reward, Eq. (5). In all cases 30 patterns had to be learned with target responses equally split between the two output classes ± 1. The pattern statistics are detailed in Methods. In each learning episode a randomly selected pattern was presented; the number of episodes was 5000 in Panel **a** and 2000 for Panels **b**,**c**. The performance values shown are averages over between 100 ($N = 1$) and 20 ($N = 33$) learning tasks, with a different set of patterns and different initial synaptic strengths in each task. The error bars represent the standard deviation of the performance fluctuations from task to task. The corresponding 1 SEM values for the mean are much smaller, in fact smaller than the symbols used in the plot. The error bars shown demonstrate that learning becomes more reliable with increasing N (for Panels b and c) since the task to task fluctuations in the population performance decrease.

where the eligibility trace $E_i^\nu(T)$ is obtained by using (1) for each neuron in the population. The reinforcement baseline $R_{\rm base}$ balances reinforcement and punishment. We shall use $R_{\rm base} = -1$, i.e synaptic updates only occur if the population response is wrong. For binary decision problems this is a robust choice whereas reinforcing correct behavior requires a careful tuning of the amount of reinforcement to the progress in learning.

We have evaluated this rule for different population sizes after a fixed number of learning episodes. The performance of the population (percentage of correct responses) is shown in Fig. 2a and compared to the average performance of the individual neurons. While for any given population size $N > 1$ the population outperforms the average single neuron, population performance nevertheless deteriorates quickly with increasing N. The reason for this is rather simple: From the perspective of the single neuron, the global reinforcement signal R is an unreliable performance measure, since the neuron may be punished for a correct response just because other neurons made a mistake. The odds of this happening increase with the size of population, average single neuron response deteriorates and this is not compensated for by the boost provided via the population response.

To investigate what level of performance is achievable in principle by a population, we have considered alternatives to the standard reinforcement prescription (2). The perhaps simplest approach is to train the neurons individually and to only use the population read-out to boost recognition. For this we assume an individual reinforcement signal $r^\nu = \pm 1$ indicating whether neuron ν did the right thing and use

$$\Delta w_i^\nu = \eta(r^\nu - 1)E_i^\nu(T) \tag{3}$$

for the synaptic updates. As shown in Fig. 2b, average single neuron performance now no longer deteriorates with increasing population size, and it is in fact independent of N. In contrast to this, population performance improves with increasing N and reaches a quite high level.

While, compared to global reward, individual reward works far better, the performance nevertheless saturates rather quickly with increasing N. The neurons are all trying to learn the same thing and this leads to correlations which are detrimental to the population performance. To address the correlation problem, we consider attenuating the learning once the population response is reliable and correct. The reliability can be assessed by introducing the population signal

$$S = \frac{1}{\sqrt{N}} \sum_{\nu=1}^{N} c(Y^\nu) \; . \tag{4}$$

The $1/\sqrt{N}$ normalization reflects the fact that, given the stimulus, neuronal responses are conditionally independent. Hence, the fluctuations in $\sum_{\nu=1}^{N} c(Y^\nu)$, due to the noisy neural processing, will be on the order of $\sqrt{N}$. In particular, if the absolute value of S is large, the sign of the population signal S (the population response) is unlikely to fluctuate and the response is reliable. The following rule implements learning attenuation

$$\Delta w_i^\nu = \eta a \, (r^\nu - 1) \, E_i^\nu(T) \quad \text{with } a = \begin{cases} 1 & \text{if } R = -1 \\ e^{-S^2} & \text{if } R = 1 \, . \end{cases} \tag{5}$$

If the population is wrong, this is the same update as in (3), whereas, due to the attenuation factor a, only a small step is made if the population output is reliable and correct. So, as for individual reward, a neuron does not adapt when it votes correctly. But even when it is wrong, only small synaptic changes occur, if the majority of the other neuron do the right thing. Due to learning attenuation, perfect performance is now approached with increasing population size without saturating earlier (Fig. 2c). The update (Eq. 5) can be understood as a gradient descent rule (Methods and Supplementary Information).

A population, properly trained, has an additional benefit besides improved average performance. The error bars in Fig. 1 measure the fluctuations in per-

formance from task to task, were a different set of stimuli-response pairs and different initial synaptic strength where used for each task. These fluctuation decrease with population size (more pronouncedly in Fig. 2c than Fig. 2b), showing that it becomes less likely that learning unexpectedly fails for a specific set of patterns. This is a definite advantage when failing to learn a specific task incurs a severe penalty (e.g. getting eaten).

2.4 On-line learning in a population

At first sight it might seem that providing an individual reward to each neuron in a population requires a biophysically implausible number of feedback signals. However just two feedback signals, global reward (R) and the strength of the population response (S) are needed, if each neuron keeps a memory of its past spiking behavior. For instance, if most neurons in the population spiked erroneously (i.e. $S > 0, R = -1$) a neuron ν that stayed silent, $c(Y^\nu) = -1$, did the right thing and therefore $r^\nu = 1$. More generally, the individual reward signal for neuron ν has the form

$$r^\nu = \mathrm{sign}\Big(R\, S\, c(Y^\nu)\Big)\ , \tag{6}$$

and hence differs from the global reward signal only if the neuron's response is at variance with the population response.

Based on this observation, we now present a fully on-line learning rule, where the delivery of feedback is explicitly modeled by changes in ambient neurotransmitter concentrations. These implicitly also encode information about the stimulus duration. Now, synapses can change in continuous time showing that there is no need to assume that plasticity is explicitly triggered by stimulus endings. Since the biophysical machinery required in this framework for plain learning with individual reward (Eq. 3) is essentially the same as for attenuated learning (Eq. 5) we shall focus on adapting the better performing rule (5). The overall feedback structure is sketched in in Fig. 3a.

