

RESEARCH ARTICLE SUMMARY

NEURAL COMPUTATION

Spiking neurons can discover predictive features by aggregate-label learning

Robert Gütiġ

INTRODUCTION: Opportunities and dangers can often be predicted on the basis of sensory clues. The attack of a predator, for example, may be preceded by the sounds of breaking twigs or whiffs of odor. Life is easier if one learns these clues. However, this is difficult when clues are hidden within distracting streams of unrelated sensory activity. Even worse, they can be separated from the events that they predict by long and variable delays. To discover those clues, a learning procedure must bridge the gap between the short epochs within which clues occur and the time when feedback arrives. This “temporal credit-assignment problem” is a core challenge in biological and machine learning.

RATIONALE: A neural detector of a sensory clue should fire whenever the clue occurs but

remain silent otherwise. Hence, the number of output spikes of this neuron should be proportional to the number of times that the clue occurred. The reversal of this observation is the core hypothesis of this study: A neuron can identify an unknown clue when it is trained to fire in proportion to the clue’s number of occurrences. This “aggregate-label” hypothesis entails that when a neuron is trained to match its number of output spikes to the magnitude of a feedback signal, it will identify a set of clues within its input activity whose occurrences predict the feedback. This learning requires neither knowledge of the time nor of the absolute number of individual clues.

RESULTS: To implement aggregate-label learning, I calculated how neurons should modify their synaptic efficacies in order to most ef-

fectively adjust their number of output spikes. Because a neuron’s discrete number of spikes does not provide a direction of gradual improvement, I derived the multi-spike tempotron learning rule in an abstract space of continuous spike threshold variables. In this space, changes in synaptic efficacies are directed along the steepest path, reducing the discrepancy between a neuron’s fixed biological spike threshold and the closest hypothetical threshold at which the neuron would fire a desired number of spikes. With the resulting synaptic learning rule, aggregate-label learning enabled simple neuron models to solve the temporal credit assignment problem. Neurons reliably identified all clues whose occurrences contributed to a delayed feedback signal. For instance, a neuron could learn to respond with different numbers of spikes to individual clues without being told how many different clues existed, when they occurred, or how much each one of them contributed to the feedback. This learning was robust to high levels of feedback and input noise and performed well on a connected speech-recognition task.

Aggregate-label learning enabled populations of neurons to solve unsupervised learning tasks by relying on internally generated feedback

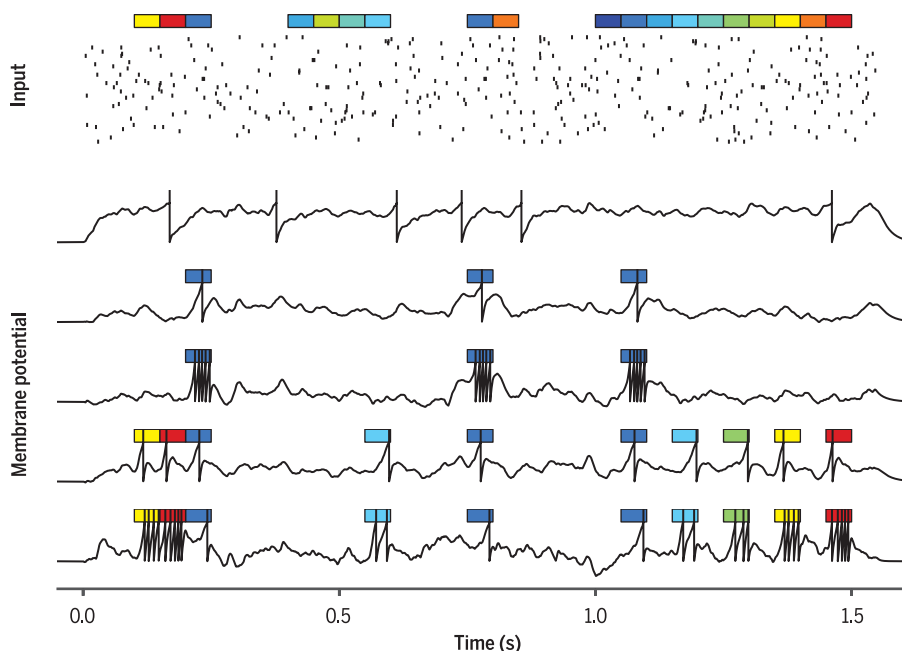
ON OUR WEB SITE

Read the full article at <http://dx.doi.org/10.1126/science.aab4113>

signals that amplified correlations between the neurons’ output spike counts. These self-supervised networks discovered reoccurring constellations of input patterns even if they were

rare and distributed over spatial and temporal scales that exceeded the receptive fields of individual neurons. Because learning in self-supervised networks is driven by aggregate numbers of feature occurrences, it does not require temporal alignment of the input activities of individual neurons. When competitive interactions between individual neurons were mediated through the internal feedback circuit, the formation of feature maps was possible even when the features’ asynchrony incapacitated lateral inhibition.

CONCLUSION: Aggregate-label learning solves the long-standing question of how neural systems can identify features within their input activity that predict a delayed feedback. This solution strongly enhances the known learning capabilities of simple neural circuit models. Because the feedback can be external or internal, these enhancements apply to supervised and unsupervised learning. In this framework, both forms of learning converge onto the same rule of synaptic plasticity, inviting future research on how they cooperate when brains learn. ■



Membrane potential traces of a model neuron before and after learning to detect reward predictive sensory clues. Before learning, top trace; after learning, second through fifth traces from top; clues, colored squares. Each clue occurrence is represented as a spike pattern within the neuron’s input activity (raster plot). After learning, the number of output spikes (vertical deflections) elicited by each clue encodes the clue’s contribution to a delayed reward.

E-mail: guetig@em.mpg.de

Cite this article as R. Gütiġ, *Science* 351, aab4113 (2016). DOI: 10.1126/science.aab4113.

RESEARCH ARTICLE

NEURAL COMPUTATION

Spiking neurons can discover predictive features by aggregate-label learning

Robert Gütig

The brain routinely discovers sensory clues that predict opportunities or dangers. However, it is unclear how neural learning processes can bridge the typically long delays between sensory clues and behavioral outcomes. Here, I introduce a learning concept, **aggregate-label learning**, that enables biologically plausible model neurons to solve this temporal credit assignment problem. Aggregate-label learning matches a neuron's number of output spikes to a feedback signal that is proportional to the number of clues but carries no information about their timing. Aggregate-label learning outperforms stochastic reinforcement learning at identifying predictive clues and is able to solve unsegmented speech-recognition tasks. Furthermore, it allows unsupervised neural networks to discover reoccurring constellations of sensory features even when they are widely dispersed across space and time.

The fitness of an individual depends on its ability to recognize circumstances that require specific actions to procure reward or avoid punishment. The approach of a predator, for example, may be preceded by one or more brief sensory clues, such as the sounds of breaking twigs or whiffs of odor. Humans and other animals are very good at learning such associations, even when clues are hidden within distracting streams of unrelated sensory activity of similar structure, and the delays between clues and outcomes are variable and orders of magnitudes longer than the durations of individual clues. To discover those clues, a learning procedure must bridge the gap between the short epochs within which clues occur and the long delays after which feedback arrives. This “temporal credit-assignment problem” is a core challenge in biological and machine learning (1–4). Often, the assignment of credit to a particular sensory clue is made even harder by the fact that the learner must not only discover when but also in which sensory channel a predictive clue can be found. This is the “spatial credit assignment problem.”

To forecast opportunities or dangers, a neural feature detector should be active whenever a predictive sensory clue occurs but remain silent otherwise. If the timing of the clues is known, synaptic learning rules (5–8) that resemble perceptron learning (9) can realize such desired responses by strengthening synapses that depolarize the cell at the time of a clue and weakening those that do otherwise. However, when the timing of the clues is unknown, an appropriate objective function that

can provide the direction of synaptic changes in spiking neurons has remained elusive (10). Applications of stochastic reinforcement learning have resulted in slow learning and limited performance (3, 4, 11–13). Other, non-neuromorphic solutions rely on complex iterative optimization algorithms such as expectation-maximization (14). Their biologically plausible implementation has remained challenging. Here, I show that the temporal credit assignment problem can be solved by training a neuron to match its number of output spikes to the magnitude of a feedback signal.

Results

Aggregate-label learning

A reliable clue detector should fire at least one action potential whenever the clue occurs but must remain silent otherwise. Hence, the number of spikes elicited within a given sensory episode should grow with the number of clue occurrences. Because we have only aggregate feedback information, I used the total number of spikes when constructing an objective function to drive learning (15). Here, I introduce the learning concept of **aggregate-label learning**: The temporal credit assignment problem is solved by a broad class of synaptic learning rules that credit individual synapses according to their ability to reduce the mismatch between the number of output spikes and the magnitude of the aggregate feedback. This feedback is only assumed to be proportional to the number of clues but provides no information about their timing or absolute numbers. Aggregate-label learning entails that neurons can learn to respond synchronously to predictive sensory clues from delayed feedback signals (16). I implemented aggregate-label learning with two specific synaptic learning rules: the multi-spike tempotron, a gradient-based learning rule that adjusts synaptic efficacies along

the steepest path toward the desired number of output spikes, and an approximation that is biologically more plausible.

The multi-spike tempotron

The “minimum disturbance principle” (17) posits that a neural implementation of aggregate-label learning should use a synaptic learning rule that adjusts a neuron's synaptic efficacies in the direction along which its number of output spikes changes most rapidly. Naive gradient-based approaches to find this direction fail in spiking neurons because of the discrete nature of the number of output spikes. Its derivatives with respect to a neuron's synaptic efficacies are zero almost everywhere and infinite at points where a new spike appears or an existing one disappears. To circumvent this problem, stochastic approaches have been tried with mixed success (11, 18, 19). Here, I introduce an alternative approach that is based on a continuous deterministic objective function for spiking neurons, the spike-threshold-surface. For each synaptic configuration and presynaptic activity pattern, the spike-threshold-surface is defined as the mapping between the neuron's firing threshold and the number of elicited output spikes (materials and methods, spike-threshold-surface) (fig. S1). On the basis of this curve, the discrete number of elicited output spikes is replaced with the distance between the fixed biological firing threshold of a neuron and the closest hypothetical threshold at which the neuron would fire the desired number of spikes. When this distance is zero, the neuron responds to the input pattern with the desired number of spikes. The multi-spike tempotron is defined as the gradient-based learning rule that approaches the desired number of spikes along the (locally) steepest path (materials and methods, multi-spike tempotron) (fig. S1D). I implemented the multi-spike tempotron by calculating the exact gradient (materials and methods, ∂^* -gradient). I also approximated the learning rule with a biologically established form of synaptic plasticity that is driven by correlations between presynaptic firing and postsynaptic depolarization. Specifically, in this correlation-based learning each synapse maintains a local trace of eligibility by correlating its own activity with the postsynaptic membrane potential (materials and methods, correlation-based learning). Beyond these eligibilities, this learning does not require a memory of past inputs or membrane potential values.

Learning predictive sensory clues

I tested aggregate-label learning within a family of generic feature learning tasks for single neurons. Each task consisted of a fixed set of brief (50 ms long) activity patterns that represented responses of a neuron's afferent periphery to occurrences of distinct features within an organism's sensory environment (Fig. 1A). In each trial, a random number of feature activity patterns was embedded within a long stream of random background activity at random times (materials and methods, embedded features). Each feature was either a clue or a distractor. The task of the neuron was to signal all clues by

Max Planck Institute of Experimental Medicine, Hermann-Rein-Strasse 3, 37075 Göttingen, Germany.
E-mail: guetig@em.mpg.de

firing a specific number of action potentials whenever the corresponding feature activity pattern occurred within its input stream. In contrast, the neuron had to remain silent during intervals of background activity or when encountering a distractor. In a first test, I trained the multi-spike tempotron to detect a single clue (out of 10 features) by matching each trial's aggregate label to the clue's number of occurrences. All remaining features were distractors. Whenever the cell fired fewer or more spikes than there were clues, the learning rule adjusted the neuron's synaptic efficacies along the direction of the steepest path toward the synaptic configuration that elicited one more or one less spike, respectively (materials and methods, multi-spike tempotron, and Eq. 7). Neural responses (materials and methods, neural responses) to random background activity and to all distractor features declined rapidly, and the cell learned to fire exactly one spike whenever the clue occurred (Fig. 1, B, second trace, and C, left; and fig. S2A) (response statistics are provided in table S1). I have checked that the correlation-

based approximation of the multi-spike tempotron resulted in a similar learning performance (fig. S3A).

Aggregate-label learning is not limited to a particular gain between a clue's number of occurrences and the feedback signal. When the desired number of output spikes corresponded to five times the number of the clues, the neuron learned to fire a burst of five spikes in response to each clue (Fig. 1B, third trace; fig. S2A; and table S1). Aggregate-label learning also enables neurons to decompose the feedback signal and identify the individual values of multiple sensory clues whose occurrences are mixed within a sensory scene (Fig. 1, B, fourth and fifth traces, and C, right; fig. S2B; and table S1). This problem arises in many ethological settings (2) whenever feedback signals incorporate contributions from several sources, such as food quality being associated with different color and odor clues. Our neural implementation of aggregate-label learning remains efficient even when applied to long sensory episodes dominated by background activity (Fig. 2A and fig. S2C) or when the number of distractor features is high

(Fig. 2A). It also tolerates a wide range of clue frequencies (Fig. 2B). The learning is robust to substantial correlations between clue and distractor occurrences (materials and methods, correlated feature occurrences) (Fig. 2C) and reproduces the phenomenon of "blocking" (2) when correlations are high (fig. S2D). When compared with a recent neural implementation of reinforcement learning (4), aggregate-label learning required much fewer synapses and learning trials to solve a feature learning task that was similar (although substantially simpler) to the ones studied here (materials and methods, reinforcement learning comparison task) (table S2) (20). Last, when sensory clues were not discrete but instead drawn from a continuous feature dimension, aggregate-label learning enabled neurons to learn continuous tuning curves (supplementary text S1 and fig. S4). The outcome of seeing a face, for instance, is likely to depend on its direction of gaze (ranging from looking straight at us to looking away) or its emotional expression (between angry and friendly).

Noisy feedback

In many learning tasks, the relation between clues and feedback can be unreliable. For instance, sensory clues can be occluded or feedback signals corrupted by noise. To test aggregate-label learning in such conditions, I modified the desired number of spikes on each trial by adding or subtracting (with equal probabilities) random Poisson noise. I found that neurons continued to reliably identify sensory clues even when the feedback noise was high (Fig. 2D). However, increasing levels of noise caused a gradual decline in the average number of spikes elicited by the clues in favor of spikes elicited by random background activity. This behavior reflects the inability of the identified clues to fully account for the (corrupted) feedback signal and could be interpreted as the neuron's attempt to search for a better predictor within the background activity. However, when the teacher is wrong, definitions of good and bad learning become circumstantial.

