Neuron
Article

Ultrastructural Analysis of Hippocampal Neuropil from the Connectomics Perspective

Yuriy Mishchenko,[1,4] Tao Hu,[1,4] Josef Spacek,[3] John Mendenhall,[2] Kristen M. Harris,[2,*] and Dmitri B. Chklovskii[1,*]
[1]Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA 20147, USA
[2]Center for Learning and Memory, Neurobiology, University of Texas, Austin, TX 78705, USA
[3]The Fingerland Department of Pathology, Charles University Hospital, 500 05 Hradec Kralove, Czech Republic
[4]These authors contributed equally to this work
*Correspondence: kharris@mail.clm.utexas.edu (K.M.H.), mitya@janelia.hhmi.org (D.B.C.)
DOI 10.1016/j.neuron.2010.08.014

SUMMARY

Complete reconstructions of vertebrate neuronal circuits on the synaptic level require new approaches. Here, serial section transmission electron microscopy was automated to densely reconstruct four volumes, totaling 670 μm^3, from the rat hippocampus as proving grounds to determine when axo-dendritic proximities predict synapses. First, in contrast with Peters' rule, the density of axons within reach of dendritic spines did not predict synaptic density along dendrites because the fraction of axons making synapses was variable. Second, an axo-dendritic touch did not predict a synapse; nevertheless, the density of synapses along a hippocampal dendrite appeared to be a universal fraction, 0.2, of the density of touches. Finally, the largest touch between an axonal bouton and spine indicated the site of actual synapses with about 80% precision but would miss about half of all synapses. Thus, it will be difficult to predict synaptic connectivity using data sets missing ultrastructural details that distinguish between axo-dendritic touches and bona fide synapses.

INTRODUCTION

Reconstructing neuronal circuits on the level of synapses is a central problem in neuroscience. Smaller invertebrate circuits can be reconstructed using serial section transmission electron microscopy (ssTEM) by identifying synapses and manually tracing pre- and postsynaptic neuronal processes to their cell bodies as has been demonstrated for the *C. elegans* nervous system (White et al., 1986; Chen et al., 2006). However, manually reconstructing vertebrate circuits using ssTEM is impractical and it remains unclear which technology will be capable of achieving this goal. Although automating ssTEM seems promising (Jurrus et al., 2008; Anderson et al., 2009; Mishchenko, 2009), the proof of principle is missing. At the same time, older approaches to reconstruct neuronal circuits are being used (Binzegger et al., 2004; Stepanyants and Chklovskii, 2005; Stepanyants et al., 2008) and alternative approaches are being developed (Briggman and Denk, 2006; Smith, 2007; Helmstaedter et al., 2008; Luo et al., 2008).

In this paper, we used manual (*RECONSTRUCT*; Fiala and Harris, 2001a, 2002; Fiala, 2005) and automated (Mishchenko, 2009) ssTEM reconstruction techniques to reconstruct densely four volumes of rat hippocampus neuropil. Although the reconstructed volumes are too small to contain complete circuits, they demonstrate that ssTEM can be scaled through automation. In addition, we used the reconstructed volumes as proving grounds to determine whether other approaches based on proximities between axons and dendrites can yield reliable predictions of synaptic connectivity.

Perhaps the oldest method for inferring synaptic connectivity used light microscopy and relies on counting proximities between axons and dendrites that can be bridged by a spine, or so-called potential synapses (Peters and Feldman, 1976; Braitenberg and Schuz, 1998; Stepanyants and Chklovskii, 2005). As the ratio of actual to potential synapses, which we call the connectivity fraction, is much less than one (Stepanyants et al., 2002), such a method can predict connectivity only probabilistically. The number of actual synapses, for example, along a dendrite is given by the number of potential synapses times the connectivity fraction. For this method to have practical value, the connectivity fraction must be invariant among dendrites, an assumption known as "Peters' rule" (Peters and Feldman, 1976; Braitenberg and Schuz, 1998). By relying on this assumption synaptic connectivity has been estimated in various neuronal circuits (Binzegger et al., 2004; Stepanyants and Chklovskii, 2005; Jefferis et al., 2007; Stepanyants et al., 2008),

The validity of Peters' rule has been explored both anatomically using sparse reconstructions (White and Rock, 1981; White, 2002; da Costa and Martin, 2009) and electrophysiologically using stimulation of neuronal classes (Shepherd et al., 2005; Petreanu et al., 2009). These studies revealed two kinds of Peters' rule violations: different classes of pre-synaptic neurons possess different connectivity fractions onto a given postsynaptic neuron class and different post-synaptic neuron classes have different connectivity fractions with a given presynaptic neuron class. Such violations indicate connection specificity among neuronal classes. However, the validity of Peters' rule within an apparently homogeneous class of neurons could not be tested because it required dense reconstructions.

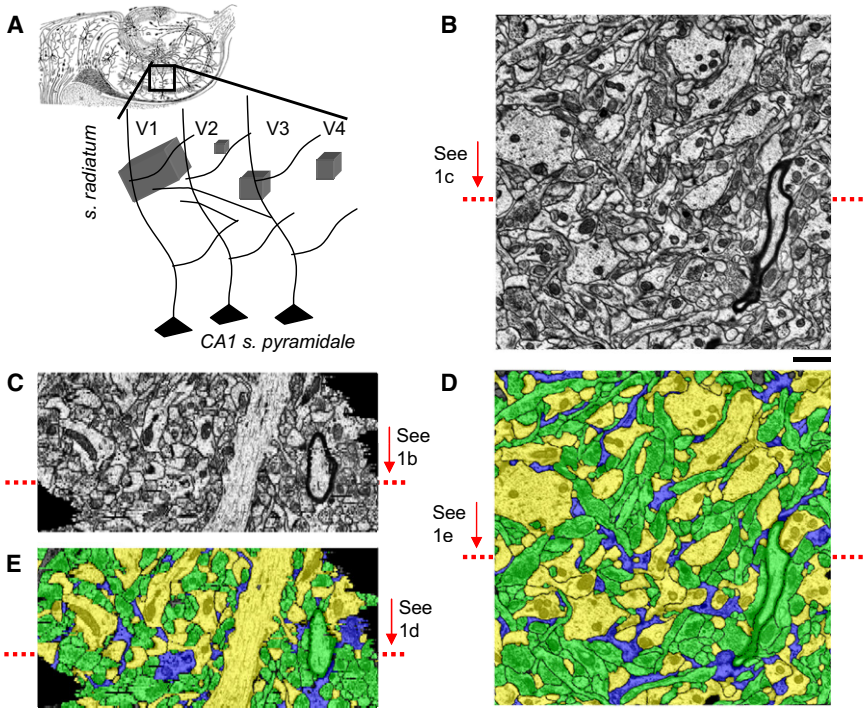


Figure 1. Reconstructed Volumes

(A) Location of the four volumes (V1–4) relative to CA1 pyramidal neuron dendrites in the hippocampus.

(B) Typical ssTEM micrograph of the hippocampus neuropil from V1.

(C) V1 resectioned orthogonal to the cutting plane at the location indicated by the red arrow in (B). Note that the stack is well aligned and the ultrastructure is visible despite lower z resolution.

(D) Electron micrograph from b after automated segmentation and proofreading colored according to the object class: axons, green; dendrites, yellow; and glia processes, blue.

(E) Segmented resection from (C).

Scale bar: 1 μm (B–E). See also Figure S1.

Among alternative approaches, serial block-face scanning electron microscopy (SBFSEM) (Denk and Horstmann, 2004) may benefit from knowing the relationship between proximities and synapses. To outline processes this technique requires high-contrast labeling, which emphasizes the extracellular space, while failing to visualize intra-cellular structures, such as synaptic vesicles and postsynaptic densities that are required for synapse identification. Hence, having a way to identify synapses based on the shape of axons and dendrites and their geometrical arrangement, such as touching, might strengthen the appeal of this and similar approaches for circuit reconstruction.

In reconstructed volumes, we identified all axons, boutons, dendrites, dendritic spines, postsynaptic densities (PSDs) and glial process, and measured the distributions of the dimensions of identified objects. The knowledge of dimensions helped to formulate quantitatively new methods to infer synaptic connectivity. We demonstrate that several formulations of Peters' rule fail to predict the density of synapses along dendrites because the probability of potential synapses being actual synapses varies among dendrites. We propose two novel methods to predict the density of synapses along dendrites using the density of touches and dendritic shaft caliber. Because the density of synapses is a small fraction (<20%) of the density of touches, the question arises whether touches can predict individual synapses without using synaptic attributes available only in ssTEM. To answer this question, we attempted to predict synapses from touches using their dimensions and found that relative areas of contact among boutons and spines can identify synapses with reasonably high probability approximating 80 percent, although many synapses are missed. The results will also help to evaluate other methods for inferring synaptic connectivity.

RESULTS

Reconstruction of Neuropil Volumes

We photographed through ssTEM four volumes of neuropil from the middle of stratum radiatum in hippocampal area CA1 at a spatial resolution of 2.2 nm/pixel and section thicknesses of 45–50 nm (Figures 1A and 1B). Volumes 1–3 (V1–3) came from a mature and volume 4 (V4) an immature postnatal day 21 rat (see Experimental Procedures). V1 centered on a radial oblique dendrite; V2 centered on a dendritic spine; V3 centered on an apical dendrite, and V4 was randomly located in s. radiatum (Figure 1A; and Table 1).

We partitioned, or segmented, these volumes along plasma membranes into three-dimensional objects using both automated and manual approaches. In the automated approach, the computer performed alignment (Figures 1B and 1C) and segmentation (Figures 1D and 1E; Mishchenko, 2009). Then a proofreading facility visually guided the user through serial sections of each object to verify or correct the segmentation. The segmentation was complete meaning that each pixel was attributed to a unique object or to a boundary between objects. In addition, we manually segmented sub-regions of V1–3 into three-dimensional objects using the RECONSTRUCT software (Fiala and Harris, 2001a, 2002; Fiala, 2005) (http://synapses.clm.utexas.edu); which allowed us to estimate the accuracy and times savings of the automated approach (see below).

We classified reconstructed three-dimensional objects into axons, dendrites, and glial processes (Figures 1D, 1E, and 2A) using the following characteristic features (Peters et al., 1991; Harris, 2008). Axons consisted of thin processes interspersed with boutons containing synaptic vesicles. Dendrites received synapses, both asymmetric (excitatory), recognized by thickened postsynaptic densities (PSDs), and symmetric (inhibitory), recognized by pleomorphic vesicles and uniform thinner densities on pre- and postsynaptic sides. Spiny dendrites were further sub-divided into shafts and spines connected to their dendritic shafts through necks and receiving only asymmetric synapses. Small astroglial processes interdigitated irregularly among axons and dendrites, and contained glycogen granules.

Table 1. Sample Volumes and Numbers of Unique 3D Objects in Each

Name	Manual	Automated	All 3D Objects	Axons	Dendrite & Spine Fragments	Glia Fragments	Unidentified Objects # (% Volume)
V1 ("Oblique")	42 μm^3	9.1 × 9.0 × 4.1 = 336 μm^3	1496	629	66 & 112	151	538 (3.9%)
V2 ("Spine")	7 μm^3	5.4 × 3.8 × 1.7 = 35 μm^3	524	345	21 & 80	35	43 (0.7%)
V3 ("Apical")	167 μm^3	6.1 × 6.1 × 4.5 = 167 μm^3	597	445	33 & 118	57	0
V4 (PN21)	NA	6.0 × 4.3 × 5.1 = 132 μm^3	548	256	29 & 75	56	132 (3.6%)
Total	219 μm^3	670 μm^3	3165	1675	149 & 385	243	713 (2.7%)

Because the above characteristic features were absent from many sections in any single ssTEM image, the three-dimensional nature of the reconstruction was essential for object identification. As automated reconstructions, with the exception of V3, extended through the whole image volume, some (<4% by volume) objects that grazed the volume edges did not contain enough features for unequivocal identification (Table 1). Since the manually reconstructed volumes did not extend to the edges of the image volumes, all of the objects could be identified unambiguously by viewing them as they passed beyond the boundaries (see Figure S1 available online).

Currently, volume reconstruction is limited by time. Tracing is the most time consuming step for the manual process and proofreading is the most time consuming step in the automated process. In both approaches, experience reduced the time needed to follow and identify small objects through sections, establish correct connections, and complete the reconstructions. Proofreading V1 (336 μm^3) by an expert (Harris) took approximately 90 hr. Proofreading V3 (167 μm^3) by an experienced electron microscopist (Mendenhall), who had limited experience in hippocampal neuropil, took approximately 160 hr: 80 hr to learn and 80 hr to complete. Hence, the average time required to proofread and correct the automatically segmented objects in V1 and V3 was about 10–20 min per μm^3. Relative to manual tracing, we estimate a tenfold savings in time.

Validation of Automated Reconstructions

We validated the automated reconstructions by comparing them against manually reconstructed sub-volumes as well as repeat viewing by more than one proofreader. In addition, we measured object dimensions and the partitioning of volume and plasma membrane area among object classes. Comparison of these values among volumes and with those previously reported provided additional validation and confirmed that the volumes were representative of hippocampal neuropil in general.

Reconstruction errors affecting the topology of the circuit, which we name "content errors," typically occurred when objects with dimensions equal to or less than section thickness overlapped and ran tangentially. Thus, the distribution of content errors is nonuniform among different classes of objects (Table 2). No content errors occurred in the reconstruction of thick dendritic shafts. A few spines were lost when their necks were obliquely sectioned and some thin axons were overlapping at some places along their lengths and may have been accidentally merged (Table 2). For example, in V3, this resulted in 26 content errors per 346 spines for an error rate of 0.08 errors per spine; and 45/447 = 0.1 per axon (0.022 errors per micron of axon).

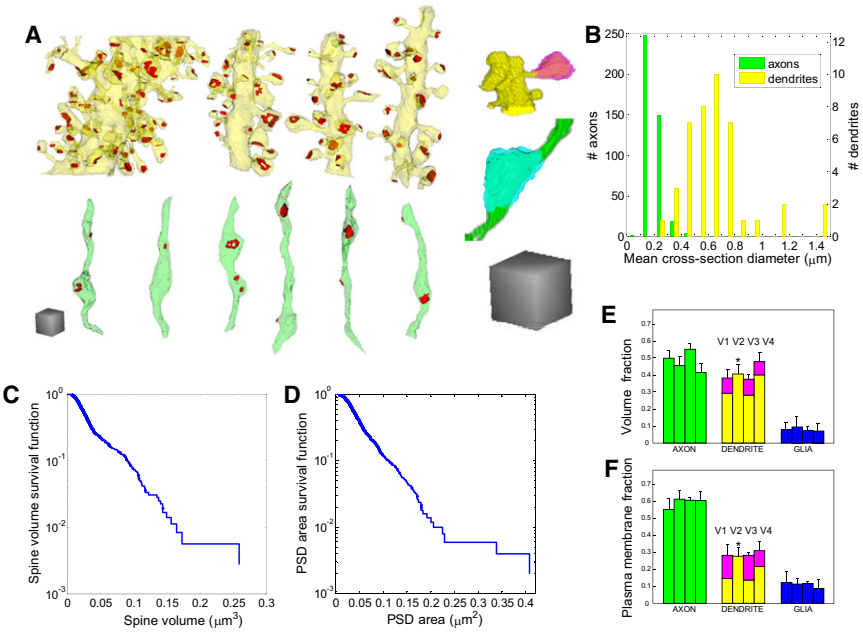


Figure 2. Shapes and Dimensions of Various Objects in the Neuropil

(A) Three-dimensional reconstruction of representative objects in V3: dendrites (yellow), axons (green), postsynaptic densities (PSDs) (red), spine (pink), and bouton (cyan).

(B) Distribution of the effective axonal and dendritic cross-section diameters in V1 and V3.

(C) Survival function of spine volume, i.e., a fraction of spines whose volume is greater than a given value.

(D) Survival function of the PSD area. Only spines and PSDs completely contained within V1 were included in (C) and (D).

(E) Distribution of volume among different object classes in the four volumes.

(F) Distribution of plasma membrane surface area among different object classes.

Scale cubes in (A) are 1 μm on the side; bars in (E) and (F) are arranged sequentially V1→V4 in each object class (axon, dendrite, glia); * in (E) and (F), calculations of the volume of spine heads and other analysis were not performed for V2 given its small size.

Table 2. Statistics of Potential Content Errors in Automated and Manual Reconstructions

	Number of Potential Content Errors					
	Dendrites	Spines		Axons		
Compared volumes	Auto	Man	Auto	Man	Auto	# of Content Errors/# of Contours = Percentage Auto
V1 42 μm^3	none	3	6	8	9	15/7,500 = 0.2%
V2 7 μm^3	none	none	7	none	none	7/1,700 = 0.4%
V3 167 μm^3	none	11	26	4	45	71/23,500 = 0.3%

We further characterized the content error rate by computing the relative fraction of all contours that had these potential errors (Table 2).

Astroglial processes could often be traced to larger processes with characteristic bundles of intermediate filaments (Ventura and Harris, 1999). Sometimes the glial processes could not be linked unambiquously to one another; nevertheless, they likely belonged to one or at most a few astrocytes. This conclusion is based on the observation that astrocytes span regions larger than reconstructed volumes and tile neuropil without substantial overlap between neighboring astrocytes (Bushong et al., 2002; Livet et al., 2007).

Discrepancies in the 3D shapes of corresponding processes reconstructed by the manual or automated approaches resulted in volume differences of less than 10% (Figure S2). The mean deviation between the same contours produced by the manual and automated approaches was 5 nm. Since the automated approach was performed with the images down sampled to a resolution of 4.4 nm/pixel, this value corresponds to a mean deviation of about one pixel. The observed volume difference is consistent with 5 nm variations in the placing of boundary contours along small processes that are 100–200 nm in diameter, which is typical for axons, the most common object in these volumes.

We found that axons, dendrites, and synapses vary widely in their dimensions both within and among classes. Axons ranged in effective cross-section from 0.10 to 0.50 μm, while dendrites ranged from 0.28 to 1.49 μm (Figure 2B; see Experimental Procedures for the algorithm used to compute the effective cross-section). Spine volumes ranged from 0.003 to 0.26 μm^3 (Figure 2C), and PSD areas ranged from 0.01 to 0.41μm^2 (Figure 2D). The breadth of these distributions suggests that the mean values (Table 3) carry only limited information about object dimensions. These distributions motivated the synapse prediction methods described below.

To verify that our sample volumes were representative of the general neuropil we computed the fraction of neuropil volume that was occupied by various classes of objects (Figure 2E). We found that axons occupied about 50% and dendrites occupied about 40% of the volume. In the immature neuropil, V4, dendrites occupied a significantly larger fraction of the volume at the expense of axons ($p < 0.01$). Glial processes occupied about 8% of all four volumes (Figure 2E). In volumes 1, 3, and 4, we distinguished spines from dendritic shafts and discovered they occupied about 9% of the total volume.

The relative distribution of plasma membrane surface area among neuropil components (Figure 2F) differed from the volume distributions (Figure 2E), which was not surprising given the differences in dimensions. Nearly 60% of the plasma membrane surface area belonged to axons, a value greater than their corresponding volume fraction consistent with their smaller caliber (Figure 2B). Likewise, the thin and tortuous glial processes provided 10%–13% of all plasma membrane surface area (Figure 2F), which was much greater than their corresponding volume fraction (Figure 2E). Dendritic shaft surface area was about 15%–20% of total membrane area (Figure 2F), substantially less than its corresponding volume fraction (Figure 2E) but also consistent with their larger caliber (Figure 2B). Spines occupied about 10%–12% of the total plasma membrane surface area (Figure 2F).

Overall, the consistency of these results among the adult samples and the general agreement with previous reports (Harris and Stevens, 1989; Lisman and Harris, 1993; Schikorski and Stevens, 1997; Sorra and Harris, 2000; Chklovskii et al., 2002) suggests that the chosen volumes are reasonable representatives of dense hippocampal neuropil.

Peters' Rule Does Not Accurately Predict Synapse Density

A long-standing proposition for estimating synaptic connectivity, known as Peters' rule, states that the number of synapses formed along a dendrite is proportional to the number of axons passing within reach of the spines emanating from the dendrite (Peters and Feldman, 1976; Braitenberg and Schuz, 1998), called potential synapses (Stepanyants et al., 2002). The coefficient of proportionality was called "filling fraction" (Stepanyants et al., 2002) and is hereby renamed to "connectivity fraction" to avoid confusion

Table 3. Summary of Measured Neuropil Parameters

Axon diameter (V1 & V3)	0.20 ± 0.06 μm
Dendrite diameter (V1 & V3)	0.67 ± 0.26 μm
Mean PSD area (V1)	0.054 μm^2
Exponential decay constant of PSD area (V1)	0.047 μm^2
Mean spine head volume (V1)	0.038 μm^3
Exponential decay constant of spine head volume (V1)	0.037 μm^3
Number of axons touching dendritic shaft per μm of dendritic length (V1 & V3)	6 ± 2
Number of axons per μm^2 volume cross-section (V1 & V3)	7
Number of axons crossing a cylinder 1 μm from dendritic shaft surface per length of dendrite (V1 & V3)	22 ± 6
Volume density of synapses (V1, V3, & V4)	2.2 ± 0.5 μm^3

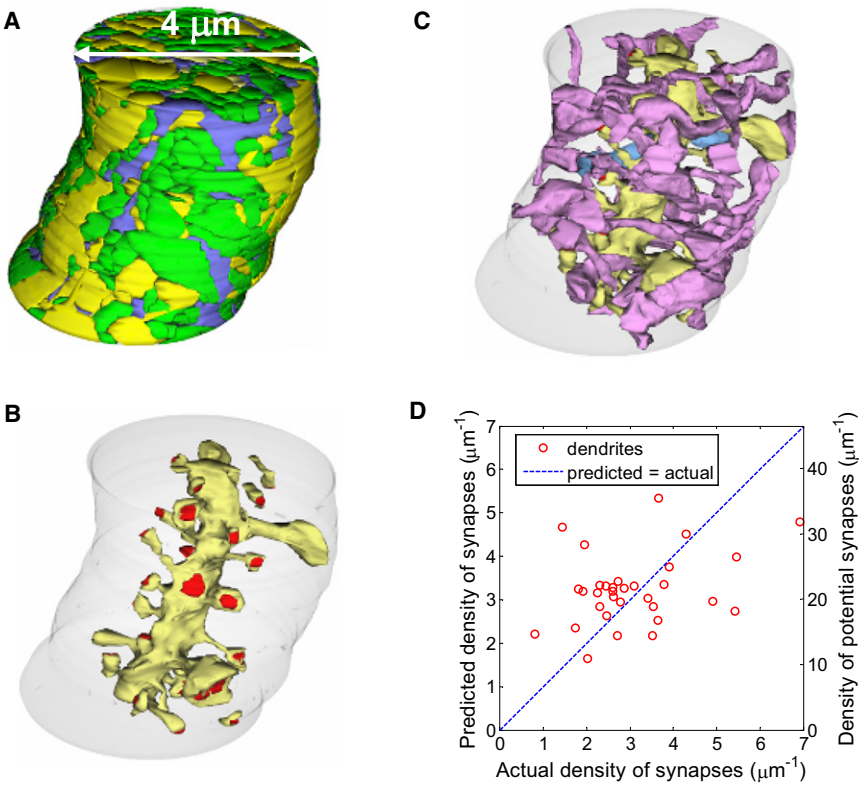


Figure 3. Comparison of Actual Density of Synapses along Individual Dendrites in V1 and V3 and Predictions Based on Maximum Reach Connectivity Fraction

(A) Manual reconstruction of cylinder centered on the central oblique dendrite coursing through V1 and containing axons (green), dendrites (yellow), and glia (blue). Double arrowed line indicates the diameter of the cylinder.

(B) Central oblique dendrite (yellow) and its associated synapses (red) located on dendritic spines. The boundary of the smallest neuropil cylinder that contained the selected oblique dendrite and all of its spines is illustrated in light gray.

(C) Subpopulation of axons (purple, to distinguish from all green axons in A) that formed synapses with the central oblique dendrite (yellow). Of these 28 axons, 27 made just one synapse and 1 made 2 synapses (light blue axon) on this dendrite.

(D) Plot of the actual density of synapses for dendrites in V1 and V3 versus the density of synapses predicted by multiplying the mean maximum-reach connectivity fraction by the local density of potential synapses. This method is a weak predictor ($r^2 \approx 0.12$).

See also Figure S3.

with the volume fractions discussed above. The maximum-reach connectivity fraction was defined as

$$\text{maximum reach connectivity fraction} = \frac{\#\text{ of axons presynaptic to reference dendrite}}{\#\text{ of all axons}(d_{spine\ reach})}, \quad (1)$$

where $d_{spine\ reach}$ is the length of the dendrite's longest spine .

We directly measured the maximum-reach connectivity fractions for the oblique and apical dendrites centered in the manually reconstructed V1 and V3, respectively (Figures 3A–3C) and compared them with the theoretical prediction (Stepanyants et al., 2002). "Maximum-reach potential connectivity cylinders" were empirically constructed around each dendritic segment at a diameter containing the longest spine. For the central oblique dendrite in V1, the cylinder encompassing the longest spine was 4 um in diameter and 3.45 μm long. Of the 159 axons that entered this cylinder, 102 made synapses within the cylinder, but only 28 of these made synapses with the central oblique dendrite for a connectivity fraction of 0.18. For the apical dendrite in V3, the cylinder encompassing the longest spine was 6 μm in diameter and 3.64 μm long. Of the 256 axons that entered that cylinder, 159 made synapses within, but only 54 made synapses with the central dendrite for a connectivity fraction of 0.19. Connectivity fractions for these oblique and apical dendrites are close to each other and to the predicted value of 0.22 (Stepanyants et al., 2002).

Next, we compared directly measured synaptic densities along dendritic segments in the automated volumes with those estimated using several formulations of Peters' rule. We considered synaptic density along dendrites rather than the number of synapses, to eliminate the dependence of number on the length of the dendritic segments. First, we calculated the average connectivity fraction by dividing the total number of synapses among all dendrites by the total number of axons within 1 μm from the surface of each dendrite's shaft, Equation 1. Second, we obtained the predicted density of synapses along each dendrite (per μm of dendrite) by multiplying the mean connectivity fraction and the density of axons (per μm of dendrite) within 1 μm of each dendrite:

$$(\text{predicted density of synapses}) = (\text{density of axons near dendrite}) * (\text{mean connectivity fraction}).$$

Multiplying the local density of axons by the mean maximum-reach connectivity fraction predicted the density of synapses along dendrites rather poorly, Figure 3D. To determine whether the discrepancy could have arisen by chance due to the small numbers of synapses on individual dendritic segments, we calculated the probability of finding this or a greater discrepancy assuming that synapses were drawn with a uniform probability that was set by the connectivity fraction, see Experimental Procedures. The probability was $p < 0.05$, suggesting that the discrepancy was unlikely to have occurred by chance; hence, the connectivity fraction varied among different dendrites (Figure S3A). Therefore, we can reject Peters' rule using the maximum-reach connectivity function as a tool to predict synaptic densities.

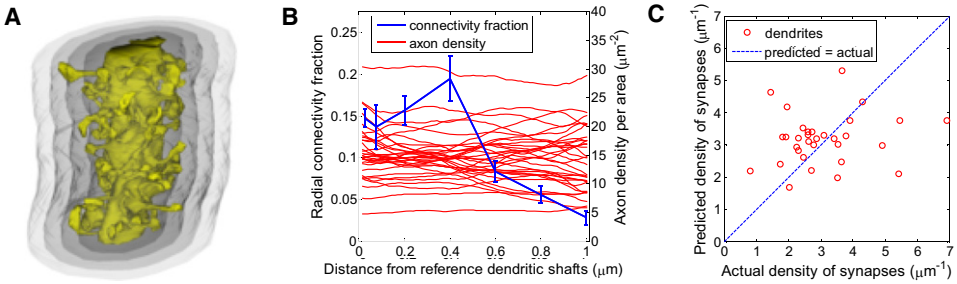


Figure 4. Comparison of Actual Density of Synapses along Individual Dendrites in V1 and V3 and Predictions Based on the Distance-Dependent Connectivity Fraction

(A) 3D illustration of one dendritic segment and four radial shells, each following the surface outline of the dendritic shaft after the spines had been truncated.
(B) Dependence of the connectivity fraction (mean ± SD) and axonal density on the distance from the surface of the dendritic shaft.
(C) Plot of the actual density of synapses along dendrites in V1 and V3 versus the density of synapses predicted by convolving the mean distance-dependent connectivity fraction (blue line in B) with the local axon density (red lines in B). This method is a weak predictor ($r^2 \approx 0.02$).
See also Figure S4.

