Somatic inhibition controls dendritic selectivity in a sparse coding network of spiking neurons.

Damien Drix[1]

[1] Humboldt Universität, Berlin, Germany

Corresponding author:
Damien Drix[1]

Email address: damien.drix@gmail.com

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

Somatic inhibition controls dendritic selectivity in a sparse coding network of spiking neurons.

Damien Drix

Humboldt Universität zu Berlin

damien.drix@gmail.com

Abstract

Sparse coding is an effective operating principle for the brain, one that can guide the discovery of features and support the learning of assocations. Here we show how spiking neurons with discrete dendrites can learn sparse codes via an online, nonlinear Hebbian rule based on the concept of somato-dendritic mismatch. The rule gives lateral inhibition direct control over the selectivity of dendritic receptive fields, without the need for a sliding threshold. The network discovers independent components that are similar to the features learned by a sparse autoencoder. This improves the linear decodability of the input: combined with a linear readout, our single-layer network performs as well as a deeper multi-layer Perceptron on the MNIST dataset. It can also produce topographic feature maps when the lateral connections are organised in a center-surround pattern, although this does not improve the quality of the encoding.

1. Introduction

There is much that animals have to learn from experience: the way the world behaves, the value of things, how to act and how to react. These forms of learning work in different ways, but they all rely, at some level, on the ability to make associations: across time, between sensorimotor modalities, between events and rewards.

This is no simple task, as we discover when we try to build machines capable of the same feats. Learning these associations is learning to perceive the world in a way that reveals some of

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

its implicit structure — the entities it is made of, the mechanisms that couple them, the regularities that matter in the agent's ecological niche. The body and peripheral nervous system already go some way towards organising perception (Lettvin et al., 1959). But the nervous activity induced by the senses still does not tell much, at first sight, about what caused it: cochlear neurons respond to their preferred frequency regardless of the nature of the sound; all sorts of visual scenes cause the same retinal cell to fire.

To make sense of sensory stimulation, the brain must discover patterns in it. That is the task of the central nervous system, the cortex in particular, and the process involves the construction of a neural code.

The coding strategies available to the brain are diverse. At one end of the spectrum are dense distributed codes, where every neuron participates in the encoding of every input. At the other end are local codes, where each neuron codes single-handedly for an entire input pattern. Neither of these two extremes are ideal for associative learning. Dense codes make it hard to distinguish situations or percepts that call for different responses. Purely local codes make it hard to generalise, because similar percepts may be encoded as entirely distinct patterns of activity.

Associative learning requires a middle ground: a transformation through which percepts that are similar in some ways, and distinct in other ways, are encoded as partially overlapping patterns. The overlap will support general associations, while other subsets of the code may be used for specific ones.

Such a transformation results in what is known as *sparse codes* (Olshausen and Field, 2004; Földiák, 2013): codes that are distributed (each stimulus is encoded over several units), but neither dense (because each unit responds only to a specific feature) nor local (because each unit participates in coding different stimuli).

From suspicious coincidences to interesting projections

Learning these sparse codes is an unsupervised problem. There is no *a priori* definition of which features should be used to encode a given set of stimuli, and not all types of features are equally

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

useful; some are trivial in the sense that they encode the input well in information-theoretic terms, but do not reveal any ecologically-relevant structure. Barlow (1987) suggested that the cortex might be looking for *suspicious coincidences*: patterns of activity that occur more often than one would expect if the coincidence was due to pure chance. This links the discovery of features to the concept of independent components[1] in statistical learning (Linsker, 1988; A. J. Bell and Sejnowski, 1995).

How can a neural network discover these suspicious coincidences? In the classical framework of artificial neural networks, each unit computes a projection $y = \varphi(\vec{w} \cdot \vec{x})$ of its inputs $\vec{x}$ onto its weight vector $\vec{w}$. The hunt for suspicious coincidences becomes a search for unusual projections. And since most random projections of high-dimensional data tend towards the Gaussian distribution (Diaconis and Freedman, 1984), it turns out that there is an effective heuristic: find those that deviate the most from Gaussianity.

That heuristic is used in Projection Pursuit and in BCM theory: a stochastic gradient descent rule tunes the weights of the neuron so as to maximise the tails of the output distribution (the third and fourth moments, skewness and kurtosis). This yields receptive fields that encode, for instance, the orientation of edges in natural images (Bienenstock et al., 1982; Cooper et al., 2004).

But Hebbian learning in its simplest, linear form $\Delta w \propto xy$ is not guaranteed to discover non-Gaussian projections. In many cases it will only find projections that maximise variance, such as principal components (Olshausen and Field, 1996; Cooper et al., 2004). To discover suspicious events requires a form of *nonlinear Hebbian learning* (Brito and Gerstner, 2016), so that inputs that are correlated with the tail of the activity distribution are potentiated preferentially.

One way to achieve this is to use a nonlinear Hebbian term of the form $\Delta w \propto x\,\Omega(y)$, where Ω is a non-monotonic function $\curvearrowright$ of the post-synaptic activity. An example is the the $y(y-\theta)x$ term in the BCM rule (Cooper et al., 2004), or the relationship between intracellular calcium levels and the direction of plasticity in biological neurons (Shouval et al., 2002).

[1] Although complete independence is neither likely nor required; it is enough if the mutual information between the patterns is lower than between their constituent dimensions. As pointed out by Graham and Field (2007), Independent Component Analysis is unlikely to find perfectly independent components in natural data.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

The same result can be obtained with a linear Hebbian term through the effect of the output nonlinearity φ, provided that it has a suitable shape. This is the mechanism used by Földiák (1990), Triesch (2007), Savin et al. (2010) and Zylberberg et al. (2011), among others.

In both cases, the nonlinearity must also be kept aligned with the activity distribution. This is usually done with an adaptive threshold that controls the inflexion points of the Ω or φ function.

The work of Brito and Gerstner (2016) sheds light on the reasons why such diverse models can produce similar results: individual neurons become selective feature detectors either because they define an explicit Ω term governing synaptic plasticity, or because the interaction of the plasticity rule with the somatic response introduces an implicit Ω relationship that constitutes the *effective Hebbian nonlinearity* of the model.

Competition

Nonlinear Hebbian learning explains how a single neuron can become a selective feature detector. A sparse coding network, however, needs to learn not just one feature detector, but a set of feature detectors that work together to encode the input. It is no use if all units learn the same feature; we need a mechanism that lets each neuron know whether it is sufficiently different from the others. One way is to have the neurons compete to respond, a paradigm known as Competitive Learning (Rumelhart and Zipser, 1985).

The simplest form of competition is to select a fixed number of winning neurons for each stimulus. This *winner-take-all* scheme is used, for instance, in the self-organising map (Kohonen, 1982). For a sparse coding network, it has the unwanted side-effect of fixing the number of features that encode each stimulus.

Instead, in the networks of Marshall (1990), Intrator (1990), Földiák (1990), and later models that use continuous (Falconbridge et al., 2006) or spiking neurons (Savin et al., 2010; Zylberberg et al., 2011; King et al., 2013), competition is implemented by recurrent inhibitory connections between the output units, also called lateral inhibition. These connections learn in a Hebbian

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

fashion[2] and since they are inhibitory, they act to decorrelate the activities of the output neurons. Initially similar receptive fields will drift apart — either they become narrower until they no longer overlap, or one of the cells picks up a different feature.

Discrete dendrites

The models mentioned so far mix the recurrent inhibition with the feedforward input before the output y is computed. The learning rule has no way to distinguish the two; it is only through deviations from mean activity — as measured by the adaptive threshold — that competition can shape the development of receptive fields.

There is, however, no reason why the feedforward input and the recurrent inhibition could not be counted separately, and used to control the learning in different ways.

The idea is biologically plausible. The majority of excitatory inputs reach the dendrites, whereas several studies have found a greater density of inhibitory synapses on the soma and axon (Andersen and Eccles, 1964; Spruston, 2008). The situation is in fact more complex than a simple dichotomy. The many classes of cortical inhibitory neurons target different locations on pyramidal cells, including the dendrites (Kubota et al., 2016). Nevertheless there are classes of interneurons that inhibit the somas specifically — basket cells are one example.

Our hypothesis is that some types of recurrent inhibition could therefore suppress somatic spikes without affecting the dendritic potentials. By comparing its own activity with the back-propagating action potentials it receives from the soma, a dendrite could thus estimate the amount of somatic inhibition and guess how many other cells are competing to respond to the same input — an information which could modulate learning in the dendrite: we know that inhibitory inputs can act as a switch for excitatory plasticity and that their effects depend on the location of inhibition (Bloss et al., 2016; Wilmes et al., 2016).

The idea, nonetheless, calls for a move away from point neurons, so that the feedforward and recurrent inputs can be distinguished. Körding and König (2000) proposed one such model, where

[2] Sometimes called *anti-Hebbian* because the connection is inhibitory. Nonetheless the *magnitude* of the weight changes according to the same law that governs excitatory plasticity in Hebbian theory. We prefer to reserve the term anti-Hebbian for cases where the more neurons A and B fire together, the less A influences the firing of B — as happens for instance in the cerebellum (C. C. Bell et al., 1997).

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

the blockage of backpropagating action potentials by lateral inhibition can switch the direction of plasticity at the dendritic synapses. More recently, Urbanczik and Senn (2014) described a two-compartment model based on the notion that the dendrite attempts to predict somatic spikes. Direct inputs to the soma are not seen by the dendrite, and this generates a prediction error that is used as a teaching signal for synaptic plasticity.

Here we apply this principle of dendritic predictions to a sparse coding network, and let the *effective Hebbian nonlinearity* emerge from somato-dendritic prediction errors caused by lateral connections. This puts lateral inhibition in direct control of the sparseness of the receptive fields, and competition can therefore decorrelate the activity of the neurons without the mediation of a sliding threshold.

2. Neural Model

Our network is a single layer of N neurons with separate dendrites (rate-based) and somas (spiking). Dendrites receive the feedforward input and function as feature detectors. Somas mediate the competition to respond: each soma receives the output of its respective dendrite and also receives direct inhibition and excitation from all the other neurons. Thus the feedforward input and the recurrent inhibition are spatially segregated: the first targets the dendrites, the other targets the somas.

Input vectors of length M are presented to the dendrites on a 50 ms clock. The resulting dendritic activation is kept constant while somas compete to respond over the 50 ms period. During that time spikes propagate through the recurrent pathway and evoke varying currents at the somatic synapses. At the end of the competition period, we count the spikes emitted by each neuron and apply the learning rules for both the feedforward and the recurrent synapses. Then the next input pattern is presented, etc.