Prompted by its robustness to noise in the above scenario, I next considered whether aggregate-label learning can be extended to a well-studied probabilistic feature learning task. In the "weather prediction task" (27), relations between clues and feedback are not deterministic, but rather, clue occurrences increase the probability of a binary reward at the end of a trial (materials and methods, probabilistic reward contingencies). In addition to introducing feedback noise through probabilistic rewards, this scenario also relaxes our previous assumption that the feedback is proportional to the number of clue occurrences. Instead, here the number of clue occurrences is related to the reward (probability) through a saturating nonlinearity whose steepness is determined by the clue's reward contingency. The higher the contingency, the more step-like this mapping becomes. Because of this task's stochastic nature, the relation between clues and rewards has to be acquired and maintained across trials—for example, the absence of a reward on a single trial should not cause the immediate unlearning

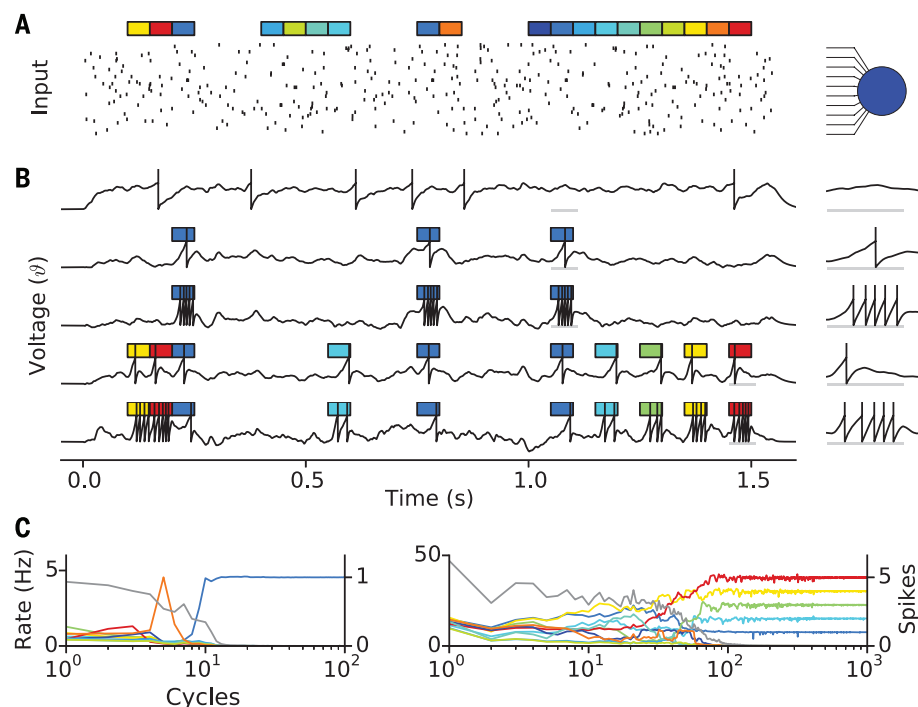


Fig. 1. Supervised learning of sensory clues. (A) Input activity (left raster plot) and schematic of a postsynaptic readout neuron (right). For visibility, only spikes (black ticks) of 10% of the neuron's 500 synaptic afferents are shown. Colored rectangles depict occurrences of $N_f = 10$ distinct sensory features (color coded) whose corresponding 50-ms activity patterns have been embedded within a stream of random background activity. (B) Voltage traces of the readout neuron before (top trace) and after being trained to fire one (second trace) or five (third trace) spikes in response to a single clue (dark blue rectangles), respectively. For the fourth and fifth traces, the neuron was trained to respond to five clues out of the 10 features (colored rectangles). The neuron either had to fire one spike in response to each of the clues (fourth trace) or one, two, three, four, and five spikes in response to the dark blue, light blue, green, yellow, and red clues, respectively (fifth trace). (Right) Enlargements of the underlined (gray horizontal lines) clue intervals. (C) Neural responses (materials and methods, neural responses) as a function of learning cycles (each containing 100 trials) for the two tasks underlying the second (left) and fifth (right) voltage traces in (B), respectively. Colored lines show the mean numbers of spikes (right y axis) elicited by each feature [color coded as in (A)], and gray lines show the mean responses to background activity (left y axis). Population learning curves are provided in fig. S3, left.

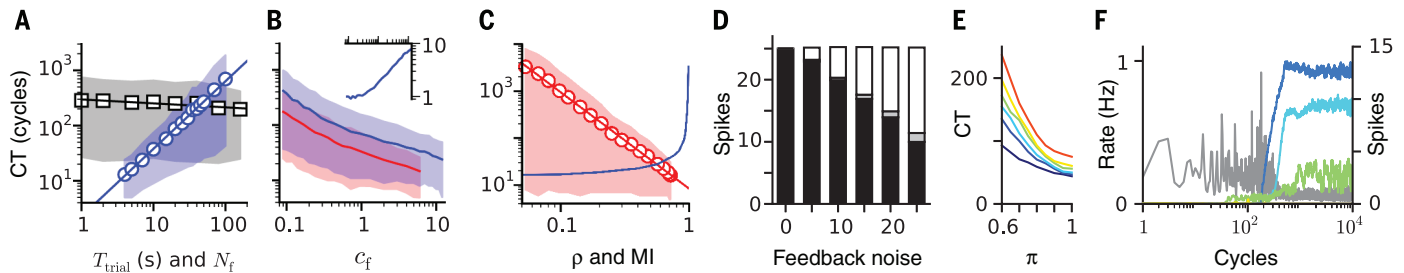


Fig. 2. Robustness of aggregate-label learning. (A to C) Mean convergence times (CT) of a neuron that was trained to fire one spike in response to a single clue for different task parameters. (A) For a fixed mean feature count $c_f^{\text{mean}} = 5$ as a function of the mean trial duration T_{trial} (blue circles) and for a fixed mean trial duration $T_{\text{trial}} = 50$ s as a function of the number of features N_f (black squares). Solid lines depict exponential fits with slopes of 1.23 and -0.07 , respectively. (B) For fixed mean trial durations $T_{\text{trial}} = 5$ s (red) and $T_{\text{trial}} = 10$ s (blue) as a function of the mean feature count c_f^{mean} . (Inset) The expected number of clue encounters before convergence in the $T_{\text{trial}} = 10$ s condition as a function of c_f^{mean} (same interval as in the main data) normalized to the value at $c_f^{\text{mean}} = 0.1$ (37.33). (C) For a fixed mean trial duration $T_{\text{trial}} = 5$ s and a fixed mean feature count $c_f^{\text{mean}} = 5$ as a function of feature correlations ρ (blue curve) or the corresponding normalized mutual information MI between the aggregate-label and the clue identity (red circles). The red line shows an inverse power-law relationship, with a nearly quadratic exponent of -2.05 . Shaded areas in (A), (B), and (C) correspond to regions between the 10th and 90th

percentile of 1000 independent simulations. (D) Learned responses of a neuron that was trained to fire five spikes in response to a single clue $c_f^{\text{mean}} = 5$ as function of the average noise that corrupted the feedback signal on each trial. Neural responses were averaged over the last 100 of 1000 cycles of training and converted to the expected number of spikes that were elicited by occurrences of the clue (black), the nine distractors (gray), or the 7.5 s of background activity (white) during each trial. All measurements were averaged over 1000 independent simulations. (E) Mean convergence times in the probabilistic single-clue detection task as a function of the clue's reward contingency π . Colors (cool to warm) depict rising noise levels in the input activity patterns implemented by an increasing spike deletion probability of 0, 0.1, 0.2, 0.3, 0.4, and 0.5. (F) Neural responses as in Fig. 1C, but for the probabilistic reward scenario. Three clues had reward contingencies of $\pi_1 \approx 1$ (dark blue), $\pi_2 \approx 0.8$ (light blue), and $\pi_3 \approx 0.6$ (green), respectively, and on each trial, a quarter of all input spikes were randomly deleted. Except the black line in (A), $N_f = 10$ throughout this figure.

of the corresponding clue. I therefore modified the learning scheme so that each rewarded trial invoked a learning step toward increasing the neuron's output spike number regardless of its response. In contrast, unrewarded trials triggered a learning step to reduce the number of output spikes only in the case of a "false alarm," when the neuron had fired at least one spike. This asymmetric learning continues to strengthen the neuron's responses to the sensory clues even after they have been identified, increasing their robustness. In addition, selectivity of the learned neural responses is enhanced by synaptic competition (22) that was induced here by bounding the overall strength of the neuron's synaptic efficacies through divisive normalization (materials and methods, probabilistic reward contingencies). The resulting learning reliably identified predictive clues, even if their reward contingencies were far from perfect and, in addition, all activity patterns were corrupted by random deletions of presynaptic spikes on each trial (Fig. 2E). I checked that the learned neural responses were hardly affected by the presence of false rewards that were not predicted by any of the available clues. The main effect of such false rewards was to slow down the clue acquisition (fig. S5A). When the reward probability resulted from the combination of several different clues, the relative strengths of the average learned responses reflected each clue's individual predictive strength (Fig. 2F and fig. S5, B and C). Thus, learned responses of multi-spike tempotrons with stochastic feedback resembled recordings in the lateral intraparietal cortex of rhesus monkeys performing a similar task (21).

Application to speech recognition

In machine learning approaches to automatic speech recognition, it is easy to count occurrences

of words or phonemes within a known text. However, their temporal alignment with a corresponding speech signal has been notoriously difficult and requires complex optimization algorithms (14). I tested whether word counts are sufficient to successfully apply aggregate-label learning to continuous speech recognition problems. I addressed this question by extending a recent neural model of spoken digit recognition (materials and methods, auditory front-end) (23) from binary to multi-spike tempotrons. In contrast to the previously used TI46 database, which is limited to utterances of single digits, I used the TIDIGITS speech corpus (24), which also contains sequences of digits of variable length (ranging from 2 to 7). With all utterances of single digits reserved to test performance, training was based on unsegmented digit sequences only. Specifically, I used the number of occurrences of a single target digit as aggregate-label to train neurons to discriminate between this target and the remaining distractor digits. I found that multi-spike tempotrons reliably learned to fire a single spike whenever their target digit occurred and to remain silent otherwise (Fig. 3A). Across all digits, the majority of simulations achieved zero errors over the test set (Fig. 3B), with an average test error of 0.0055 (materials and methods, TIDIGITS-task). For comparison with the binary tempotron model (12), I also evaluated the test error under a binary decision rule, which only required the neuron to fire at least one spike when the target digit occurred within its input. With this looser performance measure, the test error of the multi-spike tempotron reduced to 0.0021, outperforming the binary tempotron by 23%. I next exploited that $\sim 20\%$ of the training sequences in which a target digit occurred contained multiple occurrences. To expose the role of aggregate-label learning,

I repeated the comparison with a strongly reduced training set in which target digits, if present, occurred multiple times. Indeed, the performance gap between the two models widened to 47%. In contrast, the difference fell to 16% when trials with multiple target digit occurrences were excluded. This remaining difference shows that the ability of the multi-spike tempotron learning rule to enforce a single output spike, as opposed to realizing at least one, resulted in an advantage over the binary tempotron, even when the labels of a task were binary. I expect that the advantages of aggregate-label learning will become even more pronounced in applications to neural models of phoneme detection that will encounter higher numbers of target feature occurrences.

Self-supervised neural networks

Consider the above family of feature learning tasks, however, without any feedback (25, 26). How can neurons still "discover" synaptic activity patterns that represent repeated occurrences of sensory features in the environment? In general, structure is discriminated from noise through correlations. Indeed, existing models of unsupervised learning in spiking neural networks have typically relied on spatiotemporally local correlations between the pre- and postsynaptic activity of individual neurons to detect structure within a networks input activity (27)—for instance, through spike-timing-dependent synaptic plasticity (25, 26, 28). Can aggregate-label learning enable populations of neurons to detect structure across much larger scales in space and time by amplifying correlations between the neurons' output spike counts? In contrast, neurons that respond to random inputs typically generate uncorrelated numbers of output spikes. I tested this hypothesis by modeling a neural processing

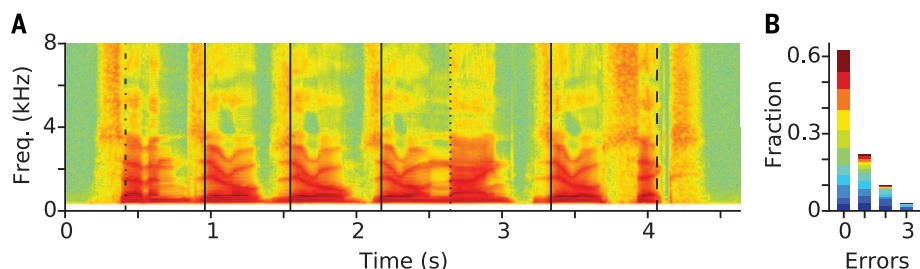


Fig. 3. Application of aggregate-label learning to continuous speech recognition. (A) Spectrogram of one example utterance of the TIDIGITS database (file “7zzz9z6a.wav” of male speaker “ka”) containing the spoken digit sequence “seven,” “zero,” “zero,” “zero,” “nine,” “zero,” “six.” Vertical lines show the output spike times of four neurons that were trained with the occurrence counts of the digits “seven” (dot-dashed), “zero” (solid), “nine” (dotted), and “six” (dashed). Each of the neurons responds to the occurrence of its target digit by firing a single spike. (B) Histogram of individual test errors over the isolated digits subset of the database. The histogram comprises error counts of 10 independent simulations for each of the 242 task conditions (11 target digits \times 22 dialects). Colors depict contributions from individual digits [starting with “zero” (darkest blue) followed by “oh,” numerical values increase with color temperature].

layer as a population of multi-spike tempotrons that were driven by a common sensory periphery. Learning in this population was controlled by a feedback signal that was generated internally by a single downstream supervisor unit (Fig. 4A) and signaled a common desired number of output spikes to all processing layer neurons. To amplify correlations between the processing layer responses—to drive neurons to respond to structure within their input activity—the common feedback signal was based on the average number of spikes fired within the processing layer during the preceding sensory episode. This self-supervision entailed that all processing neurons that fired more spikes than the population average implemented a learning step toward a reduced spike response. Vice versa, neurons that fired fewer spikes than their peers’ average response adjusted their synaptic efficacies toward an increased output spike count. If by chance a subset of neurons responded to occurrences of the same sensory feature, their elevated joint weight within the average population response biased the remaining neurons to search for a feature that occurred the same number of times. This learning will come to an end when all processing layer neurons generate an equal number of output spikes (materials and methods, zero labels, correlations in synaptic efficacies and synaptic learning noise)—in particular, when neurons cease to respond to random background activity and respond with the same number of spikes to all sensory features. Indeed, when applied to the embedded feature detection task, processing layer neurons of self-supervised networks converged (materials and methods, convergence in unsupervised learning tasks) to subsets of the recurring features and encoded each of their occurrences by firing the same number of spikes while remaining silent otherwise (Fig. 4, B and C; population data is provided in fig. S6A). The ability of aggregate-label learning to overcome long feedback delays permits the network’s supervisory circuit to integrate the activity of the processing layer over long time windows. Within these windows, features are bound together across

the individual receptive fields of different neurons on the basis of their common occurrence counts, irrespective of their timing. In particular, features can occur asynchronously across the processing layer. To demonstrate this capability, I implemented a self-supervised network architecture in which each processing layer neuron received individual input activity (Fig. 4D): For each neuron, sensory features were represented by individual activity patterns that occurred at independent random times within independent random background activity. Such individual input architectures mimic neural integration stages of segregated or even multimodal sensory input streams—such as visual, auditory, and olfactory—whose activity might not be temporally aligned. Self-supervised learning functioned also within this network architecture (Fig. 4B, inset). Detailed analyses (supplementary text S2) showed that self-supervised learning is sensitive to the number and the timing of spikes within a feature’s activity pattern (fig. S6, C and E). It can tolerate high levels of input noise (fig. S6F) and operate over a wide range of processing layer sizes (fig. S6D). Furthermore, the learning mechanism can be implemented with various functional forms of the internal feedback signal, including a neural implementation in which the feedback is driven by output spikes of a neural supervisor (fig. S6G). I have ensured that self-supervised learning was unaffected when synaptic plasticity in the processing layer used the approximate, correlation-based learning rule (fig. S3C). Last, self-supervised learning functioned analogously when the unknown features were not discrete but drawn from continuous feature dimensions (fig. S7).

