Sparse Coding with a Somato-Dendritic Rule

Damien Drix✉[1,2,3], Verena V. Hafner[2,3], and Michael Schmuker[1,3]

[1]Biocomputation group, Department of Computer Science, University of Hertfordshire, Hatfield, United Kingdom
[2]Adaptive Systems laboratory, Institut für Informatik, Humboldt-Universität zu Berlin, Berlin, Germany
[3]Bernstein Center for Computational Neuroscience, Berlin

✉ Correspondence: Damien Drix <d.drix@herts.ac.uk>

Abstract

Cortical neurons are silent most of the time. This sparse activity is energy efficient, and the resulting neural code has favourable properties for associative learning. Most neural models of sparse coding use some form of homeostasis to ensure that each neuron fires infrequently. But homeostatic plasticity acting on a fast timescale may not be biologically plausible, and could lead to catastrophic forgetting in embodied agents that learn continuously. We set out to explore whether inhibitory plasticity could play that role instead, regulating both the population sparseness and the average firing rates. We put the idea to the test in a hybrid network where rate-based dendritic compartments integrate the feedforward input, while spiking somas compete through recurrent inhibition. A somato-dendritic learning rule allows somatic inhibition to modulate nonlinear Hebbian learning in the dendrites. Trained on MNIST digits and natural images, the network discovers independent components that form a sparse encoding of the input and support linear decoding. These findings confirm that intrinsic plasticity is not strictly required for regulating sparseness: inhibitory plasticity can have the same effect, although that mechanism comes with its own stability-plasticity dilemma. Going beyond point neuron models, the network illustrates how a learning rule can make use of dendrites and compartmentalised inputs; it also suggests a functional interpretation for clustered somatic inhibition in cortical neurons.

Author Summary

Ever since the inception of neural networks in the 1950s, their engineering applications have relied on very simple artificial neurons that ignore many of the features of the cells in our brains. Research into the finer details, such as their extensive dendritic trees, was mostly the work of biologists. But the computational capabilities of dendrites are now attracting the attention of the machine learning community and there are attempts to make use of them in deep neural networks. Our work gives one example of the kind of computations that become possible once one steps beyond simple point neurons to include more biological details. Our topic is sparse coding, a field which studies how neural systems can discover structure in natural stimuli such as images. We show how adding dendrites to artificial neurons lets them solve the task in a different way. This may have benefits for creating robots that learn from experience, and suggests a number of electrophysiological experiments that could teach us more about how real neurons work.

Introduction

Activity in the brain is sparse: a given pyramidal neuron spikes infrequently (lifetime sparseness), and few neurons are active at once (population sparseness). This coding scheme is efficient in terms of energy use, and also in terms of storage capacity [1]: the same cell can participate in multiple, overlapping assemblies [2] that are active in different contexts.

Beyond energy and storage efficiency, the promise of sparse codes is that they can reveal structure in natural inputs which makes it easier to learn associative mappings — in other words, perform specific readouts [3,4]: detect a stimulus, transform a pattern of neural activity into motor commands, or prime the activation of another cell assembly.

The classical way to learn sparse codes involves the information-maximization principle of Bell & Sejnowski [5] for blind source separation. In independent component analysis [1,6,7] and sparse auto-encoders [8], the algorithm works to minimise a global cost function that includes a sparse constraint. Here, we focus on a family of single-layer networks that do not compute a global cost explicitely. Instead, these networks learn sparse codes with local learning rules thanks to the combination of two unsupervised heuristics: projection pursuit and competitive learning.

Projection pursuit looks for receptive fields with a non-Gaussian activity distribution. Diaconis & Freedman [9] note that these tend not to occur by chance,

reflecting instead some fundamental structure in the input — a characteristic that reminds us of the *suspicious coincidences* of Barlow [10].

As for competitive learning, described by Rumelhart & Zipser [11], it aims to reduce the redundancy of the code and decorrelate the output dimensions, so that each neuron responds to a different feature. This usually involves a winner-take-all system [12], or inhibitory connections between the coding neurons [13,14] — an organisation which is equivalently called lateral, recurrent or mutual inhibition.

Starting with Földiák [15], these two heuristics have been applied in a variety of sparse coding networks with rate-based [16–18] and then spiking neurons [19–22]. These networks have in common the use of Hebbian lateral inhibition to decorrelate the output, and of nonlinear Hebbian rules to perform projection pursuit on the feedforward input.

Nonlinear Hebbian learning, to follow the terminology of Brito & Gerstner [23], refers to a variant of Hebbian learning where the change of weight is proportional to the correlation between the input and a nonlinear function of the output (more precisely, of the receptive field activation). The Bienenstock–Cooper–Munro (BCM) rule [24] is an early example of such a rule, inducing depression when the output activity is below average and potentiation when it is above average. This steers gradient descent towards an activity distribution with heavy tails, which typically converges onto one of the independent components.

As noted by Brito & Gerstner [23], the precise shape of that nonlinear function is not critical. The trick is to keep it aligned with the activity distribution throughout learning, so that the potentiation region stays centered on the tail. Usually, this is done by enforcing a constant norm for the weight vectors, or by using a homeostatic term that moves the potentiation threshold according to the average activity of the neuron, as in the BCM rule. That homeostatic term has the effect to regulate the lifetime sparseness of the neuron and is also called *intrinsic plasticity* (IP) by Triesch [25], to distinguish it from synaptic plasticity.

In most models, IP needs to be faster than the Hebbian component of learning [26,27]. But in vivo, IP tends to be slower, acting over a timescale of days rather than minutes [28,29]. Besides, fast homeostasis could be particularly disruptive for animals and robots that learn continuously, and cannot assume that the feature detectors they have acquired will be stimulated at regular intervals.

Here we propose an alternative scheme that does not require fast homeostatic plasticity. The idea is to put mutual inhibition itself in control of the Hebbian nonlinearity: stimuli for which many neurons compete to respond, and neurons that are often active as well, would attract more lateral inhibition and be subject to a higher potentiation threshold. In other words, instead of using intrinsic

plasticity to enforce lifetime sparseness, this scheme would regulate both the population and the lifetime sparseness through synaptic plasticity.

To do so, we need a mechanism through which the feedforward learning rule could measure the amount of competition on an input-by-input basis and use it as a negative feedback. But artificial neural networks usually employ point neurons, where all inputs are added together into a single activity variable. The consequence is that the learning rule cannot distinguish between stronger lateral inhibition — the signal to become more selective — and weaker feedforward activity that results from synaptic plasticity or from fluctuations in the input.

The solution could be to integrate the feedforward and recurrent pathways in separate neural compartments, for instance the soma and a dendrite. The dendritic compartment could then estimate the amount of somatic inhibition by comparing its local depolarisation with the somatic activity that it perceives via backpropagating action potentials.

The idea has been tried before, although not on a sparse coding task. In Körding & König [30], lateral inhibition can prevent the backpropagating action potentials from reaching the dendrites, which induces depression in dendritic synapses via spike-timing dependent plasticity. Urbanczik & Senn [31] use probabilistic spiking neurons where the dendritic compartment tries to match the somatic potential; this results in depression when unpredicted external inputs inhibit the soma, and potentiation when these unpredicted inputs are excitatory instead.

Here we set out to investigate whether a variant of these somato-dendritic learning rules could discover sparse codes in natural stimuli. We found that one can adjust the somatic and dendritic transfer functions to produce a BCM-like curve where the threshold between depression and potentiation follows an instantaneous measure of somatic inhibition. This lets the network learn sparse codes without fast intrinsic plasticity.

Results

Network model

Our model is a network of N leaky integrate-and-fire (LIF) neurons, each with an extra dendritic compartment (fig. 1). There are two fully-connected pathways: a recurrent inhibitory pathway between the somas, and a feedforward pathway between the input and the dendrites. The network is meant to model a small patch of neural tissue where full connectivity is an acceptable approximation; hence we keep the number of neurons small ($N \leq 1024$). With respect

to the dimensionality d of our input stimuli, this translates to networks that range from undercomplete ($N/d \ll 1$) to slightly overcomplete ($N/d \approx 1.3$).

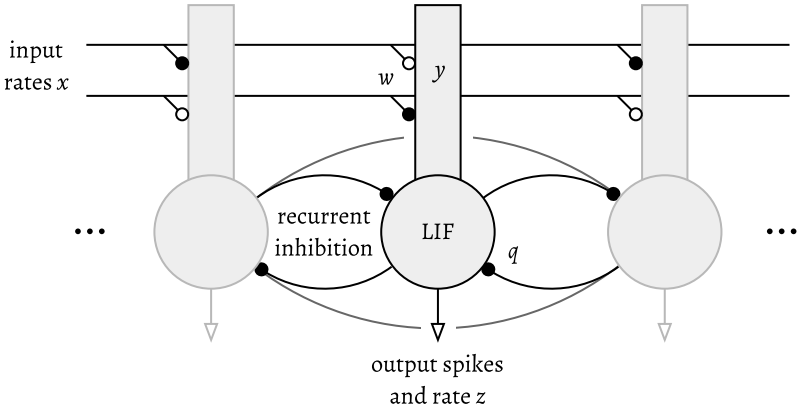

Figure 1: **Architecture of the network.** Annotations indicate the feedforward input x, leaky integrate-and-fire (LIF) somas and their firing rate z, dendritic compartments and dendritic activity y, and feedforward and recurrent pathways with weights w and q, respectively. The symbol • denotes an inhibitory synapse, ∘ an excitatory one.