Could the failure of this prediction be due to an over-simplification of the connectivity fraction as being constant up to the maximum spine reach and then dropping to zero? In reality, the connectivity fraction was a smooth function peaking at a distance around 0.4 um from the dendritic shaft (Figures 4A and 4B). This distance may seem small compared to a typical spine length, yet is consistent with spine length measurements because spines are not necessarily straight, and also do not necessarily synapse at the axon's nearest point (Harris and Stevens, 1989).

We hypothesized that using this distance-dependent connectivity fraction might improve the prediction accuracy of Peters' rule. Such an approach had an additional benefit because it did not require estimating the maximum spine reach, which fluctuated greatly along dendrites because long spines occur infrequently. Nevertheless, even with the distance-dependent connectivity fraction, Peters' rule poorly predicted the actual synaptic densities (Figure 4C). To determine whether this discrepancy could have arisen by chance due to the small numbers of synapses on each individual dendrite, we performed a statistical test similar to above, see Experimental Procedures. The probability was $p < 0.05$, (data not shown) further confirming that the connectivity fraction varies among and along dendrites. Thus, we can also reject Peters' rule with distance-dependent connectivity fraction as a tool to predict synaptic densities.

Axo-Dendritic Touches and Dendritic Caliber Are Good Predictors of Synapse Density

In this section, we report two approaches that predict the density of synapses on a dendrite more reliably than Peters' rule. First, we considered dendrite caliber as a predictor of synaptic density (Figures 5A). As the shape of dendritic cross-section can be irregular, we quantified the caliber by its circumference length (with spines truncated). Then, synaptic density is proportional to the circumference length. We found that the remaining discrepancy can happen by chance ($p > 0.5$; Figure S3B). Thus, the hypothesis that the synaptic density is linked to dendritic caliber cannot be rejected. Note that the dendrite caliber is not correlated with the density of potential synapses (Figure 5B), suggesting that axon availability is not the source of the caliber-synapse density correlation.

Second, we considered the number of axons touching dendritic spines as a predictor of the number of synapses. We defined a touch as proximity between an axon and a dendritic spine with no other intervening objects. We calculated the density of synapses on a dendrite as a function of the density

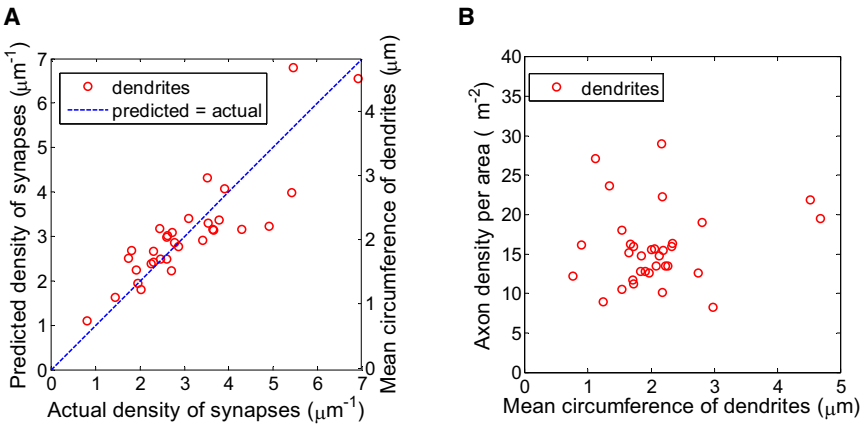


Figure 5. Relationship between Dendritic Caliber and the Density of Actual and Potential Synapses

(A) Plot of the actual density of synapses versus the density of synapses predicted by multiplying the dendritic circumference by the common coefficient. Dendritic caliber is a strong predictor of actual density of synapses along a dendrite ($r^2 \approx 0.75$).
(B) Density of available axons (per unit length of dendrite per unit distance from a dendrite) does not correlate with the dendritic caliber ($r^2 \approx 0.02$).

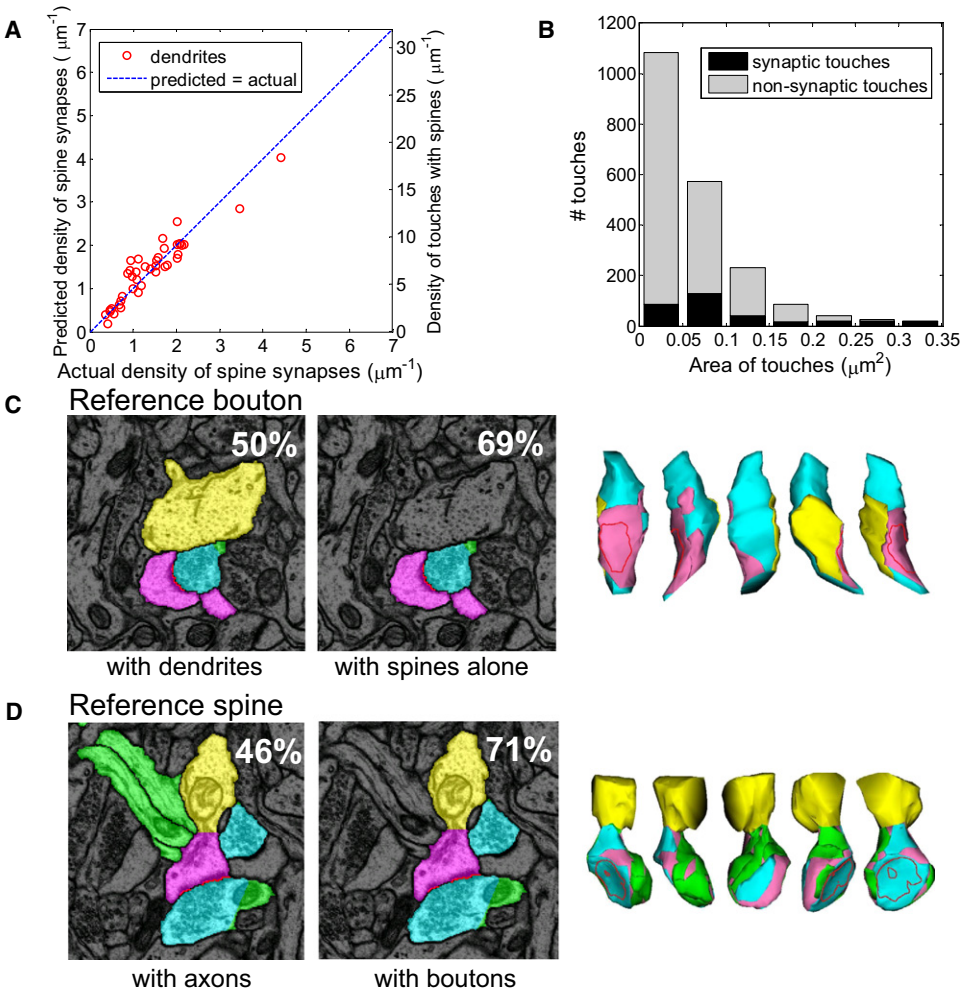


Figure 6. Relationship between Synaptic and Nonsynaptic Axo-dendritic Touches in V1

(A) Density of spine synapses along a dendrite is proportional to the density of spine touches with axons ($r^2 \approx 0.88$).

(B) Area distributions of synaptic and non-synaptic touches overlap significantly.

(C) Reference bouton whose largest touch with a spine corresponds to a synapse. Left: section containing the reference bouton (cyan) with touching dendrite (yellow) and spine (pink). Percentage of boutons with largest dendritic touch corresponding to a synapse is shown. Center: reference bouton (cyan) and touching spine (pink) form a synapse. Percentage of boutons with largest spine touch corresponding to a synapse is shown. Right: 3D views of the reference bouton colored according to the type of touching object. Visible blue areas are where other axons touched this bouton.

(D) Reference spine whose largest touch with a bouton corresponds to a synapse. Left: section containing the reference spine (pink) with touching axons (green) and boutons (cyan). Percentage of spines with largest axonal touch corresponding to a synapse is shown. Center: reference spine (pink) and touching boutons (cyan). Percentage of spines with largest bouton touch corresponding to a synapse. Right: 3D views of the reference spine surface colored according to the type of touching object. Red dotted line: position of the synapse.

See also Figure S5.

of touches using a procedure similar to that described in the previous section. First, by dividing the total number of synapses by the total number of touches we calculated the average touch connectivity fraction. Second, we calculated the predicted density of synapses on each dendrite by multiplying the density of touches on that dendrite by the average touch connectivity fraction.

The density of touches predicts the density of synapses well (Figure 6A). To determine whether the remaining discrepancy could have arisen by chance due to small counts of synapses on individual dendrites we applied the multihypothesis significance analysis again (see Experimental Procedures). We found that, for the invariant touch connectivity fraction, the probability of such discrepancy is large ($p > 0.05$; Figure S3C). Therefore, the hypothesis that synaptic density is a uniform fraction of the touch density cannot be rejected.

The discovered correlations of the synapse density should help to predict it from the density of proximities between one neuron's axons and other neuron's dendrites. The dependence of the synapse density on the caliber suggests a modification of the Peters' rule, where synapse probability is a function of both the number of proximities and the dendritic caliber. The

correlation of the synapse density with the touch density may be used in combination with the methods for touch identification such as SBFSEM and GFP recombination across synaptic partners, or GRASP (Feinberg et al., 2008) to predict the probability of a synapse.

When Do Axo-dendritic Touches Predict Individual Synapses?

Since the density of touches predicts synaptic density, it is natural to ask whether individual touches could reliably predict synapses. As the fraction of touches that correspond to synapses was much less than one (~0.2), additional information is needed to determine which touches correspond to synapses. If the target application for such a method would be a technique other than ssTEM, then, one cannot rely on synaptic attributes, such as vesicles and/or PSDs, and must instead rely on shape and geometrical proximity. We considered whether the area of a touch could predict a synapse but found that it was insufficient because the area distributions of synaptic and non-synaptic touches overlapped completely across the full range of sizes (Figure 6B).

Next, we explored a variation of this approach motivated by the observation that the sizes of boutons, spines and PSD area of a given synapse are correlated (Harris and Stevens, 1989; Lisman and Harris, 1993; Schikorski and Stevens, 1997; Pierce and Lewin, 1994). Moreover, the geometrical dimensions of hippocampal synapses correlate with the physiologically defined synaptic weight (Matsuzaki et al., 2001; Kasai et al., 2003). From these observations, we hypothesized that a spine and/or a bouton would not be bigger than that needed to accommodate a synaptic touch. Therefore, we tested (1) whether the largest touch a bouton has with adjacent dendrites or spines predicts a synapse, (2) whether the largest touch a spine has with adjacent axons or boutons predicts a synapse, (3) whether a combination of (1) and (2) predicts a synapse. In this analysis, we identified spines and boutons based only on their shapes without relying on synaptic ultrastructural attributes (see Experimental Procedures).

We started by exploring whether the relative area of touches made by a reference bouton with adjacent dendrites could predict a synapse (Figure 6C). We found that the largest-area touch corresponded to a synapse in about half of the cases. Next, we restricted our consideration to the touches among reference boutons and dendritic spines, not dendritic shafts. We found that for 69% of boutons the greatest area touch corresponded to a synapse on a spine (Figure 6C).

Although the majority of boutons' largest-area touches with spines correspond to synapses, there is a significant fraction of synapses that occur at other touches. These include synapses made with dendritic shafts and at nonlargest touches. Moreover, 17%–39% of Schaffer collateral boutons are multisynaptic (Sorra and Harris, 1993; Shepherd and Harris, 1998; Kirov et al., 1999). This means that a substantial fraction of synapses will be missed by this axo-centric largest-area touch method.

Next, we considered the relative area of touches made by a reference spine with adjacent axons (Figure 6D). The fraction of spines whose largest-area touch with axons corresponded to a synapse was less than half. This method was improved by considering only touches made with boutons, not with typically synapse-free interbouton intervals along the axon. For 71% of spines the largest area touch with adjacent boutons was synaptic (Figure 6D). As multisynaptic spines are much rarer (<1%) than multisynaptic boutons in perfusion fixed adult hippocampus (Fiala et al., 1998; Petrak et al., 2005), the fraction of spine synapses recovered by this method is also approximately 70%.

Finally, we combined these two approaches by considering a touch area relative to other touches both sharing the same bouton and the same spine. We found that 80% of the touches whose area is greatest among those sharing the same bouton and those sharing the same spine are synaptic. At the same time, this method detects only 46% of all spine synapses. Thus, the relative touch area is also an imperfect predictor of individual synapses on dendritic spines.

Our analysis focused on spine synapses because shaft synapses are rare along principal spiny dendrites in s. radiatum of area CA1. For example, in all of manual volume 3, there were only 17 asymmetric, putative excitatory shaft synapses, and only 12 symmetric, putative inhibitory synapses. Shaft synapses occur frequently along interneuron dendrites (Harris and Landis, 1986) but only two short segments of interneuron dendrites passed through volume 3. Hence, despite these being the largest volumes of hippocampal neuropil ever fully reconstructed, we were not able to analyze connectivity of the spine-free interneuron dendrites in this brain region.

DISCUSSION

In this paper, we fully reconstructed an unprecedented volume of hippocampal neuropil using ssTEM and automated registration and segmentation algorithms. Such reconstruction proves the feasibility of automating reconstructions on the scale impractical for manual reconstructions. Although the proofreading speed and the error rates are satisfactory for the analysis of the reconstructed volumes, they require radical improvement—via both hardware and software innovations—to reconstruct complete vertebrate circuits.

Full volume dense reconstruction allowed us to measure directly the numbers of nearby axons and synapses along each dendritic segment. The mean connectivity fraction calculated from these measurements is in agreement with the theoretical predictions based on light microscopy data (Stepanyants et al., 2002). Yet, the connectivity fraction varied among dendrites enough to make the use of Peters' rule unsuitable for predicting synaptic density and suggests the need to re-examine previous results (Binzegger et al., 2004; Stepanyants and Chklovskii, 2005; Jefferis et al., 2007; Stepanyants et al., 2008). Our measurements indicate possible ultrastructural causes for violations of Peters' rule obtained from light microscopy and physiology (Shepherd et al., 2005; Petreanu et al., 2009).

We found a strong correlation between the density of synapses and dendrite caliber and no correlation between the caliber and the density of available axons. This finding suggests that the density of synapses is determined not so much by the availability of axons in the local environment but more by intrinsic properties of the dendrites. The strong correlation previously

reported between dendritic cross-sectional area and microtubule number, and microtubule number and spine density further supports this hypothesis (Fiala et al., 2003; K.M. Harris et al., 2007, Soc. Neurosci., abstract). Interestingly, the scaling of synaptic density with the dendritic caliber implies the existence of a universal shaft membrane area (0.66 μm^2) per synapse, cf. (Nicol and Meinertzhagen, 1982).

The observed correlation between dendritic caliber and spine density among the segments of different dendrites is consistent with previous reports of spine density along individual dendrites as a function of distance from the cell body. In particular, the density of synapses decreases with the distance from the cell body along a given dendrite (Katz et al., 2009), which would be expected given that dendrites get thinner with distance from the cell body. Others have shown that the thickest proximal apical dendrites appear spine free, seemingly in contradiction (Megías et al., 2001). However, those proximal dendrites receive mostly inhibitory GABAergic synapses (Buhl et al., 1994; Megías et al., 2001) and our volume was taken from the middle of stratum radiatum distal to cell bodies. It will be interesting to learn whether inhibitory synapses also have a caliber to density rule in relationship to intrinsic composition or extrinsic features of the local neuropil.

We measured the distance between adjacent synapses along axons to be 4–5 μm, whereas Shepherd and Harris (1998) reported a lower intersynapse interval along axons averaging 2.7 μm. Later it was discovered that the adult hippocampal slices used in (Shepherd and Harris, 1998) had nearly 50% more synapses than adult hippocampus fixed by intracardiac perfusion (Kirov et al., 1999, 2004), as was used to obtain V1–3 reported here. This difference would account for the discrepancy in these axonal intersynapse measurements.

We found that touches between axons and dendrites (mostly spines) could be used to predict synapses on two levels. First, the density of synapses along a hippocampal dendrite appears to be a universal fraction, 0.2, of the density of touches. In contrast, the fraction of touches corresponding to synapses reported for the *C. elegans* nervous system is 0.09 (Durbin, 1987). Second, the largest touch shared by a spine and bouton predicts the presence of an actual spine synapse with about 80% precision.

Knowing the relationship between touches and synapses is valuable for techniques that do not contain the information present in ssTEM. For example, automated tracing from SBFSEM is done at lower resolution in combination with extracellular labeling that fails to reveal the two main indicators of synapses: pre-/postsynaptic zones and presynaptic vesicles. Another technique that could benefit from knowing the relationship between touches and synapses is GRASP (Feinberg et al., 2008). Although GRASP can identify synapses rather than touches by relying on synaptic proteins, this is not always done (Feinberg et al., 2008; Gordon and Scott, 2009), and may be undesirable as ectopic expression of synaptic proteins may alter connectivity (Scheiffele et al., 2000; Biederer et al., 2002; Zito et al., 2004). Finally, array tomography (Micheva and Smith, 2007) is a promising light microscopy technique with improved vertical resolution that can be used with synaptic markers to identify synaptic contacts. Our results may help one to interpret the observed proximities pre- and postsynaptic puncta.

Although our results provide guidance for reconstructing circuits with lower resolution methods, it is not clear how they would generalize beyond s. radiatum of the hippocampal area CA1. Reconstructing synaptic connectivity for each new brain region or cell type using lower resolution methods, which can be scaled to larger volumes, may require repeating this kind of analysis to determine region and dendrite-specific rules for identifying synapses. For example, the rules for identifying synaptic touches along nonspiny dendrites even within this subregion of the hippocampus may differ. Furthermore, it is also not clear what rules will apply for shaft synapses occurring on spiny densities, or small-touch synapses on multisynaptic boutons.

In conclusion, we have shown that ssTEM can be automated, in principle, but will require major advances in data acquisition and analysis to be a viable approach for reconstructing complete vertebrate circuits at the resolution of synapses. Importantly, we have used these dense reconstructions to test whether axo-dendritic proximities predict synaptic connectivity. We found that Peters rule does not predict dendritic spine density because of variations in the connectivity fraction. We found that dendritic spine density is better predicted by spine-bouton touches and dendritic caliber. Furthermore, the relative touch area predicts synapses with about 80% precision when both pre and postsynaptic dimensions of dendritic spines are considered.

EXPERIMENTAL PROCEDURES

Tissue Sources and Photographic Conditions

All procedures followed NIH guidelines for the humane care and use of laboratory animals. Volumes 1–3 were from hippocampal area CA1 of a perfusion-fixed male rat of the Long-Evans strain weighing 310 g (postnatal day 77; Harris and Stevens, 1989). Volume 4 was from a hippocampal slice that was prepared from a postnatal day 21 male rat of the Long-Evans strain and maintained in vitro for 3 hr prior to fixation as described (Fiala et al., 2003). All volumes were from the middle of s. radiatum about 150 to 200 microns from the hippocampal CA1 pyramidal cell soma. For volume 4, the series was located at a depth between 100 and 200 μm from the cut air surfaces of the slice where excellent tissue preservation occurred.

All series were cut according to our published protocols (K.M. Harris et al., 2007, Soc. Neurosci., abstract). Briefly, a diamond trimming tool (EMS, Electron Microscopy Sciences, Fort Washington, PA) was used to prepare small trapezoidal areas ~200 μm wide by 30–50 μm high. Serial thin sections were cut at ~45–50 nm on an ultramicrotome, mounted and counter stained with saturated ethanolic uranyl acetate, followed by Reynolds lead citrate, each for 5 min. Individual grids were placed in grid cassettes and stored in numbered gelatin capsules. The cassettes were mounted in a rotating stage to obtain uniform orientation of the sections on adjacent grids and the series were photographed at 10,000× (volume 4) or 5,000× (volumes 1–3) on a JEOL 1200EX or 1230 electron microscope (JEOL, Peabody, MA).

Manual Volume Reconstructions

Three-dimensional reconstructions and analyses were performed manually using the software entitled RECONSTRUCT (Fiala and Harris, 2002; Fiala, 2005), which is freely available from http://synapses.clm.utexas.edu. We digitally optimized images for brightness and contrast and colorized reconstructions to visualize structures of interest. To align manually, we indicated five or more fiducial points on adjacent pairs of serial sections that were in the same location (e.g., cross-sectioned mitochondria or microtubules). Then we chose the minimal algorithm in RECONSTRUCT to perform the alignment while blending the adjacent images. Pixel size was calibrated relative to a diffraction grating replica (Ernest F. Fullam, Latham, NY) photographed with each series, and section thickness was computed by dividing the diameters of longitudinally sectioned mitochondria by the number of sections they

spanned (Fiala and Harris, 2001b). Finally, the user manually traced outlines of all objects on each section and identified them as axon, dendrite, glia, spine, or synapse. RECONSTRUCT output had calibrated dimensions and 3D displays of reconstructed objects.

Automated Registration

Automated registration required two steps. First, the IMOD software (David Mastronarde, University of Colorado, Boulder) was used to obtain pair wise relative affine transforms between adjacent sections. In some cases, manual adjustment using the Midas tool in IMOD was required to initialize the registration algorithm. The pairwise transforms were propagated through the whole stack and an absolute transform was obtained for each section. The second step was aimed at eliminating registration mismatches remaining after affine registration. It involved calculating cross-correlations between 200 × 200 pixel image patches of adjacent sections to find a vector field of remaining miss-registration. We approximated this vector field by local distortion functions and aligned each pair of adjacent sections with sub-pixel precision using the Matlab Image Processing Toolbox (Natick, MA).

Automated Segmentation

Automated segmentation of objects used the set of image processing algorithms developed in (Mishchenko, 2009) to extract and link the 2D profiles corresponding to different neuronal processes across aligned serial sections. Each image was processed using a multiscale Gaussian-Smoothed Hessian-based ridge detector to search for plasma membranes as linear dark features of varying width. A fuzzy-logic anisotropic growth of detected membranes was used to bridge short regions where the membranes were grayer due to oblique sectioning or appeared broken. Detected profiles were filtered by retaining only closed contours, corresponding to true cell profiles, to reduce clutter in the images due to darkly stained organelles. Overlapping contours from adjacent sections were compared based on shape and texture cues to determine if they belonged to the same 3D object. All overlapping contours found to belong to the same neuronal processes were automatically grouped across serial sections.

Proofreading of Automatic Segmentations

To correct errors in automatic segmentation we developed a graphical user interface in Matlab called the ProofReading Tool (PRT) to guide proofreading in a systematic and focused manner. PRT compiled a list of significant 3D objects from the automatic segmentation. The original electron micrographs containing all intracellular organelles were used. An object's significance depended primarily on its total volume but also on its average diameter and clarity or composition of the cytoplasm, which influenced brightness. Then the PRT guided a user through this list of processes sequentially in the order of decreasing volumes. The user viewed corresponding neuronal processes through each section of the entire ssTEM stack and either confirmed or corrected them as necessary. Most of the corrections involved grouping together multiple fragments of an axon or attaching spine-necks to dendritic shafts. No manual tracing of boundaries was involved. Fragments of the same neuronal processes were continuously removed from the list so that each process was inspected only once. A final segmentation was produced in which gaps in contours were closed, contours were smoothed and holes were filled using watershed from markers performed on the inverse of the original images, where interiors of proofread objects were used as the markers. From this, a set of single-pixel lines was produced to represent boundaries of neuronal processes in different sections. The final reconstruction was stored as a bitmap of the entire volume, where each pixel carried a numerical label to identify the process containing it. We also generated RECONSTRUCT XML series from the final segmentations for visualization or quantification using RECONSTRUCT.

Distribution of Effective Cross-section Diameters of Axons and Dendrites

First, we used the Z-trace tool of RECONSTRUCT to measure each dendrite's length across serial sections. A morphological shrinking transformation was applied to each dendritic profile to get central points that were then connected from section to section by the hypotenuse of a right triangle with one side equal to the x-y distance between the two points and the other side equal to the section thickness of 50 nm. The Z-trace length of a dendrite equaled the sum of lengths of all these hypotenuses. Second we computed mean cross-sectional area, A, for each axon and dendrite by dividing the volume for each segment by its length. The effective diameter, d, then was calculated using formula $A = \pi d^2/4$. The resulting distribution is shown in Figure 2B.

Detection of Synapses

We developed a PSD recognition algorithm to detect post-synaptic densities automatically in images by searching for synapses as broader fragments of external cell boundaries with high stain density. For every point on the single-pixel boundary between an axon and a dendrite, we computed three integrals along the direction orthogonal to the boundary at that point, measuring the total integral of the image intensity and the first and the second power moments of the distance from the boundary weighted with the image intensity, out to a specified distance. The first integral measured the total darkness of the boundary at each location, and the other two integrals measured the width of such boundary. For a PSD, the first integral would indicate a very dark region, and the other two would indicate wider than usual membrane. These three measures were used as inputs to a single-layer logit neural network classifier (Haykin, 2008) trained to produce PSD score describing whether the pixel was inside a PSD. The PSD recognition algorithm was trained on 1–2 manually annotated images. For each axon-dendritic pair in contact, the total PSD score along their contact boundary was produced and used to determine if the pair made the synaptic contact. The algorithm could detect synaptic connections in a volume with 15% false negative and 20% false positive errors. The error rate was estimated by cross-validating with the manually composed list of synaptic connections. Then, we manually verified all synaptic connections in volumes 1 and 4. By repeating the manual verification process twice, we estimate that these manually verified datasets missed 7%–8% of synapses and contained 2%–3% false synapses. In volume 3, the PSD recognition algorithm was not used; instead, synapses throughout the volume were marked during the process of manual tracing in RECONSTRUCT.

After identifying the PSD traces, we computed the PSD areas as follows. We interpolated the surface between traces of PSD on adjacent slices with triangles. Then the total PSD area was calculated by adding the areas of these triangles plus the lengths of the two outer most PSD traces times 1/2 of the slice thickness. Compared to (Harris and Stevens, 1989), the PSD areas are within range although systematically smaller due to more strict inclusion criteria for the edge pixels of the PSD traces.

Computation of Distance-Dependent Connectivity Fraction

Distance-dependent connectivity fractions were calculated for each dendrite as the fraction of potential connections utilized in each radial shell following the surface outline of the dendritic shaft after the spines had been truncated. These quantities were sensitive to boundary effects when radial shells extended partially outside the sample volume. To correct for boundary effects, we divided the number of synapses in each shell by the fraction of the full radial shell actually included in the volume, and the number of axons by the fraction of full radial shell actually included in the volume with respect to one-half of the full shell's volume (but not greater than one). A factor of one-half was introduced here because each axon traversed the radial shell at two points. Each radial shell was explicitly continued outside of the volume to obtain an accurate estimate of its included fraction. As the fraction of axons grazing the shells was small, treating them the same way did not introduce a significant error. We restricted the sample of dendritic segments to include only those that were longer than 1 μm. In V3 and V4 we only included those segments that spent more than 50% of their length farther than 0.5 μm away from the volume's edge. In volume 1 this latter criterion was not necessary because V1 was big enough that dendrites on the boundaries did not affect estimates of connectivity fractions.