Neural interactions mediated through internal feedback signals

An important question in unsupervised neural learning is what mechanisms of interactions can ensure the formation of neural maps in which different neurons represent different features. One intuitive and long established mechanism for competitive interactions between neurons is lateral inhibition (27). Indeed, recurrent inhibitory

connections between the processing layers of several self-supervised networks (materials and methods, lateral inhibition) ensured that each of the networks converged to a nonoverlapping region of feature space (fig. S8, A and B). This mechanism did not require fine tuning of the strength of inhibition (fig. S8B) and accommodated different numbers of participating networks by automatically increasing each network’s selectivity when the number of networks grew (fig. S8C).

Lateral inhibition shaped the neural activity profiles that emerged from the unsupervised learning dynamics by introducing a complex interference between feedforward inputs and recurrent inhibitory activity. The complexity of these highly nonlinear interactions makes it difficult (presumably also for evolutionary processes) to control the exact shape of the emerging neural tuning profiles. In addition, this tuning remains dependent on the continued presence of the recurrent inhibition even after the learning has converged. Last, because direct lateral inhibition operates on the fast time scale of the processing layer, it requires the tight temporal alignment of all sensory feature streams. This conflicts with the inherent capability of self-supervised networks to discover temporal structures that extend over the much slower time scale of the supervisory feedback circuit. These three drawbacks can be resolved by replacing the direct interactions between processing layer neurons with indirect interactions through a more refined supervisory circuit. Specifically, I assume that instead of the common supervisory signal introduced above, a group of supervisory units maps the activity of the processing layer onto an array of supervisory signals $\vec{\ell}$ whose components ℓ_i specify the individual desired spike counts of each processing neuron (Fig. 5A). In its simplest form, these labels are given by a linear projection,

$$\ell_i = \sum_{j=1}^{N_p} S_{ij} a_j \quad (1)$$

which has a straightforward neural interpretation (materials and methods, neuron specific feedback signals in self-supervised networks) and suffices to expose the basic principle underlying the proposed mechanism. Nevertheless, nonlinearities within this supervisory pathway, such as the saturation of individual feedback signals, can have important functional consequences (compare with Fig. 6). In the above equation, N_p is the number of processing layer neurons, and the component S_{ij} of the $N_p \times N_p$ supervisory matrix specifies how much the activity a_j of the j th processing neuron contributes to the supervisory signal ℓ_i of the i th processing neuron. The regulation of this coupling between the supervisory circuit and the processing layer is assumed to take place over much slower (developmental or evolutionary) time scales than that of learning in the processing layer.

In any unsupervised neural network, the shape of the emerging neural activity profiles depends on the network’s architecture. In self-supervised networks, this dependence is particularly simple:

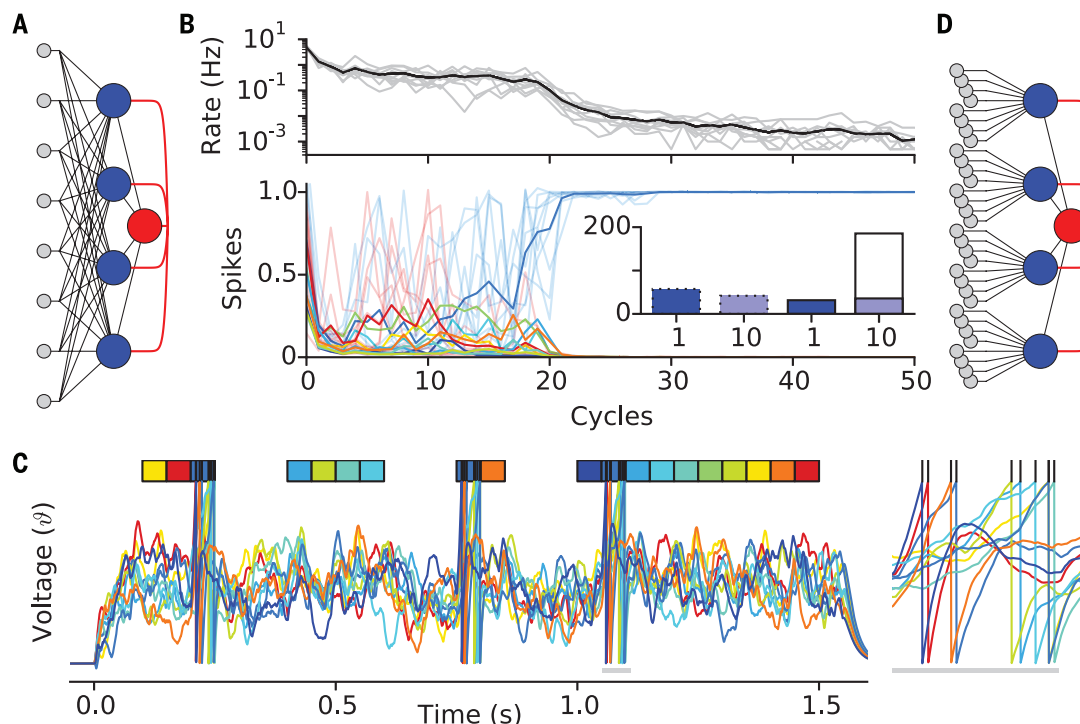


Fig. 4. Self-supervised neural networks. (A) Network scheme with shared input architecture. A layer of processing neurons (blue) receives input activity from a common input layer (gray). These multi-spike tempotrons project to a single supervisor unit (red), which feeds back a common feedback signal (red lines) to each neuron of the processing layer. (B) Learning curves of a shared input network showing the evolution of the processing-layer responses to the background activity (top, black line) and to each of the individual features [bottom; color code as in (C)]. Full colors denote the mean responses over all 10 processing-layer neurons of the network, and light tones depict responses of individual cells. (Inset) The mean convergence times (over 1000 independent

simulations) of shared [(A), dotted lines] and individual [(D), solid lines] input architectures for tasks with 1 (dark blue) and 10 (light blue) features. The open bar represents tasks in which the independent features of each processing-layer neuron were not ordered with respect to their spike counts (supplementary text S2) (C) Example voltage traces of all 10 processing-layer neurons (color coded) at the end (500 cycles) of the simulation shown in (B). Although individual voltage traces have remained diverse, all neurons have learned to respond the dark-blue feature. (Right) The 10 traces within one of the feature occurrences (gray horizontal line). (D) Network scheme as in (A), but with individual input architecture.

The learning dynamics in self-supervised networks reaches a fixed point when all supervisory signals match the activities of the recipient processing layer neurons so that all discrepancies between feedback signals and neural responses vanish. This is the case when the supervisory circuit projects the processing layer activities onto themselves—when, mathematically, \vec{a} is an eigenvector of the supervisory matrix S with eigenvalue 1. This observation immediately explains the behavior of the self-supervised networks with a single supervisor that corresponds to a uniform supervisory matrix with constant $S_{ij} = 1/N_p$. With uniform supervision, the only eigenvector of S with nonzero eigenvalue is also uniform, so that convergence requires all neurons to respond with the same number of spikes to each input pattern. Although uniform supervision offers the simplest neural implementation, neural circuits can use multiple supervisor units as an efficient mechanism to stabilize more complex neural activity profiles across feedforward sensory processing stages. In fact, with appropriate supervisory connections the learned neural responses to a discovered sensory feature can be shaped almost arbitrarily (materials and methods, supervisory matrices, neural population tuning for discrete features) (Fig. S5B and fig. S9, A

and B). When exposed to sensory features from a continuous dimension, self-supervised networks with saturating supervisory signals (materials and methods, continuous feature maps) are capable of generating continuous sensory feature maps without requiring lateral inhibition or temporal alignment of incoming sensory features (materials and methods, continuous feature maps) (Fig. 6 and fig. S9C).

Discussion

I have demonstrated that aggregate-label learning enables simple neural architectures to solve the temporal credit-assignment problem underlying learning tasks with delayed feedback. This approach remains robust even when input signals are corrupted by noise or when the feedback is stochastic and binary. Aggregate-label learning outperforms reinforcement learning and can solve connected-speech recognition tasks without temporally annotated training data. Although organisms rarely have access to the timing of an unknown clue, a clue's number of occurrences is often related to the strength or likelihood of the feedback signal that it predicts. Hence, the concept applies to a large class of ethological learning scenarios. The learning requires neural representations of the discrepancy between ag-

gregate rewards and the integrated neural responses. The existence of such reward prediction errors, such as within responses of midbrain dopamine neurons, has long been established (2, 29). I have shown that a simple binary readout (increase versus decrease) of such error signals (2, 29, 30) is sufficient to drive aggregate-label learning. At the core of aggregate-label learning lies the challenge to credit the right synapses when a neuron's discrete number of output spikes requires adjustment. The gradient-based multi-spike tempotron learning rule establishes a normative solution of this fundamental problem. Thus, it provides an important frame of reference for experimental measurements of synaptic plasticity. Its essential capabilities can be reproduced by an approximation (materials and methods, correlation-based learning) (fig. S3) that is based on temporal correlations between presynaptic activity and postsynaptic depolarization (12, 31, 32). Although performing well on easier tasks, this simple approximation becomes unstable for difficult tasks in which it requires early stopping or stabilization—for instance, through divisive weight normalization (fig. S3B). I expect that refinements, such as nonlinear eligibilities or the incorporation of corrections from elicited postsynaptic spikes (33), will further improve the performance and stability

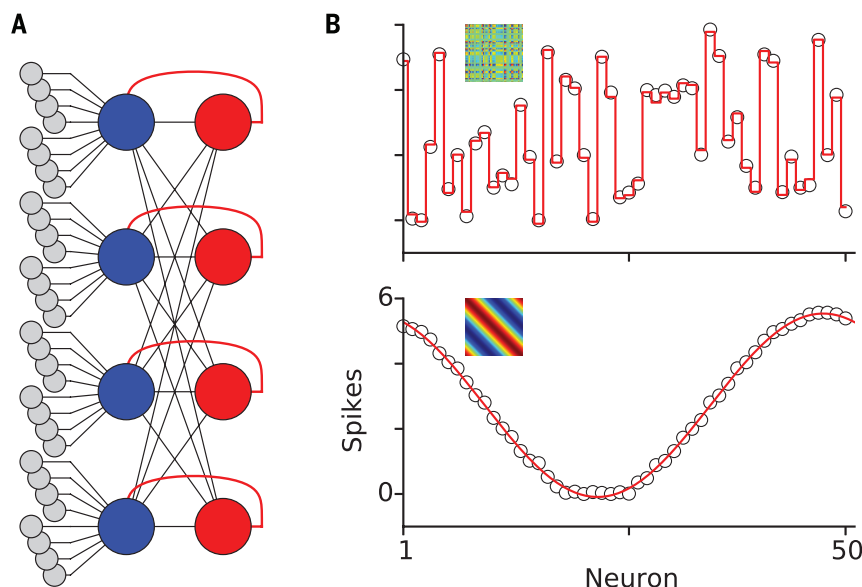


Fig. 5. Feedback-mediated neural interactions. (A) Network scheme (Fig. 4D) of a generalized self-supervised network in which each processing layer neuron receives an individual feedback signal (red lines) from a dedicated supervisor unit (red circles). (B) Learned neural responses (open circles, y axis) of 50 processing layer neurons (x axis) in a single feature task ($N_f = 1$). Responses were measured after 200 cycles of unsupervised learning (materials and methods, neural population tuning for discrete features) with two different supervisory matrices. (Top) The supervisory matrix (inset) stabilized a generic population tuning profile v whose individual components were drawn from a uniform distribution (materials and methods, supervisory matrices). The red line shows a two-parameter (baseline and scale) fit of v with $R^2 = 0.9973$. (Bottom) The circulant supervisory matrix (inset) stabilized a sinusoidal neural tuning profile. The red line shows a first-order harmonic fit (baseline, amplitude, and phase) with $R^2 = 0.9987$. Background intervals were $T_\phi = 2$ s long, and mean feature counts were $c_f^{\text{mean}} = 25$.