The recurrent pathway mediates all-to-all inhibition via spikes and conductance-based somatic synapses. For simplicity we do not use separate inhibitory interneurons. Although that architecture deviates from biology and Dale's law, King et al. [21] found that replacing direct inhibition with interneurons did not substantially alter the results of Zylberberg et al. [20].

The feedforward pathway targets the dendrites and contains both excitatory and inhibitory synapses. It carries rates instead of spikes; doing so allows us to employ a classical Hebbian formalism in the learning rule and discrete-time dendritic compartments. A spike-based input and continuous-time dendrites would be more biologically plausible, but the model would also become substantially more complex; we reserve these for future work. Here we use a rectified linear activation function in the dendrites, with some modifications to account for the overal transfer properties of biological dendrites (see *Methods* for details).

The network operates as follows. We present each input pattern x to the dendrites and compute the dendritic activation y. This results in a constant current flow from the dendrite to the soma while the somas compete to respond for 100 timesteps ($dt = 0.5$ ms). Then we compute firing rates z using both the num-

ber of spikes and the spike latencies. Finally, we apply the feedforward and recurrent learning rules. We repeat these steps for the next input pattern, etc.

Feedforward learning rules

The weight w of each feedforward, dendritic synapse is updated according to a nonlinear Hebbian rule:

$$w \leftarrow w + \mu \left[x \left(z - \delta y \right) - \alpha y w \right] \tag{1}$$

where x is the input rate, y is the dendritic activation, z is the somatic firing rate, μ is the learning rate, α controls the scale of the weights and δ sets the potentiation/depression ratio. The rule can change the sign of the weights, switching between excitatory and inhibitory synapses. It is gated by post-synaptic activity: there is no change of weight when both z and y are zero. This ensures that the weights do not fade when the neuron is silent.

The term $x\,(z - \delta y)$ at the core of that somato-dendritic rule is reminiscent of the Delta rule [32,33], and it can be seen as a rate-based variant of the rules used in Körding & König [30] and Urbanczik & Senn [31]. In Urbanczik & Senn, the purpose of learning is to correct the mismatch between the somatic activity, z, and its prediction by the dendrite. Thus, they use an error-correcting term $z - \varphi(y)$, where φ is the dendrite's own model of the somatic transfer function and $\varphi(y) \approx z$ when the dendritic prediction is correct.

In contrast, here the goal is not to achieve a perfect prediction of the somatic activity by the dendrite, but to exploit the mismatch between z and δy so that it creates a BCM-like curve modulated by inhibition (fig. 2). In the absence of somatic inhibition, we set δ so that $z - \delta y$ is non-negative and the rule behaves like a linear Hebbian rule. In the presence of somatic inhibition, $z - \delta y$ is zero for subthreshold inputs, negative for for excitatory inputs that fail to elicit enough somatic spikes, and positive for those that produce a strong response. This yields a nonlinear Hebbian rule where the effective threshold between potentiation and depression depends on the amount of competition received for each particular input, without averaging over the recent activity of the neuron.

Equation 1 can also be understood as the sum of three terms that control long-term potentiation (LTP) and long-term depression (LTD):

$$\Delta w \propto \overbrace{xz}^{\substack{\text{homosynaptic} \\ \text{LTP}}} - \underbrace{\delta x y}_{\substack{\text{homosynaptic} \\ \text{LTD}}} - \overbrace{\alpha y w}^{\substack{\text{heterosynaptic} \\ \text{LTD}}} \tag{2}$$

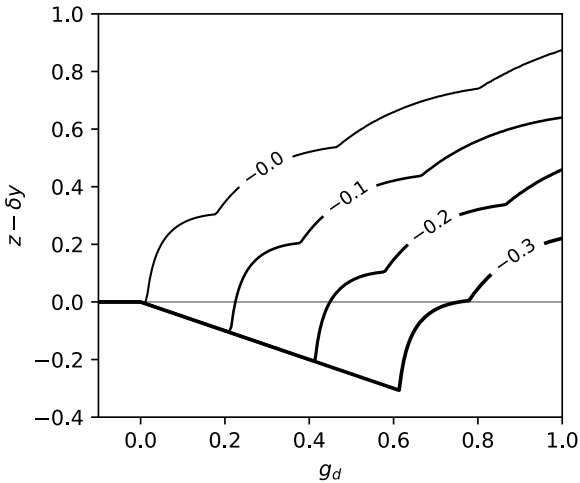

Figure 2: **The learning rule produces a BCM-like curve controlled by somatic inhibition.** Each curve plots the effective Hebbian nonlinearity $z - \delta y$ as a function of the net dendritic input g_d (assuming $w = 0$ so that we can ignore the effect of the weight decay term $-ayw$). Injecting a constant inhibitory current into the soma (marked on the curves) shifts the potentiation threshold to the right. The bumps in the curves are a consequence of the way we compute the firing rate z and mark the occurence of an extra spike. Note: this figure was generated with a finer timestep dt = 0.01 ms to smoothe the discontinuities in the curves caused by the discrete spike times.

Crucially, the two LTD terms are gated by the dendritic activity y, and will therefore not be suppressed by somatic inhibition that affects z. This allows the learning rule to depress the synapses that are active when the soma is strongly inhibited, shifting the distribution of the net dendritic input back to the left and making the neuron more selective as a result.

In contrast, Földiák [15], Zylberberg et al. [20] and King et al. [21] use a single heterosynaptic LTD term of the form $-z^p w$ that is gated by *somatic* activity and cannot induce depression in response to lateral inhibition. Instead, these networks work the other way around: they first make the output selective through IP, and then transfer that selectivity to the receptive fields by pruning the synapses that are silent when the neuron is highly active.

When $\delta > 0$ the rule is able to convert some of the recurrent inhibition into feedforward inhibition, producing receptive fields that have both ON and OFF fields even with a non-negative input. Note that homosynaptic LTP and LTD are swapped if we interpret negative weights as inhibitory synapses.

Finally, we apply a separate regularisation rule, taking care not to change the sign of the weight:

$$w \leftarrow \begin{cases} \max(0,\, w - \mu\lambda y) & \text{if } w > 0 \\ \min(0,\, w + \mu\lambda y) & \text{if } w < 0 \end{cases} \tag{3}$$

where λ determines the amount of regularisation. This does not fundamentally change the operation of the learning rule, but simplifies the receptive fields by suppressing the weights of weakly correlated input dimensions.

Recurrent learning rule

The somatic synapses that mediate lateral inhibition are plastic as well. The weight q of each recurrent, somatic synapse between a pre- and a post-synaptic neuron follows a standard Hebbian rule with pre-synaptic gating:

$$q \leftarrow q + \nu(z_{\text{pre}} z_{\text{post}} - \beta\, z_{\text{pre}} q) \tag{4}$$

where ν is the learning rate and β controls the scale of the weights. Gating by z_{pre} ensures that the inhibition from a winning neuron to a losing neuron decays, but the reciprocal connection does not. The asymmetry prevents a single neuron from taking over all the input features [34]. In practice, we use a much faster learning rate for the recurrent inhibition compared to the feedforward synapses ($\nu \gg \mu$); otherwise receptive fields are unstable and oscillate between selective and non-selective features.

Receptive fields

Our first experiment is to look at the receptive fields of the neurons after training on various types of inputs. The expectation, for a sparse coding network, is that these receptive fields should correspond to selective features (rather than complete patterns) and that the neurons should be silent most of the time.

Trained on the MNIST dataset of handwritten digits [35], the network learns receptive fields that respond to fragments of digits or pen strokes, as shown in fig. 3. These receptive fields resemble the ones learned by sparse auto-encoders [8], despite the fact that we use a different algorithm — a coincidence which can be explained if these pen-stroke shapes are indeed the independent components of MNIST digits. The activity of the network is sparse throughout the training period, both in terms of lifetime and population sparseness (figs. 4, 5).