We integrate all differential equations with the Euler method. The spike-based models (somas and somatic synapses) use a timestep $dt = 0.5\,\text{ms}$.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

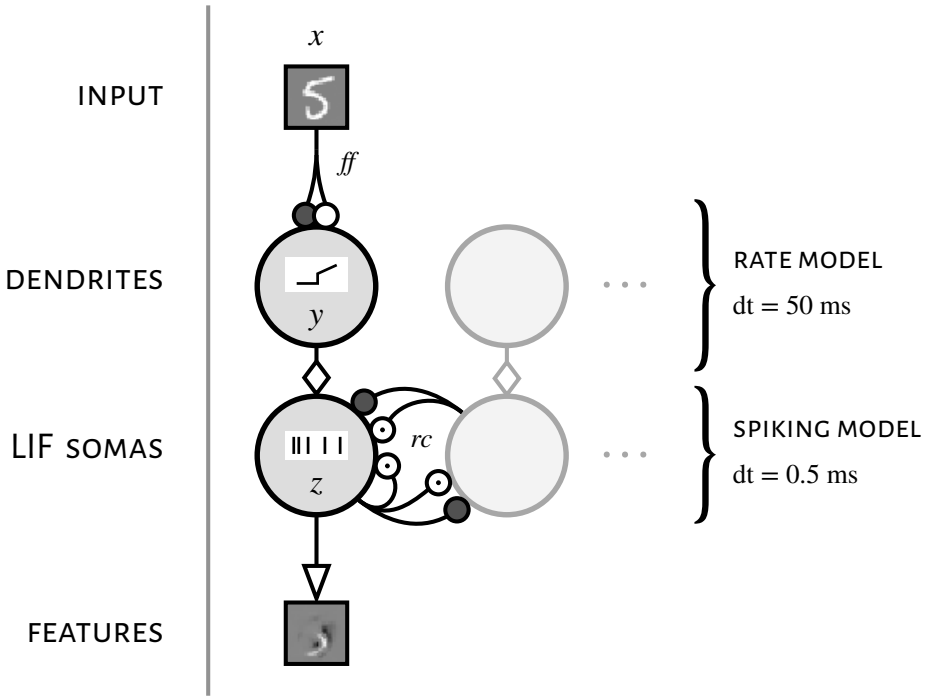


Figure 1. Structure of the network. ○ excitatory synapse; ● inhibitory synapse; ⊙ non-plastic excitatory synapse; *ff* feedforward pathway; *rc* recurrent pathway.

While it is common for spiking neuron models to be defined with units and quantities that match experimental data (mV / pS / nA) and to have a negative resting potential, here we follow the conventions of artificial neural networks: variables are dimensionless (except time) and the resting potential is set to 0.

Dendrites

In our model, the dendrites function like the neurons of a Perceptron-type network. They compute a projection of the feedforward input vector $\vec{x}$ onto their weight vector $\vec{w}_{ff}$, giving the dendritic activation $g = \vec{w}_{ff} \cdot \vec{x}$. Weights may take positive or negative values, corresponding to feedforward excitation and feedforward inhibition, respectively.

The dendritic output y is a nonlinear, non-negative function $\_\!\!\nearrow$ of the activation:

$$y = \varphi(g) = \begin{cases} 0 & \text{if } g < \theta_g \quad \text{(a)} \\ y_0 + y_1(g - \theta_g) & \text{if } g \geq \theta_g \quad \text{(b)} \end{cases} \tag{1}$$

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

The choice of that nonlinear transfer function aims to capture several aspects of dendritic responses in biological neurons, summarised in Antic et al. (2010).

First (eq. 1a), weak excitatory potentials propagate passively and attenuate before they reach the soma. Dendritic inhibition does not affect the somatic membrane potential either (Kubota et al., 2016).

Second (eq. 1b), inputs that reach a threshold trigger regenerative events, known as dendritic spikes or plateau potentials, which are able to elicit somatic spikes reliably.

Although plateau potentials have a constant, saturating peak amplitude (Polsky et al., 2004), their duration, and therefore the number of post-synaptic spikes evoked by the event, scales with the magnitude of the suprathreshold input (Milojkovic et al., 2005, fig. 2G). The variable y aims to capture that relationship, despite the fact that our model enforces a constant duration of 50 ms for plateau potentials: it models not the amplitude of dendritic spikes but rather the integral of the currents evoked by a stimulus.

As a result φ takes the form of a step-linear transfer function, capturing both the binary threshold effect and the proportional suprathreshold response. In practice we set the offset y_0 slightly above the rheobase of the soma so that at least one spike is elicited when $g = \theta_g$ (in the absence of somatic inputs), and the slope y_1 so that peak somatic responses consist of a burst of around five spikes.

The parameter θ_g controls the response threshold and can be set to zero to model a scenario where excitation and inhibition are in balance, and slight deviations from equilibrium are sufficient to trigger dendritic spikes. Higher values of θ_g yield sharper receptive fields.

Note that the initial distribution of the feedforward weights must be set so that naive dendrites can already respond to some patterns. This is because our model does not include the kind of developmental or homeostatic mechanisms that would ensure that all dendrites get sufficient inputs to bootstrap receptive field formation.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

Somas

We use a leaky integrate-and-fire (LIF) model for the somas. The membrane potential u varies as follows:

$$\tau_u \frac{du}{dt} = I_d + I_s - u \tag{2}$$

and decays to zero in the absence of inputs. I_d and I_s are the currents coming from the dendrites and from the somatic synapses, respectively. Each neuron has a single dendrite in the present model, therefore the total dendritic current is simply $I_d = y$.

A spike is emitted whenever $u \geq \theta_u$. This triggers a reset of the membrane potential: $u \leftarrow \rho$. A hard refractory period follows, and integration of the membrane potential is suspended for a duration $t_{ref} = 4\,\text{ms}$, during which $\frac{du}{dt} = 0$. Another reset happens before each new input: the membrane potential u and any pending refractoriness are set to 0.

The somatic output z computed over each competition period $t_0 \to t_1$ is a sum of the spikes emitted by the soma, weighted such that a spike that occurs at the start of the window contributes 1 to the total and a spike that arrives at the end of the window contributes 0:

$$z = \int_{t_0}^{t_1} \frac{S(t_0, t)}{t_1 - t_0} dt \tag{3}$$

where $S(t_0, t)$ is the number of spikes emitted between the times t_0 and t. This provides a continuous measure of somatic activity despite the quantization inherent to all-or-none action potentials.

Feedforward Synaptic Plasticity

Our feedforward plasticity rule is based on dendritic predictions of somatic activity, as in Urbanczik and Senn (2014). Before the competition, each dendrite computes a prediction z' of the somatic activity z that should result from the dendritic activation g, using a logarithmic approx-

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

imation of the LIF somas's response curve (I-f, or rather I-z in our model):

$$z' = \begin{cases} z_0 \ln\left(1 + z_1\left(g - \theta_g\right)\right) & \text{if } g \geq \theta_g \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We set the parameters z_0 and z_1 so that the dendritic prediction is fairly accurate in the absence of somatic inputs and at typical activity levels ($g - \theta_g < 3$). At the end of the competition period, the dendritic expectation is compared with the actual somatic activity z and the difference Ω governs the direction of synaptic plasticity in the dendrite:

$$\Omega = z - z' \tag{5}$$

The weight w_{ff} of each feedforward synapse is updated according to three different terms:

$$\begin{aligned} \Delta_{hebb} &= x\,\Omega && \text{nonlinear Hebbian term} \\ w_{ff} &\leftarrow w_{ff} + \eta_{ff}\,\Delta_{hebb} \\ \Delta_{dec} &= \kappa\, w_{ff} \max(\Omega, 0) && \text{decay term} \\ w_{ff} &\leftarrow w_{ff} - \eta_{ff}\,\Delta_{dec} \\ \Delta_{reg} &= \lambda\, g \max(\Omega, 0) && \text{regularisation term} \\ w_{ff} &\leftarrow w_{ff} - \text{sign}(w_{ff}) \min(\eta_{ff}\,\Delta_{reg},\ \text{abs}(w_{ff})) \end{aligned} \tag{6}$$

The Hebbian term is post-synaptically gated by Ω. The decay term keeps the weights bounded, and the regularisation term makes them sparse; these two terms are only applied when the dendrite is being potentiated ($\Omega > 0$). The learning rate is given by η_{ff}, and the relative weighting of the decay and regularisation terms compared to the Hebbian term is set by κ and λ, respectively.

The three updates are applied one after the other, because the regularisation term requires special care to avoid overshooting zero: the sign and magnitude of the weight must therefore be measured after applying the Hebbian and decay terms.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

A standard choice for a regularisation term that encourages sparseness would be to penalise the ℓ_1 norm of the weight vector. That would correspond to a biological mechanism based on the metabolic cost of keeping synapses functional. Since the norm of the weight vector is not readily available in our implementation, we use the activation g instead. That may correspond to one of several activity-induced mechanisms (such as NMDA receptor activation) known to exert a negative feedback on synaptic potentiation (Abraham, 2008).

Recurrent Synapses

Recurrent excitation and inhibition are conveyed by lumped conductance-based synapses. Each synapse has its own weight w_{rc}, but a single state variable r_{rc} models the total active conductance of all the synapses of the same class (excitatory/inhibitory) in each soma. The dynamics of r_{rc} are as follows:

$$\begin{aligned} \tau_{rc}\frac{d}{dt}r_{rc} &= -r_{rc} && \text{exponential decay} \\ r_{rc} &\leftarrow r_{rc} + w_{rc}(\textit{pre}) && \text{on spike from neuron } \textit{pre} \end{aligned} \tag{7}$$

The active conductance is reset to zero before every new input: $r_{rc} \leftarrow 0$. The synaptic current I_{rc} varies according to a scaling factor $\bar{g}_{rc}$, the somatic potential u and the reversal potential E_{rc}:

$$\begin{aligned} g_{rc} &= \bar{g}_{rc}\, r_{rc} \\ I_{rc} &= g_{rc}(E_{rc} - u) \end{aligned} \tag{8}$$

For inhibitory synapses, $E_{rc}^{inh} = 0$. For excitatory synapses, $E_{rc}^{exc} > \theta_u$. The total current flow from somatic synapses into the soma is $I_s = I_{rc}^{exc} + I_{rc}^{inh}$.

Recurrent somatic inhibition

Recurrent inhibition follows a global, all-to-all connection pattern with the exception of self-connections, which are not allowed. For the sake of simplicity these are direct connections between the somas of principal cells, without the mediation of inhibitory interneurons; these could

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

be added in future work, following the example of King et al. (2013) who complemented the network described in Zylberberg et al. (2011) with a separate inhibitory population.

Recurrent inhibitory synapses are plastic. The weight w_{rc}^{inh} of the recurrent synapse between each $(pre, post)$ pair of neurons is updated according to the following learning rule:

$$\Delta_{inh} = \begin{cases} z(pre)\, z(post)^2 - w_{rc}^{inh} & \text{if } z(pre) > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (9)$$
$$w_{rc}^{inh} \leftarrow w_{rc}^{inh} + \eta_{rc}^{inh}\, \Delta_{inh}$$

Recurrent weights are restricted to positive values, but the net effect is inhibitory because of the parameter E_{rc}^{inh} in eq. (8).