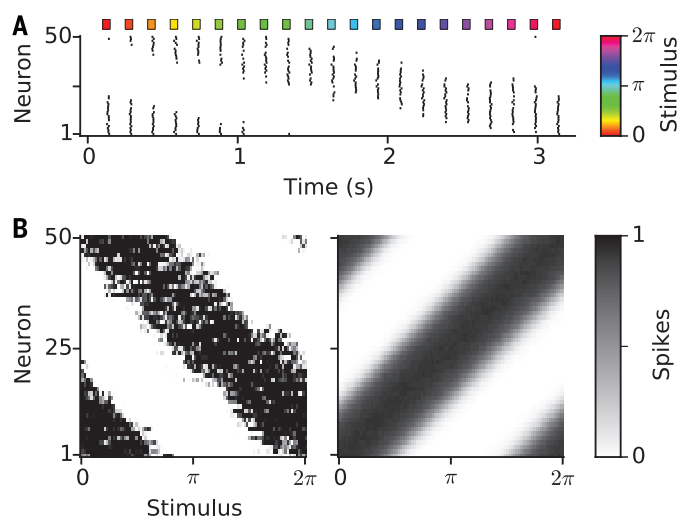


Fig. 6. Emergence of continuous feature maps. (A and B) Learned neural responses when the generalized self-supervised network from Fig. 5B, bottom, was driven with features from a continuous dimension (materials and methods, continuous feature dimensions). The emergence of these continuous maps required the supervisory signals to be saturating (materials and methods, continuous feature maps) to enforce an even distribution of neural responses across the processing layer. (A) Example output spike raster of a processing layer (after learning) when driven by a probe trial containing 21 features (colored rectangles) embedded within background activity at evenly spaced times (x axis). The features evenly span the feature dimension (color bar). (B) Mean neural responses (gray scale) of all 50 processing-layer neurons (y axis) to feature from the continuous dimension (x axis) for a single network (left) and the mean map (right) over 1000 independent simulations. Phases and directions of individual maps (fig. S9C) were aligned before averaging. Background intervals were $T_\phi = 1$ s long, and mean feature counts were $c_f^{\text{mean}} = 2$.

of neural implementations of the multi-spike tempotron. I have focused on the computationally hardest scenario and applied aggregate-label learning to single-step prediction problems (1), within which no feedback at all is available until the end of a trial. Extensions of the concept to multistep prediction problems—within which some feedback can also arrive throughout a trial, such as in the context of temporal difference learning (1)—will further boost learning performance. These extensions will need to elaborate how the neural feedback circuit should segment the learning in time when consecutive feedback signals become available. One possibility is that the circuit initiates a learning step whenever the discrepancy between the available external feedback and its internal prediction reaches a critical value. The neural implementation of aggregate-label learning can be interpreted as a continuous (temporal) form of multiple-instance learning [reviewed in (34)], establishing an unexplored cross-link between neurobiological and machine learning.

The concept of aggregate labels gives rise to a form of unsupervised learning that enables populations of neurons to discover reoccurring constellations of input patterns even if they are rare and distributed over spatial and temporal scales that exceed the receptive fields of single neurons. This detection of global structure through the amplification of correlations is related to the long established statistical technique of canonical correlation analysis (35, 36). Implemented within an information maximization approach, the underlying principle has previously enabled abstract neural networks with analog (sigmoidal) units to detect “instantaneous” spatial structure across neural populations (36, 37). Through aggregate-label learning, this principle could be realized in the time domain and implemented within a spiking neural network. The resulting self-supervised learning is consistent with observed increases in response coherence (within local groups of neurons) accompanying the emergence of direction selectivity in early cortical development (38) and captures the process’s sensitivity to stimulus statistics (39). Self-supervised networks are robust to extensive temporal misalignments between feature occurrences across their processing layers. Overcoming the inability of direct lateral inhibition to introduce competitive interactions between neurons that are processing asynchronous features, I have demonstrated that such interactions can be straightforwardly mediated through internal supervisory signals, offering a simple yet powerful mechanism with which to mold the emerging neural maps. After convergence, the supervisory circuit can disengage without affecting the learned neural responses. By detecting common numbers of occurrence, self-supervised networks realize a general principle for invariance learning that encompasses (as a special case) the recently proposed principle of “temporal contiguity” (40, 41). Nevertheless, constraints on the time evolution of the internal supervisory signals could be considered to further extend the window of “temporal contiguity” beyond the time frame of the supervisory signaling.

The mechanisms through which self-supervision segments learning in time, such as through thresholding the total number of processing layer spikes, will be important constituents of its specific neural implementations. The structural and computational simplicity of self-supervised networks permits a number of neural implementations at different spatial scales. I have shown that the internal feedback signal can be realized by the spiking activity of a simple model neuron, demonstrating that modulatory feedback of neural origin could be provided to local as well as distributed (multimodal) processing layers. Alternatively, I speculate that a local supervisory signal could be provided by astrocytes that have been implicated in modulating synaptic plasticity and also monitor the activity levels of neighboring neural circuits (42). Last, the computational similarity between a two-layer neural network and dendritic nonlinearities that converge onto a soma (43) invites the intriguing thought that the principle of self-supervision could be used by a neuron's somatic compartment to guide its dendritic branches in a search for structure within the cell's presynaptic activity (44).

Materials and methods

Embedded features

High-dimensional spike patterns of $N = 500$ sensory afferents were modeled as Poisson point processes with an average firing rate of $r_f = 5$ Hz. Each multineural input spike pattern consisted of random background activity of duration $T_{\mathcal{O}}$, which was interleaved by a random number of feature activity patterns. Feature activity patterns were $T_f = 50$ ms long and remained fixed across all trials of a given learning task. In tasks that mimicked learning of discrete sensory features, N_f independent realizations of feature activity patterns were generated from the background firing statistics. For each trial, the number of each feature's occurrences was drawn from a Poisson distribution with mean c_f^{mean} . Thus, the average total number of features within each trial was given by $N_f c_f^{\text{mean}}$, and the mean trial duration was

$$T = T_{\mathcal{O}} + N_f c_f^{\text{mean}} T_f \quad (2)$$

For each feature occurrence of a given trial, an occurrence time was drawn from a uniform distribution over $[0, T_{\mathcal{O}}]$. After sorting all feature occurrence times, the input activity pattern was constructed by successively inserting the feature activity patterns into the background. Overlap was avoided by delaying all background spikes after a feature occurrence and all subsequent feature occurrence times by T_f whenever a feature activity pattern was inserted.

Fixed feature activity spike counts

In fig. S6, D to F, I studied the convergence times of self-supervised networks with feature activity patterns whose total spike counts were fixed instead of drawn from Poisson count statistics. Specifically, for each feature activity pattern the predetermined number of spikes was randomly distributed across all N afferents by randomly

sampling (with replacement) from a uniform distribution over $\{1, 2, \dots, N\}$.

Correlated feature occurrences

In Fig. 2C and fig. S2D, homogeneous pairwise correlations $0 \leq \rho \leq 1$ between the numbers of occurrences C_i of all features $i = 1, \dots, N_f$ were introduced through a common source Y . Specifically, on each trial each feature's number of occurrences was realized through the sum of two Poisson random variables, $C_i = Y + X_i$. With the mean of the common Y set to ρc_f^{mean} and the means of the individual X_i set to $(1 - \rho)c_f^{\text{mean}}$, the resulting C_i were Poisson distributed with mean c_f^{mean} and pairwise correlations $\text{Corr}[C_i, C_j] = \rho$ for $i \neq j$. The mutual information (MI) between the labeled numbers of occurrences \vec{C}_{ℓ} —that is, $\vec{C} = (C_1, \dots, C_{N_f})$ and the corresponding aggregate-label ℓ —and the identity of the clue is given by the reduction in the clue's uncertainty, $\text{MI} = H_{\text{clue}} - H_{\text{clue}|\vec{C}_{\ell}}$ (materials and methods: mutual information between feedback signal and clue identity).

Reinforcement learning comparison task

To compare aggregate-label learning with the performance of a recent implementation of reinforcement learning (4), I studied a feature learning task in which no background activity was present and the desired response to each feature was binary: at least one spike versus no spike. In addition, the duration $T_f = 500$ ms of the $N_f = 10$ Poissonian feature activity patterns with mean firing rates of $r_f = 6$ Hz was much longer. Each trial consisted of a random sequence of all features. To mimic the feedback delay of 1350 ms (4), each input activity pattern was partitioned into 10 segments of 1350 ms duration whose starting times were locked to the beginning of each of the 10 features. Half of the features were targets and the neuron required to respond with at least one spike within the 500 ms feature interval, whereas the other half were distractors, for which the neuron had to remain silent. After a miss, the neuron was trained to increase its number of output spikes over the preceding 1350-ms input activity pattern by one spike, and after a false alarm, the neuron was trained to decrease its number of output spikes. The convergence criterion required 90% correct feature responses (4).

Probabilistic reward contingencies

In Fig. 2, E and F, and fig. S5 I study an adapted version of the “weather prediction task” (21), in which the occurrences of sensory features increment the probability p_R that a binary reward is delivered at the end of a trial. In each trial, the reward probability was given by

$$p_R = 1 - \exp \left[\sum_{i=1}^{N_f} c_i \log(1 - \pi_i) \right] \quad (3)$$

where the sum runs over all existing sensory features and $c_i \in \mathbb{N}_0$ denotes their independent Poissonian occurrence counts (with mean $c_f^{\text{mean}} = 0.1$). The reward contingency of each feature i was controlled by the parameter $\pi_i \in [0, 1]$, which determined by how much each individual occurrence

of the feature increased the reward probability. On trials that contain only a single feature—only one of the $c_i = 1$ while the others are zero—the reward probability reduces to $p_R = \pi_i$. In fig. S5A, I test the effect of false rewards. These were generated by additional features with nonzero reward contingencies, however, which were invisible to the neuron because their occurrences did not contribute to the neuron's input spike train. In learning tasks with probabilistic reward contingencies, synaptic potentiation was invoked after each rewarded trial irrespective of the neuron's response. Specifically, the neuron performed a multi-spike tempotron learning step toward increasing the current response by one spike. In contrast, depression was dependent on the neuron's response, and a learning step toward decreasing the neuron's response by one spike was only invoked when the neuron fired at least one spike during a nonrewarded trial (false alarm). I used divisive normalization to bound the resulting learning dynamics: Whenever, after a learning step, the Euclidean norm $|\vec{w}|$ of the synaptic efficacies surpassed the fivefold multiple of its initial value $|\vec{w}_{\text{ini}}|$, all efficacies were multiplied by $5|\vec{w}_{\text{ini}}|/|\vec{w}|$.

Continuous feature dimensions

To mimic sensory features from a continuous dimension I modeled a sensory encoding stage that mapped each feature parameter $\alpha \in [0, 2\pi]$ to a feature activity pattern of $T_f = 50$ ms duration. This deterministic map was constructed by specifying how the feature spike trains of each individual afferent—the number and timing of spikes within the 50 ms feature interval—depended on the feature parameter α . For each afferent, this mapping was defined through interpolation between a set of spike train templates. The number of templates assigned to each afferent was a random number $m = 1 + \Gamma$ with Γ drawn from a Poisson distribution with mean γ . Each of these m templates was randomly assigned to a fixed position α_i^* ($i = 1, \dots, m$) within the feature dimension, drawn from a uniform distribution over $[0, 2\pi]$. Each template i stored a spike count c_i drawn from the background statistics and a vector of spike times drawn from a uniform distribution over $[0, T_f]$. The number of spike times stored in each template was equal to the maximal spike count of the afferent over all of its m templates. For any feature parameter $\alpha \in [0, 2\pi]$, a vector of spike times was given by linear interpolation between the two spike time vectors of the adjacent templates k and $k + 1$ with $\alpha_k^* < \alpha < \alpha_{k+1}^*$. For afferents with only one spike train template, $m = 1$, the spike train remained unchanged for all α and did not carry any information about the feature parameter. If the two adjacent templates had the same spike counts—if $c_k = c_{k+1}$ — α was mapped to a spike train of c_k spikes that occurred at the first c_k components of the interpolated spike time vector. If the spike counts were different, $\Delta c_k = |c_k - c_{k+1}| > 0$, an additional set of transition values α_j^* ($j = 1, \dots, \Delta c_k$) from a uniform distribution over $[\alpha_k^*, \alpha_{k+1}^*]$ was stored with the templates. These transition values determined the positions of incremental changes

in the number of spikes as α transitioned through $\alpha_k^* < \alpha_1^* < \alpha_2^* < \dots < \alpha_{\Delta k}^* < \alpha_{k+1}^*$. The appearance and disappearance of spikes along the feature dimension was ordered according to a “last-in-first-out” principle, so that the most recently added spike was always the first to disappear when the spike count began to decrease. The only internal parameter of the above encoding stage is γ , which controls the density of independent spike train templates along the feature dimension and thus determines how quickly the activity patterns of two feature parameters become uncorrelated as their distance along the dimension grows (fig. S4A). Except in fig. S4A where I varied γ , I used $\gamma = 5$ throughout the present work.

Encoding schemes

In fig. S4, D to F, I compare the learning performance for different encoding stages. In addition to the above encoding in which both spike counts and spike times carry information about the feature parameter α , I implemented three additional alternatives: In a count-based encoding, the feature parameter α only determined the deterministic number of spikes that each afferent contributed to the feature activity pattern, but their timing was drawn independently for each feature realization. In a variation of this scheme, these deterministic spike counts were interpreted as the mean values of random Poisson spike count variables that were redrawn for each realization of a feature activity pattern and the corresponding number of spikes placed randomly within the feature interval. In contrast, the spike-timing-based encoding stage was constrained so that the spike counts of all spike train templates of a given afferent remained constant along the entire feature dimension. Here, only the relative timing of spikes within the feature activity pattern, but not their numbers, carried information about the feature parameter.

Continuous tuning curves

Random periodic target tuning curves of the form

$$f_O(\alpha) = A_0 + \beta \sum_{k=1}^O B_k \cos(k\alpha + \phi_k) \quad (4)$$

were generated by drawing the amplitudes B_k from a uniform distribution over $[0, 1]$ and the phases ϕ_k from a uniform distribution over $[0, 2\pi]$. The parameter $O \in \mathbb{N}$ determined the order of the highest nonzero Fourier coefficient in the above expansion. For each realization, the baseline A_0 and the global scale β were set so that the minimum and maximum of f_O were $f_O^{\min} = 0$ and $f_O^{\max} = 10$, respectively. In fig. S4C, I used the gray-level values of Leonardo da Vinci's Mona Lisa as target tuning curves. The image data was based on the file *Mona_Lisa_by_Leonardo_da_Vinci_from_C2RMF_retouched.jpg* downloaded from <http://commons.wikimedia.org/wiki> in a resolution of 687 by 1024 pixels. A 100- by 100-pixel region was extracted, converted to gray-scale, and rescaled so that the minimum value was 0 and the maximum 10. Each target tuning function corresponded to a row of the resulting

matrix. During training, feature parameters were rounded to a discrete grid that matched the column resolution of the image.

Input noise

I used two types of input noise: First, random deletion of input spikes that mimicked synaptic transmission failures (Fig. 2, E and F, and figs. S4F, S5, and S6F), and second, presynaptic spike time jitter (Fig. 3 and fig. S4E). Spike deletion was parameterized through a probability p_{del} , which determined for each spike of an input activity pattern (including background spikes) the independent probability that it was not transmitted to the postsynaptic compartment. Spike jitter was realized by adding independent Gaussian noise with zero mean and standard deviation σ to each spike time of the input activity. Neural responses were measured with the same noise settings that were used during training.