We assume that in the absence of any reinforcement information the concentration c_{rew} of the neurotransmitter signaling reward (e.g. dopamine) is maintained at a homeostatic level where a baseline release rate is balanced by a linear degradation with time constant $\tau_{\mathrm{rew}} = 10\,\mathrm{ms}$. Reinforcement information leads to a step increase ($R = 1$) or decrease ($R = -1$) in the release rate for a duration $L_{\mathrm{rew}} = 50\,\mathrm{ms}$, whereafter the release returns to its baseline level. In due course, the changes in release rate are reflected in the ambient concentration level c_{rew} (Fig. 3b, green curve) providing one signal which modulates synaptic plasticity. In Fig. 3b, and in most of the paper, we assume that the change in release rate is triggered immediately at the end of stimulus presentation ($\Delta_{\mathrm{rew}} = 0$) but our model is robust to a modest delay in the onset of

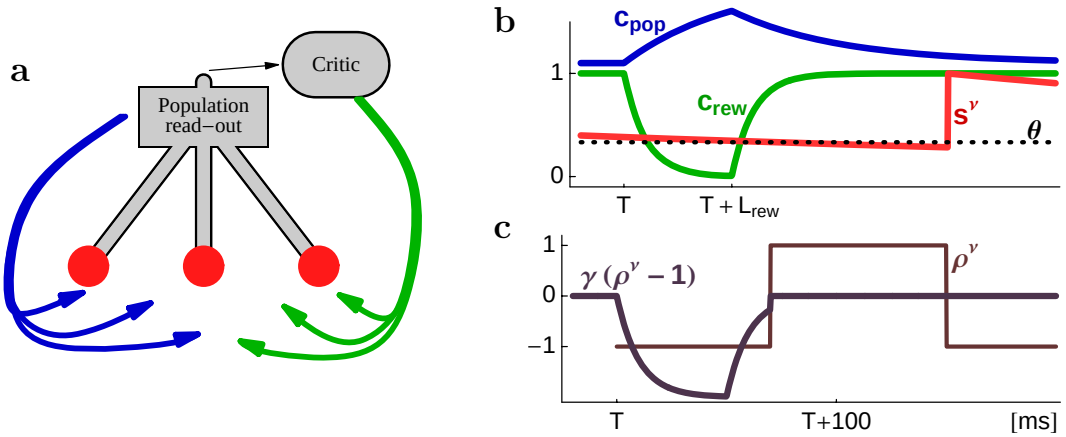

Fig. 3. On-line feedback signals interact with synaptic memory to modulate plasticity. Panel **a** sketches the overall feedback structure. In Panel **b** an example for the temporal evolution of the corresponding chemical concentrations is shown. The reward feedback $c_{\rm rew}$ and the population feedback $c_{\rm pop}$ are drawn assuming that the population signal $S > 0$, i.e. the majority of the neurons fired in response to the stimulus X ending at time T, and that this was the incorrect response ($R = -1$). The memory variable s^ν of a neuron which fired quickly in response to X is also shown. Since s^ν is above threshold for a while after time T, the neuron remembers that its responded to X by spiking. But, since it fired early, the neuron forgets this around time T+80, leading to the first jump of ρ^ν in Panel **c**. In plotting s^ν we have additionally assumed that the neuron happens to spike (at $T + 150$) in response to the current stimulus. In the considered scenario, the episodic variable r^ν equals -1 and the value of ρ^ν is thus correct initially. But since the neuron first forgets that it spiked and then spikes again, ρ^ν later changes to 1 and then flips back to -1 again. But these complications only happen at times when $c_{\rm rew}$ is again close to its homeostatic value. They hence have only a minor effect on $\gamma(\rho^\nu - 1)$ in Panel **c**, the total feedback controlling the synaptic update. For Panel **b** we have assumed no delay in reward onset ($\Delta_{\rm rew} = 0$). This is also the case in the simulations unless delayed onset is mentioned explicitly.

the reward signal ($\Delta_{\rm rew} > 0$).

Feedback about the population output is provided in a similar fashion, via the concentration level $c_{\rm pop}$ of a second neurotransmitter (Fig. 3b, blue curve). Again, the end of stimulus presentation triggers a step change in the release rate of this transmitter for a duration $L_{\rm pop} = 50\,{\rm ms}$, but now we assume that the magnitude as well as the direction of the change is controlled by the value of the population signal S (Methods). Once the release rate has returned to its baseline level, a homeostatic value of $c_{\rm pop}$ is again approached, with time constant $\tau_{\rm pop} = 50\,{\rm ms}$.

Finally, we consider the memory mechanism for the past spiking behavior enabling each neuron to determine $c(Y^\nu)$. For this, we assume a calcium like variable s^ν decaying as $\tau_{\rm M}\dot{s}^\nu = -s^\nu$ when neuron ν does not fire. But if there

is a postsynaptic spike at time t the concentration is updated to $s^\nu(t) = 1$. So the value of s^ν is directly related to the time elapsed since the last spike and comparing it to an appropriate threshold θ yields an indication if the neuron fired in response to the stimulus. Note that we are assuming the same time constant τ_M as for the synaptic eligibility trace (Eq. 1), since in both cases the relevant time scale is the typical length of a stimulus.

