Supporting Information

Supervised task learning via stimulation-induced plasticity in rate-based neural networks

Francesco Borra[1], **Simona Cocco**[1], **and Rémi Monasson**[1]

[1]Laboratoire de Physique de l'Ecole Normale Supérieure, PSL Research, CNRS UMR8023, Sorbonne Université, 24 rue Lhomond, 75005 Paris, France

1 Model parameters: values and biological significance

1.1 Values

Task 1. We use $c_E = 0.2$ and $c_I = 0.5$. The parameters are $J_0 = 0.015$, $\theta(E) = 0.08\, r_{max}$, $\theta(I) = 0.12\, r_{max}$, $\eta(E) = 1$, $\eta(I) = -1.2$, and $\beta_1 = 0.8$, $\beta_2 = 0.6\,(r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.8\, r_{max}$ and $\bar{J} = 0.25$, $f_{max} = -f_{min} = 0.2\, r_{max}$. Note that, in this setting, the associative behaviour of excitatory synapses is triggered for lower values of activity ($\theta(E) < \theta(I)$); however, inhibitory potentiation is favored by $|\eta(I)| > \eta(E)$. For the sake of inferring J, we use $n_{probe} = 10$ plus the four input stimulations related to the task. The control stimulation lasts $\Delta t = 0.003\, \tau_s$. Probing for estimating connectivty is done by choosing $f_i = 0, f_{max}$ with equal probabilities.

Task 2. Parameters are chosen as follows: $\theta(E) = 0.2\, r_{max}$, $\theta(I) = 0.16\, r_{max}$, $\eta(E) = 1$, $\eta(I) = 1$, and $\beta_1 = 0.8$, $\beta_2 = 1.25\,(r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$ and $\bar{J} = 0.1$, $f_{max} = -f_{min} = 0.4\, r_{max}$. We use $n_{probe} = 10$ and $\Delta t = 0.003\, \tau_s$. Probing for estimating connectivity is done with f_i distributed uniformly in $[f_{max}/2, f_{max}]$ for excitatory neurons and $[0, f_{max}/2]$ for inhibitory neurons; further inforamtion is available in SI, Section 5.2.

The interaction strengths J and modulations K entering the interaction kernels, see Eqs. (21)(22) in the main text, are chosen to be $J_{E,E} = J_{I,E+} = -J_{E,I} = -J_{I,I} = 0.1$, and $K_{E,E}^{-1/2} = 0.08$, $K_{I,E}^{-1/2} = 0.05$, $K_{E,I}^{-1/2} = 0.15$, $K_{I,I}^{-1/2} = 0.1$. These parameter values are compatible with the existence of two self-sustained bumps of activity, localized and diametrically opposed on the two rings.

In the cost U in Eq. (3), the weight factors are chosen as $w_{\varepsilon_i\varepsilon_j} = \left(4 \sum_{ab|\varepsilon_a=\varepsilon_i,\varepsilon_b=\varepsilon_j} [J_{ab}^{target}]^2\right)^{-1}$ to ensure that the cost assesses the average relative errors over the four classes of connections $E/I \to E/I$.

1.2 Current-to-firing rate activation function

We discuss here the expression for the activation function Φ used in Eqs. (1) and (8) in the main text. Three regimes must be distinguished.

From input currents to firing rates. Consider the linear integrate-and fire model of a spiking neuron. In the absence of leakage (zero conductance), the membrane potential V obeys the following dynamical equation

$$C\dot{V} = I, \tag{1}$$

where I is the input current (considered to be constant over time) and C the capacitance. The neuron fires with rate

$$r(I) = MF \times I \quad \text{with} \quad MF = \frac{1}{C(V_{threshold} - V_{rest})}\,, \tag{2}$$

and V_{rest} and $V_{threshold}$ are, respectively, the rest and instability potentials. Equation (2) shows that input currents can be expressed in units of firing rates through the rescaling by the multiplicative factor MF: $I \to \tilde{I} = MF \times I$. This is the convention we adopt throughout our paper.

An order of magnitude for MF is given by the slope of the current-to-frequency relationship for human pyramidal cells[1], with the result $MF \approx 50$ Hz/nA. Compatible values are reported for pyramidal cells in the primary visual cortex of cats[2], with $MF \approx 100 - 500$ Hz/nA.

High activity regime. The maximal firing rate r_{max} is the inverse of the refractory period, with values of the order of a few hundreds Hz. The linear current-rate relation in Eq. (2) can be simply modified as

$$r(\tilde{I}) = \frac{r_{max}\,\tilde{I}}{r_{max}+\tilde{I}} \qquad (\text{ large } \tilde{I}\,)\,. \tag{3}$$

This expression fixes both the maximum firing rate and the way r reaches it for large currents $\tilde{I}$. Many effects such as fatigue, temporal fluctuations (in the concentration of neurotransmitters), or short term plasticity are not taken into account.

Low activity regime. For low external input $\tilde{I}$, noisy synaptic effects can become predominant and set a baseline firing activity r_{low} for the neuron. We can interpolate between the linear response in Eq. (2) and this low current, baseline activity regime through the introduction of a smooth function,

$$r(\tilde{I}) = r_0\,\log\left(1+\exp(\tilde{I}/r_0)\right) \qquad (\text{ small } \tilde{I}\,)\,. \tag{4}$$

where $r_0 = r_{low}/\log 2$. Note that r_0 also defines the scale at which $\tilde{I}$ can be considered small. We expect r_0 to range between 0.1 and 1Hz[1]. The ratio between the maximal and baseline firing rates is thus of the order of $r_{max}/r_0 = 50-2000$. In this paper, we use $r_{max}/r_0 = 250-500$.

Complete activation function. The activation function described in Eq. (8) of Methods, main text, interpolates between the low, intermediate, and high activity regimes corresponding, respectively, to Eqs. (4), (2), (3).

1.3 Synaptic connections

A pre-synaptic neuron, depending on its type, can give rise to small excitatory or inhibitory post-synaptic potentials, referred to as EPSP or IPSP, each time it emits a spike. The mean value of the resulting current incoming onto the post-synaptic neuron is, after rescaling by MF, equal to

$$\tilde{I} = \frac{\delta V}{V_{threshold}-V_{reset}}\,r_{pre}\,, \tag{5}$$

where δV is the value of the EPSP or IPSP, and r_{pre} is the firing rate of the pre-synaptic neuron. We deduce the dimensionless expression of the synaptic connection, compare with Eq. (1) in the main text,

$$J = \frac{\delta V}{V_{threshold}-V_{reset}}\,. \tag{6}$$

Typical values for $|\delta V|$ range from tens to hundreds μV[3]. As $V_{threshold}-V_{rest}$ is about 20 mV, we find J to be of the order of 5.10^{-4} to 5.10^{-2}. These values are consistently lower than the soft bounds $\bar{J}=0.25$ in Task 1 and $\bar{J}=0.1$ in Task 2.

1.4 Control stimulation

Amplitude. The second source of input in our model is the external control f, see Eq. (1) in the main text. Literature reports accurate control of the activity of genetically-modified neurons through light activation, with firing rates up to 30 Hz with Channelrhodopsin-2[4] and up to 100 Hz with faster Chrimson variants[5]. The firing rates can be controlled both through the frequency of light pulses and the illumination power. These values are compatible with ratios f_{max}/r_{max} reaching a few tenths, as considered in this work. Inhibitory stimulations can also be obtained with optogenetic tools, resulting in strong hyperpolarization of cells by more than 10 mV, and strong reduction (50%) in the firing activity[6].