Calculation of Synaptic Density using Peters' Rule with Distance-Dependent Connectivity Fraction

Since the calculation of the average distance-dependent connectivity fraction may contain a significant uncertainty, we derived an expression for the density of synapses without explicitly using the dependence of the connectivity fraction on distance by taking advantage of the following observation. With the exception of axons touching dendritic shaft, the density of axons (per unit

length of a dendrite per unit distance from a dendrite) as a function of the distance to the dendrite is constant (Figure 4B) in agreement with prior theoretical analysis (Stepanyants et al., 2002).. Considering only those axons that do not touch the dendritic shaft, then:

$$\text{Synapses per } \mu\text{m} = \int \text{ds Density of axons (per area)(s)} \cdot \text{connectivity fraction(s)}$$

As the density of axons is independent of s (Figure 4B), it can be taken out of the integral. Then, even if the connectivity fraction varies with distance from a dendrite, as long as this function is invariant among dendrites, the integral has the same value for all of them. Therefore, we can estimate the integral by dividing the total number of synapses by the total number of axons and use this value to predict the density of synapses on each dendrite.

We also found that the number of axons touching dendritic shafts was not a good predictor of the number of synapses (Figure S4). By adding the predictions for axons touching and not touching dendritic shaft, we arrived at the total density of synapses, Figure 4C.

Delineating Axonal Boutons and Spines

The partitioning of axonal boutons from inter-bouton regions relies on their swollen shapes (Figure S5A). We computed a 3D distance transform from the surface of each axon inwards. Every voxel inside each axon was assigned the value of the shortest distance to the surface of the process. We calculated the average of the regional maxima and applied a morphological opening operation, which pinches narrow axon processes with distances to the surface shorter than 1.5 times the mean regional maxima. The remains having touches with dendrites are identified as boutons.

The detection of dendritic spines took advantage of their characteristic shapes using the following mathematical procedure. Note that, in a reconstructed dendrite, every voxel connects to the surface of the reconstructed volume by at least one path fully contained within that dendrite. After applying a morphological opening operation to the dendrite, which pinched narrow spine necks, voxels that remained connected to the surface belonged to the shaft, while those disconnected from the volume surface belonged to spines. Figure S5B shows an example of spine segmentation. The definition of a spine used here automatically is intermediate between the "spine" and "spine head" previously defined manually.

Multihypothesis Significance Analysis

To evaluate the significance of discrepancy between actual and predicted density of synapses, we calculate the probability of obtaining such discrepancy by chance due to a finite number of synapses per dendrite. We assume that synapses are drawn independently with equal probability set by the connectivity fraction and calculate the probability of observed or greater deviation from the predicted value. In the case of Peters' rule with maximum-reach connectivity fraction (Figure 3), the number of synapses on each dendrite is governed by a Binomial distribution. To avoid boundary effects, we calculate the p values only for spiny dendrites with shaft lengths greater than 2 μm and at distances more than 1 μm away from the boundary. Then we apply the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001), which has greater statistical power than the commonly used Bonferroni correction. The p values for m dendrites are arranged in ascending order, $p_1 \le p_2 \le \ldots p_m$, and adjusted to $p_i^a = \min(mp_i/i, 1)$ (Figure S3A). The multiple hypothesis corrected p value $p = \min(\{p_i^a\})$ is then compared to the standard false discovery rate $\alpha = 0.05$. We find that $p<0.05$, implying that the hypothesis of Peters' rule with maximum-reach connectivity fraction can be rejected. Similar analysis performed on the distance dependent connectivity fraction prediction also yielded $p<0.05$, thus rejecting the hypothesis (data not shown).

We performed the same significance analysis on the predictions using the number of axons touching dendrites (Figure S3B). As the multiple-hypothesis corrected $p > 0.05$, we cannot reject this hypothesis.

We performed a similar analysis for the prediction based on mean circumference of dendrites (Figure S3C). In this case, we could not use the Binomial distribution because the total number of axons was not known. As the numbers of surrounding axons are usually large (>100) and the connectivity fraction is small (Figure 4B), we approximated the Binomial distribution as a Poisson with mean equal to the predicted number of synapses.

SUPPLEMENTAL INFORMATION

Supplemental Information includes five figures and can be found with this article online at doi:10.1016/j.neuron.2010.08.014.

ACKNOWLEDGMENTS

We are grateful to Armen Stepanyants, Tom Bartol, Chandra Bajaj, Terry Sejnowski, Karel Svoboda, Jeff Magee, and Winfried Denk for helpful discussions and comments on the manuscript, to the anonymous reviewers for constructive comments, and to Stephen Clow for assistance with proofreading. K.M.H. was supported by NS21184 and EB002170; D.B.C. was partially supported by the Swartz Foundation.

Accepted: July 27, 2010
Published: September 22, 2010

REFERENCES

Anderson, J.R., Jones, B.W., Yang, J.H., Shaw, M.V., Watt, C.B., Koshevoy, P., Spaltenstein, J., Jurrus, E., U v, K., Whitaker, R.T., et al. (2009). A computational framework for ultrastructural mapping of neural circuitry. PLoS Biol. *7*, e1000074. 10.1371/journal.pbio.1000074.

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate—a practical and powerful approach to multiple testing. J. R. Stat. Soc. B *57*, 289–300.

Benjamini, Y., and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. Ann. Stat. *29*, 1165–1188.

Biederer, T., Sara, Y., Mozhayeva, M., Atasoy, D., Liu, X., Kavalali, E.T., and Südhof, T.C. (2002). SynCAM, a synaptic adhesion molecule that drives synapse assembly. Science *297*, 1525–1531.

Binzegger, T., Douglas, R.J., and Martin, K.A. (2004). A quantitative map of the circuit of cat primary visual cortex. J. Neurosci. *24*, 8441–8453.

Braitenberg, V., and Schuz, A. (1998). Cortex: Statistics and Geometry of Neuronal Connectivity (Berlin: Springer).

Briggman, K.L., and Denk, W. (2006). Towards neural circuit reconstruction with volume electron microscopy techniques. Curr. Opin. Neurobiol. *16*, 562–570.

Buhl, E.H., Halasy, K., and Somogyi, P. (1994). Diverse sources of hippocampal unitary inhibitory postsynaptic potentials and the number of synaptic release sites. Nature *368*, 823–828.

Bushong, E.A., Martone, M.E., Jones, Y.Z., and Ellisman, M.H. (2002). Protoplasmic astrocytes in CA1 stratum radiatum occupy separate anatomical domains. J. Neurosci. *22*, 183–192.

Chen, B.L., Hall, D.H., and Chklovskii, D.B. (2006). Wiring optimization can relate neuronal structure and function. Proc. Natl. Acad. Sci. USA *103*, 4723–4728.

Chklovskii, D.B., Schikorski, T., and Stevens, C.F. (2002). Wiring optimization in cortical circuits. Neuron *34*, 341–347.

da Costa, N.M., and Martin, K.A. (2009). Selective targeting of the dendrites of corticothalamic cells by thalamic afferents in area 17 of the cat. J. Neurosci. *29*, 13919–13928.

Denk, W., and Horstmann, H. (2004). Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. PLoS Biol. *2*, e329. 10.1371/journal.pbio.0020329.

Durbin, R.M. (1987). Studies on the development and organisation of the nervous system of Caenorhabditis elegans. PhD thesis. University of Cambridge, Cambridge, UK.

Feinberg, E.H., Vanhoven, M.K., Bendesky, A., Wang, G., Fetter, R.D., Shen, K., and Bargmann, C.I. (2008). GFP Reconstitution Across Synaptic Partners

(GRASP) defines cell contacts and synapses in living nervous systems. Neuron *57*, 353–363.

Fiala, J.C. (2005). Reconstruct: a free editor for serial section microscopy. J. Microsc. *218*, 52–61.

Fiala, J.C., and Harris, K.M. (2001a). Extending unbiased stereology of brain ultrastructure to three-dimensional volumes. J. Am. Med. Inform. Assoc. *8*, 1–16.

Fiala, J.C., and Harris, K.M. (2001b). Cylindrical diameters method for calibrating section thickness in serial electron microscopy. J. Microsc. *202*, 468–472.

Fiala, J.C., and Harris, K.M. (2002). Computer-based alignment and reconstruction of serial sections. Microscopy and Analysis *87*, 5–8.

Fiala, J.C., Feinberg, M., Popov, V., and Harris, K.M. (1998). Synaptogenesis via dendritic filopodia in developing hippocampal area CA1. J. Neurosci. *18*, 8900–8911.

Fiala, J.C., Kirov, S.A., Feinberg, M.D., Petrak, L.J., George, P., Goddard, C.A., and Harris, K.M. (2003). Timing of neuronal and glial ultrastructure disruption during brain slice preparation and recovery in vitro. J. Comp. Neurol. *465*, 90–103.

Gordon, M.D., and Scott, K. (2009). Motor control in a *Drosophila* taste circuit. Neuron *61*, 373–384.

Harris, K.M. (2008). Diversity in synapse structure and composition. In Structural and Functional Organization of the Synapse, J.W. Hell and M.D. Ehlers, eds. (New York: Springer).

Harris, K.M., and Landis, D.M. (1986). Membrane structure at synaptic junctions in area CA1 of the rat hippocampus. Neuroscience *19*, 857–872.

Harris, K.M., and Stevens, J.K. (1989). Dendritic spines of CA 1 pyramidal cells in the rat hippocampus: serial electron microscopy with reference to their biophysical characteristics. J. Neurosci. *9*, 2982–2997.

Haykin, S. (2008). Neural Networks and Learning Machines (Upper Saddle River, NJ: Prentice Hall).

Helmstaedter, M., Briggman, K.L., and Denk, W. (2008). 3D structural imaging of the brain with photons and electrons. Curr. Opin. Neurobiol. *18*, 633–641.

Jefferis, G.S., Potter, C.J., Chan, A.M., Marin, E.C., Rohlfing, T., Maurer, C.R., Jr., and Luo, L. (2007). Comprehensive maps of Drosophila higher olfactory centers: spatially segregated fruit and pheromone representation. Cell *128*, 1187–1203.

Jurrus, E., Whitaker, R., Jones, B.W., Marc, R., and Tasdizen, T. (2008). An optimal-path approach for neural circuit reconstruction. Proc./IEEE Int. Symp. Biomed. Imaging *2008*, 1609–1612.

Kasai, H., Matsuzaki, M., Noguchi, J., Yasumatsu, N., and Nakahara, H. (2003). Structure-stability-function relationships of dendritic spines. Trends Neurosci. *26*, 360–368.

Katz, Y., Menon, V., Nicholson, D.A., Geinisman, Y., Kath, W.L., and Spruston, N. (2009). Synapse distribution suggests a two-stage model of dendritic integration in CA1 pyramidal neurons. Neuron *63*, 171–177.

Kirov, S.A., Sorra, K.E., and Harris, K.M. (1999). Slices have more synapses than perfusion-fixed hippocampus from both young and mature rats. J. Neurosci. *19*, 2876–2886.

Kirov, S.A., Petrak, L.J., Fiala, J.C., and Harris, K.M. (2004). Dendritic spines disappear with chilling but proliferate excessively upon rewarming of mature hippocampus. Neuroscience *127*, 69–80.

Lisman, J.E., and Harris, K.M. (1993). Quantal analysis and synaptic anatomy—integrating two views of hippocampal plasticity. Trends Neurosci. *16*, 141–147.

Livet, J., Weissman, T.A., Kang, H., Draft, R.W., Lu, J., Bennis, R.A., Sanes, J.R., and Lichtman, J.W. (2007). Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. Nature *450*, 56–62.

Luo, L., Callaway, E.M., and Svoboda, K. (2008). Genetic dissection of neural circuits. Neuron *57*, 634–660.

Matsuzaki, M., Ellis-Davies, G.C., Nemoto, T., Miyashita, Y., Iino, M., and Kasai, H. (2001). Dendritic spine geometry is critical for AMPA receptor expression in hippocampal CA1 pyramidal neurons. Nat. Neurosci. *4*, 1086–1092.

Megías, M., Emri, Z., Freund, T.F., and Gulyás, A.I. (2001). Total number and distribution of inhibitory and excitatory synapses on hippocampal CA1 pyramidal cells. Neuroscience *102*, 527–540.

Micheva, K.D., and Smith, S.J. (2007). Array tomography: a new tool for imaging the molecular architecture and ultrastructure of neural circuits. Neuron *55*, 25–36.

Mishchenko, Y. (2009). Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs. J. Neurosci. Methods *176*, 276–289.

Nicol, D., and Meinertzhagen, I.A. (1982). Regulation in the number of fly photoreceptor synapses: the effects of alterations in the number of presynaptic cells. J. Comp. Neurol. *207*, 45–60.

Peters, A., and Feldman, M.L. (1976). The projection of the lateral geniculate nucleus to area 17 of the rat cerebral cortex. I. General description. J. Neurocytol. *5*, 63–84.

Peters, A., Palay, S.L., and Webster, H. (1991). The Fine Structure of the Nervous System (Oxford: Oxford University Press).

Petrak, L.J., Harris, K.M., and Kirov, S.A. (2005). Synaptogenesis on mature hippocampal dendrites occurs via filopodia and immature spines during blocked synaptic transmission. J. Comp. Neurology *484*, 183–190.

Petreanu, L., Mao, T., Sternson, S.M., and Svoboda, K. (2009). The subcellular organization of neocortical excitatory connections. Nature *457*, 1142–1145.

Pierce, J.P., and Lewin, G.R. (1994). An ultrastructural size principle. Neuroscience *58*, 441–446.

Scheiffele, P., Fan, J., Choih, J., Fetter, R., and Serafini, T. (2000). Neuroligin expressed in nonneuronal cells triggers presynaptic development in contacting axons. Cell *101*, 657–669.

Schikorski, T., and Stevens, C.F. (1997). Quantitative ultrastructural analysis of hippocampal excitatory synapses. J. Neurosci. *17*, 5858–5867.

Shepherd, G.M., and Harris, K.M. (1998). Three-dimensional structure and composition of CA3→CA1 axons in rat hippocampal slices: implications for presynaptic connectivity and compartmentalization. J. Neurosci. *18*, 8300–8310.

Shepherd, G.M., Stepanyants, A., Bureau, I., Chklovskii, D., and Svoboda, K. (2005). Geometric and functional organization of cortical circuits. Nat. Neurosci. *8*, 782–790.

Smith, S.J. (2007). Circuit reconstruction tools today. Curr. Opin. Neurobiol. *17*, 601–608.

Sorra, K.E., and Harris, K.M. (1993). Occurrence and three-dimensional structure of multiple synapses between individual radiatum axons and their target pyramidal cells in hippocampal area CA1. J. Neurosci. *13*, 3736–3748.

Sorra, K.E., and Harris, K.M. (2000). Overview on the structure, composition, function, development, and plasticity of hippocampal dendritic spines. Hippocampus *10*, 501–511.

Stepanyants, A., and Chklovskii, D.B. (2005). Neurogeometry and potential synaptic connectivity. Trends Neurosci. *28*, 387–394.

Stepanyants, A., Hof, P.R., and Chklovskii, D.B. (2002). Geometry and structural plasticity of synaptic connectivity. Neuron *34*, 275–288.

Stepanyants, A., Hirsch, J.A., Martinez, L.M., Kisvárday, Z.F., Ferecskó, A.S., and Chklovskii, D.B. (2008). Local potential connectivity in cat primary visual cortex. Cereb. Cortex *18*, 13–28.

Ventura, R., and Harris, K.M. (1999). Three-dimensional relationships between hippocampal synapses and astrocytes. J. Neurosci. *19*, 6897–6906.

White, E.L. (2002). Specificity of cortical synaptic connectivity: emphasis on perspectives gained from quantitative electron microscopy. J. Neurocytol. *31*, 195–202.

White, E.L., and Rock, M.P. (1981). A comparison of thalamocortical and other synaptic inputs to dendrites of two non-spiny neurons in a single barrel of mouse SmI cortex. J. Comp. Neurol. *195*, 265–277.

White, J.G., Southgate, E., Thompson, J.N., and Brenner, S. (1986). The structure of the nervous system of the nematode caenorhabditis elegans. Philos. Trans. R. Soc. Lond. B Biol. Sci. *314*, 1–340.

Zito, K., Knott, G., Shepherd, G.M., Shenolikar, S., and Svoboda, K. (2004). Induction of spine growth and synapse formation by regulation of the spine actin cytoskeleton. Neuron *44*, 321–334.

A network of spiking neurons for computing sparse representations in an energy efficient way

Tao Hu[1]
hut@janelia.hhmi.org

Alexander Genkin[2]
alexgenkin@iname.com

Dmitri B. Chklovskii[1]
mitya@janelia.hhmi.org

[1]*Howard Hughes Medical Institute, Janelia Farm Research Campus, Ashburn, VA 20147, USA*
[2]*AVG Consulting, Brooklyn, NY, USA*

Computing sparse redundant representations is an important problem both in applied mathematics and neuroscience. In many applications, this problem must be solved in an energy efficient way. Here, we propose a hybrid distributed algorithm (HDA), which solves this problem on a network of simple nodes communicating via low-bandwidth channels. HDA nodes perform both gradient-descent-like steps on analog internal variables and coordinate-descent-like steps via quantized external variables communicated to each other. Interestingly, such operation is equivalent to a network of integrate-and-fire neurons, suggesting that HDA may serve as a model of neural computation. We compare the numerical performance of HDA with existing algorithms and show that in the asymptotic regime the representation error of HDA decays with time, *t*, as 1/*t*. We show that HDA is stable against time-varying noise, specifically, the representation error decays as $1/\sqrt{t}$ for Gaussian white noise.

1 Introduction

Many natural signals can be represented as linear combinations of a few feature vectors (or elements) chosen from a redundant (or overcomplete) dictionary. Such representations are called sparse because most dictionary elements enter with zero coefficients. The importance of sparse representations has been long recognized in applied mathematics (Chen et al., 1998, Baraniuk, 2007) and in neuroscience, where electrophysiological recordings (DeWeese et al., 2003) and theoretical arguments (Attwell and Laughlin, 2001, Lennie, 2003) demonstrate that most neurons are silent at any given moment (Olshausen and Field, 1996, Gallant and Vinje, 2000, Olshausen and Field, 2004).

In applied mathematics, sparse representations lie at the heart of many important developments. In signal processing, such solutions serve as a foundation for basis pursuit (Chen et al., 1998) de-noising, compressive sensing (Baraniuk, 2007) and object recognition (Kavukcuoglu et al., 2010). In statistics, regularized multivariate regression algorithms, such as the Lasso (Tibshirani, 1996) or the elastic net (Zou and Hastie, 2005), rely on sparse representations to perform feature subset selection along with coefficient fitting. Given the importance of finding sparse representations, it is not surprising that many algorithms have been proposed for the task (Efron et al., 2004, Zou and Hastie, 2005, Friedman et al., 2007, Yin et al., 2008, Cai et al., 2009a, b, Li and Osher, 2009, Xiao, 2010). However, most algorithms are designed for CPU architectures and are computationally and energy intensive.

Given the ubiquity of sparse representations in neuroscience, how can neural networks generate sparse representations remains a central question. Building on the seminal work of Olshausen and Field (Olshausen and Field, 1996), Rozell et al. have proposed an algorithm for sparse

representations by neural networks called Local Competitive Algorithm (LCA) (Rozell et al., 2008). Such algorithm computes a sparse representation on a network of nodes that communicate analog variables with each other. Although a step towards biological realism, the LCA neglects the fact that most neurons communicate using action potentials (or spikes) - quantized all-or-none electrical signals. Although spiking neurons can communicate analog variables by firing rates, their punctuate nature leads to computational inferiority relative to pure analog unless the limit of large number of spikes is taken (Deneve and Boerlin, 2011, Shapero et al., 2011). However, this limit erases the advantage of spiking in terms of energy efficiency, an important consideration in brain design (Attwell and Laughlin, 2001, Laughlin and Sejnowski, 2003).

In this paper, we introduce an energy efficient algorithm called hybrid distributed algorithm (HDA), which computes sparse redundant representations on the architecture of (Rozell et al., 2008) but using neurons that spike. We demonstrate that such algorithm performs as well as the analog one, thus suggesting that spikes may not detrimentally affect computational capabilities of neural circuits. Moreover, HDA can serve as a plausible model of neural computation because local operations are described by the biologically inspired integrate-and-fire neurons (Koch, 1999, Dayan and Abbott, 2001). Other spiking neuron models have been proposed for sensory integration, working memory (Boerlin and Deneve, 2011) and implementing dynamical systems (Deneve and Boerlin, 2011, Shapero et al., 2011).

Because spiking communication requires smaller bandwidth, HDA may also be useful for sensor networks, which must discover sparse causes in distributed signals. In particular, large networks of small autonomous nodes are commonly deployed both for civilian and military applications, such as monitoring the motion of tornado or forest fires, tracking traffic conditions, security surveillance in shopping malls and parking facilities, locating and tracking enemy movements, detection of terrorist threats and attacks, (Tubaishat and Madria, 2003). The nodes of such networks use finite-life or slowly charging batteries and, hence, must operate under limited energy budget. Therefore, low-energy computations and limited bandwidth communication are two central design principles of such networks. Because correlations are often present among distributed sensor nodes, computing sparse redundant representations is an important task.

The paper is organized as follows. In §2 we describe the Bregman iteration method for computing sparse representations and briefly introduce two other distributed methods. We then consider a refined Bregman iteration method with coordinate descent modifications (§3) and continue in §4 by deriving our hybrid distributed algorithm. In §5 we prove the asymptotic performance guarantee of HDA, and demonstrate its numerical performance in §6. Finally, we conclude with the discussion of the advantages of HDA (§7).

2 Problem statement and existing distributed algorithms

A sparse solution $\boldsymbol{u} \in \mathbb{R}^n$ of the equation $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$, where $\boldsymbol{f} \in \mathbb{R}^m$, and wide matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ $(n > m)$ can be found by solving the following constrained optimization problem:

$$\min \|\boldsymbol{u}\|_1 \text{ s.t. } \boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}, \tag{1}$$

which is known as basis pursuit (Chen et al., 1998). In practical applications, where $\boldsymbol{f}$ contains noise, one typically formulates the problem differently, in terms of an unconstrained optimization problem known as the Lasso (Tibshirani, 1996):

$$\min \frac{1}{2}\|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 + \lambda\|\boldsymbol{u}\|_1, \tag{2}$$

where λ is the regularization parameter which controls the trade-off between representation error and sparsity. The choice of regularization by l_1-norm assures that the problem both remains convex (Boyd and Vandenberghe, 2004, Dattorro, 2008, Bertsekas, 2009) and favors sparse solutions (Tibshirani, 1996, Chen et al., 1998). In this paper we introduce an energy efficient algorithm that searches for a solution to the constrained optimization problem (1) by taking steps towards solving a small number of unconstrained optimization problems (2). Our algorithm is closest to the family of algorithms called Bregman iterations (Yin et al., 2008, Cai et al., 2009a, b, Osher et al., 2010), which take their name from the replacement of the l_1-norm by its Bregman

divergence (Bregman, 1967), $D(\boldsymbol{u}, \boldsymbol{u}^k) = \lambda\|\boldsymbol{u}\|_1 - \lambda\|\boldsymbol{u}^k\|_1 - \langle \boldsymbol{p}^k, \boldsymbol{u} - \boldsymbol{u}^k \rangle$, where $\boldsymbol{p}$ is a sub-gradient of $\lambda\|\boldsymbol{u}\|_1$ (Boyd and Vandenberghe, 2004).. The iterations start with $\boldsymbol{u}^0$=$\boldsymbol{p}^0$=0 and consist of two steps:

$$\boldsymbol{u}^{k+1} = \operatorname{argmin}_u E$$
$$= \operatorname{argmin}_u \left\{\tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 + \lambda\|\boldsymbol{u}\|_1 - \lambda\|\boldsymbol{u}^k\|_1 - \langle \boldsymbol{p}^k, \boldsymbol{u} - \boldsymbol{u}^k \rangle\right\}. \quad (3)$$

$$\boldsymbol{p}^{k+1} = \boldsymbol{p}^k - \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{u}^{k+1} - \boldsymbol{f}). \quad (4)$$

Throughout the paper, we assume that $\boldsymbol{A}$ is column normalized, i.e. if $\boldsymbol{A}_i$ is the i-th column of $\boldsymbol{A}$, $\boldsymbol{A}_i^T\boldsymbol{A}_i = 1$. Note that, because $n > m$, $\boldsymbol{A}$ defines a (redundant) frame. Moreover, we assume that $|\boldsymbol{A}_i^T\boldsymbol{A}_j| < 1$ for any $i \neq j$.

A practical algorithm for solving (1) called linearized Bregman iterations (LBI) is derived by solving the optimization problem (3) approximately (Yin et al., 2008, Cai et al., 2009a, b). The square error term in Eq. (3) is replaced by its linear approximation $\langle \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}), \boldsymbol{u} - \boldsymbol{u}^k \rangle$ around $\boldsymbol{u}^k$ and a proximity term $\frac{1}{2\delta}\|\boldsymbol{u} - \boldsymbol{u}^k\|_2^2$ is added to reflect the limited range of validity of the linear approximation. After some algebra the steps (3) and (4) reduce to the following two-step LBI (Yin et al., 2008, Cai et al., 2009a, b):

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k - \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{u}^k - \boldsymbol{f}), \quad (5)$$

$$\boldsymbol{u}^{k+1} = \delta\ \text{shrink}(\boldsymbol{v}^{k+1}, \lambda), \quad (6)$$

where $\boldsymbol{v}^k = \boldsymbol{p}^k + \boldsymbol{u}^k/\delta$ and the component wise operation $\text{shrink}(x, \lambda) = \begin{cases} x - \lambda, & \text{if } x > \lambda \\ 0, & \text{if } -\lambda < x < \lambda \\ x + \lambda, & \text{if } x < -\lambda \end{cases}$ (Elad et al., 2007).

The LBI can be naturally implemented by a network of n parallel nodes, Figure 1, an architecture previously proposed to implement LCA (Rozell et al., 2008). Such a network combines feedforward projections, $\boldsymbol{A}^T$, and inhibitory lateral connections, $-\boldsymbol{A}^T\boldsymbol{A}$, which implement "explaining away" (Pearl, 1988). At every step, each node updates its component of the internal variable, $\boldsymbol{v}$, by adding the corresponding components of the feedforward signal, $\boldsymbol{A}^T\boldsymbol{f}$, and the broadcast external variable, $-\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{u}$. Then, each node computes the new value of its component in $\boldsymbol{u}$ by shrinking its component in $\boldsymbol{v}$. Another distributed algorithm called RDA (Xiao, 2010) can also be implemented by such a network.

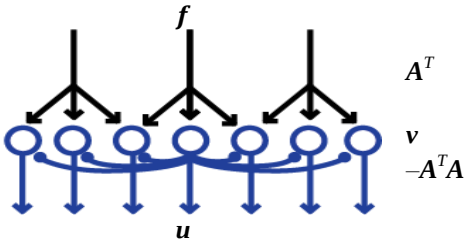


Figure 1: A network architecture for LCA, RDA, LBI, or HDA. Feedforward projections multiply the input $\boldsymbol{f}$ by a matrix $\boldsymbol{A}^T$, while lateral connections update internal node activity $\boldsymbol{v}$ by a product of matrix $-\boldsymbol{A}^T\boldsymbol{A}$ and external activity $\boldsymbol{u}$.

Although LBI, LCA or RDA achieve sparse approximation of the incoming signal, implementing these algorithms in man-made or biological hardware using the network architecture of Fig. 1 would be challenging in practice. The reason is that all these algorithms require real-time communication of analog variables, thus placing high demands on the energy consumption and bandwidth of lateral connections. Considering that the potential number of lateral connections is $O(n^2)$, and both volume and energy are often a limited resource in the brain (Attwell and Laughlin, 2001, Chklovskii et al., 2002, Laughlin and Sejnowski, 2003) and in sensor networks (Tubaishat and Madria, 2003) we search for a more efficient solution.

