Unifying sparse coding, predictive coding, and divisive normalization

Author Names and Affiliations: Yanbo Lian[1], Anthony N. Burkitt[1,2]

[1] Department of Biomedical Engineering, The University of Melbourne, Parkville, Victoria, 3010, Australia

[2] Greame Clark Institute of Biomedical Engineering, The University of Melbourne, Parkville, Victoria, 3010, Australia

Correspondence: yanbo.lian@unimelb.edu.au

Acknowledgements: This work received funding from the Australian Government, via grant AUS-MURIB000001 associated with ONR MURI grant N00014-19-1-2571.

Conflict of interest statement: The authors declare no competing financial interests.

Abstract

Sparse coding, predictive coding and divisive normalization have each been found to be principles that underlie the function of neural circuits in many parts of the brain, supported by substantial experimental evidence. However, the connections between these related principles are still poorly understood. In this paper, we show how sparse coding, predictive coding, and divisive normalization can be described within a unified framework, and illustrate this explicitly within the context of a two-layer neural model. This two-layer model is constructed in a way that implements sparse coding with a network structure that is constructed by implementing predictive coding. The results demonstrate how a homeostatic function that regulates neural responses in the model can shape the nonlinearity of neural responses in a way that replicates different forms of divisive normalization. The equivalence of the model to divisive normalization in a single-neuron case is shown analytically. Simulations show that the model can learn simple cells in the primary visual cortex with the property of contrast saturation, which has previously been explained by divisive normalization. In summary, the study demonstrates that the three principles of sparse coding, predictive coding, and divisive normalization can be unified to provide a learning framework based on biophysical properties, such as Hebbian learning and homeostasis, and this framework incorporates both learning and more diverse response nonlinearities observed experimentally. This framework has the potential to also be used to explain how the brain learns to integrate input from different sensory modalities.

1 Introduction

The structure of the brain's networks enables it to perform complex daily tasks. Numerous experimental studies of the visual system, as well as other areas of the brain, have been conducted to uncover the underlying network structures, such as those responsible for simple cells and complex cells in cat's primary visual cortex (V1) (Hubel and Wiesel, 1959, 1962, 1968). Different models have been proposed to account for various features of the experimental data, and this has resulted in a number of proposed neuronal models capable of describing some of the underlying mechanisms of the brain's networks. Three such important models of network function are: (i) divisive normalization (Heeger, 1992), (ii) sparse coding (Olshausen and Field, 1996, 1997), and (iii) predictive coding (Rao and Ballard, 1999), each of which has had a significant

impact due to their success in accounting for experimental data and providing specific mechanisms for the way in which the neural circuits function.

Divisive normalization, originally proposed by Heeger (1992), is a mathematical model that normalizes the response of individual model units (representing neurons in the real neural circuits) by the division of population response of the whole neural network; i.e., that the response of each neuron not only depends on its own activity but also on the responses of other neurons in the network. Divisive normalization was introduced to explain how responses of cells observed in cat V1, can display significant nonlinearities that linear models failed to account for. Heeger (1992) showed that a simple mathematical model of divisive normalization can explain many nonlinear response properties of V1 cells, such as contrast saturation and contrast invariance. Due to the wide diversity of nonlinear structure introduced through divisive normalization with different model parameters, divisive normalization has been used to explain a wide range of physiological data, from the visual system (Heeger, 1992) to the olfactory system (Olsen et al., 2010) and also multisensory integration (Ohshiro et al., 2011, 2017). For a review of divisive normalization see Carandini and Heeger (2012).

Sparse coding, originally proposed by Olshausen and Field (1996, 1997), finds an efficient representation of the sensory input using a linear combination of a set of basis vectors. Sparse coding is based upon two important assumptions, namely that the model generates a linear reconstruction of the input and this linear reconstruction is efficient. The notion of efficiency adopted in sparse coding is based on the principle of redundancy reduction proposed by Barlow (1961, 1989). Olshausen and Field (1996, 1997) applied sparse coding to natural images and showed that this model can learn Gabor-like features after learning, which resemble the observed receptive fields (RFs) of simple cells in V1. An appealing feature of sparse coding is its learning ability, namely that the model describes how the connection weights can be learnt based on the statistics of the input visual data, in a way that is similar to synaptic plasticity. Apart from its success in learning RFs of cells in the visual cortex, sparse coding has also been successfully applied to explain experimental results in other brain areas, such as auditory cortex (Smith and Lewicki, 2006) and hippocampal formation (Lian and Burkitt, 2021, 2022; Lian et al., 2023). For a review see Beyeler et al. (2019).

Predictive coding, originally proposed by Rao and Ballard (1999), provides a hierarchical predictive framework that is inspired by the observation of the extensive feedback in the cortex. Predictive coding assumes a hierarchical network structure, where units in the lower layers of the hierarchy project to the successive higher layers via feedforward connections, while units in the higher layers project back to the lower layers via feedback connections between the adjacent layers. The residual error, defined as the difference between the feedforward input to a layer and the feedback from its adjacent higher layer, is computed and generates the response of the units in the lower layer that project to the upper layer. The feedback from higher layers of the hierarchy represents the prediction at that layer of the input, and thus the residual error is often referred to as *prediction error* in predictive coding. Note that this framework is predictive in a hierarchical sense rather than a temporal sense. The essence of predictive coding is that the higher layers in the network hierarchy constantly make prediction of the current input, compute the mismatch between input and prediction, and only transmit that mismatch (prediction error) to the higher layer. The notion of only transmitting mismatch to the higher layer is similar to the principle of efficiency in sparse coding. Rao and Ballard (1999) showed that predictive coding, when used to model the visual pathways of the brain, can learn simple-cell-like RFs and display some non-classical RF properties such as end-stopping. A Bayesian framework was later constructed using predictive coding, which provides an appealing model for many different brain areas (Knill and Pouget, 2004; Friston, 2010; Bastos et al., 2012).

Built upon these three significant neural mechanisms - sparse coding, predictive coding and divisive normalization - many computational and experimental studies have successfully been able to expand our understanding of particular brain functions. However, a unified understanding or framework for these three models is still lacking. Though the direct connection between these three models may not be immediately apparent, we propose that these three models are closely related through the underlying network structures by which they are implemented. In this paper, we provide a framework that unifies sparse coding, predictive coding and divisive normalization from first principles.

2 Methods

2.1 Sparse coding

Sparse coding finds an efficient representation of the sensory input using a linear combination of a set of basis vectors (Olshausen and Field, 1996, 1997). Denote

$$\begin{aligned} \mathbf{x} &= [x_1, x_2, \cdots, x_L]^T \\ \mathbf{s} &= [s_1, s_2, \cdots, s_M]^T \\ \mathbf{A} &= [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_M], \end{aligned} \tag{1}$$

where $\mathbf{x}$ is a $L \times 1$ vector that represents the sensory input such as visual or auditory stimuli that has been pre-processed upstream, $\mathbf{s}$ is a $M \times 1$ vector that represents the response of M model units, and $\mathbf{A}$ is a $L \times M$ matrix that contains M basis vectors $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_M$. Each basis vector, $\mathbf{a}_i$ ($i = 1, 2, \cdots, M$) is a $L \times 1$ vector that represents one feature used to reconstruct the input such that the input, $\mathbf{x}$, can be represented by a linear combination of basis vectors weighted by model responses:

$$\mathbf{x} \approx \mathbf{A}\mathbf{s} = \sum_{i=1}^{M} s\, \mathbf{a}_i. \tag{2}$$

Sparse coding finds the linear representation (Eq. 2) using sparse responses, namely that only part of the model responses, s_i ($i = 1, 2, \cdots, M$), will be active. This can be formulated as an optimization problem of minimizing the following cost function:

$$C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s}) = \frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{s}\|^2 + \lambda Q(\mathbf{s}), \tag{3}$$

where $Q(\mathbf{s}) = \sum_{i=1}^{M} Q(s_i)$, $Q(\cdot)$ is a function that penalizes large responses, and λ is a constant parameter that scales the penalty function. In other words, the first term on the right-hand-side of Eq. 3 represents the reconstruction error and the second term determines the sparsity of model units. Therefore, by minimizing $C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s})$, one finds the linear reconstruction of the input using sparse responses. However, there is always a trade-off between the reconstruction fidelity and the response sparsity. Note that Eq. 3 is the same as Eq. 2

in Olshausen and Field (1996) and Eq. 14 in Olshausen and Field (1997).

Assuming that $Q(\cdot)$ is differentiable and given $\mathbf{A}$, the dynamics of $\mathbf{s}$ can be derived from taking the partial derivative of $C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s})$:

$$\dot{\mathbf{s}} \propto -\frac{\partial C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s})}{\partial \mathbf{s}} = \mathbf{A}^T(\mathbf{x} - \mathbf{A}\mathbf{s}) - \lambda Q'(\mathbf{s}), \tag{4}$$

where $Q'(\cdot)$ represents the derivative of $Q(\cdot)$ and $Q'(\mathbf{s}) = [Q'(s_1), Q'(s_2), \cdots, Q'(s_M)]^T$. Eq. 4 describes how, given input $\mathbf{x}$, an iterative procedure enables us to calculate the response of model units. The response $\mathbf{s}$ may or may not reach a stable equilibrium, depending on the properties of this dynamic system.

The learning rule for the connection weights $\mathbf{A}$ can also be derived by taking the partial derivative of $C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s})$:

$$\dot{\mathbf{A}} \propto -\frac{\partial C^{\mathrm{SC}}(\mathbf{A}, \mathbf{s})}{\partial \mathbf{A}} = (\mathbf{x} - \mathbf{A}\mathbf{s})\mathbf{s}^T \tag{5}$$

L2 norm of $\mathbf{A}$ is then adapted.

Generally, when simulating sparse coding models, the learning rule in Eq. 5 is applied once only after some iterations of computing the model response according to the model dynamics described by Eq. 4, with the assumptions that (i) the time scale of learning/plasticity is longer than that of model responses, and (ii) the model response iterates to some stable value. The learning rule in Eq. 5 may cause basis vectors in $\mathbf{A}$ to grow without bound, so L2 norm normalization of each vector, $\mathbf{a}_i$, is used in the original work of sparse coding (Olshausen and Field, 1997) and some subsequent studies (Zhu and Rozell, 2013; Lian et al., 2019).

2.2 Predictive coding

Predictive coding, originally proposed by Rao and Ballard (1999), is a hierarchical model in which prediction errors are propagated to the higher levels of the network hierarchy. Predictive coding can have multiple layers of the network, but in this study we only look at a two-layer predictive coding network model because it is the fundamental building block of multi-layer predictive coding and it can be explicitly shown how it is related to sparse coding and divisive normalization. Combining Eqs. 4 and 5 in Rao and Ballard (1999), the cost function of predictive coding can be written as

$$C^{\mathrm{PC}}(\mathbf{A}, \mathbf{s}) = \frac{1}{2}\|\mathbf{x} - f(\mathbf{A}\mathbf{s})\|^2 + \lambda Q(\mathbf{s}) + \beta W(\mathbf{A}), \tag{6}$$

where $f(\cdot)$ can be interpreted as the neuronal activation function, $f(\mathbf{As})$ is the prediction of the input, $Q(\mathbf{s})$ is the prior constraint on the model responses, and $W(\mathbf{A})$ puts some regularization on the basis vectors that are the columns of $\mathbf{A}$.