We also test a variant of MNIST called Fashion-MNIST [36], which uses the same format but consists of small images of items of clothing like shoes and

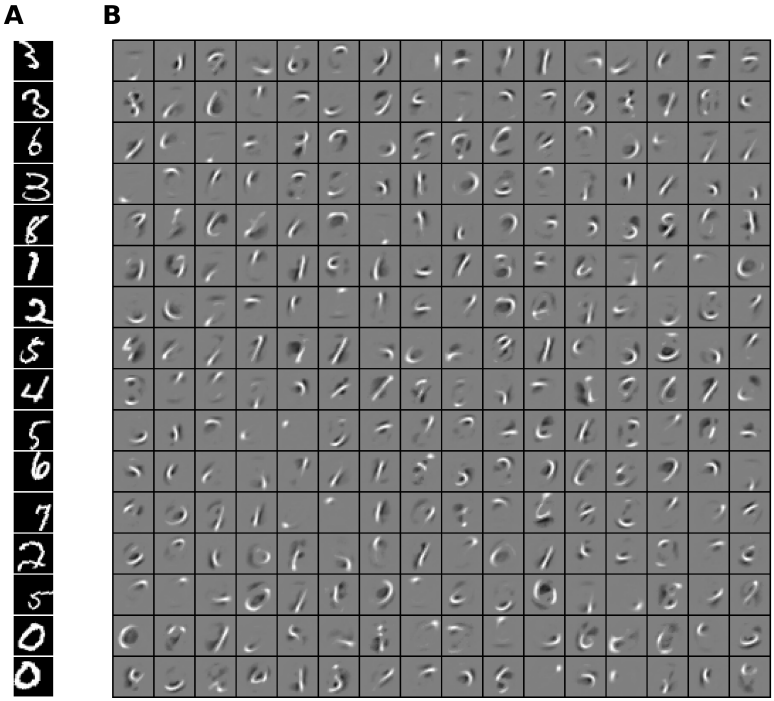

Figure 3: **The network learns pen-stroke shapes from MNIST digits. A**: sample input stimuli. Black corresponds to zero and white to one. **B**: receptive fields (weights) of a network with 256 neurons after training on 120,000 digits (28×28 pixels) with random distortions. Middle gray corresponds to zero, lighter pixels to excitatory weights, and darker pixels to inhibitory weights.

shirts. Training the network on that dataset extracts the outlines of the input stimuli and also separates some of their constituent parts (fig. 6).

We then train the network on two photographic datasets. The first one is the dataset used by Olshausen & Field [1], which consists of landscapes and close-ups of outdoors scenes. The second one is a selection of public-domain images by NASA [37] that show mostly people and indoor scenes. Sparse coding networks have often been applied to natural images [1,17,19,20], from which they learn Gabor-like filters that resemble the receptive fields of simple cells in the visual cortex [38]. Images are typically not presented to the network in their raw form, but first processed either by a *difference-of-Gaussians* filter that models the transformations happening in the retina, or by a whitening transform that equalises the variance across spatial frequencies [39]. Both types of pre-

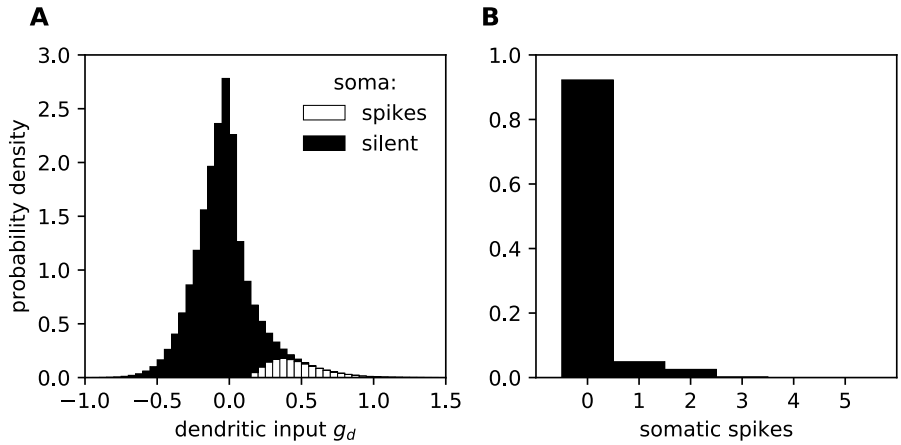

Figure 4: **Neural activity is sparse after training on the MNIST dataset. A:** the net dendritic input (receptive field activation) has a heavy-tailed distribution (excess kurtosis: 2.81, skewness: 0.9). The two conditions (spikes / no spikes) are stacked. **B:** neurons are silent most of the time, as shown by the distribution of the number of spikes per neuron per stimulus.

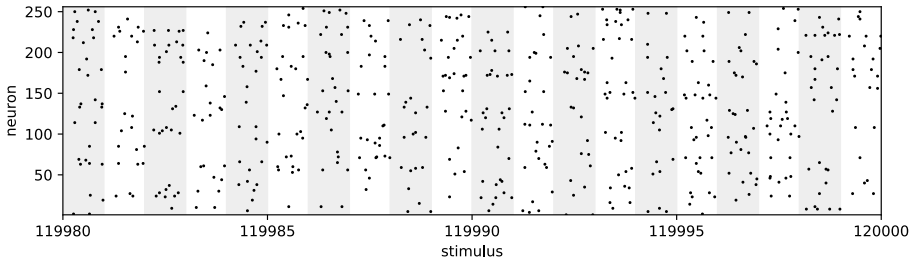

Figure 5: **Network activity is sparse: only a few spikes are emited for each stimulus.** The alternating columns in the background are 50.0 ms wide and correspond to successive MNIST patterns during a one-second period at the end of the training run.

processing have the effect to suppress low spatial frequencies and highlight edges. For this experiment we adopt the whitening transform of Olshausen & Field [1].

After training on small patches drawn at random from different image locations, the model learns oriented edge filters, in line with other sparse coding algorithms (figs. 7 and 8). Compared to the outdoor scenes used by Olshausen &

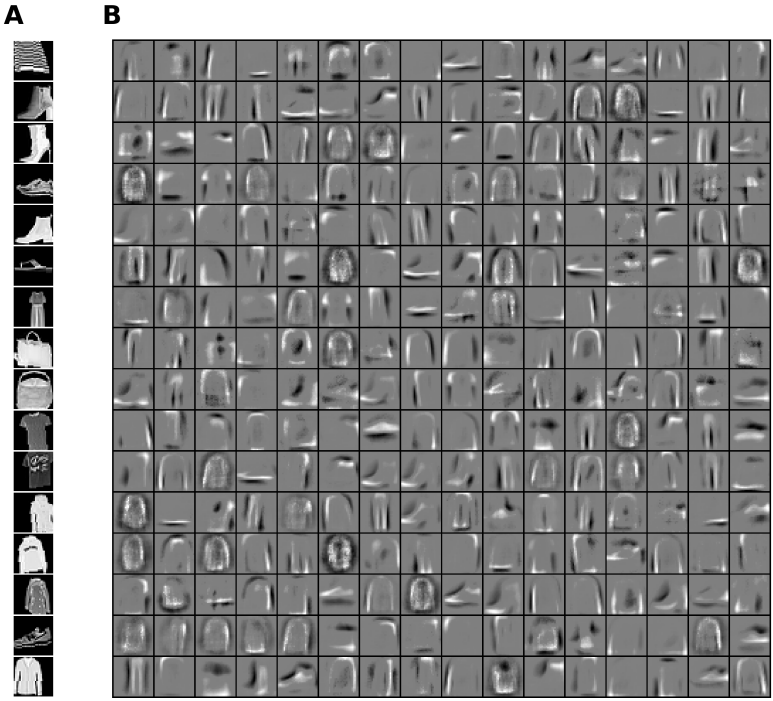

Figure 6: **The network learns the outlines and parts of the various items of clothing in the Fashion-MNIST dataset.** For instance, the neuron in the top right corner detects a short sleeve. **A**: sample input stimuli. **B**: receptive fields. All other details are the same as for the regular MNIST dataset (fig. 3).

Field [1], the NASA dataset yields sligtly more elongated receptive fields but gives otherwise similar results. This is probably due to the more frequent occurence of straight edges in indoor scenes.

Linear decoding

The next series of experiments aims to check whether the network's output is indeed a good encoding of the input. This does not necessarily follow from an analysis of the receptive fields; for instance, a network could succeed in extracting individual independent components, but still fail to encode the mixture of components present in any given input. More specifically, we would like to check whether the sparse encoding produced by the network can be linearly

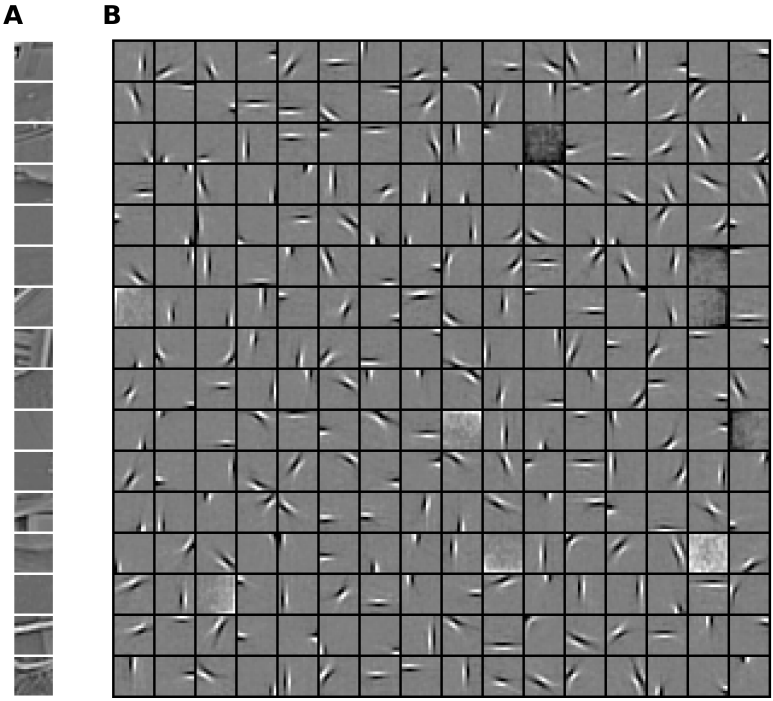

Figure 7: **The model learns oriented edge filters from pre-whitened images from the NASA dataset. A**: sample input stimuli. Middle gray corresponds to zero. **B**: receptive fields after training on 500,000 patches (16×16 pixels) (see *Methods*). In addition to the edge filters, the network develops a pair of non-selective ON and OFF receptive fields (uniform dark and bright patches). These encode the mean of the input, which we do not subtract. Similarly, seven cells learn oriented gradients.

decoded, enabling cheap multiple readouts of cell assemblies as envisioned by Fusi et al. [4].