Based on $c_{\rm rew}$, $c_{\rm pop}$ and s^ν, an approximation ρ^ν to the individual reward signal r^ν in Eq. (6) can now be computed at the synaptic level (Fig. 3c). Explicitly, we use

$$\rho^\nu = \mathrm{sign}\Big(c^*_{\rm rew}\, c^*_{\rm pop}\, (s^\nu - \theta)\Big)\ , \tag{7}$$

where $c^*_{\rm rew}$ and $c^*_{\rm pop}$ are the deviations of the neurotransmitter concentrations from their respective homeostatic levels. Hence, $c^*_{\rm rew}$ and $c^*_{\rm pop}$ can be positive or negative and decay to zero once enough time has elapsed since stimulus presentation. For appropriate values of the threshold θ (and of the time constant $\tau_{\rm M}$) the value of $\rho^\nu(t)$ is a good approximation to r^ν for quite a while after the end of the stimulus. But the approximation becomes less reliable as time goes by (reflected by the changing value of ρ^ν in Fig. 3c). Large synaptic updates are confined to a time window after the reward signal by introducing the factor

$$\gamma = \begin{cases} |c^*_{\rm rew}| & \text{if } c^*_{\rm rew} < 0 \\ |c^*_{\rm rew}||c^*_{\rm pop}| & \text{if } c^*_{\rm rew} > 0 \end{cases}$$

where $|c^*_{\rm rew}|$ ensures eventual decay to zero. Further, the inclusion of $|c^*_{\rm pop}|$ for positive $c^*_{\rm rew}$ implements the attenuated learning from reward in analogy to Eq. 5. In terms of these quantities the on-line version of (5) for the synaptic updates is now simply given by

$$\dot{w}^\nu_i = \eta\, \gamma(t)\, (\rho^\nu(t) - 1)\, E^\nu_i(t)\,. \tag{8}$$

An example of the effective feedback determining synaptic plasticity is shown in Fig. 3c. Using Eq. 8, we arrive at a fully on-line scheme where the learning of the previously presented stimulus occurs concurrently with the processing of the current one.

Simulations results comparing the on-line procedure to episodic learning show that using our biologically reasonable model for the feedbacks hardly slows down learning (Fig. 4a, red vs black). We also tested the learning with stimuli of variable lengths (duration randomly chosen between 400 and 600 ms). In contrast to the 500 ms stimuli used in the previous simulation, the time constant $\tau_{\rm M}$ for the eligibility and the memory traces is now no longer precisely matched to stimulus duration. The green learning curve shows that the on-line procedure is insensitive to such deviations. For a further check on robustness, we simulated delayed onset of reward by setting $\Delta_{\rm rew}$ to 100 ms, corresponding to 20% of stimulus duration. Even though reward onset now occurs during the

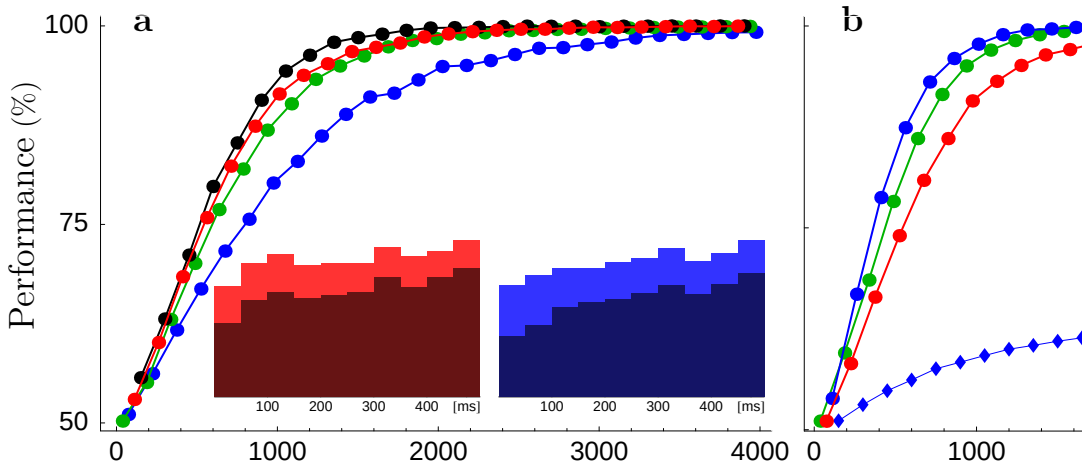

Fig. 4. Learning curves (performance vs. number of pattern presentations) for on–line reinforcement learning. **a:** The red curve shows the performance of the on-line procedure on the same tasks as in Fig 1. For comparison, the results obtained with episodic learning (Eq. 5) are given by the black curve. To check for robustness, we tested the on-line procedure on variable length patterns (green curve) with a duration of between 400 and 600 ms (in contrast to the fixed 500 ms length previously assumed). We also simulated delayed onset of reward ($\Delta_{\rm rew} = 100$ms, blue curve) for the fixed length patterns. The insets show the distribution of postsynaptic spike times after learning without (red) and with (blue) delay in reward onset. The x-axis is time elapsed from start of stimulus to spike; the contributions from the patterns where the goal is to spike is highlighted by the use of a dark color. The histograms are based on the postsynaptic spikes of all neurons in the population. All results in the panel are for $N = 33$, averaged over 20 tasks. **b:** Performance for different population sizes: $N = 33$ (red, same curve as in Panel a), $N = 67$ (green), $N = 135$ (blue circles). The blue diamonds represent mean single neuron performance for $N = 135$.

presentation of the subsequent stimulus, perfect performance is nevertheless approached (blue curve in Fig. 4a). But there is a noticeable slow down in learning. While it may be possible to improve performance by adjusting the learning rate or the memory time constant $\tau_{\rm M}$ to the additional delay, to focus on robustness, we refrained from such re-tuning and used the same parameter values in the on-line procedure for the three kinds of tasks.

Postsynaptic spike timing after learning (Fig. 4a, insets) is distributed quite uniformly with only a slightly reduced frequency towards the start of the stimuli. The noticeable difference in activity between the spike and the no-spike patterns, already in the first few time bins of the histograms, shows that the eligibility trace can bridge a delay of some 500 ms between action and reward delivery.