The rule is pre-synaptically gated, since weights can change only when the presynaptic neuron is active. It is not symmetrical: the rate of change of the weight from a weakly active neuron to a strongly active one ($z(post) > 1$) is greater than that of the reciprocal connection. The fact that the rule is not post-synaptically gated, unlike the feedforward learning rule, ensures that lateral inhibition to a silent cell will eventually vanish, allowing the cell to pick up a pattern of its own. This is critical for the self-organisation of the network, as discussed in Marshall (1990).

The learning rate η_{rc}^{inh} is set so that the effective modification speed for recurrent inhibitory synapses is much faster than that of feedforward synapses, otherwise the network is unstable. While very fast plasticity rates may seem implausible, there is some evidence for inhibitory plasticity operating on the order of seconds in the hippocampus (Hartmann et al., 2008). Fast effective plasticity is also plausible if it involves a large number of synapses (Yger et al., 2015).

Recurrent somatic excitation

The final step is to add recurrent somatic excitation to our model. Without it no receptive fields would develop, because the dendritic prediction z' could then only be (a) correct ($\Omega \approx 0$), in the absence of inhibition, or (b) overestimated ($\Omega < 0$), when the soma is inhibited. Recurrent excitation allows Ω to become positive, provided that recurrent inhibition is sufficiently low.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

These synapses are non-plastic; the synapse strength w_{rc}^{exc} between each $(pre, post)$ pair of neurons is constant and defined as follows.

For experiments where the neurons form a topographic map, the connectivity pattern of recurrent excitation is all-to-all within a local neighbourhood. w_{rc}^{exc} is given by a normalised Gaussian kernel:

$$w_{rc}^{exc} = \exp\left(-\frac{\text{dist}(pre, post)^2}{2\sigma_{topo}^2}\right)\left(\sum_j \exp\left(-\frac{\text{dist}(j, post)^2}{2\sigma_{topo}^2}\right)\right)^{-1} \tag{10}$$

where $\text{dist}(pre, post)$ is the Euclidean distance between neurons *pre* and *post* on the map, and σ_{topo} defines the size of the neighbourhood. Boundary conditions are periodic and self-excitation is allowed.

For all other experiments, the size of the neighbourhood is reduced to the point where *only* self-connections are allowed:

$$w_{rc}^{exc} = \begin{cases} 1 & \text{if } pre = post \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

In that case the effect of self-excitation is equivalent to a systematic underestimation of the soma's current-frequency response curve by the dendrite; the same outcome could be achieved without recurrent excitation through the parameters z_0 and z_1.

3. Results

Inhibitory control of the effective Hebbian nonlinearity

First let us look at how an *effective Hebbian nonlinearity* emerges in our model. The relevant term is the post-synaptic factor $\Omega = z - z'$ in the learning rule for the feedforward synapses. To study how Ω varies as a function of the feedforward and recurrent inputs, we isolate a single neuron and replace the dendritic activation g and somatic inhibition I_{rc}^{inh} with constant inputs that we can manipulate (we retain somatic self-excitation, but no lateral excitation). We then compute the Selectivity Index, as defined in Brito and Gerstner (2016), for the various combinations of values. This index measures the learning rule's preference for a heavy-tailed activity distribution

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

(Laplace) over a Gaussian distribution of the same mean and variance. The results are summarised in figures 2 and 3.

In the absence of somatic excitation or inhibition, dendritic expectations are more or less correct ($z' \approx z$) and the $\Omega(g)$ relationship is flat: ⎯. No learning takes place.

Now add somatic self-excitation. This creates more somatic activity than the dendrite expects, and the Ω term increases with g above the threshold: ⌒. This rule is only selective if the dendritic threshold θ_g is located towards the tail of the dendritic activity distribution (fig. 3, left). This is typically not the case in our network, because there is no mechanism in place to enforce that constraint and $\langle g \rangle$ will tend to overtake θ_g. As a result neurons with only self-excitation will acquire broad, unselective receptive fields.

Finally, add somatic inhibition. Ω becomes negative for low values of g, where recurrent inhibition is stronger than self-excitation: ∽. Increasing somatic inhibition shifts the zero-crossing point to the right (fig. 2), and widens the range of input current distributions for which selective receptive fields can develop (fig. 3, middle and right). This also means that more competition will result in narrower receptive fields, providing a mechanism through which the network can self-organise to cover the entire feature space.

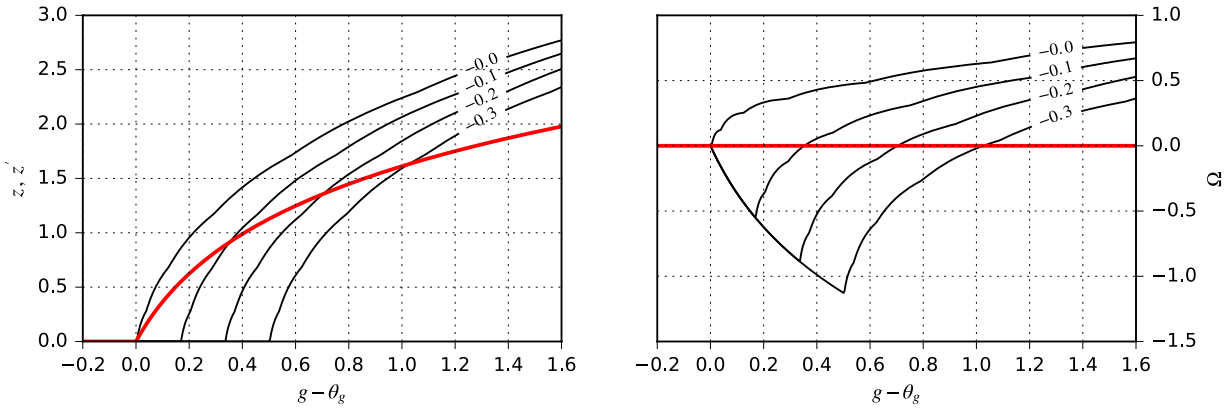


Figure 2. Effect of somatic excitation and inhibition on the Hebbian nonlinearity. Each labeled curve corresponds to a particular value of I_{rc}^{inh}. Dendritic prediction in red, actual somatic values in black. **Left:** Because of self- excitation, the somatic response (z) is steeper than predicted by the dendrite (z'), while somatic inhibition shifts it to the right. **Right:** The prediction error Ω produces the effective Hebbian nonlinearity. A smaller timestep $dt = 0.005$ ms was used while generating this figure, to make the curves smoother.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

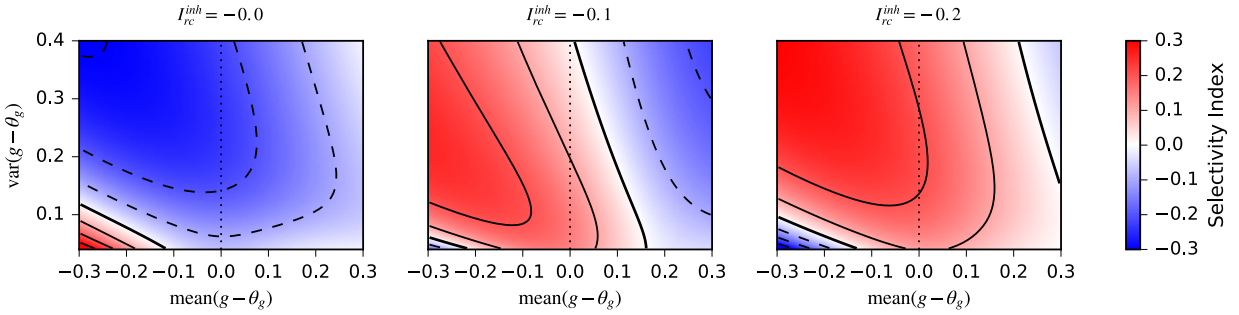


Figure 3. Selectivity Index for various levels of somatic inhibition and dendritic current distributions. Positive values indicate rules that can form selective receptive fields.

Solving the Bars Problem

We revisit a task used notably by Földiák (1990) to test their network. The input patterns in that task consist of horizontal and vertical bars; each bar can be present or absent with a certain probability and independently of others bars. This is considered a nonlinear ICA problem, because the bars overlap non-additively: pixel values are zero where there is no bar, and one where there is at least one bar.

The expected behaviour of the network is straightforward. The independent components of the input are individual bars, therefore there should be at least one neuron that codes specifically for the presence of each particular bar.

We consider three variants or perturbations of the task, which were also explored in Spratling (2011). The first variant adds Gaussian noise to each pixel, with the final value clipped to the range $[0, 1]$. This amounts to a substantial corruption of the input patterns; the author himself struggles to discern individual bars. The second variant introduces unequal probabilities for horizontal and vertical bars, and the third variant changes the shape of the pattern from a square to an elongated rectangle, so that horizontal bars are longer than vertical bars. These test for a possible failure mode of competitive learning models, where the largest or most frequent features invade the entire coding space and the network becomes blind to other patterns. Finally we also combine all three variants together (figure 4).

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

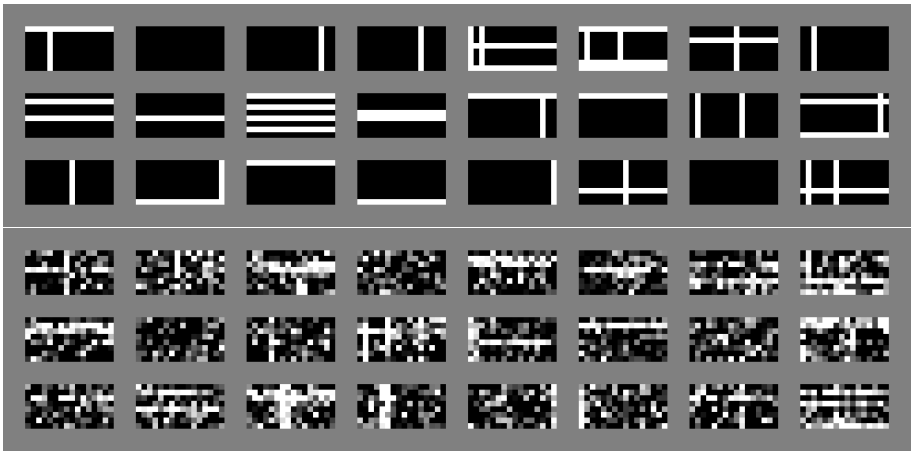


Figure 4. Sample 8×16 bars patterns. Bar probabilities: 0.12 (horizontal), 0.06 (vertical). **Top:** without noise. **Bottom:** with Gaussian noise ($\mu = 0, \sigma^2 = 0.3$).