Neuron model

The multi-spike tempotron was implemented as current-based leaky integrate-and-fire neuron model with reset. Specifically, the neuron's postsynaptic membrane potential $V(t)$ was given by integration of exponentially decaying currents from $N = 500$ synaptic afferents, yielding

$$V(t) = V_{\text{rest}} + \sum_{i=1}^N \omega_i \sum_{t_i^j < t} K(t - t_i^j) - \vartheta \sum_{t_{\text{spike}}^j < t} \exp\left(-\frac{t - t_{\text{spike}}^j}{\tau_m}\right) \quad (5)$$

Here, t_i^j denote the arrival times of individual presynaptic spikes of the i th afferent, and t_{spike}^j are the times of output spikes elicited in the postsynaptic compartment. Each input spike at time t_i^j contributes a postsynaptic potential (PSP), whose shape is given by the kernel $K(t - t_i^j) = V_{\text{norm}} \left[\exp\left(-\frac{t - t_i^j}{\tau_m}\right) - \exp\left(-\frac{t - t_i^j}{\tau_s}\right) \right]$ and whose peak amplitude is determined by the afferent's individual synaptic efficacy ω_i . Normalization of K to unit amplitude resulted from setting $V_{\text{norm}} = \eta^{[\eta/(\eta - 1)]}/(\eta - 1)$ with $\eta = \tau_m/\tau_s$. The temporal shape of PSPs is governed by the integration time constant of the postsynaptic membrane, $\tau_m = 20$ ms, and the decay time constant of synaptic currents, $\tau_s = 5$ ms. PSPs are causal; $K(t - t_i^j)$ is defined to vanish for $t < t_i^j$. Whenever $V(t)$ crosses the firing threshold ϑ , the neuron emits an output spike, and the voltage resets to $V_{\text{rest}} = 0$ by the last term in Eq. 5. I used $\vartheta = 1$ for the fixed biological firing threshold of the neuron. Numerical simulations of the neuron model were event-driven and based on exact solutions of Eq. 5. In simulations underlying table S2, I also tested readout neurons with $N = 80$ and $N = 20$ synaptic afferents.

Spike-threshold-surface

To formulate a continuous objective function for a neuron's discrete number of spikes, I defined, for a given input spike pattern, the spike-threshold-surface (STS) of a neuron as the function $\text{STS} :$

$\mathbb{R}^+ \rightarrow \mathbb{N}_0$, which maps each threshold value ϑ to the number of elicited output spikes, $\vartheta \mapsto \text{STS}(\vartheta)$ (fig. SIC). The STS can be characterized by the decreasing sequence of critical threshold values ϑ_k^* :

$$\vartheta_k^* = \sup\{\vartheta \in \mathbb{R}^+ : \text{STS}(\vartheta) = k\}, k \in \mathbb{N} \quad (6)$$

The critical threshold ϑ_k^* denotes the threshold value (supremum) at which the number of output spikes jumps from $k - 1$ to k . For thresholds between two critical values, the STS is constant, $\text{STS}(\vartheta_{k+1}^* < \vartheta < \vartheta_k^*) = k$. Because a neuron remains silent if its firing threshold lies above the maximum postsynaptic voltage V_{max} , the STS is zero for thresholds above V_{max} , $\text{STS}(\vartheta > V_{\text{max}}) = 0$. The first output spike is generated when the threshold equals to V_{max} , $\vartheta_1^* = V_{\text{max}}$. In general, each ϑ_k^* corresponds to a voltage value described by Eq. 5 and hence is a function of the neurons synaptic efficacies ω and differentiable with respect to them. When I use the STS for the definition of gradient-based synaptic learning rules, I ignore the possible existence of singular points in the space of synaptic efficacies where a required ϑ_k^* and/or its derivatives might not exist—for example, when two adjacent critical thresholds coalesce and the output spike count of a neuron jumps by two spikes at a single critical value. I have not encountered such pathologies in any of the simulations underlying the present work.

Multi-spike tempotron

The goal of multi-spike tempotron learning is to modify a neuron's synaptic efficacies so that it fires a desired number of output spikes in response to a given input spike pattern. A simplified version of this problem has been solved with the binary tempotron (12), which is a supervised synaptic learning rule that can train a neuron to respond to a given input spike pattern by either firing at least one spike or remaining silent. Because of the original tempotron's binary decision, the learning could be formulated as a gradient-based learning rule operating on the maximum postsynaptic potential: increasing it toward the cell's firing threshold when the neuron should fire at least one spike, and lowering it below the firing threshold when the neuron should remain silent. However, this reliance on the maximal postsynaptic potential as its objective function prevents the binary tempotron learning rule from controlling the number of spikes beyond one, when the nonlinearity of spike generation renders the maximum postsynaptic potential insensitive to the number of elicited spikes. To overcome this problem, the multi-spike tempotron implements a gradient-based learning rule whose objective is to shape the neuron's STS by modifying the positions of the critical thresholds ϑ_k^* (Eq. 6). Although many objective functions can be defined on the basis of the STS, I focus here on the simplest form in which each learning step operates along the gradient of only a single ϑ_k^* with $k \in \mathbb{N}$; that is, the neuron's synaptic efficacies are changed by $\Delta \omega \propto \nabla_{\omega} \vartheta_k^*$ after each error trial.

To generate the desired number of output spikes d in response to a given input spike pattern,

a neuron's STS must be aligned with its firing threshold ϑ so that $\vartheta_{d+1}^* < \vartheta < \vartheta_d^*$. If this condition is violated on a given trial—if the neuron misclassifies its input spike pattern—an “absolute” learning rule would decrease ϑ_{d+1}^* through $\bar{\Delta}\omega \propto -\nabla_{\omega} \vartheta_{d+1}^*$ whenever the neuron fired more spikes than required. If the neuron had fired less spikes than desired, absolute learning would increase ϑ_d^* through $\bar{\Delta}\omega \propto \nabla_{\omega} \vartheta_d^*$. Such absolute learning (fig. S2C) would require the supervisory signal to provide the exact value of desired output spike count. In contrast, a simpler, biologically more plausible, and from an incremental learning perspective perhaps also more intuitive learning rule is based on a binary feedback signal that only specifies whether the neuron should increase or decrease its current output spike count o :

$$\bar{\Delta}\omega = \begin{cases} -\lambda \nabla_{\omega} \vartheta_o^* & \text{if } o > d \\ \lambda \nabla_{\omega} \vartheta_{o+1}^* & \text{if } o < d \end{cases} \quad (7)$$

where the learning rate $\lambda > 0$ controls the step size of individual updates. The present work is based on this “relative” multi-spike tempotron learning rule (Eq. 7) (materials and methods, ϑ^* -gradient).

Correlation-based learning

In correlation-based learning (12), the synaptic credit assignment of the multi-spike tempotron learning rule is approximated on the basis of the correlation between presynaptic activity and postsynaptic voltage. Given the i th synapse's presynaptic spike train with spike times t_i^j , its eligibility $\mathcal{E}_i = \sum_{t_i^j} V_i^j$ was defined by the sum over all individual correlations $V_i^j = \int_{t_i^j}^{\infty} dt V(t) K(t - t_i^j)$, where $V(t)$ is the postsynaptic voltage given by Eq. 5 and K is the PSP kernel. In neurons, eligibilities of this or similar form could be realized on the basis of intracellular calcium signals, which are sensitive to the coincidence of pre- and postsynaptic activity through the voltage dependence of synaptic *N*-methyl-D-aspartate (NMDA) receptors. It is well established that the induction of long-term synaptic changes requires these calcium signals to reach specific plasticity induction thresholds (31, 45, 46). Correspondingly, I defined correlation-based learning so that all synapses whose eligibility reached a plasticity induction threshold ψ on a given error trial became eligible to undergo potentiation or depression. To ensure sufficient specificity of this learning, the induction threshold should be adjusted so that on average, only a limited fraction of synapses undergo change. In the present work, I have simplified this adjustment of ψ by automatically selecting the 10% most eligible synapses on each error trial. Specifically, all synapses whose V_i exceeded the 9th, (the highest) decile D_9 were updated:

$$\Delta\omega_i^{\pm} = \begin{cases} \pm\lambda_V & \text{if } \mathcal{E}_i > D_9 \\ 0 & \text{if } \mathcal{E}_i \leq D_9 \end{cases} \quad (8)$$

Positive updates applied to trials in which the neuron fired less than the desired number of

spikes, and negative updates applied to trials in which the neuron fired too many.

Noninteger labels in supervised learning tasks

In tasks involving supervised learning of continuous tuning curves (fig. S4), aggregate-labels were not restricted to integer values (supplementary materials, equation 41) and could not be directly interpreted as the desired number of output spikes d . Although technically the relative learning rule defined in Eq. 7 does not require d to be an integer, I probabilistically mapped noninteger labels ℓ to integer values d , being either the nearest lower L^- or higher L^+ integer adjacent to ℓ . Specifically, on each trial ℓ was mapped to $d = L^-$ with probability $p = L^+ - \ell$ and to $d = L^+$ with probability $1 - p = \ell - L^-$. As a result, the average integer label $\langle d \rangle = pL^- + (1 - p)L^+ = (L^+ - \ell)L^- + (\ell - L^-)L^+ = \ell$ matched the noninteger aggregate.

Learning rate and annealing

In simulations of the multi-spike tempotron, I used a learning rate of $\lambda = 10^{-5}$ (Eq. 7) in all supervised learning tasks with discrete target features. This value resulted in near optimal convergence times in the single-clue detection tasks, but the minimum was shallow (fig. S2C, inset). In supervised learning tasks of continuous tuning curves (fig. S4), learning convergence was improved by the use of a simple annealing schedule: If the fraction of error trials within the past 2500 trials failed to decrease relative to the preceding 2500 trial interval, the learning rate λ was reduced by a multiplicative factor of 0.9. Correspondingly, a higher initial learning rate of $\lambda = 10^{-4}$ was used in these simulations. In all self-supervised network simulations with a single supervisor unit, the learning rate was $\lambda = 5 \times 10^{-4}$ and remained fixed (Fig. 4 and figs. S6, S7, and S8). In simulations of self-supervised networks with generalized supervisory architectures, I lowered the learning rate to $\lambda = 1 \times 10^{-5}$ during the critical and consolidation phases of the discrete feature learning task (Fig. 5 and fig. S9, A and B) and used $\lambda = 5 \times 10^{-5}$ when studying the formation of continuous maps (Fig. 6 and fig. S9C). For correlation-based learning (Eq. 8), I used a constant learning rate of $\lambda_V = 1 \times 10^{-5}$ for supervised learning and $\lambda_V = 5 \times 10^{-4}$ in self-supervised networks. In the simulations summarized in table S2, I used the optimized learning rates given there.

Momentum heuristic

Following previous studies (12, 23), I used a momentum heuristic (47) to accelerate convergence in all supervised learning tasks. Specifically, synaptic updates combined the learning rule correction $\bar{\Delta}\omega$ of the current error trial (Eq. 7) with a fraction of the previous synaptic change $\bar{\Delta}\omega^{\text{previous}}$, so that $\bar{\Delta}\omega^{\text{current}} = \bar{\Delta}\omega + \mu \bar{\Delta}\omega^{\text{previous}}$. If on a given error trial the individual learning rule correction of an efficacy $i \in \{1, \dots, N\}$ was zero ($\Delta\omega_i = 0$), neither the synapse nor its previous change were updated. I used $\mu = 0.99$ for the momentum parameter, implementing a decaying trace of former synaptic changes (12). The momen-

tum heuristic was not engaged in self-supervised networks, in which I set $\mu = 0$.

Initialization

Before learning, the initial synaptic efficacies $\bar{\omega}_{\text{ini}}$ of all multi-spike tempotrons were initialized so that the neurons fired with ~5 Hz when driven by the 5-Hz Poisson background activity. Specifically, after efficacies were drawn from a Gaussian distribution with zero mean and a standard deviation of 0.01, neurons were trained with 1-s-long trials of background activity with random Poisson labels with a mean of 5. During this initialization, the momentum heuristic was disengaged ($\mu_{\text{ini}} = 0$), and the learning rate was set to $\lambda_{\text{ini}} = 10^{-3}$. Initialization proceeded over independent blocks of 100 trials and terminated when the mean firing rate of the neuron over the last block exceeded 5 Hz. Subsequently, the parameters λ and μ were set to their simulation values, and previous synaptic changes (materials and methods, momentum heuristic) were initialized to zero ($\bar{\Delta}\omega^{\text{previous}} = 0$). The above initialization was important for self-supervised networks to avoid the trivial fixed point at which all processing layer neurons remain silent. When self-supervised networks were studied with noisy input spike trains (fig. S6F), the spike deletion probability p_{del} was also applied to the Poisson background activity used during the initialization, but the target spike rate remained at 5 Hz. For supervised learning tasks, initialization was not crucial and had only appreciable effects on the convergence times when convergence was fast (fig. S2C).

Neural responses

The mean response R_y^{mean} of a neuron to a feature activity pattern y during and after learning was measured through dedicated batches of probe trials during which the learning rule was not engaged. The index y refers either to a particular feature $f \in \{1, \dots, N_f\}$ from the set of discrete features or to the feature activity pattern of a particular feature parameter $\alpha \in [0, 2\pi]$ when feature activity patterns belonged to a continuous manifold. Each probe trial consisted of a $T_{\text{probe}} = 2$ s segment of background activity within which a central gap of 50 ms was inserted. For each probe trial, the output spike count of a neuron was measured twice: once giving $s_{\mathcal{O}}$, with the central gap being empty, and once giving s_y , with the feature activity pattern placed within the gap. The response R_y^{mean} was defined as the mean difference in the number of output spikes elicited in these two conditions, $R_y^{\text{mean}} = \langle s_y - s_{\mathcal{O}} \rangle$, with the average taken over 1000 independent probe trials. The embedding of the feature activity pattern within background activity ensured that the neuron's responses were measured in the same conditions as present during learning. The averaging captured the stochasticity of the neuron's responses due to the random background activity that surrounded the feature activity pattern or additional input noise, whose statistics were kept identical to the training conditions of a given task. Responses to background activity $R_{\mathcal{O}}^{\text{mean}}$ were defined as the mean spike rate elicited in the absence of the feature activity pattern, $R_{\mathcal{O}}^{\text{mean}} = \langle s_{\mathcal{O}} \rangle / T_{\text{probe}}$.

Response variability (table S1) was quantified through the corresponding standard deviations $R_y^{\text{std}} = \text{STD}[s_y - s_{\mathcal{O}}]$ and $R_{\mathcal{O}}^{\text{std}} = \text{STD}[s_{\mathcal{O}}]/T_{\text{probe}}$.

Neural tuning curves

The tuning of neurons that were exposed to features from a continuous dimension (Fig. 6 and figs. S4, S7, S8, and S9C) was characterized by measuring the neural responses to 100 probe features that were linearly spaced along the continuous feature dimension. The mean responses shown in fig. S7A are the average of these responses over the networks processing layer. Tuning width in figs. S7, B and C, and S8C were defined as the fraction of the continuous feature dimension over which the processing layer's average response exceeded 0.5 spikes per feature.