The on-line procedure speeds up considerably with increasing population size as shown in Fig. 3b. But the figure also highlights a second advantage of a large population. Essentially perfect population performance is attained despite of

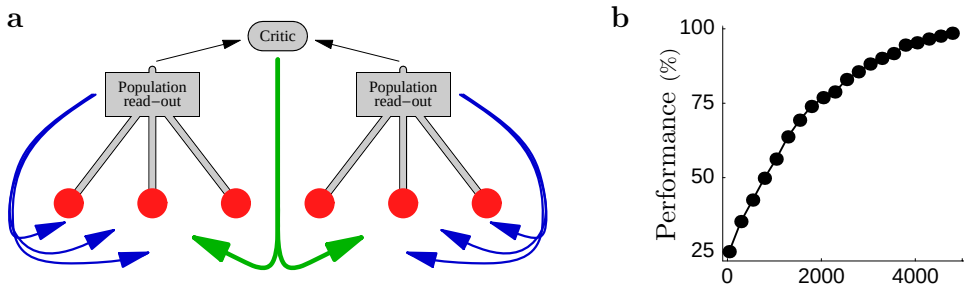

Fig. 5. Learning of 4-way decisions by two populations. **a**: Sketch of the feedback structure used for the task. **b:** Learning curve for two populations with $N = 25$ neurons each. The number of patterns learned was 24, with 6 pattern allocated to each of the four output classes. The reported values are averages over 20 tasks.

the fact that mean single neuron performance does not increase much during learning. So, in a large population, the single neuron has to learn only very little and can stay close to a homeostatic regime of operation.

For a fixed number of neurons in the population and a given synaptic load, one expects total learning time to scale linearly with problem size. So, if twice as many patterns have to be learned but neurons have also twice as many afferents (thus doubling number of synaptic weights), the number of times each pattern needs to be presented in order to achieve a given performance level should not increase. For the $N = 33$ population, we checked that this is the case (data not shown).

2.5 Flexibility of the on-line scheme

Behavioral adaptation takes many different forms, and there is certainly more to learning than just binary decisions tasks based on a spike/no-spike code. While the network architecture may be task dependent, it seems unlikely that the underlying synaptic learning mechanisms are highly specific. Here we provide some examples showing that the above on-line scheme applies to different learning scenarios with little or no modification.

We first consider the non-binary case where one of $n > 2$ responses must be chosen based on a stimulus. Within the current framework this can be addressed by assuming that there are several (m) populations of neurons, each responding with a binary decision to a stimulus. The behavioral response is determined by the combined output of the populations and can thus have one of $n = 2^m$ values. The global reinforcement signal R then encodes whether this combined output is correct. We now assume that each of the m populations has its individual population feedback and then use the above online procedure for each of the populations. So, as sketched in Fig. 5a learning at the level of the single neuron is based on its own response to the stimulus, on

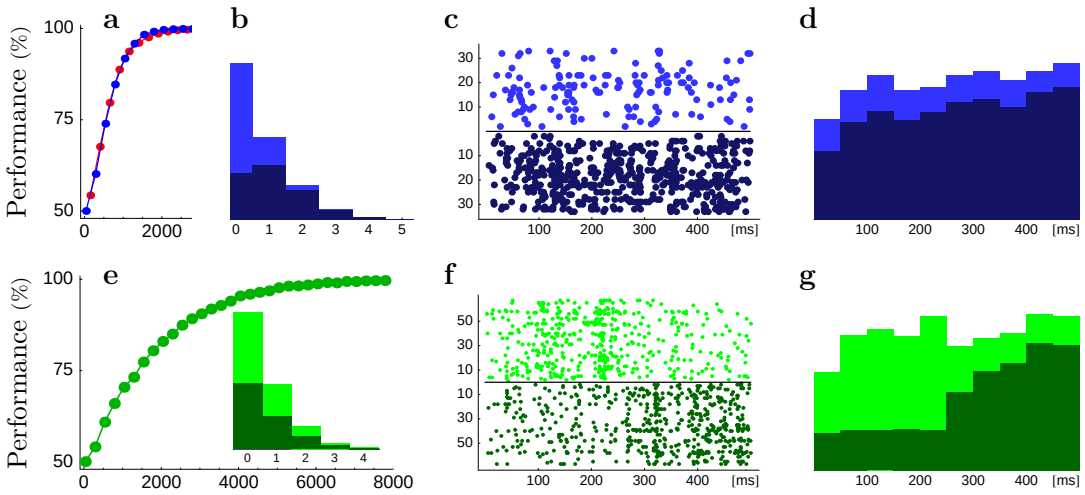

Fig. 6. Different coding strategies for a binary decision task. **a:** Learning curve (blue) when the population read-out assumes a firing rate code ($N = 33$ and 30 patterns). For comparison the corresponding curve for the spike/no-spike code is shown in red (same curve as in Fig 3). After learning with the firing rate code, **b:** distribution of the number of spikes within stimulus length (500 ms), **c:** scatter-plot of the firings for each neuron, **d:** distribution of the spike times. In the three panels dark (light) blue gives the contribution from the patterns with target output 1 (-1). **e:** Learning curve for the spike-early/spike-late code ($N = 67$, 30 patterns) and the corresponding spike number histogram (Panel **e**, inset), firing scatterplot (Panel **f**), and spike timing histogram (Panel **g**). Dark (light) green is for target 1 (-1) patterns. The values reported are averages over 20 tasks except in the scatter-plots, which are for a single trained population.

feedback about the output of the population it belongs to, and on the global reinforcement assessing the behavioral response. Simulation results for two populations learning a 4-way decision task are shown in Fig. 5b. While 4-way decisions are harder than binary decisions since the combined output is incorrect if just one of the two population responses is wrong, learning nevertheless succeeds rather quickly.

