

A Normative Theory of Adaptive Dimensionality Reduction in Neural Networks

Authored by:

Cengiz Pehlevan
Dmitri Chklovskii

Abstract

To make sense of the world our brains must analyze high-dimensional datasets streamed by our sensory organs. Because such analysis begins with dimensionality reduction, modelling early sensory processing requires biologically plausible online dimensionality reduction algorithms. Recently, we derived such an algorithm, termed similarity matching, from a Multidimensional Scaling (MDS) objective function. However, in the existing algorithm, the number of output dimensions is set a priori by the number of output neurons and cannot be changed. Because the number of informative dimensions in sensory inputs is variable there is a need for adaptive dimensionality reduction. Here, we derive biologically plausible dimensionality reduction algorithms which adapt the number of output dimensions to the eigenspectrum of the input covariance matrix. We formulate three objective functions which, in the offline setting, are optimized by the projections of the input dataset onto its principal subspace scaled by the eigenvalues of the output covariance matrix. In turn, the output eigenvalues are computed as i) soft-thresholded, ii) hard-thresholded, iii) equalized thresholded eigenvalues of the input covariance matrix. In the online setting, we derive the three corresponding adaptive algorithms and map them onto the dynamics of neuronal activity in networks with biologically plausible local learning rules. Remarkably, in the last two networks, neurons are divided into two classes which we identify with principal neurons and interneurons in biological circuits.

1 Paper Body

Our brains analyze high-dimensional datasets streamed by our sensory organs with efficiency and speed rivaling modern computers. At the early stage of such analysis, the dimensionality of sensory inputs is drastically reduced as evidenced by anatomical measurements. Human retina, for example, conveys signals from 125 million photoreceptors to the rest of the brain via 1 million ganglion cells [1] suggesting a hundred-fold dimensionality reduction. Therefore, biologically

plausible dimensionality reduction algorithms may offer a model of early sensory processing. In a seminal work [2] Oja proposed that a single neuron may compute the first principal component of activity in upstream neurons. At each time point, Oja’s neuron projects a vector composed of firing rates of upstream neurons onto the vector of synaptic weights by summing up currents generated by its synapses. In turn, synaptic weights are adjusted according to a Hebbian rule depending on the activities of only the postsynaptic and corresponding presynaptic neurons [2]. Following Oja’s work, many multineuron circuits were proposed to extract multiple principal components of the input, for a review see [3]. However, most multineuron algorithms did not meet the same level of rigor and biological plausibility as the single-neuron algorithm [2, 4] which can be derived using a normative approach, from a principled objective function [5], and contains only local

cal Hebbian learning rules. Algorithms derived from principled objective functions either did not possess local learning rules [6, 4, 7, 8] or had other biologically implausible features [9]. In other algorithms, local rules were chosen heuristically rather than derived from a principled objective function [10, 11, 12, 9, 3, 13, 14, 15, 16]. There is a notable exception to the above observation but it has other shortcomings. The twolayer circuit with reciprocal synapses [17, 18, 19] can be derived from the minimization of the representation error. However, the activity of principal neurons in the circuit is a dummy variable without its own dynamics. Therefore, such principal neurons do not integrate their input in time, contradicting existing experimental observations. Other normative approaches use an information theoretical objective to compare theoretical limits with experimentally measured information in single neurons or populations [20, 21, 22] or to calculate optimal synaptic weights in a postulated neural network [23, 22]. Recently, a novel approach to the problem has been proposed [24]. Starting with the Multidimensional Scaling (MDS) strain cost function [25, 26] we derived an algorithm which maps onto a neuronal circuit with local learning rules. However, [24] had major limitations, which are shared by various other multineuron algorithms: 1. The number of output dimensions was determined by the fixed number of output neurons precluding adaptation to the varying number of informative components. A better solution would be to let the network decide, depending on the input statistics, how many dimensions to represent [14, 15]. The dimensionality of neural activity in such a network would be usually less than the maximum set by the number of neurons. 2. Because output neurons were coupled by anti-Hebbian synapses which are most naturally implemented by inhibitory synapses, if these neurons were to have excitatory outputs, as suggested by cortical anatomy, they would violate Dale’s law (i.e. each neuron uses only one fast neurotransmitter). Here, following [10], by anti-Hebbian we mean synaptic weights that get more negative with correlated activity of pre- and postsynaptic neurons. 3. The output had a wide dynamic range which is difficult to implement using biological neurons with a limited range. A better solution [27, 13] is to equalize the output variance across neurons. In this paper, we advance the normative approach of [24] by proposing three new objective functions which allow us to overcome the above limitations.