Duration. Due to the choice of the parameters and the values of firing rates, all terms on the right hand side of Eq. (2) in the main text take values of the order of unity. The order of magnitude of the time required for a variation δJ in a synaptic connection is therefore

$$\delta t \simeq \tau_s\,\delta J\,. \tag{7}$$

Substituting δJ with $\bar{J}/10$ in the above equation, where $\bar{J}$ is the soft bound on the (absolute) value of connections, we find that a significant change to connection requires time $\delta t \simeq \tau_s \times \bar{J}/10 \simeq 0.01\,\tau_s$.

Hebbian plasticity is associated to a variety of times scales δt, ranging from few seconds to few minutes, with homeostatic effects taking place over a large interval of time scales too[7]. Choosing δt of the order of 1 minute gives τ_s in Eq. (2) of the order of a few hours. The duration of one cycle of control stimulation is then approximately equal to $\Delta t = 0.003\,\tau_s \simeq 30$ sec. This time scale is consistently larger than the time τ_r for convergence of the network activity, see next Section and Discussion in the main text.

2 Stationary activity of the rate model

2.1 Convergence of dynamics

The stationary states of activity are the roots of the N coupled implicit equations

$$\mathbf{r} = \Phi\big(\mathbf{J}\cdot\mathbf{r}+\mathbf{f}\big)\ . \tag{8}$$

Solutions, in general, are not unique. Moreover, since the external input $\mathbf{f}$ is arbitrary, uniqueness may hold for some $\mathbf{f}$ and not for others. However, if the largest singular value of $\mathbf{J}$, $\sigma(\mathbf{J})$, is smaller than unity, the solution $\mathbf{r}^\star$ of (8) is guaranteed to be unique[1]. Moreover, $\mathbf{r}^\star$ is a globally attractive fixed point of the dynamical equations (expressing time in units of τ_n to lighten notations):

$$\dot{\mathbf{r}} = -\mathbf{r}+\Phi(\mathbf{J}\cdot\mathbf{r}+\mathbf{f}). \tag{9}$$

and, therefore, no limit cycle can exist.

Uniqueness of solution. Consider the discrete-time dynamics

$$\mathbf{r}(k+1) = \Phi(\mathbf{J}\cdot\mathbf{r}(k)+\mathbf{f})\ . \tag{10}$$

As $\sigma(\mathbf{J}) = \max_{\mathbf{x}}\|J\,\mathbf{x}\|/\|\mathbf{x}\| < 1$, the map $L_{J,f} : r \mapsto \mathbf{J}\cdot\mathbf{r}+\mathbf{f}$ is a global contraction for any $\mathbf{f}$. Moreover, since $|\Phi'| < 1$, Φ is a global contraction, too. Hence, the map in (10), which can be written as $\Phi\circ L_{J,\mathbf{f}}$, is a global contraction. According to the contraction mapping theorem, Eq. (10) has a unique fixed point given by $\mathbf{r}^\star = \lim_{k\to\infty}\mathbf{r}(k)$ for any $\mathbf{r}(0)$. This result gives us a practical and fast way to compute $\mathbf{r}^\star$.

Attractivity of solution. We now show that the unique solution of the stationary solution is an attractive fixed point of the continuous-time dynamics. Assume $\mathbf{r}\neq\mathbf{r}^\star$ and note that, under dynamics (9),

$$\frac{1}{2}\frac{d}{dt}\|\mathbf{r}-\mathbf{\Phi}\|^2 = (\mathbf{r}-\mathbf{\Phi})^T\cdot\mathbf{M}(\mathbf{r})\cdot(\mathbf{r}-\mathbf{\Phi}) - \|\mathbf{r}-\mathbf{\Phi}\|^2 \tag{11}$$

where we have used the shorthand notation $\mathbf{\Phi} = \Phi(\mathbf{J}\cdot\mathbf{r}+\mathbf{f})$ and $M_{ij}(\mathbf{r}) = \Phi_i'(\mathbf{J}\cdot\mathbf{r}+\mathbf{f})\,J_{ij}$. Note that since $\Phi_i' < 1$ for all i, the element-wise multiplication by Φ_i' is a contraction. Then, if $\sigma(\mathbf{J}) < 1$, $\|\mathbf{J}(\mathbf{r})\cdot(\mathbf{r}-\mathbf{\Phi})\| < \|\mathbf{r}-\mathbf{\Phi}\|$, from which it follows that

$$\frac{d}{dt}\|\mathbf{r}-\mathbf{\Phi}\|^2 < 0\ \ \forall\mathbf{r}\neq\mathbf{r}^\star. \tag{12}$$

We conclude that $L(\mathbf{r}) = \|\mathbf{r}-\mathbf{\Phi}\|^2$ is a global Lyapunov function for (9) for any $\mathbf{f}$ in the sense that it satisfies Lyapunov's second method for stability; note that $L(\mathbf{r}^\star) = 0$. Hence, whatever the initial condition, $\mathbf{r}$ converges asymptotically to $\mathbf{r}^\star$ under the dynamics (9).

In practice, to find the fixed point, we iterate (10). Note that $\sigma(\mathbf{J}) < 1$ is a sufficient but not a necessary condition. Violating it does not always appear to be too detrimental to convergence. While we softly enforce this condition in the classification Task 1, we violate it on purpose in the case of Task 2, since a continuous attractor is expected to have multiple solutions by construction.

2.2 Relaxation time

We can also estimate how fast the firing rates converge towards the stationary solution. Linearizing the dynamics in Eq. (9) to the first order in $\delta\mathbf{r} = \mathbf{r}-\mathbf{r}^\star$, we obtain

$$\frac{d}{dt}\delta\mathbf{r} = (-\mathbf{Id}+\mathbf{M}(\mathbf{r}^\star))\cdot\delta\mathbf{r}\ . \tag{13}$$

Hence, the relaxation time τ_r is given by

$$\tau_r = \frac{\tau_n}{1-\max\{\text{Real part of eigenvalues of }\mathbf{M}(\mathbf{r}^\star)\}}\ . \tag{14}$$

The relaxation time τ_r includes network effects, and is expected to be larger than the single-neuron time scale τ_n. Note that τ_r does not depend only on the connectivity $\mathbf{J}$, but also on the input $\mathbf{f}$. We show how the ratio τ_r/τ_n varies during training in Fig. 1, both for Tasks 1 and 2. We observe that the introduction of a regularization over the connections for Task 1 consistently reduces the relaxation time. However, even in the absence of this extra cost as in Task 2, the value of τ_r/τ_n remains moderate up to the end of the training phase.

[1] $\sigma(\mathbf{J})$ coincides with the spectral radius of $\sqrt{\mathbf{J}^T\mathbf{J}}$