Next we investigate the use of different coding strategies at the level of the single neuron. Until now we have assumed that in decoding postsynaptic spike trains the population read-out only considers whether the neuron does or does not fire. While this spike/no-spike code suggests itself for its theoretical simplicity other codes are possible as well. In particular, for a proper firing rate code we redefine the scoring function $c(Y^\nu)$ to be the number of spikes in the output spike train of neuron ν. Correspondingly we redefine the population signal (Eq. 4) by setting

$$S = \frac{1}{\sqrt{N}} \left(\sum_{\nu=1}^{N} c(Y^\nu) - \vartheta \right)$$

with a threshold $\vartheta = \frac{2}{3}N$. Since the population response is the sign of S, choosing a threshold which is greater that $\frac{1}{2}N$ takes into account that a neuron may occasionally spike more than once in response to a stimulus. As the learning curve in Fig. 6a shows, using a firing rate code instead of the spike/no-spike code does not discernibly change performance. Note, that for simplicity we are still using the same learning mechanisms at the neuronal and synaptic level as for the spike/no-spike code. So the memory trace s^ν of each neuron now provides only a rather rough approximation of $c(Y^\nu)$, since the exact number of postsynaptic spikes cannot be determined from s^ν. But since the output activity level is fairly low (Fig. 6b) the limited information encoded in s^ν is sufficient for learning to succeed.

As inputs we have throughout assumed fixed low activity spike patterns, so the neuronal outputs are highly dependent on relative input spike times. But the two output codes considered upto now do not take postsynaptic spike timing into account and there is thus a code switch between inputs and outputs. While this could be a avoided by using mean firing rate inputs, it is nevertheless of interest to ask if population learning itself can be based on a spike timing dependent output code. For this, we study a spike-early/spike-late code. In particular we use as scoring function $c(Y^\nu) = 1$ if there are more spikes in Y^ν during the second half of stimulus duration than during the first, otherwise $c(Y^\nu) = -1$. (For an equal number of spikes and in the case of no spike, $c(Y^\nu) = 0$). Since the output code is now balanced around 0 we revert to our standard measure of population activity (Eq. 4). As shown in Fig. 5e population learning can be based on such a spike timing dependent output, even if it is slower than for the rate codes. The spike early/spike-late coding is easily seen as a the difference between target 1 and target -1 patterns in the timing histogram (Fig. 3g) whereas firing rates do no distinguish between the two target classes (Fig. 5e, inset).

For learning to succeed with the timing dependent code the neuronal and synaptic memory traces had to be modified. First, a larger value of the threshold θ in (Eq. 7) is needed since the distinction is now between an early and a late firing and not between spike/no-spike. But a more subtle modification is necessary as well. Assume there are two or more spikes in the spike train Y^ν and the last spike occurs in the second half of stimulus duration. With our usual computation for the neuronal trace variable s^ν, the last spike erases the memory of all previous spikes due to the deterministic update. Hence, the neuron only remembers that it spiked late. This creates a systematic mismatch to the value of $c(Y^\nu)$ which depends on the number as well as on the timing of spikes. Since this mismatch can prevent successful learning, as a simple remedy we used a stochastic update of the trace s^ν: A postsynaptic spike at time t, does not always update s^ν to 1 but only with a probability of $1 - s^\nu(t)$. Hence the memory about early spikes may be retained even if the neuron fires during the second half of the stimulus duration. While the stochastic update does not

ensure that the memory trace correctly reflects the score $c(Y^\nu)$, the errors are now no longer systematic and the simulations show that the resulting memory trace is reliable enough for successful learning (Fig. 6).

The stochastic memory trace is more flexible than the deterministic one, since it can also be used in conjunction with the other coding strategies. The flexibility, however, comes at a price. For the spike/no-spike code, we observed a twofold increase in learning times when using the stochastic instead of the deterministic memory trace (data not shown).

3 Discussion

We have presented a theory of reinforcement learning in populations of spiking neurons where synaptic plasticity is modulated by global reinforcement, feedback about the population response as well as a memory trace encoding the neuron's past firing behavior. Learning now speeds up with increasing population size, in contrast to the case where only global reinforcement is available. In presenting simulation results we have assumed a specific neuronal model and synaptic reinforcement procedure, the escape noise neuron presented in [11]. The model suggests itself because of its flexibility and because it leads to a spike-timing dependent learning rule. However, our population approach is not confined to this synaptic plasticity model and could readily be adapted to use other reinforcement learning procedures [4–6] at the single neuron level. Indeed, the population neurons could even be tempotrons with the associated supervised plasticity rule [12]. But the tempotron applies just to the episodic learning of binary decisions and these restrictions would then equally apply to the population learning. Obviously, in absolute terms, population performance will depend on the specifics of the neuronal model and the associated plasticity rule. But considering the scaling of the performance we expect our findings to be generic: With just global reinforcement performance degrades with increasing population size, but it improves when plasticity is properly modulated by the population response and the neuronal memory trace.

These results also throw light on the biophysical mechanisms implied by mean firing rate models of learning where, for processing speed, the postsynaptic rate is often taken to represent the average over a population of neurons (in lieu of a single neuron average over an extended time period). But then, even in cases where the mean firing rate description of the spiking population is itself carefully established [13], it is often unclear what the mean firing rate plasticity rule actually means at the level of the single spiking neuron. In particular, this interpretational conundrum arises when plasticity is modulated by the mean postsynaptic rate, as in e.g. Hebbian learning. Since this rate is really a population average it is (i) not immediately available at the synaptic level

and may (ii) be at variance with the true postsynaptic behavior of any single neuron. Our model resolves this conundrum, with regard to the first point, by explicitly describing a biophysically reasonable delivery mechanism for the population response. With regard to the second point, we have shown how differences between the population averaged and the neuronal postsynaptic rate can be resolved when each neuron keeps a memory trace of its recent spiking behavior. But the focus here was not on training a population of spiking neurons to just emulate a single mean firing rate unit. This would amount to using a learning rule similar to the one we considered in Fig. 1b which tries to force all of the neurons to march in lock-step. Then, learning performance eventually becomes independent of population size, whereas it can be increased by using a better plasticity rule. This suggests that modeling a spiking population by a mean firing rate unit underestimates its learning capacity.