We optimize these objective functions by proceeding as follows. In Section 2, we formulate and solve three optimization problems of the form: Offline setting : $Y^* = \arg \min_Y L(X, Y)$.

(1)
Y

Here, the input to the network, $X = [x_1, \dots, x_T]$ is an $n \times T$ matrix with T centered input data samples in \mathbb{R}^n as its columns and the output of the network, $Y = [y_1, \dots, y_T]$ is a $k \times T$ matrix with corresponding outputs in \mathbb{R}^k as its columns. We assume $T \ll k$ and $T \ll n$. Such optimization problems are posed in the so-called offline setting where outputs are computed after seeing all data. Whereas the optimization problems in the offline setting admit closed-form solution, such setting is ill-suited for modeling neural computation on the mechanistic level and must be replaced by the online setting. Indeed, neurons compute an output, y_T , for each data sample presentation, x_T , before the next data sample is presented and past outputs cannot be altered. In such online setting, optimization is performed at every time step, T , on the objective which is a function of all inputs and outputs up to time T . Moreover, an online algorithm (also known as streaming) is not capable of storing all previous inputs and outputs and must rely on a smaller number of state variables. In Section 3, we formulate three corresponding online optimization problems with respect to y_T , while keeping all the previous outputs fixed: Online setting : $y_T^* = \arg \min_{y_T} L(X, Y)$.

(2)
 y_T

Then we derive algorithms solving these problems online and map their steps onto the dynamics of neuronal activity and local learning rules for synaptic weights in three neural networks. We show that the solutions of the optimization problems and the corresponding online algorithms remove the limitations outlined above by performing the following computational tasks: 2

C output eig.

?

B output eig.

output eig.

A

??

input eig. E

D y_k

$y_1 x_1$

z_1

x_2

Principal

x_1

...

z_1

z_l

$y_1 x_n$

x_1
 y_k x_2
 \dots
 x_n
 anti-Hebbian synapses
 Hebbian
 Inter-neurons
 input eig.
 F
 z_1 y_k
 x_2
 $?$
 input eig.
 y_1
 \dots x_n
 $?$

Figure 1: Input-output functions of the three offline solutions and neural network implementations of the corresponding online algorithms. A-C. Input-output functions of covariance eigenvalues. A. Soft-thresholding. B. Hard-thresholding. C. Equalization after thresholding. D-F. Corresponding network architectures.

1. Soft-thresholding the eigenvalues of the input covariance matrix, Figure 1A: eigenvalues below the threshold are set to zero and the rest are shrunk by the threshold magnitude. Thus, the number of output dimensions is chosen adaptively. This algorithm maps onto a single-layer neural network with the same architecture as in [24], Figure 1D, but with modified learning rules. 2. Hard-thresholding of input eigenvalues, Figure 1B: eigenvalues below the threshold vanish as before, but eigenvalues above the threshold remain unchanged. The steps of such algorithm map onto the dynamics of neuronal activity in a network which, in addition to principal neurons, has a layer of interneurons reciprocally connected with principal neurons and each other, Figure 1E. 3. Equalization of non-zero eigenvalues, Figure 1C. The corresponding network's architecture, Figure 1F, lacks reciprocal connections among interneurons. As before, the number of above-threshold eigenvalues is chosen adaptively and cannot exceed the number of principal neurons. If the two are equal, this network whitens the output. In Section 4, we demonstrate that the online algorithms perform well on a synthetic dataset and, in Discussion, we compare our neural circuits with biological observations.