Similar to the sparse coding case, the model dynamics and learning rule of predictive coding can be derived by taking the corresponding partial derivatives of $C^{\mathrm{PC}}(\mathbf{A}, \mathbf{s})$:

$$\dot{\mathbf{s}} \propto -\frac{\partial C^{\mathrm{PC}}(\mathbf{A}, \mathbf{s})}{\partial \mathbf{s}} = \mathbf{A}^T \frac{\partial f^T(\mathbf{As})}{\partial \mathbf{As}} (\mathbf{x} - f(\mathbf{As})) - \lambda Q'(\mathbf{s}) \tag{7}$$

and

$$\dot{\mathbf{A}} \propto -\frac{\partial C^{\mathrm{PC}}(\mathbf{A}, \mathbf{s})}{\partial \mathbf{A}} = \frac{\partial f^T(\mathbf{As})}{\partial \mathbf{As}} (\mathbf{x} - f(\mathbf{As}))\mathbf{s}^T - \beta W'(\mathbf{A}). \tag{8}$$

The learning rule of predictive coding is applied after some iterations of computing the model responses.

2.3 Divisive normalization

Divisive normalization was first proposed by Heeger (1992) to account for the nonlinear physiological responses of V1 cells. As reviewed by Carandini and Heeger (2012), one commonly used variant of divisive normalization is given by

$$\begin{aligned} s_i^{\mathrm{DN}} &= \gamma \frac{(s_i^{\mathrm{L}})^n}{\rho^n + \sum_k (s_k^{\mathrm{L}})^n} \\ s_i^{\mathrm{L}} &= \mathbf{a}_i^T \mathbf{x}, \end{aligned} \tag{9}$$

where $\mathbf{x}$ is the sensory input, $\mathbf{a}_i$ represents the synaptic weights connecting input with neuron i and is normally hand-wired, s_i^{L} is the linear response, s_i^{DN} is the response of neuron i after divisive normalization, and γ, ρ and n are the parameters that determine the shape of the response curve of divisive normalization. Note that s_i^{L} can also incorporate a rectifying function, but we use a linear function for the purpose of analytical analysis below. The model structure of divisive normalization is shown in Fig. 1.

Divisive normalization introduces nonlinearity into the model through the calculation of the neural responses in Eq. 9, which also introduces competition between model units through the scaling of the response of each unit by the response of the whole population. Note that divisive normalization, similar to sparse coding and predictive coding, involves the definition of connection weights, but unlike sparse coding and predictive

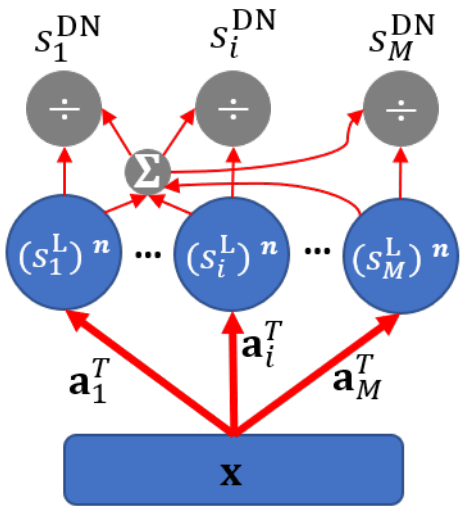

Figure 1: **Model structure of divisive normalization (DN).** The linear response is first generated; some form of nonlinearity (exponential function with exponent n) is then applied; next, the response of each unit is divided by the sum of responses of all the units in the network.

coding, the synaptic weights are not learnt through a learning process.

2.4 Hebbian learning

Hebbian learning is the principle that the change of synaptic weight connecting two neurons depends on the responses of pre- and post-synaptic neurons. The learning rules of both sparse coding (Eq. 5) and predictive coding (Eq. 8) have a Hebbian component, namely $(\mathbf{x} - \mathbf{A}\mathbf{s})\mathbf{s}^T$ for sparse coding and $(\mathbf{x} - f(\mathbf{A}\mathbf{s}))\mathbf{s}^T$ for predictive coding. By interpreting $(\mathbf{x} - \mathbf{A}\mathbf{s})$ or $(\mathbf{x} - f(\mathbf{A}\mathbf{s}))$ as the response of error neurons that compute the difference between input and model reconstruction, these learning rules can be seen to obey the Hebbian learning principle.

2.5 Homeostasis

The brain has self-regulating mechanisms called homeostatic plasticity that stabilizes the neural system via synaptic scaling and response adaptation (Turrigiano, 2011, 2017). Sparse coding and predictive coding have components that are similar to homeostasis. The weight normalization in sparse coding (Eq. 5) and weight regularization in predictive coding (Eq. 8) scale weights so that the learning process is stable, playing a role similar to homeostatic synaptic scaling. In addition, the derivative of $Q(s)$ in the model dynamics of

both sparse coding (Eq. 4) and predictive coding (Eq. 7) regulates the model response, which is similar to homeostatic firing regulation.

3 Results

3.1 Relationship between sparse coding and predictive coding

Predictive coding is well known to be closely related to sparse coding, or efficient coding more broadly speaking, because only the prediction error is fed to higher levels of the network hierarchy. Furthermore, it can be shown from first principles that sparse coding and predictive coding are identical in a two-layer neural network under some parameter choices.

In the original cost function of predictive coding (Eq. 6), the term $\beta W(\mathbf{A})$ applies some penalties on the connection weights $\mathbf{A}$ during the learning process, since its derivative appears in the learning rule described by Eq. 8. Rao and Ballard (1999) used the sum of squared response in the cost function that regulates the L2 norm of learned basis vectors ($\mathbf{A}$), which is essentially similar to L2 normalization used in sparse coding models during the learning process. Therefore, $\beta W(\mathbf{A})$ in Eq. 6 can be removed and $-\beta W'(\mathbf{A})$ in Eq. 8 can be approximated by a L2 normalization. Additionally, in the case where $f(\cdot)$ is the linear function, $f(y) = y$, then the cost function (Eq. 6), model dynamics (Eq. 7) and learning rule (Eq. 8) of predictive coding become identical to those of sparse coding (Eqs. 3, 4 and 5). Note that $f(y) = y$ was used in end-stopping simulations to illustrate predictive coding in the original study by Rao and Ballard (1999).

In summary, when the weight regularization term is replaced by L2 normalization and the feedback activation function is linear, sparse coding is identical to predictive coding. A unified framework with model diagram and equations is given below.

The exposition above shows how sparse coding and predictive coding can be described in the same framework, although their original formulation had different goals, namely sparse coding is about the sparsity in the network model while predictive coding is concerned with the goal of the resulting neural processing. Here a unified framework for sparse coding and predictive coding is illustrated by implementing the principle of sparse coding in a two-layer model with the structure of predictive coding. We refer to this model

as sparse/predictive coding (SPC) model, similar to Boutin et al. (2020). The structure of the SPC model is shown in Fig. 2.

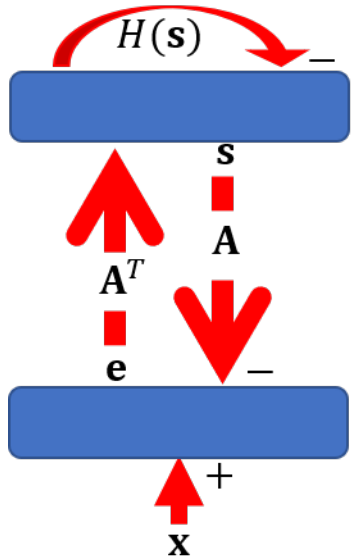

Figure 2: **Model structure of sparse/predictive coding (SPC).** The first layer computes the prediction error between the input and reconstruction of the model, and then sends error signals to the second layer. The second layer takes input from the first layer and incorporates the response-regulating mechanism, $H(\cdot)$. This two-layer network implements the dynamics described in Eq. 10.

The similarity between sparse coding (Eq. 4) and predictive coding (Eq. 7) enables the dynamics and learning rule of the SPC model to be described as a stage of *response iteration*:

$$\begin{aligned} \dot{\mathbf{s}} &= \eta_s(\mathbf{A}^T\mathbf{e} - H(\mathbf{s})) \ , \\ \mathbf{e} &= \mathbf{x} - \mathbf{A}\mathbf{s}, \end{aligned} \tag{10}$$

followed by the *learning step*:

$$\dot{\mathbf{A}} = \eta_a \mathbf{e}\mathbf{s}^T \ \text{ with } \ \|\mathbf{A}\|_2 = 1 \ , \tag{11}$$

where $\mathbf{e}$ is the prediction error, $H(\mathbf{s}) = \lambda Q'(\mathbf{s}) = [H(s_1), H(s_1), \cdots, H(s_M)]^T$ is the mechanism that regulates the responses of model units, η_s is the update rate when computing the responses, η_a is the learning rate of updating connection weights, and the magnitude of $\mathbf{A}$ is regulated by weight normalization ($\|\mathbf{A}\|_2 = 1$) at the end of each learning step, similar to previous studies (Olshausen and Field, 1997; Rozell et al., 2008; Zhu and Rozell, 2013; Lian et al., 2019).

The model responses are computed in response iteration using Eq. 10 and then the learning rule of SPC is applied in the learning step using Eq. 11. Note that Eq. 11 is a local Hebbian learning rule in this model structure. η_s needs to be chosen carefully to ensure reasonable convergence of the model responses (see Fig. 5B for an example). η_a also needs to be much larger than η_s (so that the weight dynamics is on a much slower timescale than the response dynamics) and to be chosen such that $\mathbf{A}$ can learn meaningful structure at a reasonable speed.

When this SPC model is related to the neural circuits of the brain, neurons in the first layer compute the prediction error and feed the error signals to the second layer, where the responses are computed according to the dynamics in Eq. 10. $\mathbf{A}^T$ and $\mathbf{A}$ can be interpreted as the feedforward and feedback synaptic connections between two layers. This structure was used by Lian et al. (2019) to implement sparse coding of V1 simple cells with separate ON and OFF LGN channels and the learnt receptive fields, given by $\mathbf{A}^T$, agree well with experimental data and can account for many experimental phenomena observed in the LGN-V1 pathway.

An alternative way of implementing sparse coding in a feedforward network is through inhibitory lateral connections, which promote competition that eventually leads to sparse responses (Rozell et al., 2008; Zylberberg et al., 2011; Brito and Gerstner, 2016), and the role of the lateral connections is to decorrelate responses across the network, so that the model can learn different features. However, the mathematical equations underlying different implementations of sparse coding are essentially the same. Therefore, the framework of SPC proposed here can unify a range of different models related with sparse coding and predictive coding.

In the cost functions of sparse coding (Eq. 3) and predictive coding (Eq. 6), $Q(\cdot)$ is the sparsity term that promotes sparse responses of the model units and its derivative is denoted as $H(\cdot)$, as appeared in the SPC model (Eq. 10). Consequently, $H(\cdot)$ can be understood as a homeostasis function that self-regulates individual neuronal responses and should only depend on the homeostatic mechanism of individual neurons.