The network succeeds in learning a full set of single bars in all cases, even when all the perturbations are combined (figure 5). Learning speed is affected by the various perturbations: smaller bar sizes, lower bar probabilities and higher noise variance all delay the formation of the receptive fields, considerably so in the case of the noise. Note that throughout this paper we use the feed-forward weights of each neuron as a proxy for its receptive field; this did not differ appreciably from measures based on spike-triggered averages.

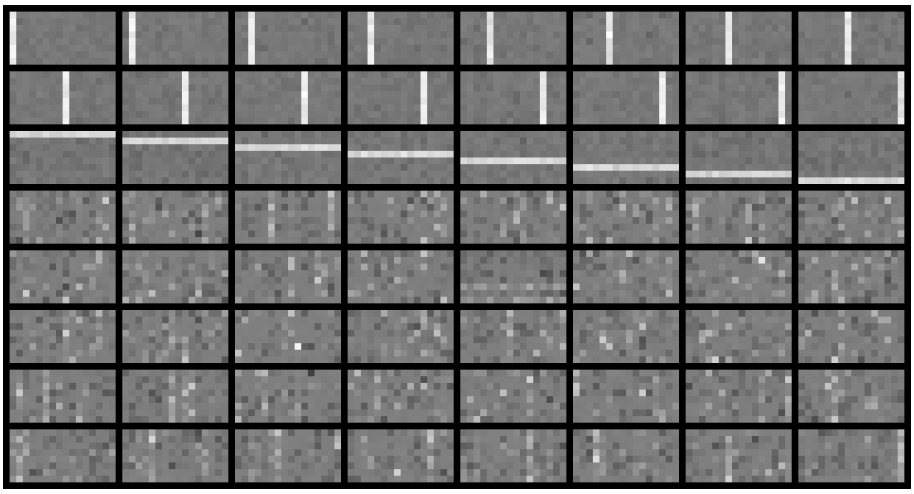


Figure 5. Receptive fields (sorted by type) learned by a network ($N = 64$) trained on 36,000 noisy patterns. Some neurons learn the full set of 24 bars (top rows), the other receptive fields are mostly random. This network uses stronger lateral inhibition than in the rest of the paper to avoid redundant bars: $\bar{g}_{rc}^{inh} = 30/N$.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

Learning from MNIST handwritten digits

We now turn to a more complex dataset, the MNIST database (LeCun and Cortes, 1998). This dataset consists of 70,000 grayscale images of handwritten digits that have been normalised and centered in a 28x28 pixel frame. It is divided into a training set of 60,000 images and a testing set of 10,000 images; we use only the training set for learning receptive fields. Each pattern is shown only once during training, unless otherwise indicated. We apply random translations in the X and Y axis to each pattern, drawn from the discrete uniform distribution $\mathcal{U}\{-1, 1\}$. A sample of the training patterns is shown in figure 6. The MNIST dataset contains contiguous blocks of digits by the same writer; we do not randomise these away, because such runs occur in natural data and online learning algorithms should be able to deal with them. Instead we randomise the starting position.

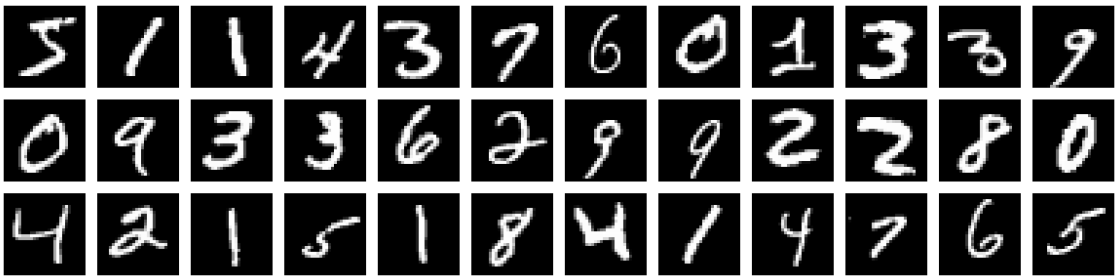


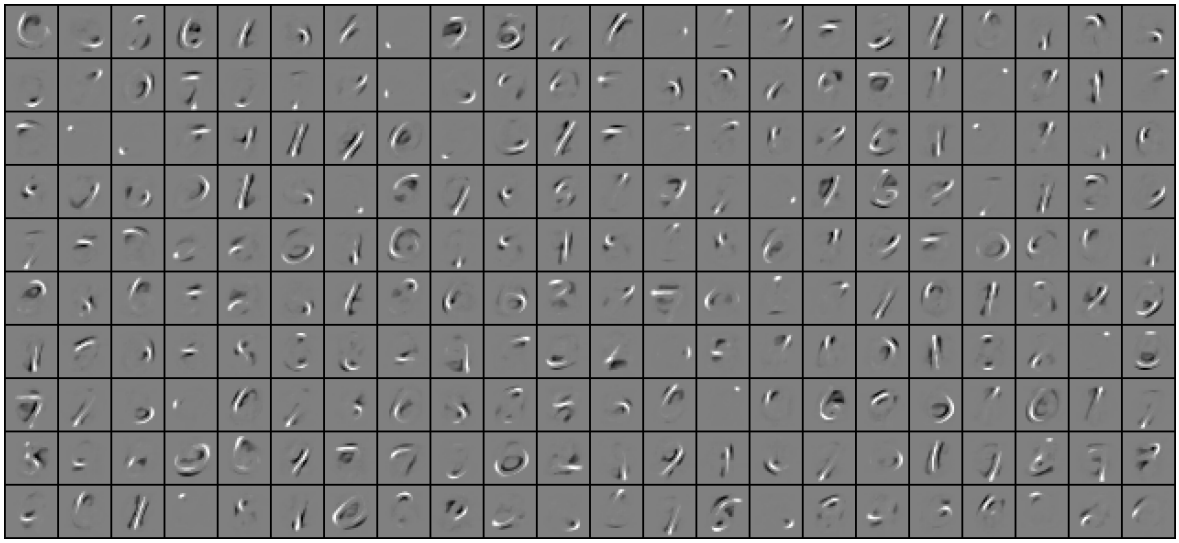
Figure 6. Sample of the MNIST training patterns.

The MNIST dataset is more complex than the bars problem of Földiák. There is more variability and it is not known *a priori* what the independent components of handwritten digits are. One can expect that each pattern must be decomposed into a greater number of components, and that these components overlap with each other more, than in the bars problem.

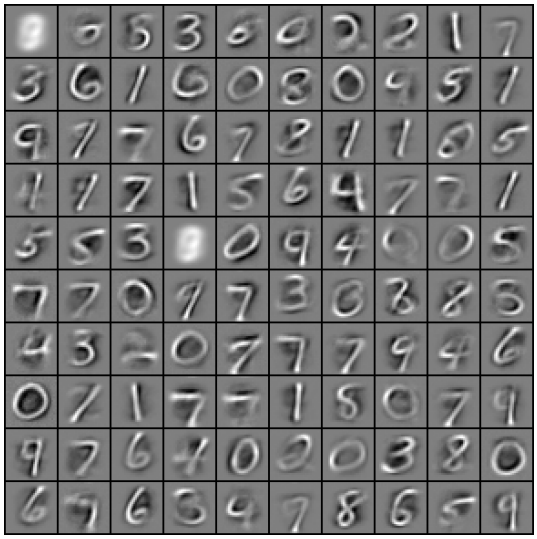
The results are shown in figure 7. The independent components learned by our network resemble pen strokes: straight lines delimited by inhibitory margins and curved lines surrounding an inhibitory center. A few pixel-like blobs appear in the periphery of the 28x28 frame, where data is rare since digits are centered. The early stages of receptive field development give us a clue as to how these pen strokes form: each receptive field starts as a more-or-less complete digit and

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

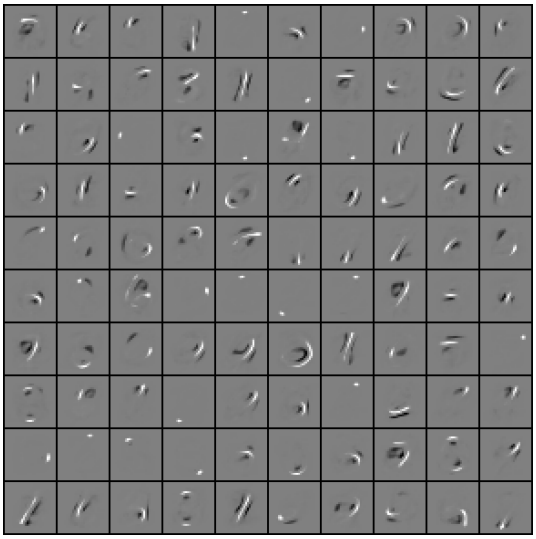
then shrinks to a smaller fragment that is also part of other digits (figure 8). This process depends primarily on the regularisation parameter λ: the higher the value, the smaller and simpler the receptive fields.



(a) $\lambda = 0.1$ (default)



(b) $\lambda = 0.0$



(c) $\lambda = 0.2$

Figure 7. Receptive fields after presenting 60,000 MNIST digits. Each plot shows a subset of a network ($N = 484$) trained with different values of the regularisation parameter λ. Receptive fields are normalised individually: middle gray is neutral, lighter colours are excitatory, darker colours are inhibitory.

Interestingly, these features are similar to those extracted by various implementations of sparse autoencoders, for instance Makhzani and Frey (2013), despite significant differences in the network architecture and learning rule.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

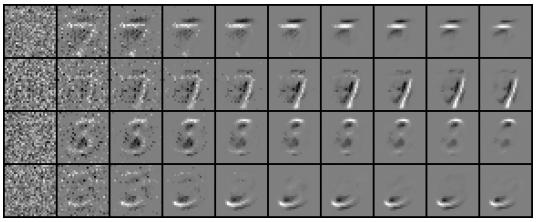


Figure 8. Evolution of four selected receptive fields during training. In the early stages (left) one can recognise individual digits (7, 3). After training (right) the receptive fields become selective for smaller components that are found in several classes of digits.

The network's activity is sparse and average firing rates differ across neurons, as shown in figure 9. Feedforward weights converge smoothly. The fine temporal structure visible in the spike raster reflects the blocks of digits written in the same hand, and disappears when the dataset is randomised.

Our implementation takes 2 to 3 minutes to train a network of 484 neurons on 60,000 MNIST patterns, using one core of an Intel® Core™ i7-2635QM and single-precision numerical kernels, compiled to SIMD vector instructions with the ispc compiler (Pharr and Mark, 2012).

Reconstructing MNIST digits

With the MNIST dataset, evaluating the quality of the learned code is not as simple as counting bars in the network's receptive fields. Instead, we come back to the idea that the goal of sparse coding is to produce a transformation of the input that preserves information while improving decodability, linear decodability in particular. First we ask how well a linear model can reconstruct the input patterns from the network's output.