We first test whether sparse codes make it easier to classify MNIST digits (tbl. 1). Trained on the raw pixels, a linear Support Vector Machine (SVM) classifier performs poorly on MNIST, with an error rate of 8.2 %. But the same linear classifier reaches a much better performance if we train it on the output of the sparse coding network instead of the raw pixels. With $N = 512$ neurons, that combination outperforms the non-parametric k-Nearest Neighbors method (kNN). It also compares with a Multi-Layer Perceptron (MLP) with three layers — in that particular case, the unsupervised sparse coding layer effectively replaces

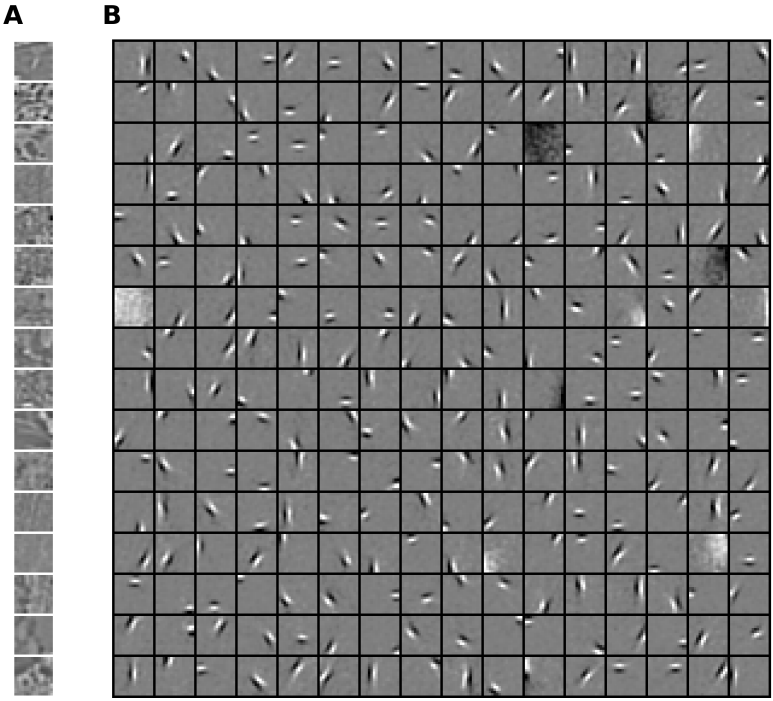

Figure 8: **The model also learns oriented edge filters from the Olshausen & Field dataset. A**: sample input stimuli. **B**: receptive fields. These tend to be shorter and more localized than with the NASA dataset (fig. 7).

two hidden layers trained using backpropagation. With $N = 1024$ neurons, the accuracy reaches a value of 0.981 that is higher than most non-convolutional methods with the exception of polynomial SVM [36,40].

We obtain broadly similar results with the Fashion-MNIST variant (tbl. 2): sparse coding improves linear decoding. However, this dataset is more challenging than the standard MNIST digits for non-convolutional algorithms, and the improvement is consequently smaller. Ensemble learning methods as well as polynomial SVM yield a better performance [36].

Table 1: **The network learns features that improve linear decoding.** Mean and standard deviation of the test error on MNIST. For comparison we include data for a MLP (28×28–300–100–10) from LeCun et al. [40]; other results are our own work.

Input	Classifier	Error (%)
Raw pixels	SVM (linear)	8.2
Raw pixels	kNN ($k = 4$)	3.2
Raw pixels	MLP (3 layers)	2.5 ± 0.1 [40]
Sparse ($N = 256$)	SVM (linear)	3.3 ± 0.2
Sparse ($N = 512$)	SVM (linear)	2.3 ± 0.1
Sparse ($N = 768$)	SVM (linear)	2.1 ± 0.1
Sparse ($N = 1024$)	SVM (linear)	1.9 ± 0.1

Table 2: **Sparse features also improve linear decoding on the Fashion-MNIST dataset, although less than with the standard MNIST.**

Input	Classifier	Error (%)
Raw pixels	SVM (linear)	16.0
Raw pixels	kNN ($k = 4$)	14.2
Sparse ($N = 256$)	SVM (linear)	18.2 ± 0.4
Sparse ($N = 512$)	SVM (linear)	15.5 ± 0.2
Sparse ($N = 768$)	SVM (linear)	14.5 ± 0.3
Sparse ($N = 1024$)	SVM (linear)	14.3 ± 0.2

After that classification task, we turn to linear regression and attempt to reconstruct natural images from the output of the network. While Zylberberg et al. [20] inverted the transformation manually by reusing the network's encoding weights for decoding, here we train a linear model to predict the input patch given the sparse output of the network. We did not attempt to quantify the reconstruction error: pixel-wise measures such as the peak signal-to-noise ratio are neither very informative of how much structure is preserved, nor easy to interpret when comparing different scenes, and better metrics based on structural similarity are non-trivial to compute [41]. Qualitatively, we find that even a small network with 64 neurons preserves the general features of the scene (fig. 9), despite reducing the dimensionality of the data by a factor of 4. Larger networks are able to encode finer details — the larger text on the sample image starts to be legible with 256 sparse coding neurons.

Original image | Decoded (N=64) | Decoded (N=256)

Figure 9: **Input images can be linearly decoded from the sparse output**. Reconstructions from networks of 64 to 256 neurons show increasing fidelity to the original image.

Stability and response to perturbations

In most machine learning experiments, the input data is randomised so that its distribution is mostly homogeneous over time. This is not the case for embodied agents that learn continuously: an animal samples from small regions of the input space as it moves from one place or activity to the next. Thus an important challenge in artificial neural networks is to learn online on non-homogeneous data. Sparse coding networks with a homeostatic term make an explicit assumption that the average firing rate of each neuron is constant, and the violation of that assumption could be a factor in catastrophic forgetting. The next experiment aims to explore whether the absence of a homeostatic term in our model makes it more robust to perturbations.

In fig. 10, we first train the network on the full MNIST dataset with Gaussian noise ($\sigma = 0.2$) added to the digits and clipped to $[0, 1]$. After 150000 stimuli, we remove the MNIST input and continue training on the background noise. We restore the input and train again on the full MNIST dataset for 150000 stimuli. Finally, we perform one last training round on a subset of MNIST that contains only the zeroes, with all other digits removed.

We find that the receptive fields retain their selectivity despite fading during the period when the network receives only background noise, and recover with minimal changes when the original input is restored (fig. 11): thanks to the lack of fast IP, input deprivation does not induce catastrophic forgetting. As long as the distribution of the independent components remains the same, there is

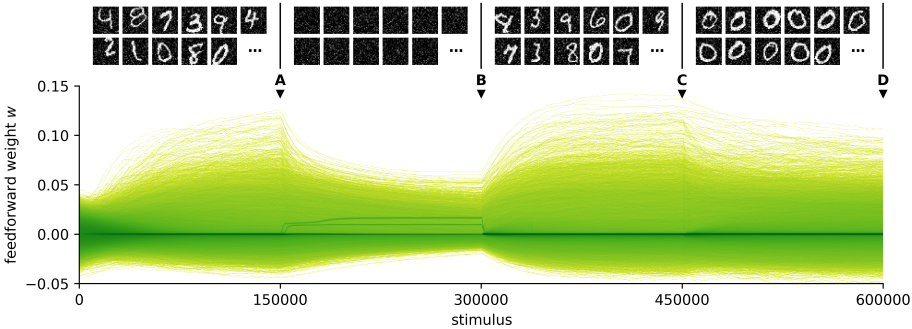

Figure 10: **Weights converge smoothly to a steady state after perturbations in the input. Top**: sample training stimuli for each of the four training periods. **Bottom**: density plot of the trajectories of the 112896 feedforward weights w during training ($N = 144$). The color mapping is logarithmic to account for the high density of zero weights. The letters **A**, **B**, **C** and **D** mark the times when snapshots of the receptive fields were taken (fig. 11).

also no drift with continued learning (compare **A** and **C** in fig. 11). In contrast, we observed a constant shifting of the receptive fields when replicating other models such as the one by Zylberberg et al. [20].