3.2 The homeostasis function in sparse/predictive coding determines response nonlinearity

The dynamics of SPC model described in Eq. 10 can be reformulated into one equation:

$$\dot{\mathbf{s}} = \eta_s \left[\mathbf{A}^T (\mathbf{x} - \mathbf{A}\mathbf{s}) - H(\mathbf{s}) \right]. \tag{12}$$

The equilibrium, $\mathbf{s}^{\mathrm{EQ}}$, of this differential equation is reached when $\dot{\mathbf{s}} = \mathbb{0}$, namely when

$$-\mathbf{A}^T\mathbf{A}\mathbf{s}^{\mathrm{EQ}} + \mathbf{A}^T\mathbf{x} = H(\mathbf{s}^{\mathrm{EQ}}). \tag{13}$$

Solutions to Eq. 13 are the steady state response when the iterative process of solving Eq. 12 converges. If we define $Y(\mathbf{s})$ as a linear function of $\mathbf{s}$ with form $Y(\mathbf{s}) := -\mathbf{A}^T\mathbf{A}\mathbf{s} + \mathbf{A}^T\mathbf{x}$, where $-\mathbf{A}\mathbf{A}^T$ is the slope and $\mathbf{A}^T\mathbf{x}$ is the offset of $Y(\mathbf{s})$, then the solutions $\mathbf{s}^{\mathrm{EQ}}$ to Eq. 13 are just the intersects between $Y(\mathbf{s})$ and $H(\mathbf{s})$. Function $H(\cdot)$ is fixed for any given neuron because it depends on the self-regulating homeostatic mechanisms of that particular neuron. As $Y(\mathbf{s})$ shifts, corresponding to its offset $\mathbf{A}^T\mathbf{x}$, the intersect between these two functions determines the analytical equilibrium of the activity, $\mathbf{s}^{\mathrm{EQ}}$, given the current input $\mathbf{x}$. A simplified one-dimensional illustration is shown in Fig. 3.

Consider now the case of the spontaneous activity of the neuron, namely the response when the input $\mathbf{x}$ is zero. When there is no external input to the network (i.e., $\mathbf{x} = \mathbb{0}$), the spontaneous response of the SPC network is determined by the intersect between the curves $Y(\mathbf{s}) = -\mathbf{A}^T\mathbf{A}\mathbf{s}$ and $H(\mathbf{s})$. Therefore, given $\mathbf{A}$, the choice of $H(\cdot)$ determines the spontaneous response of the model. When input $\mathbf{x}$ changes linearly, $Y(\mathbf{s})$ also changes linearly, so how the intersect $\mathbf{s}^{\mathrm{EQ}}$ changes solely depends on the shape of $H(\mathbf{s})$. In other words, $H(\cdot)$ determines the response nonlinearity of the model. For example, if $H(\mathbf{s})$ is a linear function of $\mathbf{s}$, model equilibrium is linearly dependant on the input $\mathbf{x}$; if $H(\mathbf{s})$ is highly nonlinear, there will be a great nonlinearity between model equilibrium and input; if $H(\mathbf{s})$ approaches infinity with a vertical asymptote $\mathbf{s} = \mathbf{s}^{\mathrm{M}}$, then the model equilibrium will saturate at $\mathbf{s}^{\mathrm{M}}$ although the input $\mathbf{x}$ keeps increasing.

When there is no homeostasis namely that $H(s) = 0$, the x-axis intersect of function $Y(\mathbf{s})$ will be the equilibrium of the model (see detailed analysis in Appendix A.3).

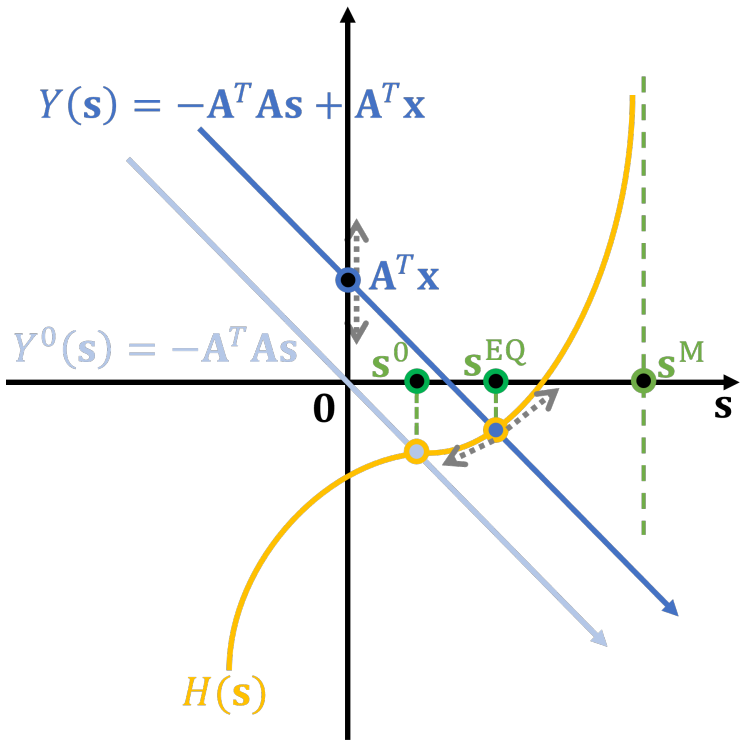

Figure 3: **Equilibrium of sparse/predictive coding model explained in one-dimensional space.** The intersect between $Y(\mathbf{s}) = -\mathbf{A}^T\mathbf{A}\mathbf{s} + \mathbf{A}^T\mathbf{x}$ and $H(\mathbf{s})$ determines the model's analytical equilibrium, $\mathbf{s}^{\mathrm{EQ}}$. As input $\mathbf{x}$ changes, $Y(\mathbf{s})$ shifts with the corresponding offset $\mathbf{A}^T\mathbf{x}$ and the intersect moves along the curve of function $H(\mathbf{s})$. When $\mathbf{x} = \mathbb{0}$, the intersect is the model's spontaneous response, $\mathbf{s}^0$. If $H(\mathbf{s})$ has a vertical asymptote at $\mathbf{s} = \mathbf{s}^{\mathrm{M}}$, then $\mathbf{s}^{\mathrm{M}}$ becomes the analytical upper bound upon response of the model.

3.3 Relationship between sparse/predictive coding and divisive normalization

Divisive normalization (DN) is fundamentally different from sparse/predictive Coding (SPC). Unlike SPC, DN is based upon a physiological model that was introduced to explain the nonlinear neural response properties in the brain, rather than a learning model with neural dynamics and synaptic plasticity. Specifically, for the model responses, SPC applies an iterative procedure described by Eq. 10, while the response of DN can be directly calculated by Eq. 9; for the learning rule, SPC incorporates learning and can describe how some brain properties can emerge via learning (Olshausen and Field, 1996, 1997; Zhu and Rozell, 2013; Lian et al., 2019), while DN has no learning rule and can only be used for explaining physiological neural response data. DN has a wide diversity of response nonlinearity with different parameters of the model, so it has been successful to account for the neural response nonlinearities found in experimental studies (Carandini and Heeger, 2012).

As demonstrated in the previous section (Section 3.2), the shape of the homeostasis function, $H(\cdot)$, determines the response nonlinearity of the SPC model, so SPC may potentially display the response nonlinearity of DN models, given a suitable choice of $H(\cdot)$.

3.3.1 SPC and DN can be identical for a single-neuron model

Consider the case where there is only one neuron in the first layer and one neuron in the second layer for both the SPC and DN models, that the input is nonnegative, and the connection weight is 1. The DN model computes the response as

$$s^{\mathrm{DN}} = \gamma \frac{x^n}{\rho^n + x^n} \ , \tag{14}$$

where γ determines the saturation value for the response, ρ determines the point where the response is $\gamma/2$, and n determines the curvature of the normalization curve. Some examples of DN described by Eq. 14 are shown in Fig. 4A.

For the SPC model, the equation that solves the analytical equilibrium of the output neuron (Eq. 13) becomes

$$x - s^{\mathrm{EQ}} = H(s^{\mathrm{EQ}}). \tag{15}$$

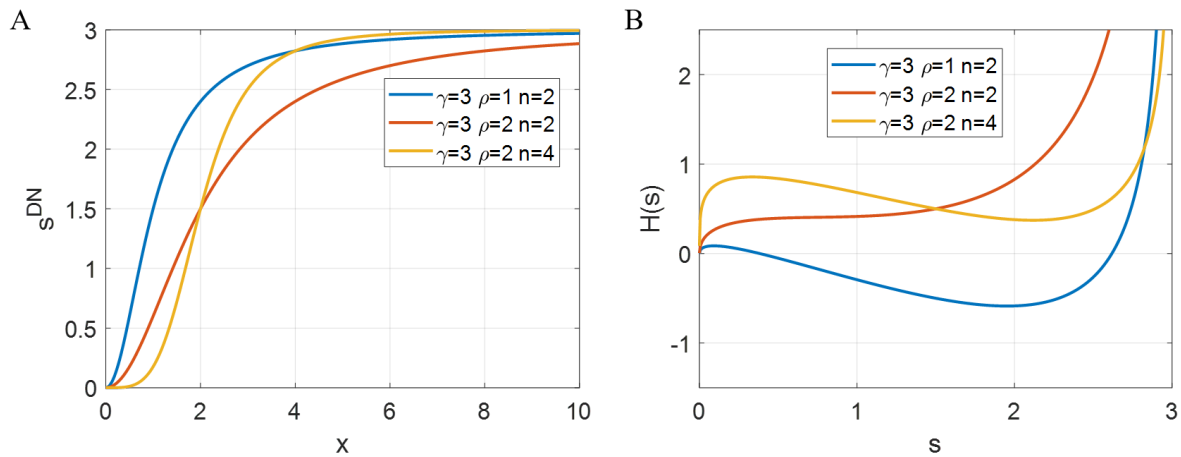

Figure 4: **Divisive normalization (DN) can be identical to sparse/predictive coding (SPC) for single-neuron scenario.** (A) DN described by Eq. 14. (B) Homeostasis function, $H(s)$, of the equivalent SPC model, described by Eq.17.

From Eq. 14 of DN, the solution for x is given by

$$x = \rho \left(\frac{\gamma}{s^{\mathrm{DN}}} - 1 \right)^{-\frac{1}{n}}. \tag{16}$$

If DN and SPC are equivalent for this scenario (i.e., $s^{\mathrm{DN}} = s^{\mathrm{EQ}}$), the homeostasis function in SPC can be derived from Eqs. 15 & 16:

$$H(s) = x - s = \rho \left(\frac{\gamma}{s} - 1 \right)^{-\frac{1}{n}} - s \quad \text{for} \;\; s < \gamma. \tag{17}$$

Some homeostasis functions of SPC described by Eq. 17 are shown in Fig. 4B. When s is small, $H(s)$ has a small cost on responses, though it displays various shapes for different sets of parameters. As s approaches 3 (the value of γ), $H(s)$ has a very high cost that prevents the response from exceeding the saturation. Therefore, for a single neuron scenario, SPC with a suitable choice of the homeostasis function can be equivalent to DN model that displays response nonlinearity and response saturation. Note that the homeostasis function (Eq. 17) is derived from Eq. 14 where s^{DN} is always smaller than the saturation value γ. Consequently, $H(s)$ in Eq. 17 only applies for values $s < \gamma$.