The process has four separate phases: (1) First, we train the network's receptive fields with one presentation of the training set. (2) Then, we disable weight updates and go through the training set once more while recording the network's responses. This generates 60,000 feature vectors. (3) Using stochastic gradient descent, we compute regression coefficients Q for a linear model that maps these feature vectors back to the corresponding input patterns. (4) Finally, we test the reconstruction performance using the 10,000 patterns of the testing set: each test pattern is first

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

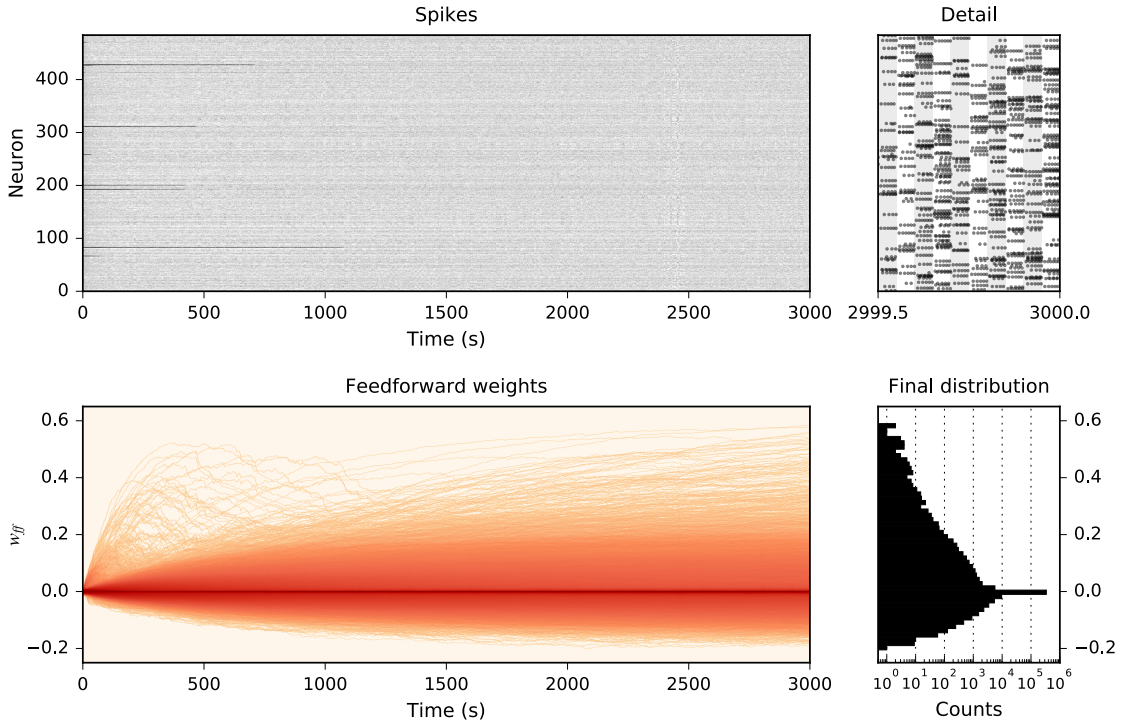


Figure 9. Spike raster (top) and evolution of the feedforward weights w_{ff} (bottom) during the presentation of the 60,000 training pattern (20 patterns per second of simulated time). The alternating bands in the detail plot indicate successive input patterns. The plots on the left use nonlinear color maps (spikes: $d^{0.5}$, weights: $log(1 + d^{0.6})$ to accommodate the high dynamic range.

presented to the sparse coding network, and we use the resulting feature vector to reconstruct the input using the coefficients learned at step 3.

The reconstruction for each pixel i is $x'_i = \max\left(\sum_j (Q_{ij} z_j)\,,\, 0\right)$ and the gradient descent rule is as follows:

$$Q_{ij} \leftarrow Q_{ij} + \eta_q (x_i - x'_i) z_j \tag{12}$$

with the learning rate $\eta_q = 1 \times 10^{-3}$.

Quantifying the reconstruction quality for our images of handwritten digits is difficult: pixel-wise errors that affect the underlying structure of the digits should be given a higher weight than those that don't, but we lack a formal definition of that underlying structure — in fact, that is precisely what our network is trying to discover. We could attempt to compute some measure of mutual information between the input patterns and the reconstructions, but that is

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

not straightforward in the multivariate case with high-dimensional data. Instead we fall back to showing a small subset of the patterns for visual comparison (fig. 10).

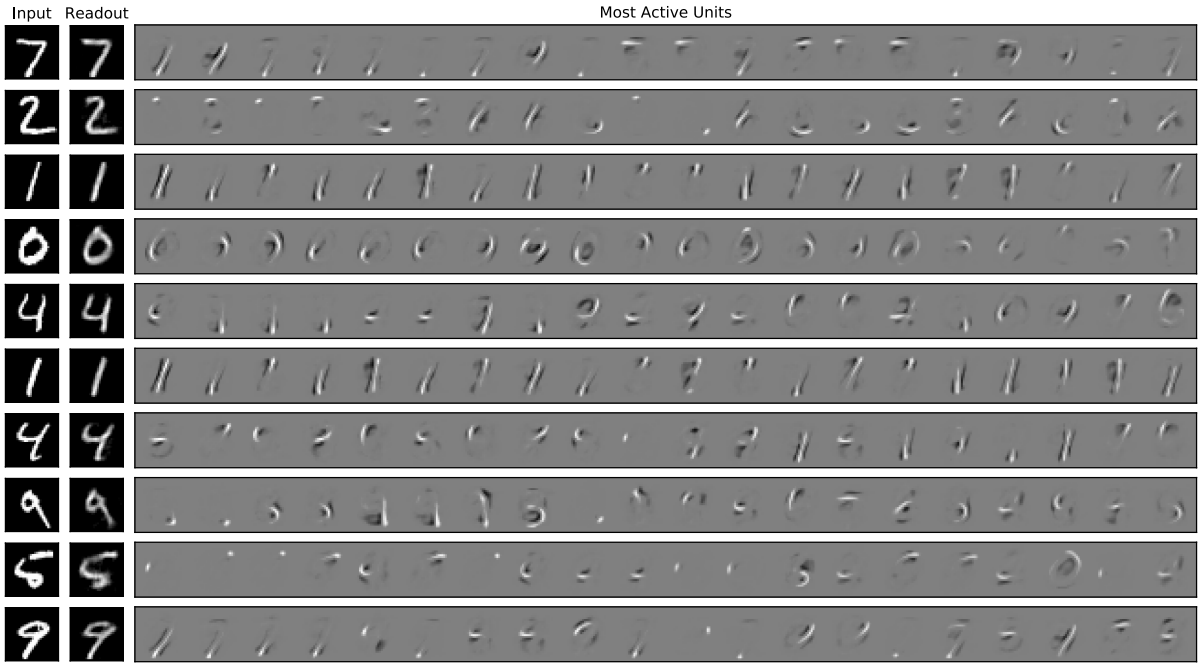


Figure 10. Sample reconstructions of MNIST digits, showing the receptive fields of the 20 most active neurons (sorted and normalised by firing rate).

Classifying MNIST digits

The second experiment tests whether the learned encoding is suitable for associative learning, and in particular whether it is linearly decodable: we compare how well a linear classifier can associate digit labels to the feature vectors, versus the raw input patterns.

The MNIST database has been used as a benchmark for classification algorithms, therefore we can also compare the performance of our model to that of others. However, our intent here is not to advance the state of the art in classification, but merely to test whether the sparse code learned by our network is more linearly separable than the input patterns.

The method is the same as in the reconstruction task, but we train a linear Support Vector Machine (SVM) to predict the class labels instead of reconstructing the input patterns. We use the LinearSVC implementation from scikit-learn (Pedregosa et al., 2011) with the default parameters

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

	Linear SVM	kNN
raw input	8.2 ± 0.0 %	2.8 %
features ($N = 484$)	2.6 ± 0.1 %	3.6 ± 0.1 %
features ($N = 784$)	2.6 ± 0.2 %	3.4 ± 0.1 %
features ($N = 784$, $\sigma_{topo} = 0.5$)	3.8 ± 0.2 %	
features ($N = 784$, random)	5.1 ± 0.2 %	

Table 1. Error rate for various classifiers and input transformations (for non-deterministic methods we show the mean and standard deviation over 10 runs, rounded to 1 decimal).

(ℓ_2 penalty, squared hinge loss). We also evaluate a k-Nearest Neighbours (kNN) classifier ($k = 3$, weighted by Euclidean distance). A non-parametric method, kNN is often effective despite its simplicity; its main drawback is that it is very expensive in computations and in memory. Here it serves as a baseline and as a control for parametric methods.

There is one potential confound to be aware of: the possibility that our network improves classification performance not because it learns a particularly good encoding of the input, but simply because it adds a nonlinear projection step. This is how the kernel trick improves the performance of a nonlinear SVM, and we are reminded of a long history of neural networks that make use of random hidden weights (Wang and Wan, 2008). To control for this we also need to run a network without coherent receptive fields. We tested several configurations of untrained networks and networks where the weights were shuffled after learning; the one that performed best had random feedforward weights (w_{ff} as per the initial distribution) and no recurrent inhibition ($w_{rc}^{inh} = 0$).

Results can be seen in table 1. The combination of sparse features and linear SVM has a lower error rate than the linear SVM alone. According to the results collected by LeCun et al. (1998), that makes it comparable to a multilayer Perceptron with about the same number of neurons and two hidden layers. It is expectedly worse than convolutional networks — these can take advantage of spatial invariants, which our network ignores. As for random nonlinear projections, they are indeed better than the raw input, but not as good as trained features; this confirms that our learning rule is doing something useful.

In contrast, the error rate of the kNN classifier is worse when using the feature vectors. We do not analyse that result further since our focus is on linear decodability, but offer a tentative

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

explanation: sparse codes do not help the classifier because the kNN algorithm does not rely on linear separability; the transformation is also not completely lossless.

Learning topographic maps

In several mammalian species, the neurons of the primary visual area are organised into feature maps with smoothly varying receptive fields (Hubel and Wiesel, 1962; Mountcastle, 1997). Some neural network models generate similar maps through a center/surround organisation of the lateral connectivity: a neuron receives net excitation from its immediate neighbours and net inhibition from neurons further away, in a way that is reminiscent of the self-organising map of Kohonen (1982). See for instance the work of Linsker (1986), or, more recently, Butko and Triesch (2007) and Stevens et al. (2013).

Our model already incorporates recurrent excitation, therefore we can easily check whether imposing a topographic pattern on these connections, as per equation (10), would also yield feature maps.

The results are shown in figure 11. The receptive fields are indeed organised into patterns that resemble the pinwheels of smoothly varying features seen in cortical maps. Individual receptive fields are not obviously different from those learned with self-excitation only, but they seem less diverse. Classification performance is also worse compared to the non-topographic case (table 1).