Convergence in supervised tasks

Convergence times for the supervised single-feature learning tasks in Fig. 2, A to C and E, and fig. S2C were defined as the number of learning cycles required until neural responses converged to firing one spike in response to the clue and remaining silent in response to distractors and background activity. Specifically, neural responses were evaluated after each cycle of 100 independent learning trials. Convergence required all distractor feature and background responses to satisfy $R_{\text{distractor}}^{\text{mean}} < 0.01$ and $R_{\mathcal{O}}^{\text{mean}} < 0.01/T_r$, respectively. In addition, in deterministic feedback tasks (Fig. 2, A to C, and fig. S2C) convergence required the mean discrepancy between the clue responses and the desired single spike to fall below $|\langle s_{\text{target}} - s_{\mathcal{O}} - 1 \rangle| < 0.01$. In probabilistic feedback tasks (Fig. 2E), convergence required the mean clue response to exceed one spike, $R_{\text{clue}}^{\text{mean}} > 1$. Mean convergence times report the number of cycles after which the criterion was first satisfied, averaged over 1000 independent simulations.

Convergence in unsupervised tasks

Depending on the learning task, I used different definitions of convergence for self-supervised networks. In Fig. 4B, inset, and fig. S6, C, E, and G, I studied the convergence times of networks with a fixed number of $N_p = 10$ processing layer neurons that were engaged in discrete feature detection tasks without input noise. In these simulations, I used a direct convergence criterion that required convergence of all individual processing-layer neurons. As in the supervised tasks, learning proceeded in cycles of 100 learning trials that were interleaved by measurements of the neural responses, giving the individual mean feature iR_f^{mean} ($f \in \{1, \dots, N_f\}$) and background $iR_{\mathcal{O}}^{\text{mean}}$ responses of each processing layer neuron $i \in \{1, \dots, N_p\}$ (materials and methods, neural responses). On the basis of these, I defined the population feature responses \mathcal{R}_f as the average feature responses across the processing layer $\mathcal{R}_f = \sum_{i=1}^{N_p} iR_f^{\text{mean}}/N_p$ and its rounded integer value as $\mathcal{R}_f^{\text{int}} = \text{round}(\mathcal{R}_f)$. Convergence required the background responses of all individual processing neurons to fall below $iR_{\mathcal{O}}^{\text{mean}} < 0.05$ Hz, while at least one of the rounded population feature responses $\mathcal{R}_f^{\text{int}}$ had to be greater than zero.

In addition, for all feature activity patterns f , the deviations between the individual responses of each processing layer neuron and the population mean had to lie below 5%, $|iR_f^{\text{mean}} - \mathcal{R}_f| < 0.05\mathcal{R}_f^{\text{int}}$, when $\mathcal{R}_f^{\text{int}} \geq 1$ or $|iR_f^{\text{mean}} - \mathcal{R}_f| < 0.05$ when $\mathcal{R}_f^{\text{int}} = 0$.

When comparing convergence times of networks with varying processing-layer sizes and also different levels of input noise, the above convergence criterion was not suitable. Instead, successful detection of the single feature in the tasks of fig. S6F was based on a population criterion that required the total number of spikes elicited within the processing layer in response to the feature activity pattern $S_f = \sum_{i=1}^{N_p} (i s_{\text{f}} - i s_{\mathcal{O}})$ to exceed the population response to the 2-s background activity interval $S_{\mathcal{O}} = \sum_{i=1}^{N_p} i s_{\mathcal{O}}$ of the probe trials. Specifically, I required that over 1000 independent probe trials, the minimum of S_f was greater than the maximum of $S_{\mathcal{O}}$. The comparison of the above two convergence measures in the absence of noise is shown in fig. S6D. Although both measures give similar values for small populations, the strictness of the second, population-based convergence criterion does not increase with growing network size, so that convergence times do not grow.

In fig. S7, B, C, and E, I report the tuning statistics of self-supervised networks that were exposed to feature activity patterns from a continuous feature dimension. For continuous dimensions, the computational costs of the above convergence criteria that required an ongoing evaluation of the processing layers' responses were not practical. As an efficient albeit indirect alternative, I defined convergence on the basis of spike count coherence (SCC), which is the agreement between the spike counts of all processing layer neurons during successive learning cycles of 100 trials (fig. S7D, bottom). Specifically, convergence required that throughout a cycle, an average of 99% of the processing-layer neurons fired in agreement with the mode value of the processing-layer spike counts—that the peak in the processing layer's spike count histogram would on average contain over 99% of the individual neural responses. The tuning of multiple self-supervised networks with recurrent inhibition (fig. S8) was evaluated after a fixed simulation length of 1000 cycles.

Temporal alignment of sensory features

To highlight the capability of self-supervised networks to identify structure within their input activity across long temporal distances, I did not align the random occurrence times of feature activity patterns in simulations of individual input networks. However, as motivated in the text, I did enforce such alignment in simulations of networks with recurrent lateral inhibition between multiple self-supervised groups (fig. S8).

Scalar feedback signals in self-supervised networks

In self-supervised networks, all processing-layer neurons implemented the same relative multi-

spike tempotron learning rule that was used in the supervised tasks. However, in contrast to receiving an external feedback signal, self-supervised networks generated their training labels internally as a function of the spiking activity within their processing layers. After each learning trial, the components of the activity vector $\vec{a} \in \mathbb{N}_0^{N_p}$ consisted of the N_p individual spike counts a_i ($i = 1, \dots, N_p$) that were elicited in the processing layer throughout a given learning trial. In self-supervised networks with a single supervisor (Fig. 4 and figs. S3C, S6, S7, and S8), each activity vector \vec{a} was mapped to a common label ℓ through a scalar feedback function $\mathcal{L}: \mathbb{N}_0^{N_p} \rightarrow \mathbb{N}_0$ with $\vec{a} \mapsto \ell = \mathcal{L}(\vec{a})$. I defined \mathcal{L} to be the mean spike count, $\mathcal{L}(\vec{a}) = \sum_{i=1}^{N_p} a_i/N_p$, except in fig. S6G, where I also tested self-supervised learning with nonlinear feedback functions that realized the mode and the median of the processing-layer spike counts. When \mathcal{L} implemented the mean spike count, noninteger labels were rounded to the nearest integers. In fig. S6G, I also tested the learning performance under a neural implementation of the scalar feedback function \mathcal{L} , defined as the normalized spike count of a supervisor neuron. Specifically, the same integrate-and-fire model that was used for the multi-spike tempotrons of the processing layer, but with fast synaptic time constants of $\tau_s = 0.5$ ms, received excitatory projections from all processing-layer neurons. These connections had fixed strong individual efficacies of $\omega = 1.01$ so that in essence, every processing-layer spike caused a spike in the supervisor neuron. As a result, when normalized by the number of processing-layer neurons N_p , its spike count approximated the mean spike count within the processing layer.

Neuron-specific feedback signals in self-supervised networks

I generalized the concept of self-supervised neural networks by replacing the scalar feedback function \mathcal{L} by a vector valued projection $\vec{\mathcal{L}}: \mathbb{N}_0^{N_p} \rightarrow \mathbb{N}_0^{N_s}$ with $\vec{a} \mapsto \vec{\ell} = \vec{\mathcal{L}}(\vec{a})$, whose components specify an individual label for each processing-layer neuron. Having in mind a neural implementation of this feedback function through a back-projecting layer of N_s supervisor units, I considered $\vec{\mathcal{L}}$ to be a linear projection of the form $\vec{\mathcal{L}}(\vec{a}) = S\vec{a}$ (Eq. 1). In addition, all labels were rounded to the nearest non-negative integers. In the context of a neural implementation, the $N_p \times N_p$ supervisor matrix S can be thought of as the product of two sets of connections, $S = \mathcal{BF}$: A $N_s \times N_p$ matrix \mathcal{F} denoting feed-forward connections of neural activity from the processing to the supervisor layer, and a $N_p \times N_s$ matrix \mathcal{B} denoting feedback connections of supervisory signals from the supervisor units to the processing neurons. In this interpretation, the number of supervisory units required for the implementation of a particular projection S is given by its rank.

Zero labels

The learning dynamics of self-supervised networks has a trivial fixed point at which all

processing-layer neurons remain silent ($\vec{a} = 0$) for all input spike patterns. To prevent convergence to this state, I enforced that any given processing-layer neuron would refrain from learning on an individual trial when its feedback was zero.

Stability of self-supervised learning dynamics

When feedback signals were implemented as the rounded mean of the processing-layer activity, the resulting self-supervised learning dynamics was susceptible to a runaway instability in which all processing-layer responses continued to grow in response to background activity. This was in contrast to the stable learning dynamics that emerged under the nonlinear scalar feedback functions that implemented the mode or median because of the well-known robustness of these measures to outliers. Although this instability could be effectively countered by a small bias in the rounding operation that mapped the mean processing-layer spike counts to integer-valued labels, I followed an alternative approach: Whenever a neuron's output exceeded its label, the multi-spike tempotron learning step was followed by an additional attenuation of all weights by a factor of 0.995. In addition to stabilizing the learning dynamics, this attenuation step substantially accelerated the convergence of the processing-layer neurons to stop firing in response to background activity. It was implemented in all reported simulations of self-supervised networks, except in the critical and consolidation phases of Figs. 5B and 6 and fig. S9.

Lateral inhibition

Lateral inhibition between multiple self-supervised networks was introduced by recurrent inhibitory connections between all processing neurons that did not belong to the same self-supervised network. Specifically, each multi-spike tempotron in these networks received inhibitory afferents, with a fixed efficacy $\omega_l < 0$, that were driven by the output spikes of all processing-layer neurons of the other networks.

Correlations in synaptic efficacies and synaptic learning noise

In biological implementations, neurons in the processing layer would neither be identical in terms of their morphologies or biophysical properties, nor would they receive exactly identical input spike trains. However, as a worst-case scenario I also studied this limiting case in which the self-supervised learning dynamics of the shared input architectures had a pathological fixed point at which all neurons converged to identical synaptic efficacies. In this case, the learning would stop even if the processing-layer neurons had failed to lock onto a sensory feature and instead responded to the random background activity. Indeed, in the shared input scenario, correlations between the synaptic efficacies within the processing layer grew during learning (fig. S6B). Although the probability that a network would enter the pathological state be-

fore converging onto a sensory feature was limited for a broad range of task parameters that I explored in this study, the problem could be alleviated by introducing an additive Gaussian learning noise at each synaptic update. Specifically, after any learning step invoked in a processing-layer neuron, all synaptic efficacies received additive noise, so that

$$\vec{\omega} = \vec{\omega} + \vec{\xi} \quad (9)$$

with individual ξ_i drawn independently from a Gaussian distribution with zero mean and standard deviation $\sigma_{\omega} = 10^{-3}$. With such synaptic learning noise, which also highlights the robustness of the present learning scheme, neural responses remained diverse even over extended learning periods. For comparability, the synaptic update noise was also applied to individual input architectures in Fig. 4 and figs. S3C, S6 and S7, although by construction these architectures could not develop correlations between the efficacies of their processing-layer neurons (fig. S6B).

Supervisory matrices

I studied supervisor-mediated interactions between processing-layer neurons with two types of supervisory matrices S that stabilized different tuning profiles across the processing-layer population. In the first type, S corresponded to the sum of two matrices $S = S_{\text{hom}} + \lambda_{\text{inhom}} S_v$, each of which contributed one eigenvector with nonzero eigenvalue, λ_{hom} and λ_{inhom} , respectively, to S . The constant matrix S_{hom} with elements $(S_{\text{hom}})_{ij} = (1/N_p)$ was normalized by the processing-layer size N_p so that the eigenvalue of the homogeneous eigenvector became one, $\lambda_{\text{hom}} = 1$. This component of the supervisory circuit ensured that the learning dynamics preserved the mean processing layer activity. In contrast, the second contribution $S_v = \vec{v} \vec{v}^T$ with eigenvector \vec{v} is used to stabilize an arbitrary activity profile \vec{v} with zero mean. Specifically, in the simulations of Fig. 5B, top, and fig. S9, A, top, and B, top, the components of \vec{v} were drawn randomly from a uniform distribution, centralized by subtraction of their mean and scaled so that each \vec{v} had unit norm. In the second type of supervisory matrices (Figs. 5B, bottom, and 6 and figs. S9, A, bottom, B, bottom, and C, S was constructed as a circulant matrix whose only nonzero eigenvalues λ_{hom} and λ_{inhom} corresponded to the homogeneous (one eigenvector) and the first-order harmonic modes (two eigenvectors), respectively. Specifically, I set the first row of S to $S_{1,j} = [\lambda_{\text{hom}} + 2\lambda_{\text{inhom}} \cos(2\pi j/N_p)]/N_p$ where $j = 1, \dots, N_p$, which fixed the circulant matrix S .

Neural population tuning for discrete features

To demonstrate that the steady-states of the self-supervised learning dynamics can be controlled by the structure of the supervisory matrix S , I used a triphasic learning schedule that consisted of an initialization period, a critical period, and a consolidation period. Specifically, each simulation started by a brief initialization period of 50 cycles, during which the supervisory matrix

was neutral ($\lambda_{\text{hom}} = 1$ and $\lambda_{\text{inhom}} = 1$), and the processing layer ceased responding to background activity. In the following critical period, which lasted only a single cycle, the supervisory signaling amplified the desired mode of the processing layer responses by operating with $\lambda_{\text{inhom}} = 4$. During a subsequent consolidation phase, the supervisory circuit returned to $\lambda_{\text{inhom}} = 1$, and the responses stabilized. Fit parameters between the learned processing layer feature responses $iR_i^{\text{mean}} (i = 1, \dots, N_p)$ and the inhomogeneous mode are shown in fig. S9B. Specifically, each random tuning profile \vec{v} (fig. S9B, top) was fitted (least squares) to the learned response profile by optimizing a baseline b and an amplitude a parameter, minimizing $\sum_{i=1}^{N_p} [iR_i^{\text{mean}} - (b + av_i)]^2$. Because of the two first-order harmonic eigenvectors of the circulant S , harmonic fits (fig. S9B, bottom) required an additional phase parameter ϕ so that the corresponding fits minimized the discrepancies between iR_i^{mean} and $b + a \cos[2\pi(i-1)/N_p + \phi]$.

Continuous feature maps

The emergence of continuous maps required longer periods of amplifying supervisory signaling. Specifically, I used the circulant supervisory matrix (materials and methods, supervisory matrices) with a strong amplification of the first harmonic mode, whose eigenvalue was set to $\lambda_{\text{inhom}} = 20$. Task parameters were as in lateral inhibition case, $T_{\text{O}} = 1$ s and $c_i^{\text{mean}} = 2$. A saturation of each neuron's individual feedback signal at $\Delta = 2$ spikes per trial resulted in the distribution of neural activity across the processing layer. Specifically, each neuron's individual label ℓ (Eq. 1) was mapped to $\tilde{\ell}$ by the saturating nonlinearity

$$\tilde{\ell} = \Delta \left[1 - \exp(-\ell/\Delta) \right] \quad (10)$$

In addition, this saturation of neural feedback signals prevented the unbounded growth of neural responses within the processing layer due to the strongly amplifying harmonic mode. Results in Fig. 6 and fig. S9C were obtained by running the above self-supervised learning dynamics for 1000 cycles. For each simulation, the learned neural response map (Fig. 6B, left) was characterized by the dependence of the center response (across the processing layer) on the continuous stimulus parameter. Specifically, for each stimulus parameter $a_j = 2\pi j/100$ with $j = 0, \dots, 99$ of the learned response map, the center of the processing layer's response was measured by the weighted circular mean

$$\hat{m}(a_j) = \text{atan2} \left[\frac{1}{N_p} \sum_{k=1}^{N_p} k R_{a_j}^{\text{mean}} \sin \left(\frac{2\pi k}{N_p} \right), \frac{1}{N_p} \sum_{k=1}^{N_p} k R_{a_j}^{\text{mean}} \cos \left(\frac{2\pi k}{N_p} \right) \right] \quad (11)$$

and converted back to the processing layer locations (neuron id) of the center responses $\hat{a}(a_j) = N_p \hat{m}(a_j) / (2\pi)$. These center responses were fitted

by a straight line whose y -intercept (phase), slope, and coefficient of determination (R^2) values were used to characterize the learned maps over 1000 independent simulations (fig. S9C).