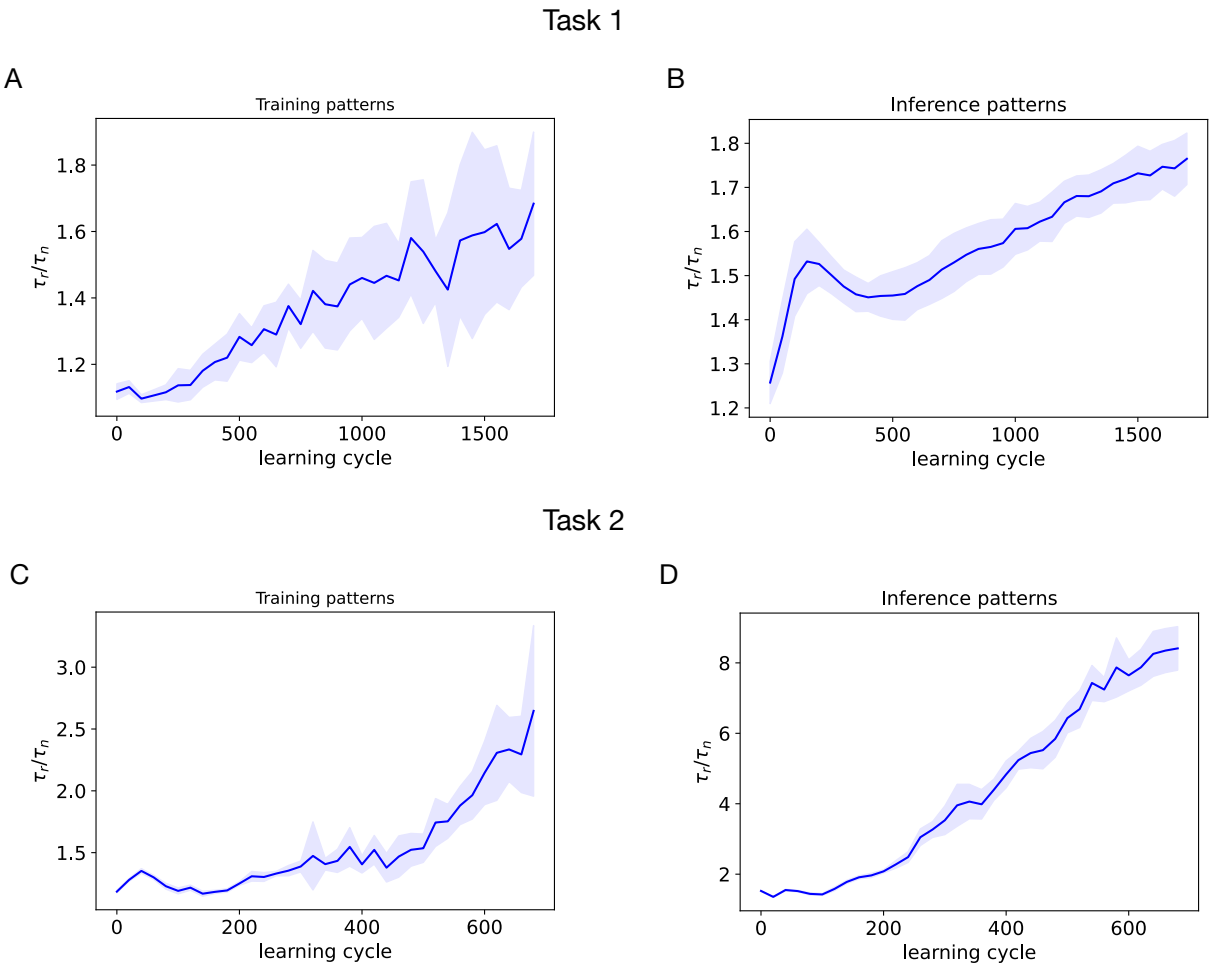

Figure 1. Relaxation times τ_r/τ_n for Task 1 (top row, A and B), and Task 2 (bottom row, C and D). In the left column we show results for the training patterns, while the outcome for random probing patterns (with random entries 0 or 1) used for connectivity estimation are shown in the right column. Results are averaged over 10 instances for training examples, and over 20 examples in the case of the inference patterns; the blue shaded area corresponds to one standard deviation from the mean.

3 Optimization loop over the control

The control $\mathbf{f}$ can be found with a training loop that minimizes the auxiliary cost

$$W(\mathbf{f}) = \frac{\Delta U(\mathbf{f})}{\Delta t} + \gamma \|\Delta \mathbf{J}(\mathbf{f})\|^2 \tag{15}$$

with

$$\Delta U(\mathbf{f}) = U(\mathbf{J} + \Delta \mathbf{J}(\mathbf{f})) - U(\mathbf{J}) \tag{16}$$

and $\gamma = 1/(\bar{J}/10)^2/cN$. Exit from the minimization loop takes place if W is sufficiently low, *i.e.* below a hyper-parameter $W^{threshold}$, which is negative at the beginning to make sure that the algorithm does not get stuck immediately, but is then set to 0. This condition is checked once W is not changing anymore, *i.e.* either the gradient of W is zero or the running average of the differences ΔW of W between two successive loop iterations is small than a threshold $\Delta W^{threshold}$ (defining another hyper-parameter). If either of these conditions is verified, the loop is broken if $\Delta U < \Delta U^{threshold}$ and re-initialized with a new initial condition for f otherwise. A pseudocode for the training loop to find $\mathbf{f}$ at a given time step is provided below. A number of hyper-parameters and checks are involved in this optimization, which depend on the setting and can be found in the code.

Initialization:

Take $\mathbf{f}_{old}$ from previous control stimulation (if there is one) and a random new one $\mathbf{f}_{random}$. Check which one has lowest scalar product with $\partial \Delta U / \partial \mathbf{f}$ and pick this as initial condition for the loop. This step aims at preventing the control to get stuck in local minima.

Loop:

1. compute $\dfrac{dW}{d\mathbf{f}}$.
2. If $\dfrac{dW}{d\mathbf{f}} = 0$ or $\langle \Delta W \rangle < \Delta W^{threshold}$ then
   - if $\Delta U < \Delta U_{threshold}$ then exit
   - else reset. Randomly draw a new $\mathbf{f}$ and reset the running average $\langle \Delta W \rangle$ to a high value.
3. else update the control as follows

$$\mathbf{f} \leftarrow \mathbf{f} - \alpha \frac{dW}{d\mathbf{f}} \tag{17}$$

   where the learning rate α is chosen to minimize ΔW by scanning multiple values; this step is necessary because the optimum learning rate significantly changes with the connectivity $\mathbf{J}$. Clip element-wise $\mathbf{f}$ in the range $[f_{min} - q, f_{max} + q]$ for a small value of q (introduced to improve the numerical procedure).
4. compute W and ΔU and update the running average $\langle \Delta W \rangle$.

The variation of the control variable $\mathbf{f}$ between two successive time steps is often small. However, large discontinuities may be occasionally present, in particular at the end of the protocol, see Task 1 and Fig. 8 in the main text. The fact that a discontinuous control can emerge in dynamical problems without discontinuities is well known, for in stance in optimal control[8]. These discontinuities may have two origins. The first is an external reset, see step 2 in the loop above. If the control variable is stuck in a minimum with positive $\Delta U(\mathbf{f})$ (or higher than the threshold), the gradient descent is reset with a different initial condition, see Fig. 2A. This may also happen if, before the gradient descent starts, if the previous control $\mathbf{f}$ is worse than a randomly drawn new one. The second possibility is that the gradient descent of W finds a path to access a new distant solution, which was previously inaccessible, see Fig. 2B.

The choice of hyper-parameters is the following: $q = 1/10\, f_{max}$, $n_{av} = 10$ (size of the sliding window for running average), α is chosen from $\alpha = 10^{-s} \left\| \frac{dW}{d\mathbf{f}} \right\|^{-1}$ with $s \in \{-1 + k/2 | k = 0, 1, ..., 10\}$. A few hyper-parameters are task-specific. Task 1: $\Delta U^{threshold} = 0$; $\Delta W^{threshold} = (\Delta U^{threshold} - W)/500$ if $\Delta U^{threshold} > W$, else $\Delta W^{threshold} = (\Delta U^{threshold} - W)/(500 + 50 \min(n_{resets}, 15) - k)$ where k is the iteration number after the last reset (or beginning of the loop) and n_{reset} is the number of resets; when resetting $\mathbf{f}$, components f_i are drawn independently and uniformly between $[f_{min}, f_{max}]$. Task 2: $U^{threshold} = 0.001/\Delta t$ if $U > 0.7$ and 0 otherwise and $\Delta W^{threshold}$ as in Task 1; when resetting $\mathbf{f}$, components f_i are drawn independently and uniformly between $[0, f_{max}]$. Additional details can be found in the code.

4 Further information on Task 1

4.1 Effect of parameters

Depending on the choice of parameters and hyper-parameters the training process can change from fast and easy to very hard or even impossible. We consider in particular the role of the size of the network, the control strength range and the choice of parameter in the learning rule. We stress that the hyper-parameters (including the time after which the training was interrupted) were optimized for the setting reported in the main text and were not changed as we varied these parameters.