2

Dimensionality reduction in the offline setting

In this Section, we introduce and solve, in the offline setting, three novel optimization problems whose solutions reduce the dimensionality of the input. We state our results in three Theorems which are proved in the Supplementary Material. 2.1

Soft-thresholding of covariance eigenvalues

We consider the following optimization problem in the offline setting:

$$\min_Y \|X - Y\|_F^2 + \lambda \text{Tr}(Y^T Y) \quad (3)$$

where $\lambda \geq 0$ and I_T is the $T \times T$ identity matrix. To gain intuition behind this choice of the objective function let us expand the squared norm and keep only the Y -dependent terms:

$$\min_Y \|X - Y\|_F^2 + \lambda \text{Tr}(Y^T Y) = \min_Y \|X\|_F^2 - 2\text{Tr}(X^T Y) + \text{Tr}(Y^T Y) + \lambda \text{Tr}(Y^T Y) \quad (4)$$

where the first term matches the similarity of input and output [24] and the second term is a nuclear norm of Y known to be a convex relaxation of the matrix rank used for low-rank matrix modeling [28]. Thus, objective function (3) enforces low-rank similarity matching. We show that the optimal output Y is a projection of the input data, X , onto its principal subspace. The subspace dimensionality is set by m , the number of eigenvalues of the data covariance matrix, $P^T C = P^T X X^T P = \sum_{t=1}^T x_t x_t^T$, that are greater than or equal to the parameter λ . 3

Theorem 1. Suppose an eigen-decomposition of $X X^T = V \Lambda V^T$, where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_T)$ with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_T \geq 0$. Note that λ has at most n nonzero eigenvalues coinciding with those of $P^T C$. Then, $\hat{Y} = U_k S^k (\Lambda^{-1/2} V_k^T X)$, (5)

$X X^T$ are optima of (3), where $S^k(\lambda, \lambda) = \text{diag}(s_1, \dots, s_T)$, $s_k = \max(\lambda - \lambda_k, 0)$, V_k consists of the columns of V corresponding to the top k eigenvalues, i.e. $V_k = [v_1, \dots, v_k]$ and U_k is any $k \times k$ orthogonal matrix, i.e. $U_k^T U_k = I(k)$. The form (5) uniquely defines all optima of (3), except when $k = m$, $\lambda_k = \lambda_{k+1}$ and $X X^T k = \lambda_{k+1}$. 2.2

Hard-thresholding of covariance eigenvalues

Consider the following minimax problem in the offline setting:

$$\min_Y \max_Z \|X - Y\|_F^2 + \lambda \text{Tr}(Y^T Y) + \mu \text{Tr}(Z^T Z) \quad (6)$$

where $\mu \geq 0$ and we introduced an internal variable Z , which is an $l \times T$ matrix $Z = [z_1, \dots, z_T]$ with $z_t \in \mathbb{R}^l$. The intuition behind this objective function is again based on similarity matching but rank regularization is applied indirectly via the internal variable, Z . 4

Theorem 2. Suppose an eigen-decomposition of $X X^T = V \Lambda V^T$, where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_T)$ with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_T \geq 0$. Assume $l \leq \min(k, m)$. Then, $\hat{Y} =$

$$Y = U_k H T_k (\lambda X, \lambda T)^{1/2} V_k X,$$

$$Z = U_l S T_{l, \min(k, m)} (\lambda X, \lambda T)^{1/2} V_l X, \quad (7)$$
 $X X X$ are optima of (6), where $H T_k (\lambda, \lambda T) = \text{diag } H T \lambda^1, \lambda T, \dots, H T \lambda^k, \lambda T, H T(a, b) = a \lambda(a \lambda b)$ with $\lambda()$ being the step function: $\lambda(a \lambda b) = 1$ if $a \lambda b$ and $\lambda(a \lambda b) = 0$ if $a \lambda$