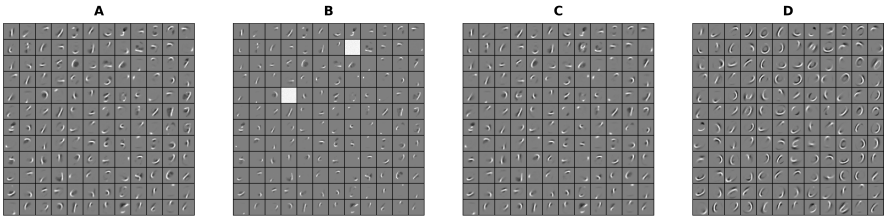

Figure 11: **Snapshots of the receptive fields at the times marked in fig. 10. A:** receptive fields at the end of the initial training period. **B:** receptive fields fade during input deprivation, but do not drift. **C:** they recover with minimal changes after the original input is restored. **D:** reorganisation occurs after further training with zeros only.

However, the receptive fields do change rapidly when we switch from the full MNIST to zeroes only: they adapt to match the new distribution of the independent components and forget the features that were specific to other digits (such as straight lines). Thus the lack of IP protects against forgetting during input deprivation but does not block continual adaptation to the input, as long as the new stimuli overlap with existing receptive fields.

A small number of neurons (typically one or two) respond strongly to the noise during the period of input deprivation (bright receptive fields in fig. 11; dark lines in figs. 10, 12). Average firing rates for the other cells are low; again, this can be explained by the absence of a homeostatic term that would drive every neuron towards a target firing rate. Since the background noise does not contain any structure, these few active cells are sufficient to encode it and inhibit other neurons, protecting their receptive fields.

The transient increase in activity when the input is restored does not exceed three times the baseline: spikes remain sparse throughout, and come back to normal after 10 seconds (fig. 12). Since the neurons have fixed somatic and dendritic thresholds, that increase must come from the decay of lateral inhibition or from a shift in the excitatory/inhibitory balance of the feedforward weights. In contrast, in a network with IP, homeostatic adjustment of the thresholds to the background noise would cause a temporary saturation of the transfer function and loss of sparseness when the input is restored.

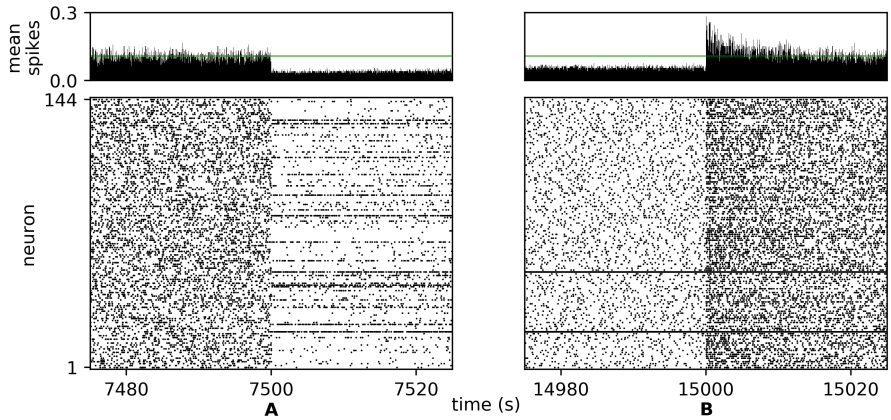

Figure 12: **The network is robust to input deprivation. Top:** mean number of spikes per neuron per stimulus; the green line marks the average value before mark A. **Bottom:** raster plot of the output spikes. Following input deprivation (mark **A**), firing patterns return to normal within 20 seconds after the input is restored (mark **B**), except for the neurons that responded strongly to the noise (these take somewhat longer).

Discussion

Dendritic learning and compartmentalised inputs

Biological neurons are more than just thresholding devices [42]: they can perform computations that are significantly more complex than the traditional artificial neurons with a single summation stage feeding into a sigmoidal or rectified linear (ReLU) transfer function. These include temporal processing at various levels from the synapses to the soma, and multi-stage integration of inputs in the dendrititic tree [43,44], achieving capabilities in individual cells that would normally require a network of point neurons.

With this paper, we give one example of the types of learning that become possible in neurons with separate compartments. Point neurons are capable of sparse coding, but they must approach the problem of from the angle of lifetime sparseness. Our contribution is to show how the addition of a dendritic compartment gives the learning rule access to more information that lets it modulate nonlinear Hebbian learning via population sparseness as well.

In that respect, the model by Körding & König [30] is perhaps the closest to our work. In their network, lateral inhibition decides whether a neuron is losing or winning the competition by blocking the backpropagating action potentials, but it does so without blocking the spikes that travel down the axon. In contrast, lateral inhibition in our model suppresses both the internal teaching signal and the output of the neurons. The distinction could have its relevance in multilayer networks; but it is likely that both architectures can perform sparse coding with the right dendritic learning rule — something that Körding & König [30] could not explore, as they used simpler stimuli made of a single independent component.

A single dendritic compartment is still a stark simplification over the finely branched structure of dendritic trees. Legenstein & Maass [45] use multiple dendrites to solve a nonlinear binding task where each neuron learns to respond to multiple patterns (for instance *AB* and *CD*), while ignoring other combinations of the same input dimensions (*AC* and *BD*); Hawkins & Ahmad [46] exploit a similar mechanism. As for Schiess et al. [47], they extend the somatodendritic learning rule of Urbanczik & Senn [31] to reward-modulated learning with multiple dendrites. One area of future research is therefore to apply our sparse coding rule to neurons with more than two compartments, for instance for the purpose of learning sparse codes from a bottom-up input and predictive associations from top-down sources at the same time.

The wave of interest in modelling neural networks with compartmentalised inputs now extends to the neuromorphic hardware that can simulate them efficiently, with some experimental support for dendrites on the SpiNNaker chip

[48], and multiple compartments and input traces on Intel's Loihi [49]. As the idea makes its way from biology to machine learning [46,50], we expect to see a shift from a paradigm where artificial neural networks employ very simple units and rely on supervised learning to distribute a task over this generic computational substrate, to a paradigm where single neurons perform a substantial amount of computation and where the structure of these neurons already encodes a particular approach towards solving the task.

Learning sparse codes without intrinsic plasticity

Our findings confirm that intrinsic plasticity is not strictly required to learn sparse codes. In addition to its role in decorrelating the population responses, plastic lateral inhibition can also regulate sparseness through its effect on the nonlinear Hebbian learning rule. Freed from the need to provide that fast negative feedback, IP could instead act on timescales slower than Hebbian plasticity, and help recruit previously silent dendrites.

Without fast IP, the network can be made more robust to temporary input deprivation. Because of their selective receptive fields, the neurons respond only weakly to background noise, and the post-synaptic gating in the learning rule protects the synaptic weights from rapid changes.

But replacing intrinsic plasticity with synaptic plasticity is still not enough to cope with the changes that an animal or robot would encounter as it switches between tasks and environments: the network remains susceptible to rapid and extensive reorganisation when novel inputs overlap the existing receptive fields, or when the distribution of the independent components changes. On the one hand, that kind of adaptability is desirable as natural environments are not static and the quick acquisition of novel stimuli can be critical for survival. But on the other hand, it should disturb existing receptive fields as little as possible so as not to erase previous experiences and all the associations that build upon them.

Although increasing sparseness and careful tuning of learning rates could help, it is likely that solving that stability-plasticity dilemma will require ad-hoc gating mechanisms. Some candidates are the conditional consolidation of synaptic changes [51], neuromodulation and attention [52,53], or a mechanism based on top-down prediction errors like the Adaptive Resonance Theory [54].

On sparse coding and associative readouts

In Buzsáki's perspective [3], every pathway that links two populations of cells involves a readout or transformation of one neural code into another. But in

machine learning, complex transformations often require multiple layers. Does the brain use interneurons for that purpose, or does it somehow solve the problem without them? Although there are excitatory interneurons in the cortex (layer IV stellate cells), we know that inputs from distal areas converge directly onto the dendrites of pyramidal cells [55], which favours the direct readout hypothesis. That architecture would also scale better to larger networks.

Compared to dense codes (where every cell participates in coding every stimulus), there are two reasons why sparse codes should make direct readouts easier to learn. First, because fewer active units mean fewer weights to tune for any given mapping — in that sense, sparseness could act as a form of regularisation that prevents overfitting. Second, because separating the independent components of the signal should also help to disentangle the factors of variation, and make the problem more linearly tractable.

Compared to a local code (where every stimulus has its own dedicated cells), sparse codes should allow the readouts to generalise to novel inputs while retaining the ability to encode small differences. Instead of encoding each stimulus as a whole (as happens in nearest-neighbour clustering, self-organising maps and strict winner-take-all networks), a sparse coding network encodes each stimulus as a combination of features that it shares with other stimuli. It is thus able to respond to the familiar features of an unfamiliar input.