Auditory front-end

Following the auditory brain stem model used in (23), speech samples from the TIDIGITS database were converted to spike patterns through the following sequence of processing steps: Audio files (20 kHz sampling frequency) were converted into spectrograms by using the spectrogram function of the matplotlib python package, with a sliding window of 512 samples and a temporal resolution of 1 ms. These spectrograms were smoothed in frequency space into 16 overlapping channels between 360 and 8000 Hz. Specifically, the signal in each channel corresponded to the dot product between the spectrogram and a triangular frequency filter. Each of the 16 filters spanned three adjacent values of the logarithmic (Mel-scale) frequency list (360, 500, 658, 837, 1039, 1269, 1528, 1822, 2155, 2531, 2957, 3439, 3984, 4602, 5301, 6092, 6987, and 8000) and peaked at the center value. The resulting frequency averaged spectrogram S was normalized to its global maximum, logarithmically rescaled by $S = \log(S + \epsilon) - \log(\epsilon)$ with $\epsilon = 10^{-5}$, and again peak normalized. Last, the signal in each channel of S was smoothed in time by a Gaussian filter with a time constant of 20 ms. For spike generation, the final signals were rescaled to the interval $[0, 1]$ by subtraction of the global minimum and subsequent division by the global maximum. Spikes were generated by threshold crossings. Specifically, in each frequency band 15 threshold values (0.0625, 0.125, 0.1875, 0.25, 0.3125, 0.375, 0.4375, 0.5, 0.5625, 0.625, 0.6875, 0.75, 0.8125, 0.875, and 0.9375) triggered onset (crossings from below) and offset (crossings from above) spikes, respectively, and fed these into separate output channels. One additional threshold placed at 1 fed into a single output channel. In total, each speech sample was converted into a spike pattern over $16 \times (2 \times 15 + 1) = 496$ afferents that traced the envelope of the spectrotemporally smoothed spectrogram.

TIDIGITS-task

In Fig. 3, I used continuous digit sequences from the TIDIGITS database to train multi-spike tempotrons with aggregate-label learning to detect one out of the 11 English digit words “zero,” “oh,” “one,” “two,” “three,” “four,” “five,” “six,” “seven,” “eight,” and “nine,” which served as the clue. For each training example, the neuron’s supervisory signal consisted only of the aggregate-label, the number of times that the clue occurred within the speech sequence. The neuron’s task was to learn to fire one spike in response to each clue occurrence. I divided the speech data from all 111 male speakers (21 to 70 years of age) of the database into a training and a test set. The test set consisted of the isolated digit subset of the database and contained 22 utterances of isolated digits (two productions of each digit) from each speaker. In contrast, I defined the training set as

the remaining 55 connected digit sequences from each speaker. The lengths of these sequences were variable, comprising two, three, four, five, and seven digits (11 each). The TIDIGITS database comprises 22 distinct dialect groups (five to six speakers each), which were used separately in the present application and resulted in $22 \times 11 = 242$ individual tasks. Generalization performance was improved by training margins and spike time noise (23), which I optimized over a three-dimensional grid spanning Gaussian spike time jitters $\sigma \in \{0, 5, 10\}$ ms and independent spike threshold margins $\kappa^+ = \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The two margin parameters refer to a positive margin, $\vartheta = 1 + \kappa^+$, when at least one output spike was required, and a negative margin, $\vartheta = 1 - \kappa^-$ when the neuron should remain silent. For each of the $3 \times 6 \times 6$ different parameter combinations, the performance was measured by the mean minimal error (over 1000 learning cycles) over 10 independent simulations. The histogram in Fig. 3B reports the test errors of all 10 simulations for the best parameter combination for each digit and dialect—a total of $10 \times 11 \times 22 = 2420$ values. Accommodating the dimensions of the front-end, the number of afferents in this application of the multi-spike tempotron was set to $N = 496$. All other parameters—except the learning rate, which was increased to $\lambda = 5 \times 10^{-5}$ —remained unaltered.

ϑ^* -gradient

The multi-spike tempotron learning rule requires the evaluation of the gradient $\nabla_{\vec{\omega}} \vartheta^*$ of the appropriate critical threshold ϑ^* in the space of synaptic efficacies $\vec{\omega}$. After an error trial, ϑ^* is determined numerically in two steps: First, the ϑ^* is bracketed by interval halving until the extra spike that is elicited when the spike threshold ϑ corresponds to the lower bound can be analytically associated with a local subthreshold voltage maximum $v_{\max}(\vartheta)$ when ϑ is set to the upper bound. In the second step, ϑ^* is determined numerically (with accuracy 10^{-13}) as the root of the analytic expression for $[\vartheta - v_{\max}(\vartheta)]$. At the critical spike threshold ϑ^* , the post-synaptic voltage $V(t)$ (Eq. 5) that is elicited by the presynaptic spike pattern with spike times t_s^j is given by

$$V(t) = V_o(t) - \vartheta^* \sum_{t_s^j < t} \exp\left[-\frac{(t - t_s^j)}{\tau_m}\right] \quad (12)$$

where

$$V_o(t) = \sum_{i=1}^N \omega_i \sum_{t_s^j < t} K(t - t_s^j) \quad (13)$$

is the unreset subthreshold voltage and the sum over t_s^j in Eq. 12 runs over all output spikes elicited before time t . We assume (materials and methods, spike-threshold-surface) that for any given ϑ^* there exists a unique t^* so that

$$\begin{aligned} \vartheta^* &= V(t^*) \\ &= V_o(t^*) - \vartheta^* \sum_{j=1}^m \exp\left[-\frac{(t^* - t_s^j)}{\tau_m}\right] \end{aligned} \quad (14)$$

where m denotes the number of output spikes that the neuron fires before t^* , $t_s^j < t^*$ for $j \in \{1, \dots, m\}$. By definition of the neuron model and the critical threshold ϑ^* , all spike times t_s^j and t^* satisfy

$$\vartheta^* = V(t^*) = V(t_s^j) \quad (15)$$

so that for each afferent $i \in \{1, \dots, N\}$

$$\vartheta_i^{*'} \equiv \frac{d}{d\omega_i} \vartheta^* = \frac{d}{d\omega_i} V(t^*) = \frac{d}{d\omega_i} V(t_s^j) \quad (16)$$

where we introduced the abbreviation ϑ_i^{*} for the i th component of the wanted gradient.

Because ϑ^* depends on the synaptic efficacy ω_i also through the previous spike times $t_s^j < t^*$ (Eq. 14), its derivative with respect to ω_i is given by

$$\vartheta_i^{*'} = \frac{\partial}{\partial \omega_i} V(t^*) + \sum_{j=1}^m \frac{\partial}{\partial t_s^j} V(t^*) \frac{d}{d\omega_i} t_s^j \quad (17)$$

where we dropped the vanishing

$$\frac{\partial}{\partial t^*} V(t^*) \frac{d}{d\omega_i} t^* = 0 \quad (18)$$

The above relation holds because $V(t^*)$ is either a local maximum with $\partial V(t^*)/\partial t^* = 0$ or t^* is the time of an inhibitory input spike whose arrival time does not depend on ω_i . Similarly, for each $k \in \{1, \dots, m\}$

$$\begin{aligned} \frac{d}{d\omega_i} V(t_s^k) &= \frac{\partial}{\partial \omega_i} V(t_s^k) \\ &+ \sum_{j=1}^k \frac{\partial}{\partial t_s^j} V(t_s^k) \frac{d}{d\omega_i} t_s^j \end{aligned} \quad (19)$$

from which we obtain

$$\begin{aligned} \frac{d}{d\omega_i} t_s^k &= \frac{1}{\dot{V}(t_s^k)} \left[\vartheta_i^{*'} - \frac{\partial}{\partial \omega_i} V(t_s^k) - \sum_{j=1}^{k-1} \frac{\partial}{\partial t_s^j} V(t_s^k) \frac{d}{d\omega_i} t_s^j \right] \end{aligned} \quad (20)$$

with the time derivatives

$$\dot{V}(t_s^k) \equiv \frac{\partial}{\partial t} V(t) \Big|_{t=t_s^k} \quad (21)$$

evaluated from the left, before the spike reset.

To solve Eq. 17 for ϑ_i^{*} , we refactor the right hand side of Eq. 20 into the form

$$\frac{d}{d\omega_i} t_s^k = \frac{1}{\dot{V}(t_s^k)} [\vartheta_i^{*'} A_k + B_k] \quad (22)$$

where the coefficients A_k and B_k are given by the recursive equations

$$A_k = 1 - \sum_{j=1}^{k-1} \frac{A_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t_s^k) \quad (23)$$

and

$$B_k = -\frac{\partial}{\partial \omega_i} V(t_s^k) - \sum_{j=1}^{k-1} \frac{B_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t_s^k) \quad (24)$$

With the analogous definitions

$$A_* = 1 - \sum_{j=1}^m \frac{A_j}{\dot{V}(t_s^j)} \frac{\partial}{\partial t_s^j} V(t^*) \quad (25)$$

and

$$B_* = -\frac{\partial}{\partial \omega_i} V(t^*) - \sum_{j=1}^m \frac{B_j}{V(t_s^j)} \frac{\partial}{\partial t_s^j} V(t^*) \quad (26)$$

for t^* , insertion of Eq. 22 into Eq. 17 yields

$$\vartheta_i^* = -\frac{B_*}{A_*} \quad (27)$$

To calculate A_* and B_* , we consider all times t_x at which the voltage reaches the spike threshold, when $t_x \in \{t_s^1, t_s^2, \dots, t_s^m, t^*\}$. At these times, Eq. 12 reduces to

$$V(t_x) = \frac{V_o(t_x)}{C_{t_x}} \quad (28)$$

with

$$C_{t_x} \equiv 1 + \sum_{t_s^j < t_x} \exp\left[\frac{-(t_x - t_s^j)}{\tau_m}\right] \quad (29)$$

and gives the derivatives

$$\begin{aligned} \frac{\partial}{\partial \omega_i} V(t_x) &= \frac{1}{C_{t_x}} \frac{\partial}{\partial \omega_i} V_o(t_x) \\ &= \frac{1}{C_{t_x}} \sum_{t_s^j < t_x} K(t_x - t_s^j) \end{aligned} \quad (30)$$

$$\frac{\partial}{\partial t_s^k} V(t_x) = \frac{-V_o(t_x)}{C_{t_x}^2} \frac{\exp\left[\frac{-(t_x - t_s^k)}{\tau_m}\right]}{\tau_m} \text{ for } t_s^k < t_x \quad (31)$$

and

$$\begin{aligned} \dot{V}(t_x) &= \frac{1}{C_{t_x}^2} \left\{ C_{t_x} \frac{\partial}{\partial t_x} V_o(t_x) \right. \\ &\quad \left. + \frac{V_o(t_x)}{\tau_m} \sum_{t_s^j < t_x} \exp\left[\frac{-(t_x - t_s^j)}{\tau_m}\right] \right\} \end{aligned} \quad (32)$$

Mutual information between feedback signal and clue identity

In the discrete feature detection task, the sensory environment is composed of N_f distinct features, one of which is randomly selected as target feature $t \in \{1, \dots, N_f\}$. Hence, a priori the uncertainty about the target feature identity is given by the entropy $H_t = \ln N_f$. On a given learning trial, the occurrence counts of all sensory features $i \in \{1, \dots, N_f\}$ are realizations c_i of Poisson random variables C_i each with mean rate c_i^{mean} . Homogeneous pairwise correlations of strength $\rho \in [0, 1]$ between these occurrences counts are introduced by implementing each individual feature count $C_i = Y + X_i$ as the sum of a common count variable Y and a feature specific variable X_i , with mean rates ρc_i^{mean} and $(1 - \rho) c_i^{\text{mean}}$, respectively. On each learning trial, information about the target feature identity is provided by a label ℓ that specifies the occurrence count of the target feature, $\ell = c_\ell$. To quantify the amount of information that the resulting labeled vector of occurrence counts $\vec{C}_\ell = (c_1, c_2, \dots, c_{N_f}; \ell)$ provides about the target feature identity, we calculate the mutual information

$$I_{t, \vec{C}_\ell} = H_t - H_{t|\vec{C}_\ell} = \ln N_f - H_{t|\vec{C}_\ell} \quad (33)$$

where the conditional entropy

$$H_{t|\vec{C}_\ell} = \sum_{\vec{c}} p(\vec{c}) H_{t|\vec{C}_\ell = \vec{c}} \quad (34)$$

quantifies the average uncertainty in the target feature identity after a labeled occurrence count vector has been provided.

For a specific realization of the labeled occurrence count vector \vec{c}_ℓ , the entropy $H_{t|\vec{C}_\ell = \vec{c}_\ell} = \ln m$ only depends on the number of features m whose occurrence counts are equal to the label. Hence,

$$H_{t|\vec{c}} = \sum_{\vec{c}} p(\vec{c}) H_{t|\vec{C}_\ell = \vec{c}} = \sum_{m=1}^{N_f} p(m) \ln m \quad (35)$$

where $p(m)$ can be written as the total probability

$$p(m) = \sum_{k=0}^m p(X_t = k) p(m|X_t = k) \quad (36)$$

Because all individual feature counts X_i for $i \in \{1, \dots, N_f\}$ are *iid*, we can define

$$\begin{aligned} p_k &\equiv p(X_i = k) \\ &= \frac{[(1 - \rho) c_i^{\text{mean}}]^k \exp[-(1 - \rho) c_i^{\text{mean}}]}{k!} \end{aligned} \quad (37)$$

and obtain

$$p(m|X_t = k) = \binom{N_f - 1}{m - 1} p_k^{m-1} (1 - p_k)^{N_f - m} \quad (38)$$

so that

$$p(m) = \binom{N_f - 1}{m - 1} \sum_{k=0}^m p_k^m (1 - p_k)^{N_f - m} \quad (39)$$

and

$$\begin{aligned} I_{t, \vec{C}_\ell} &= \ln N_f - \sum_{m=1}^{N_f} \ln m \binom{N_f - 1}{m - 1} \times \\ &\quad \sum_{k=0}^m p_k^m (1 - p_k)^{N_f - m} \end{aligned} \quad (40)$$

In Fig. 2C, I use the normalized mutual information, $I_{t, \vec{C}_\ell} = I_{t, \vec{C}_\ell} / \ln N_f$, which measures the fraction of bits of the target feature uncertainty that can on average be inferred from a given trial. The pairwise correlations ρ only enter the mutual information through the probabilities p_k of the individual count variables X_i .