Our model does not rely on the assumption that the neurons code by firing rate and can be used in conjunction with spike timing dependent codes. But it does suggest that there should be a relationship between postsynaptic coding and plasticity. In its basic form, the learning rule for the escape noise neuron we use, just changes the probability that a postsynaptic spike train generated in response to a stimulus is produced again on a further presentation of the same stimulus. So the neuron can in principle be reinforced to learn any output code. (This is in contrast to e.g. the tempotron where the spike/no-spike code is hard-wired into the plasticity rule.) But in the present framework, the generality of the escape noise rule is compromised when plasticity is modulated by a comparison between the population response and the neuronal memory trace. Since the outcome of such a comparison depends on the code assumed in reading-out the population, the resulting plasticity rule must depend on the postsynaptic code. One should keep in mind though, that matching code and plasticity may need only minor adjustments. The essential step in going from the spike/no-spike to the spike-early/spike-late code is the adjustment of a threshold parameter in the synaptic plasticity rule, whereas no modification at all was needed for the mean firing rate code.

We have not considered how the postsynaptic code is read out. Since this entails monitoring the population during the duration of the stimulus, a neural integrator is needed which is likely to involve a combination of cellular mechanisms (e.g. plateau potentials) and recurrent network connectivity [14]. In addition, to read a spike timing dependent output code, a way to measure the time elapsed since stimulus onset is needed. While this can be seen as a special case of a neural integrator, dedicated implementations of such neural clocks are also possible [15,16]. A detailed model of a decision making circuitry has been presented in [17]. There a neural integrator is formed by a group of pyramidal neurons with recurrent excitatory connectivity mediated by AMPA and NMDA receptors. For assessing the accumulated activity difference between

two input populations, two such integrators were used, each receiving input from one of the populations. Since a pool of interneurons provides mutual inhibition between the integrators, the circuitry amplifies the difference between the accumulated population signals, leading to a binary decision.

This circuitry readily specializes to a read-out for our model when decisions are encoded in the firing rates of a single population. For this, just assume that the population projects to one of the neural integrators whereas the second integrator receives input with a constant, stimulus-independent, firing rate providing the threshold. But our rule could also be used to train the two input populations driving the decision circuitry in [17]. Then we would no longer needs to assume that the duration of stimuli is known at the level of the population read-out. Instead, the duration itself could be learned. The reason is that with two populations the downstream decision making circuitry will not generate a binary decision as long as the difference in the accumulated population activities is small. So the system knows that it has to wait for further evidence until the excess accumulated activity of one population is large enough to trigger the corresponding decision. Further, the decision making could be based on a leaky accumulation of the populations activities. Then the read-out does not have to be reset at stimulus onset, so not even the onset need not be known prior to learning.

We believe that such reinforcement learning with competing input populations may become an important avenue for research since one can consider a behaviorally very natural framework: embedded at possibly unknown times within a continuous stream of events, initially unknown subsequences (stimuli) appear and a response results in reward if it is timely as well as appropriate. An instance of this framework is the learning of reaction times when detecting coherent motion within a random motion field [18]. An analysis of behavioral data [19] indicates that performance improvements in this task largely result from an increase in the evidence rate. So the input into the decision making circuitry becomes more reliable, as it would in our population learning. In contrast, plasticity in the decision making circuitry itself was found to play a smaller role. While the onset of stimuli was obvious in the random motion field experiments, this need not always be the case. For instance, temporal segmentation is crucial in the processing of complex auditory stimuli such a speech [20,21] and it would be fascinating, if rather ambitious, to computationally model how infants learn to parse sequences of syllables into words.

4 Methods

4.1 Single neuron model

The input to the model neuron is a spike pattern X consisting of M spike trains X_i ($i = 1, \ldots, M$) where each X_i is a list of the spike times in afferent i. The resulting postsynaptic spike train Y is also a list of spike times. If the neuron, with synaptic vector w, produces the output Y in response to X its membrane potential at a time t is:

$$u(t) = U_{rest} + \sum_{i=1}^{M} w_i \sum_{s \in X_i} \epsilon(t-s) - \sum_{s \in Y} \kappa(t-s)\,.$$

Here $U_{rest} = -1$ (arbitrary units) is the resting potential, $\epsilon(t)$ is the postsynaptic and $\kappa(t)$ the reset kernel. For $t \leq 0$ the kernels vanish and for $t > 0$ they are given by

$$\epsilon(t) = \frac{1}{\tau_{\mathrm{m}} - \tau_{\mathrm{s}}} \left(e^{-t/\tau_{\mathrm{m}}} - e^{-t/\tau_{\mathrm{s}}} \right) \quad \text{and} \quad \kappa(t) = \frac{1}{\tau_{\mathrm{m}}} e^{-t/\tau_{\mathrm{m}}}\,,$$

where $\tau_{\mathrm{m}} = 10\,\mathrm{ms}$ is used for the membrane time constant and $\tau_{\mathrm{s}} = 1.4\,\mathrm{ms}$ for the synaptic time constant.