In Fig. 3A, we study the dependency of the training curves with the size N of the network for $f_{max} = 0.2\, r_{max}$ as in the main text. We observe that learning becomes harder (longer) and shows increased variability for smaller networks. Note the note the non monotonic behaviour in N, which should, however, be interpreted with caution since the parameter scaling with the size should be treated more carefully.

We now consider the role of the control strength f_{max} for networks with $N = 100$ neurons. Intuitively, the larger the range, the easier the control. This is confirmed by Fig. 3B, where the training curves are shown for five values of $f_{max}/r_{max} = 0.1$, 0.15, 0.2, 0.25, 0.3; 0.2 is the value used in the main text). It is important to highlight that if the control is too weak, training seems to fail. While this minimal value of f_{max} correspond to Task 1, and is not expected to hold for other tasks[2], we expect the

[2] In particular, it is likely that the value of the minimal f_{max} could be slightly improved by optimizing over the hyper-parameters.

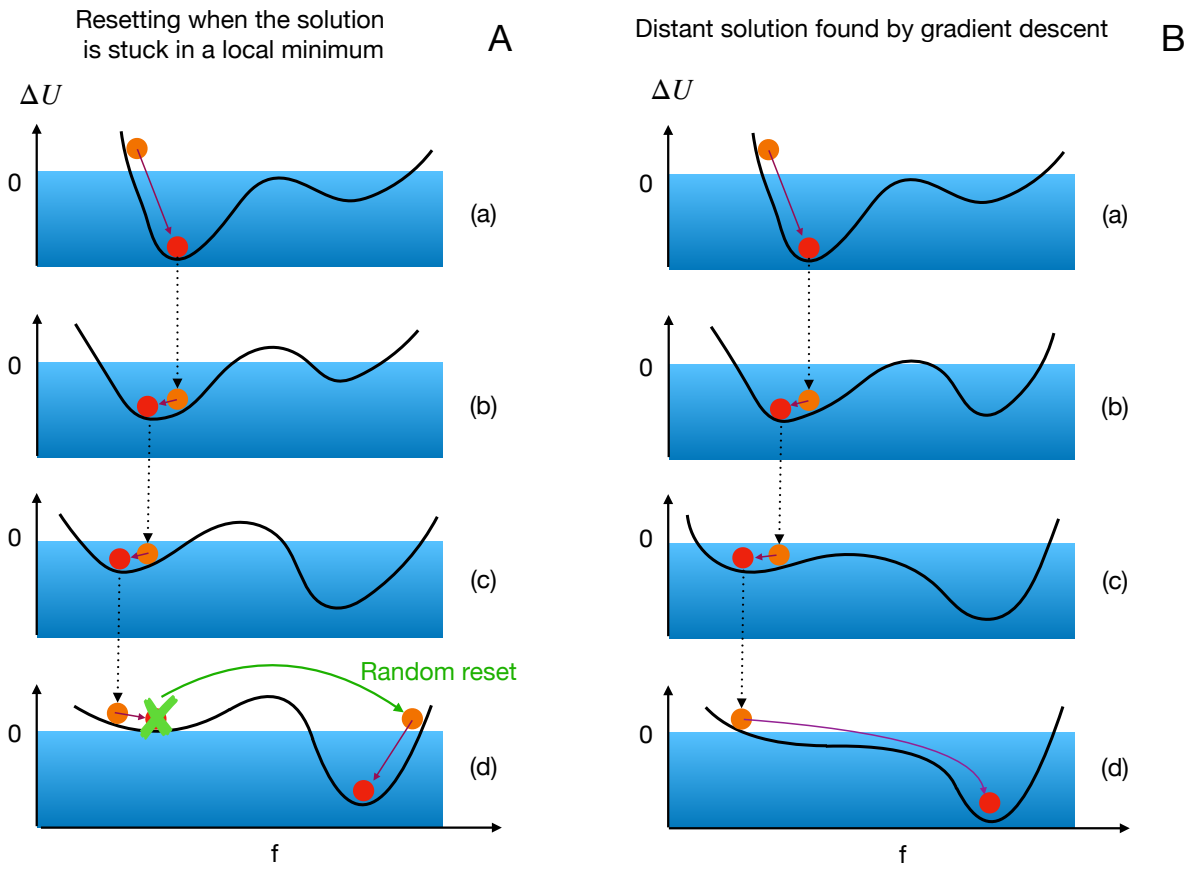

Figure 2. The two mechanisms generating large discontinuities in the control variable. Panel A: external reset. Panel B distant solution found by gradient descent. The horizontal axis symbolizes the control variable **f** on a line. The vertical axis the value of ΔU, with negative values colored in blue. The cartoon shows four successive time steps (a)-(d). At each time step, the initial condition (orange) goes to a local minimum (red) which is used as initial condition for the following iteration. However, at step (e), in panel A, the local minimum is positive so an external reset is needed. Note how two different mechanisms at time step (e) in panels A and B produce the same effect.

existence of a critical value of f_{max} below which learning through stimulation cannot be implemented to be quite generic.

In Fig. 3C, we change some parameter values, focusing on β_1,$\theta(E)$ and $\theta(I)$.

- Setting 1: $\theta(E) = 0.08\, r_{max}$, $\theta(I) = 0.12\, r_{max}$, $\beta_1 = 4$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$;
- Setting 2 (as in the main text): $\theta(E) = 0.08\, r_{max}$, $\theta(I) = 0.12\, r_{max}$, $\beta_1 = 0.8$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$;
- Setting 3: $\theta(E) = 0.1\, r_{max}$, $\theta(I) = 0.06\, r_{max}$, $\beta_1 = 4.$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$;
- Setting 4: $\theta(E) = 0.1\, r_{max}$, $\theta(I) = 0.06\, r_{max}$ $\beta_1 = 0.8$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$;
- Setting 5: $\theta(E) = 0.12\, r_{max}$, $\theta(I) = 0.1\, r_{max}$, $\beta_1 = 4$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$;
- Setting 6: $\theta(E) = 0.12\, r_{max}$, $\theta(I) = 0.1\, r_{max}$, $\beta_1 = 0.8$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$.

All the other parameters keep the same values as in the main text. In particular all results in Fig. 3C were obtained with $N = 100$ neurons.