Response to input deprivation

In networks with a sliding threshold, removing the feedforward input induces a compensatory adaptation of the threshold that eventually causes the neurons to respond to background noise, erasing their receptive fields and inducing large transients when the feedforward input is re-established.

Since our model lacks an adaptive threshold, one might wonder how the network reacts to the same scenario. We train a network on the MNIST dataset, this time with additive Gaussian

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

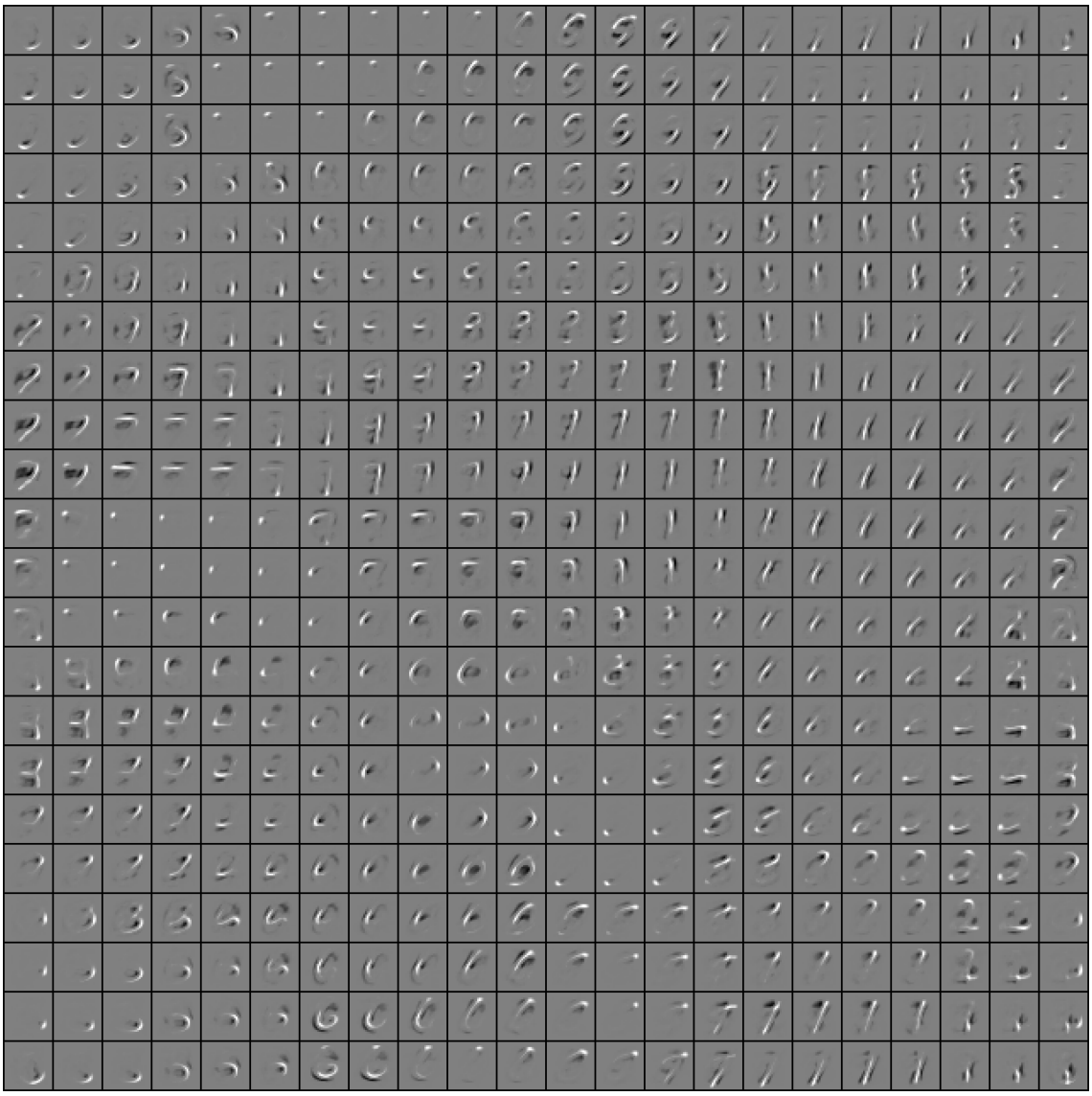


Figure 11. Receptive fields learned with topographic recurrent excitation ($N = 484, \sigma_{topo} = 0.5$)

noise. Once receptive fields have stabilised, we interrupt the stream of MNIST digits, showing only background noise for a period of time before re-establishing the normal input.

The input deprivation protocol does induce a compensatory response in our network (figure 12). This is evident when comparing the spike raster plots just after the input is turned off, and just before it is turned back on: most neurons eventually starting responding to the background noise. Receptive fields start to fade but their structure is preserved; they recover quickly when the input is re-established. Activity transients are limited in duration and amplitude: sparse re-

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

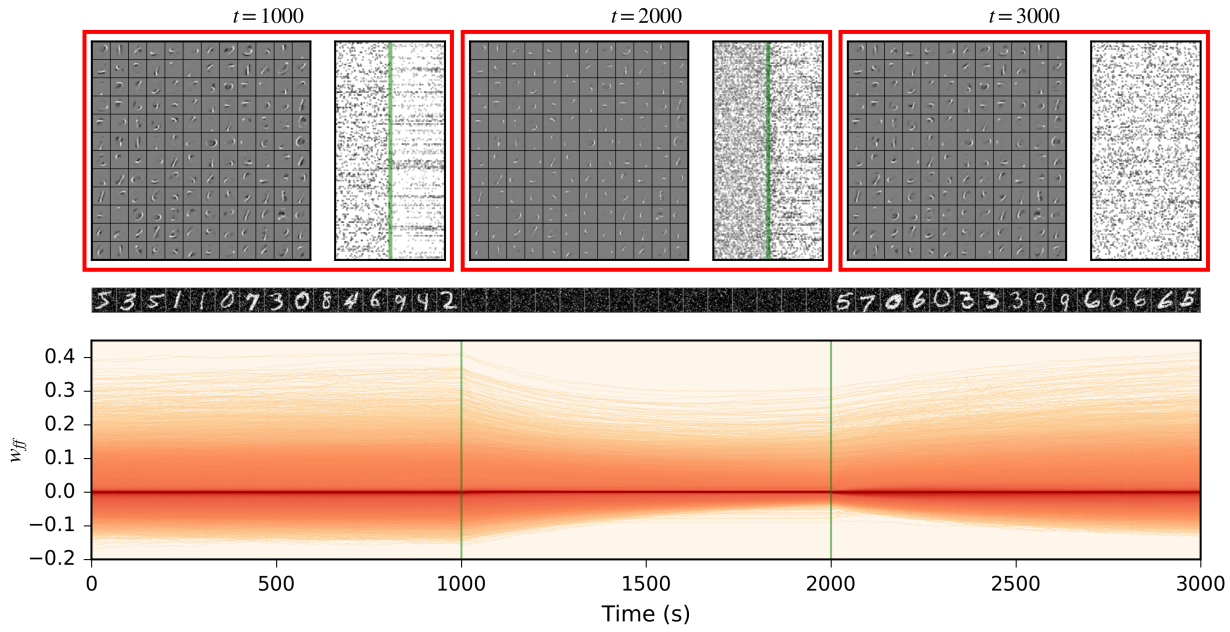


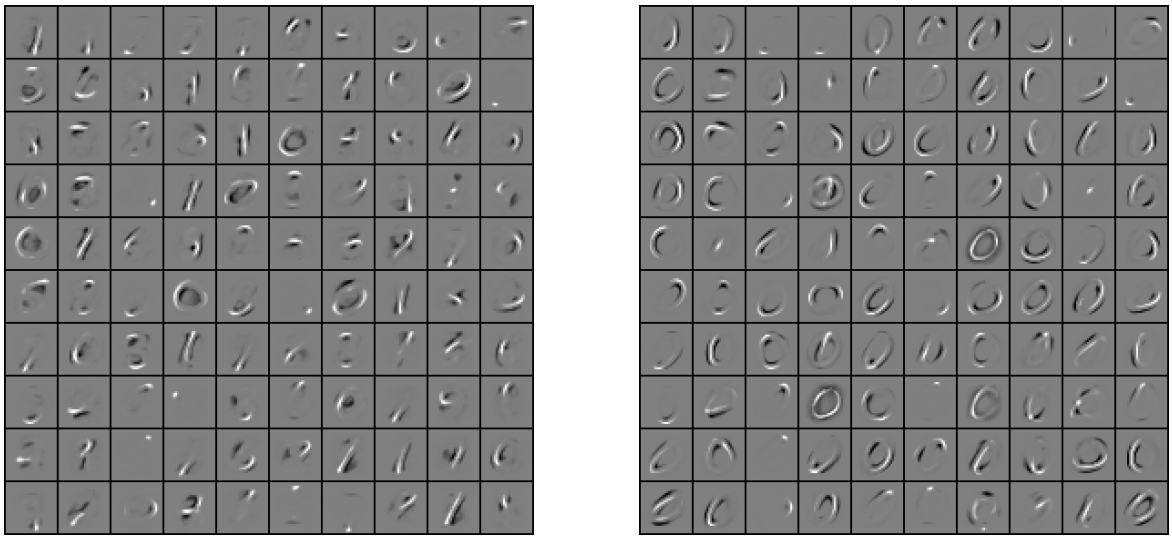
Figure 12. Response of a pre-trained network to input deprivation in the presence of noise. MNIST digits with a Gaussian noise overlay ($\mu = 0.0$, $\sigma = 0.3$, final value clipped to $[0, 1]$) are presented as usual until $t = 1000$ and after $t = 2000$. In-between, only noise is presented to the network. **Top:** receptive fields and activity of the first 144 neurons of the network at selected points in time. Spike raster plots show 10 seconds of activity. **Middle**: sample input patterns. **Bottom:** evolution of the feedforward weights.

sponses are retained throughout the transitions. Note that without noise no compensation would occur, because weights don't change unless $z > 0$ or $y > 0$.

Adaptation to changes in input statistics

The situation is different when, instead of just turning off the input, we alter its underlying structure. One kind of change is to freeze the input and present the same pattern over and over again. This causes a small number of neurons to change their receptive fields to match that pattern, with the other neurons remaining inactive. On the other hand, if the whole training dataset is replaced by another one with different statistics, extensive reorganisation occurs, as shown in figure 13.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:



(a) before (b) after

Figure 13. Left: sample receptive fields learned on the full MNIST dataset. **Right:** receptive fields of the same neurons after further training on zeroes only.

4. Discussion

Sparse coding without adaptive thresholds

In sparse coding models that use a sliding threshold, inhibition modulates the net activity ahead of the nonlinearity. The adaptive threshold adjusts to these variations, so that the nonlinearity stays aligned with the activity distribution. But fast, step-like changes — for instance turning the input on and off — will leave it saturated until the threshold catches up.