$$X X X b, S T_{l, \min(k, m)} (\lambda, \lambda T) = \text{diag } S T \lambda^1, \lambda T, \dots, S T \lambda^{\min(k, m)}, \lambda T, 0, \dots, 0, V_p X = \{z\}_{l=1}^{\min(k, m)} X$$
 $v_1, \dots, v_p X$ and $U_p \lambda O(p)$. The form (7) uniquely defines all optima (6) except when either $X \lambda^1 \lambda$ is an eigenvalue of C or $2) k \lambda m$ and $\lambda X k = \lambda^{k+1} \lambda 2.3$

Equalizing thresholded covariance eigenvalues

Consider the following minimax problem in the offline setting:

$$\min_Z \max_Y \text{Tr } \lambda X \lambda X Y \lambda Y + Y \lambda Y Z \lambda Z + \lambda T Y \lambda Y \lambda T Z \lambda Z, Y$$

(8)

where $\lambda \lambda 0$ and $\lambda \lambda 0$. This objective function follows from (6) after dropping the quartic Z term. λ

Theorem 3. Suppose of $X \lambda X$ is $X \lambda X = V X \lambda X V X$, where λX = an eigen-decomposition $X X X \text{diag } \lambda X, \dots, \lambda$ with $\lambda \lambda \dots \lambda \lambda 0$. Assume $l \lambda \min(k, m)$. Then, $1 \lambda T \lambda p \lambda \lambda Y = U_k \lambda T \lambda^k (\lambda X, \lambda T)^{1/2} V_k X, Z = U_l \lambda T O \lambda Y \lambda V X, (9)$

$X X X$ are optima of (8), where $\lambda k (\lambda, \lambda T) = \text{diag } \lambda \lambda^1 \lambda \lambda T, \dots, \lambda \lambda^k \lambda \lambda T, \lambda T$ is an $l \lambda T$ rectangular diagonal matrix with top $\min(k, m)$ diagonals are set to arbitrary nonnegative constants and the rest are zero, $O \lambda Y$ is a block-diagonal orthogonal matrix that has two blocks: the $\min(k, m)$ dimensional and the bottom block is $T \lambda \min(k, m)$ dimensional, $V_p = X$ top block X is v_1, \dots, v_p , and $U_p \lambda O(p)$. The form (9) uniquely defines all optima of (8) except when either $X \lambda^1 \lambda$ is an eigenvalue of C or $2) k \lambda m$ and $\lambda X k = \lambda^{k+1}$. Remark 1. If $k = m$, then Y is full-rank and equalizing variance across all channels.

3

$1 T$

$Y Y \lambda = \lambda I_k$, implying that the output is whitened,

Online dimensionality reduction using Hebbian/anti-Hebbian neural nets

In this Section, we formulate online versions of the dimensionality reduction optimization problems presented in the previous Section, derive corresponding online algorithms and map them onto the dynamics of neural networks with biologically plausible local learning rules. The order of subsections corresponds to that in the previous Section. 4

3.1

Online soft-thresholding of eigenvalues

Consider the following optimization problem in the online setting:

$$2 y^T \lambda \arg \min X \lambda X \lambda Y \lambda Y \lambda T I T F.$$

(10)

y^T

By keeping only the terms that depend on y^T we get the following objective for (2):
$$L = \sum_{t=1}^T 4x^T x_t y_t y^T + 2y^T y_t y^T + \sum_{t=1}^T 2kx^T k y^T k + ky^T k. \quad (11)$$

In the large- T limit, the last two terms can be dropped since the first two terms grow linearly with T and dominate. The remaining cost is a positive definite quadratic form in y^T and the optimization problem is convex. At its minimum, the following equality holds:
$$y^T = \sum_{t=1}^T y_t x^T. \quad (12)$$