The simplest scenario described above is implemented with $\gamma = 3$, $\rho = 2$ and $n = 2$. DN is implemented

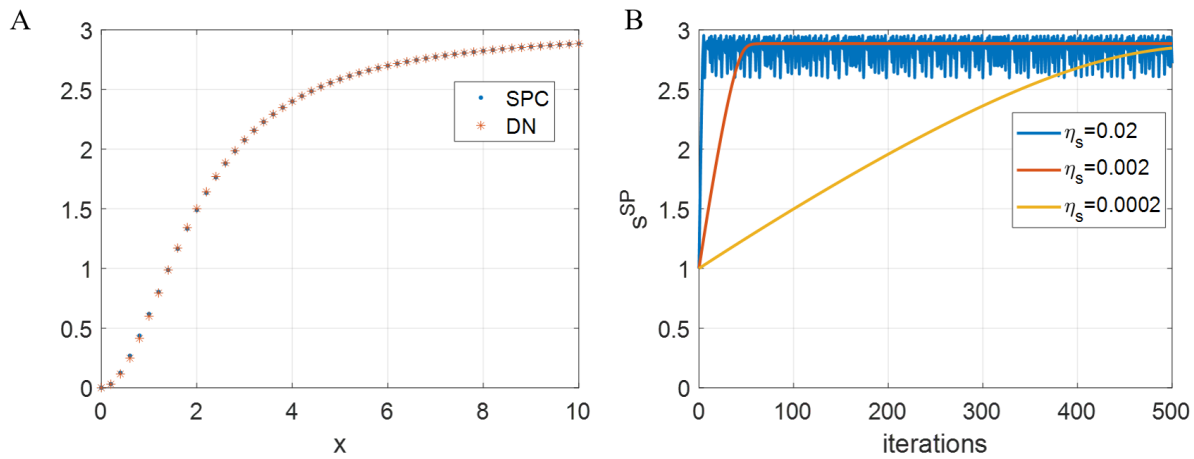

Figure 5: **Divisive normalization (DN) and sparse/predictive coding (SPC) in the single-neuron scenario.** (A) Responses of DN and SPC with $\eta_s = 0.002$ vs. different inputs x. (B) Response of SPC vs. number of iterations for input, $x = 10$, with different values of η_s (inserted panel).

using Eq. 14. SPC is implemented by an iterative procedure described by Eq. 10 with $H(s)$ given by Eq. 17 and 500 iterations of computing the response to any input.

Simulation results are shown in Fig. 5. Fig. 5A shows that SPC and DN can generate equivalent results where the model response of SPC after convergence is identical to the model response of DN for all inputs $x \geq 0$. When the input is $x = 10$, the response of SPC model vs. number of iterations is plotted in Fig. 5B, which shows that the response of SPC gradually evolves to the equilibrium and its dynamics depends upon the value of η_s. s^{SP} starts with a value of 1 because it is initialized to 1 prior to the iterative process. When computing the response of the SPC model, the value of η_s requires careful consideration. If η_s is too small, it takes more iterations to reach the equilibrium (e.g., yellow line in Fig. 5B). However, if η_s is too large, s^{SP} may become oscillatory (e.g., blue line in Fig. 5B). Additionally, if s^{SP} overshoots the saturation value, γ, during simulation of the response iteration (Eq. 10), this will cause the SPC model to break down because $H(s)$, described by Eq. 17, is well-defined only for $s < \gamma$.

The analysis of this simple scenario illustrates that although SPC and DN are different in terms of their model dynamics and model structure, they can nevertheless generate equivalent network activity given a suitable choice of the homeostasis function in the SPC model. However, the homeostasis function in SPC enforces a rigid constraint upon the response, namely that the response has to be always smaller than γ

during the iterative procedure of computing responses. When the network has more complexity, namely as the number of neurons increases, this assumption can be violated.

3.3.2 Relationship between SPC and DN in a multi-neuron network

If there are multiple neurons in the network, it is no longer straightforward to obtain a mathematical formula that provides the equivalent SPC model for any given DN model. However, we can still estimate how SPC is related to the response nonlinearity of DN through the homeostasis function $H(\cdot)$ of the SPC model.

Denote $\mathbf{x} = \alpha \mathbf{x}_0$ where $\mathbf{x}_0$ is a normalized stimulus and α is a scalar that determines the strength of the input $\mathbf{x}$. The divisive normalization definition, Eq. 9, then becomes

$$\begin{aligned} s_i^{\mathrm{DN}} &= \gamma \frac{\alpha^n (s_i^{\mathrm{L},0})^n}{\rho^n + \alpha^n \sum_k (s_k^{\mathrm{L},0})^n} \ , \\ s_i^{\mathrm{L},0} &= \mathbf{a}_i^T \mathbf{x}_0 \end{aligned} \tag{18}$$

where $\sum_k (s_k^{\mathrm{L},0})^n$ represents the population response to a normalized stimulus $\mathbf{x}_0$ with some nonlinearities determined by the exponent n. Denoting $z := \sqrt[n]{\sum_k (s_k^{\mathrm{L},0})^n}$, then Eq. 18 becomes

$$s_i^{\mathrm{DN}} = \gamma \frac{\alpha^n (\mathbf{a}_i^T \mathbf{x}_0)^n}{\rho^n + \alpha^n z^n} = \frac{\gamma (\mathbf{a}_i^T \mathbf{x}_0)^n}{z^n} \frac{\alpha^n}{(\rho/z)^n + \alpha^n} = \gamma_0 \frac{\alpha^n}{\rho_0^n + \alpha^n} \ , \tag{19}$$

where $\gamma_0 = \frac{\gamma (\mathbf{a}_i^T \mathbf{x}_0)^n}{z^n}$ and $\rho_0 = \rho / z$. Comparison of Eq. 14 and Eq. 19 shows that the relationship between the model response of DN and the input magnitude α can be reduced to expression for the single-neuron scenario analyzed in previous section. Consequently, $H(\cdot)$ of SPC can be chosen to account for response nonlinearities described by DN.

3.4 Sparse/Predictive coding can learn Gabor-like simple cells with contrast saturation previously explained by divisive normalization

One notable difference between DN and SPC is the learning ability: DN is a model of only the physiological responses, and it can account for many nonlinear response properties, whereas the SPC model has a learning process; i.e., the SPC model, unlike the DN model, can learn important features of the input, given a suitable

choice of the homeostasis function (Olshausen and Field, 1996, 1997; Rao and Ballard, 1999). As demonstrated by Zhu and Rozell (2013), sparse coding can be used to explain a number of nonlinear receptive field properties of V1 cells such as contrast invariance and end-stopping. However, contrast saturation of V1 cells, which is a classical nonlinear property(Ohzawa et al., 1982, 1985; Freeman et al., 2002), can not be captured in the sparse coding model of Zhu and Rozell (2013), whereas it can easily be captured by DN (Heeger, 1992). In a simple scenario in the section above, DN can be equivalent to SPC with suitable choice of the homeostasis function, $H(s)$ (Eq. 17 and Fig. 4B), which suggests that SPC can account for saturation at least for a single-neuron scenario (Fig. 5A). Saturation in the SPC model is built into the homeostasis function, $H(s)$ (Eq. 17), that has a very large penalty (infinity in this case) at the saturation point.

In this section, we show that SPC with a novel homeostasis function can learn Gabor-like simple cells as well as displaying the property of contrast saturation that is previously explained by DN. When implementing SPC, we incorporate the nonnegativity constraints for the response (i.e., $\mathbf{s} \geq 0$) so that the model response represents the firing rate, given that nonnegative sparse coding keeps the same expressiveness as standard sparse coding (Papyan et al., 2017).

3.4.1 A homeostasis function for SPC

When using sparse coding to learn Gabor-like features that are similar to simple cells in V1 (Olshausen and Field, 1996, 1997), homeostasis functions were chosen to be of the particular forms given by

$$H(s) = \lambda \frac{s}{1 + (s/\sigma)^2}\ ,\ \ \lambda s e^{-(\frac{s}{\sigma})^2} \ \text{ or } \ \lambda s^{\alpha - 1} \quad (\text{with } s \geq 0)$$

where $\alpha = 1$ (Olshausen and Field, 1996, 1997; Zhu and Rozell, 2013; Lian et al., 2019). $H(s) = \lambda s^{\alpha-1}$ with $\alpha = 2$ is used by Rao and Ballard (1999), but the model does not generate Gabor-like receptive fields. When $H(s)$ takes the form of $\lambda s^{\alpha-1}$ (with $s \geq 0$), from our simulation, we find that it is necessary to choose α in the range $1 \leq \alpha < 2$ to enable the model to learn Gabor-like receptive fields, as will be shown in Section 3.4.2 and illustrated in Fig. 7.

After examining the homeostasis functions that can learn Gabor-like receptive fields for SPC models, one thing in common is that the homeostasis function is concave starting from 0. In order to introduce response

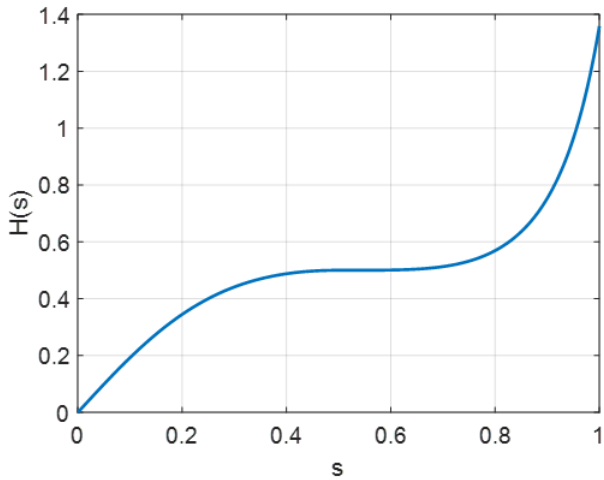

Figure 6: **The proposed homeostasis function described by Eq. 20.** In this example, parameters are chosen as follow: $\lambda = 0.5$, $x_0 = 0.5$, $\sigma = 0.5$ and $n = 4$.

saturation into the homeostasis function, the homeostasis function should approach infinity as the input increases, and a convex curvature of the function is necessary. Therefore, we propose a homeostasis function for the SPC model described as follows:

$$H(s) = \begin{cases} \lambda \frac{2s/x_0}{1+(s/x_0)^2}, & s \leq x_0 \\ \lambda e^{(\frac{s-x_0}{\sigma})^n}, & s > x_0 \end{cases}, \tag{20}$$

where $x_0 > 0$ is the turning point at which the homeostatic function, $H(s)$, becomes convex instead of concave, σ determines the steepness of $H(s)$ when s is larger than x_0, and n is the index that determines the steepness of the curve when s is large. An example of $H(s)$ given by Eq. 20 is shown in Fig. 6. Note that this homeostasis function has a similar shape to the corresponding homeostasis function of DN in the single-neuron scenario (Eq. 17 and Fig. 4B).

The proposed homeostasis function, $H(s)$, in Eq. 20 has the following properties: (1) it is concave when $s < x_0$, which facilitates the emergence of features after learning; (2) it is convex when $s > x_0$; (3) the derivative at $s = x_0$ is 0 such that the transition between concave and convex is smooth and continuous; (4) it approaches large values fairly quickly when s is large, which leads to response saturation.

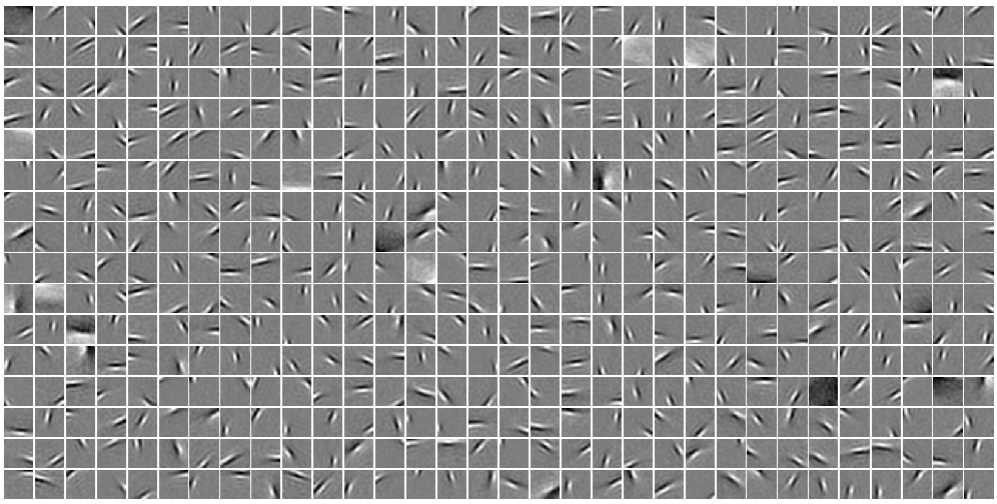

Figure 7: **Sparse/Predictive coding (SPC) model can learn Gabor-like features.** Receptive fields of 512 model units. Each square is a 16×16 image that shows the corresponding learnt feature. The homeostasis function used here is defined in Eq. 20.

3.4.2 The model can learn Gabor-like simple cells

The SPC model can learn Gabor-like simple cells when a homeostasis function, $H(s)$, as described in Fig. 6 is used in the model dynamics (Eq. 10) and learning rule (Eq. 11). To illustrate this, a model is trained using input consisting of 16×16 image patches of pre-whitened natural images (Olshausen and Field, 1997), the model has 512 model units and model response are constrained to be nonnegative. Fig. 7 shows 512 model neuron receptive fields learnt by this SPC model with the above homeostasis function, Eq. 20. The pattern of responses illustrates that the model learns Gabor-like features similar to those of simple cells in V1 and similar to previous modelling studies based on sparse coding (Olshausen and Field, 1996, 1997; Zylberberg et al., 2011; Zhu and Rozell, 2013; Lian et al., 2019). This shows that SPC with a suitable homeostasis function can learn features and it is not confined to a specific function as used in previous models.

3.4.3 The response of model units shows contrast saturation that has not been observed in previous sparse coding models

After learning, the response vs. contrast curve was examined to investigate the response nonlinearity of the model using the above proposed homeostasis function. Fig. 8 shows that SPC with the homeostasis function (Eq. 20) can not only learn Gabor-like features but also display contrast saturation.

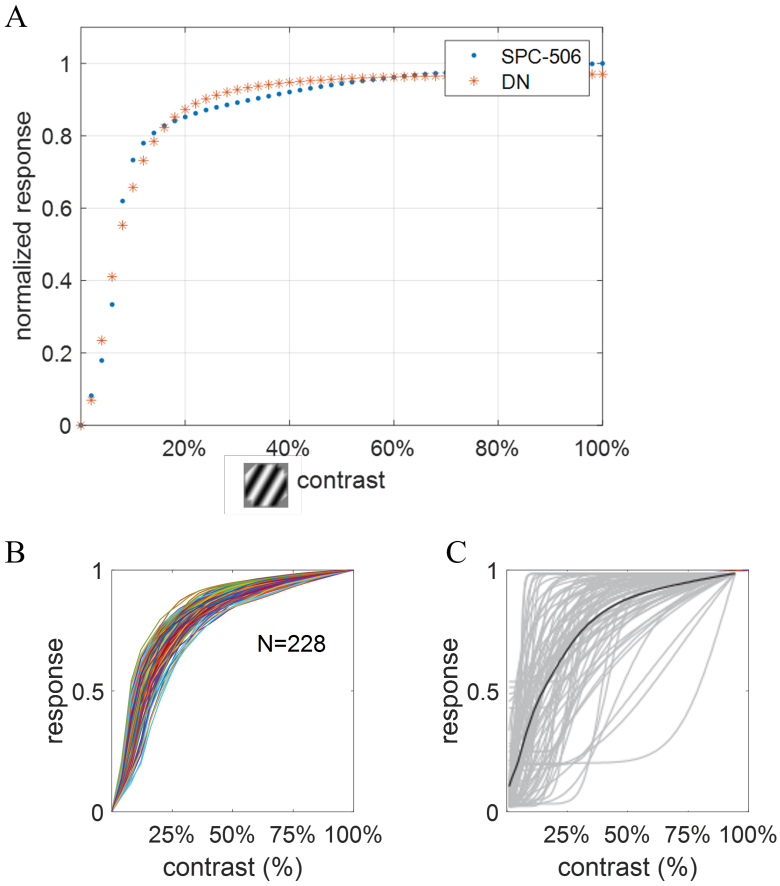

Figure 8: **Sparse/Predictive coding (SPC) model can generate contrast saturation similar to divisive normalization (DN).** (A) 'SPC-506' represents normalized response of the 506-th unit of the SPC model using the above homeostasis function, Eq.20. 'DN' shows the closest DN fit by Eq. 14. The fitted parameters of DN are: $\gamma = 0.97$, $\rho = 0.35$ and $n = 2.05$. (B) Response vs. contrast curves for 228 out of 512 learnt simple cells whose receptive fields are centered well within the 16×16 visual field. The response is scaled to a maximum of 1 for each learnt cell. (C) Response vs. contrast curves for cells in cat V1 from an experimental study (adapted from (Busse et al., 2009)). The response is also scaled to a maximum of 1 for each cell. The black line represents the population average.

For the learnt 512 simple cells, we exclude cells whose Gabor-like receptive fields are not well within the 16×16 pixel image, which leave 228 learnt cells for further analysis. Next, an exhaustive search was performed for each cell to find the preferred sinusoidal grating with preferred spatial frequency, orientation and spatial phase that gives the cell maximal response. Then the contrast (amplitude of the preferred sinusoidal grating) was changed and the response of the cell was collected to obtain its response vs. contrast curve. The response vs. contrast curve of each cell is then fitted by the equation of single-neuron DN (Eq. 14), similar

to previous studies (Heeger, 1992; Busse et al., 2009). Fig. 8A shows the response vs. contrast curve of one example unit of the SPC model with homeostasis function described in Fig. 6, which shows response saturation and the response curve can be well fitted by the equation of single-neuron DN (Eq.14). Fig. 8B shows response vs. contrast curves of 228 out of 512 learnt simple cells whose Gabor-like receptive fields are well centered within the 16×16 visual field. These 228 learnt cells display different degrees of contrast saturation, which is similar to the distribution of values for experimentally measured cells shown in Fig. 8C.

Overall, using the homeostasis function defined in Eq. 20 (Fig. 6), the SPC model possesses two important features: learning receptive field-like features and displaying nonlinearities similar to those observed experimentally. The simulations shown here, trained on natural images, demonstrate that the SPC model can learn Gabor-like simple cells as well as display contrast-saturation. Consequently, this SPC framework has great potential of describing and explaining how the neural circuits in the visual pathway possess nonlinear properties of the visual input that are observed experimentally.

4 Discussion

In this study, we aim to unify three important principles of neural information processing: sparse coding, predictive coding and divisive normalization (DN). Sparse coding and predictive coding can be identical in a two-layered neural network and unified as a sparse/predictive coding (SPC) framework whose model dynamics is determined by the predictive structure and a homeostasis function. We analytically showed that the SPC model can be identical to divisive normalization for a single-neuron scenario, namely that SPC with a homeostasis function can be derived from any divisive normalization function. Due to the difficulties of mathematically analyzing the relationship between SPC and divisive normalization within a network with multiple neurons, only simulation results of a network implementation of SPC are given. In a neural network model of V1, simulation results here show that SPC with a specific homeostasis function can learn Gabor-like simple cells, which gives similar receptive field properties to previous sparse and predictive coding model results (Olshausen and Field, 1996, 1997; Rao and Ballard, 1999), as well as displaying contrast saturation, which has previously been explained by DN (Heeger, 1992).

The work presented here is also related with other previous studies. Rubin et al. (2015) demonstrate that a

neural network with excitatory and inhibitory neurons, input-output power function, and fine-tuned recurrent connection weights can account for nonlinearities previously explained by DN. However, their work does not incorporate any learning ability of the network, nor the connection with sparse/predictive coding. Chalk et al. (2018) provides a framework of sparse and predictive coding from the perspective of information theory. However, their work does not consider divisive normalization and, unlike classical hierarchical predictive coding (Rao and Ballard, 1999), it addresses predictive coding in the temporal domain. The sparse/predictive coding model implemented here is similar to the study by Boutin et al. (2020), but their work does not consider the relationship with divisive normalization. Our work unifies sparse coding, predictive coding and divisive normalization from first principles, showing how the non-linear response properties arise through learning (i.e., synaptic plasticity).

Winner-take-all, the mechanism in which only the neuron with the highest firing rate of the neural network is allowed to fire, has been used in many studies of computational neuroscience. The beneficial property of winner-take-all is that it provides strong competition to the neural network, which assists with learning useful features from the input (Einhäuser et al., 2002; Rennó-Costa and Tort, 2017), and can be accomplished by lateral inhibition within the network (Coultrip et al., 1992). The SPC framework proposed here is closely related to the winner-take-all mechanism; indeed winner-take-all can be viewed as an extreme sparse coding model where only a single neuron or small number of neurons fire, which can be implemented in the SPC model with a strong sparsity constraint. For example, when the homeostasis function, $H(\cdot)$, is a constant function, and the value of the constant determines the fraction of active neurons in the network.

Homeostasis is a well-established and prominent feature of the brain's neural circuits, in which the brain employs a balancing mechanism to regulate neuronal excitatory and inhibitory activity (Turrigiano, 2011). In the SPC framework proposed in this study, the homeostasis function in the model dynamics, Eq. 10, regulates the response of each single neuron in a dynamic fashion, analogous to the homeostatic regulation of neuronal firing of biological neural circuits (Turrigiano, 2011). In the learning step of the SPC framework, connection weights are normalized after each update, which is equivalent to the effect of homeostatic synaptic rescaling (Turrigiano, 2011). The homeostasis function in the SPC model, however, has its origin in the mathematical derivation of the original sparse and predictive coding model, and its homeostatic role in regulating the neural circuits is somewhat fortuitous. The combined SPC-DN framework proposed

in this study combines Hebbian and homeostatic learning and may serve as a more general framework for the understanding of nonlinear neural response properties and neural learning in brain circuits (Turrigiano, 2017).

The original predictive coding formulation (Rao and Ballard, 1999) is predictive in a hierarchical fashion, i.e., the model predicts the input in lower levels of the hierarchy from the activity in the higher levels, rather than being predicting in a temporal fashion, i.e., predicting future input. However, the input to the brain typically changes over time and the brain is constantly making predictions of the future on the basis of previous sensory stimuli and neural activity. In this paper, the visual input to the model is considered as static, consisting of a sequence of stationary images. However, this framework can also be used to investigate the problem of temporal prediction by incorporating synaptic delay between adjacent layers of the network hierarchy, as done in recent studies investigating predictive coding with temporal alignment (Hogendoorn and Burkitt, 2019; Burkitt and Hogendoorn, 2021).

The SPC framework proposed in this paper unifies sparse coding and predictive coding, and bridges its connection with divisive normalization. A predictive structure with feedforward input and feedback from higher layers preserves the essence of predictive coding and the model's dynamics, given in Eq. 10, implements the dynamics of the neural activity in the network. However, a more fully biological implementation of this model would still require numerous additional features. First, the network would need to obey Dale's Law by having separate populations of excitatory and inhibitory neurons. Second, spiking neurons that learn using spike-based synaptic plasticity, such as spike-timing-dependant-plasticity, would need to be used rather than the rate-based neurons typically used. Third, a more biological realization of the homeostasis function, which could differ for different neural populations, would be required. In addition, the choice of time constants of the dynamic system, synaptic delay between subsequent layers of the hierarchy, and many other parameters would need to be matched with experimental data for the particular neural circuit being modelled. Though some of these aspects have been considered in previous studies (Denève and Machens, 2016; Lian et al., 2019), much research still remains to be undertaken.

Given the success of divisive normalization in modelling data of multisensory integration (Ohshiro et al., 2011, 2017), this SPC framework might be able to uncover how the brain learns to integrate input from

multiple sensory systems.

References

Barlow HB (1961) Possible principles underlying the transformation of sensory messages. *Sens. Commun.* 1.

Barlow HB (1989) Unsupervised learning. *Neural Comput.* 1:295–311.

Bastos AM, Usrey WM, Adams RA, Mangun GR, Fries P, Friston KJ (2012) Canonical microcircuits for predictive coding. *Neuron* 76:695–711.

Beyeler M, Rounds EL, Carlson KD, Dutt N, Krichmar JL (2019) Neural correlates of sparse coding and dimensionality reduction. *PLoS Comput. Biol.* 15:e1006908.

Boutin V, Franciosini A, Ruffier F, Perrinet L (2020) Effect of top-down connections in hierarchical sparse coding. *Neural Comput.* 32:2279–2309.

Brito CS, Gerstner W (2016) Nonlinear hebbian learning as a unifying principle in receptive field formation. *PLoS Comput. Biol.* 12:e1005070.

Burkitt AN, Hogendoorn H (2021) Predictive visual motion extrapolation emerges spontaneously and without supervision at each layer of a hierarchical neural network with spike-timing-dependent plasticity. *J. Neurosci.* 41:4428–4438.

Busse L, Wade AR, Carandini M (2009) Representation of concurrent stimuli by population activity in visual cortex. *Neuron* 64:931–942.

Carandini M, Heeger DJ (2012) Normalization as a canonical neural computation. *Nat. Rev. Neurosci.* 13:51–62.

Chalk M, Marre O, Tkačik G (2018) Toward a unified theory of efficient, predictive, and sparse coding. *Proc. Natl. Acad. Sci.* 115:186–191.

Coultrip R, Granger R, Lynch G (1992) A cortical model of winner-take-all competition via lateral inhibition. *Neural Netw.* 5:47–54.

Denève S, Machens CK (2016) Efficient codes and balanced networks. *Nat. Neurosci.* 19:375–382.

Einhäuser W, Kayser C, König P, Kording KP (2002) Learning the invariance properties of complex cells from their responses to natural stimuli. *Eur. J. Neurosci.* 15:475–486.

Freeman TC, Durand S, Kiper DC, Carandini M (2002) Suppression without inhibition in visual cortex. *Neuron* 35:759–771.

Friston K (2010) The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci.* 11:127–138.

Harpur GF, Prager RW (1996) Development of low entropy coding in a recurrent network. *Network: Comput. Neural Syst.* 7:277–284.

Heeger DJ (1992) Normalization of cell responses in cat striate cortex. *Vis. Neurosci.* 9:181–197.

Hogendoorn H, Burkitt AN (2019) Predictive coding with neural transmission delays: a real-time temporal alignment hypothesis. *eNeuro* 6.

Hubel DH, Wiesel TN (1959) Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.* 148:574–591.

Hubel DH, Wiesel TN (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* 160:106–154.

Hubel DH, Wiesel TN (1968) Receptive fields and functional architecture of monkey striate cortex. *J. Physiol.* 195:215–243.

Knill DC, Pouget A (2004) The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends Neurosci.* 27:712–719.

Lian Y, Almasi A, Grayden DB, Kameneva T, Burkitt AN, Meffin H (2021) Learning receptive field properties of complex cells in V1. *PLoS Comput. Biol.* 17:e1007957.

Lian Y, Burkitt AN (2021) Learning an efficient hippocampal place map from entorhinal inputs using non-negative sparse coding. *eNeuro* 8:1–19.

Lian Y, Burkitt AN (2022) Learning spatiotemporal properties of hippocampal place cells. *eNeuro* 9.

Lian Y, Grayden DB, Kameneva T, Meffin H, Burkitt AN (2019) Toward a biologically plausible model of LGN-V1 pathways based on efficient coding. *Front. Neural Circuits* 13:13.

Lian Y, Williams S, Alexander AS, Hasselmo ME, Burkitt AN (2023) Learning the vector coding of egocentric boundary cells from visual data. *J. Neurosci.* 43:5180–5190.

Ohshiro T, Angelaki DE, DeAngelis GC (2011) A normalization model of multisensory integration. *Nat. Neurosci.* 14:775–782.

Ohshiro T, Angelaki DE, DeAngelis GC (2017) A neural signature of divisive normalization at the level of multisensory integration in primate cortex. *Neuron* 95:399–411.

Ohzawa I, Sclar G, Freeman RD (1985) Contrast gain control in the cat's visual system. *J. Neurophysiol.* 54:651–667.

Ohzawa I, Sclar G, Freeman R (1982) Contrast gain control in the cat visual cortex. *Nature* 298:266–268.

Olsen SR, Bhandawat V, Wilson RI (2010) Divisive normalization in olfactory population codes. *Neuron* 66:287–299.

Olshausen BA, Field DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381:607–609.

Olshausen BA, Field DJ (1997) Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Res.* 37:3311–3325.

Papyan V, Romano Y, Elad M (2017) Convolutional neural networks analyzed via convolutional sparse coding. *J. Mach. Learn. Res.* 18:2887–2938.

Rao RP, Ballard DH (1999) Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nat. Neurosci.* 2:79–87.

Rennó-Costa C, Tort AB (2017) Place and grid cells in a loop: implications for memory function and spatial coding. *J. Neurosci.* 37:8062–8076.

Rozell CJ, Johnson DH, Baraniuk RG, Olshausen BA (2008) Sparse coding via thresholding and local competition in neural circuits. *Neural Comput.* 20:2526–2563.

Rubin DB, Van Hooser SD, Miller KD (2015) The stabilized supralinear network: a unifying circuit motif underlying multi-input integration in sensory cortex. *Neuron* 85:402–417.

Smith EC, Lewicki MS (2006) Efficient auditory coding. *Nature* 439:978–982.

Spratling MW (2008) Reconciling predictive coding and biased competition models of cortical function. *Front. Comput. Neurosci.* 2:4.

Turrigiano G (2011) Too many cooks? Intrinsic and synaptic homeostatic mechanisms in cortical circuit refinement. *Ann. Rev. Neurosci.* 34:89–103.

Turrigiano GG (2017) The dialectic of hebb and homeostasis. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 372:20160258.

Zhu M, Rozell CJ (2013) Visual nonclassical receptive field effects emerge from sparse coding in a dynamical system. *PLoS Comput. Biol.* 9:e1003191.

Zylberberg J, Murphy J, DeWeese M (2011) A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of V1 simple cell receptive fields. *PLoS Comput. Biol.* 7:e1002250.

A Appendix

A.1 Choice of $H(\cdot)$

In the original papers of sparse coding (Olshausen and Field, 1997) and predictive coding (Rao and Ballard, 1999), the choices of $Q(s)$ considered were

$$Q(s) = \ln\left[1 + \left(\frac{s}{\sigma}\right)^2\right] \ , \ -e^{-\left(\frac{s}{\sigma}\right)^2} \ , \ |s| \ \text{ or } \ s^2 \ , \tag{A.1}$$

where σ is a positive scaling factor. With the assumption of nonnegative responses, we consider here the three different functions for $Q(s)$:

$$Q(s) = \ln\left[1 + (\frac{s}{\sigma})^2\right] \ -e^{-(\frac{s}{\sigma})^2} \ , \ \text{ or } \ s^\alpha, \tag{A.2}$$

where α is a positive power index. The corresponding homeostasis functions are $\lambda Q'(s)$, namely:

$$H(s) = \lambda \frac{s}{1 + (s/\sigma)^2} \ , \ \lambda s e^{-(\frac{s}{\sigma})^2} \ \text{ or } \ \lambda s^{\alpha-1} \quad (\text{with } s \geq 0) \ , \tag{A.3}$$

where the constant parameters are absorbed into the single constant parameter, λ.

Some examples of homeostasis functions, $H(\cdot)$, are shown in Fig. A.1. For power functions, when $H(s) = 1$ ($\alpha = 1$), the cost the homeostasis puts on model neurons is the same for different firing rates; when α is smaller than 1, the cost of generating small firing rates is extremely high (infinite cost for zero firing rate) and the cost decreases dramatically for larger firing rates, i.e., when $\alpha > 1$, the cost of generating firing rates starts from 0 and increases without bound as the firing rate increases. For $H(s) = 2s/(1 + s^2)$ and $H(s) = 2se^{-s^2}$, the cost starts from zero, reaches the maximum and then reduces to zero as the firing rate increases. Overall, the choice of $H(\cdot)$ is crucial because it partly determines the dynamics of the model, as described by Eq. 10.

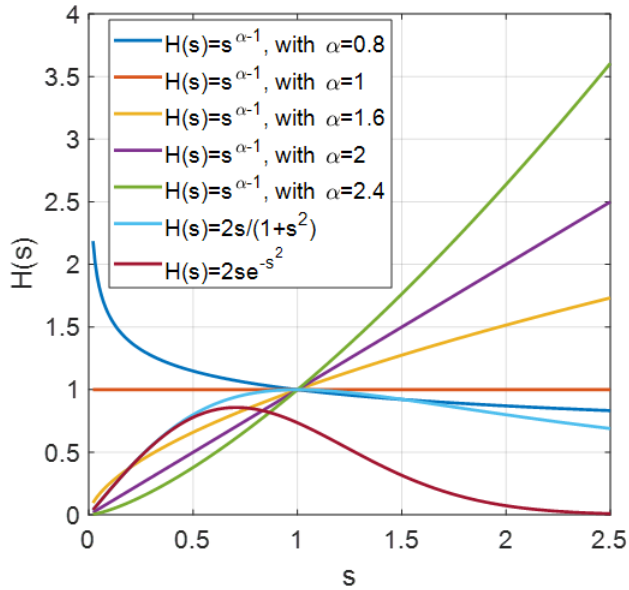

Figure A.1: **Examples of homeostasis functions.** These functions are examples of Eq. A.3. λ is set to 1 for power index functions. For the remaining functions, λ is set to 2 and σ is set to 1 so that all functions can be plotted in the same range.

A.2 When $H(s) \neq 0$

If there is a homeostatic fixed point of the activity, namely a value of the firing rate $\mathbf{s}^{\mathrm{EQ}}$ such that $H(\mathbf{s}^{\mathrm{EQ}})$ is non-zero, then generally the equilibrium value $\mathbf{s}^{\mathrm{EQ}}$ does not have an explicit analytical formula, except for some special choices of $H(\cdot)$ such as $H(s) = c$ where c is a constant.

If $H(s) = c$, Eq. 13 becomes

$$-\mathbf{A}^T\mathbf{A}\mathbf{s}^{\mathrm{EQ}} + \mathbf{A}^T\mathbf{x} = \mathbf{c}, \tag{A.4}$$

where $\mathbf{c}$ is a vector whose elements are all equal to c. Similar to the analysis above, the equilibrium, $\mathbf{s}^{\mathrm{EQ}}$ can then be written as

$$\mathbf{s}^{\mathrm{EQ}} = (\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{x} - \mathbf{c}), \tag{A.5}$$

when $\mathbf{A}^T\mathbf{A}$ is invertible.

For other more complex choices of $H(\cdot)$, an explicit formula for the analytical equilibrium has not been found, though its value maybe determined from the intersection between $Y(\mathbf{s}) = -\mathbf{A}^T\mathbf{A}\mathbf{s} + \mathbf{A}^T\mathbf{x}$ and

$H(\mathbf{s})$, as show in Fig. 3.

A.3 When $H(s) = 0$

If there is no homeostasis (i.e., $H(s) = 0$), Eq. 13 becomes

$$\mathbf{A}^T\mathbf{A}\mathbf{s}^{\mathrm{EQ}} = \mathbf{A}^T\mathbf{x}. \tag{A.6}$$

Since $\mathbf{A}^T\mathbf{A}$ is positive semi-definite, $\mathbf{A}^T\mathbf{A}$ will be invertible (i.e., positive definite) if we assume $\mathbf{A}\mathbf{z} \neq \mathbf{0}$ for any vector $\mathbf{z} \neq \mathbf{0}$. This condition is generally met for sensory system in the brain, since there will be neurons responding to a non-zero input stimulus. The equilibrium, $\mathbf{s}^{\mathrm{EQ}}$ can then be written as

$$\mathbf{s}^{\mathrm{EQ}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x}. \tag{A.7}$$

The equilibrium of SPC model with no homeostasis is still linear, but the equivalent feedforward connection becomes $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$. In addition, the amplitude of the model responses will increase linearly with the amplitude of the input. If $\mathbf{A}^T\mathbf{A} = \mathbb{1}$, the equilibrium of the SPC model will be identical to the response of a linear feedforward model with input $\mathbf{x}$ and feedforward connection weight matrix $\mathbf{A}^T$. Given that the dynamic equations Eq. 12 describe a first order differential equation, the model responses, $\mathbf{s}$, will approach the equilibrium, $\mathbf{s}^{\mathrm{EQ}}$, as

$$\mathbf{s}(t) = [\mathbf{v}_1\ \mathbf{v}_2\ \cdots \mathbf{v}_M] \begin{bmatrix} \mathbf{v}_1^T\mathbf{s}(0)e^{\lambda_1 t} \\ \mathbf{v}_2^T\mathbf{s}(0)e^{\lambda_2 t} \\ \vdots \\ \mathbf{v}_M^T\mathbf{s}(0)e^{\lambda_M t} \end{bmatrix} + \mathbf{s}^{\mathrm{EQ}}\ , \tag{A.8}$$

where $\mathbf{v}_i$ $(i = 1, 2, \cdots, M)$ are the eigenvectors and λ_i $(i = 1, 2, \cdots, M)$ are the corresponding eigenvalues of $-\mathbf{A}^T\mathbf{A}$, and $\mathbf{s}(0)$ is the initial response at time 0. Because $-\mathbf{A}^T\mathbf{A}$ is negative definite with the assumptions above, λ_i $(i = 1, 2, \cdots, M)$ will be all negative, meaning that the first term on the right hand side of Eq. A.8 will reduce to zero exponentially.

Figs. A.2 & A.3 show the simulation results to validate the analysis above when there is no homeostasis in the model. In the simulation there are 256 units in the first layer ($L = 256$) and 256 units in the second

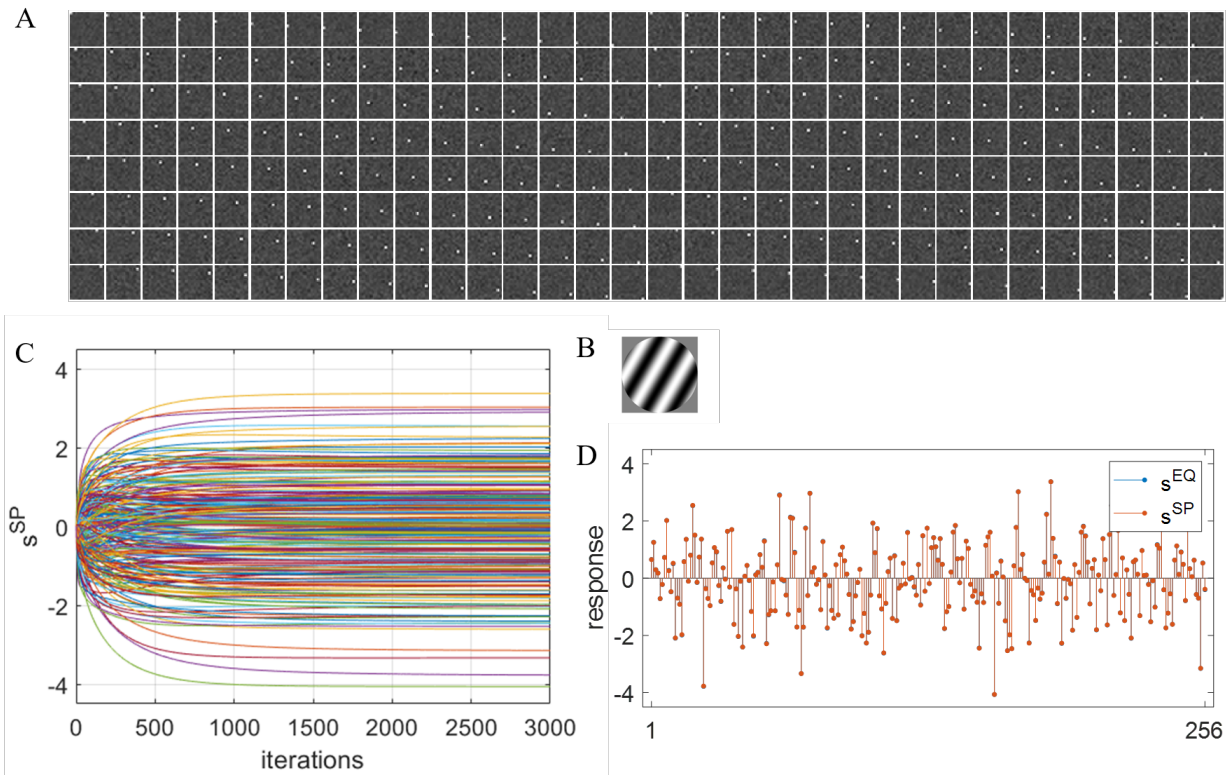

Figure A.2: **Model response converges to the analytical equilibrium.** There is no homeostasis in this simulation. (A) Basis vectors of $\mathbf{A}$. Each square block is a 16×16 image. $\mathbf{A}$ is defined in Eq A.9 and close to an identity matrix, so each basis vector has one dominant element. (B) The 16×16 input image. (C) Response trajectory. (D) The response of analytical equilibrium and response of SPC model after 3000 iterations of computing model responses with $\eta_s = 0.05$.

layer ($M = 256$). Therefore $\mathbf{A}$ is a 256×256 matrix and each column is a basis vector used to reconstruct the input image. The input is a 16×16 image of sinusoidal grating (Fig. A.2B). There are 3000 iterations of Eq. 10 to compute the model responses with $\eta_s = 0.05$ and there is no learning process of $\mathbf{A}$ in this simulation. $\mathbf{A}$ is chosen differently for Figs. A.2 and A.3.

The connection $\mathbf{A}$ in Fig. A.2 is chosen as

$$\mathbf{A} = \mathbb{1} + \mathbb{N} \;\text{ with }\; \|\mathbf{A}\|_2 = 1 \;, \tag{A.9}$$

where $\mathbb{1}$ is an 256×256 identity matrix and $\mathbb{N}$ is an 256×256 matrix whose elements are sampled from a Gaussian distribution with mean 0 and standard deviation 0.05. $\mathbf{A}$ is then normalized to have l_2 norm 1.

$\mathbf{A}$ is chosen this way to ensure that $\mathbf{A}^T\mathbf{A}$ is a full-rank matrix and the eigenvalues of $\mathbf{A}^T\mathbf{A}$ are all different but separated from 0. Fig. A.2A displays the 256 basis vectors and these vectors are used to reconstruct the input image (Fig. A.2B). Fig. A.2C shows that the model response becomes stable within 3000 iterations. The analytical equilibrium $\mathbf{s}^{\mathrm{EQ}}$ (Eq. A.7) is plotted as red spikes and the model response $\mathbf{s}^{\mathrm{SP}}$ after 3000 iterations is plotted in blue spikes in Fig. A.2D. Fig. A.2D shows that the model response is identical to the analytical equilibrium. The summed squared error between $\mathbf{s}^{\mathrm{EQ}}$ and $\mathbf{s}^{\mathrm{SP}}$ is only 0.0036% and the summed squared error between input ($\mathbf{x}$) and reconstruction ($\mathbf{As}$) is 0. In other words, given the current input and fixed connection, the model can essentially represent the input perfectly and the model response converges to the analytical value.

Although the model has an analytical equilibrium described by Eq. A.7, it does not guarantee that model response will converge to the analytical equilibrium within any reasonable time. On the contrary, the iterated model response might be far away from the equilibrium while the model response can still reconstruct the input very well, which is illustrated in Fig. A.3.

In Fig. A.3, the connection weights $\mathbf{A}$ are chosen to be some Gabor-like filters that are learnt by a standard sparse coding model (Olshausen and Field, 1996, 1997; Rozell et al., 2008) and $\|\mathbf{A}\|_2 = 1$. Fig. A.3A shows the 256 basis filters of $\mathbf{A}$, which look similar to simple cells found in the visual cortex (Hubel and Wiesel, 1959). Though $\mathbf{A}^T\mathbf{A}$ is invertible with all positive eigenvalues, some eigenvalues are very small (the smallest is 4×10^{-10}). Therefore, the analytical response described by Eq. A.7 may become very large. This is illustrated in Fig. A.3B, which shows that the analytical equilibrium is quite large and the amplitude in this example can grow as large as 500. Fig. A.3C shows the trajectory of model response over these 3000 iterations of implementing model dynamics with $\eta_s = 0.05$ (the same as Fig. A.2). However, it appears that model responses do not converge after 3000 iterations. The model responses after 3000 iterations are plotted in Fig. A.3D. Compared with the analytical equilibrium shown in Fig. A.3B, the amplitude of model response is much smaller than the analytical equilibrium. Apart from the different amplitudes of responses, the analytical equilibrium and model response also differ in the population responses, which are illustrated in Fig. A.3E, that shows that the responses are very different even after normalizing the responses to the same scale. The summed squared error between the equilibrium and model responses is 99.99%. The difference between analytical and simulated results may be caused by the discrete implementation of the model and the