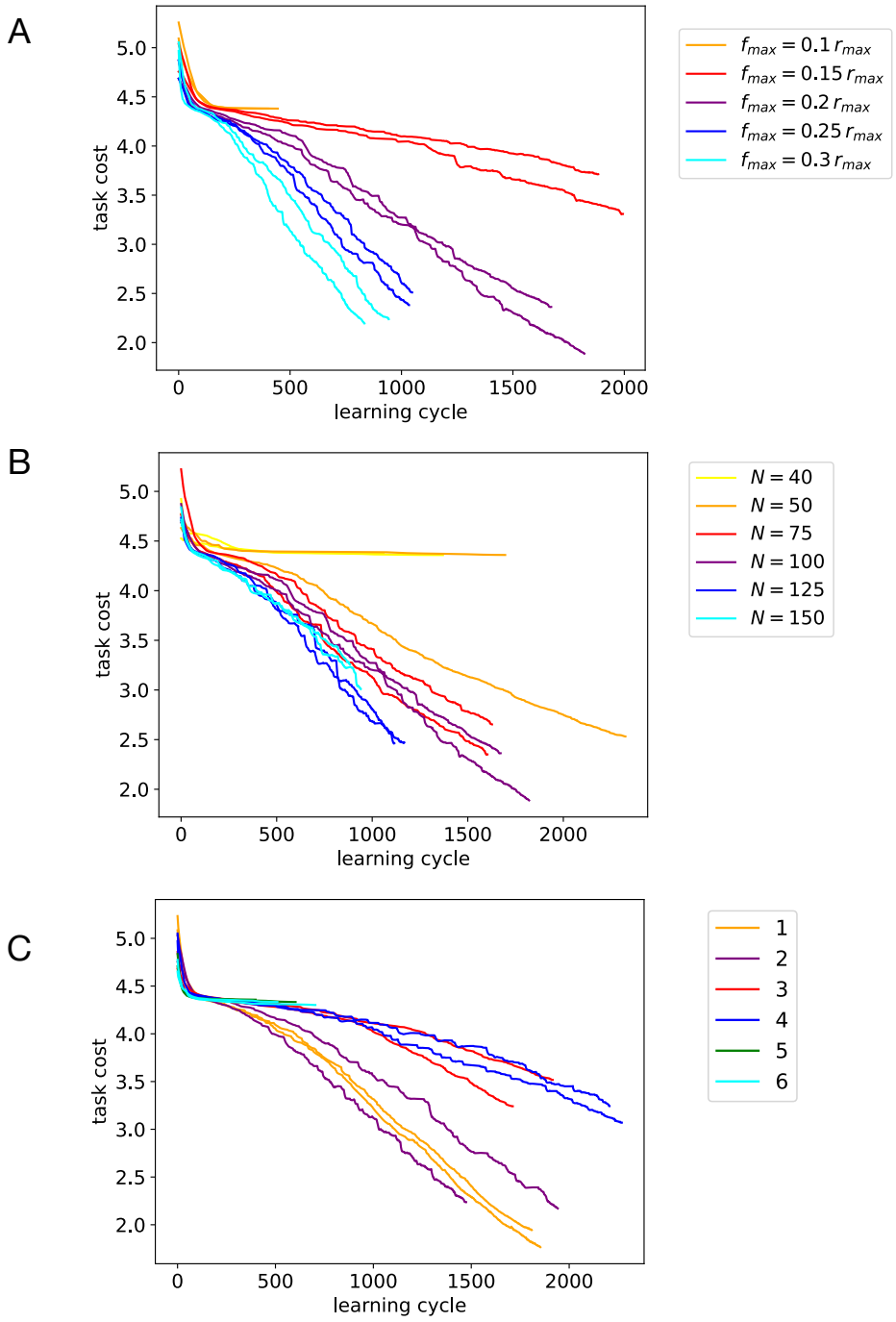

Figure 3. Training cost as a function of the learning cycle for Task 1 with parameter variations. Two examples, colored in the same way, are shown for each case. A: Changing the control maximal value f_{max}. Notice that, for $f_{max} = 0.1\, r_{max}$, training fails. B: Changing the network size N; in practice, N_{in} and N_{out} are left unchanged, while $N_{proc} = N - 30$ is varied. C: Changing other parameters, see descriptions of settings 1 to 6 in the text.

4.2 Case of Hebbian rule for inhibitory neurons

Last of all we show, in Fig. 4, that Task 1 can also be done with Hebbian learning for inhibitory synapses. The picture is not much different from the anti-Hebbian case. We show a few settings with $N = 100$. The settings are the following.

- Setting 1: $\theta(E) = 0.2 r_{max}$, $\theta(I) = 0.12 r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08\, r_{max}$, $\beta_1 = 3.125$, $\beta_2 = 0.5 \times (r_{max}/\bar{J})^2$.
- Setting 2: $\theta(E) = 0.06 r_{max}$, $\theta(I) = 0.1 r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08\, r_{max}$, $\beta_1 = 3.125$, $\beta_2 = 0.5 \times (r_{max}/\bar{J})^2$.
- Setting 3: $\theta(E) = 0.12 r_{max}$, $\theta(I) = 0.08 r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08 r_{max}$, $\beta_1 = 3.125$, $\beta_2 = 0.5 \times (r_{max}/\bar{J})^2$.
- Setting 4: $\theta(E) = 0.08 r_{max}$, $\theta(I) = 0.12 r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08\, r_{max}$,$\beta_1 = 3.125$, $\beta_2 = 0.5 \times (r_{max}/\bar{J})^2$.
- Setting 5: $\theta(E) = 0.10 r_{max}$, $\theta(I) = 0.06 r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08\, r_{max}$, $\beta_1 = 3.125$, $\beta_2 = 0.5 \times (r_{max}/\bar{J})^2$.
- Setting 6: $\theta(E) = 0.08 r_{max}$, $\theta(I) = 0.12 r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08\, r_{max}$,$\beta_1 = 3.125$, $\beta_2 = 0.5 \times (r_{max}/\bar{J})^2$.

4.3 Neural activation and input representation

If Fig. 5 we provide a picture of how the neural network represents different inputs in the *proc* region of the network and how the differentiation emerges. First we show the fraction of neuron activating (defined as being surpassing a certain activity threshold). We see in Fig. 5A that, no metter the input, a large fraction of neurons is always responding. In Fig. 5B we show the fraction of neurons which contribute in differentiating the representation of different digits. It is interesting to see that inhibitory neurons contribute the most.

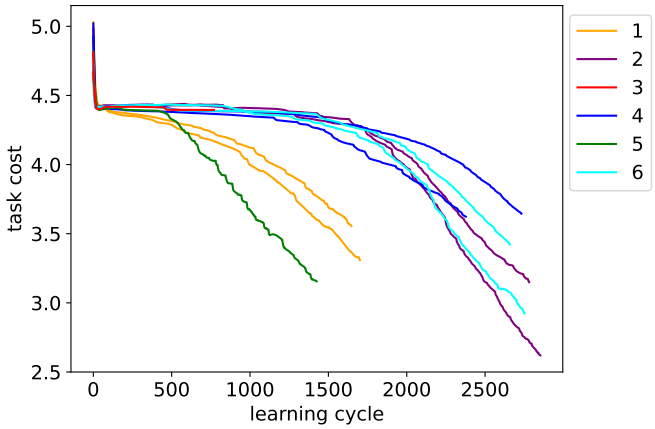

Figure 4. Hebbian rule for inhibitory neurons in Task 1.

5 Further information on Task 2

5.1 Multiple solutions for the stationary activity

As the target connectivity in Figure 5A has its largest singular value larger than one, the stationary equations for the neural activity may have multiple solutions. Multiple solutions are rare but do occur occasionally (notice that limits cycles are in principle possible but were never observed during the stimulation protocol).

When multiple solutions are present, it is possible that the control optimization loop selects a certain contrl stimulation pattern $\mathbf{f}^*$ associated to a specific state of neural activity $\mathbf{r}^*$ but, once the control stimulation is applied , another state of activity, say, $\mathbf{r}'$, is reached by the neural dynamics. We stress that we are addressing here the case of a large difference between $\mathbf{r}^*$ and $\mathbf{r}'$. Small differences are always expected to take place due to the small discrepancies between the true (and unknown) connectivity and its estimate used for the computation of the optimal control. Since the plasticity inducing stimulation is long compared to τ_r, we assume it can be interrupted in time should this scenario take place. Then, the control stimulation is applied again. This goes on until the correct fixed point is selected. This procedure is effective only if there is a limited number of coexisting fixed points, as appears to be the case. In practice, if the wrong solution is reached during the control stimulation phase, the fixed point iteration of the neural dynamics is restarted with a new initial condition.

5.2 Choice of probing stimulations for connectivity inference

While the activation function Φ in Eq. (8) (Methods, main text) is invertible, Φ^{-1} is ill conditioned for very low firing rates. To avoid numerical issues in the estimation of the connectivity, we consider probing stimulations such that the least active neuron still has detectable activity. We use stimulations stronger for excitatory than for inhibitory neurons. Excitatory neurons are stimulated with a random forcing in the range $[f_{max}/2, f_{max}]$, while for inhibitory neurons, the range considered is $[0, f_{max}/2]$.