While a closed-form analytical solution via matrix inversion exists for y^T , we are interested in biologically plausible algorithms. Instead, we use a weighted Jacobi iteration where y^T is updated according to:

$$y^T \leftarrow (1 - \eta) y^T + \eta \sum_{t=1}^T W^T Y X x^T + \eta \sum_{t=1}^T W^T Y Y y^T, \quad (13)$$

where η is the weight parameter, and $W^T Y X$ and $W^T Y Y$ are normalized input-output and output-output covariances, $\sum_{t=1}^T y_{t,i} x_{t,k} y_{t,i} y_{t,j} = \sum_{t=1}^T Y Y Y X$, $W^T_{i,k} = \sum_{t=1}^T y_{t,i} x_{t,k}$, $W^T_{i,j} = \sum_{t=1}^T y_{t,i} y_{t,j}$, $W^T_{i,i} = 0$.
$$y^T \leftarrow y^T + \sum_{t=1}^T y_{t,i} y_{t,i} \quad (14)$$

Iteration (13) can be implemented by the dynamics of neuronal activity in a single-layer network, Figure 1D. Then, $W^T Y X$ and $W^T Y Y$ represent the weights of feedforward ($x^T \rightarrow y^T$) and lateral ($y^T \rightarrow y^T$) synaptic connections, respectively. Remarkably, synaptic weights appear in the online solution despite their absence in the optimization problem formulation (3). Previously, nonnormalized covariances have been used as state variables in an online dictionary learning algorithm [29]. To formulate a fully online algorithm, we rewrite (14) in a recursive form. This requires introducing $Y Y =$ representing cumulative activity of a neuron i up to time $T - 1$, $DT_{i,j}$ a scalar variable $\sum_{t=1}^T y_{t,i} y_{t,j}$. Then, at each data sample presentation, T , after the output y^T converges to a steady state,

the following updates are performed: $Y Y \leftarrow Y Y + y_{T,i} y_{T,i}$, $DT_{i,j} \leftarrow DT_{i,j} + y_{T,i} y_{T,j}$, $W^T_{i,j} \leftarrow W^T_{i,j} + y_{T,i} x_{T,j}$, $W^T_{i,i} \leftarrow W^T_{i,i} - y_{T,i}^2$.
$$Y Y \leftarrow Y Y + y_{T,i} y_{T,i}, \quad W^T_{i,j} \leftarrow W^T_{i,j} + y_{T,i} x_{T,j}, \quad W^T_{i,i} \leftarrow W^T_{i,i} - y_{T,i}^2. \quad (15)$$

Hence, we arrive at a neural network algorithm that solves the optimization problem (10) for streaming data by alternating between two phases. After a data sample is presented at time T , in the first phase of the algorithm (13), neuron activities are updated until convergence to a fixed point. In the second phase of the algorithm, synaptic weights are updated for feedforward connections according to a local Hebbian rule (15) and for lateral connections according to a local anti-Hebbian rule (due to the $(-)$ sign in equation (13)). Interestingly, in the $\eta = 0$ limit, these updates have the same form as the single-neuron Oja rule

[24, 2], except that the learning rate is not a free parameter but is determined by the cumulative neuronal activity $1/DTY_{+1,i}$ [4, 5]. 3.2

Online hard-thresholding of eigenvalues

Consider the following minimax problem in the online setting, where we assume $\epsilon > 0$:

$$\begin{aligned} & \min_{\mathbf{Y}^T, \mathbf{Z}^T} \max_{\mathbf{X}_i, \mathbf{Y}_i, \mathbf{Z}_i} \sum_{t=1}^T \left(\mathbf{X}_i^T \mathbf{Y}_i - \mathbf{Y}_i^T \mathbf{Z}_i + \frac{\epsilon}{2} \left(\|\mathbf{X}_i\|^2 + \|\mathbf{Y}_i\|^2 + \|\mathbf{Z}_i\|^2 \right) \right) \\ & \text{subject to } \mathbf{X}_i^T \mathbf{X}_i = \mathbf{Y}_i^T \mathbf{Y}_i = \mathbf{Z}_i^T \mathbf{Z}_i = \mathbf{I} \end{aligned} \quad (16)$$