This leads to a dilemma of time scales that is particularly problematic when the sliding threshold controls not just the Hebbian nonlinearity as in the BCM rule (Bienenstock et al., 1982), but also the excitability of the neuron (Földiák, 1990; Triesch, 2007). The threshold must adapt slowly compared to the interval between recurrences of the feature, otherwise the neuron will give false positives as it strives to find something to respond to. Yet it must respond quickly to changes in input statistics, because as long as the network remains in a saturated regime its output will not be reliable.

In contrast, our model — like the BCM rule — has a fixed response threshold. The selectivity of the neurons depends mainly on their dendritic receptive fields; these adapt at a speed set by the

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

feedforward learning rate η_{ff}, which may be made as slow as required to get a good sample of the input. The average firing rates and population sparseness, on the other hand, depend on recurrent inhibition, which adapts in seconds and keeps the network functional throughout perturbations.

But the BCM rule still uses an adaptive threshold to adjust the Hebbian nonlinearity. That is because BCM theory started as a model of a single neuron — *something* was needed to control the nonlinearity and ensure that it would latch onto one of the independent components.

What we show here is that the adaptive threshold becomes redundant in a network. Lateral inhibition can play the same role, provided that it is plastic and that the feedforward learning rule lets it control the effective Hebbian nonlinearity directly. Selectivity is a single-neuron phenomenon, but one that is enabled by competition — it is because other units compete to respond that a neuron can afford to become silent most of the time. Outside of a network, our neuron model reverts to the strategy that will transmit the most information about the input: learning the principal component.

On the role of gating mechanisms

Does this mean that there is no role for adaptive thresholds and intrinsic plasticity in sparse coding networks? They are not an absolute requirement for the development of selective receptive fields. But they could be still be useful for other aspects of learning, in particular whenever there are slower changes that must be compensated — for instance developmental changes related to the growth of neurons and synapses, or changes in the network's operating conditions caused by glial cells or diffusing substances such as nitric oxide.

More generally, our results highlight the need for an adaptive *gating* of plasticity. There is more to learning online than just using stochastic gradient descent. When learning happens in realistic conditions, we can no longer assume that the input data will be properly randomised over the timescale of plasticity; it may linger in small areas of the feature space for long periods of time. Our results show that this will result in extensive reorganisation of the network — a

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

desirable effect in some cases, when it is called adaptation, but a catastrophe in many others, when it is called forgetting, as per the old stability-plasticity dilemma.

The solution is probably not as simple as having the learning slow enough that all these perturbations average out: there is a role for gating mechanisms, both intrinsic and external. These could take the shape of critical periods for learning, conditional consolidation of synaptic changes (Redondo and Morris, 2011), neuromodulation and attention (Hasselmo, 1995; Krichmar, 2008), and rules based on the mismatch between top-down and bottom-up inputs (Grossberg, 1980).

Sparse coding with spikes

Sparse coding has been studied with continuous neurons (Földiák, 1990; Olshausen and Field, 1997) and spiking neurons (Savin et al., 2010; Zylberberg et al., 2011), with fairly similar results. Is there any advantage to the use of spiking neurons then, beyond demonstrating how spiking neurons can perform sparse coding?

We argue that the main benefit is speed. Lateral inhibition is particularly expensive to compute in large continuous-time recurrent networks that must be iterated to convergence for every input pattern, like the network of Földiák (1990). The process is not unlike solving a continuous Hopfield network, which can require thousands of iterations with the Euler method (Talaván and Yáñez, 2005). In contrast, spiking neurons communicate with their neighbours only a small fraction of the time. We speculate that this may lead to a faster convergence of the competition.

At any rate, a spiking network simulator can take advantage of the sparse activity to speed up the simulation, even with a fully-connected network. With or without neuromorphic hardware, spiking models could therefore be of some use in machine learning.

Topographic maps

As the primary visual area (V1) has been an inspiration for sparse coding models, it is natural that some also aim to reproduce the topographic maps that can been seen in V1 in a number of

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

mammalian species. Yet we find that enforcing a topographical organisation through short-range lateral excitation does not improve the quality of the code. In fact, that may encourage redundant receptive fields and reduce the coverage of the feature space. The specifics of our model may be at fault; more experiments are needed to conclude on that topic.

But should sparse coding models try to produce maps at all? Feature maps are appealing to us, human observers, because they helps us make sense of cortical representations. That does not mean the brain has them for the same reason. Some mammals (such as grey squirrels) seem to do just fine without them:

> These results suggest that [...] an orientation map is not essential for strong orientation tuning. We suggest that an orderly arrangement of functional properties is not a universal characteristic of cortical architecture. (Van Hooser, 2005)

Besides, while map models assume a Mexican hat pattern of short-range excitation and long-range inhibition, the cortex, it seems, wears its hat upside down: studies show that the range of inhibition is actually smaller than that of excitation (Muir and Cook, 2014). Chklovskii and Koulakov (2004) suggest that cortical maps may simply be the side-effect of an optimisation to minimise wiring length. Would it be possible to reproduce the formation of maps with a developmental rule of that sort? Maps that develop through the creation and pruning of synapses may yield a better encoding than maps that emerge through the influence of lateral connectivity.

On linear separability and associative learning

Our network learns a sparse code that improves the performance of a linear classifier. That classification task is an artificial one, in the sense that the class labels are far removed from any sensory modality. Nonetheless, the improvement in linear separability should be broadly useful to many associative paradigms, such as the learning of predictive and causal relationships across sensorimotor modalities, or the association of a stimulus with an adapted response.

There is a case for linear separability as a coding principle in the brain. It is not very plausible, in terms of resource economy and scaling properties, that the process of decoding the information

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

converging to each pyramidal cell from other parts of the cortex requires multiple neurons. It is more likely that this process can be performed entirely within the dendritic tree of the target neuron, acting as a small cascade of linear filters and nonlinearities (Häusser and Mel, 2003; Poirazi et al., 2003). This would require each item of distal context to be linearly separable, with the cascade structure learning to respond to specific *combinations* of these items. Sparse codes fulfil the first condition; the second one calls for new models of integration across dendritic branches.

Legenstein and Maass (2011) showed for instance how branch-specific potentiation can lead to competition between dendrites that allows a single neuron to respond to several overlapping patterns, but not to shuffled combinations of their parts. As the concept of somato-dendritic predictions can be generalised to multiple dendrites (Schiess et al., 2016), it would make sense to combine these mechanisms with sparse coding in the same neuron, modelling the basal and apical dendritic domains of pyramidal cells. This may allow the simultaneous discovery of features and of associative relationships between features in a single layer.

In the present work we do not consider the temporal structure of the input. But the model we describe is similar in structure to other models of single-neuron processing that do take it into account. For instance, the linear-nonlinear-Poisson (LNP) model (Brunel et al., 2014) also contains a static nonlinearity that feeds into a spike generation mechanism, and adds a temporal filter at the level of the dendrite and synapses. The similarity suggests further experiments in that direction. Another possibility would be to use delays in recurrent excitation to learn pattern transitions, as suggested in Rodriguez et al. (2004).

Acknowledgments

The research behind this work was partly funded by the GRK 1589/1 and 1589/2 *Sensory Computation in Neural Systems* of the Deutsche Forschungsgemeinschaft. It also received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 609465.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

We thank Xavier Hinaut, Oswald Berthold and Walter Senn for their comments on the manuscript, as well as SIP Café in Bordeaux for their psychological support while it was being written.

Annex: Model Parameters

These parameters are common to all figures unless otherwise indicated in the figure or caption. Literal values are not rounded. Single precision (32-bit floats) is assumed throughout the model.

Dendrites		Somas		Feedforward Synapses		Recurrent Synapses	
θ_g	0.0	τ_u	10 ms	η_{ff}	1.5×10^{-3}	η_{rc}^{inh}	1.5×10^{-1}
y_0	1.005	θ_u	1.0	κ	1.0	$\bar{g}_{rc}^{exc}$	0.3
y_1	0.6	ρ	0.0	λ	0.1	$\bar{g}_{rc}^{inh}$	$6/N$
z_0	0.9	t_{ref}	4 ms	w_{ff} (initial)	$\mathcal{N}\left(\mu = 0, \sigma = \frac{2\sqrt{\pi}}{M\sqrt{2}}\right)$	τ_{rc}^{exc}	40 ms
z_1	5.0					τ_{rc}^{inh}	5 ms
						E_{rc}^{exc}	3.0
						E_{rc}^{inh}	0.0
						w_{rc}^{inh} (initial)	$Exp(\beta = 0.01)$

References

Abraham WC. 2008. Metaplasticity: tuning synapses and networks for plasticity. *Nature Reviews Neuroscience* 9:387–387. DOI: 10.1038/nrn2356

Andersen P., Eccles JC. 1964. Location of postsynaptic inhibitory synapses on hippocampal pyramids. *Journal of neurophysiology*.

Antic SD., Zhou W-L., Moore AR., Short SM., Ikonomu KD. 2010. The decade of the dendritic NMDA spike. *Journal of Neuroscience Research* 88:2991–3001. DOI: 10.1002/jnr.22444

Barlow H. 1987. Cerebral Cortex as Model Builder. In: *Matters of Intelligence*. Dordrecht: Springer Netherlands, 395–406. DOI: 10.1007/978-94-009-3833-5_18

578 Bell CC., Han VZ., Sugawara Y., Grant K. 1997. Synaptic plasticity in a cerebellum-like structure
579 depends on temporal order. *Nature* 387:278–281. DOI: 10.1038/387278a0

580 Bell AJ., Sejnowski TJ. 1995. An Information-Maximization Approach to Blind Separation and
581 Blind Deconvolution. *Neural Computation* 7:1129–1159. DOI: 10.1016/0165-1684(91)90081-S

582 Bienenstock EL., Cooper LN., Munro PW. 1982. Theory for the development of neuron selectivity:
583 orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience* 2:32–
584 48.

585 Bloss EB., Cembrowski MS., Karsh B., Colonell J., Fetter RD., Spruston N. 2016. Structured
586 Dendritic Inhibition Supports Branch-Selective Integration in CA1 Pyramidal Cells. *Neuron*
587 89:1016–1030. DOI: 10.1016/j.neuron.2016.01.029

588 Brito CSN., Gerstner W. 2016. Nonlinear Hebbian Learning as a Unifying Principle in
589 Receptive Field Formation. *PLoS Computational Biology* 12:e1005070. DOI: 10.1371/jour-
590 nal.pcbi.1005070.t001

591 Brunel N., Hakim V., Richardson MJ. 2014. Single neuron dynamics and computation. *Synaptic*
592 *structure and function* 25:149–155. DOI: 10.1016/j.conb.2014.01.005

593 Butko NJ., Triesch J. 2007. Learning sensory representations with intrinsic plasticity. *Neurocom-*
594 *puting* 70:1130–1138. DOI: 10.1016/j.neucom.2006.11.006

595 Chklovskii DB., Koulakov AA. 2004. Maps In the Brain: What Can We Learn from Them? *Annu.*
596 *Rev. Neurosci.* 27:369–392. DOI: 10.1146/annurev.neuro.27.070203.144226

597 Cooper LN., Intrator N., Blais BS., Shouval HZ. 2004. *Theory of Cortical Plasticity*. World Scientific.

598 Diaconis P., Freedman D. 1984. Asymptotics of Graphical Projection Pursuit. *The annals of statis-*
599 *tics.*

600 Falconbridge MS., Stamps RL., Badcock DR. 2006. A Simple Hebbian/Anti-Hebbian Network
601 Learns the Sparse, Independent Components of Natural Images. *Neural Computation* 18:415–
602 429. DOI: 10.1162/089976606775093891

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

Földiák P. 1990. Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics*.