The emission of postsynaptic spikes is controlled by a stochastic firing intensity $\phi(u)$ which increases with the membrane potential: At each point t in time the firing probability is $\phi(u(t))\,\Delta t$ where Δt represents an infinitesimal time window (we use $\Delta t = 0.2\,\mathrm{ms}$ in the simulations). Our stochastic intensity is

$$\phi(u) = k e^{\beta u},$$

with $k = 0.01$ and $\beta = 5$; in the limit of $\beta \to \infty$ one would recover the deterministic model with a spiking threshold $\theta = 0$. As shown in [11] the log-likelihood of actually producing, upto some time t, the output spike train Y_t is given by

$$\mathcal{L}_w(Y_t|X) = \sum_{s \in Y_t} \log \phi(u(s)) - \int_0^t \phi(u(s))\mathrm{d}s\,.$$

From this the basic quantity for computing the learning updates is obtained as

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial}{\partial w_i}\mathcal{L}_w(Y_t|X) = \begin{cases} -\phi(u(t))\,\beta\,\mathrm{PSP}_i(t) & \text{if } t \notin Y \\ \delta(t)\,\beta\,\mathrm{PSP}_i(t) & \text{if } t \in Y \end{cases}$$

Here $\mathrm{PSP}_i(t) = \sum_{s \in X_i} \epsilon(t-s)$ is the contribution of the postsynaptic potential of synapse i and $\delta(t)$ is Dirac's delta function.

In all the simulations initial values for the synaptic strength were picked from a Gaussian distribution with mean and standard deviation equal to 1.7, independently for each afferent and each neuron.

4.2 Pattern statistics

Input patterns are made up of 50 independent Poisson spike trains with a mean firing rate of 6 Hz, independent realizations are used for each pattern. An input layer with 50 nodes presents patterns to the neuronal population. The connectivity from the input layer to the neurons is random, with each neuron receiving a connection from an input node with a probability of 0.8. So the input X^ν effectively seen by the ν-th neuron consists of roughly 40 parallel spike trains, and inputs to different neurons are different but highly correlated. Except for the simulations with stimuli of variable length, the duration of stimuli is 500 ms. The presentation order of the stimuli-response pairs was random.

4.3 Episodic learning

Simulations. With just global reinforcement (Eq. 2) the learning rate was $\eta = 1250/N$. Decreasing the learning rate with increasing population size was essential to compensate for the increasingly loose relationship between single neuron performance and reward. We also tested higher learning rates by using the scaling $\eta = 1250/\sqrt{N}$. For $N > 1$, this results in a significant deterioration in performance both on the population and the single neuron level. For learning with individual reward (Eq. 3) the learning rate was $\eta = 625$, and $\eta = 2500$ was used for attenuated learning (Eq. 5).

Gradient property. The update (5) can be understood as a stochastic gradient step. A detailed derivation is given in the Supplementary Information, here we just sketch the main ideas. Instead of optimizing expected reward (R), we optimize the expected value of $g(R|S|)$, where $|S|$ denotes the absolute value. The function g is increasing but it saturates for large positive values, leading to learning attenuation. Using this objective function in conjunction with the standard gradient estimator [22], does not yield an update with individualized reward. To achieve this, we analytically average the standard estimator over the two outcomes $c(Y^\nu) = \pm 1$, instead of leaving this to the sampling procedure. Hence compared to the standard estimator our update has reduced variance and this speeds up the learning.

4.4 On-line learning

Modeling of the feedback signals. For the reward feedback, we assume a temporary increase in the release rate of a neurotransmitter (e.g. dopamine) in case of success, whereas failure results in a decrease. Assuming that the stimulus X ends at time T, a simple model for the concentration c_{rew} of the neurotransmitter is

$$\tau_{\text{rew}}\dot{c}_{\text{rew}} = -c_{\text{rew}} + 1 + R\,\Theta(t; T + \Delta_{\text{rew}}, L_{\text{rew}})\,.$$

Here $\Theta(t; T+\Delta_{\text{rew}}, L_{\text{rew}})$ is equal to 1 for t between $T+\Delta_{\text{rew}}$ and $T+\Delta_{\text{rew}}+L_{\text{rew}}$ and is zero otherwise. So L_{rew} gives the time during which the release rate is changed, and the parameter Δ_{rew} allows to model delayed onset of the reward. A typical time course of c_{rew} is shown in Fig. 3. The above describes the time course of c_{rew} upto the time $T' > T$ when the stimulus X' immediately following X ends. Then the reward variable R is replaced by the value appropriate for the response to stimulus X'.

Feedback about the population output is modeled similarly. In the time interval from T to T' the concentration c_{pop} of the corresponding neurotransmitter evolves as

$$\tau_{\text{pop}}\dot{c}_{\text{pop}} = -c_{\text{pop}} + \beta + \alpha\,\text{sign}(S)e^{-S^2}\Theta(t; T, L_{\text{pop}})$$

where we assume that the change in release rate is modulated by the strength of the population signal via the e^{-S^2} term. Further, the onset of population feedback occurs immediately when the stimulus ends ($\Delta_{\text{pop}} = 0$).

The explicit formulas for the deviations of the concentrations from their homeostatic levels (used above in Eq. 7) are simply $c^*_{\text{rew}} = c_{\text{rew}} - 1$ and $c^*_{\text{pop}} = c_{\text{pop}} - \beta\,$.