By keeping only those terms that depend on \mathbf{y}^T or \mathbf{z}^T and considering the large- T limit, we get the following objective:

$$\begin{aligned} L = & \sum_{t=1}^T \left(\mathbf{y}_t^T \mathbf{x}_t - \mathbf{y}_t^T \mathbf{z}_t + \frac{\epsilon}{2} \left(\|\mathbf{x}_t\|^2 + \|\mathbf{y}_t\|^2 + \|\mathbf{z}_t\|^2 \right) \right) \\ & \text{subject to } \mathbf{x}_t^T \mathbf{x}_t = \mathbf{y}_t^T \mathbf{y}_t = \mathbf{z}_t^T \mathbf{z}_t = \mathbf{I} \end{aligned} \quad (17)$$

Note that this objective is strongly convex in \mathbf{y}^T and strongly concave in \mathbf{z}^T . The solution of this minimax problem is the saddle-point of the objective function, which is found by setting the gradient of the objective with respect to $\{\mathbf{y}^T, \mathbf{z}^T\}$ to zero [30]:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{y}_t} &= \mathbf{x}_t - \mathbf{z}_t + \epsilon \mathbf{y}_t = 0 \\ \frac{\partial L}{\partial \mathbf{z}_t} &= -\mathbf{y}_t + \mathbf{x}_t - \epsilon \mathbf{z}_t = 0 \end{aligned}$$

To obtain a neurally plausible algorithm, we solve these equations by a weighted Jacobi iteration:

$\mathbf{y}^T \leftarrow (1 - \alpha) \mathbf{y}^T + \alpha \mathbf{W}^T \mathbf{X} \mathbf{x}^T + \alpha \mathbf{W}^T \mathbf{Z} \mathbf{z}^T$, $\mathbf{z}^T \leftarrow (1 - \alpha) \mathbf{z}^T + \alpha \mathbf{W}^T \mathbf{Y} \mathbf{y}^T + \alpha \mathbf{W}^T \mathbf{Z} \mathbf{z}^T$. (19) Here, similarly to (14), \mathbf{W}^T are normalized covariances that can be updated recursively: $\mathbf{Y}^T \mathbf{D}^T \mathbf{Y} + 1, i \leftarrow \mathbf{D}^T, i + \epsilon$,

$$\begin{aligned} & \mathbf{Z}^T \mathbf{D}^T \mathbf{Z} + 1, i \leftarrow \mathbf{D}^T, i + \epsilon + \mathbf{z}^T, i \\ & \mathbf{X}^T \mathbf{Y} \mathbf{X} + \mathbf{W}^T \mathbf{Y} + 1, i \leftarrow \mathbf{W}^T, i + \mathbf{y}^T, i \mathbf{x}^T, j + \epsilon \mathbf{W}^T, i / \mathbf{D}^T \mathbf{Y} + 1, i \\ & \mathbf{Z}^T \mathbf{Y} \mathbf{Z} + \mathbf{W}^T \mathbf{Y} + 1, i \leftarrow \mathbf{W}^T, i + \mathbf{y}^T, i \mathbf{z}^T, j + \epsilon \mathbf{W}^T, i / \mathbf{D}^T \mathbf{Y} + 1, i \\ & \mathbf{Z}^T \mathbf{Y} \mathbf{Z} + 1, i, j \leftarrow \mathbf{W}^T, i + \mathbf{z}^T, i \mathbf{y}^T, j + \epsilon \mathbf{W}^T, i / \mathbf{D}^T + 1, i \\ & \mathbf{Z}^T \mathbf{Z} \mathbf{Z} + \mathbf{W}^T \mathbf{Z} \mathbf{Z} \mathbf{W}^T, ii = 0. \quad (20) \end{aligned}$$