Földiák P. 2013. Sparse and Explicit Neural Coding. In: *Principles of Neural Coding*. CRC Press, 379–390. DOI: 10.1201/b14756-22

Graham DJ., Field DJ. 2007. Sparse Coding in the Neocortex. In: Kaas JH ed. *Evolution of Nervous Systems*. Oxford: Academic Press, 181–187.

Grossberg S. 1980. How does a brain build a cognitive code? *Psychological Review* 87:1.

Hartmann K., Bruehl C., Golovko T., Draguhn A. 2008. Fast Homeostatic Plasticity of Inhibition via Activity-Dependent Vesicular Filling. *PLoS ONE* 3:e2979 EP –. DOI: 10.1371/journal.pone.0002979

Hasselmo ME. 1995. Neuromodulation and cortical function: modeling the physiological basis of behavior. *Behavioural Brain Research* 67:1–27. DOI: 10.1016/0166-4328(94)00113-T

Häusser M., Mel B. 2003. Dendrites: bug or feature? *Current opinion in neurobiology* 13:372–383. DOI: 10.1016/S0959-4388(03)00075-8

Hubel DH., Wiesel TN. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*. DOI: 10.1113/jphysiol.1962.sp006837

Intrator N. 1990. A neural network for feature extraction. In: *Advances in Neural Information Processing Systems*.

King PD., Zylberberg J., DeWeese MR. 2013. Inhibitory interneurons decorrelate excitatory cells to drive sparse code formation in a spiking model of V1. *The Journal of Neuroscience* 33:5475–5485. DOI: 10.1523/JNEUROSCI.4188-12.2013

Kohonen T. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43:59–69. DOI: 10.1007/BF00337288

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

626 Körding KP., König P. 2000. A learning rule for dynamic recruitment and decorrelation. *Neural*
627 *Networks* 13:1–9. DOI: 10.1016/S0893-6080(99)00088-X

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

630 Kubota Y., Karube F., Nomura M., Kawaguchi Y. 2016. The Diversity of Cortical Inhibitory
631 Synapses. *Frontiers in neural circuits* 10. DOI: 10.3389/fncir.2016.00027

632 LeCun Y., Bottou L., Bengio Y., Haffner P. 1998. Gradient-based learning applied to document
633 recognition. *Proceedings of the IEEE* 86:2278–2324. DOI: 10.1109/5.726791

634 LeCun Y., Cortes C. 1998. The MNIST database of handwritten digits.

635 Legenstein R., Maass W. 2011. Branch-Specific Plasticity Enables Self-Organization of
636 Nonlinear Computation in Single Neurons. *Journal of Neuroscience* 31:10787–10802.
637 DOI: 10.1523/JNEUROSCI.5684-10.2011

638 Lettvin JY., Maturana HR., McCulloch WS., Pitts WH. 1959. What the Frog's Eye Tells the Frog's
639 Brain. *Proceedings of the IRE* 47:1940–1951. DOI: 10.1109/JRPROC.1959.287207

640 Linsker R. 1986. From basic network principles to neural architecture: emergence of orientation
641 columns. *Proceedings of the national academy of sciences*.

642 Linsker R. 1988. Self-organization in a perceptual network. *Computer* 21:105–117.
643 DOI: 10.1109/2.36

644 Makhzani A., Frey B. 2013. k-Sparse Autoencoders. *eprint arXiv:1312.5663*:-.

645 Marshall JA. 1990. A self-organizing scale-sensitive neural network. In: *1990 IJCNN International*
646 *Joint Conference on Neural Networks*. IEEE, 649–654 vol.3. DOI: 10.1109/IJCNN.1990.137911

647 Milojkovic BA., Radojicic MS., Antic SD. 2005. A strict correlation between dendritic and somatic
648 plateau depolarizations in the rat prefrontal cortex pyramidal neurons. *The Journal of Neuro-*
649 *science* 25:3940–3951. DOI: 10.1523/JNEUROSCI.5314-04.2005

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

650 Mountcastle V. 1997. The columnar organization of the neocortex. *Brain* 120:701–722.
651 DOI: 10.1093/brain/120.4.701

652 Muir DR., Cook M. 2014. Anatomical Constraints on Lateral Competition in Columnar Cortical
653 Architectures. *Neural Computation* 26:1624–1666. DOI: 10.1152/jn.00747.2006

654 Olshausen BA., Field DJ. 1996. Natural image statistics and efficient coding. *Network: Computation*
655 *in Neural Systems* 7:333–339. DOI: 10.1088/0954-898X_7_2_014

656 Olshausen BA., Field DJ. 1997. Sparse coding with an overcomplete basis set: A strategy employed
657 by V1? *Vision Research* 37:3311–3325. DOI: 10.1016/S0042-6989(97)00169-7

658 Olshausen BA., Field DJ. 2004. Sparse coding of sensory inputs. *Current opinion in neurobiology.*
659 DOI: 10.1016/j.conb.2004.07.007

660 Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Pretten-
661 hofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot
662 M., Duchesnay Édouard. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine*
663 *Learning Research* 12:2825–2830.

664 Pharr M., Mark WR. 2012. ispc: A SPMD compiler for high-performance CPU programming. In:
665 *2012 Innovative Parallel Computing (InPar)*. IEEE, 1–13. DOI: 10.1109/InPar.2012.6339601

666 Poirazi P., Brannon T., Mel BW. 2003. Pyramidal Neuron as Two-Layer Neural Network. *Neuron*
667 37:989–999. DOI: 10.1016/S0896-6273(03)00149-1

668 Polsky A., Mel BW., Schiller J. 2004. Computational subunits in thin dendrites of pyramidal cells.
669 *Nature neuroscience* 7:621–627. DOI: 10.1038/nn1253

670 Redondo RL., Morris RGM. 2011. Making memories last: the synaptic tagging and capture hy-
671 pothesis. *Nature Reviews Neuroscience* 12:17–30. DOI: 10.1038/nrn2963

672 Rodriguez A., Whitson J., Granger R. 2004. Derivation and Analysis of Basic Computa-
673 tional Operations of Thalamocortical Circuits. *Journal of cognitive neuroscience* 16:856–877.
674 DOI: 10.1162/089892904970690

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

675 Rumelhart DE., Zipser D. 1985. Feature Discovery by Competitive Learning. *Cognitive Science*
676 9:75–112. DOI: 10.1207/s15516709cog0901_5

677 Savin C., Joshi P., Triesch J. 2010. Independent Component Analysis in Spiking Neurons. *PLoS*
678 *Computational Biology* 6:e1000757. DOI: 10.1371/journal.pcbi.1000757

679 Schiess M., Urbanczik R., Senn W. 2016. Somato-dendritic Synaptic Plasticity and
680 Error-backpropagation in Active Dendrites. *PLoS Computational Biology* 12:e1004638.
681 DOI: 10.1371/journal.pcbi.1004638.s003

682 Shouval HZ., Bear MF., Cooper LN. 2002. A unified model of NMDA receptor-dependent bidi-
683 rectional synaptic plasticity. *Proceedings of the national academy of sciences* 99:10831–10836.
684 DOI: 10.1073/pnas.152343099

685 Spratling MW. 2011. Unsupervised Learning of Generative and Discriminative Weights Encod-
686 ing Elementary Image Components in a Predictive Coding Model of Cortical Function. *Neural*
687 *Computation* 24:60–103. DOI: 10.1162/NECO_a_00222

688 Spruston N. 2008. Pyramidal neurons: dendritic structure and synaptic integration. *Nature Re-*
689 *views Neuroscience* 9:206–221. DOI: 10.1038/nrn2286

690 Stevens JLR., Law JS., Antolik J., Bednar JA. 2013. Mechanisms for Stable, Robust, and Adap-
691 tive Development of Orientation Maps in the Primary Visual Cortex. *Journal of Neuroscience*
692 33:15747–15766. DOI: 10.1523/JNEUROSCI.1037-13.2013

693 Talaván PM., Yáñez J. 2005. A continuous Hopfield network equilibrium points algorithm. *Com-*
694 *puters & Operations Research* 32:2179–2196. DOI: 10.1016/j.cor.2004.02.008

695 Triesch J. 2007. Synergies Between Intrinsic and Synaptic Plasticity Mechanisms. *Neural Compu-*
696 *tation* 19:885–909. DOI: 10.1162/neco.2007.19.4.885

697 Urbanczik R., Senn W. 2014. Learning by the Dendritic Prediction of Somatic Spiking. *Neuron*
698 81:521–528. DOI: 10.1016/j.neuron.2013.11.030

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ:

699 Van Hooser SD. 2005. Orientation Selectivity without Orientation Maps in Visual Cortex of
700 a Highly Visual Mammal. *Journal of Neuroscience* 25:19–28. DOI: 10.1523/JNEUROSCI.4042-
701 04.2005

702 Wang LP., Wan CR. 2008. Comments on "The Extreme Learning Machine. *Neural Networks, IEEE*
703 *Transactions on* 19:1494–1495. DOI: 10.1109/TNN.2008.2002273

704 Wilmes KA., Sprekeler H., Schreiber S. 2016. Inhibition as a Binary Switch for Excitatory Plas-
705 ticity in Pyramidal Neurons. *PLoS Computational Biology* 12:e1004768 EP –. DOI: 10.1371/jour-
706 nal.pcbi.1004768

707 Yger P., Stimberg M., Brette R. 2015. Fast Learning with Weak Synaptic Plasticity. *J. Neurosci.*
708 35:13351. DOI: 10.1523/JNEUROSCI.0607-15.2015

709 Zylberberg J., Murphy JT., DeWeese MR. 2011. A Sparse Coding Model with Synaptically Local
710 Plasticity and Spiking Neurons Can Account for the Diverse Shapes of V1 Simple Cell Receptive
711 Fields. *PLoS Computational Biology* 7:e1002250 EP –. DOI: 10.1371/journal.pcbi.1002250

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2595v1 | CC BY 4.0 Open Access | rec: 15 Nov 2016, publ: