Systems/Circuits

The Sparseness of Mixed Selectivity Neurons Controls the Generalization–Discrimination Trade-Off

Omri Barak,[1] **Mattia Rigotti,**[1,2] **and Stefano Fusi**[1,3]

[1]Center for Theoretical Neuroscience, Department of Neuroscience, Columbia University Medical Center, New York, New York 10032, [2]Center for Neural Science, New York University, New York 10003, and [3]Kavli Institute for Brain Sciences, Columbia University Medical Center, New York, New York 10032

Intelligent behavior requires integrating several sources of information in a meaningful fashion—be it context with stimulus or shape with color and size. This requires the underlying neural mechanism to respond in a different manner to similar inputs (discrimination), while maintaining a consistent response for noisy variations of the same input (generalization). We show that neurons that mix information sources via random connectivity can form an easy to read representation of input combinations. Using analytical and numerical tools, we show that the coding level or sparseness of these neurons' activity controls a trade-off between generalization and discrimination, with the optimal level depending on the task at hand. In all realistic situations that we analyzed, the optimal fraction of inputs to which a neuron responds is close to 0.1. Finally, we predict a relation between a measurable property of the neural representation and task performance.

Introduction

How do we determine whether a neural representation is good or bad? In general the answer depends on several factors, which include the statistics of the quantity that is represented, the task to be executed, and the neural readout that utilizes the representation.

Previous work evaluated neural representations on the basis of the information they encode (Atick and Redlich, 1992; Jazayeri and Movshon, 2006). This is often the only viable approach when it is not known how the representations are used or read out by downstream structures (e.g., in the case of early sensory areas).

Here we evaluate a neural representation by the information that is accessible to individual readout neurons, which we assume simply compute a weighted sum of the inputs followed by a thresholding operation. In general, this information is smaller than the total information contained in the input, as it is constrained to be in a more "explicit" format (DiCarlo et al., 2012) suitable for being processed by simple readouts.

Previous studies evaluated neural representations of natural visual scenes by computing the reconstruction error of a population of linear readout neurons (Olshausen and Field, 2004). These elegant works showed that sparseness is an important feature of the neural representations, not only because it naturally leads to the receptive fields observed in cortical recordings, but it also increases the dimensionality of the input, facilitates learning, and reduces the effects of input noise.

We focus on the capacity of a readout neuron to produce a large set of diverse responses to the same inputs (i.e., to implement a large number of input–output functions). This capacity clearly depends on the input representation, and it is functionally important as it can be harnessed to generate rich dynamics and perform complex tasks (Hinton and Anderson, 1989; Rigotti et al., 2010b). We consider a specific class of problems in which readout neurons receive inputs from multiple sources (Fig. 1*A*). This situation is encountered in many cases, which include integration of sensory modalities, combining an internally represented context with a sensory stimulus, or mixing the recurrent and the external input of a neural circuit. These are typical situations in almost every brain area, especially in those integrating inputs from multiple brain systems, such as the prefrontal cortex (Miller and Cohen, 2001). As the readout is linear, in these situations there are some input–output functions that cannot be implemented (Fig. 1*B*). For instance, the ability to differentiate between external inputs that are received in different contexts is known to potentially generate a large number of non-implementable functions (McClelland and Rumelhart, 1985; Rigotti et al., 2010b). The difficulty stems from the high correlations between the input patterns that only differ by the state of one segregated information source (e.g., the one encoding the context).

Fortunately, there are transformations implemented by simple neuronal circuits that decorrelate the inputs by mixing different sources of information in a nonlinear way. We will focus on one such transformation that is implemented by introducing an intermediate layer of randomly connected neurons (RCNs; Fig. 1*C*). Each RCN responds nonlinearly to the weighted sum of the original inputs, and its weights are random and statistically independent. These neurons typically respond to complex combina-

Received June 8, 2012; revised Oct. 31, 2012; accepted Dec. 26, 2012.

Author contributions: O.B., M.R., and S.F. designed research; O.B., M.R., and S.F. performed research; O.B., M.R., and S.F. wrote the paper.

This work was supported by funding from the Gatsby Foundation, the Kavli Foundation, the Sloan-Swartz Foundation, and Defense Advanced Research Projects Agency Grant SyNAPSE HR0011-09-C-0002. M.R. is supported by Swiss National Science Foundation Grant PBSKP3-133357. We are grateful to Larry Abbott, Marcus Benna, René Hen, Dani Martí, and Nicole Rust for comments on this manuscript.

Correspondence should be addressed to Stefano Fusi, Center for Theoretical Neuroscience, Department of Neuroscience, 1051 Riverside Drive, New York, NY 10032. E-mail: sf2237@columbia.edu.

DOI:10.1523/JNEUROSCI.2753-12.2013

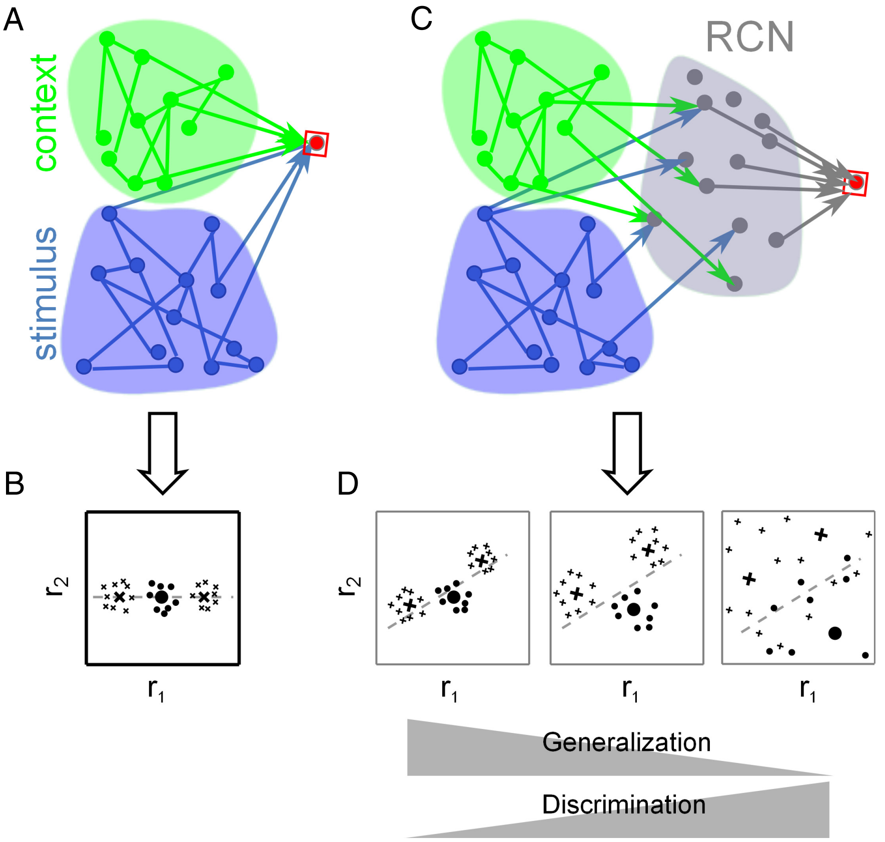

Figure 1. The challenge of integrating sources of information in the presence of noise. ***A***, A single neuron (red) receiving input from two sources (green and blue, representing for instance a sensory stimulus and the context in which it appears). ***B***, Representations in the input space that are not linearly separable (typical situation when multiple sources of information are integrated). The axes of the plane represent independent patterns of activity of the input neurons, for instance the firing rate of two different neurons. Each point on the plane represents a different input activity pattern, and the symbols represent how patterns should be classified. For example, crosses are inputs that should activate the readout neuron, and circles are inputs that should inactivate it. The inputs to be classified are constructed as noisy variations of three prototypes (large symbols) that represent three different classes. In this example, the correlations between the inputs constrain the large symbols to lie on a line, making the classification problem linearly nonseparable (i.e., there is no single line separating the crosses from the circles). ***C***, An intermediate layer of randomly connected neurons, RCNs, solves the linear separability problem. ***D***, Neural representations in the RCN space for three different transformations performed by the RCNs on the input space. The axes now represent the activity of two RCNs. For all the transformations, the dimensionality of the inputs increases (the prototypes spread out on a plane), aiding discrimination (distance between large symbols). But too much decorrelation (right) can amplify noise and degrade the generalization ability (dispersion of small symbols). The sparseness or the coding level of the RCNs mediates this generalization–discrimination tradeoff.

tions of the parameters characterizing the different sources of information (mixed selectivity), such as a sensory stimulus only when it appears in a specific context. Mixed selectivity neurons have been widely observed in the cortex (Asaad et al., 1998; Rigotti et al., 2010a, 2013; Warden and Miller, 2010), although they are rarely studied, as their response properties are difficult to interpret. Neural representations that contain RCNs allow a linear readout to implement a large number of input–output functions (Marr, 1969; Hinton and Anderson, 1989; Maass et al., 2002; Lukoševičius and Jaeger, 2009; Rigotti et al., 2010b).

Any transformation that mixes multiple sources of information, should reconcile the two opposing needs of the discrimination–generalization trade-off (Fig. 1*D*). It should decorrelate the representations sufficiently to increase classification capacity, which is related to the ability to discriminate between similar inputs. Unfortunately, as we will demonstrate (Fig. 4*B*), transformations that decorrelate tend to destroy the information about relative distances in the original space, making it harder for the readout neurons to generalize (i.e., generate the same output to unknown variations of the inputs).

We will show that RCNs can efficiently decorrelate the inputs without sacrificing the ability to generalize. The discrimination–generalization trade-off can be biased by varying the sparseness of the RCN representations, and there is an optimal sparseness that minimizes the classification error.

Materials and Methods

Definition of the task. For simplicity we report here the analysis of the case with two sources of information. The case with more than two sources is a straightforward extension and is briefly discussed at the end of this section. We consider two network architectures—one with an RCN layer (Fig. 4*A*), and one without (Fig. 2*A*). The activity of all neurons is approximated as binary.

In both cases, the first layer is an input composed of two sources containing N neurons each. The first source $\psi^x \in \{\pm 1\}^N$ can be in one of m_1 states, $x = 1, \ldots, m_1$, and the second source is denoted by $\phi^a \in \{\pm 1\}^N$ with $a = 1, \ldots, m_2$. An input pattern ξ^μ is composed of one subpattern from each source $\xi^{xa} = (\psi^x\ \phi^a)^T$, where each pattern μ can be denoted by its constituent subpatterns $\mu = (x, a)$. All subpatterns are random and uncorrelated with equal probability for $+1$ or -1. There are $p = m_1 m_2$ possible composite patterns composed of all possible combinations of the subpatterns.

Each pattern is assigned a random desired output $\eta^\mu \in \{\pm 1\}$, and the task is to find a linear readout defined by weights W such that the sign of the projection of the activity of the last layer (input or RCN, Fig. 1, *A* and *B*, respectively) onto it will match the desired output.

No RCNs. In this case the task can be written in vector notation as:

$$\mathrm{sign}(W^T Q) = \eta, \qquad (1)$$

where $Q_{i,\mu} = \xi_i^\mu$ is a $2N \times p$ matrix that contains all input patterns. Note that we assume a zero threshold for the readout for simplicity. We show below that this choice has no effect on the scaling properties we are interested in.

Since we are using random outputs, the classification ability depends only on the structure of the input. We first show that the matrix Q is low dimensional. Consider a case of $m_1 = 2$ and $m_2 = 3$:

$$Q = \underbrace{\begin{pmatrix} \psi^1 & \psi^1 & \psi^1 & \psi^2 & \psi^2 & \psi^2 \\ \phi^1 & \phi^2 & \phi^3 & \phi^1 & \phi^2 & \phi^3 \end{pmatrix}}_{6} \Big\} 2N. \qquad (2)$$

This matrix can be written as

$$Q = \begin{pmatrix} \psi^1 & \psi^2 - \psi^1 & 0 & 0 \\ \phi^1 & 0 & \phi^2 - \phi^1 & \phi^3 - \phi^1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \qquad (3)$$

showing that it is in fact only of rank 4. In general, the rank will be $(m_1 - 1) + (m_2 - 1) + 1$.

The rank of this matrix determines the effective number of inputs to the readout neuron (Barak and Rigotti, 2011), which in turn affects the possible number of patterns that can be classified.

Once the number of patterns exceeds the capacity, which is two times the number of independent inputs or rank, we expect classification to be at chance level (Cover, 1965; Hertz et al., 1991; Barak and Rigotti, 2011). To verify this, we considered $m_1 = m_2 = 5$ and a subset of $\tilde{p} = 1, \ldots ,25$ patterns. For each value of $\tilde{p}$ we computed the rank of Q and the fraction of patterns that were classified correctly (average from 500 random choices of η). This was repeated for $m_1 = m_2 = 10$ and $m_1 = m_2 = 15$ (Fig. 2*D*,*E*).

The RCN layer. To solve the linear separability problem, we introduce an intermediate layer of randomly connected neurons. The input patterns are projected to N_{RCN} randomly connected neurons through weights $J_{ij} \approx \mathcal{N}\ (0, N/2)$, where $i = 1, \ldots, N_{RCN}$, $j = 1, \ldots, 2N$ and $\mathcal{N}_x(\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2.}}$. A threshold θ is applied to the RCNs, defining a coding level $f = \text{erfc}\ (\theta/\sqrt{2})/2$, which is the fraction of all possible input patterns that activate a given RCN. For a pattern $\xi^{xa} = (\psi^x\ \phi^a)^T$, the activity of the ith RCN is given by

$$S_i^{xa} = \text{sign}\left(\sum_{j=1}^{2N} J_{ij}\xi_j^{xa} - \theta\right)$$

$$= \text{sign}\left(\sum_{j=1}^{N} J_{ij}\psi_j^x + \sum_{j=N+1}^{2N} J_{ij}\phi_{j-N}^a - \theta\right). \quad (4)$$

The task can now be written as finding a W such that sign $(W^T S) = \eta$, where the $N_{RCN} \times p$ matrix S is the activity of the RCNs due to all input patterns. The different layers can be schematically described in the following diagram:

$$\xi_j^\mu \xrightarrow{J_{ij}} g_i^\mu \xrightarrow{\theta,\text{sign}} S_i^\mu \xrightarrow{W_i} h^\mu \xrightarrow{\text{sign}} \eta^\mu. \quad (5)$$

RCN classification without noise. As before, we consider the rank of the matrix S. A single RCN can add at most 1 to the rank of the pattern matrix, and Figure 3 shows that this is indeed the case for sufficiently large p and sufficiently high f. We quantified this behavior by determining, for each value of p and each coding level f, the minimal number of RCNs required to classify 95% of the patterns correctly. This value was 0.5p for the high f (dense coding) case, as expected from Cover's Theorem (Cover, 1965). We determined the critical coding level at which this fraction increased to 0.75p and saw that it decreased approximately as $p^{-0.8}$.

RCN classification with noise. We introduce noise by flipping the activity of a random fraction n of the $2N$ elements of the input patterns ξ and propagating this noise to the RCN patterns S.

The heuristic calculations of generalization and discrimination in Figure 4 were done by choosing $m_1 = m_2 = 2$ and $n = 0.1$. Consistent RCNs were defined as those that maintained the same activity level (sign of their input) in response to two noisy versions of the same input pattern. Discriminating RCNs were defined as those that had a different activity level in response to two patterns differing by only one subpattern.

The test error of the readout from the RCNs depends on how the readout weights are set. If we have p patterns in N_{RCN} dimensions and the noise is isotropic in this space, the readout that minimizes errors is the one that has the maximal margin between the patterns and the separating hyperplane defined by the weights W (Krauth et al., 1988). Because the noise originates in the input layer, each RCN has a different probability to flip, and the isotropic assumption is not true. Nevertheless, we use the mean patterns in RCN space to derive the maximal margin hyperplane and use this weight vector as a readout (for a possible alternative, Xue et al. (2011)). We also trained a readout using online learning from noisy inputs, and while the error decreased, the qualitative features we report did not differ.

Because we are interested in the shape of the error curve and not its absolute values, we adjusted the number of RCNs to avoid floor and ceiling effects. Specifically, as we varied the noise we used Equation 20 to estimate the number of RCNs that would produce a minimal error of 10%.

Approximation of the test error. While the paper presents extensive numerical simulations of a wide range of parameters, we are also interested in analytical approximations that can provide better insight and help understand various scaling properties of the system. Furthermore, we would like to estimate the error from experimentally accessible quantities, and our analytical approximations help us in this regard. Given readout weights W, every pattern μ has a distribution of projections onto this readout due to the input noise. We approximate these as Gaussian, defining κ_μ and Σ_μ as the mean and variance, respectively:

$$W^T S^\mu \approx \mathcal{N}(\kappa_\mu, \textstyle\sum_\mu^2).$$

The test error (probability of misclassification) is then given by:

$$\text{err}_{\text{test}}^\mu = \frac{1}{2}\,\text{erfc}\left(\frac{\kappa_\mu}{\sqrt{2\sum_\mu^2}}\right). \quad (6)$$

We approximate the average test error by inserting the averages inside the nonlinearity (which is somewhat reasonable given that we are interested in errors far from saturation effects):

$$\langle \text{err}_{\text{test}}^\mu \rangle_\mu \approx \frac{1}{2}\,\text{erfc}\left(\frac{\langle \kappa_\mu \rangle_\mu}{\sqrt{2\left\langle \sum_\mu^2 \right\rangle_\mu}}\right) \overset{def}{=} \frac{1}{2}\,\text{erfc}\left(\frac{\kappa}{\sqrt{2\sum^2}}\right). \quad (7)$$

To approximate κ, we note that the perceptron margin, $\min_\mu \kappa_\mu$ can be bounded by the minimal eigenvalue λ of the matrix $M = S^T S$ (Barak and Rigotti, 2011):

$$\min_\mu \kappa_\mu \geq \sqrt{\frac{\lambda}{p}}. \quad (8)$$

Because we are in the regime where there are many more RCNs than patterns, and classification is hampered by noise, we expect the margin to be a good approximation to κ. We thus proceed to estimate λ from the matrix M.

The matrix M defined above is a random matrix (across realizations of ξ, which are assumed to be random). To obtain the distribution of its minimal eigenvalue, we should first derive the eigenvalue and only then average over realizations of the matrix. Nevertheless, as an approximation we consider the minimal eigenvalue of the average matrix $\bar{M} = \langle M \rangle_\xi$. This provides an upper bound on λ, which in turn gives a lower bound on the margin.

As stated above, it is useful to expand the pattern indices into their constituents: $\mu = (x, a)$, $\nu = (y, b)$. Because we are analyzing the average matrix, each element $\bar{M}_{\mu\nu}$ only depends on the number of matching subpatterns between μ and ν so we can decompose the matrix in the following form:

$$\bar{M}_{xa,yb} = N_{RCN}[\gamma\delta_{xy}\delta_{ab} + \gamma_1(\delta_{xy} + \delta_{ab}) + \gamma_2 1], \quad (9)$$

where δ is the Kroenecker delta and γ, γ_1, and γ_2 are scalar coefficients to be determined. This equation simply states that there are three possible values for the entries of $\bar{M}_{\mu\nu}$, corresponding to whether μ and ν share zero, one, or two subpatterns. The right hand side of the equation is composed of three matrices that commute with each other, and hence we can study their eigenvalues separately. The matrices multiplied by γ_1 and γ_2 are both low rank and thus do not contribute to the minimal eigenvalue. Thus, the minimal eigenvalue is determined by the first matrix $N_{RCN}\gamma\delta_{xy}\delta_{ab}$, and using Equation 8, the value of κ is given by

$$\kappa \approx \sqrt{\frac{\gamma N_{RCN}}{p}}. \quad (10)$$

Using Equation 9, we can express γ in terms of the squared differences of activity due to different patterns:

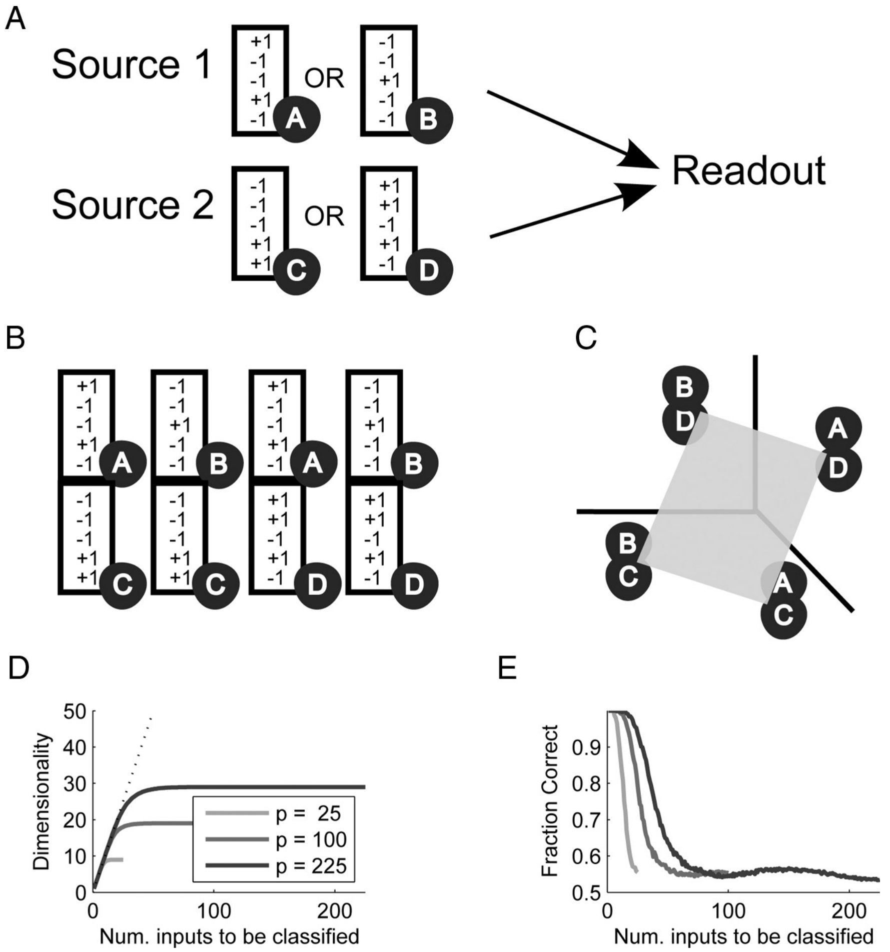

Figure 2. Segregated representations are not linearly separable. ***A***, Two sources of *N* neurons each are each in one of two configurations (A,B for the first source, C,D for the second one), and they are read out by a linear classifier. ***B***, The four possible input patterns in a 2*N* dimensional space. ***C***, Despite being 2*N* dimensional, the four patterns are actually on a 2D plane due to their structure. Four points on a 2D plane cannot be arbitrarily classified (e.g., AD and BC cannot be separated from AC and BD). The spatial arrangement of the four points is a consequence of the correlations between all the patterns (e.g., AC has a large overlap or correlation with AD, as the first source is in the same state). ***D***, For more than four patterns, the gap between the number of patterns and their corresponding dimension increases, impairing linear separability. The three curves show the number of dimensions versus the number (Num.) of inputs to be classified for two input sources with 5, 10, or 15 states each ($p = 25, 100,$ or 225 possible patterns). ***E***, Classification errors arise when there are more patterns than dimensions, even when only a subset of possible patterns is used. The graph shows that the fraction of correctly classified patterns drops rapidly once the number of patterns used exceeds the input dimensionality (compare to ***D***).

$$\frac{1}{2}\left\langle (\bar{S}_{xa} - \bar{S}_{yb})^2 \right\rangle = \frac{1}{2}\left\langle (\bar{M}_{xa,xa} - 2\bar{M}_{xa,yb} + \bar{M}_{yb,yb})^2 \right\rangle \tag{11}$$

$$= \gamma(1 - \delta_{xy}\delta_{ab}) + \gamma_1(2 - \delta_{xy} - \delta_{ab}) \tag{12}$$

$$\gamma = (\Delta_1)^2 - \frac{1}{2}(\Delta_2)^2, \tag{13}$$

where $(\Delta_1)^2$ and $(\Delta_2)^2$ are the squared differences between RCN activity due to two patterns differing by one and two subpatterns respectively (average across all RCNs). Equation 13 provides a recipe for estimating γ from experimental data. To compare this estimate with the true value of κ, we define $\Gamma = \frac{\kappa^2 p}{N_{RCN}}$ (Eq. 10 and Fig. 8*C*,*D*).

We now turn to the estimation of Σ^2. Because W is normalized, Σ^2 is simply a weighted average of the trial-to-trial variability of the RCNs. We approximate it by σ^2, which is the unweighted average (Fig. 8*E*,*F*). The final estimate of the test error is:

$$\mathrm{err}_{\mathrm{test}} \approx \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{\gamma}{\sigma^2}\frac{N_{RCN}}{p}}\right). \tag{14}$$

Note that a somewhat similar analysis of signal and noise using dimensionality of matrices was performed by Büsing et al. (2010).

A non-zero threshold of the readout does not change the scaling properties. Our analysis and simulations were performed assuming that the threshold of the readout unit is at zero. We verified that our numerical results do not depend on the choice of the threshold (data not shown). The reason for this can be understood by considering what happens to the matrix elements of M when an additional constant input implementing a non-zero threshold is added. In this case the modified matrix $\tilde{M}$ becomes:

$$\tilde{M}_{\mu\nu} = M_{\mu\nu} + 1, \tag{15}$$

thus adding a low rank matrix (all ones) that does not contribute to the rank. Hence, γ does not change, and neither does the performance.

Generalizing to more than two sources. An equivalent form to Equation 13 is

$$\gamma = M_0 - 2M_1 + M_2, \tag{16}$$

where M_0 is the average of the diagonal elements of the matrix $\bar{M}$ (of the form $\bar{M}_{xa,xa}$), M_1 is the average of those elements of M of the form $\bar{M}_{xa,xb}$, and M_2 is the average of those elements of M of the form $\bar{M}_{xa,yb}$. This form readily generalizes for more than two sources of information:

$$\gamma = \sum_{k=1}^{m} \binom{m}{k} (-1)^{k+1} M_k, \tag{17}$$

where M_k is the average value of all elements of the matrix $\bar{M}$ corresponding to the activity the RCNs when presented with two patterns differing by k subpatterns.

Simulations. All simulations were performed in Matlab (MathWorks). N was always chosen to be 500, and the rest of the parameters are noted in the main text. The readout weights were derived from the matrix $\bar{S}$ of average RCN activations. The entries of this matrix can be calculated by considering the mean and variance of the input $\tilde{g}_i$ to an RCN due to a noisy pattern:

$$\bar{\tilde{g}} = g_i(1 - 2n) \tag{18}$$

$$\mathrm{Var}\,(\tilde{g}_i) = 4n(1 - n), \tag{19}$$

where g_i is the noiseless input to that RCN. Approximating this input by a Gaussian distribution, we can derive the probability for this RCN to be activated as $q_i = \mathrm{erfc}\left(\frac{\theta - \bar{\tilde{g}}}{\sqrt{\mathrm{Var}(\tilde{g}_i)}}\right)$, and its mean state as $\bar{S}_i = 2q_i - 1$. Once we have the p patterns $\bar{S}^{\mu}$, we use quadratic programming to find the weight vector W that gives the maximal margin (Wills and Ninness, 2010).

Quality of approximation from experimentally accessible data. Because γ and σ^2 are only approximations (Fig. 8), we checked the quality of predicting the relative benefit of sparseness from experimentally accessible data. To this end, we measured for 20 realizations of 12 noise levels between 3% and 20% the ratio between the error obtained with a dense coding of $f = 0.5$ or an ultra-sparse coding of $f = 0.001$ to that obtained with a sparse coding of $f = 0.1$. We also estimated γ and σ^2 from 30 trials of each pattern using a subset of 100 RCNs (Fig. 7).

Results

The neural representation of information should be in a format that is accessible to the elements of a neural circuit. We took the point of view of an individual neuron reading out a population of input neurons that encode multiple noisy sources of information. In particular, we studied the number of input–output functions that can be robustly implemented by such a neuron. We first explain the problems arising when integrating multiple sources of information. We then show that the classification performance greatly increases when the input neurons mix the information sources by integrating the activity of the source populations through random connections (randomly connected neurons or RCNs). We show that the threshold of the RCNs, which determines their coding level (i.e., the average fraction of stimuli to which each individual neuron responds), biases a tradeoff between generalization and discrimination. Finally, we provide a prescription for measuring the components of this tradeoff from neural data.

The problem with segregated neural representations

Consider a single neuron receiving input from several sources. For ease of presentation and visualization, we consider only two sources. For example, one source may represent a sensory input and the other the internally represented task to be executed. Each source is segregated from the other and is represented by N neurons (Fig. 2*A*), each of which can be inactive (-1) or active (1). A state of one of the sources corresponds to a specific configuration of all N of its neurons.

In general, the classification capacity of a linear readout is determined by the structure (i.e., correlations) of the inputs and by the desired output. The desired output depends on the type of representations that will be needed by downstream processing stages in the brain. To remain general, we estimated the classification performance for all possible outputs. Specifically, we assume that the output neurons can only be either active or inactive in response to each input (two-way classification). If there are p different inputs, then there are 2^p input–output functions. The classification performance can be estimated by going over all these functions and counting how many can be implemented (i.e., when there is a set of synaptic weights that allow the output neuron to respond to all the inputs as specified by the function). As it is impractical to consider such a large number of input–output functions, we estimate the performance on randomly chosen outputs, which is a good approximation of the average performance over all possible outputs, provided the sample is large enough.

Under the assumption that we consider all possible outputs, the classification performance depends only on the properties of the input. In particular, the performance depends on the input correlations, which in our case are the correlations between the vectors representing the input patterns of activity. These correlations are due to the specific choice of the statistics of the inputs. A useful way to represent the correlations that are relevant for the performance is to consider the spatial arrangement of the points that represent the inputs in an activity space. Each input can be regarded as a point in an N_t dimensional space, where N_t is the total number of input neurons ($N_t = 2N$ in the example of Fig. 2). Our correlations are the consequence of a particular arrangement of the points representing the inputs. Indeed, in our case the points live in a low dimensional space (i.e., a space that has a dimensionality that is smaller than the minimum between N_t and p), and this can greatly limit the classification performance (Hinton, 1981; Barak and Rigotti, 2011). Figure 2*B* shows a simple example that illustrates the problem. The four possible configurations of the two populations of N input neurons are four points in a $2N$ dimensional space. Four points span at most a 3D space (i.e., a solid) and, more in general, p points span at most $p - 1$ dimensions (less if $N_t < p$). In our example, the four inputs are all on a 2D plane because of their correlations (Fig. 2*C*). One dimension is spanned by the line connecting the two patterns of the first source, and the other dimension goes along the line connecting the two patterns of the second source. The fact that there are more inputs to be classified than dimensions can lead to the existence of input–output functions that are not implementable by a linear readout. In other words, there will be sets of desired outputs that cannot be realized by a readout neuron. In these situations the inputs are said to be not linearly separable. For instance, it is not possible to draw a plane that separates patterns AD and BC from patterns BD and AC. This is equivalent to saying that there is no linear readout with a set of synaptic weights that implements an input–output function for which the inputs AD and BC should produce an output that is different from the one generated by inputs BD and AC (Hertz et al., 1991).

As the number of information sources and states within those sources increases, so does the gap between the number of patterns to be classified and the dimensionality of the space that they span, leading to a vanishing probability that the classification problem is linearly separable (see Materials and Methods) (Rigotti et al., 2010b). In Figure 2, *D* and *E* show this scaling for two sources of 5, 10, and 15 states each ($m = 5, 10, 15$). The number of neurons representing each source, $N = 500$, is significantly larger than the number of states. The dimensionality is more formally defined as the rank of the matrix that contains all the vectors that represent the $p = m^2$ different inputs (see Materials and Methods). Full rank (i.e., rank equal to the maximum, which in our case is p) indicates that all the p vectors representing the input patterns are linearly independent and hence span a p dimensional space. Because neurons within each source only encode that specific source, the dimensionality is always smaller than p (see Materials and Methods). Indeed, it scales as m, whereas p grows like m^2. This problem exists even when only a subset $\tilde{p} < p$ of all the m^2 combinations need to be correctly classified. This is shown in Figure 2*D*, where the dimensionality increases linearly with the number of inputs to be classified and then saturates. The upper bound determined by the source segregation is already reached at $\tilde{p} \sim m$, which is much smaller than the total number of m^2 combinations. Figure 2*E* shows that the probability for linear separability drops rapidly once the number of input patterns is higher than the dimensionality of the inputs.

Randomly connected neurons solve the problem

To solve the linear separability problem generated by the segregated representations, the information sources should be mixed in a nonlinear way. This can be achieved by introducing an intermediate layer of neurons that are randomly connected to the segregated inputs. These neurons increase the dimensionality of the neural representations (dimensional expansion), thereby increasing the probability that the problem becomes linearly separable for all possible outputs. Figures 3, *A* and *B* show that the dimensionality increases as more RCNs are added until it reaches the maximal dimensionality permitted by the number of inputs.

RCNs are surprisingly efficient at increasing the dimensionality. In the dense case in which the RCNs are activated by half of all

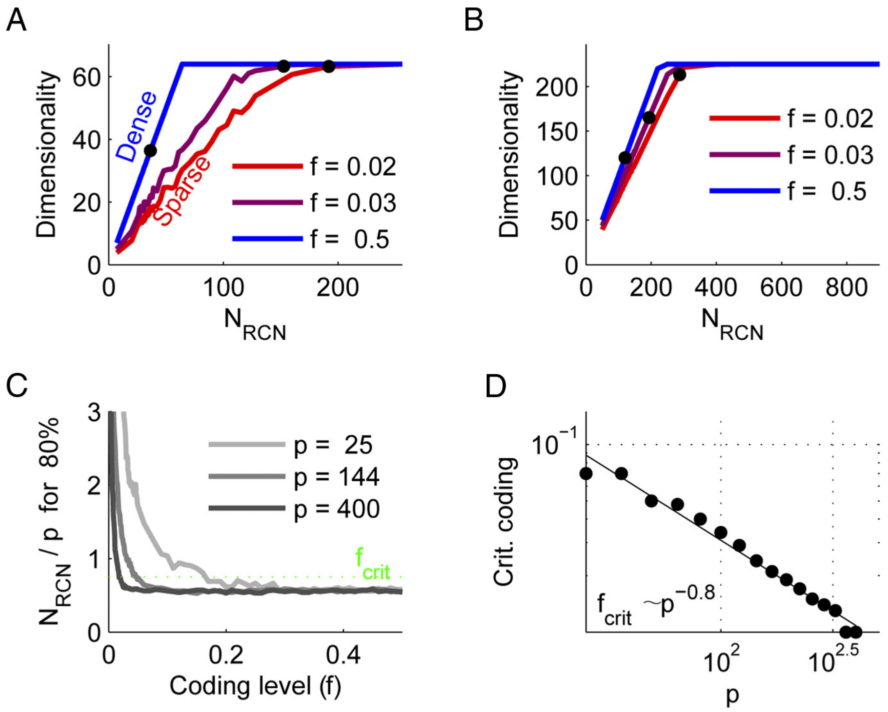

Figure 3. RCNs enable linear separability by increasing input dimensionality. ***A***, The dimensionality of the representation in RCN space as a function of the number of RCNs for 64 patterns (two sources of eight states each). For dense representations, every RCN increases the dimensionality by 1, while for sparse ones this slope is smaller. Correct classification requires a high enough dimensionality, and the black markers denote the point where 95% of the patterns could be classified correctly. Sparseness is measured by *f*, the fraction of patterns that activate a given RCN. ***B***, Similar to ***A***, but for 225 patterns. Note that the detrimental effect of sparse coding is reduced. ***C***, The ordinate denotes the minimum number of RCNs required to classify 80% of the patterns correctly normalized by the number of patterns considered. Note that as the number of patterns (and thereby RCNs) increases, sparser representations become more efficient. For each number of patterns there is a critical (Crit.) coding level (f_{crit}) below which performance deteriorates (green dotted line; see Materials and Methods). ***D***, This coding level scales as a power law of the number of patterns with an exponent of approximately −0.8.

possible inputs, if the number *p* of input patterns is sufficiently large, every RCN adds, on average, one dimension to the representations. This scaling is as good as the case in which the response properties of the neurons in the intermediate layer are carefully chosen using a learning algorithm. It is important to note that for simplicity we analyzed the case in which all combinations of input patterns are considered. In many realistic situations only a subset of those combinations may be needed to be classified to perform a given task. Although it is not possible to make statements about these general cases without further assumptions, we can note that if the combinations are picked uniformly at random from all the possible ones, the scaling of the number of dimensions versus the number of RCNs remains the same. In other words, the number of RCNs grows linearly with the number of inputs that have actually to be classified (Rigotti et al., 2010b). For this reason, in what follows we will consider only the case in which all the combinations have to be classified.

If one changes the threshold for activating the RCNs, and hence modifies the coding level *f* (i.e., the average fraction of the inputs that activate a RCN), the convergence to full dimensionality slows down. This is shown in Figure 3, where it is clear from the slope of the curves that the dimensionality increase per RCN is smaller for sparser neural representations. This is due to finite size effects—there are simply not enough RCNs to sample the entire input space. As the total number *p* of inputs increases and the space spanned by the inputs grows, the RCNs become progressively more efficient at increasing the dimensionality because they have more chances to be activated (Figure 3*B*).

The output neuron (Figs. 1*B* and 2*A*) can then be tested to determine how many input–output functions it can implement when the outputs are chosen randomly. In our situation it behaves like a perceptron that classifies uncorrelated inputs, although it is important to note that the inputs are not uncorrelated. The number of correctly classified inputs is approximately twice the number of RCNs. Figure 3*C* shows that this also holds for all but the sparsest coding levels, with the reason for failure being finite size effects of N_{RCN}. We can quantify the breakdown for sparse coding levels by defining a critical coding level, f_{crit}, at which the number of RCNs needed increases to 0.75 of the number of inputs. Figure 3*D* shows the scaling of this finite size effect; the coding level at which classification performance deteriorates scales as a power of the number of patterns: $f_{crit} \sim p^{-0.8}$.

RCN coding level biases the discrimination–generalization trade-off

In the previous section we analyzed the ability of the output neuron to classify inputs that contain multiple sources of information when the inputs are first transformed by RCNs. The next issue we address is whether the encouraging results on the scaling properties of the RCNs still hold when the output neuron is required to generalize. Generalization is the ability to respond in the same way to familiar and unfamiliar members of the same class of inputs. For example, in visual object recognition, the members of a class are the retinal images of all possible variations of the same object (e.g., when it is rotated), including those that have never been seen before. To study generalization it is important to know how to generate all members of a class. To make the analysis treatable, we studied a specific form of generalization in which the members of a class are noisy variations of the same pattern of activity. In our case, generalization is the ability to respond in the same way to multiple noisy variations of the inputs. Some of the noisy variations are used for training (training set), and some others for testing the generalization ability (testing set). The noise added to the patterns of activity is independent for each neuron, as in studies on generalization in attractor neural networks and pattern completion (see Discussion for more details).

We also make the further assumption that the number of RCNs is sufficient to reach the maximal dimensionality in the noiseless case, as we intend to focus on the generalization performance. We basically assume that there enough RCNs is to classify correctly the inputs in all possible ways (i.e., for all possible outputs) in the absence of noise.

As illustrated in Figure 1*D*, the transformation of the inputs performed by the RCN layer has to decorrelate them sufficiently to ensure linear separability while maintaining the representation of different versions of the same input similar enough to allow for generalization. The decorrelation increases the ability of the readout neurons to discriminate between similar inputs, but it is important to note that not all forms of discrimination lead to linear separability. The decorrelation operated by the RCNs has the peculiarity that it not only increases the dissimilarity between inputs, but it also makes the neural representations linearly separable.

We now study the features of the transformation performed by the RCNs and how the parameters of the transformation bias the discrimination–generalization tradeoff, with a particular emphasis on the RCN coding level *f*. *f* is the average fraction of RCNs that are activated in response to each input. *f* close to 0.5 means dense representations, small *f* corresponds to sparse representations. In our model, *f* is controlled by varying the threshold for the activation of the RCNs. Figure 4*B* shows how the relative Hamming distances between inputs are transformed by the randomly connected neurons for two different coding levels. These distances express the similarity between the neural representations (two identical inputs are at zero distance if they are identical). Note that the ranking of distances is preserved—if point A is closer to B than to C in the input space, the same will hold in the RCN space. In other words, if input A is more similar to B than to C, this relation will be preserved in the corresponding patterns of activity represented by the RCNs.

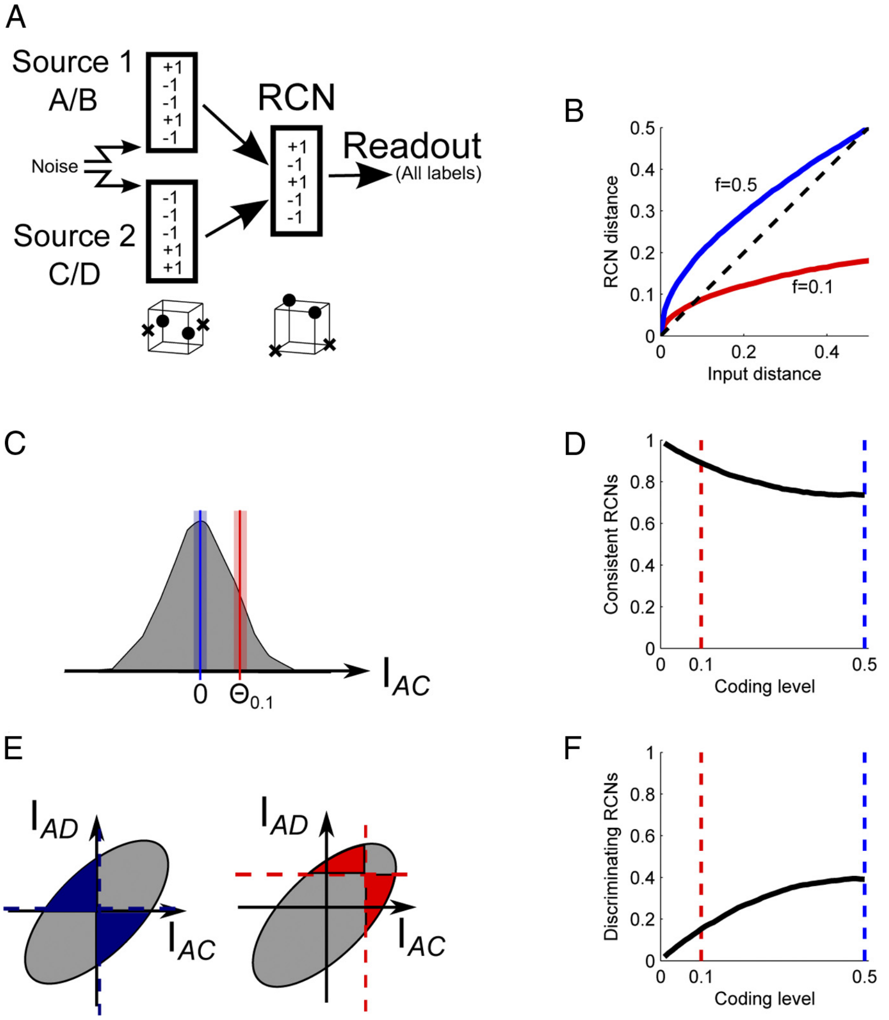

Figure 4. RCN coding level shifts the balance between discrimination and generalization. ***A***, Neural architecture, as in Figure 1*C*. The crosses and circles represent the patterns to be classified, as in Figure 1*D*. The original segregated representations are nonlinearly separable for a classification problem analogous to the exclusive OR (opposite points should produce the same output). The RCNs increase the dimensionality, making the problem linearly separable (now a plane can separate crosses from circles). ***B***, Transformation of Hamming distances in the RCN space for two coding levels (blue, 0.5; red, 0.1). The distance in the RCN space is plotted versus the distance in the input space. Although distances are distorted, their ranking is preserved (e.g., small distances map to small distances). ***C–F***, How generalization and discrimination abilities vary with the coding level of RCN representations. ***C***, The generalization ability is estimated as the fraction of RCNs that respond in a consistent manner to noisy realizations of the same input ($n = 0.1$). The shaded area represents the distribution of input currents to different RCNs for a particular input pattern (A, C, for the two sources, respectively). For dense representations (blue, threshold at zero) there is a larger fraction of RCNs that is around the activation threshold compared to the sparse case. ***D***, The fraction of consistent RCNs decreases with coding level. ***E***, The discrimination ability is estimated as the fraction of RCNs that respond differentially to a pair of patterns, differing only by the state of one source. The gray area represents the distribution of the currents to the RCNs for AC and AD inputs (differing in the second source). The area of colored shading represents the fraction of RCNs that respond differentially to the two combinations of inputs (for one input the current is positive and for the other it is negative). ***F***, Discrimination increases with coding level.

To understand how sparseness affects the classification performance, we first need to discuss the effects on both the discrimination and the generalization ability. Figure 4, *C-F* illustrate an intuitive argument explaining how sparseness biases the discrimination-generalization tradeoff (see Materials and Methods for details). We first consider generalization. Figure 4*C* shows the distribution of inputs to all RCNs when a generic input made of two sources of information is presented. For dense coding, the threshold is set to zero (blue line), and all the RCNs on the right of the threshold (half of all RCNs) are active. The noise in the input ($n = 0.1$ for this example) can cause those RCNs that receive near threshold input to flip their activity, as denoted by the blue shading. Similarly, for a sparse coding of $f = 0.1$, 10% of the RCNs are on the right of the red line, and a smaller number of RCNs are affected by noise. We estimate the generalization ability by measuring the fraction of RCNs that preserve their activity for different noisy versions of the same input pattern. This quantity increases as the representations become sparser (Fig. 4*D*).

As for the discrimination ability, we consider again the input currents to the RCNs. Figure 4*E* shows the two-dimensional input distribution to all RCNs for two inputs that share the same value for one of the two information sources (i.e., half of all input neurons are the same). As above, the blue and red lines denote threshold for $f = 0.5$ and $f = 0.1$, respectively. To enable discrimination, we need RCNs that respond differentially to these two patterns—their input is above threshold for one pattern and below threshold for the other, as denoted by colored areas. For this measure the fraction of RCNs with a differential response decreases as the representations become sparser (Fig. 4*F*).

The optimal coding level is approximately 0.1

To check how these two opposing trends affect the final performance of the classifier, we trained the output neuron to classify noisy versions of the inputs (obtained by flipping a random subset of *n* percent of the source neurons' activities), and then measured the fraction of wrong

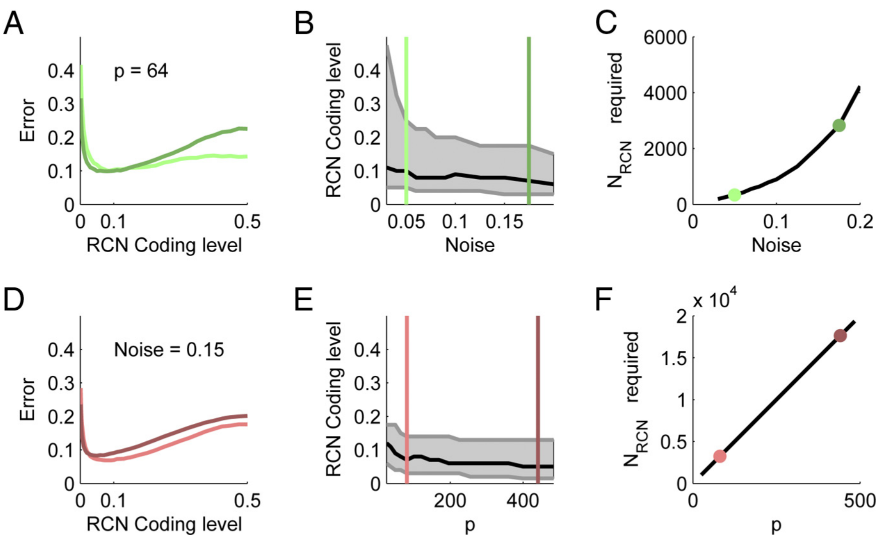

Figure 5. The optimal coding level is ~10%. ***A***, The dependence of classification error on RCN coding level for two different noise levels: $n = 0.05$ (light) and $n = 0.175$ (dark). Two sources of eight states each were used. ***B***, Extension of ***A*** for many noise levels, showing the dependence of the optimal coding level (black curve) on the input noise. The sensitivity of the error to coding level is indicated by shading those coding levels that result in an error up to 20% worse than the optimal one. The colored lines denote the values used in ***A***. ***C***, The number of RCNs required to maintain the minimal error at roughly 0.1 (i.e., 10% of the patterns misclassified). ***D***–***F***, Similar to the panels in the top row, but varying the number of patterns. Patterns were generated by two input sources with identical numbers of states. The two parameter values shown are $p = 81$ (light) and $p = 441$ (dark).

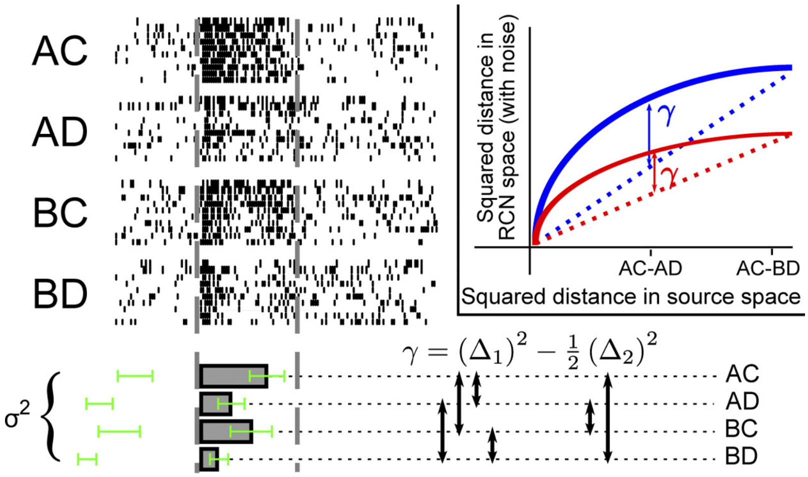
AC
AD
BC
BD
Squared distance in RCN space (with noise)
γ
γ
AC-AD
AC-BD
Squared distance in source space
γ = (Δ_1)^2 − ½ (Δ_2)^2
σ^2
AC
AD
BC
BD

Figure 6. Measuring the discrimination (γ) and generalization ($1/\sigma^2$) factors from neural data (simulated data). The rasters show the spikes of a single hypothetical neuron in response to several presentations of four combinations of stimuli (A/B) and contexts (C/D). The mean firing rates for each combination of input sources are calculated (bars below the rasters). The squared differences of firing rates are averaged for two cases: inputs differing by either stimulus or context, but not both (left cluster of arrows that point to the two dashed lines that correspond to these pairs, Δ_1), and inputs differing in both stimulus and context (right cluster, Δ_2). The green error bars denote the trial to trial variability used to compute σ^2. Inset, γ is a measure of the nonlinearity in the transformation to RCN space. In the input space, the difference between pairs of inputs belonging to Δ_1 is half the difference between pairs belonging to Δ_2. The factor γ measures the deviation from this relation in RCN space.

responses to new noisy realizations. Figure 5*A* shows numerical results of the fraction of errors made by a linear readout from a population of RCNs when the activity of a random 5% (light) or 17.5% (dark) of the source neurons is randomly flipped in each presentation. The abscissa shows *f*, the RCNs' coding level, which is varied by changing the activation threshold of the RCNs. The figure reveals that there is an optimal coding level of about 0.1 that decreases the error rate more than twofold compared to the maximally dense coding of 0.5. The advantage of sparser coding is more substantial as the input noise increases. Because we are interested in the shape of this curve, we increased the number of RCNs to avoid floor and ceiling effects as the noise increases (from 336 for 5% to 2824 for 17.5%). Otherwise, for small noise, the generalization error could be zero for all values of *f*, or it could be maximal and again constant for large noise.

Figure 5*B* shows the optimal coding level (black) for increasing levels of noise. The shaded area around the black curves is delimited by the coding levels at which the performance decreases by 20% compared to the black curve. The optimal coding level hardly shifts, but the relative advantage of being at the minimum increases when noise increases. The number of RCNs necessary to compensate for the increased noise is shown in Figure 5*C*.

In the noiseless case (Fig. 3) we saw that the sparse coding levels are adversely affected by finite size effects. To ascertain whether this is the cause for the increase in error as the coding level decreases below $f = 0.1$, we increased the number of patterns and RCNs. Figure 5*D* shows that a fivefold increase in the number of patterns only slightly moves the optimal coding level. Indeed, Figure 5*E* shows that the optimal level does decrease with increasing system size, but at a very slow rate that is probably not relevant for realistic connectivities. Indeed, even for 20,000 RCNs (Fig. 5*F*) the optimal coding level is still above 5%. Note that when we vary the number of patterns, the required number of RCNs grows linearly (Fig. 5*F*).

Components of the discrimination–generalization tradeoff

The numerical results reveal several phenomena. First, the required number of RCNs grows linearly with the number of input patterns. Second, a coding level of approximately $f = 0.1$ is better than dense coding ($f = 0.5$) for correct classification. Third, an ultra sparse coding level of $f = 0.01$–0.03 is significantly worse than intermediate values. We derived an approximate analytical expression of the test error that allows us to understand the scaling properties of the RCN transformation and relies on experimentally accessible factors (see Materials and Methods):

$$\mathrm{err}_{\mathrm{test}} \approx \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{\gamma(\theta, n)}{\sigma^2(\theta, n, p)}\frac{N_{RCN}}{p}}\right), \tag{20}$$

where γ is the discrimination factor that depends on the threshold θ for activating the RCNs (and hence on the coding level of the RCN representations) and on the noise n in the inputs. $1/\sigma^2$ is the generalization factor, which depends on θ, n, and the total number p of classes of inputs. The inverse of the generalization factor, σ^2, is simply defined as the average intertrial variability of RCN responses (Fig. 6, green error bars).

The discrimination factor γ is related to the similarities between the RCN representations induced by similar inputs. Figure 6 defines γ more precisely and shows how to measure it from neural data. Consider, for example, the case in which one source of information represents a sensory stimulus and the other represents the temporal context in which the stimulus appears (as in Fig. 1). We assume that the recorded neurons contain a representative subpopulation of the RCNs, which presumably are the majority of neurons. We also assume that the recorded RCNs receive an equal contribution from the inputs representing the two sources. For each neuron we consider the mean firing rate for every combination of the inputs. For simplicity, we assume that there are two possible stimuli and two contexts for a total of four cases. The four bars corresponding to the four rasters in Figure 6 represent the mean firing rates in these cases. We now focus on pairs of inputs that differ only by the state of one source (e.g., as in the pair of cases in which the same sensory stimulus appears in two different contexts). For each such pair, we compute the squared difference in the firing rate of the neuron. This quantity should be averaged across all conditions that contain analogous pairs of inputs. We name this average $(\Delta_1)^2$. In a similar manner, $(\Delta_2)^2$ is the average squared difference between the firing rates corresponding to the cases in which both sources are different (e.g., different sensory stimuli appearing in different contexts, right cluster of arrows). The discrimination factor γ is then given by the average across neurons of:

$$\gamma = (\Delta_1)^2 - \frac{1}{2}(\Delta_2)^2, \tag{21}$$

This quantity can be computed from the recorded activity under the assumption that the two sources of information have an equal weight in driving the RCNs. This is a reasonable assumption every time the two sources of information can be considered similar enough for symmetry reasons (e.g., when they represent two visual stimuli that in general have the same statistics). In the other cases it is possible to derive an expression for γ that takes into account the different weights of the two sources. However, the relative weights should be estimated from the data in an independent way (e.g., by recording in the areas that provide the RCNs with the input).

To help understand the meaning of γ in the case that we analyzed (i.e., when the two sources have the same weight), we show in the inset of Figure 6 how γ is related to the shape of the curve that represents the squared distance in the RCN space as a function of the squared distance in the input space. In particular, γ expresses the deviation from a linear transformation. Notice that in contrast to Figure 4*B*, on the *y*-axis we now represent the expected squared distance in RCN space between pairs of noisy patterns. The distances in RCN space are contracted by the presence of noise in the inputs (see Materials and Methods, Eq. 18).

The deviation from a linear function is intuitively related to the ability to discriminate between patterns that are not linearly separable. Indeed, for a linear transformation ($\gamma = 0$) the dimensionality of the original input space does not increase, and the neural representations would remain nonlinearly separable.

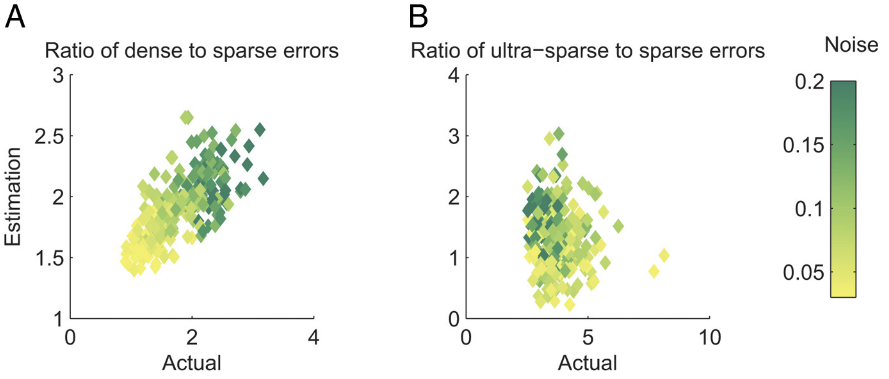

Figure 7. Estimating error from experimentally accessible factors versus the actual error. The figure shows the ratio between the error obtained using either dense (***A***, $f = 0.5$) or ultra-sparse (***B***, $f = 0.01$) to that obtained using sparse ($f = 0.1$) coding. Ratios are shown for various levels of the input noise. The *x*-axis shows the ratio derived from the full simulation, while the *y*-axis is computed using the formula and calculating γ and σ^2 from 30 trials of 100 RCNs. The color bar shows the noise level (same values as in Fig. 5*A*–*C*, using 64 patterns).

While the exact values of the error are not captured by the experimentally accessible factors, the general trends are. To illustrate this point, we computed the expected error from a subset of 100 RCNs simulated during 30 trials of 64 patterns. Figure 7*A* shows the ratio of the test error for dense (0.5) and sparse (0.1) coding as derived from this estimation versus the actual one obtained from the full simulation. Note that the correlation is very good (correlation coefficient, 0.7), even though for the high noise levels the network contains >4000 RCNs. This result is especially important in cases where N_{RCN} and p are unknown but fixed—for instance, when a neuromodulator changes the activity level of the network. In such cases, estimating γ and σ^2 from neural recordings can provide a useful measure of the effect on network performance. The case of ultra-sparse coding is not captured as well by the approximation (Fig. 7*B*; correlation coefficient, −0.1), and the reason for this is explained below.

Equation 20 already confirms the first phenomenon mentioned above—linear scaling of RCN number with the number of input patterns. If we ignore the weak p dependence of σ^2 and rearrange the terms of the equation, we can see this linear scaling. As indicated by Figure 5*F*, the dependence of σ^2 on p is small and does not affect the scaling. We also verified that, similarly to the noiseless case (Fig. 2*D*,*E*), using a fraction of possible input combinations does not alter this scaling (data not shown).

To understand the remaining phenomena—namely why a coding level of ~0.1 is optimal—we consider the interplay between the discrimination and generalization factors. Figure 8, *A* and *B* show the dependence of test error as a function of the coding level of the RCN representation for two different noise levels. We computed these quantities either using the full numerical simulation (solid line) or the approximation of Equation 20 (dashed line). While the approximation captures the general trend of the error, it underestimates the error for low coding levels. This is more evident in the low noise case, which also requires fewer RCNs.

The reason for the failure of the formula in the ultra sparse case is that γ and σ^2 are actually approximations of Γ and Σ^2—quantities that are not directly accessible experimentally (see Materials and Methods). Briefly, Γ is a measure of the average distance from a pattern to the decision hyperplane in RCN space. Σ^2 is the average noise of the patterns in the

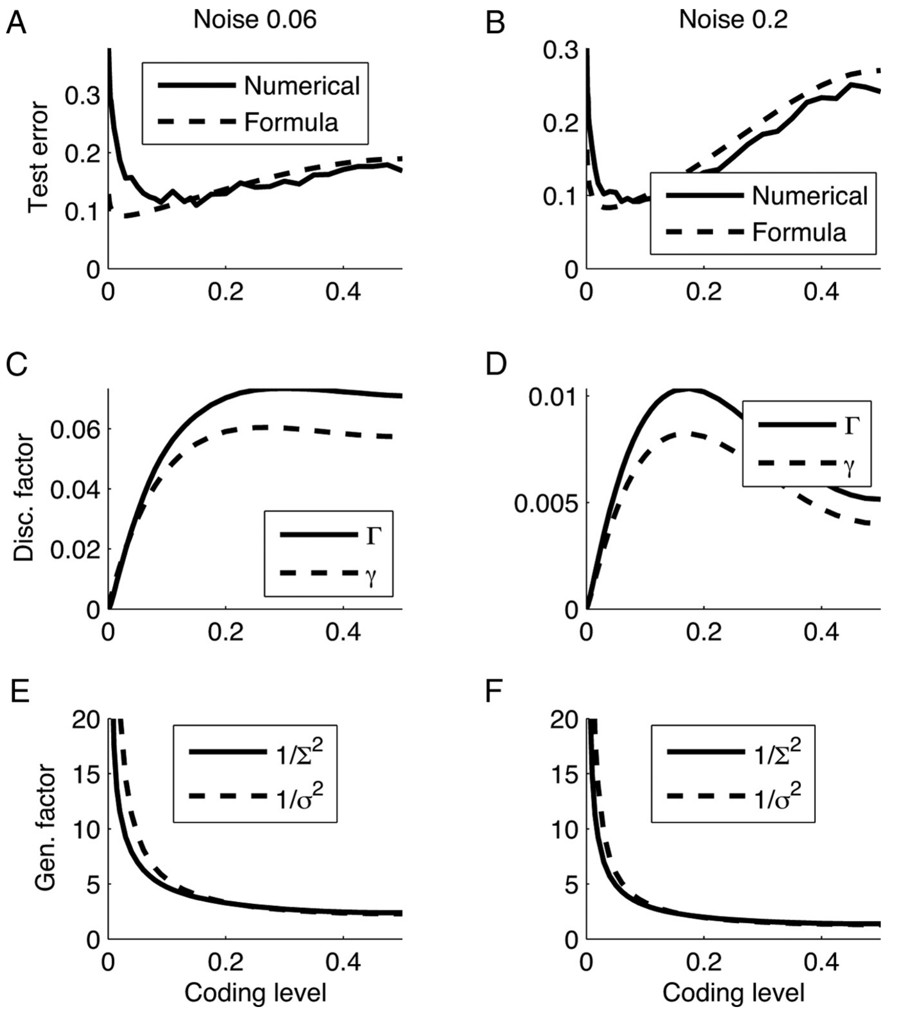

Figure 8. Components of generalization discrimination tradeoff. ***A***, ***B***, Comparing the actual test error with the one predicted by Equation 20. Note the discrepancy in sparse coding levels, which is caused mainly by the small denominator of Equation 20. ***C***, ***D***, Discrimination factor, exact Γ and approximate γ, as a function of coding level. ***E***, ***F***, Generalization factor, exact $1/\Sigma^2$ and approximate $1/\sigma^2$, as a function of coding level. All values were derived using 64 input patterns and 424 (4227) RCNs for 0.06 (0.2) noise.

direction perpendicular to the decision hyperplane. Using these quantities in the formula produces a curve that is almost indistinguishable from the one obtained with the full simulation (data not shown). We thus look at the deviations of the factors from their exact counterparts.

Figure 8, *C* and *D*, show that the discrimination factor in general increases with coding level, but for high levels of noise it shows a nonmonotonic behavior. The generalization factor shown in Figure 8, *E* and *F*, increases with coding level, giving rise to the discrimination generalization trade-off. Note that both of these estimates follow the general trend of their more exact counterparts—Γ and Σ^2—but have some systematic deviations. Specifically, σ^2 underestimates the noise for sparse coding levels, leading to the discrepancy in Figures 8, *A* and *B*, and 7*B*.

To summarize the reason for optimality of 0.1 coding—it is better than 0.5 because dense coding amplifies noise more than it aids discrimination. This is more evident in the cases with high input noise. A coding level lower than 0.05 is suboptimal due to finite size effects, which are probably inevitable for biologically plausible parameter values. These trends are summarized in Figure 9. Increasing inhibition or changing the balance between excitation and inhibition will cause the neural representations to become sparser, decreasing the ability to discriminate between similar inputs. This performance degradation, however, is overcompensated by the increased robustness to noise (generalization), with the net effect that the generalization error decreases for sparser representations. This trend does not hold for too sparse representations ($f < 0.1$), because finite size effects start playing the dominant role, and the generalization ability does not improve fast enough to compensate for the degradation in the discrimination ability. In this regime any increase in global inhibition leads to a degradation in the performance.

Discussion

Most cognitive functions require the integration of multiple sources of information. These sources may be within or across sensory modalities (e.g., distinct features of a visual stimulus are different sources of information), and they may include information that is internally represented by the brain (e.g., the current goal, the rule in effect or the context in which a task is performed). In all these situations, what are the most efficient neural representations of the sources of information? The answer depends on the readout. We focused on a simple linear readout, which is what presumably can be implemented by individual neurons. We showed that the sources must be mixed in a nonlinear way to implement a large number of input–output functions. Segregated representations composed of highly specialized neurons that encode the different sources of information independently are highly inefficient, because the points representing the possible inputs span a low dimensional space due to their correlation structure (as in the case of semantic memories; Hinton, 1981).

Segregated representations can be transformed into efficient representations with a single layer of randomly connected neurons. This transformation can efficiently increase the dimensionality of the neural representations without compromising the ability to generalize. The best performance (minimal classification error) is achieved for a coding level $f \sim 0.1$, as the result of a particular balance between discrimination and generalization.

Why a linear readout?

Our results hinge on the choice of a linear readout that limits classification ability. We proposed RCNs as a possible solution, which is compatible with the observation that neurons with nonlinear mixed selectivity are widely observed in the brain (Asaad et al., 1998; Rigotti et al., 2010a; Warden and Miller, 2010). One may legitimately wonder whether there are other biologically plausible solutions involving different forms of nonlinearities. For example, it is possible that neurons harness the nonlinear dendritic integration of synaptic inputs so that a full or a partial network of RCNs is implemented in an individual dendritic tree.

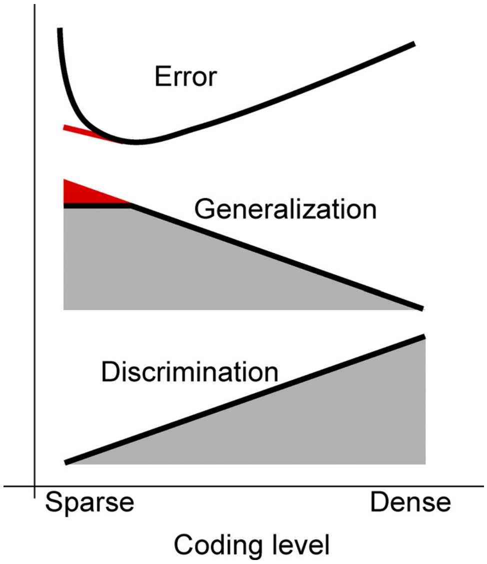

Figure 9. Summarizing our understanding of the generalization discrimination trade-off. Dense coding is optimal for discriminating between similar inputs but has a detrimental effect on the ability to generalize. Decreasing coding level shifts this balance and improves the overall classification ability, but for very sparse coding levels ($f < 0.1$) finite size effects limit the generalization ability, and thus the classification error increases. In the limit of very large neural systems (red lines), the classification error would keep increasing as the representations become sparser, and this would compensate for the decrease in the ability to discriminate.

In this scenario, some of the "units" with nonlinear mixed selectivity, analogous to our RCNs, are implemented by a specific branch or set of dendritic branches, and hence they would not be visible to extracellular recordings. Some others must be implemented at the level of the soma and expressed by the recordable firing rates, as mixed selectivity is observed in extracellular recordings. Our results about the statistical properties of the RCNs apply to both hidden and visible units if they implement similar nonlinearities. As a consequence, our predictions about the activity of the RCNs are likely to remain unchanged in the presence of dendritic nonlinearities. However, future studies may reveal that the dendritic nonlinearities play an important role in strongly reducing the number of RCNs.

Our choice of a linear readout is also motivated by recent studies on a wide class of neural network dynamical models. Inspired by the results on the effects of the dimensional expansion performed in support vector machines (Cortes and Vapnik, 1995), many researchers realized that recurrent networks with randomly connected neurons can generate surprisingly rich dynamics and perform complex tasks even when the readout is just linear (Jaeger, 2001; Maass et al., 2002; Buonomano and Maass, 2009; Sussillo and Abbott, 2009; Rigotti et al., 2010b). Studying RCNs in a feedforward setting has enabled us to derive analytical expressions for the scaling properties of the circuit. Our results probably have important implications for the dynamics of models that rely on RCNs to expand the dimensionality of the input, even outside the feedforward realm. In particular, our analysis can already predict the relevant dynamical properties of a recurrent network model implementing rule-based behavior (Rigotti et al., 2010b).

Why randomly connected neurons?

RCNs solve efficiently the problem of low dimensionality of the input by mixing nonlinearly multiple sources of information. Intermediate layers of mixed selectivity neurons can be obtained in many other ways (Rumelhart et al., 1986). However, RCNs offer an alternative that is appealing for several reasons. First, there is accumulating experimental evidence that for some neural systems random connectivity is an important representational and computational substrate (Stettler and Axel, 2009). Second, the number of needed RCNs (that are not trained) scales linearly with the number of inputs that should be classified. This is the same scaling as in the case in which the synaptic weights are carefully chosen with an efficient algorithm (Rigotti et al., 2010b). Third, many of the algorithms for determining the weights of hidden neurons require random initial conditions. The importance of this component of the algorithms is often underestimated (Schmidhuber and Hochreiter, 1996). Indeed, there are many situations in which learning improves the signal-to-noise ratio but does not change the statistics of the response properties that the neurons had before learning, which are probably due to the initial random connectivity. This consideration does not decrease the importance of learning, as it is clear that in many situations it is important to increase the signal-to-noise ratio (see e.g., the spiking networks with plastic inhibitory-to-excitatory connections analyzed by Bourjaily and Miller, 2011). However, it indicates that our study could be relevant also in the case in which the neurons of the hidden layer are highly plastic.

In all the above mentioned cases learning can improve the performance. There are situations, as those studied in recurrent networks (Bourjaily and Miller, 2011), in which different forms of synaptic plasticity can lead either to beneficial or disruptive effects. Synaptic plasticity between inhibitory and excitatory neurons increases the signal-to-noise ratio, as mentioned above, but STDP (spike timing-dependent plasticity) between excitatory neurons actually disrupts the diversity of the neural responses, requiring a larger number of RCNs. These forms of learning, which are disruptive for the heterogeneity, are probably present in the brain to solve other problems in which it is important to link together neurons that fire together. Classical examples are the formation of invariant representations of visual objects (DiCarlo et al., 2012) or learning of temporal context (Rigotti et al., 2010a).

How general are our results?

When multiple sources of information are represented in segregated neuronal populations, the correlations in the inputs can limit the number of input–output functions that are implementable by a linear readout. We showed that RCNs can mix these sources of information efficiently and solve the nonseparability problems related to this type of correlations. The correlations that we considered are presumably widespread in the brain, as they are likely to emerge every time a neuron integrates two or more sources of information, as in the case in which it receives external and recurrent inputs. For example, neurons in layer 2/3 of the cortex receive a sensory input from layer 4 and a recurrent input from other neurons in the same layer (Feldmeyer, 2012). It is important in any case to notice that the correlations and the noise that we studied are specific and that there are important computational problems which involve different types of correlations and more complex forms of generalization. For example, the classification of visual objects is a different difficult problem because the retinal representations of the variations of the same object can be more different than the representations of different objects. The manifolds (sets of points in the high dimensional space of neural activities) representing the variations of specific objects are highly curvilinear (with many twists and turns) and "tangled," requiring the neural classifiers to implement a large number of variations (Bengio and LeCun, 2007; DiCarlo et al., 2012). The shallow neural architecture that we considered (only one intermediate layer) can deal only with "smooth" classes (i.e., a small variation of the input should not require a change in the response of the output neuron that classifies the inputs). Indeed for non-smooth classes, a prohibitive number of RCNs and a huge training set would be required. To classify visual objects efficiently, one would require a sophisticated preprocessing stage that extracts the relevant features from the retinal input so that the neural representations of the visual object become "smooth." Deep networks (Bengio and Le Cun, 2007) that contain multiple layers of processing can be efficiently trained to solve these problems.

It is difficult to say whether our results about the efficiency of RCNs and optimal sparseness apply also to problems like vision, and further studies will be required to make more general statements. We speculate that our results probably apply to the late stages of visual processing and to some of the components of the early stages (e.g., when multiple features should be combined together). Interestingly, some of the procedures used to extract features in deep networks can generate neural architectures that are similar to those obtained with random connectivity. Networks with random weights and no learning can already represent features well suited to object recognition tasks when the neurons are wired to preserve the topology of the inputs (i.e., neurons with limited receptive fields) (Saxe et al., 2011). These semi-structured patterns of connectivity could also be an important substrate for learning the features used in deep networks.

Why sparse representations?

One of our main results is that there is an optimal sparseness for the neural representations. The coding level that minimizes the generalization error is, for most realistic situations, close to $f = 0.1$.

Besides our results and the obvious and important argument related to metabolic costs, there are other computational reasons for preferring a high degree of sparseness. These reasons lead to different estimates of the optimal f, typically to lower values than what we determined.

The first reason is related to memory capacity: sparse neural representations can strongly reduce the interference between stored memories (Willshaw et al., 1969; Tsodyks and Feigel'man, 1988; Amit and Fusi, 1994). The number of retrievable memories can be as large as f^{-2} when the proper learning rule is chosen. When f goes to zero, the capacity can become arbitrarily large, but the amount of information stored per memory decreases. If one imposes that the amount of information per memory remains finite in the limit $N \rightarrow \infty$, where N is the number of neurons in a fully connected recurrent network, then the number of random and uncorrelated patterns that can be stored scales as $N^2/(\log N)^2$ when $f = \log N/N$. f is significantly smaller than our estimate when one replaces N with the number of connections per neuron (in the cortex $N \sim 10^4$ would lead to $f \sim 10^{-3}$). The discrepancy becomes larger when one considers wider brain areas (Ben Dayan Rubin and Fusi, 2007).

A second reason mentioned in the Introduction is the ability of sparse over-complete representations to increase input dimensionality, facilitate learning, and reduce noise (Olshausen and Field, 2004).

All of these computational reasons lead to different estimates of the optimal f, as they deal with different problems. The brain is probably dealing with all these problems, and for this reason it may use different and sometimes adaptive coding levels in different areas, but also within the same area (indeed, there is a lot of variability in f across different neurons).

Estimates of the sparseness of neural representations recorded in the brain vary over a wide range, depending on the method for defining the coding level, the sensory stimuli used, and the brain area considered. Many estimates are close to our optimal value of 0.1, especially in some cortical areas (e.g., in V4 and IT it ranges between 0.1 and 0.3) (Sato et al., 2004; Rolls and Tovee, 1995; J. J. DiCarlo and N. Rust, unpublished observations).

The hippocampus exhibits significantly lower coding level (0.01–0.04) (Barnes et al., 1990; Jung and McNaughton, 1993; Quiroga et al., 2005). These estimates are lower bounds for f, as the authors used very strict criteria to define a cell as responsive to a stimulus. For example, in Quiroga et al. (2005) a cell was considered to be selective to a particular stimulus if the response was at least five standard deviations above the baseline. On the other hand, many of these estimates are probably biased by the technique used to record neural activity (extracellular recording). Active cells tend to be selected for recording more often than quiet cells, shifting the estimate of f toward higher values (Shoham et al., 2006). Recent experiments (Rust and DiCarlo, 2012), designed to accurately estimate f, indicate that for V4 and IT $f \sim 0.1$.

Biasing the generalization–discrimination tradeoff

The generalization–discrimination tradeoff resulted in an optimal coding level of 0.1 under general assumptions about the statistics of the inputs and the outputs of individual neurons. Specific behavioral tasks may impose additional constraints on these statistics, resulting in different optimal coding levels. This is probably why the brain is endowed with several mechanisms for rapidly and reversibly modifying the sparseness of the neural representations (e.g., by means of neuromodulation; Disney et al. (2007); Hasselmo and McGaughy (2004)). In other situations, neural systems that become dysfunctional (due e.g., to stress, aging or sensory deprivation) may cause a long term disruptive imbalance in the discrimination–generalization trade-off.

These types of shifts have been studied systematically in experiments aimed at understanding the role of the dentate gyrus (DG) and CA3 in pattern separation and pattern completion (Sahay et al., 2011). The DG has been proposed to be involved in pattern separation, which is defined as the process by which similar inputs are transformed into more separable (dissimilar) inputs. It is analogous to our definition of pattern discrimination, suggesting that the neurons in the DG may have similar properties as our RCNs. CA3 seems to play an important role in pattern completion, which is the reconstruction of complete stored representations from partial inputs. Neurons in CA3 would be represented by the output neurons of our theoretical framework, and pattern completion is related to the generalization ability. Neurogenesis, which is observed in the DG, may alter in many ways (e.g., by changing the global level of inhibition in the DG) the balance between pattern separation and pattern completion (Sahay et al., 2011). In the future, it will be interesting to analyze specific tasks and determine to what extent our simple model can explain the observed consequences of the shifts in the balance between pattern separation and pattern completion.

References

Amit DJ, Fusi S (1994) Learning in neural networks with material synapses. Neural Comput 6:957–982. CrossRef Medline

Asaad WF, Rainer G, Miller EK (1998) Neural activity in the primate prefrontal cortex during associative learning. Neuron 21:1399–1407. CrossRef Medline

Atick JJ, AN Redlich (1992) What does the retina know about natural scenes? Neural Comput 4:196–210. CrossRef

Barak O, Rigotti M (2011) A simple derivation of a bound on the perceptron margin using singular value decomposition. Neural Comput 23:1935–1943. CrossRef

Barnes CA, McNaughton BL, Mizumori SJ, Leonard BW, Lin LH (1990) Comparison of spatial and temporal characteristics of neuronal activity in sequential stages of hippocampal processing. Prog Brain Res 83:287–300. CrossRef Medline

Ben Dayan Rubin DD, Fusi S (2007) Long memory lifetimes require complex synapses and limited sparseness. Front Comput Neurosci 1:7. CrossRef Medline

Bengio Y, LeCun Y (2007) Scaling learning algorithms towards AI. In: Large-scale kernel machines, pp 321–360. Cambridge, MA: MIT.

Bourjaily MA, Miller P (2011) Synaptic plasticity and connectivity requirements to produce stimulus-pair specific responses in recurrent networks of spiking neurons. PLoS Comput Biol 7:e1001091. CrossRef Medline

Buonomano DV, Maass W (2009) State-dependent computations: spatiotemporal processing in cortical networks. Nat Rev Neurosci 10:113–125. CrossRef Medline

Büsing L, Schrauwen B, Legenstein R (2010) Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. Neural Comput 22:1272–1311. CrossRef Medline

Cortes C, Vapnik V (1995) Support-vector networks. Machine Learning 20: 273–297. CrossRef

Cover TM (1965) Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans Electronic Comput 14:326–334. CrossRef

DiCarlo JJ, Zoccolan D, Rust NC (2012) How does the brain solve visual object recognition? Neuron 73:415–434. CrossRef Medline

Disney AA, Aoki C, Hawken MJ (2007) Gain modulation by nicotine in macaque v1. Neuron 56:701–713. CrossRef Medline

Feldmeyer, D (2012) Excitatory neuronal connectivity in the barrel cortex. Front Neuroanat 6:24. CrossRef Medline

Hasselmo ME, McGaughy J (2004) High acetylcholine levels set circuit dynamics for attention and encoding and low acetylcholine levels set dynamics for consolidation. Prog Brain Res 145:207–231. CrossRef Medline

Hertz J, Krogh A, Palmer RG (1991) Introduction to the theory of neural computation, Vol 1. Reading, MA: Addison-Wesley.

Hinton GE (1981) Implementing semantic networks in parallel hardware. In: Parallel models of associative memory (Hinton GE, Anderson JA), pp 161–187. Hillsdale, NJ: Lawrence Erlbaum.

Hinton GE, Anderson JA (1989) Parallel models of associative memory. Hillsdale, NJ: Lawrence Erlbaum.

Jaeger H (2001) The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report 148.

Jazayeri M, Movshon JA (2006) Optimal representation of sensory information by neural populations. Nat Neurosci 9:690–696. CrossRef Medline

Jung MW, McNaughton BL (1993) Spatial selectivity of unit activity in the hippocampal granular layer. Hippocampus 3:165–182. CrossRef Medline

Krauth W, Mézard M, Nadal JP (1988) Basins of attraction in a perceptron-like neural network. Complex Sys 2:387–408.

Lukoševičius M, Jaeger H (2009) Reservoir computing approaches to recurrent neural network training. Comp Sci Rev 3:127–149. CrossRef

Maass W, Natschlger T, Markram H (2002) Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput 14:2531–2560. CrossRef Medline

Marr D (1969) A theory of cerebellar cortex. J Physiol 202:437–470. Medline

McClelland JL, Rumelhart DE (1985) Distributed memory and the representation of general and specific information. J Exp Psychol Gen 114: 159–197. CrossRef Medline

Miller EK, Cohen JD (2001) An integrative theory of prefrontal cortex function. Annu Rev Neurosci 24:167–202. CrossRef Medline

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

Olshausen BA, Field DJ (2004) Sparse coding of sensory inputs. Curr Opin Neurobiol 14:481–487. CrossRef Medline

Quiroga RQ, Reddy L, Kreiman G, Koch C, Fried I (2005) Invariant visual representation by single neurons in the human brain. Nature 435:1102–1107. CrossRef Medline

Rigotti M, Ben Dayan Rubin D, Morrison SE, Salzman CD, Fusi S (2010a) Attractor concretion as a mechanism for the formation of context representations. Neuroimage 52:833–847. CrossRef Medline

Rigotti M, Ben Dayan Rubin D, Wang XJ, Fusi S (2010b) Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. Front Comput Neurosci 4:24. CrossRef Medline

Rolls ET, Tovee MJ (1995) Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. J Neurophysiol 73:713726. Medline

Rumelhart DE, Hintont GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536. CrossRef

Rust NC, DiCarlo JJ (2012) Balanced increases in selectivity and tolerance produce constant sparseness along the ventral visual stream. J Neurosci 32:10170–10182. CrossRef Medline

Sahay A, Wilson DA, Hen R (2011) Pattern separation: a common function for new neurons in hippocampus and olfactory bulb. Neuron 70:582–588. CrossRef Medline

Sato T, Uchida G, Tanifuji M (2004) The nature of neuronal clustering in inferotemporal cortex of macaque monkey revealed by optical imaging and extracellular recording. Soc Neurosci Abstr 34:300–312.

Saxe A, Koh P, Chen Z, Bhand M, Suresh B, Ng A (2011) On random weights and unsupervised feature learning. In: Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, June-July.

Schmidhuber J, Hochreiter S (1996) Guessing can outperform many long time lag algorithms. Technical note: IDSIA-19-96.

Shoham S, O'Connor DH, Segev R (2006) How silent is the brain: is there a "dark matter" problem in neuroscience? J Comp Physiol A Neuroethol Sens Neural Behav Physiol 192:777–784. CrossRef Medline

Stettler DD, Axel R (2009) Representations of odor in the piriform cortex. Neuron 63:854–864. CrossRef Medline

Sussillo D, Abbott LF (2009) Generating coherent patterns of activity from chaotic neural networks. Neuron 63:544–557. CrossRef Medline

Tsodyks MV, Feigel'man MV (1988) The enhanced storage capacity in neural networks with low activity level. Europhys Lett 6:101–105. CrossRef

Warden MR, Miller EK (2010) Task-dependent changes in short-term memory in the prefrontal cortex. J Neurosci 30:15801–15810. CrossRef Medline

Wills AG, Ninness B (2010) QPC—quadratic programming in c. http://sigpromu.org/quadprog/.

Willshaw DJ, Buneman OP, Longuet-Higgins HC (1969) Non-holographic associative memory. Nature 222:960–962. CrossRef Medline

Xue H, Chen S, Yang Q (2011) Structural regularized support vector machine: a framework for structural large margin classifier. IEEE Trans Neural Netw 22:573–587. CrossRef Medline