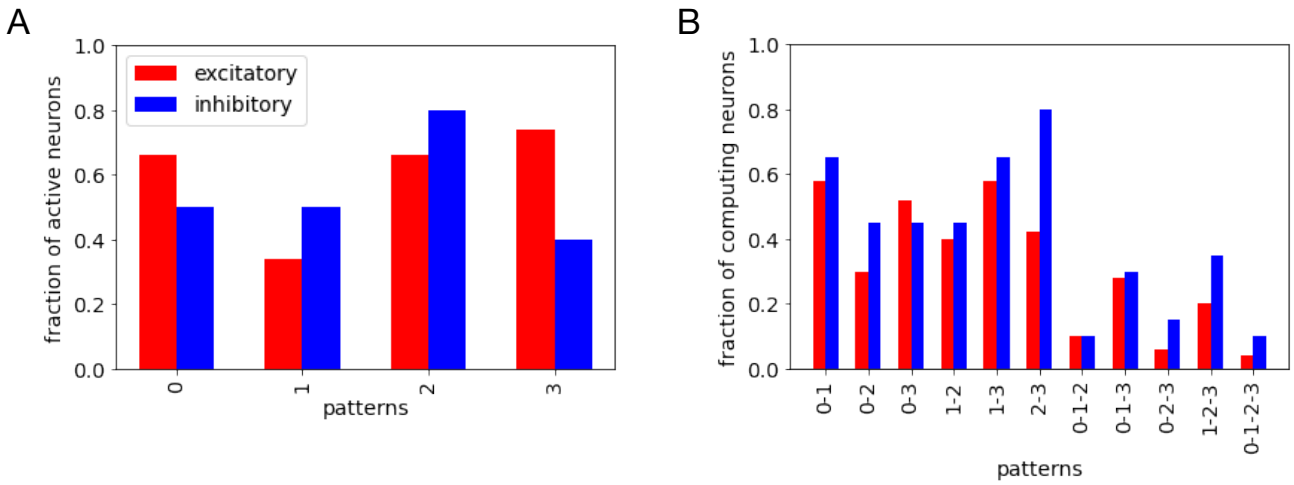

Figure 5. A: Bar plots of the fractions of active bulk neurons ($r/r_{max} > 0.01$) for each input. B: fractions of neurons that separate the representations of two or more inputs. A neuron is said to separate two inputs μ and ν if its activity significantly varies with the applied input, *i.e.* $|r(\mathbf{f}_\mu) - r(\mathbf{f}_\nu)|/r_{max} > 0.01$; this inequality should hold for any pair of inputs if more than two are considered.

Despite this procedure, if a neuron j is found to have low activity (less than $0.2\ 10^{-3}\, r_{max}$), then its stimulation f_j is increased by a random amount in the range $[0, f_{max}/2]$ (and the value is then cropped at f_{max} if necessary). If, after correcting all the components f_j, any neuron remain inactive, the stimulation pattern is discarded and another one is randomly drawn.

At the beginning of the training phase, this procedure is largely unnecessary, but at later (training) times, large quasi-disconnected (*i.e.* with very weak synaptic connections) groups of neurons appear. At this point, the pattern correction procedure become essential. After some times, all neurons will be again connected together and corrections are less and less frequent. We stress that correcting a probing stimulation pattern does not require to stimulate the system more times as, based on our relatively accurate estimate of the network connectivity, we can predict the response of the network in a reliable way.

5.3 Case of anti-Hebbian rule for inhibitory neurons

In this section we train out network on Task 2 with anti-Hebbian learning rule for inhibitory synapses (see 6). We show twelve different settings, highlighting a certain variability in the learning curves.Indeed, the protocol works as intended in the sense that the structural cost decreases. However, if we look at the continuous attractor behaviour, we get a different picture. For instance, we can focus on setting 3 and 7, for which the final cost is approximately the same at the end of the protocol. Likewise, the connectivity structures are very similar (see Fig. 6B). However, in the setting 7, receptive fields appear, while in setting 3 they do not, since. The reason is that the final result is not given by a random but unbiased perturbation of the target connectivity state but, rather, a biased version of it, due to the effect of plasticity constraints in the learning process. Specifically, we can clearly see the interference of the formation of different "stripes" in the connectivity matrix.

The settings are the following (we only specify parameters differing from those of the main text).

- $\theta(E) = 0.175\, r_{max}$, $\theta(I) = 0.1\, r_{max}$, $\eta(I) = -1.1$, $\beta_1 = 6.25$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$
- $\theta(E) = 7\, r_{max}/30$, $\theta(I) = 4\, r_{max}/30$, $\eta(I) = -1$, $\beta_1 = 6.25$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$
- $\theta(E) = 6\, r_{max}/70$, $\theta(I) = 6\, r_{max}/70$, $\eta(I) = -1.5$, $\beta_1 = 6.25$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$
- $\theta(E) = 4\, r_{max}/35$, $\theta(I) = 2\, r_{max}/35$, $\eta(I) = -0.9$, $\beta_1 = 3.125$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$
- $\theta(E) = 2\, r_{max}/35$, $\theta(I) = 3\, r_{max}/35$, $\eta(I) = -1.5$, $\beta_1 = 1.25$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$
- $\theta(E) = 0.12\, r_{max}$, $\theta(I) = 0.08\, r_{max}$, $\eta(I) = -2.$, $\beta_1 = 3.125$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$
- $\theta(E) = 0.16\, r_{max}$, $\theta(I) = 0.08\, r_{max}$, $\eta(I) = -1.1$, $\beta_1 = 3.125$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$
- $\theta(E) = 0.16\, r_{max}$, $\theta(I) = 0.12\, r_{max}$, $\eta(I) = -1.1$, $\beta_1 = 3.125$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$

- $\theta(E) = 0.16\, r_{max}$, $\theta(I) = 0.12\, r_{max}$, $\eta(I) = -3.$, $\beta_1 = 3.125$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$
- $\theta(E) = 0.16\, r_{max}$, $\theta(I) = 0.12\, r_{max}$, $\eta(I) = -1.3$, $\beta_1 = 1.25$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$
- $\theta(E) = 0.12\, r_{max}$, $\theta(I) = 0.12\, r_{max}$, $\eta(I) = -1.3$, $\beta_1 = 3.125$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$
- $\theta(E) = 0.12\, r_{max}$, $\theta(I) = 0.08\, r_{max}$, $\eta(I) = -1.3$, $\beta_1 = 1.25$, $\beta_2 = 2.5 \times (r_{max}/\bar{J})^2$, $\theta_0(E) = \theta_0(I) = 0.16\, r_{max}$

5.4 Effect of variation of parameters

In Fig. 7 we show the learning curves associated to different sets of parameters. We only mention parameters differing from the setting in the main text and, in particular, we provide some example of the effect of perturbation of the hebbian post-synaptic threshold

- $\theta(E) = 0.12\, r_{max}$, $\theta(I) = 0.08\, r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08\, r_{max}$
- $\theta(E) = 0.08\, r_{max}$, $\theta(I) = 0.12\, r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08\, r_{max}$
- $\theta(E) = 0.2\, r_{max}$, $\theta(I) = 0.16\, r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08\, r_{max}$ (as in the main text)
- $\theta(E) = 0.2\, r_{max}$, $\theta(I) = 0.08\, r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08\, r_{max}$
- $\theta(E) = 0.16\, r_{max}$, $\theta(I) = 0.12\, r_{max}$, $\theta_0(E) = \theta_0(I) = 0.08\, r_{max}$

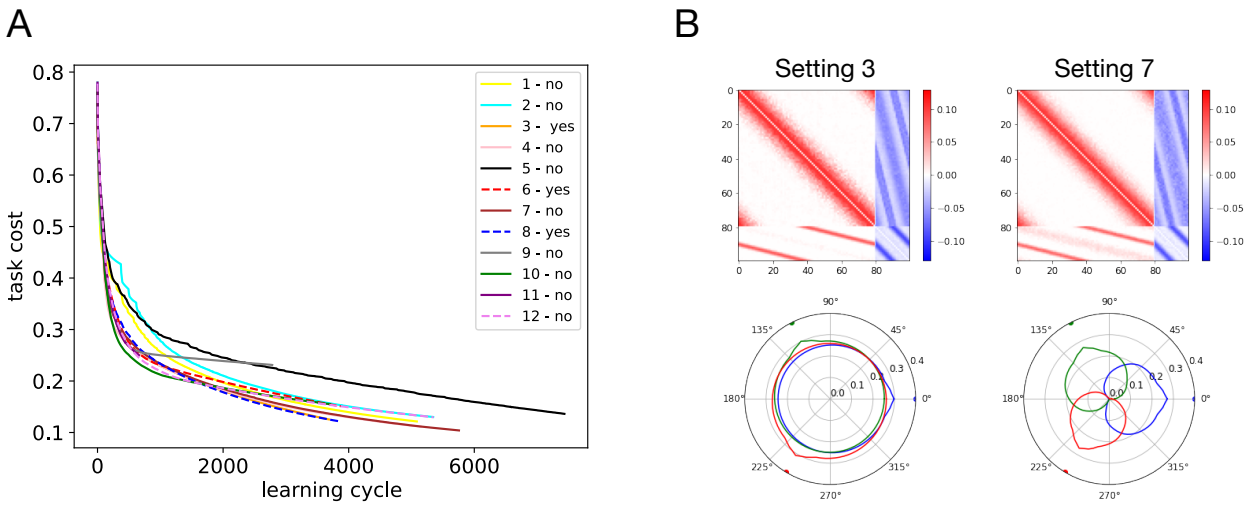

Figure 6. Panel A: anti-hebbian inhibitory synapses. We show different learning curves for 12 different parameter settings for Task 2. The label on each curve (yes/no) tells if the attractor was generated or not. Dotted lines are used to make overlapping lines visible. Panel B shows the connectivity matrix **J** and the receptive fields for settings 3 and 7 at the end of the training, while panel. Notice how similar connectivity structures and cost values are associated to very different performances in term of receptive field formation.

6 Additional Task 3: logical AND function in a small circuit

Definition of the task. We propose an additional simple task in a feed-forward network, with few neurons and simplified activation function, which lacks the complexity and theoretical interest of the tasks in the main text, but offers a detailed visualization of the quantities involved in learning. We set $r_{max} = \infty$ and $r_0 = 0$ (i.e. we use the standard , i.e. the standard ReLU as activation function). Hence, we cannot express quantities in units of r_{max}. Instead, we use as a unit of measure a value $\bar{r}$ which is assumed to be an intermediate frequency $r_0 \ll \bar{r} \ll r_{max}$. The task consists in the realization of a logical AND between two inputs in a circuit made of $N = 4$ neurons, out which $N_E = 3$ are excitatory and $N_I = 1$ is inhibitory. We assume neurons 1 and 2 can receive binary input stimulations, in practice $f_1, f_2 = 0 \times \bar{r}$ or $1 \times \bar{r}$ (which we identify, respectively, with

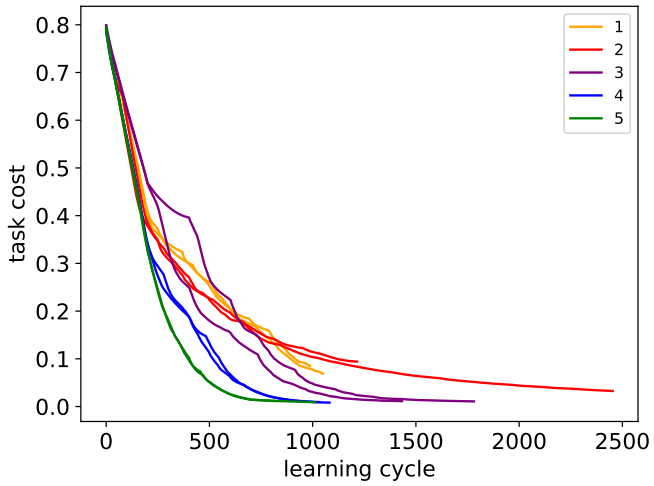

Figure 7. Learning curves for Task 2 with Hebbian inhibitory connections (as in the main text) but with different parameter choices.

logical values False='0' and True='1'), while we set $f_3 = 1\,\bar{r}$. We demand that the activity r_4 of neuron 4 expresses the logical AND, i.e. $r_4 = \bar{r}(=\text{'1'})$ if both f_1 and f_2 are '1', and $r_4 = 0(=\text{'0'})$ otherwise. A possible circuit implementing this task in shown in Fig. 8. Two excitatory neurons (nodes 1 and 2) and an interneuron (node 3) are connected to an output neuron (node 4). The non-zero connections are set to $J_{41} = J_{42} = -J_{43} = 1$: the inhibitory input to neuron 4 coming from neuron 3 is large enough to suppress the inputs from either neuron 1 or 2, but not from both.

We next train a network with random initial connectivity with the following cost, see Eq. (4),

$$U_{task}(\mathbf{J}) = \frac{1}{4}\left[\left(r_4(1,1)-\bar{r}\right)^2 + \left(r_4(0,1)-0\right)^2 + \left(r_4(1,0)-0\right)^2 + \left(r_4(0,0)-0\right)^2\right], \tag{18}$$

which is the sum of squared errors for the four possible combinations of binary inputs. The activities of neurons 1, 2 and 3 are not considered in the cost. This cost is functional and not structural: it does not impose *a priori* any target value for the connectivity matrix, but rather constrains the function to the achieved.

Training: stimulations and network. Despite its simplicity, the AND task case highlights the underlying difficulty in training a plastic network and the difference with the standard gradient descent (GD) used in machine learning. Figure 8B shows the cost dynamics with GD and with our stimulation protocol. While both algorithms successfully learn the task, GD is faster, as we expect. For the plastic network, the three synapses cannot be updated independently, since the activity of the post-synaptic neuron 4, which enters both the Hebbian and the homeostatic part of the plasticity rule, influences at once the evolution of all the connections we are trying to update. As a result of this constraint, the changes in the $\mathbf{J}$ matrix cannot align along the gradient of the cost as is the case for the GD dynamics, see Fig. 8D. Our protocol is nevertheless able to make the cost decrease at all times. This decrease demonstrated that the task is progressively learned, as visible from the responses of neuron 4 to all possible pairs of inputs reported in Fig. 8E.

The behaviour of the connectivity is displayed in Fig. 8C. We observe that our stimulation protocols drives the network connectivity to the circuit shown in Fig. 8A. It is easy to check that this network is the only one achieving zero cost, see Eq. (18) and Methods. For larger networks, we expect the task to be achieved by many connectivity networks[9], and our learning protocol will select one of them.

The time course of the applied stimulations $f_i(t)$ is displayed in Fig. 8F. We observe that, most of the time, the control smoothly changes across stimulation periods, but occasionally shows large discontinuities (as explained in Fig. 2) in this case, no discontinuity is due to an external reset.

Some remarks about the solution. Assuming symmetry between neurons 1 and 2, we can call $J_+ = J_{41} = J_{42}$ and $J_- = -J_{43}$. Then, essentially, the task is solved when we have $1 = \max(2J_+ - J_-, 0) = 1$ and $\max(J_+ - J_-, 0) = 0$ (and $\max(-J_-, 0) = 0$ which is always true). It means that the solution is given by the curve $J_- = 2J_+ - 1$ in the $\{J_- \geq J_+\} \cap \{J_\pm > 0\}$ space, which is the $J_- = 2J_+ - 1$ curve for $J_+ \geq 1$. Since we start with small synaptic strength values, it is not surprising the smallest possible values for the connections are selected by optimization procedure, as shown in Fig. 8C.

Parameters. We use $\Delta t = 10^{-2}\tau_n$, $\theta(E) = \theta(I) = \theta_0(E) = \theta_0(I) = \bar{r}/2$, $\beta_1 = 0.2$, $\beta_2 = 0$.

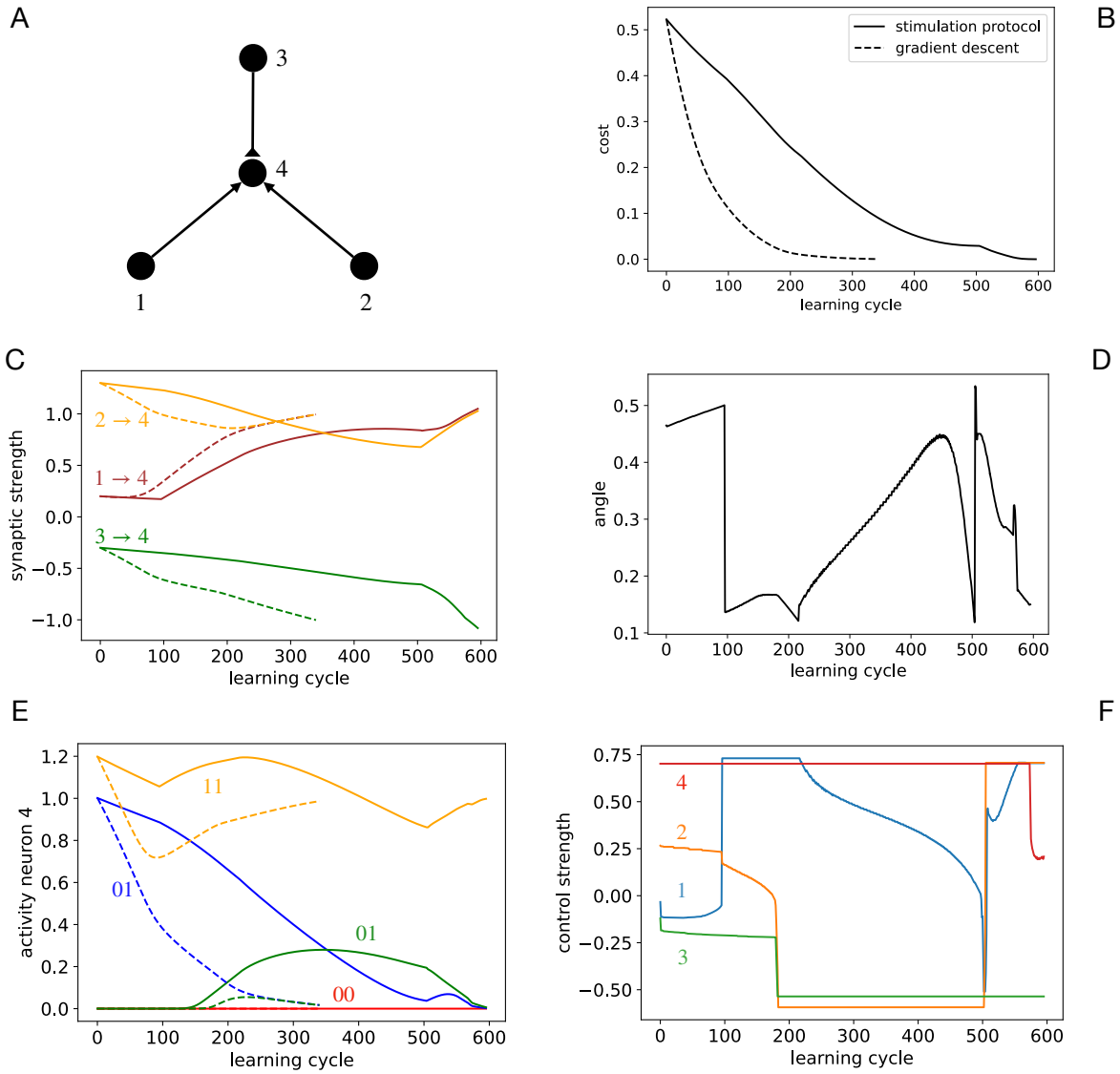

Figure 8. A Example of a 4-neuron circuit implementing a logical AND between two inputs. Two excitatory (nodes 1 and 2) and one inhibitory (node 3) neurons are connected to an output neuron (node 4). The three non zero connections are approximately $J_{41} = 1$, $J_{42} = 1$ and $J_{43} = -1$ at the end of the training. We set $f_{max} = 0.7\,\bar{r}$ and $f_{min} = -0.5\,\bar{r}$. B: Cost U_{task} as a function of the time step (time $t/\Delta t$) during learning with our stimulation protocol (full curve) and standard gradient descent (dashed line). For the latter, we choose the learning rate in such a way that the first connectivity change has the same norm $\|\Delta \mathbf{J}\|$ as with the stimulation protocol. C: activity r_4 of neuron 4 (in $\bar{r}$ units)in response to the four possible pairs of inputs onto neurons 1 and 2 through the training process. D: Cosine similarity between (minus) the gradient of the cost $\partial U/\partial \mathbf{J}$ and the time derivative of the connectivity $\dot{\mathbf{J}}$. The gap between the similarity and one (corresponding to gradient descent) reflects the limited accessibility of the connectivity space imposed by the plasticity rule, see Fig. 3 from the main text. Notice that the similarity remains positive at all times, implying that the plasticity rule can be exploited to decrease the cost and learn the task. (E) Evolution of the synaptic strengths during learning with our stimulation protocol (full lines) and standard gradient descent (dashed lines). (F) control $f_i(t)$ (in $\bar{r}$ units) applied to the neurons $i = 1,2,3,4$. The control stimulations smoothly vary for a certain period of time, then abruptly jump when no local solution for the control optimization process are found near that at previous step.

References

1. Moradi Chameh, H. *et al.* Diversity amongst human cortical pyramidal neurons revealed via their sag currents and frequency preferences. *Nat. Comm.* **12**, 2497 (2021).

2. Nowak, L., Azouz, R., Sanchez-Vives, M., Gray, C. & McCormick, D. Electrophysiological classes of cat primary visual cortical neurons in vivo as revealed by quantitative analyses. *J Neurophysiol* **89**, 1541–66 (2003).

3. Sayer, R., Friedlander, M. & Redman, S. The time course and amplitude of epsps evoked at synapses between pairs of ca3/cal neurons in the hippocampal slices. *J. Neurosci.* **10**, 826–836 (1990).

4. Wang, H. *et al.* High-speed mapping of synaptic connectivity using photostimulation in channelrhodopsin-2 transgenic mice. *Proc. Natl. Acad. Sci.* **104**, 8143–8148 (2007).

5. Mager, T., Lopez de la Morena, D., Senn, V. & al. High frequency neural spiking and auditory signaling by ultrafast red-shifted optogenetics. *Nat. Commun.* **9**, 1750 (2018).

6. Rodgers, J., Bano-Otalora, D., Belle, M. & al. Using a bistable animal opsin for switchable and scalable optogenetic inhibition of neurons. *EMBO Reports* **22**, e51866 (2021).

7. Zenke, F., Gerstner, W. & Ganguli, S. The temporal paradox of hebbian learning and homeostatic plasticity. *Curr. opinion neurobiology* **43**, 166–176 (2017).

8. Lenhart, S. & Workman, J. T. *Optimal control applied to biological models* (CRC press, 2007).

9. Marder, E. & Taylor, A. Multiple models to capture the variability in biological neurons and networks. *Nat Neurosci* **14**, 133–138 (2011).