3 Bregman coordinate descent

In an attempt to find a distributed algorithm for solving (1) under bandwidth limitations, we explore a different strategy, called coordinate descent, where only one component of $\boldsymbol{u}$ is updated at a given iteration (Friedman et al., 2007). Inspired by (Li and Osher, 2009) we derive a novel Bregman coordinate descent algorithm. We start from (3) and rewrite the energy function on the right hand side by substituting matrix notation with explicit summation over vector components:

$$E=\frac{1}{2}\left\|\sum_{j=1}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right\|_2^2 + \lambda \sum_{j=1}^{n} \left\|u_j\right\|_1 - \lambda \sum_{j=1}^{n} \left\|u_j^k\right\|_1 - \sum_{j=1}^{n} p_j^k (u_j - u_j^k). \quad (7)$$

Assuming that in the (k+1)-th iteration, the i-th component of $\boldsymbol{u}$ is to be updated, and the values of all other components of $\boldsymbol{u}$ remain unchanged, then the updated value u_i' is obtained from

$$u_i' = \text{argmin}_{u_i} E = \text{argmin}_{u_i} \left\{\frac{1}{2}\left\|u_i \boldsymbol{A}_i + \sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right\|_2^2 + \lambda \|u_i\|_1 - p_i u_i\right\}, \quad (8)$$

where we drop terms independent of u_i and do not keep track of the iteration number k. The condition for the minimum in iteration (8) is

$$\partial[\lambda \|u_i'\|_1] \ni -\boldsymbol{A}_i^T \left(u_i' \boldsymbol{A}_i + \sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right) + p_i, \quad (9)$$

where ∂ [.] designates a subdifferential (Boyd and Vandenberghe, 2004). Noticing $\boldsymbol{A}_i^T \boldsymbol{A}_i = 1$ and $\sum_{j\neq i}^{n} u_j \boldsymbol{A}_j = \boldsymbol{A}u - u_i \boldsymbol{A}_i$, we rewrite (9) as

$$\partial[\lambda \|u_i'\|_1] \ni -u_i' + u_i - \boldsymbol{A}_i^T (Au - \boldsymbol{f}) + p_i, \quad (10)$$

From the optimality condition (10), we get the update formula of p_i (Yin et al., 2008),

$$p_i' + u_i' = p_i + u_i - \boldsymbol{A}_i^T (\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}), \quad (11)$$

where $\partial[\lambda \|u_i'\|_1] \ni p_i'$. By defining $v_i = p_i + u_i$, we get:

$$v_i' = v_i - \boldsymbol{A}_i^T (\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}). \quad (12)$$

Then we derive the update formula of u_i. Noticing

$$\begin{aligned} &\left\|u_i \boldsymbol{A}_i + \sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right\|_2^2 = u_i^2 + 2u_i \boldsymbol{A}_i^T \left(\sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right) + \left\|\sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right\|_2^2 \\ &= \left\|u_i + \boldsymbol{A}_i^T \left(\sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right)\right\|_2^2 - \left\|\boldsymbol{A}_i^T \left(\sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right)\right\|_2^2 + \left\|\sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right\|_2^2 \\ &= \left\|u_i + \boldsymbol{A}_i^T \left(\sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right)\right\|_2^2 + const, \end{aligned} \quad (13)$$

we rewrite (8) as

$$\begin{aligned} u_i' &= \text{argmin}_{u_i} \left\{\frac{1}{2}\left\|u_i + \boldsymbol{A}_i^T \left(\sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right)\right\|_2^2 + \lambda \|u_i\|_1 - p_i u_i\right\} \\ &= \text{argmin}_{u_i} \left\{\frac{1}{2}\left\|u_i - p_i + \boldsymbol{A}_i^T \left(\sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right)\right\|_2^2 + \lambda \|u_i\|_1\right\} \\ &= \text{shrink}\left(p_i - \boldsymbol{A}_i^T \left(\sum_{j\neq i}^{n} u_j \boldsymbol{A}_j - \boldsymbol{f}\right), \lambda\right) \\ &= \text{shrink}(p_i + u_i - \boldsymbol{A}_i^T (\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}), \lambda). \end{aligned} \quad (14)$$

By substituting Eqns. (11) and (12) into (14), we get:

$$u_i' = \text{shrink}(v_i', \lambda). \quad (15)$$

These iterations appear similar to that in LBI (5, 6), but are performed in a component-wise manner resulting in the following algorithm.

Algorithm 1: Bregman coordinate descent
Initialize: $\boldsymbol{v}$=0, $\boldsymbol{u}$=0
while "$\|\boldsymbol{f} - \boldsymbol{A}\boldsymbol{u}\|_2^2$ not converge" **do**

"choose $i \in \{1:n\}$"

$$v_i \longleftarrow v_i - \boldsymbol{A}_i^T(\boldsymbol{Au} - \boldsymbol{f}), \tag{16}$$

$$u_i \longleftarrow \text{shrink}(v_i, \lambda). \tag{17}$$

end while

In addition to specifying component-wise iterations in Algorithm 1, we must also specify the order in which the components of $\boldsymbol{u}$ are updated. Previous proposal include updating components sequentially based on the index *i* (Friedman et al., 2007, Genkin et al., 2007), randomly, or based on the gradient of the objective function (Li and Osher, 2009). In general, choosing *i* in a distributed architecture requires additional communication between nodes and, therefore, places additional demands on energy consumption and communication bandwidth.

4 Derivation of the Hybrid Distributed Algorithm (HDA)

Here, we present our central contribution, a distributed algorithm for solving (1), which has lower communication bandwidth requirements than the existing ones and does not require additional communication for determining the update order. We name our algorithm Hybrid Distributed Algorithm (HDA) because it combines a gradient-descent-like update of $\boldsymbol{v}$, as in Eq. (5), and a coordinate-descent-like update of u_i, as in Eq. (17). The key to this combination is the quantization of the external variable, arising from replacing the shrinkage operation with thresholding. As a result:

1. Due to quantization of the external variable, communication between nodes requires only low bandwidth and is kept to a minimum.
2. The choice of a component of $\boldsymbol{u}$ to be updated, in the sense of coordinate descent, is computed autonomously by each node.

To reduce bandwidth requirements, instead of communicating the analog variable $\boldsymbol{u}$, HDA nodes communicate a quantized variable $\boldsymbol{s} \in \{-1,0,1\}^n$ to each other. The variable $\boldsymbol{u}$, which solves (1) is obtained from $\boldsymbol{s}$ by averaging it over time: $\boldsymbol{u} = \lambda\bar{\boldsymbol{s}} = \frac{\lambda}{t}\sum_{k=0}^{t}\boldsymbol{s}^k$.
In HDA, components of $\boldsymbol{s}$ are obtained from the internal variable $\boldsymbol{v}$:

$$\boldsymbol{s} \longleftarrow \text{threshold}(\boldsymbol{v}, \lambda), \tag{18}$$

where threshold function is component wise, $\text{threshold}(x,\lambda) = \begin{cases} 1, & \text{if } x > \lambda \\ 0, & \text{if } -\lambda \le x \le \lambda \\ -1, & \text{if } x < -\lambda \end{cases}$.

An update for the internal variable $\boldsymbol{v}$ is similar to (5) but with substitution of $\boldsymbol{u}$ by $\lambda\boldsymbol{s}$:

$$\boldsymbol{v} \longleftarrow \boldsymbol{v} - \boldsymbol{A}^T(\lambda\boldsymbol{As} - \boldsymbol{f}). \tag{19}$$

Note that in HDA there is no need to explicitly specify the order in which the components of $\boldsymbol{u}$ are updated because the threshold operation (18) automatically updates the components in the order they reach threshold. Updates (18, 19) lead to the following computer algorithm.

Algorithm 2: Discrete-time HDA
Initialize: $\boldsymbol{v}$=0, $\boldsymbol{u}$=0, $\boldsymbol{s}$=0, t=0.
while "$\|\boldsymbol{f} - \boldsymbol{Au}\|_2^2$ not converge" **do**
$t \leftarrow t + 1$
$\boldsymbol{v} \leftarrow \boldsymbol{v} - \boldsymbol{A}^T(\lambda\boldsymbol{As} - \boldsymbol{f})$,
$\boldsymbol{s} \leftarrow \text{threshold}(\boldsymbol{v}, \lambda)$,
$\boldsymbol{u} \leftarrow ((t-1)\boldsymbol{u} + \lambda\boldsymbol{s})/t$.
end while
Although not necessary, precomputing $\boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{A}^T\boldsymbol{f}$ may speed up algorithm execution.

To gain some intuition for Algorithm 2 consider an example, where $\boldsymbol{f}$ is chosen to coincide with some column of $\boldsymbol{A}$, i.e. $\boldsymbol{f}=\boldsymbol{A}_i$. Then the solution of problem (1) must be $u_i=1$, $u_{j\neq i}=0$. Now, let us see how the algorithm computes this solution.

The algorithm starts with $\boldsymbol{v}=0$, $\boldsymbol{u}=0$, $\boldsymbol{s}=0$. Initially, each component v_j changes at a rate of $\boldsymbol{A}_j^T\boldsymbol{A}_i$ and, while the i-th component is below the threshold, $\boldsymbol{u}$ stays at 0. Assuming $\lambda \gg 1$, after $\lambda/(\boldsymbol{A}_i^T\boldsymbol{A}_i)=\lambda$ iterations, v_i reaches the threshold λ and s_i switches from 0 to 1. At that time, the other components of $\boldsymbol{v}$ are still below threshold, $\left|v_{j\neq i}\right|=\left|\lambda\boldsymbol{A}_{j\neq i}^T f\right|=\left|\lambda\boldsymbol{A}_{j\neq i}^T\boldsymbol{A}_i\right|<\lambda$ and, therefore the components $s_{j\neq i}$ stay at 0. Note that choosing large λ guarantees that no more than one component reaches the threshold at any iteration.

Knowing $\boldsymbol{s}$, we can compute the next iteration for $\boldsymbol{v}$ (19), which is $\boldsymbol{v}=\lambda\boldsymbol{A}^T\boldsymbol{f}-\boldsymbol{A}^T(\lambda\boldsymbol{A}_i s_i-\boldsymbol{f})=\lambda\boldsymbol{A}^T\boldsymbol{A}_i-\boldsymbol{A}^T\lambda\boldsymbol{A}_i+\boldsymbol{A}^T\boldsymbol{f}=\boldsymbol{A}^T\boldsymbol{f}$. Note that the first and the second terms cancelled because the change in v accumulated over previous λ iterations is canceled by receiving broadcast s_i. Because s_i switches back to 0, $u_i=\lambda\bar{s}_i=\lambda/\lambda=1$ as required. From this point on, the above sequence repeats itself. The above cancellation maintains $s_{j\neq i}=0$ and ensures sparsity of the solution, $u_{j\neq i}=0$.

The HDA updates (18, 19) can be immediately translated into the continuous-time evolution of the physical variables $\boldsymbol{s}(t)$ and $\boldsymbol{v}(t)$ in a hardware implementation.

Continuous-time evolution:

$$\boldsymbol{v}(t)=\int_0^t\boldsymbol{A}^T[\boldsymbol{f}-\lambda\boldsymbol{A}\boldsymbol{s}(t')]dt' \tag{20}$$

$$\boldsymbol{s}(t)=\text{spike}(\boldsymbol{v}(t),\lambda), \tag{21}$$

where the spike function is component wise, $\text{spike}(v_i(t),\lambda)=\begin{cases}\delta(t), & \text{if } v_i(t)=\lambda\\ 0, & \text{if } -\lambda<v_i(t)<\lambda.\\ -\delta(t), & \text{if } v_i(t)=-\lambda\end{cases}$

In this continuous-time evolution, the solution to (1) is given by the scaled temporal average $\boldsymbol{u}(\text{t})=\frac{\lambda}{t}\int_0^t\boldsymbol{s}(t')d\,t'$.

The HDA can be naturally implemented on a neuronal network, Fig 1. Unlike the LCA (Rozell et al., 2008) and the LBI (Yin et al., 2008, Cai et al., 2009a, b), which require neurons continuously communicating graded potentials, the HDA uses perfect, or non-leaky, integrate-and-fire neurons (Koch, 1999, Dayan and Abbott, 2001). Ideal, or non-leaky, integrate-and-fire neurons integrate inputs over time in their membrane voltage, $\boldsymbol{v}$, (20) and fire a unitary action potential (or spike) when the membrane voltage reaches the threshold, λ, (21). The inputs come from the stimulus, $\boldsymbol{A}^T\boldsymbol{f}$, and from other neurons, via the off-diagonal elements of $-\boldsymbol{A}^T\boldsymbol{A}$. After the spike is emitted, the membrane voltage is reset to zero due to the unitary diagonal elements of $\boldsymbol{A}^T\boldsymbol{A}$. We emphasize that, in discrete-time simulations, the membrane potential of HDA integrate-and-fire neurons after spiking is reset by subtracting the threshold magnitude rather than by setting it to zero (Brette et al., 2007).

Unlike thresholding in the HDA nodes (21), in biological neurons, thresholding is one-sided (Koch, 1999, Dayan and Abbott, 2001). Such discrepancy is easily resolved by substituting each node with two opposing (on- and off-) nodes. In fact, neurons in some brain areas are known to come in two types (on- and off-) (Masland, 2001).

Therefore, the HDA can be used as a model of computation with integrate-and-fire neurons. In the next section, we prove that $\boldsymbol{u}$, a time-average of $\boldsymbol{s}$, which can be viewed as a firing rate, converges to a solution of $\boldsymbol{f}=\boldsymbol{A}\boldsymbol{u}$.

Finally, for the sake of completeness, we propose the following "hopping" version of the HDA, which does not reduce energy consumption of communication bandwidth, yet is convenient for fast implementation on the CPU architecture for the sake of modeling.

Algorithm 3: hopping HDA
Initialize: $\boldsymbol{v}$=0, $\boldsymbol{u}$=0, $\boldsymbol{s}$=0, t=0.
While "$\|\boldsymbol{f}-\boldsymbol{A}\boldsymbol{u}\|_2^2$ not converge" **do**
$r=\max|v_i|$,
$j=\mathrm{argmax}_i|v_i|$,
if $r<\lambda$ **then**
$t_w=\min[(\lambda\ \mathrm{sign}(\boldsymbol{A}_i^T\boldsymbol{f})-v_i)/(\boldsymbol{A}_i^T\boldsymbol{f})]$,
$j=\mathrm{argmin}_i[(\lambda\ \mathrm{sign}(\boldsymbol{A}_i^T\boldsymbol{f})-v_i)/(\boldsymbol{A}_i^T\boldsymbol{f})]$,
$t\leftarrow t+t_w$,
$s_j\leftarrow\ \mathrm{sign}(\boldsymbol{A}_j^T\boldsymbol{f})$
$\boldsymbol{v}\leftarrow\boldsymbol{v}+t_w\boldsymbol{A}^T\boldsymbol{f}-\lambda s_j\boldsymbol{A}^T\boldsymbol{A}_j$,
$u_j\leftarrow((t-1)u_j+s_j)/t$,
else
$s_j\leftarrow\ \mathrm{sign}(v_j)$
$\boldsymbol{v}\leftarrow\boldsymbol{v}-\lambda\boldsymbol{A}^T\boldsymbol{A}_j s_j$,
$u_j\leftarrow((t-1)u_j+\lambda s_j)/t$,
end if
end while

As before, precomputing $\boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{A}^T\boldsymbol{f}$ may speed up algorithm execution.

The name "hopping HDA" comes from the fact that, instead of waiting for many iterations to reach the threshold, λ, the algorithm directly determines the component of $\boldsymbol{v}$ which will be the next to reach the threshold and computes the required integration time in t_w. Thus, the idea of hopping is similar to the ideas behind LARS (Efron et al., 2004) and "kicking" (Osher et al., 2010). When that component of v reaches the threshold, it broadcasts $-\boldsymbol{A}^T\boldsymbol{A}$ to other neurons instantaneously. We note that in practice, several nodes may exceed the threshold at the same time. In this case, we update super-threshold components based on the magnitude of v_i starting with the largest.

5 Asymptotic performance guarantees

In this section, we analyze the asymptotic performance of the HDA by proving three theorems. Theorem 1 demonstrates that the HDA can be viewed as taking steps towards the solutions of a sequence of the Lasso problems whose regularizer coefficient decays in the course of iterations. Theorem 2 demonstrates that the representation error decays as *1/t* in the asymptotic limit. Theorem 3 demonstrates that, in the presence of time-varying noise, the representation error in the asymptotic limit decays also as a power of *t*. All the results are proven for the evolution described by Eqns. (20, 21), but can be easily adapted for the discrete-time case.

Importantly, Theorems 1 and 2 together suggest an intuition for why HDA finds a sparse solution. As the solution of a Lasso problem is known to be sparse (Tibshirani, 1996), it may seem possible that solving a sequence of the Lasso problems, as shown in Theorem 1, would yield a sparse solution. Yet, one may argue that, according to Theorem 1, the regularizer coefficient decays in the course of iterations and, because smaller regularization coefficients should yield less sparse solutions, the final outcome may not be sparse. Note, however, that the driving force for the growth of components of $\boldsymbol{u}$ is given by the representation error, which itself shrinks in the course of iterations according to Theorem 2. Because the error decays with the same asymptotic rate as the regularization coefficient we may still expect that the ultimate solution remains sparse. Indeed, such intuition is born out by numerical simulations as will be demonstrated in Section 6.

Theorem 1: Define average external variable at time t as $\bar{\boldsymbol{s}}(t):=\frac{1}{t}\int_0^t\boldsymbol{s}(t')dt'$. Then, provided $\|\bar{\boldsymbol{s}}(t)\|_1\neq 0$, the energy function $E(t):=\|\boldsymbol{f}-\lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2^2+(\lambda^2/t)\|\bar{\boldsymbol{s}}(t)\|_1$ generated by (20,21) decreases monotonically.

Proof: To prove this theorem, we consider separately the change in $E(t)$ during the interval between spikes and the change in $E(t)$ during a spike. We define $\boldsymbol{w} := \int_0^t \boldsymbol{s}(t')dt'$, which does not change during the interval between spikes. Then we replace $\bar{\boldsymbol{s}}(t)$ in $E(t)$ by $\boldsymbol{w}/t$ and obtain after simple algebra:

$$\begin{aligned} dE(t)/dt &= \tfrac{2\lambda}{t^3}\boldsymbol{w}^T\boldsymbol{A}^T(\boldsymbol{f}t - \lambda\boldsymbol{A}\boldsymbol{w}) - \tfrac{2\lambda^2}{t^3}\|\boldsymbol{w}\|_1 \\ &= \tfrac{2\lambda}{t^3}[\boldsymbol{w}^T\boldsymbol{v}(t) - \lambda\|\boldsymbol{w}\|_1] \\ &= \tfrac{2\lambda}{t^3}\textstyle\sum_{i=1}^n [w_i v(t)_i - \lambda|w_i|]. \end{aligned} \tag{22}$$

The second equality follows from Eq. (20). Since $|v(t)_i| < \lambda$, if $\|\boldsymbol{w}\|_1 \neq 0$, $dE(t)/dt < 0$. Therefore, during the interval between spikes, $E(t)$ decreases.

If the *i*-th neuron fires a spike at *t*, $|s(t)_i| = 1$ and $s(t)_{j\neq i} = 0$, then the difference between $E(t)$, just after the spike, and $E(t^-)$, just before the spike is given by (notation t^- means arbitrarily close to *t* from below),

$$\begin{aligned} E(t) - E(t^-) &= \|\boldsymbol{f} - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2^2 + (\lambda^2/t)\|\bar{\boldsymbol{s}}(t)\|_1 - \|\boldsymbol{f} - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t^-)\|_2^2 - (\lambda^2/t)\|\bar{\boldsymbol{s}}(t^-)\|_1 \\ &= \|\boldsymbol{f} - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t^-) - \lambda s(t)_i\boldsymbol{A}_i/t\|_2^2 - \|\boldsymbol{f} - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t^-)\|_2^2 \\ &\quad + \tfrac{\lambda^2}{t}\left(\textstyle\sum_{j\neq i}|\bar{s}(t^-)_j| + |\bar{s}(t)_i|\right) - \tfrac{\lambda^2}{t}\left(\textstyle\sum_{j\neq i}|\bar{s}(t^-)_j| + |\bar{s}(t^-)_i|\right) \\ &= -\tfrac{2\lambda}{t}s(t)_i\boldsymbol{A}_i^T(f - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t^-)) + \tfrac{\lambda^2}{t^2}\|\boldsymbol{A}_i\|_2^2\|s(t)_i\|_2^2 + \tfrac{\lambda^2}{t}(|\bar{s}(t)_i| - |\bar{s}(t^-)_i|) \\ &= -\tfrac{2\lambda}{t^2}s(t)_i v(t^-)_i + \tfrac{\lambda^2}{t^2}\|s(t)_i\|_2^2 + \tfrac{\lambda^2}{t}(|\bar{s}(t^-)_i + s(t)_i/t| - |\bar{s}(t^-)_i|). \\ &= \tfrac{\lambda^2}{t^2}[\|s(t)_i\|_2^2 + |s_i^t|\mathrm{sign}(\bar{s}(t^-)_i s(t)_i) - 2s(t)_i v(t^-)_i/\lambda] \end{aligned} \tag{23}$$

In the above equation, we used the relation $\bar{\boldsymbol{s}}(t) = \bar{\boldsymbol{s}}(t^-) + \boldsymbol{s}(t)/t$, which can be written separately for each component as $\bar{s}(t)_i = \bar{s}(t^-)_i + s(t)_i/t$ and $\bar{s}(t)_{j\neq i} = \bar{s}(t^-)_{j\neq i}$ (because $s(t)_{j\neq i} = 0$). Since $s(t)_i v(t^-)_i \to \lambda$, $\|s(t)_i\|_2^2 = |s(t)_i| = 1$, $E(t) - E(t^-) \to 0$ when $\mathrm{sign}(\bar{s}(t^-)_i s(t)_i) = 1$ and $E(t) - E(t^-) < 0$ when $\mathrm{sign}(\bar{s}(t^-)_i s(t)_i) = -1$. Therefore, at spike time, $E(t)$ does not increase. Combining (22) and (23) concludes the proof.

Similarly, for the discrete-time HDA, Algorithm 2, it is easy to show that, for sufficiently large λ, if $\|\bar{\boldsymbol{s}}(t) := (1/t)\sum_{k=0}^{t-1}\boldsymbol{s}^k\|_1 \neq 0$, the sequence $\{E(t) := \|\boldsymbol{f} - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2^2 + (\lambda^2/t)\|\bar{\boldsymbol{s}}(t)\|_1\}$ generated by Algorithm 2 decreases monotonically.

Theorem 2: There exists an upper bound on the representation error, $\|\boldsymbol{f} - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2$, which decays as O(1/*t*).

Proof: In the continuous-time evolution, $v(t)_i = \boldsymbol{A}_i^T\int_0^t[\boldsymbol{f} - \lambda\boldsymbol{A}\boldsymbol{s}(t')]dt'$. Because of the threshold operation, $|v(t)_i| \leq \lambda$ and, therefore,

$$\left|A_i^T\int_0^t[\boldsymbol{f} - \lambda\boldsymbol{A}\boldsymbol{s}(t')]dt'\right| \leq \lambda. \tag{24}$$

Then, assuming that $\boldsymbol{A}$ has full row rank, $\left\|\int_0^t[\boldsymbol{f} - \lambda\boldsymbol{A}\boldsymbol{s}(t')]dt'\right\|_2$ must be also bounded from above. Then, the representation error can be expressed as:

$$\|\boldsymbol{f} - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2^2 = \tfrac{1}{t^2}\left\|\int_0^t[\boldsymbol{f} - \lambda\boldsymbol{A}\boldsymbol{s}(t')]dt'\right\|_2^2 \leq \tfrac{\mathrm{const}}{t^2}. \tag{25}$$

Therefore, $\|\boldsymbol{f} - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2 \leq \frac{const}{t}$, which concludes the proof.

Similar proof can be given for the discrete-time HDA, although with a different constant.

Theorem 3: Assume the signal $\boldsymbol{f}$ is subject to time varying noise, i.e. $\boldsymbol{f}(t) = \boldsymbol{f}^0 + \boldsymbol{\varepsilon}(t)$. If

$\left\|\int_0^t \boldsymbol{\varepsilon}(t')dt'\right\|_2^2 = O(t^{\alpha<1})$, then $\lim_{t\to\infty}\|\boldsymbol{f}^0 - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2 = 0$ and there exist some upper bound of $\|\boldsymbol{f}^0 - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2$, which decays as $t^{-\min(1,1-\alpha)}$.

Proof: Because of the threshold operation, $\boldsymbol{v}$ is bounded from above:

$$\|\boldsymbol{f}^0 - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2^2 = \frac{1}{t^2}\left\|\int_0^t[\boldsymbol{f}(t') - \boldsymbol{\varepsilon}(t') - \lambda\boldsymbol{A}\boldsymbol{s}(t')]dt'\right\|_2^2 \quad (26)$$

$$= \frac{1}{t^2}\left\|\int_0^t[\boldsymbol{f}(t') - \lambda\boldsymbol{A}\boldsymbol{s}(t')]dt'\right\|_2^2 - \frac{2}{t^2}\langle\int_0^t \boldsymbol{\varepsilon}(t')dt', \int_0^t[\boldsymbol{f}(t') - \lambda\boldsymbol{A}\boldsymbol{s}(t')]dt'\rangle + \frac{2}{t^2}\left\|\int_0^t \boldsymbol{\varepsilon}(t')dt'\right\|_2^2.$$

Using again the fact that $|\boldsymbol{v}(t)_i| = \left|\boldsymbol{A}_i^T\int_0^t[\boldsymbol{f}(t') - \lambda\boldsymbol{A}\boldsymbol{s}(t')]dt'\right|$ is bounded from above, we obtain

$$\|\boldsymbol{f}^0 - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2^2 \le \frac{C^2}{t^2} - \frac{2C}{t^2}\left\|\int_0^t \boldsymbol{\varepsilon}(t')dt'\right\|_2 + \frac{2}{t^2}\left\|\int_0^t \boldsymbol{\varepsilon}(t')dt'\right\|_2^2 = O\left(t^{-\min(2,2-2\alpha)}\right). \quad (27)$$

This concludes the proof. Next, we consider several examples of noise.

In the case of $\boldsymbol{f}$ contaminated by the white noise, $\int_0^t \boldsymbol{\varepsilon}(t')dt' = O(\sqrt{t})$, and the representation error converges as $1/\sqrt{t}$.

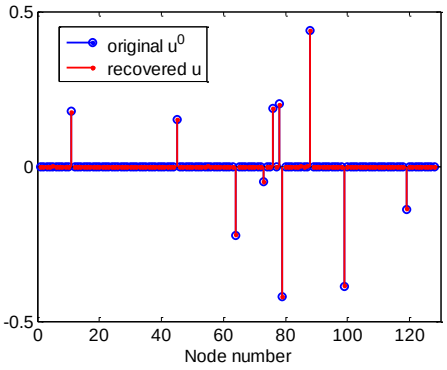
In the case of static noise where $\boldsymbol{\varepsilon}(t) = \boldsymbol{\varepsilon}$, we obtain:

$$\|\boldsymbol{f}^0 - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}(t)\|_2^2 \le \frac{C^2}{t^2} - \frac{2C}{t}\|\boldsymbol{\varepsilon}\|_2 + 2\|\boldsymbol{\varepsilon}\|_2^2, \quad (28)$$

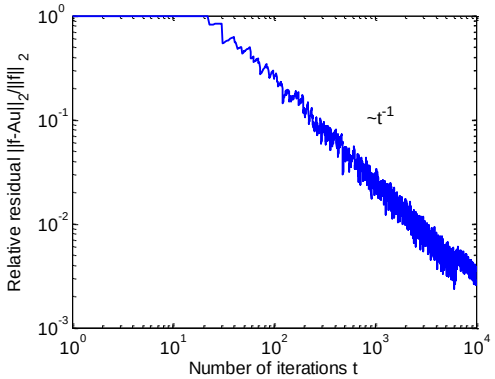
which can be used as a stopping criterion in a de-noising application to prevent over-fitting.