Our results indicate that the use of sparse codes can help, allowing a simple linear readout to reach the same accuracy as a multi-layer network trained on the raw input. This suggests that one could learn transformations from one sparse code to another sparse code by adding just an extra set of synapses to the target neurons. However, the decoding tasks we performed in this paper are not necessarily representative of the kind of readouts that a neural system embodied in an animal or robot needs to perform: in the case of MNIST, the target classes are few and mutually exclusive; and in the case of natural images, the output space is the same as the input space.

It would therefore be of value to test the idea with more realistic types of inputs and readout — for instance, encoding sensory and motor information and learning predictive associations between one modality and another. And the lesser improvement in linear decoding on the Fashion-MNIST variant shows that detecting a complex arrangement between the parts of a sparse code may still require more than a single linear readout — not just an extra set of synapses, but an extra set of compartments as well, as in Legenstein & Maass [45], leveraging dendritic arithmetics to solve the task [44].

Biophysical interpretation

In certain aspects, the architecture of our network ressembles cortical networks. Pyramidal cells are also the convergence point of distal inputs and local recurrent pathways, and their somas receive targeted inhibition from parvalbumin-containing basket cells that are activated by neighbouring neurons [56]. This suggests that the cortex might be making a similar use of compartmentalised inhibition for the purpose of learning sparse codes.

However, given the diversity of cortical inhibitory pathways [57], the roles of compartmentalised inhibition in cortical neurons must be considerably more complex than our model can possibly account for. For instance, other types of inhibitory interneurons bring recurrent inhibition to the dendrites [58,59]; one could attempt to include these in a computational model, following up on the work of Spratling & Johnson [60]. Wilmes et al. [61] also modelled the mechanism postulated by Körding & König [30], where inhibition does not suppress somatic spiking, but blocks the backpropagating action potentials on their way to the dendrites — a mechanism which could also sustain a sparse coding rule.

As it stands, our somato-dendritic learning rule (eqs. 1 and 2) contains a number of hypotheses about plasticity in biological neurons. Loosely speaking, the z term corresponds to backpropagating action potentials (bAPs), while the y term signals dendritic activity. This implies that somatic spikes should lead to long-term potentiation (LTP) of active excitatory synapses, while dendritic activity should cause depression (LTD). For dendritic inhibitory synapses, the situation should be reversed, with dendritic activity leading to LTP and bAPs leading to LTD.

There is some evidence in support of the first hypothesis: bAPs are a classical trigger of LTP, while dendritic spikes [62,63] and NMDA receptor activation [64,65] have both been linked to LTD or blockage of LTP. But there is also evidence to the contrary: NMDAR activation is central to LTP as well [64,66,67], and dendritic spikes can induce LTP without bAPs [68].

Nonetheless it seems that dendritic LTP without bAPs requires a local sodium spikelet [69,70]: a fast, spike-like depolarization that sometimes, but not always, accompanies NMDA spikes [71]. This suggests rephrasing our biophysical interpretation and equating the z term with fast voltage transients — either bAPs or sodium spikelets — while the y term would correspond to slower dendritic events like elevated calcium. The question is then what sort of conditions can trigger sodium spikelets in the absence of bAPs, and how the operation of the learning rule would change if these were included in the model.

As for inhibitory plasticity, Holmgren & Zilberter [72] bring some support to the notion that plasticity at dendritic inhibitory synapses could be reversed com-

pared to excitatory synapses. They report LTD of inhibitory inputs coincident with bAPs, and LTP for those that come up to 800 ms *after* the train of action potentials. The latter fact also hints at a dimension of temporal processing which we ignore in this model and which we could explore in further work — for instance adapting the learning rule to learn transitions and sequences.

More generally, these questions call for further electrophysiological investigations of somato-dendritic plasticity rules. There would be much to learn from experiments that vary dendritic and somatic activity independently — controlling the number of somatic action potentials emitted during a dendritic spike, or the relative timing of somatic and dendritic events.

Models & Methods

Somatic compartments and somatic synapses

The somatic compartments are standard LIF neurons. The membrane potential u follows the following equation:

$$\tau_m \frac{du}{dt} = I_d + I_s - u \tag{5}$$

where I_d and I_s are the currents from the dendrite and somatic synapses, respectively.

We use a fixed spiking threshold θ and after-spike reset ρ without a refractory period:

$$u \leftarrow \rho \quad \text{if } u \geq \theta \tag{6}$$

We compute a firing rate z that takes into account the number of spikes and also their latency relative to the stimulus onset t_0. First we define a trace ζ that increases after each spike and decays exponentially:

$$\begin{aligned} \zeta &\leftarrow \zeta + 1 \quad \text{if } u \geq \theta \\ \tau_\zeta \frac{d\zeta}{dt} &= -\zeta \end{aligned} \tag{7}$$

Then we normalise so that the area under the curve is the number of spikes, and integrate over the stimulus window:

$$z = \int_{t=t_0}^{t_0+50ms} \frac{\zeta}{\tau_\zeta} dt \tag{8}$$

Thus a spike that occurs towards the end of the window contributes less to the total than a spike that occurs early.

Somatic inhibition comes from lumped, conductance-based synapses where the fraction g_s of active conductance and somatic current I_s evolve as follows for each neuron post:

$$\begin{aligned} g_s &\leftarrow g_s + q_{\text{pre}\to\text{post}} \quad \text{on spike from neuron pre} \\ \tau_s \frac{dg_s}{dt} &= -g_s \\ I_s &= -g_s u \end{aligned} \tag{9}$$

The initial weights q are drawn from an exponential distribution (mean = 0.01).

Before each new stimulus we reset the continuous-time variables of the model, as in Zylberberg et al. [20]:

$$\begin{aligned} u &\leftarrow \rho \\ \zeta &\leftarrow 0 \\ g_s &\leftarrow 0 \end{aligned} \tag{10}$$

That reset does not seem to be critical for our findings, but we did not explore the issue further.

Dendritic compartments

Dendrites are rate- and current-based. The net dendritic input g_d and dendritic activation y for each neuron post are as follows:

$$\begin{aligned} g_d &= \sum_{\text{pre}} x_{\text{pre}} w_{\text{pre}\to\text{post}} \\ y &= \max(g_d - \theta_d, 0) \end{aligned} \tag{11}$$

The initial weights w are drawn from a normal distribution (std = 0.01).

The current I_d from the dendrite to the soma is a nonlinear function of the dendritic activation:

$$I_d = \begin{cases} y_0 + \kappa y & \text{if } y > 0 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Here the goal is to reproduce the active properties of biological dendrites. Above a certain input threshold, regenerative activation of the NMDA receptors causes dendritic spikes. These lead to a sharp increase in membrane potential followed

by a plateau where stronger inputs cause no further increase in voltage [73,74]. We model this with a step function and the offset y_0. However, stronger inputs do increase the duration, and reduce the rise time of the plateau, producing more somatic spikes. We model this with the linear term κy. In practice we adjust y_0 to cancel out the somatic rheobase, so that suprathreshold dendritic activation elicits at least one spike in the absence of somatic inhibition. We find that θ_d is not critical for our findings as long as all dendrites respond to some inputs at the start of the simulation. Here we set it to zero, although a small positive value would better reproduce the data in Milojkovic et al. [73] and Oikonomou et al. [74].

Parameters

Table 3 summarises the parameters for the neuron model. All simulations are performed with a timestep $dt = 0.5$ ms and 100 steps per stimulus, except for fig. 2 which uses a finer timestep.

Table 3: **Model parameters used for all experiments in this paper.**

Soma and somatic synapses	Dendrites
$\theta = 1.0$	$\theta_d = 0.0$
$\rho = 0.0$	$y_0 = 1.0$
$\tau_m = 10.0$ ms	$\kappa = 0.5$
$\tau_\zeta = 50.0$ ms	$\delta = 0.5$
$\tau_s = 5.0$ ms	$\lambda = 0.01$
$\nu = 1 \times 10^{-1}$	$\mu = 4 \times 10^{-4}$
$\beta = N/250$	$\alpha = 1.0$

Receptive Fields

Throughout this paper we use the weights of the neurons as a proxy for their actual receptive fields. Showing all the weights of the network on the same image requires that we normalise each receptive field separately, because neurons that respond to narrow features have larger absolute weights than those that respond to broad ones. Nonetheless, we make sure that zero weights appear as the same middle gray for all neurons, allowing quick identification of ON (brighter) and OFF (darker) areas. Thus we normalise the receptive field $W_i = \begin{bmatrix} w_{1\to i} & \cdots & w_{d\to i} \end{bmatrix}$ of each neuron i as follows when generating the figures:

$$W_i' = \frac{1 + a_i W_i}{2} \tag{13}$$

where $a_i = \left(\max_{1 \le j \le d} |w_{j \to i}|\right)^{-1}$ and d is the number of input dimensions.

MNIST

We use both the standard MNIST dataset [35] and the Fashion-MNIST variant [36], each with 60,000 training samples and 10,000 test samples. We map the full range of the data to the interval $[0, 1]$. When training the sparse coding network, we shuffle the patterns and distort them with random shears and translations, as done in LeCun et al. [40]. The purpose of these distortions is to increase the number of distinct training samples, and also to remove the correlations introduced by the centering of the patterns. We do this by applying the following affine transformation with the origin at the center of the pattern:

$$M = \begin{bmatrix} 1 & A_1 & T_1 \\ A_2 & 1 & T_2 \\ 0 & 0 & 1 \end{bmatrix}$$

where each A_i is a random variable drawn from $\mathcal{N}(\sigma = 0.1)$, and each T_i is a random variable drawn from $\mathcal{N}(\sigma = 2.0)$. Distorted digits produce more localised receptive fields than the centered patterns, which in turn improves the performance of classifiers trained on the output of the network. When training and testing the classifiers themselves, we freeze the weights of the sparse coding network and we use the plain stimuli without distortions.

In tbl. 1 we use the following classifiers from `scikit-learn` [75] version 0.19.1:

SVM: `LinearSVC(C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, loss=squared_hinge, max_iter=1000, multi_class=ovr, penalty=l2, random_state=2136146589, tol=0.0001, verbose=0)`

kNN: `KNeighborsClassifier(algorithm=auto, leaf_size=30, metric=minkowski, metric_params=None, n_jobs=1, n_neighbors=4, p=2, weights=uniform)`

Natural images

We use two datasets of photographic images: one by Olshausen & Field, which consists of natural outdoors scenes [76], and one compiled from public-domain images by NASA [37], which can be found in the supporting information of this paper (S1 File). In both cases, each image was converted to grayscale, resized to an area of 200,000 pixels, preprocessed using the same whitening transform as Olshausen & Field [1], and then normalised to unit variance. No further

normalisation was applied to the individual patches used for training; in particular, the patch mean was not subtracted from the input. Note that in contrast to MNIST the natural image stimuli contain both positive and negative values. We interpret these as ON and OFF channels from the retina; while it would be more realistic to split the ON and OFF values into separate, non-negative channels, we did not attempt this here.

For the reconstruction experiment, the input image was tiled into overlapping patches with a width of 16 pixels and a stride of 8 pixels. Each input patch was run through a sparse coding network pre-trained on the NASA dataset. The sparse output was then fed as the input to a linear model trained with ridge regression to recontruct the original patches. Finally, the predicted patches were placed at their original locations and averaged to account for the stride overlap.

References

1. Olshausen BA, Field DJ. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 1997. doi: 10.1016/S0042-6989(97)00169-7.

2. Hebb DO. The Organization of Behavior: A Neuropsychological Theory. John Wiley & Sons, Inc., 1949. ISBN: 9781410612403.

3. Buzsáki G. Neural Syntax: Cell Assemblies, Synapsembles, and Readers. *Neuron*, 2010. doi: 10.1016/j.neuron.2010.09.023.

4. Fusi S, Miller EK, Rigotti M. Why neurons mix: high dimensionality for higher cognition. *Current Opinion in Neurobiology*, 2016. doi: 10.1016/j.conb.2016.01.010.

5. Bell AJ, Sejnowski TJ. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 1995. doi: 10.1016/0165-1684(91)90081-S.

6. Hyvarinen A, Oja E. Independent component analysis: algorithms and applications. *Neural Networks*, 2000. doi: 10.1016/S0893-6080(00)00026-5.

7. Hafner VV, Fend M, König P, Körding KP. Predicting properties of the rat somatosensory system by sparse coding. *Neural Information Processing Letters and Reviews*, 2004; 4:11–18. Available from: http://bsrc.kaist.ac.kr/nip-lr/V04N01/V04N01P2-11-18.pdf.

8. Makhzani A, Frey BJ. Winner-take-all autoencoders. In: *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015; 2791–2799.

Available from: http://papers.nips.cc/paper/5783-winner-take-all-autoencoders.pdf.

9. Diaconis P, Freedman D. Asymptotics of Graphical Projection Pursuit. *The annals of statistics*, 1984; 12:793–815. Available from: http://www.jstor.org/stable/2240961.

10. Barlow HB. Cerebral Cortex as Model Builder. In: Vaina LM, editor. *Matters of Intelligence*, Springer, Dordrecht, 1987. doi: 10.1007/978-94-009-3833-5_18.

11. Rumelhart DE, Zipser D. Feature Discovery by Competitive Learning. *Cognitive Science*, 1985. doi: 10.1207/s15516709cog0901_5.

12. Kohonen T. The self-organizing map. *Proceedings of the IEEE*, 1990. doi: 10.1109/5.58325.

13. Marshall JA. A self-organizing scale-sensitive neural network. In: *1990 IJCNN International Joint Conference on Neural Networks*, IEEE, 1990. doi: 10.1109/IJCNN.1990.137911.

14. Marshall JA. Development of perceptual context-sensitivity in unsupervised neural networks: parsing, grouping, and segmentation. In: *1992 IJCNN International Joint Conference on Neural Networks*, IEEE, 1992. doi: 10.1109/IJCNN.1992.227155.

15. Földiák P. Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics*, 1990. doi: 10.1007/BF02331346.

16. Falconbridge MS, Stamps RL, Badcock DR. A Simple Hebbian/Anti-Hebbian Network Learns the Sparse, Independent Components of Natural Images. *Neural Computation*, 2006. doi: 10.1162/089976606775093891.

17. Butko NJ, Triesch J. Learning sensory representations with intrinsic plasticity. *Neurocomputing*, 2007. doi: 10.1016/j.neucom.2006.11.006.

18. Lucke J. A Dynamical Model for Receptive Field Self-organization in V1 Cortical Columns. In: de Sá JM, Alexandre LA, Duch W, Mandic D, editors. *Artificial Neural Networks – ICANN 2007*, Springer Berlin Heidelberg, 2007. doi: 10.1007/978-3-540-74695-9_40.

19. Savin C, Joshi P, Triesch J. Independent Component Analysis in Spiking Neurons. *PLoS Computational Biology*, 2010. doi: 10.1371/journal.pcbi.1000757.

20. Zylberberg J, Murphy JT, DeWeese MR. A Sparse Coding Model with Synaptically Local Plasticity and Spiking Neurons Can Account for the Diverse

Shapes of V1 Simple Cell Receptive Fields. *PLoS Computational Biology*, 2011. doi: 10.1371/journal.pcbi.1002250.

21. King PD, Zylberberg J, DeWeese MR. Inhibitory interneurons decorrelate excitatory cells to drive sparse code formation in a spiking model of V1. *Journal of Neuroscience*, 2013. doi: 10.1523/JNEUROSCI.4188-12.2013.

22. Ferré P, Mamalet F, Thorpe SJ. Unsupervised Feature Learning With Winner-Takes-All Based STDP. *Frontiers in Computational Neuroscience*, 2018. doi: 10.3389/fncom.2018.00024.

23. Brito CSN, Gerstner W. Nonlinear Hebbian Learning as a Unifying Principle in Receptive Field Formation. *PLoS Computational Biology*, 2016. doi: 10.1371/journal.pcbi.1005070.t001.

24. Bienenstock EL, Cooper LN, Munro PW. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 1982. doi: 10.1523/JNEUROSCI.02-01-00032.1982.

25. Triesch J. A Gradient Rule for the Plasticity of a Neuron's Intrinsic Excitability. In: *Artificial Neural Networks: Biological Inspirations – ICANN 2005*, Springer Berlin Heidelberg, 2005. doi: 10.1007/11550822_11.

26. Triesch J. Synergies Between Intrinsic and Synaptic Plasticity Mechanisms. *Neural Computation*, 2007. doi: 10.1162/neco.2007.19.4.885.

27. Zenke F, Hennequin G, Gerstner W. Synaptic Plasticity in Neural Networks Needs Homeostasis with a Fast Rate Detector. *PLoS Computational Biology*, 2013. doi: 10.1371/journal.pcbi.1003330.s002.

28. Toyoizumi T, Kaneko M, Stryker MP, Miller KD. Modeling the Dynamic Interaction of Hebbian and Homeostatic Plasticity. *Neuron*, 2014. doi: 10.1016/j.neuron.2014.09.036.

29. Chistiakova M, Bannon NM, Chen J, Bazhenov M, Volgushev M. Homeostatic role of heterosynaptic plasticity: models and experiments. *Frontiers in Computational Neuroscience*, 2015. doi: 10.3389/fncom.2015.00089.

30. Körding KP, König P. A learning rule for dynamic recruitment and decorrelation. *Neural Networks*, 2000. doi: 10.1016/S0893-6080(99)00088-X.

31. Urbanczik R, Senn W. Learning by the Dendritic Prediction of Somatic Spiking. *Neuron*, 2014. doi: 10.1016/j.neuron.2013.11.030.

32. Widrow B, Hoff ME. Adaptive switching circuits. Technical report 1553-1, Stanford Electronics Laboratories, Stanford University, 1960. Available from: http://www.dtic.mil/get-tr-doc/pdf?AD=AD0241531.

33. Sutton RS, Barto AG. Toward a modern theory of adaptive networks: expectation and prediction. *Psychological Review*, 1981. doi: 10.1037/0033-295X.88.2.135.

34. Marshall JA. Adaptive perceptual pattern recognition by self-organizing neural networks: Context, uncertainty, multiplicity, and scale. *Neural Networks*, 1995. doi: 10.1016/0893-6080(94)00099-8.

35. LeCun Y, Cortes C. The MNIST database of handwritten digits. [Internet] 1998. Available from: http://yann.lecun.com/exdb/mnist/. Cited 13 September 2016.

36. Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. Preprint, 2017. Available from: ArXiv:1708.07747. Cited 08 May 2018.

37. NASA on The Commons: Women of NASA. [Internet] National Aeronautics and Space Administration. Available from: https://www.flickr.com/photos/nasacommons/albums/72157630503052344. Cited 09 May 2018.

38. van Hateren JH, van der Schaaf A. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society B: Biological Sciences*, 1998. doi: 10.1098/rspb.1998.0303.

39. Blais BS, Intrator N, Shouval HZ, Cooper LN. Receptive Field Formation in Natural Scene Environments: Comparison of Single-Cell Learning Rules. *Neural Computation*, 1998. doi: 10.1088/0954-898X/7/3/003.

40. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. doi: 10.1109/5.726791.

41. Thung K, Raveendran P. A survey of image quality measures. In: *2009 International Conference for Technical Postgraduates (TECHPOS)*, IEEE, 2009. doi: 10.1109/TECHPOS.2009.5412098.

42. Koch C, Segev I. The role of single neurons in information processing. *Nature neuroscience*, 2000. doi: 10.1038/81444.

43. Poirazi P, Brannon T, Mel BW. Pyramidal Neuron as Two-Layer Neural Network. *Neuron*, 2003. doi: 10.1016/S0896-6273(03)00149-1.

44. Spruston N, Kath WL. Dendritic arithmetic. *Nature neuroscience*, 2004. doi: 10.1038/nn0604-567.

45. Legenstein R, Maass W. Branch-Specific Plasticity Enables Self-Organization of Nonlinear Computation in Single Neurons. *Journal of Neuroscience*, 2011. doi: 10.1523/JNEUROSCI.5684-10.2011.

46. Hawkins J, Ahmad S. Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex. *Frontiers in Neural Circuits*, 2016. doi: 10.3389/fncir.2016.00023.

47. Schiess M, Urbanczik R, Senn W. Somato-dendritic Synaptic Plasticity and Error-backpropagation in Active Dendrites. *PLoS Computational Biology*, 2016. doi: 10.1371/journal.pcbi.1004638.s003.

48. Hopkins M, Pineda-Garc G, Bogdan PA, Furber SB. Spiking neural networks for computer vision. *Interface Focus*, 2018. doi: 10.1098/rsfs.2018.0007.

49. Davies M, Srinivasa N, Lin T, Chinya G, Cao Y, et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*, 2018. doi: 10.1109/MM.2018.112130359.

50. Guerguiev J, Lillicrap TP, Richards BA. Towards deep learning with segregated dendrites. *eLife*, 2017. doi: 10.7554/eLife.22901.

51. Redondo RL, Morris RGM. Making memories last: the synaptic tagging and capture hypothesis. *Nature Reviews Neuroscience*, 2011. doi: 10.1038/nrn2963.

52. Hasselmo ME. Neuromodulation and cortical function: modeling the physiological basis of behavior. *Behavioural Brain Research*, 1995. doi: 10.1016/0166-4328(94)00113-T.

53. Krichmar JL. The neuromodulatory system: a framework for survival and adaptive behavior in a challenging world. *Adaptive Behavior*, 2008. doi: 10.1177/1059712308095775.

54. Grossberg S. How does a brain build a cognitive code? *Psychological Review*, 1980. doi: 10.1037/0033-295X.87.1.1.

55. Izhikevich EM, Edelman GM. Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences*, 2008. doi: 10.1073/pnas.0712231105.

56. Freund TF, Katona I. Perisomatic Inhibition. *Neuron*, 2007. doi: 10.1016/j.neuron.2007.09.012.

57. Kubota Y, Karube F, Nomura M, Kawaguchi Y. The Diversity of Cortical Inhibitory Synapses. *Frontiers in Neural Circuits*, 2016. doi: 10.3389/fncir.2016.00027.

58. Jadi M, Polsky A, Schiller J, Mel BW. Location-Dependent Effects of Inhibition on Local Spiking in Pyramidal Neuron Dendrites. *PLoS Computational Biology*, 2012. doi: 10.1371/journal.pcbi.1002550.

59. Bloss EB, Cembrowski MS, Karsh B, Colonell J, Fetter RD, et al. Structured Dendritic Inhibition Supports Branch-Selective Integration in CA1 Pyramidal Cells. *Neuron*, 2016. doi: 10.1016/j.neuron.2016.01.029.

60. Spratling MW, Johnson MH. Preintegration Lateral Inhibition Enhances Unsupervised Learning. *Neural Computation*, 2002. doi: 10.1162/089976602320264033.

61. Wilmes KA, Sprekeler H, Schreiber S. Inhibition as a Binary Switch for Excitatory Plasticity in Pyramidal Neurons. *PLoS Computational Biology*, 2016. doi: 10.1371/journal.pcbi.1004768.

62. Holthoff K, Kovalchuk Y, Yuste R, Konnerth A. Single-shock LTD by local dendritic spikes in pyramidal neurons of mouse visual cortex. *The Journal of physiology*, 2004. doi: 10.1113/jphysiol.2004.072678.

63. Holthoff K, Kovalchuk Y, Konnerth A. Dendritic spikes and activity-dependent synaptic plasticity. *Cell and Tissue Research*, 2006. doi: 10.1007/s00441-006-0263-8.

64. Abraham WC. Metaplasticity: tuning synapses and networks for plasticity. *Nature Reviews Neuroscience*, 2008. doi: 10.1038/nrn2356.

65. Nabavi S, Kessels HW, Alfonso S, Aow J, Fox R, et al. Metabotropic NMDA receptor function is required for NMDA receptor-dependent long-term depression. *Proceedings of the National Academy of Sciences*, 2013. doi: 10.1073/pnas.1219454110.

66. Lüscher C, Malenka RC. NMDA Receptor-Dependent Long-Term Potentiation and Long-Term Depression (LTP/LTD). *Cold Spring Harbor Perspectives in Biology*, 2012. doi: 10.1101/cshperspect.a005710.

67. Brandalise F, Carta S, Helmchen F, Lisman JE, Gerber U. Dendritic NMDA spikes are necessary for timing-dependent associative LTP in CA3 pyramidal cells. *Nature Communications*, 2016. doi: 10.1038/ncomms13480.

68. Golding NL, Staff NP, Spruston N. Dendritic spikes as a mechanism for cooperative long-term potentiation. *Nature*, 2002. doi: 10.1038/nature00854.

69. Kim Y, Hsu C, Cembrowski MS, Mensh BD, Spruston N. Dendritic sodium spikes are required for long-term potentiation at distal synapses on hippocampal pyramidal neurons. *eLife*, 2015. doi: 10.7554/eLife.06414.

70. Kim S, Kim Y, Lee S, Ho W. Dendritic spikes in hippocampal granule cells are necessary for long-term potentiation at the perforant path synapse. *eLife*, 2018. doi: 10.7554/eLife.35269.

71. Milojkovic BA, Wuskell JP, Loew LM, Antic SD. Initiation of Sodium Spikelets in Basal Dendrites of Neocortical Pyramidal Neurons. *The Journal of Membrane Biology*, 2005. doi: 10.1007/s00232-005-0827-7.

72. Holmgren CD, Zilberter Y. Coincident Spiking Activity Induces Long-Term Changes in Inhibition of Neocortical Pyramidal Cells. *Journal of Neuroscience*, 2001. doi: 10.1523/JNEUROSCI.21-20-08270.2001.

73. Milojkovic BA, Radojicic MS, Antic SD. A strict correlation between dendritic and somatic plateau depolarizations in the rat prefrontal cortex pyramidal neurons. *Journal of Neuroscience*, 2005. doi: 10.1523/JNEUROSCI.5314-04.2005.

74. Oikonomou KD, Short SM, Rich MT, Antic SD. Extrasynaptic Glutamate Receptor Activation as Cellular Bases for Dynamic Range Compression in Pyramidal Neurons. *Frontiers in Physiology*, 2012. doi: 10.3389/fphys.2012.00334.

75. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2011; 12:2825–2830. Available from: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.

76. Olshausen BA. Sparse Coding Simulation Software. 1997. Available from: http://redwood.berkeley.edu/bruno/sparsenet/.

Supporting Information

S1 File. NASA images dataset. Natural images dataset compiled from selected public-domain images from NASA on the Commons.