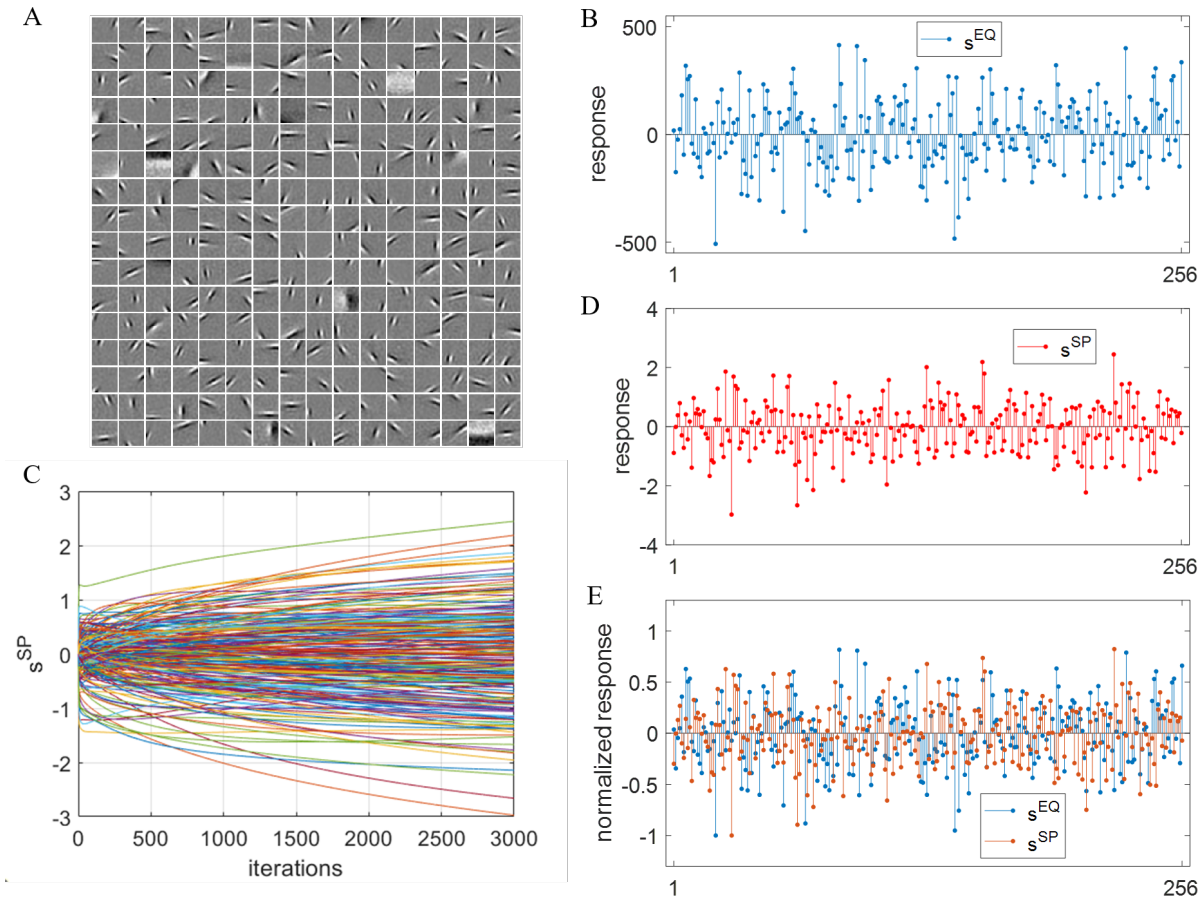

Figure A.3: **Model response does not converge to the analytical equilibrium.** There is no homeostasis in this simulation and the input to the model is the same as Fig. A.2B. (A) Basis vectors of $\mathbf{A}$. Each square block is a 16×16 image. $\mathbf{A}$ is learned using a standard sparse coding model and each basis filter is similar to simple cells in V1. (B) $\mathbf{s}^{\mathrm{EQ}}$, the analytical equilibrium determined by Eq. A.7. (C) Response trajectory. (D) Model response, $\mathbf{s}^{\mathrm{SP}}$, after 3000 iterations. (E) The normalized response of analytical equilibrium and model response after 3000 iterations with $\eta_s = 0.05$.

very small eigenvalues of $\mathbf{A}^T\mathbf{A}$, which takes much longer for the model response to converge, as described by Eq. A.8. Despite the significant difference between the analytical equilibrium and model simulation response, the model can still reconstruct the input well, with a small summed squared error of 0.8% and the average energy of the model response (mean of the squared response) is 0.77. However, if there are only 300 iterations, the reconstruction error of the model is 1.5% and the average energy is 0.21. Therefore, there seems to be a trade-off between the model accuracy and computation cost (including both time and energy). In practice, although $\mathbf{A}^T\mathbf{A}$ is mostly invertible, some of the eigenvalues can easily be very small, as

illustrated in Fig. A.3. Also, if the standard deviation of $\mathbb{N}$ in Eq. A.9 becomes 0.1, the smallest eigenvalues of $\mathbf{A}^T\mathbf{A}$ would be 9.9×10^{-7}. Therefore, though the discrete online implementation of the model is not as accurate as the analytical result, it nevertheless provides a reasonable response within a reasonable time.

A.4 Nonlinearity

The SPC model implemented in our simulations have non-negative firing rates. The nonlinearity introduced into the SPC model is two-fold: through the nonnegativity of the firing rates and through the homeostasis function.

As analyzed in the previous section, the equilibrium of the SPC model is linearly dependent on the input when there is no constraint of nonnegative firing rate and homeostasis (Eq. A.7), given that $\mathbf{A}^T\mathbf{A}$ is invertible. However, $\mathbf{s} \geq \mathbf{0}$ has a dramatic effect on the network, because this constraint introduces great nonlinearity. Somewhat surprisingly, the nonnegativity constraint accelerates the convergence of the model responses and it becomes stable after 1500 iterations (Fig. A.4A) compared with the case without the nonnegativity constraint (Fig. A.3C). However, because of the nonnegativity of $\mathbf{s}$, the model cannot reconstructs the input with negative responses though the basis vectors $\mathbf{A}$ in Fig. A.3 are learned in a standard sparse coding that allows model responses to be negative, which leads to a larger reconstruction error of 11.1% after incorporating $\mathbf{s} \geq \mathbf{0}$. Fig. A.4B illustrates the model responses after 3000 iterations when $\mathbf{s} \geq \mathbf{0}$ is incorporated and it shows a clear difference from the analytical response ($\mathbf{s}^{\mathrm{EQ}}$ in Fig. A.3B) and the model responses without the nonnegativity constraint (Fig. A.3D). After the nonnegativity of model response is incorporated, one way to preserve the expressiveness of the basis vectors is to extend the basis vectors to be $[\mathbf{A} \ -\mathbf{A}]$; i.e., similar to the study by Zhu and Rozell (2013). Therefore, there are 512 model units, instead of 256 model units in this case. Figs. A.4C & D show that the convergence of model units is lost again and more model units are active, which leads to a better reconstruction of the input with 0.8% error.

Apart from the nonnegativity of the model response, the homeostasis, $H(s)$, also introduces nonlinearity into the model, since the system becomes nonlinear if $H(s)$ is not a first-order polynomial of s. Even when there is no nonnegative constraint on the model response, the choice of $H(s)$ can determine the model responses and ultimately determines whether the model can learn structures from the input. For example, only suitable choices of sparsity functions can lead to Gabor-like simple cells after learning (Olshausen and

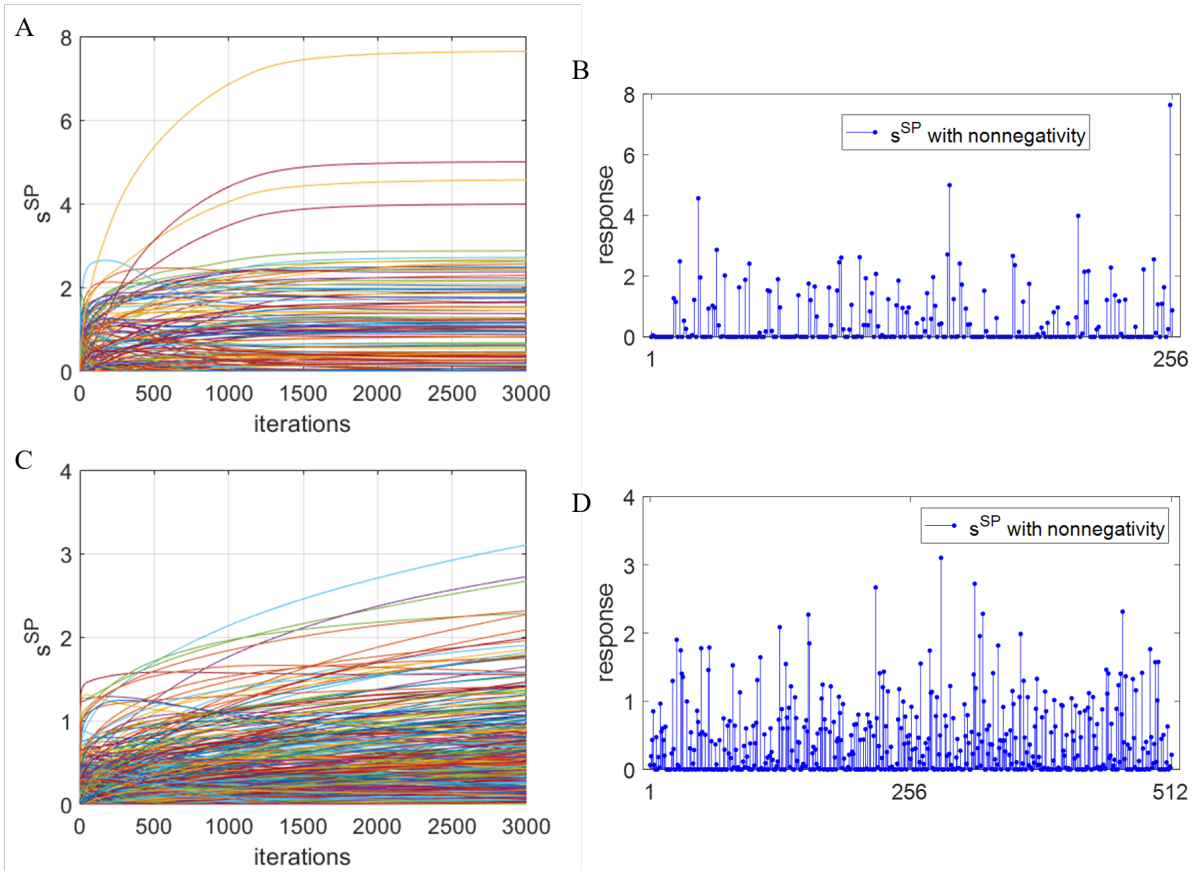

Figure A.4: **Model responses with nonnegative constraint.** The input to the model is the same as Fig. A.2B. (A, C) Response trajectory. (B, D) $\mathbf{s}^{\mathrm{SP}}$, after 3000 iterations. For (A) and (B), the same basis vectors $\mathbf{A}$ as in Fig. A.3A are used. For (C) and (D), the basis vectors are extend to be $[\mathbf{A} \ -\mathbf{A}]$ and the number of model units also doubles, so there are 512 model units in (C) and (D).

Field, 1996, 1997; Rao and Ballard, 1999).

A.5 Competition

SPC model incorporates competition into the neural network via the homeostasis function and the model structure.

The homeostasis function, $H(\cdot)$ in Eq. 10, introduces competition because it directly regulates the response of model units and it can put different penalties on different units based on their response level.

Apart from the homeostasis function, the structure of SPC model itself (Fig. 2) has a competitive nature because it explicitly implements a form of competition, as demonstrated by Spratling (2008). Suppose

there is no homeostasis (i.e., $H(s) = 0$) in the dynamic equation of SPC, Eq. 10 essentially becomes the implementation of Harpur's "negative feedback network" (Harpur and Prager, 1996) that causes competition between model units such that they compete to represent the input. That competition can also be seen from the analytical equilibrium described in Eq. A.7, where $(\mathbf{A}^T\mathbf{A})^{-1}$ will always scale the linear response $(\mathbf{A}^T\mathbf{x})$ based on the correlation between different basis vectors. Lian et al. (2021) show that the competition incorporated by this structure can be very strong such that learning complex cells from simple cells can be very challenging.

Because of this competitive nature of SPC model, sparse coding has been used to explain some nonclassical receptive field properties of V1 such as end-stopping, contrast invariance of orientation tuning (Zhu and Rozell, 2013). However, this study failed to explain contrast saturation, which is another important property of V1 neurons.