REFERENCES AND NOTES

- R. S. Sutton, Learning to predict by the methods of temporal differences. *Mach. Learn.* **3**, 9 (1988).
- W. Schultz, P. Dayan, P. R. Montague, A neural substrate of prediction and reward. *Science* **275**, 1593–1599 (1997). doi: [10.1126/science.275.5306.1593](https://doi.org/10.1126/science.275.5306.1593); pmid: [9054347](https://pubmed.ncbi.nlm.nih.gov/9054347/)
- E. M. Izhikevich, Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cereb. Cortex* **17**, 2443–2452 (2007). doi: [10.1093/cercor/bhl152](https://doi.org/10.1093/cercor/bhl152); pmid: [17220510](https://pubmed.ncbi.nlm.nih.gov/17220510/)
- J. Friedrich, R. Urbanczik, W. Senn, Spatio-temporal credit assignment in neuronal population learning. *PLOS Comput. Biol.* **7**, e1002092 (2011). doi: [10.1371/journal.pcbi.1002092](https://doi.org/10.1371/journal.pcbi.1002092); pmid: [21738460](https://pubmed.ncbi.nlm.nih.gov/21738460/)
- F. Ponulak, A. Kasinski, Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting. *Neural Comput.* **22**, 467–510 (2010). doi: [10.1162/neco.2009.11.08.901](https://doi.org/10.1162/neco.2009.11.08.901); pmid: [19842989](https://pubmed.ncbi.nlm.nih.gov/19842989/)

- R. V. Florian, The chronotron: A neuron that learns to fire temporally precise spike patterns. *PLOS ONE* **7**, e40233 (2012). doi: [10.1371/journal.pone.0040233](https://doi.org/10.1371/journal.pone.0040233); pmid: [22879876](https://pubmed.ncbi.nlm.nih.gov/22879876/)
- Y. Xu, X. Zeng, S. Zhong, A new supervised learning algorithm for spiking neurons. *Neural Comput.* **25**, 1472–1511 (2013). doi: [10.1162/NECO_a.00450](https://doi.org/10.1162/NECO_a.00450); pmid: [23517101](https://pubmed.ncbi.nlm.nih.gov/23517101/)
- R. M. Memmesheimer, R. Rubin, B. P. Olveczky, H. Sompolinsky, Learning precisely timed spikes. *Neuron* **82**, 925–938 (2014). doi: [10.1016/j.neuron.2014.03.026](https://doi.org/10.1016/j.neuron.2014.03.026); pmid: [24768299](https://pubmed.ncbi.nlm.nih.gov/24768299/)
- F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms* (Spartan Books, 1962).
- R. Güti, To spike, or when to spike? *Curr. Opin. Neurobiol.* **25**, 134–139 (2014). doi: [10.1016/j.conb.2014.01.004](https://doi.org/10.1016/j.conb.2014.01.004); pmid: [24468508](https://pubmed.ncbi.nlm.nih.gov/24468508/)
- H. S. Seung, Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron* **40**, 1063–1073 (2003). doi: [10.1016/S0896-6273\(03\)00761-X](https://doi.org/10.1016/S0896-6273(03)00761-X); pmid: [14687542](https://pubmed.ncbi.nlm.nih.gov/14687542/)
- R. Güti, H. Sompolinsky, The tempotron: A neuron that learns spike timing-based decisions. *Nat. Neurosci.* **9**, 420–428 (2006). doi: [10.1038/nn1643](https://doi.org/10.1038/nn1643); pmid: [16474393](https://pubmed.ncbi.nlm.nih.gov/16474393/)
- I. R. Fiete, M. S. Fee, H. S. Seung, Model of birdsong learning based on gradient estimation by dynamic perturbation of neural conductances. *J. Neurophysiol.* **98**, 2038–2057 (2007). doi: [10.1152/jn.01311.2006](https://doi.org/10.1152/jn.01311.2006); pmid: [17652414](https://pubmed.ncbi.nlm.nih.gov/17652414/)
- L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **77**, 257–286 (1989). doi: [10.1109/5.18626](https://doi.org/10.1109/5.18626)
- At first sight, it seems that losing the clue timing eliminates almost all the necessary feedback information, and asking for a specific number of spikes seems irrelevant because we do not really care about the total number of spikes if we only get at least one spike for each clue. However, a varying spike count per clue would indicate that the detector is operating in a regime in which noise could suppress or add spikes, which implies that the detector would not function reliably when the clues are embedded in background activity. This suggests that the number of spikes could serve as an objective function to stabilize an effective synaptic configuration and might also suffice for learning such a configuration from random initial conditions.
- Because a neuron's memory for previous inputs decays rapidly (neural membrane time constants rarely exceed a few tens of milliseconds), it has to fire rather promptly in response to a clue, if at all.
- B. Widrow, M. A. Lehr, 30 years of adaptive neural networks: Perceptron, Madaline, and backpropagation. *Proc. IEEE* **78**, 1415–1442 (1990). doi: [10.1109/5.58323](https://doi.org/10.1109/5.58323)
- D. Barber, "Learning in spiking neural assemblies" in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, K. Obermayer, Eds. (MIT Press, 2002), vol. 15, pp. 149–156.
- J. P. Pfister, T. Toyozumi, D. Barber, W. Gerstner, Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural Comput.* **18**, 1318–1348 (2006). doi: [10.1162/neco.2006.18.6.1318](https://doi.org/10.1162/neco.2006.18.6.1318); pmid: [16764506](https://pubmed.ncbi.nlm.nih.gov/16764506/)
- Although a population of 135 neurons (with 80 synapses each) required more than 3000 trials of reinforcement learning (4), the single integrate-and-fire neuron underlying the present study required an average of eight or four trials, when learning was implemented with the multi-spike tempotron or the correlation-based approximation (discussion and materials and methods, correlation-based learning), respectively. In fact, based on aggregate labels the task could be readily solved by a single neuron with only 20 synapses (table S2). Similarly strong performance differences between other neural implementations of gradient-based and reinforcement learning (11) have been reported (12) in the context of a binary classification task.
- T. Yang, M. N. Shadlen, Probabilistic reasoning by neurons. *Nature* **447**, 1075–1080 (2007). doi: [10.1038/nature05852](https://doi.org/10.1038/nature05852); pmid: [17546027](https://pubmed.ncbi.nlm.nih.gov/17546027/)
- L. F. Abbott, S. B. Nelson, Synaptic plasticity: Taming the beast. *Nat. Neurosci.* **3** (suppl.), 1178–1183 (2000). doi: [10.1038/81453](https://doi.org/10.1038/81453); pmid: [11127835](https://pubmed.ncbi.nlm.nih.gov/11127835/)
- R. Güti, H. Sompolinsky, Time-warp-invariant neuronal processing. *PLOS Biol.* **7**, e1000141 (2009). doi: [10.1371/journal.pbio.1000141](https://doi.org/10.1371/journal.pbio.1000141); pmid: [19582146](https://pubmed.ncbi.nlm.nih.gov/19582146/)
- R. G. Leonard, G. Doddington, *TIDIGITS*, vol. LDC93S10 (Linguistic Data Consortium, 1993).
- T. Masquelier, R. Guyonnet, S. J. Thorpe, Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PLOS ONE* **3**, e1377 (2008). doi: [10.1371/journal.pone.0001377](https://doi.org/10.1371/journal.pone.0001377); pmid: [18167538](https://pubmed.ncbi.nlm.nih.gov/18167538/)

26. B. Nessler, M. Pfeiffer, L. Buesing, W. Maass, Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Comput. Biol.* **9**, e1003037 (2013). pmid: [23633941](#)
27. R. Linsker, Perceptual neural organization: Some approaches based on network models and information theory. *Annu. Rev. Neurosci.* **13**, 257–281 (1990). doi: [10.1146/annurev.ne.13.030190.001353](#); pmid: [2183677](#)
28. R. Güttig, R. Aharonov, S. Rotter, H. Sompolinsky, Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity. *J. Neurosci.* **23**, 3697–3714 (2003). pmid: [12736341](#)
29. W. Schultz, Updating dopamine reward signals. *Curr. Opin. Neurobiol.* **23**, 229–238 (2013). doi: [10.1016/j.conb.2012.11.012](#); pmid: [23267662](#)
30. V. Pawlak, J. R. Wickens, A. Kirkwood, J. N. Kerr, Timing is not everything: Neuromodulation opens the STDP gate. *Front. Synaptic Neurosci.* **2**, 146 (2010). doi: [10.3389/fnsyn.2010.00146](#); pmid: [21423532](#)
31. A. Artola, S. Bröcher, W. Singer, Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature* **347**, 69–72 (1990). doi: [10.1038/347069a0](#); pmid: [1975639](#)
32. D. E. Feldman, The spike-timing dependence of plasticity. *Neuron* **75**, 556–571 (2012). doi: [10.1016/j.neuron.2012.08.001](#); pmid: [22920249](#)
33. V. Pawlak, D. S. Greenberg, H. Sprekeler, W. Gerstner, J. N. Kerr, Changing the responses of cortical neurons from sub- to suprathreshold using single spikes in vivo. *eLife* **2**, e00012 (2013). doi: [10.7554/eLife.00012](#); pmid: [23359858](#)
34. J. Amores, Multiple instance classification: Review, taxonomy and comparative study. *Artif. Intell.* **201**, 81–105 (2013). doi: [10.1016/j.artint.2013.06.003](#)
35. H. Hotelling, Relations between two sets of variates. *Biometrika* **28**, 321–377 (1936). doi: [10.1093/biomet/28.3-4.321](#)
36. S. Becker, G. E. Hinton, Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature* **355**, 161–163 (1992). doi: [10.1038/355161a0](#); pmid: [1729650](#)
37. S. Becker, Mutual information maximization: models of cortical self-organization. *Network* **7**, 7–31 (1996). doi: [10.1088/0954-898X/7/1/003](#)
38. Y. Li, S. D. Van Hooser, M. Mazurek, L. E. White, D. Fitzpatrick, Experience with moving visual stimuli drives the early development of cortical direction selectivity. *Nature* **456**, 952–956 (2008). doi: [10.1038/nature07417](#); pmid: [18946471](#)
39. S. D. Van Hooser *et al.*, Initial neighborhood biases and the quality of motion stimulation jointly influence the rapid emergence of direction preference in visual cortex. *J. Neurosci.* **32**, 7258–7266 (2012). doi: [10.1523/JNEUROSCI.0230-12.2012](#); pmid: [22623671](#)
40. L. Wiskott, T. J. Sejnowski, Slow feature analysis: Unsupervised learning of invariances. *Neural Comput.* **14**, 715–770 (2002). doi: [10.1162/089976602317318938](#); pmid: [11936959](#)
41. N. Li, J. J. DiCarlo, Unsupervised natural experience rapidly alters invariant object representation in visual cortex. *Science* **321**, 1502–1507 (2008). pmid: [18787171](#)
42. R. Min, M. Santello, T. Nevian, The computational power of astrocyte mediated synaptic plasticity. *Front. Comput. Neurosci.* **6**, 93 (2012). doi: [10.3389/fncom.2012.00093](#); pmid: [23125832](#)
43. P. Poirazi, T. Brannon, B. W. Mel, Pyramidal neuron as two-layer neural network. *Neuron* **37**, 989–999 (2003). doi: [10.1016/S0896-6273\(03\)00149-1](#); pmid: [12670427](#)
44. M. Larkum, A cellular mechanism for cortical associations: An organizing principle for the cerebral cortex. *Trends Neurosci.* **36**, 141–151 (2013). doi: [10.1016/j.tins.2012.11.006](#); pmid: [23273272](#)
45. J. A. Cummings, R. M. Mulkey, R. A. Nicoll, R. C. Malenka, Ca²⁺ signaling requirements for long-term depression in the hippocampus. *Neuron* **16**, 825–833 (1996). doi: [10.1016/S0896-6273\(00\)80102-6](#); pmid: [8608000](#)
46. R. C. Malenka, R. A. Nicoll, Long-term potentiation—A decade of progress? *Science* **285**, 1870–1874 (1999). doi: [10.1126/science.285.5435.1870](#); pmid: [10489359](#)
47. J. Hertz, A. Krough, R. G. Palmer, *Introduction to the Theory of Neural Computation* (Westview Press, 1991).

ACKNOWLEDGMENTS

It is a pleasure to thank W. Denk, D. Hansel, and P. Dayan for very insightful discussions throughout all stages of this work and L. Abbott, J. Hillmann, A. Schaefer, S. Shamma, T. Sharpee, and F. Wolf for helpful comments on this manuscript. I thank D. Fliegner and U. Haase for outstanding technical support and the Max Planck Society for making this research possible.

SUPPLEMENTARY MATERIALS

www.sciencemag.org/content/351/6277/aab4113/suppl/DC1
Materials and Methods
Supplementary Text S1 and S2
Figs. S1 to S9
Tables S1 and S2
References (48–56)

26 April 2015; accepted 20 January 2016
10.1126/science.aab4113



**Spiking neurons can discover predictive features by
aggregate-label learning**

Robert Gütiğ (March 3, 2016)

Science Translational Medicine **351** (6277), . [doi:
10.1126/science.aab4113]

Editor's Summary

Credit assignment in the brain

To discover relevant clues for survival, an organism must bridge the gap between the short time periods when a clue occurs and the potentially long waiting times after which feedback arrives. This so-called temporal credit-assignment problem is also a major challenge in machine learning. Gütiğ developed a representation of the responses of spiking neurons, whose derivative defines the direction along which a neuron's response changes most rapidly. By using a learning rule that follows this development, the temporal credit-assignment problem can be solved by training a neuron to match its number of output spikes to the number of clues. The same learning rule endows unsupervised neural networks with powerful learning capabilities.

Science, this issue p. 10.1126/science.aab4113

This copy is for your personal, non-commercial use only.

Article Tools

Visit the online version of this article to access the personalization and article tools:

<http://science.sciencemag.org/content/351/6277/aab4113>

Permissions

Obtain information about reproducing this article:

<http://www.sciencemag.org/about/permissions.dtl>

Science (print ISSN 0036-8075; online ISSN 1095-9203) is published weekly, except the last week in December, by the American Association for the Advancement of Science, 1200 New York Avenue NW, Washington, DC 20005. Copyright 2016 by the American Association for the Advancement of Science; all rights reserved. The title *Science* is a registered trademark of AAAS.