6 Numerical results

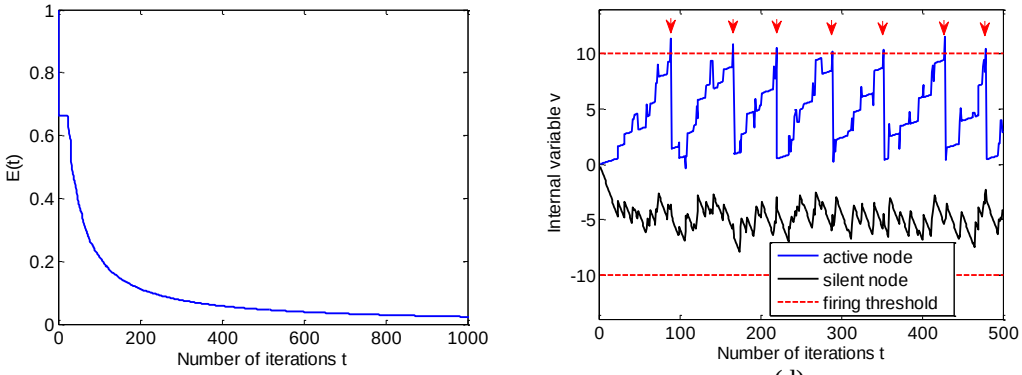
In this section, we report the results of numerical experiments. In the first experiment, we search for sparse representation (1) of synthesized data using HDA. The elements of the matrix $\boldsymbol{A} \in \mathbb{R}^{64\times128}$ are chosen from a normal distribution and column-normalized by dividing each element by the l_2 norm of its column. For the noiseless case, we construct vector $\boldsymbol{f}$ as $\boldsymbol{A}\boldsymbol{u}^0$, where $\boldsymbol{u}^0 \in \mathbb{R}^{128}$ is generated by randomly selecting $nz = 10$ locations for non-zero entries sampled from a flat distribution between -0.5 and 0.5. Then, we apply the discrete-time HDA (Algorithm 2) using the network (Fig. 1) with 128 nodes. We set the spiking threshold λ=10 and obtain a solution, $\boldsymbol{u} = \lambda\bar{\boldsymbol{s}}$, which is compared with $\boldsymbol{u}^0$, Fig. 2.



(a)



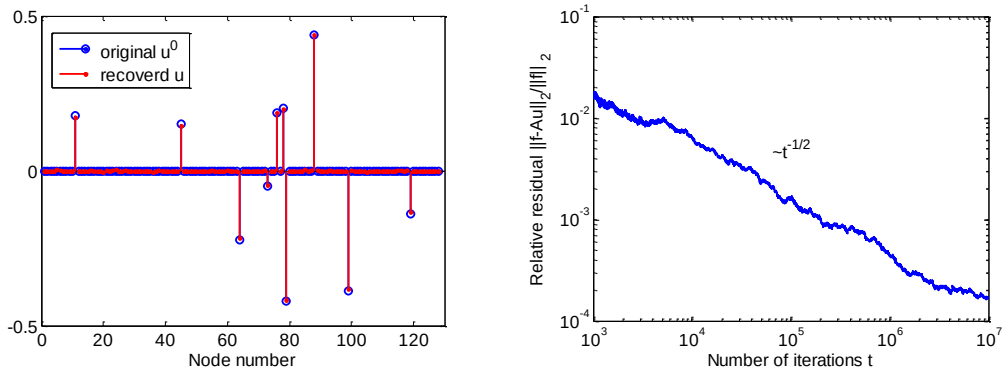
(b)



(c) (d)

Figure 2: Computing sparse representation, $\boldsymbol{u}$, from noiseless $\boldsymbol{f} = \boldsymbol{A}\boldsymbol{u}^0$ using HDA. (a) The reconstructed $\boldsymbol{u} = \lambda\bar{\boldsymbol{s}}$ (stemmed red dots) at t = 10000 coincides with the original $\boldsymbol{u}^0$ (blue circles). (b) The relative residual $\|\boldsymbol{f} - \lambda\boldsymbol{A}\bar{\boldsymbol{s}}\|_2/\|\boldsymbol{f}\|_2$ decays as *1/t* (note log-scale axes) in agreement with the upper bound (Theorem 2). The wiggles are due to the discreteness of $\boldsymbol{s}$. (c) Energy, E^t, as defined in Theorem 5.1 decays monotonically. (d) Representative evolution of internal variable, $\boldsymbol{v}$, of a broadcasting node (blue) and a silent node (black). Red arrows indicate time points when the component of $\boldsymbol{s}$ corresponding to the broadcasting node is non-zero. The firing thresholds (for λ=10) are shown by dashed red lines.

As hardware implementations of HDA or neural circuits must operate on the incoming signal $\boldsymbol{f}$ contaminated by noise, which varies during the iterative computation, we analyze the performance of HDA in the presence of noise. To model such a situation we add time varying Gaussian white noise to the original signal $\boldsymbol{f}^0 = \boldsymbol{A}\boldsymbol{u}^0$. On each iteration step, we set each component $f_i^k = f_i^0\left(1 + 0.5\varepsilon_i^k\right)$, where the noise ε_i^k is independently picked from a normal distribution, $N(0,1)$. We found that, despite such a low signal-to-noise ratio, the HDA yields $\boldsymbol{u}$, which is close to the original $\boldsymbol{u}^0$, Fig. 3a. The relative residual decays as $1/\sqrt{t}$, Fig. 4b, as expected from $\sum_{k=1}^{t}\varepsilon^k = O(\sqrt{t})$.

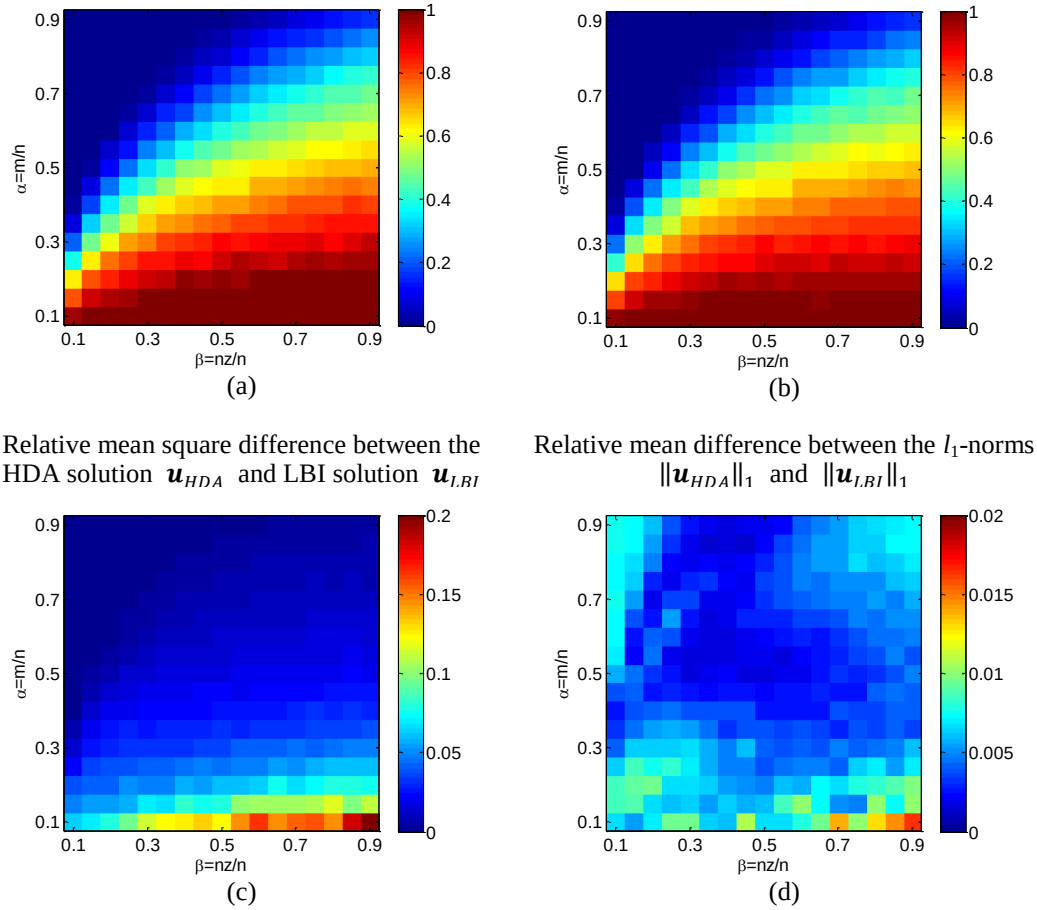


(a) (b)

Figure 3: The HDA is robust to noise in the input. Computing sparse representation on the same dataset as Figure 2 but contaminated by strong time-varying noise.

Relative mean square difference between the HDA solution $\boldsymbol{u}_{HDA}$ and $\boldsymbol{u}^0$

Relative mean square difference between the LBI solution $\boldsymbol{u}_{LBI}$ and $\boldsymbol{u}^0$



(a) Relative mean square difference between the HDA solution $\boldsymbol{u}_{HDA}$ and LBI solution $\boldsymbol{u}_{LBI}$

(b) Relative mean difference between the l_1-norms $\|\boldsymbol{u}_{HDA}\|_1$ and $\|\boldsymbol{u}_{LBI}\|_1$

(c)

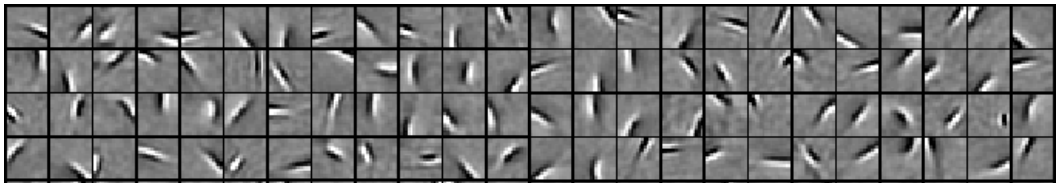
(d)

Figure 4: For a wide range of parameters the HDA solution, $\boldsymbol{u}_{HDA}$, is on par with that of the LBI, $\boldsymbol{u}_{LBI}$. The relative mean square difference between $\boldsymbol{u}_{HDA}$ and the predefined sparse signal $\boldsymbol{u}^0$ (a) and the relative mean square difference between $\boldsymbol{u}_{LBI}$ and $\boldsymbol{u}_0$ (b) demonstrate both HDA and LBI both find the unique solution to the basis pursuit problem (1) when it exists (upper left corner). Indeed, the solutions $\boldsymbol{u}_{HDA}$ and $\boldsymbol{u}_{LBI}$are essentially identical (c) and have the same l_1-norms $\|\boldsymbol{u}_{HDA}\|_1$ and $\|\boldsymbol{u}_{LBI}\|_1$ (d).

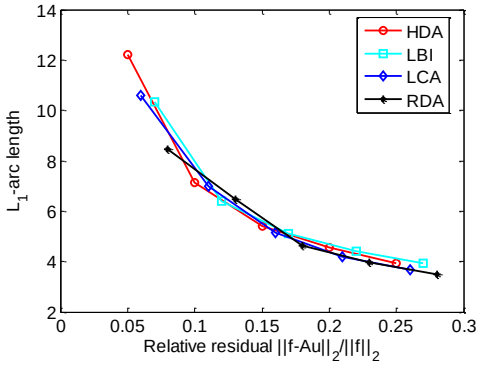
Next we explore the performance of HDA relative to that of the LBI for a wide range of parameters. We present the results as a function of two variables: system indeterminacy $\alpha = m/n$ and system sparsity $\beta = nz/n$ (Charles et al., 2011), Fig. 4. We pick $n = 200$ and vary (α, β) in the range between 0.1 and 0.9. For each pair (α, β), we calculate the corresponding (m, nz) and sample 50 different realizations of the over-complete dictionary $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and the sparse signal $\boldsymbol{u}^0 \in \mathbb{R}^{200}$ satisfying $\|\boldsymbol{u}^0\|_0 = nz$. We then use HDA and LBI to calculate the corresponding sparse solutions $\boldsymbol{u}_{HDA}$ and $\boldsymbol{u}_{LBI}$. We compare the solution of each algorithm to $\boldsymbol{u}^0$ and plot the relative mean square error $\|\boldsymbol{u}_{HDA/LBI} - \boldsymbol{u}_0\|_2^2 / \|\boldsymbol{u}_0\|_2^2$ in Fig. 4a and b. When the system is sufficiently sparse (small β) and determinate (large α), upper left corners of Fig. 4a and b, the solution to the basis pursuit problem (1) is unique and $\boldsymbol{u}_0$ is perfectly recovered (Chen et al., 1998). Under such condition, the solution of HDA is essentially identical to that of LBI as demonstrated in Fig. 4c, which shows the relative mean square difference between the HDA and the LBI solutions $\|\boldsymbol{u}_{HDA} - \boldsymbol{u}_{LBI}\|_2^2 / \|\boldsymbol{u}_{LBI}\|_2^2$. When β gets larger and α gets smaller, the recovery is poor for both algorithms because the predefined $\boldsymbol{u}_0$ is not necessarily the solution with minimum l_1-norm and the solution to (1) is not unique (Chen et al., 1998). Therefore the sparse solutions found by HDA and LBI can be very different as revealed by the large difference in the bottom right corner of Fig. 3c, but they still have near identical l_1-norms, Fig. 4d. We

calculate the relative mean difference between the l_1-norms as $\mathrm{abs}(\|\boldsymbol{u}_{LBI}\|_1 - \|\boldsymbol{u}_{HDA}\|_1)/\|\boldsymbol{u}_{LB}\|_1$ and find that the difference averaged over all points in Fig. 4d is only 5×10^{-3}.

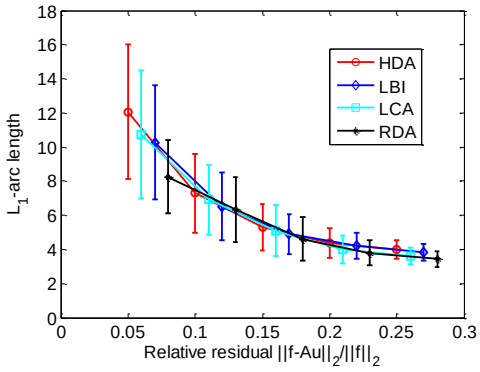
To demonstrate that HDA also serves as model of neural computation, we test it with biologically relevant inputs and dictionary. We use SPAMS (Mairal et al., 2010) to train a four times over complete dictionary with 1024 elements from 16×16 image patches randomly sampled form whitened natural images (Olshausen and Field, 1996). These image patches are further processed by subtracting the mean and normalizing contrast by setting variance to unity. The resulting dictionary elements have spatial properties resembling those of V1 receptive fields, Fig. 5a, (Olshausen and Field, 1996). Then we create a test data set containing 1000 image patches prepared in the same fashion as training image patches. We decompose these image patches using HDA over the learned dictionary and record the mean l_1-arc length of the representation coefficients $\|\boldsymbol{u}\|_1$ at various stopping relative residual. As a comparison we also simulate the decompositions using LBI, LCA and RDA. We found that HDA achieves similar representation error – sparsity tradeoff, Fig. 5b and c.



(a)



(b)



(c)

Figure 5: HDA achieves error – sparsity tradeoff comparable with LBI, LCA and RDA. (a) Representative dictionary elements learned from whitened natural image patches. (b) Tradeoff for a typical natural image patch and (c) Mean tradeoff for an ensemble of 1000 contrast normalized image patches.

7 Summary

In this paper, we propose an algorithm called HDA, which computes sparse redundant representation using a network of simple nodes communicating using punctuate spikes. Compared to the existing distributed algorithms such as LCA and RDA, the HDA has lower energy consumption and demands on the communication bandwidth. Also, HDA is robust to noise in the input signal. Therefore, HDA is a highly promising algorithm for hardware implementations for energy constrained applications.

We propose three implementations of the HDA: a discrete-time HDA (Algorithm 2), a continuous-time evolution of the physical variable in a hardware implementation, and a hopping HDA (Algorithm 3) for fast computation on a CPU architecture.

Finally, HDA operation combines analog and digital steps (Sarpeshkar, 1998) and is equivalent to a network of non-leaky integrate-and-fire neurons suggesting that it can be used as a model for

neural computation.

References

Attwell D, Laughlin SB (2001) An energy budget for signaling in the grey matter of the brain. J Cereb Blood Flow Metab 21:1133-1145.

Baraniuk RG (2007) Compressive sensing. Ieee Signal Proc Mag 24:118-124.

Bertsekas DP (2009) Convex optimization theory. Belmont, MA: Athena Scientific.

Boerlin M, Deneve S (2011) Spike-Based Population Coding and Working Memory. Plos Comput Biol 7.

Boyd S, Vandenberghe L (2004) Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press.

Bregman LM (1967) The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Computational Mathematics and Mathematical Physics 7:200-217.

Brette R, Rudolph M, Carnevale T, Hines M, Beeman D, Bower JM, Diesmann M, Morrison A, Goodman PH, Harris FC, Zirpe M, Natschlager T, Pecevski D, Ermentrout B, Djurfeldt M, Lansner A, Rochel O, Vieville T, Muller E, Davison AP, El Boustani S, Destexhe A (2007) Simulation of networks of spiking neurons: A review of tools and strategies. J Comput Neurosci 23:349-398.

Cai JF, Osher S, Shen ZW (2009a) Convergence of the Linearized Bregman Iteration for L(1)-Norm Minimization. Mathematics of Computation 78:2127-2136.

Cai JF, Osher S, Shen ZW (2009b) Linearized Bregman Iterations for Compressed Sensing. Mathematics of Computation 78:1515-1536.

Charles AS, Garrigues P, Rozell CJ (2011) Analog Sparse Approximation with Applications to Compressed Sensing. arXiv:11114118.

Chen SSB, Donoho DL, Saunders MA (1998) Atomic decomposition by basis pursuit. Siam Journal on Scientific Computing 20:33-61.

Chklovskii DB, Schikorski T, Stevens CF (2002) Wiring optimization in cortical circuits. Neuron 34:341-347.

Dattorro J (2008) Convex Optimization & Euclidean Distance Geometry. Palo Alto, CA: Meboo Publishing.

Dayan P, Abbott LF (2001) Theoretical Neuroscience. Computational and Mathematical Modeling of Neural System. Cambridge, MA: MIT Press.

Deneve S, Boerlin M (2011) Implementing Dynamical systems with spiking neurons. In: COSYNE.

DeWeese MR, Wehr M, Zador AM (2003) Binary spiking in auditory cortex. J Neurosci 23:7940-7949.

Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Annals of Statistics 32:407-451.

Elad M, Matalon B, Shtok J, Zibulevsky M (2007) Wide-angle view at iterated shrinkage algorithms - art. no. 670102. P Soc Photo-Opt Ins 6701:70102-70102.

Friedman J, Hastie T, Hofling H, Tibshirani R (2007) Pathwise Coordinate Optimization. Annals of Applied Statistics 1:302-332.

Gallant JL, Vinje WE (2000) Sparse coding and decorrelation in primary visual cortex during natural vision. Science 287:1273-1276.

Genkin A, Lewis DD, Madigan D (2007) Large-scale Bayesian logistic regression for text categorization. Technometrics 49:291-304.

Kavukcuoglu K, Sermanet P, Boureau Y, Gregor K, Mathieu M, LeCun Y (2010) Learning Convolutional Feature Hierachies for Visual Recognition. Advances in Neural Information Processing Systems.

Koch C (1999) Biophysics of Computation. New York: Oxford University Press.

Laughlin SB, Sejnowski TJ (2003) Communication in neuronal networks. Science 301:1870-1874.

Lennie P (2003) The cost of cortical computation. Curr Biol 13:493-497.

Li YY, Osher S (2009) COORDINATE DESCENT OPTIMIZATION FOR l(1) MINIMIZATION WITH APPLICATION TO COMPRESSED SENSING; A GREEDY ALGORITHM. Inverse Problems and Imaging 3:487-503.
Mairal J, Bach F, Ponce J, Sapiro G (2010) Online Learning for Matrix Factorization and Sparse Coding. J Mach Learn Res 11:19-60.
Masland RH (2001) The fundamental plan of the retina. Nat Neurosci 4:877-886.
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 (2004) Sparse coding of sensory inputs. Curr Opin Neurobiol 14:481-487.
Osher S, Mao Y, Dong B, Yin WT (2010) Fast Linearized Bregman Iteration for Compressive Sensing and Sparse Denoising. Commun Math Sci 8:93-111.
Pearl J (1988) Embracing Causality in Default Reasoning. Artificial Intelligence 35:259-271.
Rozell CJ, Johnson DH, Baraniuk RG, Olshausen BA (2008) Sparse coding via thresholding and local competition in neural circuits. Neural Comput 20:2526-2563.
Sarpeshkar R (1998) Analog versus digital: extrapolating from electronics to neurobiology. Neural Comput 10:1601-1638.
Shapero S, Brüderle D, Hasler P, Rozell C (2011) Sparse approximation on a network of locally competitive integrate and fire neurons. In: Cosyne.
Tibshirani R (1996) Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society Series B-Methodological 58:267-288.
Tubaishat M, Madria S (2003) Sensor Networks: An Overview. IEEE Potentials 22.
Xiao L (2010) Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization. J Mach Learn Res 11:2543-2596.
Yin WT, Osher S, Goldfarb D, Darbon J (2008) Bregman Iterative Algorithms for l(1)-Minimization with Applications to Compressed Sensing. Siam Journal on Imaging Sciences 1:143-168.
Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society Series B-Statistical Methodology 67:301-320.

NON-LINEAR PREDICTIVE CODING AS A MODEL OF EARLY SENSORY PROCESSING

Shaul Druckmann*, Tao Hu* and Dmitri B. Chklovskii

Janelia Farm Research Campus

*** - Equal contribution**

Abstract

Early stages of sensory systems face the challenge of compressing information from a large number of receptors onto a much smaller number of projection neurons, a so called communication bottleneck. To make more efficient use of limited bandwidth, compression may be achieved using predictive coding, whereby predictable, or redundant, components of the stimulus are removed. In the case of the retina, Srinivasan et al. (1982) suggested that feedforward subtraction of a linear prediction generated from nearby receptors implements such compression, resulting in biphasic center-surround receptive fields. However, inhibition often operates in a feedback manner and with non-linear input output transformations, considerably complicating the dynamics of such circuits. Here, we solve the transient *non-linear* recurrent dynamics of a generic early sensory circuit in response to a step-like stimulus. We then demonstrate that the non-linearity improves compression. We show that interneuron activity in time constructs progressively less sparse but more accurate representations of the stimulus, thus providing a powerful theoretical framework to understand the dynamics of early sensory processing in a variety of physiological experiments. More generally, our results demonstrate that highly non-trivial computations, at the forefront of modern signal processing, can be mapped onto a concrete neuronal circuit.

Introduction

Receptor neurons in early sensory systems are more numerous than the projection neurons that transmit sensory information to higher brain areas, implying that sensory signals must be compressed to pass through a limited bandwidth channel known as “Barlow’s bottleneck” (Attneave, 1954; Barlow and Levick, 1976). Since natural signals arise from physical objects, which are contiguous in space and time, they are highly spatially and temporally correlated (Dong and Atick, 1995a; Field, 1987; Ruderman and Bialek, 1994). Such signals are ideally suited for predictive coding, a compression strategy borrowed from engineering (Elias, 1955) whereby redundant, or predictable components of the signal are subtracted and only the residual is transmitted (Srinivasan et al., 1982). For highly correlated signals, compression using predictive coding results in a significant dynamic range reduction (Huang and Rao, 2011; Srinivasan et al., 1982).

Consider, for example, the processing of natural images in the retina. Instead of transmitting photoreceptor signals, which are highly correlated in space and time, ganglion cells can transmit differences in signal between nearby pixels or consecutive time points. Indeed, the well known center surround spatial receptive fields or biphasic temporal receptive fields of ganglion cells (Victor, 1999) may be viewed as evidence of predictive coding because they effectively code

such differences (Atick, 1992; Hosoya et al., 2005; Huang and Rao, 2011; Laughlin, 1981; Meister and Berry, 1999; Nirenberg et al., 2010; Srinivasan et al., 1982). In a related line of research optimal spatio-temporal linear filters were derived from the principle of maximizing information transmission (Atick et al., 1992; Atick and Redlich, 1990; Dong and Atick, 1995b; van Hateren, 1992).

How is predictive coding implemented on the circuit level? The seminal work (Srinivasan et al., 1982), which introduced predictive coding to neuroscience, proposed that a subtraction, via feedforward inhibition, of a linear prediction generated from nearby receptors could implement predictive coding. Although their model captures the essence of predictive coding, it does not reflect two important biological facts. First, in the retina, and other early sensory systems, inhibition has a significant feedback component (Masland, 2001; Olsen et al., 2010; Shepherd et al., 2007). Second, interneuron transfer functions are often non-linear (Arevian et al., 2008; Baccus, 2007; Rieke and Schwartz, 2011).

Here, we consider a simplified circuit that captures the key features of sensory processing in early sensory systems, such as retina and olfactory bulb in vertebrates or antennal lobes in invertebrates (Figure 1c). The circuit contains the two major classes of neurons present in such systems: principal neurons, corresponding to bipolar cells, mitral cells, or projection neurons, respectively, which receive input from the sensory receptors and project to higher brain areas; secondly interneurons, representing amacrine cells, granule cells, or local interneurons, which perform local processing. To keep the mathematical analysis tractable we assume a simplified connectivity of reciprocal connections between interneurons and principal neurons, as found in the olfactory bulb (Shepherd et al., 2007) and studied by (Koulakov and Rinberg, 2011) (Figure 1c). Although the steady state activity of such networks for non-linear interneuron transfer functions has been addressed (Dayan, 1999, Koulakov & Rinberg 2011, Lee & Seung 1997, Olshausen & Field 1997), the presence of the non-linearity precluded solving network dynamics in the transient regime.

By taking advantage of recent developments in applied mathematics and signal processing we are able to solve the non-linear recurrent dynamics of such a circuit, for an arbitrary number of sensory channels and interneurons. Using the techniques developed for sparse redundant representations we compute neuronal activity in response to an arbitrary step-stimulus. Moreover, we calculate how reconstruction error decays in time and demonstrate that threshold-linear neurons are superior to both linear neurons and direct transmission. Lastly, our theory makes non-trivial predictions for the different dynamics of early sensory circuits in response to more or less naturalistic stimuli.

Results

Linear predictive coding in a single channel

We start by considering predictive coding in a single channel, where a principal neuron is reciprocally connected with an inhibitory interneuron forming a negative feedback loop (Figure

1a), (Shapley and Victor, 1978; Wilson, 1999). Initially, we assume that both neurons are linear first-order elements (see Methods). To transmit faithfully fast changes in the stimulus, the principal neuron's time constant (the ratio of membrane capacitance and conductance) will be set to zero. For simplicity of exposition we assume that the interneuron time constant is infinite. Denoting the sensory stimulus by *s(t)*, the activity of the principal neuron as *p(t)*, the interneuron activity, *n(t)*,we can express the dynamics as:

$$\begin{cases} p(t) = s(t) - wn(t) \\ \delta \frac{dn(t)}{dt} = wp(t), \end{cases} \tag{1}$$

where *w* in the second equation is the weight of the synapse from the principal neuron to the interneuron. For the sake of simplicity, we assumed that the weight of the synapse from the interneuron to the principal neuron is the same in magnitude but with negative sign, -*w*. The time constant, δ, is the ratio of interneuron membrane capacitance and synaptic conductance, characterizing the rate of the interneuron response (see Methods).

To guarantee that the signal coded by the negative feedback loop is potentially decodable, the prediction made by the interneuron must be strictly causal. In other words, there must be a delay between the input to the interneuron, *p(t)*, and the output of the interneuron, *n(t+e)*. Given that feedback requires signals passing through a synapse, such delay is biologically plausible. When discussing analytical solutions below, we assume that $e \to 0$ to avoid clutter.

In response to a step stimulus, $s(t) = \theta(t)s$, where $\theta(t) = \begin{cases} 0, t < 0 \\ 1, t \geq 0 \end{cases}$ the dynamics in such a negative feedback loop is (Figure 1a):

$$\begin{cases} n(t) = \frac{s}{w}\,\theta(t)\left(1 - \exp\left(-w^2\frac{t}{\delta}\right)\right) \\ p(t) = s\,\theta(t)\exp\left(-w^2\frac{t}{\delta}\right) \end{cases}. \tag{2}$$

The interneuron's activity, *n(t)*, grows with time as it integrates the output of the principal neuron, *p(t)*. In turn, the principal neuron's output, *p(t)*, is the difference between the incoming stimulus and the interneuron's activity, *n(t)*, i.e. a residual, which decays with time from the onset of the stimulus. In the limit considered here (infinite interneuron time constant), the interneuron's feedback will approach the incoming stimulus and the residual will decay to zero. To summarize, in the predictive coding framework one can view the interneuron's activity as a series of progressively more accurate predictions of the stimulus. The principal neuron subtracts these predictions and sends the series of residuals to higher brain areas.

The transient response to a step stimulus (Figure 1b) is consistent with electrophysiological measurements from principal neurons in invertebrate and vertebrate retina (Laughlin 1981, Shapley & Victor 1978). For example, in flies, cells post-synaptic to photoreceptors (the LMCs) have graded potential response (Laughlin et al., 1987) consistent with Equation 2. In the vertebrate retina, most recordings are performed on ganglion cells, which read out signals from bipolar cells. In response to a step-stimulus the firing rate of ganglion cells has the form

consistent with Equation 2 (Shapley & Victor 1978). Determining whether negative feedback originates from intra-cellular or circuit mechanisms is difficult in the single-channel case, (Weckstrom and Laughlin, 2010). Such a distinction is simpler in the multi-channel case considered in the next sub-section.

Linear predictive coding in multiple channels

In most sensory systems, stimuli are transmitted along multiple parallel sensory channels, such as mitral cells in the olfactory bulb, or bipolar cells in the retina. Although a multiple channel circuit could implement predictive coding by replicating the negative feedback loop in each channel, this solution is likely suboptimal. Indeed, due to the contiguous nature of physical objects in space, stimuli are often correlated across different channels. Therefore, interneurons that combine inputs across channels may generate an accurate prediction more rapidly. Subtracting predictions generated by such shared interneurons would result in a lower residual, hence reducing the bandwidth requirements even further (Figure 2).

The dynamics of a multi-channel negative feedback loop in response to a multi-dimensional stimulus can be described similarly to that of a single channel (Equation 1):

$$\begin{cases} \boldsymbol{p} = \boldsymbol{s} - \boldsymbol{W}\boldsymbol{n} \\ \delta \frac{d\boldsymbol{n}}{dt} = \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{p} \end{cases}, \tag{3}$$

where boldface lowercase letters are column vectors representing input stimulus, $\boldsymbol{s} = (s_1, s_2, s_3, \dots)^T$, activity of principal neurons, $\boldsymbol{p} = (p_1, p_2, p_3, \dots)^T$, and interneurons, $\boldsymbol{n} = (n_1, n_2, n_3, \dots)^T$ (superscript *T* stands for matrix transpose), Figure 2a. Boldface uppercase letters designate synaptic weight matrices. Synaptic weights from principal neurons to interneurons are $\boldsymbol{W}^T$, and synaptic weights from interneurons to principal neurons are, again for simplicity, the same in magnitude but with the negative sign, $-\boldsymbol{W}$. Each column of $\boldsymbol{W}$ contains the weights of synapses from correlated principal neurons to a given interneuron, thus defining that interneuron's receptive field (Figure 2b). To avoid clutter, we do not explicitly indicate the time dependence of the vectors $\boldsymbol{p}$, $\boldsymbol{s}$, and $\boldsymbol{n}$.

Linear dynamics of the multi-channel feedback loop in response to a multi-dimensional step stimulus can be solved similarly to that of a single channel:

$$\begin{cases} \boldsymbol{n} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\left(1 - \exp\left(-\boldsymbol{W}^T\boldsymbol{W}\frac{t}{\delta}\right)\right)\boldsymbol{W}^T\boldsymbol{s} \\ \boldsymbol{p} = \left[\mathbf{1} - \boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{W})^{-1}\left(1 - \exp\left(-\boldsymbol{W}^T\boldsymbol{W}\frac{t}{\delta}\right)\right)\boldsymbol{W}^T\right]\boldsymbol{s} \end{cases}, \tag{4}$$

provided $\boldsymbol{W}^T\boldsymbol{W}$ is invertible. When the matrix $\boldsymbol{W}^T\boldsymbol{W}$ is not full rank, for instance if the number of interneurons exceeds the number of sensory channels, the solution of Equation 4 is given by a different expression (see Methods).

Just as in the single-channel case, the circuit constructs a series of progressively more accurate stimulus predictions, $\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{n}$, which reduce the transmitted residual, $\boldsymbol{p} = \boldsymbol{s} - \hat{\boldsymbol{s}}$ (Lee & Seung 1997, Olshausen & Field 1997, Rao & Ballard 1999) (Figure 2c,d). Therefore, the multi-channel feedback loop reduces bandwidth requirements for transmitting the steady state residual.

Can the steady state interneuron representation fully cancel multi-dimensional sensory stimuli? If the interneuron receptive fields fully span the space defined by the stimulus ensemble, their representation of the stimuli, and hence cancellation of activity, can be perfect. Otherwise, the interneuron representation minimizes the residual for given receptive fields and the cancellation is only partial.

A biological example of linear predictive coding in multiple channels may be the photoreceptor-horizontal cell circuit in the vertebrate retina (Kamermans and Spekreijse, 1999).

Non-linear predictive coding in multiple channels

Solving the circuit dynamics in the previous sub-section relied on the assumption that neurons act as linear elements, which in view of various non-linearities exhibited by real neurons, represents a drastic simplification. The most obvious kind of nonlinearity arises from the fact that spiking neurons cannot have negative firing rate, or that graded potential neurons cannot have a negative synaptic vesicle release rate. Yet, such a limitation can be dealt with by substituting a pair of on- and off- neurons with identical, but sign reversed, receptive fields in place of a single linear neuron. Namely, the on- neuron could encode the positive half of the linear transfer function and zero in the negative half, whereas (positive) activity in the off- neuron encodes the negative half of the linear function. Thus, in the vertebrate retina, pairs of on- and off- bipolar cells can mimic a linear neuron.

Another response non-linearity, which is often found in interneurons (Arevian et al 2008), is the existence of a non-zero input threshold below which neurons do not fire. A pair of such on- and off- neurons is described by the following threshold function, Figure 2e:

$$\mathrm{Thresh}(n) = \begin{cases} n - \lambda, n > \lambda \\ 0, |n| \le \lambda \\ n + \lambda, n < -\lambda \end{cases}, \tag{5}$$

which has a "gap" or "deadzone" around zero activity and is not equivalent to a linear neuron. The feedback circuit dynamics is given by the following non-linear equations (see Methods, Eq.S3),

$$\begin{cases} \boldsymbol{p} = \boldsymbol{s} - \boldsymbol{W}\boldsymbol{a} \\ \delta \frac{d\boldsymbol{n}}{dt} = \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{p} \\ \boldsymbol{a} = \mathrm{Thresh}(\boldsymbol{n}) \end{cases}, \tag{6}$$

where vector thresholding is performed in a component-wise manner. The vector $\boldsymbol{a}$ stands for the external activity of interneurons, which affects their post-synaptic neurons, such as firing rate or

synaptic release, while $\boldsymbol{n}$ represents internal activity of interneurons, given by the membrane potential. Due to the relative linearity of principal neurons (Arevian et al., 2008; Matthews and Fuchs, 2010), we do not differentiate between their external and internal activity. As before, these equations are derived in the limit of zero time constant for the principal neurons and infinite time constant for the interneurons (see Methods).

In response to a presentation of a step stimulus, the reciprocal circuit dynamics goes through a transient phase and settles at a steady state. The steady state solutions for the non-linear dynamics in a multi-channel negative feedback loop have been addressed previously (Koulakov & Rinberg 2010, Lee & Seung 1997, Olshausen & Field 1997), albeit using somewhat different equations (see Discussion). If the interneuron receptive fields span the stimulus space, their steady-state activity vector cancels out the stimulus fully and reduces the residual to zero.

Our proposal to view circuit dynamics from the predictive coding perspective leads to the following two related insights. First, previous observations that the circuit dynamics constructs stimulus representation within local interneurons, which do not project directly to higher brain areas, seemed paradoxical and requiring additional explanation (Koulakov & Rinberg 2011). Predictive coding accounts for this naturally because it calls for removing predictable components from the input and transmitting only the residual. The representation of the interneurons is meant therefore not to be transmitted but to serve as a prediction that cancels out activity.

Second, if the interneuron receptive fields cover the space of stimuli they will cancel out the stimulus and the steady state of the circuit will be at zero activity of principal neurons, which seems to present a problem for decoding. Indeed, to prevent full cancellation, Koulakov and Rinberg, 2011, suggested that the interneuron receptive fields do not span the stimulus space fully and the stimulus cancelation is only partial. In contrast, in the predictive coding framework, full cancelation of the sensory stimulus in steady state is not an issue: a representation of the stimulus during steady state can be constructed from the prior transient activity of the principal neurons.

These insights indicate that addressing transient dynamics is crucial from the perspective of predictive coding. However, understanding the transient dynamics of a non-linear recurrent network is usually difficult, and indeed it has not been previously solved for this circuit.

The central contribution of this paper is an analysis of the transient regime based on a rigorous solution of the dynamics in the multi-channel feedback circuit with threshold-linear interneurons in response to an arbitrary step-stimulus, e.g. switching on of an image (see Methods, section *Solution of the threshold-linear negative feedback circuit dynamics*). Remarkably, we were able to solve the non-linear circuit dynamics by mapping Equation 9 onto a signal processing algorithm called linearized Bregman iteration (Osher et al 2009, Yin et al 2008). This algorithm constructs a faithful representation of the stimulus $\boldsymbol{Wn} = \boldsymbol{s}$, while minimizing the L_1-L_2 norm of the interneuron activity (Zou & Hastie 2005) (see Methods, section *Linearized Bregman iteration*).

Next, we describe in words the mathematical expressions for the response of the feedback circuit to a step-stimulus (see Methods), see Figure 2f-g. Since initially inhibitory interneurons are silent, the principal neurons transmit the stimulus fully, just as in the absence of feedback (or in direct

transmission). The internal activity of the interneurons grows with a rate proportional to the projection of the sensory stimulus on their receptive fields, $\boldsymbol{W^T s}$. Unlike in the linear circuit, interneurons do not inhibit principal neurons until their internal activity crosses threshold, Figure 2f.

With time, interneurons cross threshold one by one and start contributing to the representation of the stimulus, thereby constructing a more accurate representation of the stimulus, Figure 2f,g. The first interneuron to cross threshold is the one for which the projection of the sensory stimulus on its receptive field, $\boldsymbol{W^T s}$ is highest. As its contribution is subtracted from the activity of the principal neurons, the driving force on other interneurons $\boldsymbol{W^T(s - Wa)}$ changes. Therefore, the order by which interneurons cross threshold depends also on the correlation between the receptive fields, Figure 2b,f. Collectively the representation progresses from sparse to dense, but individual interneurons may first be active then become silent. Eventually, the interneurons will accurately represent the input, i.e. their external activity times their receptive fields is equal to the stimulus ($\boldsymbol{s = Wa}$), and will fully subtract it from the principal cells' activity, resulting in no further excitation to the interneurons, Figure 2g,h.

Accordingly, at each point in time, the circuit subtracts a prediction, $\hat{\boldsymbol{s}} = \boldsymbol{Wa}$, which was constructed in the interneurons from previous incoming sensory signals, from the current sensory stimulus and the principal neurons transmit the residual, $\boldsymbol{p = s - \hat{s}}$, to higher brain areas. We note that initially the interneurons are silent and the principal neurons transmit the stimulus directly. If there were no bandwidth limitation, the stimulus could be decoded just from this initial transmission. However, the bandwidth limitation results in coarse, or noisy, principal neuron transmission, an issue we will return to later.

Understanding circuit dynamics in the predictive coding framework provides the following important insights. First, consider the length of transient activity for different types of stimuli. The time from stimulus onset to cancellation of the stimulus depends on the rate of the interneurons' activation, which in turn is proportional to the projection of the stimulus on the interneurons' receptive fields. Presumably, interneuron receptive fields are adapted to the most common stimuli, e.g. natural images in the case of the retina, therefore this type of stimulus should be relatively quickly cancelled out. In contrast, non-natural stimuli, like white noise patterns, will be represented by interneuron activity only after a longer delay. Accordingly, it will take longer to cancel out non-natural stimuli, leading to longer transients in the principal neurons (Figure S1). This stands as a prediction of the theory.

Second, since interneurons require time to charge up through threshold, fast switching between different patterns of stimuli, especially non-natural stimuli such as temporal white noise, should lead to cancellation and, hence, to relatively weak activation of interneurons. We believe this explains the relatively weak inhibitory surround often found in reverse correlation experiments when stimuli uncorrelated in time, e.g. white noise, are used to map ganglion cell receptive fields (Field et al 2010). In general, non-natural stimuli would activate interneurons only weakly and, therefore, to probe interneurons experimentally one should use natural stimuli.

Advantages of non-linear predictive coding for information transmission

In neuroscience, the predictive coding strategy was originally suggested to allow efficient transmission through a limited bandwidth channel (Srinivasan et al., 1982). Below, we show that the feedback circuit with threshold-linear neurons is indeed more efficient than the existing alternatives. We first consider a scenario in which bandwidth does not have a fixed limitation, but rather a cost associated with it. We are able to analyze this case analytically for additive noise. To do so we return to the single channel case, calculate transmission costs and show that, whereas direct transmission and the linear feedback circuit are suited for low and high signal-to-noise regimes respectively, the threshold linear circuit achieves low transmission cost in both. We then move on to multiple channels and generalize the single-channel results for that case. Lastly, we consider a different model, often used in neurobiology, with limited bandwidth transmission, where transmission bandwidth is set by the discreteness of a Poisson process.

We first consider transmission of two components, a step-like signal, s, and an uncorrelated, or unpredictable, component modeled by a white Gaussian $(0,\boldsymbol{N}^2)$ "noise". We quantify cumulative transmission cost by the integral of the squared activity in the principal neuron:

$$Transmission\ cost = \int_0^t p(t')^2\, dt' \quad , \tag{7}$$

We consider an encoding circuit composed of a single sensory channel, principal neuron and interneuron, and a decoding circuit of identical neurons, Figure 1, which guarantees lossless transmission. Admittedly, it is unclear that the brain is ultimately interested in transmission that allows full reconstruction of a stimulus and may instead only extract some part of the information. However, given that it is unknown which part of the stimulus the brain is interested in extracting, considering full reconstruction seems to be the most neutral, assumption free choice.

We calculated the cumulative transmission cost for a linear negative feedback circuit, Figure 1a and for direct transmission, Figure 1b, (for details see Methods, section *Transmission cost for additive noise with discrete-time dynamics*) and the summed expected value of transmission cost, for large t, is given by the following expression:

$$\begin{aligned} Direct\ transmission\ cost &\simeq (N^2 + s^2)t \\ Negative\ feedback\ transmission\ cost &\simeq (N^2 + N^2)t \end{aligned} \quad , \tag{8}$$

The above expressions can be understood in the following fashion. In the case of direct transmission one constantly transmits the value of the signal plus noise, hence the two components of the cost. In the linear negative feedback circuit the value of the signal is quickly subtracted away and is not repeatedly transmitted and thus the signal does not appear in the cost. However attempting to predict the noise increases its contribution to the bandwidth. This occurs since the noise is uncorrelated (unpredictable) between any two moments in time, and thus subtracting it will just reinject the noise, increasing the summed transmission bandwidth due to noise.

Thus, Equation 8 indicates that for signal-to-noise ratio (SNR) smaller than one it is more effective to use direct transmission (Figure 3a) whereas for higher SNR it is more effective to use the negative feedback circuit strategy (Figure 3b).

The threshold linear negative feedback circuit is effective in both high and low SNR regimes. Transmission cost can be directly calculated in the limits of strong or weak signal (see Methods, section: *transmission cost for additive noise with discrete dynamics*). Intuitively, if the signal is low the interneurons will not cross threshold for a long time. Thus, for most of the time the circuit will be operating in direct transmission mode since it is only the suprathreshold activity of the interneurons that affects the principal cells (Figure 3a). Conversely, for high signals the interneurons will quickly cross threshold and the circuit will act like the linear negative feedback circuit (Figure 3b). In this case again for most of the time the circuit will be operating in the more efficient mode.

What should the value of threshold λ be? In the limit of high and low signal-to-noise ratio the optimal value of the threshold can be calculated (see Methods, section: *Setting threshold value in the negative feedback circuit*). The parametric plot of the transmission costs as a function of λ is in Figure 3c. The threshold value of λ is chosen based on the relative frequency of the two regimes.

The calculations of transmission cost for the case of multiple channels with orthogonal interneuron receptive fields ($\boldsymbol{W^T W = I}$) can be obtained from the single channel case. Due to the orthogonality of interneuron receptive fields, the activity of interneurons is independent of each other, i.e., dependent only on the stimulus. Thus, the projection of the stimulus on each receptive field can be treated as a single, separate channel. The cost of transmission can then simply be summed across all these channels. If there are less interneurons than principal neurons, the remaining channels are treated simply as operating by direct transmission. Depending on the signal-to-noise ratios in the different channels, it will be more efficient to use direct transmission or a linear negative feedback circuit for multi-channel stimuli. As in the single channel case, the threshold linear circuit offers a good solution for both low and high signal-to-noise ratio channels.

If the interneuron receptive fields are not orthogonal one cannot simply sum the costs that would be incurred in independent channels because the activity of an interneuron in one channel will affect the activity of another interneuron in a different channel. Our simulations show that also in this case the threshold-linear circuit is more effective than both the linear negative feedback circuit and direct transmission (Figure 3d).

Why do threshold linear interneurons reduce transmission bandwidth? To answer this question, we recall that achieving an accurate representation of the stimulus requires spanning stimulus space with interneuron receptive fields. However, different stimulus ensembles may have drastically different dimensionality making it difficult to match the dimensionality of the stimulus and receptive field spaces. One possible solution would be for the interneuron receptive fields to span the space of all the stimuli ensembles, which would result in a highly redundant representation for some stimuli.

However, there is a problem with having highly redundant interneuron receptive fields, which is similar to over-fitting in statistics. Recall that real-world stimuli contain unpredictable components that are uncorrelated in time, which we refer to as "noise". By charging interneurons, such noise degrades the quality of their prediction because subtraction may compound the noise instead of cancelling it. Since noise can be present in every channel, having more degrees of freedom in the prediction than in the stimulus is detrimental.

To overcome the detrimental effect of noise on predictive coding we would like to reduce the number of degrees of freedom available for prediction of lower dimensional stimuli while preserving the capability to fit a higher dimensional stimulus with more degrees of freedom. Next, we argue that such flexible adjustment of the number of degrees of freedom used for prediction is achieved by linear thresholding. The sequential activation of more and more interneurons in time during the dynamics of the circuit can be seen as a process of representing the stimulus with predictions of increasing number of degrees of freedom (conceptually similar to a regularization path in statistics (Hastie et al., 2009)). The feature vectors that participate in the representation (e.g. high vs. low (spatial) frequency components) are determined by those components that have large activation and are able to cross threshold. At early times only the strongest activated, most reliable components, will cross threshold. As interneurons integrate over time the number of interneurons that cross threshold will increase, including components with weaker activation. Thus, the dynamics of the circuit afford a whole series of representations, from highly regularized predictions to fully accurate matching of the stimulus. Unlike the linear circuit where all feature vectors, i.e., interneuron receptive fields, participate in representing the stimulus, in the non-linear circuit only active interneurons, those whose membrane potential exceeds threshold, participate in the representation. Thus, different subsets of feature vectors may be used to represent the stimulus.

In addition to considering lossless transmission of correlated and uncorrelated components, we also studied a different model. In this model, we consider transmission where bandwidth limitation is set by the discreteness of spiking, modeled by a Poisson process. Instead of calculating the cost of transmission, in the new model we compute the reconstruction error. Although the discreteness of transmission can be overcome by averaging over time, this comes at the cost of longer perceptual delays, or lower transmission rates, as longer integration takes place. Therefore, we characterize transmission efficiency by reconstruction error as a function of time, Figure 4.

We find that, for Poisson transmission, predictive coding provides more accurate stimulus reconstruction than direct transmission for all times but the brief interval until the first interneuron has crossed threshold (Figure 4).

Discussion

In this study we analyzed early sensory processing by considering dynamics in a simplified model - a multi-channel negative feedback circuit with non-linear interneurons - in response to a step stimulus presentation. We showed that the negative feedback loop can be seen as implementing predictive coding by subtracting predictable components. This explains the seemingly paradoxical role of forming a stimulus representation within interneurons which lack long range

projections to transmit this representation to higher brain areas. Solving the non-linear dynamics for the first time, we are able to address the transient regime, which is critical for predictive coding, and explicitly show that in many cases the threshold-linear circuit deals with transmission bandwidth limitations better than other alternatives. Lastly, our theory provides straightforward predictions, such as briefer transients of principal neuron activity for natural versus non-natural stimuli.

To allow rigorous mathematical treatment of the dynamics we considered a model of early sensory circuits simplified in three main ways. First, we greatly simplified the complex temporal dynamics found in single neurons. Second, we lumped the multiple different cell types of inhibitory interneurons and different principal neurons into two uniform populations. Third, we neglected additional adaptation mechanisms that might occur in early sensory circuits. We briefly address each of these simplifications below.

Our simplifying assumption of principal neurons having zero time constants and interneurons having infinitely long time constants is motivated by the temporal structure of the step stimulus. Indeed, to encode rapid changes in stimuli, the principal neuron time constant must be shorter than the fastest timescale of change. The interneuron time constant must match the stimulus autocorrelation time, which for the step stimulus is infinite. For more realistic stimuli, one would expect interneuron time constant to be finite. The steady state of such circuit in response to a step stimulus will have non-zero activity in principal neurons.

Sensory systems often contain multiple distinct types of both interneurons, such as amacrine cell subclasses (Masland, 2001) in the retina or different local interneurons of the antennal lobe (Chou et al., 2010)) and principal neurons, such a bipolar cell subclasses (Masland, 2001). In addition to the cells that could be described by our mechanisms, there are examples of more specialized (non-linear) predictors, such as motion sensitive interneurons (Borst and Euler, 2011; Euler et al., 2002). Further study is required to determine which cell types are modeled by our theory perhaps with different time scales and which cell types generate more complex predictions beyond the current theoretical framework.

Clearly the gain control mechanism we describe is but one of multiple mechanisms allowing early sensory circuits to effectively encode information. The circuit adapts to the changes in the stimulus via synaptic plasticity on multiple time scales: from short term synaptic facilitation and depression on the scale of tens or hundreds of milliseconds (Dunn and Rieke, 2008), to history dependent adaptation effects on the scale of multiple seconds (Hosoya et al., 2005). On the single neuron level additional changes may occur affecting how synaptic currents are transformed into neural activity (Nagel and Wilson, 2011). Though the contributions of these different mechanisms may be conflated in some experiments, the multi-channel feedback mechanism described in this paper can potentially be teased apart from others mechanisms since it has two distinct attributes. First, it involves multiple channels and is thus distinct from single channel mechanisms. Second, it occurs on the time scale of neural activity, tens or hundreds of milliseconds, and is thus distinct from slower adaptive processes.

Circuit dynamics in response to a specific stimulus is affected by the number of interneurons and the shape of individual interneuron receptive fields. In particular, the presence of steady-state

activity depends on whether interneurons fully span the stimulus space. One might be tempted to declare a representation "overcomplete" if the number of interneurons is greater than the number of principal cells, as, for example, in the retina (Masland 2001) and olfactory bulb (Shepherd et al., 2007). However, the question of whether early sensory circuits are overcomplete is not as straightforward as it seems. Even if the number of interneurons is greater than the number of principal cells, interneurons may be composed of distinct types that code for very different properties of the stimulus (e.g., motion detection (Borst and Euler, 2011; Euler et al., 2002)) in which case it might be more accurate to consider the number of interneurons in each type. Conversely, the true dimensionality of the stimulus might be smaller than that of the number of principal neurons, for instance if different principal neurons are linearly dependent.

From a computational point of view there are three main advantages to overcompleteness in the negative feedback circuit. First, the delay until subtraction of prediction, which occurs when the first interneuron crosses threshold, will be briefer as the number of receptive fields grows since the maximal projection of the stimulus on the interneurons' receptive fields will be higher. Second, the larger the number of receptive fields the fewer the number of interneurons with suprathreshold activity, which may be energetically more efficient. Third, if stimuli come from different statistical ensembles, it could be advantageous to have receptive fields tailored to the different stimulus ensembles, which may result in more receptive fields, i.e., interneurons than principle neurons.

Because direct experimental measurements of interneuron receptive fields (de Vries et al., 2011) are few and far between we assumed a simple shape consistent with the known large width of receptive fields. Interneuron receptive fields were assumed to be Gaussian with a width between three and five times that of principal neuron receptive fields, which tiled the space. Interneuron receptive fields were placed on a regular two-dimensional grid with a spacing of three principal neuron receptive fields, resulting in an overlap of interneuron receptive fields (for additional details see methods). In addition, we confirmed that the results do not qualitatively change for other assumptions, including random, not grid-like, interneuron receptive field centers, or assuming other simple wide shapes for interneuron receptive fields such as uniform circles or squares.

Interneuron receptive fields may be learned in development. Computationally, a representation of the stimulus ensemble as a linear combination of the interneuron receptive fields (Dayan 1999, Olshausen & Field 1997) can be viewed as an instance of a statistical method called factor analysis. Factor analysis is a generalization of the principal component analysis (PCA) to the case where components are not necessarily orthogonal. From the statistical point of view, interneuron receptive fields serve as factors and interneuron activity as factor loadings. Accordingly, interneuron receptive fields can be learned from natural stimuli by adjusting synaptic weights using Hebbian plasticity rule (Dayan 1999, Olshausen & Field 1997). If the multi-channel circuit contains only a single interneuron, its receptive field approaches the first principal component of a presented stimuli set, thus insuring that the residual magnitude is as small as possible. If there are multiple interneurons, their receptive fields span the principal component subspace with the number of dimensions equal to the number of interneurons.

Whereas Srinivasan et al. originally suggested that interneurons subtract a prediction in a feedforward fashion, we considered feedback subtraction. There are two main reasons for focusing on the feedback nature of early sensory circuits. First, in many early sensory systems inhibition has a significant feedback component (Masland, 2001; Olsen et al., 2010; Shepherd et al., 2007). Second, although feedforward prediction can be faster than feedback, the predictor does not have easy access to prediction error and is therefore less robust to variation in circuit parameters (Astrom and Murray, 2008). Therefore, feedforward prediction alone would be brittle and must be used in combination with feedback.

Recently, sparse representations were studied in a single-layer circuit with lateral inhibitory connections (Figure S5), which constructs the stimulus representation in the projection neurons themselves and directly transmits it downstream (Rehn and Sommer, 2007; Rozell et al., 2008). Such a circuit was proposed as a model of primary cortical areas (Olshausen and Field, 1996), but, we believe such circuits do not model early sensory systems as well as the negative feedback circuit for a number of reasons. First, anatomical data is more consistent with the reciprocally connected interneuron layer than lateral connections between principal neurons (Masland, 2001; Shepherd et al., 2007). Second, direct transmission of the representation would result in greater perceptual delays after stimulus onset since no information is transmitted while the representation is being built up in the sub-threshold range (Figure 4) (though the transmission delay might be alleviated by multi-scale receptive fields (Perrinet, 2007)). In contrast, in the predictive coding model the projection neurons pass forth (a coarse and possibly noisy version of) the input stimulus from the very beginning.

There are subtle but important distinctions between our solution, which finds a faithful stimulus representation (if such is available) while minimizing a cost associated with the representation (namely the L_1-L_2 norms) and other solutions that converge to the minimum of a cost function composed of a representation error term, $\|\boldsymbol{s} - \boldsymbol{W}\boldsymbol{a}\|_2^2$, and a weighted representation norm, e.g., $\lambda\|\boldsymbol{a}\|_1$ (Dayan, 1999; Koulakov and Rinberg, 2011; Olshausen and Field, 1997; Rehn and Sommer, 2007; Rozell et al., 2008). First, in the combined cost function approach the representation is never precisely equal to the stimulus. Since any non-zero stimulus will have a non-zero activity representation it will have a non-zero representation cost. Therefore, unless the coefficient $\lambda=0$, minimizing the representation norm drives the minimum of the cost function away from the zero representation error.

Second, our solution scales with the intensity of the stimulus, i.e., multiplying the intensity of the stimulus by some factor would result in interneuron representation that is identical except for the multiplication of the activity by the same factor. In contrast, scaling of stimulus intensity in the combined representation cost and representation error cost function will result in a different relative weighting of representation error to representation cost (since the magnitude of the signal increased) and thus a different (as determined by the magnitude of λ) minimum will be found. Thus, our approach would predict that scaling of the intensity of the stimulus would not yield different sets of active interneurons, but rather a scaling up of their activity.

The principle of predictive coding has been previously applied to the operation of cortical loops (Jehee and Ballard, 2009; Rao and Ballard, 1999). These papers focus on the sequential interaction between the different levels of a hierarchical predictor with increasingly larger

receptive fields which is able to make predictions based on higher-level information, which then inform the lower sensory levels. Another application of predictive coding (Beck et al., 2011) considers feedback loop between the cortex and the olfactory bulb implementing Bayesian inference. In contrast, we focus on a specific feedback inhibition circuit in early sensory systems (prior to the thalamus), where hierarchies do not appear to play such an important role, propose a mechanistic description of the negative feedback circuit operation and show an explicit solution for threshold linear neurons.

In summary, by solving the dynamics of the negative feedback circuit through equivalence to linearized Bregman iteration we have shown that the development of activity in a simplified early sensory circuit can be viewed as implementing an efficient, non-linear, intrinsically parallel algorithm for predictive coding. Our study maps the steps of the algorithm onto specific neuronal substrates, providing a solid theoretical framework for understanding physiological experiments on early sensory processing as well as experimentally testing predictive coding ideas on a finer, more quantitative level.

Figures

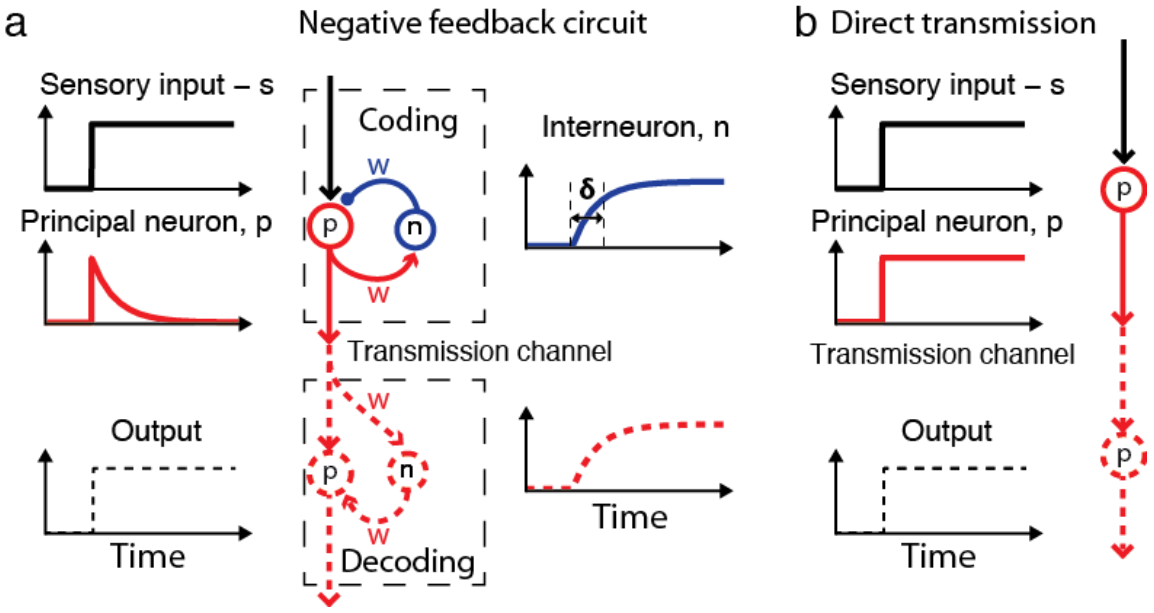


Figure 1. Schematic view of early processing in a single sensory channel in response to a step stimulus. a. A predictive coding model consists of a coding circuit, transmission channel and, for theoretical analysis only, a virtual decoding circuit. Coding is performed in a negative feedback circuit containing a principal neuron, *p*, and an inhibitory interneuron, *n*. In response to a step-stimulus (top left) the interneuron charges up with time (top right) till it reaches the value of the stimulus. Principal neuron (middle left) transmits the difference between the interneuron activity and the stimulus, resulting in a transient signal. **b.** Direct transmission model used for comparison of transmission costs.

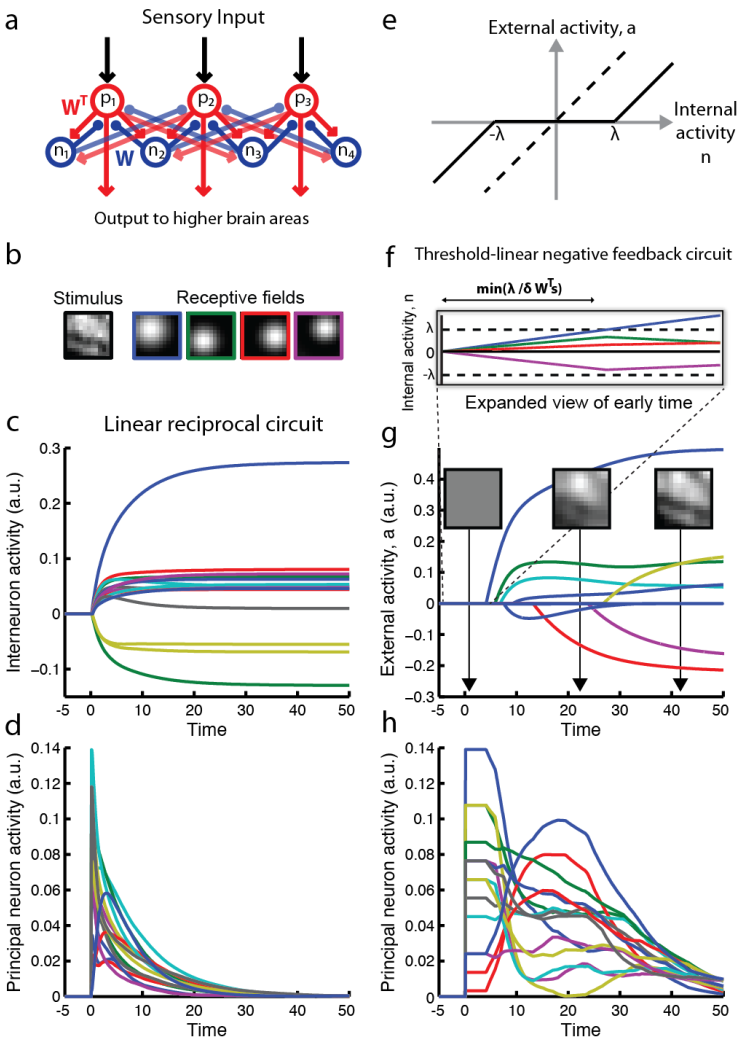


Figure 2. Predictive coding in a multi-channel negative feedback circuit in response to a step stimulus. a. Circuit diagram for multiple channel negative feedback circuit. **b.** Stimulus (grayscale in black box left) and a subset of interneuron's receptive field (grayscale in boxes). **c-d.** Response of linear negative feedback circuit to a step stimulus at time zero in interneurons (**c**) and principal neurons (**d**). **e.** Threshold-linear transfer function relating internal, *n*, and external, *a*, activity of interneurons. Dashed line shows diagonal. **f-h**. Response of interneurons (**f-g**) and principal neurons to a step stimulus at time zero. **f.** Initially, internal activity of the interneurons (only some are shown) grows proportionally to the correlation of the stimulus (grayscale in black box) and the interneuron's receptive field (see grayscale in boxes color coded to match traces in **b**). Once an interneuron (e.g. blue) crosses threshold (dashed black line) it inhibits other interneurons via the shared principal neurons. Note that green interneuron is inhibited more strongly than red because blue and green receptive fields are more similar. **g.** External activity of a larger subset of interneurons over a longer time period. Grayscale boxes show the stimulus represented by the interneuron layer at various times marked by arrows. **h.** Principal neuron activity as a function of time. Initial activity is determined by the stimulus. As more and more interneurons cross threshold they more closely represent the stimulus and cancel out more and more of the activity of the principal cells. Eventually, the interneuron representation (right box in **g**) is nearly identical to the stimulus (left box in **b**) and the principal neurons' activity drops almost to zero.

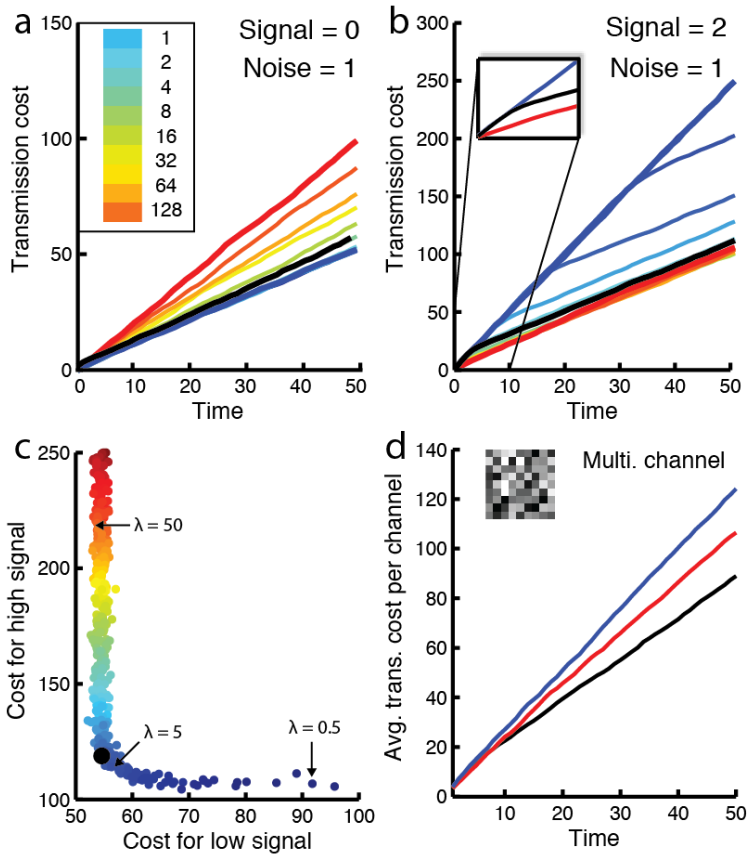


Figure 3. Integrated transmission cost for direct transmission and negative feedback circuit. Transmission cost as a function of time is plotted for non-linear circuits with different threshold values from zero, i.e., a linear negative feedback circuit (red), to infinity, i.e., direct transmission (blue), and values in between color coded as per inset in a. The optimal threshold value circuit (see Methods) is marked in black. **a.** Cost is plotted for low signal, SNR = 0. **b.** Cost is plotted for high signal, SNR = 2. Inset shows early times. **c.** Tradeoff for different threshold values. Each value of threshold is plotted as a circle at the x value equal to the low transmission cost at the final transmission time and the y value equal to the high signal cost. Large black circle shows optimal threshold value as in a, b. **d.** Average transmission cost per channel for a multichannel stimulus with SNR distributed uniformly between zero and 2 for each channel.

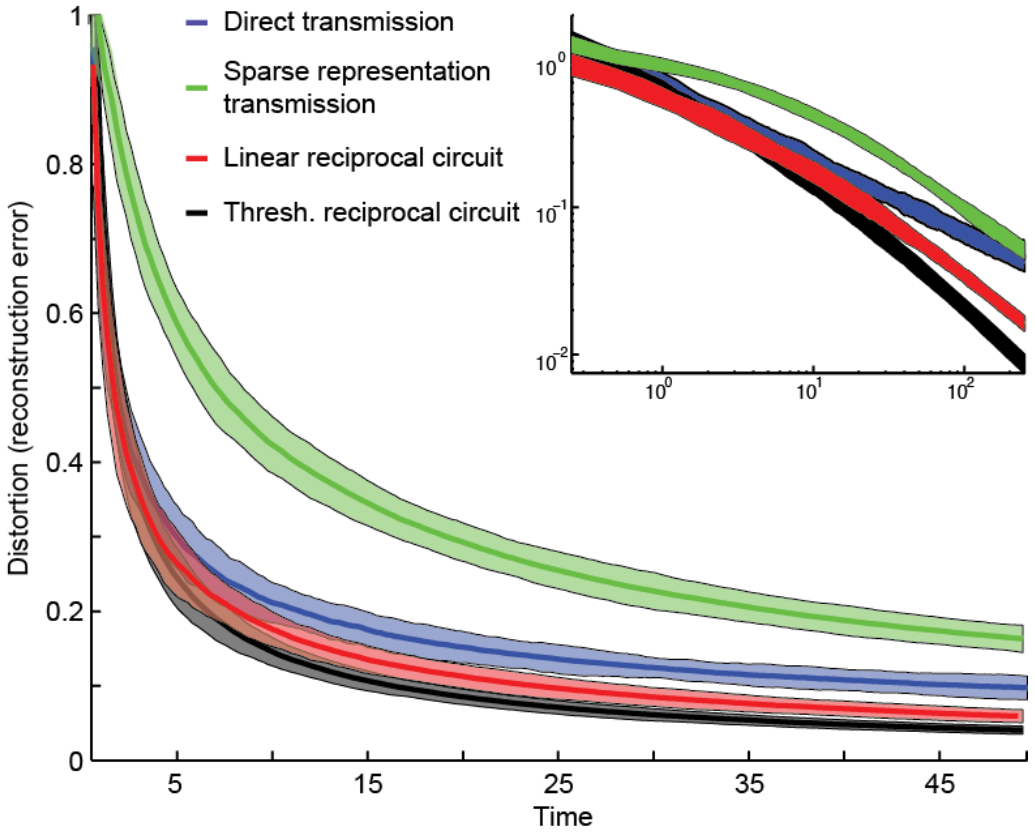


Figure 4. Effective use of bandwidth by negative feedback circuit. An image is transmitted through principal neurons that are bandwidth limited and noisy. The reconstruction error as a function of time following the presentation of stimulus is shown for the full non-linear negative feedback circuit (black), for a linear negative feedback circuit (red), for a direct transmission circuit (blue), and for a circuit where the sparse approximation itself is transmitted instead of the residual (green). Time on the x-axis is measured in units of the time length in which a single noisy transmission occurs (see Methods).

Methods

Derivation of the single channel circuit dynamics from neuron biophysics

Here we derive the dynamics of a single channel negative feedback loop, Equation 1, from the biophysical equations for the principal neuron, *p*, and an inhibitory interneuron, *n*. We start with a linear neuron model (Koch, 1999):

$$C_p^m \frac{dp}{dt} = -\frac{p}{R_p^m} + g_p^s(s - wn),$$

$$C_n^m \frac{dn}{dt} = -\frac{n}{R_n^m} + g_n^s wp \qquad \text{Eq. S1}$$

where R^m is the membrane resistance, C^m the membrane capacitance, g^s synaptic conductance and the subscript designates the neuron class (principal and interneuron). The connection strength is marked by w.

By rearranging the terms in Eq. S1 we obtain:

$$\tau_p \frac{dp}{dt} = -p + R_p^m g_p^s(s - wn),$$

$$\tau_n \frac{dn}{dt} = -n + R_n^m g_n^s wp, \qquad \text{Eq. S2}$$

where $\tau = RC$ is the membrane time constant.

If we assume that the time constant of the principal cells is small compared to that of the interneurons and the auto-correlation time of the stimulus, we can assume that the first equation reaches equilibrium instantaneously:

$$p = R_p^m g_p^s(s - wn)$$

$$\tau_n \frac{dn}{dt} = -n + R_n^m g_n^s wp, \qquad \text{Eq. S3}$$

As the purpose of integration is to construct stimulus representation, the integration time should be on the order of the auto-correlation time in the stimulus. Since here we are studying the simplified case of the semi-infinite step-stimulus, the time constant of the neuron should be approaching infinity. We assume this occurs by the interneurons having a very large membrane resistance (or correspondingly a very small conductance) and moderate capacitance. Therefore, the leakage term, *-n*, which is the only term in the second line of Eq. S3 that doesn't grow with the membrane resistance, can be neglected in the dynamics of interneurons. By this assumption and substituting the first equation into the second, we find:

$$p = R_p^m g_p^s(s - wn)$$

$$\tau_n \frac{dn}{dt} = R_n^m g_n^s R_p^m g_p^s w(s - wn), \qquad \text{Eq. S4}$$

If we define $\alpha = R_p^m g_p^s$ we obtain:

$$p = \alpha(s - wn)$$

$$\tau_n \frac{dn}{dt} = R_n^m g_n^s \alpha w(s - wn), \qquad \text{Eq. S5}$$

Defining the time constant $\delta = C_n^m R_n^s$ we have:

$$p = \alpha(s - wn)$$

$$\delta \frac{dn}{dt} = wp, \qquad \text{Eq. S6}$$

As in Equation 1 in the main text where we assumed for simplicity that α = 1.

Equivalence of multi-channel negative feedback loop dynamics and linearized Bregman iteration

Here we show that the dynamics of neurons in the negative feedback circuit, Figure 1a, is equivalent, under some assumptions, to the linearized Bregman iteration, which in turn finds a sparse representation of the stimulus, see section below. We assume that each interneuron is characterized by its internal activity, such as membrane voltage, $\boldsymbol{n}$, and its effect on other neurons is quantified by the external activity, $\boldsymbol{a}$, such as synaptic release. Assuming that the principal cells act as linear units we have the following equations:

$$C_p^m \frac{d\boldsymbol{p}}{dt} = -\frac{\boldsymbol{p}}{R_p^m} + g_p^s(\boldsymbol{s} - \boldsymbol{W}\boldsymbol{a})$$

$$C_n^m \frac{d\boldsymbol{n}}{dt} = -\frac{\boldsymbol{n}}{R_n^m} + g_n^s \boldsymbol{W}^T \boldsymbol{p}$$

$$\boldsymbol{a} = \mathrm{Thresh}_\lambda(\boldsymbol{n}), \qquad \text{Eq. S7}$$

where the same notation is used as in Equation S1 in the previous section. Finally, $\mathrm{Thresh}_\lambda(\boldsymbol{n})$ is a threshold linear function, which models non-linearity of synaptic release. Provided we combine On- and Off- neuronal pairs with the same receptive fields into a single unit, it can be written as:

$$\mathrm{Thresh}_\lambda(x) \equiv \begin{cases} 0, & \text{if } |x| \le \lambda \\ \mathrm{sign}(x)(|x| - \lambda), & \text{if } |x| > \lambda \end{cases} \qquad \text{Eq. S8}$$

Following the same derivation as in the section above we obtain:

$$\boldsymbol{p} = \alpha(\boldsymbol{s} - \boldsymbol{W}\boldsymbol{a})$$

$$\delta \frac{d\boldsymbol{n}}{dt} = \boldsymbol{W}^T \boldsymbol{p}$$

$$\boldsymbol{a} = \mathrm{Thresh}_\lambda(\boldsymbol{n}), \qquad \text{Eq. S9}$$

we can convert the differential equations into difference equations:

$$\Delta\boldsymbol{n} = \frac{\Delta t}{\delta} \boldsymbol{W}^T \left(\boldsymbol{s} - \boldsymbol{W}\boldsymbol{a}^k\right)$$

$$\boldsymbol{a} = \mathrm{Thresh}_\lambda(\boldsymbol{n}), \qquad \text{Eq. S10}$$

or:

$$\boldsymbol{n}^{k+1} = \boldsymbol{n}^k + \tilde{\delta}\boldsymbol{W}^T(\boldsymbol{s} - \boldsymbol{W}\boldsymbol{a}^k)$$

$$\boldsymbol{a}^{k+1} = \mathrm{Thresh}_\lambda(\boldsymbol{n}^{k+1}) \qquad \text{Eq. S11}$$

where $\tilde{\delta}$ is a time step measured in units of δ. The equations above represent linearized Bregman iteration (Osher et al., 2009; Yin et al., 2008), which provides a faithful reconstruction of the stimulus while minimizing an L_1, L_2 norm of the representation as we show next.

Linearized Bregman iteration

Here we find a represention of vector *s* in terms of a sparse combination of redundant features (columns of matrix $\boldsymbol{W}$), whose weights are given by vector $\boldsymbol{a}$. Formally, the problem is defined as follows:

$\min_{\boldsymbol{a}}\{J(\boldsymbol{a})\}\, s.t. \boldsymbol{Wa} = \boldsymbol{s}$,

$\mathrm{L}_1, \mathrm{L}_2$ cost (Elastic net): $J(\boldsymbol{a}) \equiv \mu||\boldsymbol{a}||_1 + \frac{1}{2\delta}||\boldsymbol{a}||_2^2$ Eq. S12

Remarkably, this high-dimensional non-linear optimization problem can be solved by a simple iterative scheme, Eq. S11, combining a linear step, which looks like gradient descent on the representation error, and a component-wise threshold-linear step.

The idea behind linearized Bregman iteration (Eq. S11), is to start with $\boldsymbol{a}^0 = 0$ and, at each iteration, to seek to update $\boldsymbol{a}$ so as to minimize the square error plus the distance from the previous value of $\boldsymbol{a}$. Thus, we perform the following update:

$$\boldsymbol{a}^{\boldsymbol{k+1}} = \mathrm{argmin}_{\boldsymbol{a}}\left\{D_J^{\boldsymbol{p}^k}(\boldsymbol{a}, \boldsymbol{a}^k) + \frac{1}{2}||\boldsymbol{s} - \boldsymbol{W}\boldsymbol{a}||^2\right\} \quad \text{Eq. S13}$$

where we used a notation $D_J^{\boldsymbol{p}}(\boldsymbol{a}, \boldsymbol{b})$ for the Bregman divergence (Bregman, 1967) between the two points $\boldsymbol{a}$ and $\boldsymbol{b}$ induced by the convex function J. The Bregman divergence is an appropriate measure for such problems and can handle the non-differentiable nature of the cost. It is defined by:

$$D_J^{\boldsymbol{p}}(\boldsymbol{a}, \boldsymbol{b}) = J(\boldsymbol{a}) - J(\boldsymbol{b}) - \langle \boldsymbol{p}, \boldsymbol{a} - \boldsymbol{b}\rangle \quad \text{Eq. S14}$$

where $\boldsymbol{p} \in \partial J(\boldsymbol{b})$ is an element of the subgradient of J at the point $\boldsymbol{b}$.

The Bregman divergence for the elastic net cost function J defined in Eq. S12 is:

$$D_J^{\boldsymbol{p}}(\boldsymbol{a}, \boldsymbol{a}^k) = \mu||\boldsymbol{a}||_1 - \mu\left||\boldsymbol{a}^k\right|_1 + \frac{1}{2\delta}||\boldsymbol{a}||_2^2 - \frac{1}{2\delta}\left|\boldsymbol{a}^k\right|_2^2 - \langle \boldsymbol{p}, \boldsymbol{a} - \boldsymbol{a}^k\rangle, \quad \text{Eq. S15}$$

where $\boldsymbol{p}$ is a subgradient of J at $\boldsymbol{a}^k$. The condition for the minimum in Eq. S13 is:

$$\partial\left[\mu\left||\boldsymbol{a}^{k+1}\right|_1 + \frac{1}{2\delta}\left|\boldsymbol{a}^{k+1}\right|_2^2\right] \ni \boldsymbol{p}^k + \boldsymbol{W}^T(\boldsymbol{s} - \boldsymbol{W}\boldsymbol{a}^k), \quad \text{Eq. S16}$$

where ∂ [.] designates a subdifferential. Consistency of the iteration scheme requires that the update $\boldsymbol{p}^{k+1}$ be a subgradient of J as well.

$$\partial\left[\mu\left||\boldsymbol{a}^{k+1}\right|_1 + \frac{1}{2\delta}\left|\boldsymbol{a}^{k+1}\right|_2^2\right] \ni \boldsymbol{p}^{k+1}. \quad \text{Eq. S17}$$

By combining Eqs. S16, 17 we set:

$$\boldsymbol{p}^{k+1} = \boldsymbol{p}^k + \boldsymbol{W}^T(\boldsymbol{s} - \boldsymbol{W}\boldsymbol{a}^k). \quad \text{Eq. S18}$$

By substituting Eq. S18 into Eq. S16 and simplifying we get:

$$\boldsymbol{a}^{k+1} = \mathrm{argmin}_u\left\{\mu\|\boldsymbol{a}\|_1 + \frac{1}{2\delta}\left||\boldsymbol{a} - \delta\boldsymbol{p}^{k+1}\right||^2\right\}, \quad \text{Eq. S19}$$

which can be solved explicitly:

$$\boldsymbol{a}^{k+1} = \mathrm{Thresh}_{\delta\mu}(\delta\boldsymbol{p}^{k+1}) \quad \text{Eq.S20}$$

By defining

$$\boldsymbol{n}^k = \delta\boldsymbol{p}^k \quad \text{Eq.S21}$$

and expressing it in Eqs. S18, 20 we get Eq. S10 with substitution $\lambda = \mu\delta$.

Solution of the threshold-linear negative feedback circuit dynamics

To detail the closed form solution for the threshold-linear negative feedback circuit we divide time into the intervals in between interneurons crossing threshold and consider them each in

sequence since between threshold crossings the dynamics are linear first order and can be explicitly solved.

At time $t^0 = 0$, and define the first threshold crossing, t^1 and so on.

Let us first define some auxiliary variables:

$\mathbf{\Gamma}^k$: an indicator function for which interneurons are above threshold at time t^k. For instance, $\mathbf{\Gamma}^0 = (0, 0, \dots, 0)$.

$\mathbf{d}^k$: the distance between each interneuron and its threshold at time t^k. For instance, $\mathbf{d}^0 = (\lambda, \lambda, \dots, \lambda)$.

Let us write down the dynamics for $t^0 < t < t^1$:

$$\boldsymbol{p} = \boldsymbol{s} - \boldsymbol{W}\boldsymbol{a}$$

$$\delta \frac{d\boldsymbol{n}}{dt} = \boldsymbol{W}^T\boldsymbol{p}$$

$$\boldsymbol{a} = \mathbf{0}, \qquad \text{Eq. S22}$$

since none of the interneurons are active the equations can be easily solved:

$$\boldsymbol{n}(t) = \boldsymbol{n}(t = 0) + \frac{1}{\delta}\boldsymbol{W}^T\boldsymbol{s}t = \frac{1}{\delta}\boldsymbol{W}^T\boldsymbol{s}t\,, \qquad \text{Eq. S23}$$

since we assume as usual that interneurons are initially at rest.