Magnitude of the population feedback. For episodic attenuated learning the magnitude of the weight vector change given by Eq. 5 is essentially the same for $R = 1$ and $R = -1$ if S is close to zero. For binary decision tasks this is reasonable since success and error yield the same amount of information about the desired output, and a small value of $|S|$ means that population performance is unreliable. To achieve the same effect in the on-line procedure we have to consider the magnitude of $\gamma(t)$ in Eq. 8. For $R = 1$ this depends on the magnitude of the population feedback which is controlled by the parameter α. For the timing parameters of the feedback signals we use, the value of the integral of $\gamma(t)$ over a reward period is approximately the same for $R = 1$ as for $R = -1$ if $\alpha = 2.5$ (and $|S|$ is small). Hence, we always use $\alpha = 2.5$ for binary decision problems. In contrast, success is much more informative than failure in the four-way decision task, since in the case of an error any of the

remaining three output values can be the correct one. So for small $|S|$ a larger learning step is appropriate if $R = 1$ and, hence, we use $\alpha = 5$ in the four-way task.

Read-out of the neuronal memory traces. The threshold value in Eq. 7 is $\theta = e^{-1.1}$ in all simulations but the ones using the spike-early/spike-late code. The rationale behind this choice is the following. If the time constant τ_{M} is equal to stimulus duration, then a spike occurring in response to a stimulus yields a memory trace $s^\nu \geq 1/e$ when the stimulus ends. Hence, choosing $\theta = 1/e$ gives $\mathrm{sign}(s^\nu - \theta) = c(Y^\nu)$ for the spike/no-spike code at the time T when the stimulus ends. So, at this point in time, the read-out of the memory trace accurately reflects what the neuron was doing. But, due to the latency of c_{rew}, the strongest synaptic changes occur around $T + 50$ (Fig. 3b,c) even if reward onset is instantaneous and so a θ slightly smaller than $1/e$ is used in simulations.

For the spike-early/spike-late code $\theta = e^{-0.55}$ since the scoring function now distinguishes between spiking during the first and the second half of stimulus duration. In addition Eq. 7 is now used only if $s^\nu \geq \theta^2$, otherwise, i.e. when s^ν indicates that the neuron has not fired for a long time, we set $r^\nu = 0$. This reflects the fact that the spike-early/spike-late scoring function has the value of zero, $c(Y^\nu) = 0$, if the stimulus elicited no postsynaptic spike. (The cases where $c(Y^\nu) = 0$ because the same number of spikes are produced during the first and second half, are approximately accounted for by the stochastic update already described in main text.)

Learning rate and performance measure A learning rate of $\eta = 8$ was used in all of the simulations of the on-line procedure (Eq. 8), except for the ones using the spike-early/spike-late code. There $\eta = 2$ was used. The performance percentages shown in the learning curves are computed as a running mean $\bar{p}$ which is updated after each pattern presentation by $\bar{p} \leftarrow (1-\lambda)\bar{p} + \lambda p$. Here $p = 100\%$ if the presented stimulus was classified correctly, otherwise $p = 0$. The timing parameter was set to $\lambda = 0.2/P$, where P is the total number of patterns to be learned.

References

[1] A. Pouget, P. Dayan, and R. Zemel. Information processing with population codes. *Nature Rev. Neurosci.*, 1:125–132, 2000.

[2] B. Averbeck, P. Latham, and A. Pouget. Neural correlations, population coding and computation. *Nature Rev. Neurosci.*, 7:358–366, 2006.

[3] P. Dayan and L. Abbott. *Theoretical Neuroscience*. The MIT Press, 2001.

[4] H. Seung. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, 40:1063–1073, 2003.

[5] I. Fiete and H. Seung. Gradient learning in spiking neural networks by dynamic perturbation of conductances. *Phys. Rev. Letts.*, 97:048104, 2006.

[6] R. Florian. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural Computation*, 19:1468–1502, 2007.

[7] E. Izhikevich. Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cerebral Cortex*, 17:2443–2452, 2007.

[8] S. Wirth, M. Yanike, L. Frank, A. Smith, W. Brown, and W. Suzuki. Single neurons in the monkey hippocampus and learning of new associations. *Science*, 300(5625):1578–1581, 2003.

[9] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing. Vol. I.* MIT Press, 1986.

[10] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City etc., 1991.

[11] J. Pfister, T. Toyoizumi, D. Barber, and W. Gerstner. Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural Computation*, 18:1318–1348, 2006.

[12] R. Gütig and H. Sompolinsky. The tempotron: a neuron that learns spike timing-based decision. *Nature Neuroscience*, 9:420–428, 2006.

[13] S. Fusi, W.F. Asaad, E.K. Miller, and X-J. Wang. A neural circuit model of flexible sensori-motor mapping: learning and forgetting on multiple timescales. *Neuron*, 54:319–333, 2007.

[14] G. Major and D. Tank. Persistent neural activity: prevalence and mechanisms. *Current Opinion in Neurobiology*, 14:675–684, 2004.

[15] D. Durstewitz. Neural representation of interval time. *Neuroreport*, 15:745–749, 2004.

[16] J. Reutimann, V. Yakovlev, S. Fusi, and W. Senn. Climbing neuronal activity as an event-based cortical representation of time. *J. Neuroscience*, 24(13):3295–330, 2004.

[17] X. Wang. Probabilistic decision making by slow reverberation in cortical circuits. *Neuron*, 36:955–968, 2002.

[18] J. Gold and M. Shadlen. Representation of a perceptual decision in developing oculomotor commands. *Nature*, 404:390–394, 2000.

[19] P. Eckhoff, P. Holmes, C. Law, P. Connolly, and J. Gold. On diffusion processes with variable drift rates as models for decision making during learnings. *New Journal of Physics*, 10:0150060, 2008.

[20] L. Sanders, E. Newport, and H. Neville. Segmenting nonsense: an event related potential index of perceived onsets in continuous speech. *Nat. Neurosci.*, 5:700–703, 2002.

[21] K. McNealy, J. Mazziotta, and M. Dapretto. Cracking the language code: Neural mechanisms underlying speech parsing. *J. Neurosci.*, 26:7629–7639, 2006.

[22] R. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.