Let us define l^k as a vector of the time at which each interneuron would cross threshold if the dynamics stay the same. We note that this time will be different for each interneuron and that these are not the true times of threshold crossing since once a neuron crosses threshold the dynamics are no longer valid. To find the components of l^k we find when each neuron's activity has changed by the value of d^k. Thus, for l^1:

$$\boldsymbol{n_j}(t = l^1) - \boldsymbol{n_j}(t = 0) = d_j^0\,, \qquad \text{Eq. S24}$$

from the equation above we have:

$$l_j^1 = \frac{\delta\lambda}{(\boldsymbol{W}^T\boldsymbol{s})_j}\,, \qquad \text{Eq. S25}$$

Where by $\left(\boldsymbol{W}^T\boldsymbol{s}\right)_j$ we denote the j[th] component of the vector $\boldsymbol{W}^T\boldsymbol{s}$. We are interested in the first threshold crossing:

$$t^1 = \min(\boldsymbol{l}^1) = \frac{\delta\lambda}{\max(\boldsymbol{W}^T\boldsymbol{s})}\,, \qquad \text{Eq. S26}$$

Finding t^1 we update:

$$\mathbf{\Gamma}^1 = (0, 0, \dots, 1, \dots, 0) \text{ and } \mathbf{d}^1 = \left(\lambda - \frac{(\boldsymbol{W}^T\boldsymbol{s})_{\mathbf{1}}}{\delta}, \lambda - \frac{(\boldsymbol{W}^T\boldsymbol{s})_{\mathbf{2}}}{\delta}, \dots, 0, \dots, \lambda - \frac{(\boldsymbol{W}^T\boldsymbol{s})_{\boldsymbol{n}}}{\delta}\right), \qquad \text{Eq. S27}$$

Let us know find the solution for $t^i < t < t^{i+1}$

To make the calculations more concise let us work with variables $\tilde{t} = t - t^i$, so that time starts at zero and $\widetilde{\boldsymbol{n}} = \boldsymbol{n} - \boldsymbol{n}(t = t^i)$ so that $\boldsymbol{n}$ starts at zero.

Lastly, we denote $\boldsymbol{W}_{>}^{\boldsymbol{i}}$ as the matrix collecting those columns of $\boldsymbol{W}$ for which the interneurons are above threshold immediately following the i[th] threshold crossing, i.e. the indices for which $\mathbf{\Gamma}^k = 1$. Conversely, we denote $\boldsymbol{W}_{<}^{\boldsymbol{i}}$ collecting the columns of $\boldsymbol{W}$ for which the interneurons are

under threshold immediately following the ith threshold crossing. To avoid clutter we drop the superscript i above $\boldsymbol{W}$.

Thus, we can write the dynamics for the above threshold interneurons marked as $\widetilde{\boldsymbol{n}}_{>}$ and the subthreshold interneurons marked as $\widetilde{\boldsymbol{n}}_{<}$ as follows:

$$\boldsymbol{p} = \boldsymbol{s} - \boldsymbol{W}\boldsymbol{a} = \boldsymbol{s} - \boldsymbol{W}_{>}(\widetilde{\boldsymbol{n}}_{>} - \lambda \mathrm{sign}(\boldsymbol{n}_{>}))$$

$$\delta \frac{d\widetilde{\boldsymbol{n}}_{>}}{dt} = \boldsymbol{W}_{>}^{\boldsymbol{T}}\boldsymbol{p} = \boldsymbol{W}_{>}^{\boldsymbol{T}}\boldsymbol{s} - \boldsymbol{W}_{>}^{\boldsymbol{T}}\boldsymbol{W}_{>}(\widetilde{\boldsymbol{n}}_{>} - \lambda \mathrm{sign}(\boldsymbol{n}_{>}))\,, \qquad \text{Eq. S28}$$

Defining $\boldsymbol{A} \equiv \boldsymbol{W}_{>}^{\boldsymbol{T}}\boldsymbol{W}_{>}$ and $\boldsymbol{B} \equiv \boldsymbol{W}_{>}^{\boldsymbol{T}}\boldsymbol{W}_{>}\lambda \mathrm{sign}(\boldsymbol{n}_{>})$ we can simplify the equation above to:

$$\delta \frac{d\widetilde{\boldsymbol{n}}_{>}}{dt} = -\boldsymbol{A}\widetilde{\boldsymbol{n}}_{>} + \left(\boldsymbol{W}_{>}^{\boldsymbol{T}}\boldsymbol{s} + \boldsymbol{B}\right), \qquad \text{Eq. S29}$$

Assuming $\boldsymbol{A}$ is invertible we can write down the solution:

$$\widetilde{\boldsymbol{n}}_{>} = \boldsymbol{A}^{-\boldsymbol{1}}\exp\left(\frac{-\boldsymbol{A}}{\delta}\tilde{t}\right)\widetilde{\boldsymbol{C}} + \boldsymbol{A}^{-\boldsymbol{1}}\left(\boldsymbol{W}_{>}^{\boldsymbol{T}}\boldsymbol{s} + \boldsymbol{B}\right), \qquad \text{Eq. S30}$$

where $\widetilde{\boldsymbol{C}}$ is an integration constant determined by the initial conditions, $\widetilde{\boldsymbol{n}}(\tilde{t} = 0) = 0$, yielding:

$$\widetilde{\boldsymbol{n}}_{>} = \boldsymbol{A}^{-\boldsymbol{1}}\left(1 - \exp\left(\frac{-\boldsymbol{A}}{\delta}\tilde{t}\right)\right)\left(\boldsymbol{W}_{>}^{\boldsymbol{T}}\boldsymbol{s} + \boldsymbol{B}\right), \qquad \text{Eq. S31}$$

For the subthreshold interneurons:

$$\delta \frac{d\widetilde{\boldsymbol{n}}_{<}}{dt} = \boldsymbol{W}_{<}^{\boldsymbol{T}}\boldsymbol{p} = \boldsymbol{W}_{<}^{\boldsymbol{T}}\boldsymbol{s} - \boldsymbol{W}_{<}^{\boldsymbol{T}}\boldsymbol{W}_{>}(\widetilde{\boldsymbol{n}}_{>} - \lambda \mathrm{sign}(\boldsymbol{n}_{>}))\,, \qquad \text{Eq. S32}$$

Therefore:

$$\widetilde{\boldsymbol{n}}_{<} = \frac{1}{\delta}\boldsymbol{W}_{<}^{\boldsymbol{T}}\left(\boldsymbol{s} + \boldsymbol{W}_{>}\lambda \mathrm{sign}(\boldsymbol{n}_{>})\right)\tilde{t} - \frac{1}{\delta}\boldsymbol{W}_{<}^{\boldsymbol{T}}\boldsymbol{W}_{>}\int_{\boldsymbol{0}}^{\tilde{\boldsymbol{t}}}\widetilde{\boldsymbol{n}}_{>}(\tau)d\tau\,,$$

$$\widetilde{\boldsymbol{n}}_{<} = \frac{1}{\delta}\boldsymbol{W}_{<}^{\boldsymbol{T}}\left(\boldsymbol{s} + \boldsymbol{W}_{>}\lambda \mathrm{sign}(\boldsymbol{n}_{>})\right)\tilde{t} - \frac{1}{\delta}\boldsymbol{W}_{<}^{\boldsymbol{T}}\boldsymbol{W}_{>}\left[\boldsymbol{A}^{-\boldsymbol{1}}\left(\tilde{t} - \delta\boldsymbol{A}^{-1}\left(1 - \exp\left(\frac{-\boldsymbol{A}}{\delta}\tilde{t}\right)\right)\right)\left(\boldsymbol{W}_{>}^{\boldsymbol{T}}\boldsymbol{s} + \boldsymbol{B}\right)\right]. \qquad \text{Eq. S33}$$

Recall that these expressions hold only until the next threshold crossing:

$$t^{i+1} = \min\left[l^{i+1}\right], \qquad \text{Eq. S34}$$

Once the time and the neuron crossing threshold is found, we update the relevant variables and repeat the calculation. Thus, the solution proceeds from threshold crossing to threshold crossing.

Computing transmission reconstruction accuracy for Poisson transmission

To calculate transmission accuracy for Poisson transmission we simulated the activity of a noisy negative feedback circuit for a set of patches extracted from natural images. We performed this simulation for the threshold-linear negative feedback circuit, for a linear feedback circuit, for direct transmission by principal cells with no inhibition and for transmission of the sparse approximation itself. The activity of the principal cells was not transmitted as an analog value, but rather a discrete value generated by a Poisson process whose mean is equal to principal neuron activity.

In the case of direct transmission, signals were decoded by averaging the responses from the principal cells from all previous time points. For negative feedback circuit transmission decoding was performed by a circuit that reverses the process of prediction (Figure 1). Namely, the

transmission received was fed into a downstream predictor, $\tilde{n}$, that had the same structure as the inhibitory layer downstream, $\tilde{n} = n$, (an unrealistic assumption performed in the interest of simplicity); this prediction is then summed with the principal neuron output and the combined signal is averaged across previous time points:

$$\hat{s}^t = \frac{1}{t}\sum_{i=1}^{t}\left(p^i + \tilde{n}^i\right) \ , \qquad \text{Eq. S35}$$

For transmission of the interneuron activity itself, the signal was taken to be the sum over the receptive fields of active interneurons times their activity. This signal was averaged across all previous time points.

Reconstruction error was defined as the root of the squared difference between the reconstructed stimulus, averaged across previous time points, and the actual stimulus, divided by the norm of the stimulus. Results were averaged across 100 different patches to obtain variability estimates for the plot.

Transmission cost for additive noise with discrete-time dynamics

Here we consider discrete-time dynamics of a linear feedback and direct transmission circuits in response to a step stimulus, s, and white noise, ζ, injected into principal neuron. Because the principal neuron is linear such noise can be viewed either as an uncorrelated part of the sensory stimulus or as a transmission noise.

The dynamics of *the linear negative feedback circuit* with additive noise, ζ, are given by:

$$p^i = s + \zeta^i - wn^i$$
$$n^{i+1} = n^i + wp^i \ , \qquad \text{Eq. S36}$$

with initial values of: $p^0 = n^1 = 0$. Note that to maintain causality in the negative feedback loop we introduce a one-step delay in the second equation, which could correspond to a synaptic transmission delay.

By substituting the first equation into the second we get a recursive relation for n^i:

$$n^{i+1} = n^i + w\left(s + \zeta^i - wn^i\right) = ws + w\zeta^i + (1 - w^2)n^i \qquad \text{Eq. S37}$$

Following the recursion back to the initial values one obtains:

$$n^{i+1} = ws\sum_{k=1}^{i}(1 - w^2)^{i-k} + w\sum_{k=1}^{i}(1 - w^2)^{i-k}\zeta^k \ , \qquad \text{Eq. S38}$$

By summing the geometric series in the first term we get:

$$n^{i+1} = \frac{s}{w}\left[1 - (1 - w^2)^i\right] + w\sum_{k=1}^{i}(1 - w^2)^{i-k}\zeta^k \ , \qquad \text{Eq. S39}$$

By substituting this into Eq. S36, the principal neuron activity is given by:

$$p^i = s(1 - w^2)^{i-1} - w^2\sum_{k=1}^{i-1}(1 - w^2)^{i-1-k}\zeta^k + \zeta^i \ . \qquad \text{Eq. S40}$$

The cost of transmission bandwidth at time point *i* is given by:

$$C^i = \left(p^i\right)^2 . \qquad \text{Eq. S41}$$

Assuming that the noise, ζ, has zero mean and variance N^2, the expected value of the cost function is thus:

$$E\left[C^i\right] = s^2(1-w^2)^{2i-2} + w^4N^2\sum_{k=1}^{i-1}(1-w^2)^{2(i-1-k)} + N^2\,. \quad \text{Eq. S42}$$

Computing the second sum:

$$E\left[C^i\right] = s^2(1-w^2)^{2i-2} + w^2N^2\frac{\left(1-(1-w^2)^{2i-2}\right)}{2-w^2} + N^2$$

$$E\left[C^i\right] = s^2(1-w^2)^{2i-2} + N^2\frac{\left(2-w^2(1-w^2)^{2i-2}\right)}{2-w^2}. \quad \text{Eq. S43}$$

We are interested in the cumulative cost over transmission:

$$\sum_{i=1}^{t} E\left[C^i\right] = s^2\sum_{i=1}^{t}(1-w^2)^{2i-2} + \frac{N^2}{2-w^2}\left(\sum_{i=1}^{t}(2) - w^2\sum_{i=1}^{t}(1-w^2)^{2i-2}\right). \quad \text{Eq. S44}$$

For large t and $0 < w \leq 1$ the sum can be approximated by the sum over an infinite series:

$$\sum_{k=1}^{t} E\left[C^i\right] \simeq \frac{N^2}{2-w^2}\left(2t - \frac{1}{2-w^2}\right) + \frac{s^2}{2w^2-w^4}. \quad \text{Eq. S45}$$

For unitary gain, $w = 1$, this simplifies to:

$$\sum_{k=1}^{t} E[C^k] \simeq N^2(2t-1) + s^2\,. \quad \text{Eq. S46}$$

For *direct transmission* the principal neuron activity is given by:

$$p^i = s + \zeta^i\,. \quad \text{Eq. S47}$$

Plugging this result into the transmission cost:

$$C^i = \left(s + \zeta^i\right)^2, \quad \text{Eq. S48}$$

following the same calculation as above, the expected value for the transmission cost gives:

$$E\left[C^i\right] = s^2 + N^2\,. \quad \text{Eq. S49}$$

Finally the summed transmission cost:

$$\sum E\left[C^i\right] = (s^2 + N^2)t\ . \quad \text{Eq. S50}$$

Equations S46 and S50 are combined in equation 8 of the main text.

Below we calculate the average cost of transmission in *the threshold-linear negative feedback circuit*. Explicitly expressing the cost of transmission for the threshold-linear negative feedback circuit for one particular noisy instantiation is difficult since the time of threshold crossing depends on the particular instantiation of the noise process. Since the cost of transmission depends on whether the interneuron is active or not, as seen in the equations above, different threshold crossing times will lead to different transmission costs. However, the average time of threshold crossing is immediately obtainable in the limit of large signal since before threshold crossing the interneuron simply sums the activity of the principal neuron and the time of threshold crossing, denoted by t_λ is simply $t_\lambda = \frac{\lambda}{\delta s}$. In what follows we assume $\delta = 1$ to avoid clutter.

Before threshold crossing the circuit behaves exactly as a direct transmission circuit. Thus, we can write down the expression for transmission bandwidth cost up until that point (Eq. S52):

$$E\left[C^i\right] = s^2 + N^2\,. \quad \text{Eq. S51}$$

If we neglect the possibility of crossing threshold and then having noise drive the circuit down back under threshold (the higher the signal-to-noise ratio the safer this approximation) then after threshold crossing the circuit behaves as a linear circuit starting at zero activation of the interneuron (Eq. S49). Thus we have the following expressions following threshold crossing:

$$E\left[C^{i=t_\lambda}\right] = s^2 + N^2;\ \ E\left[C^{i>t_\lambda}\right] = 2N^2 \ . \qquad \text{Eq. S52}$$

Finally we can combine the costs before and after threshold crossing to obtain:

$$\sum E\left[C^i\right] = \frac{\lambda}{s}(s^2 + N^2) + 2N^2\left(t - \frac{\lambda}{s} - 1\right) + s^2 \ . \qquad \text{Eq. S53}$$

We note that this expression grows asymptotically just as the linear negative feedback circuit but has an additional offset due to the time spent under threshold.

In the low signal limit a different calculation needs to be performed since there is no assurance that a neuron that crosses threshold will remain over threshold. Consider the extreme case where signal is zero. The cost is given by:

$$E\left[C^i\right] = E\left[\left(p^i\right)^2\right] = E\left[\left(\zeta^i - a^i\right)^2\right] = E\left[\left(\zeta^i\right)^2\right] - 2E\left[\zeta^i a^i\right] + E\left[\left(a^i\right)^2\right] . \qquad \text{Eq. S54}$$

Since a^i is just a sum of the previous noise terms it is uncorrelated with ζ^i. Thus:

$$E\left[C^i\right] = N^2 + E\left[\left(a^i\right)^2\right] . \qquad \text{Eq. S55}$$

As long as the interneuron is subthreshold, the value of *a* is zero. Thus we can write:

$$E\left[\left(a^i\right)^2\right] = \text{prob}\left(n^i > \lambda\right)\left(n^i - \lambda\right)^2 . \qquad \text{Eq. S56}$$

Accordingly, we need to calculate when *n* crosses threshold. Since there is no signal term, n is just a summation of uncorrelated noise terms, or a random walk. Once *n* crosses threshold the negative feedback circuit will spring into action and return n towards threshold. Thus, given that enough time has passed, we can approximate *n* as being distributed uniformly between $-\lambda$ and λ. Threshold crossing will occur when *n* is close enough to threshold and the arriving noise term is of the right magnitude (positive for positive threshold). Assume for simplicity we have jumps of exactly size N. Thus any *n* of value between λ and $\lambda - N$ will have a fifty percent chance of crossing threshold. The fraction of such values is $\frac{N}{\lambda}$. Assume for the sake of simplicity that when such points cross over threshold they have a value of $\lambda + N$. Thus:

$$E\left[\left(a^i\right)^2\right] = \text{prob}\left(n^i > \lambda\right)\left(n^i - \lambda\right)^2 \simeq \left(\frac{N}{\lambda}\right)(N^2)$$

$$E\left[\left(a^i\right)^2\right] \simeq \frac{N^3}{\lambda} , \qquad \text{Eq. S57}$$

and finally:

$$E\left[C^i\right] = N^2 + \frac{N^3}{\lambda} . \qquad \text{Eq. S58}$$

Summing transmission cost:

$$\sum E\left[C^i\right] = \left(N^2 + \frac{N^3}{\lambda}\right) t . \qquad \text{Eq. S59}$$

Setting threshold value in the negative feedback circuit

As shown in the previous section for SNR smaller than one direct transmission is more effective for all time points and hence the threshold should be set at infinity. Conversely, if SNR is greater than one, the linear negative feedback circuit is more effective and threshold should be set to zero. However, if different SNR values may occur for a given channel, some greater than one and some smaller than one, then a threshold-linear circuit may be more effective than either solution since it functions as direct transmission under threshold and as a linear negative feedback circuit above threshold.

Continuing in the limit of high SNR and very low SNR, consider a simple example where a channel with fixed noise either receives a strong signal, *s*, which occurs a fraction *f* of the time or no signal at all. We continue to assume that the stimuli changes in a step like fashion where the length of each step is long. The cost of transmission is therefore given by:

$$C_{total} = fC_{high} + (1-f)C_{low} \,. \qquad \text{Eq. S60}$$

Plugging in the transmission cost expressions:

$$C_{total} = f\left(\frac{\lambda}{s}(s^2+N^2) + 2N^2\left(t - \frac{\lambda}{s} - 1\right) + s^2\right) + (1-f)\left(N^2 + \frac{N^3}{2\lambda}\right)t \,. \qquad \text{Eq. S61}$$

Figure 3c illustrates the dependence of C_{high} and C_{low} as a function of threshold, λ, for $N = 1$. The optimal threshold is found by taking the derivative with respect to the threshold:

$$\lambda^* = \sqrt{\frac{(1-f)N^3 ts}{2f(s^2-N^2)}} \,. \qquad \text{Eq. S62}$$

If there is only high signal in this channel, f=1, we find that the threshold should be zero, as expected since the linear negative feedback circuit is the most efficient for this case. If there is only low signal in this channel, f=0, we find that the threshold should be infinite, as expected since direct transmission is more efficient in this case. We note that the divergence of the expression if noise and signal are equal is artificial recalling our initial assumption that when the signal is on it is at a high SNR so $s^2 > N^2$.

Simulating the dynamics of the negative feedback circuit

The dynamics of the negative feedback circuit were simulated according to the difference equations described above. Numerical integration was performed in Matlab with a time step of 0.1 δ, i.e. one tenth of the relevant time constant. 169 principal neurons were simulated. Stimuli were 2D patches of size 13x13 (169). Interneuron receptive fields were chosen as Gaussians with widths and centers chosen at random from a uniform distribution. 338 interneurons were simulated making the representation twice overcomplete. The threshold for the interneurons was set at a value of 5 where the norm of the receptive fields was set to 1. Simulations were carried on until the stimulus was accurately represented by the interneuron layer (representation mismatch equal to 0.001 of the norm of the stimulus) and subtracted from the principal cells.

Arevian, A.C., Kapoor, V., and Urban, N.N. (2008). Activity-dependent gating of lateral inhibition in the mouse olfactory bulb. Nature neuroscience *11*, 80-87.
Atick, J.J. (1992). Could Information-Theory Provide an Ecological Theory of Sensory Processing. Network-Computation in Neural Systems *3*, 213-251.
Atick, J.J., Li, Z., and Redlich, A.N. (1992). Understanding retinal color coding from first principles. Neural Comput *4*, 559--572.
Atick, J.J., and Redlich, A.N. (1990). Towards a theory of early visual processing. Neural Comput *2*, 308--320.
Attneave, F. (1954). Some informational aspects of visual perception. Psychological review *61*, 183-193.
Baccus, S.A. (2007). Timing and computation in inner retinal circuitry. Annu Rev Physiol *69*, 271-290.
Barlow, H.B., and Levick, W.R. (1976). Threshold setting by the surround of cat retinal ganglion cells. J Physiol *259*, 737-757.
Beck, J.M., Pouget, A., and Latham, P. (2011). A neural network which approximates Bayesian inference and learning in a model of odour segmentation. In Computational and Systems Neuroscience (Salt Lake City, UT).
Borst, A., and Euler, T. (2011). Seeing things in motion: models, circuits, and mechanisms. Neuron 71, 974-994.
Bregman, L.M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming* 1. USSR computational mathematics and mathematical physics *7*, 200--217.
Chou, Y.H., Spletter, M.L., Yaksi, E., Leong, J.C., Wilson, R.I., and Luo, L. (2010). Diversity and wiring variability of olfactory local interneurons in the Drosophila antennal lobe. Nature neuroscience *13*, 439-449.
Dayan, P. (1999). Recurrent sampling models for the Helmholtz machine. Neural Comput *11*, 653-678.
de Vries, S.E., Baccus, S.A., and Meister, M. (2011). The projective field of a retinal amacrine cell. J Neurosci *31*, 8595-8604.
Dong, D.W., and Atick, J.J. (1995a). Statistics of natural time-varying images. Network: Computation in Neural Systems *6*, 345--358.
Dong, D.W., and Atick, J.J. (1995b). Temporal decorrelation: a theory of lagged and nonlagged responses in the lateral geniculate nucleus. Network: Computation in Neural Systems *6*, 159-178.
Dunn, F.A., and Rieke, F. (2008). Single-Photon Absorptions Evoke Synaptic Depression in the Retina to Extend the Operational Range of Rod Vision. Neuron *57*, 894-904.
Elias, P. (1955). Predictive coding. Information Theory, IRE Transactions on *1*, 16--24.
Euler, T., Detwiler, P.B., and Denk, W. (2002). Directionally selective calcium signals in dendrites of starburst amacrine cells. Nature *418*, 845-852.
Field, D.J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. J Opt Soc Am A *4*, 2379-2394.
Hosoya, T., Baccus, S.A., and Meister, M. (2005). Dynamic predictive coding by the retina. Nature *436*, 71-77.
Huang, Y., and Rao, R.P.N. (2011). Predictive coding. Wiley Interdisciplinary Reviews: Cognitive Science *2*, 580-593.

Jehee, J.F.M., and Ballard, D.H. (2009). Predictive Feedback Can Account for Biphasic Responses in the Lateral Geniculate Nucleus. PLoS Comput Biol *5*, e1000373.
Kamermans, M., and Spekreijse, H. (1999). The feedback pathway from horizontal cells to cones. A mini review with a look ahead. Vision Res *39*, 2449-2468.
Koch, C. (1999). Biophysics of computation : information processing in single neurons (New York, Oxford University Press).
Koulakov, A.A., and Rinberg, D. (2011). Sparse Incomplete Representations: A Potential Role of Olfactory Granule Cells. Neuron 72, 124-136.
Laughlin, S. (1981). A simple coding procedure enhances a neuron's information capacity. Zeitschrift fur Naturforschung Section C: Biosciences *36*, 910-912.
Laughlin, S.B., Howard, J., and Blakeslee, B. (1987). Synaptic limitations to contrast coding in the retina of the blowfly Calliphora. Proc R Soc Lond B Biol Sci *231*, 437-467.
Masland, R.H. (2001). The fundamental plan of the retina. Nature neuroscience *4*, 877-886.
Matthews, G., and Fuchs, P. (2010). The diverse roles of ribbon synapses in sensory neurotransmission. Nat Rev Neurosci *11*, 812-822.
Meister, M., and Berry, M.J., 2nd (1999). The neural code of the retina. Neuron *22*, 435-450.
Nagel, K.I., and Wilson, R.I. (2011). Biophysical mechanisms underlying olfactory receptor neuron dynamics. Nature neuroscience *14*, 208-216.
Nirenberg, S., Bomash, I., Pillow, J.W., and Victor, J.D. (2010). Heterogeneous response dynamics in retinal ganglion cells: the interplay of predictive coding and adaptation. In J Neurophysiol, pp. 3184-3194.
Olsen, S.R., Bhandawat, V., and Wilson, R.I. (2010). Divisive normalization in olfactory population codes. Neuron *66*, 287-299.
Olshausen, B.A., and Field, D.J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature *381*, 607-609.
Olshausen, B.A., and Field, D.J. (1997). Sparse coding with an overcomplete basis set: a strategy employed by V1? Vision Res *37*, 3311-3325.
Osher, S., Mao, Y., Dong, B., and Yin, W. (2009). Fast linearized Bregman iteration for compressive sensing and sparse denoising. Communications in Mathematical Sciences.
Perrinet, L. (2007). Dynamical neural networks: Modeling low-level vision at short latencies. The European Physical Journal-Special Topics *142*, 163--225.
Rao, R.P.N., and Ballard, D.H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. nature neuroscience *2*, 79--87.
Rehn, M., and Sommer, F.T. (2007). A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. Journal of computational neuroscience *22*, 135-146.
Rieke, F., and Schwartz, G. (2011). Nonlinear spatial encoding by retinal ganglion cells: when 1+1 not equal 2. Journal of General Physiology *138*, 283-290.
Rozell, C.J., Johnson, D.H., Baraniuk, R.G., and Olshausen, B.A. (2008). Sparse coding via thresholding and local competition in neural circuits. Neural Comput *20*, 2526-2563.
Ruderman, D.L., and Bialek, W. (1994). Statistics of natural images: Scaling in the woods. Phys Rev Lett *73*, 814-817.
Shapley, R.M., and Victor, J.D. (1978). The effect of contrast on the transfer properties of cat retinal ganglion cells. J Physiol *285*, 275-298.
Shepherd, G.M., Chen, W.R., Willhite, D., Migliore, M., and Greer, C.A. (2007). The olfactory granule cell: from classical enigma to central role in olfactory processing. Brain Res Rev *55*, 373-382.

Srinivasan, M.V., Laughlin, S.B., and Dubs, A. (1982). Predictive coding: a fresh view of inhibition in the retina. In Proc R Soc Lond, B, Biol Sci, pp. 427-459.
van Hateren, J.H. (1992). A theory of maximizing sensory information. Biological cybernetics *68*, 23-29.
Victor, J.D. (1999). Temporal aspects of neural coding in the retina and lateral geniculate. Network-Computation in Neural Systems *10*, R1-R66.
Weckstrom, M., and Laughlin, S. (2010). Extracellular potentials modify the transfer of information at photoreceptor output synapses in the blowfly compound eye. J Neurosci *30*, 9557-9566.
Wilson, H.R. (1999). Spikes, decisions, and actions: The dynamical foundations of neuroscience (Oxford University Press).
Yin, W., Osher, S., Goldfarb, D., and Darbon, J. (2008). Bregman iterative algorithms for l1-minimization with applications to compressed sensing. SIAM Journal on Imaging Sciences *1*, 143--168.