$+ \mathbf{z}^T, i \mathbf{W}^T, i / \mathbf{D}^T + 1, i$, Equations (19) and (20) define an online algorithm

that can be naturally implemented by a neural network with two populations of neurons: principal and interneurons, Figure 1E. Again, after each data sample presentation, T , the algorithm proceeds in two phases. First, (19) is iterated until convergence by the dynamics of neuronal activities. Second, synaptic weights are updated according to local, anti-Hebbian (for synapses from interneurons) and Hebbian (for all other synapses) rules.

3.3

Online thresholding and equalization of eigenvalues

Consider the following minimax problem in the online setting, where we assume $\lambda > 0$ and $\mu > 0$:

$$\{y^T, z^T\} \arg \min \arg \max \text{Tr} \{X_L^T X Y_L^T Y + Y_L^T Y Z_L^T Z + \lambda Y_L^T Y + \mu Z_L^T Z\} \quad (21)$$

By keeping only those terms that depend on y^T or z^T and considering the large- T limit, we get the following objective:
$$\sum_{t=1}^T \text{Tr} \{X_L^T X X_L^T X + Y_L^T Y Y_L^T Y + \lambda Y_L^T Y + \mu Z_L^T Z\} \quad (22)$$

This objective is strongly convex in y^T and strongly concave in z^T and its saddle point is given by:
$$\sum_{t=1}^T \text{Tr} \{X_L^T X X_L^T X + Y_L^T Y Y_L^T Y + \lambda Y_L^T Y + \mu Z_L^T Z\} \quad (23)$$

To obtain a neurally plausible algorithm, we solve these equations by a weighted Jacobi iteration:

$y^T \leftarrow (1 - \alpha) y^T + \alpha W^T Y X X^T W Y Z z^T, z^T \leftarrow (1 - \beta) z^T + \beta W^T Z Y y^T$, (24) As before, W are normalized covariances which can be updated recursively: $Y^T D^T Y + 1, i \leftarrow D^T, i + \alpha$,

$$Z^T D^T Z + 1, i \leftarrow D^T, i + \beta$$

$$Y X Y X Y X W^T + 1, ij \leftarrow W^T, ij + y^T, i x^T, j \leftarrow W^T, ij / D^T Y + 1, i$$

$$Z Y Z Y Z W^T + 1, ij \leftarrow W^T, ij + y^T, i z^T, j \leftarrow W^T, ij / D^T Y + 1, i$$

$$Z Y Z Y Z W^T Z Y (25) + 1, ij \leftarrow W^T, ij + z^T, i y^T, j \leftarrow W^T, ij / D^T + 1, i$$

. Equations (24) and (25) define an online algorithm that can be naturally implemented by a neural network with principal neurons and interneurons. As before, after each data sample presentation at

6

10

10 1 10

102 103 104

T

? T -1.56

-1

10-2 10 10-4

-3

1

10

102 103 104

T
 C ?T ? T -1.80 10
 102 103 104
 T
 10
 ? T -1.53
 1 10-1
 ?T
 10-2
 -1.33
 -1.43
 10-3 1
 10
 Eigenvalue Error
 ?T
 -1.50
 103 102 10 1 10-1 10-2 10-3 1
 102 103 104
 T
 Subspace Error
 B Eigenvalue Error
 103 102 10 1 10-1 10-2 10-3 1
 Subspace Error
 Subspace Error
 Eigenvalue Error
 A
 103 102 10 1 10-1 10-2
 ? T -1.48
 1
 10
 102 103 104
 T
 10 1 10-1 10-2 10-3 1
 ? T -1.41 ? T -1.38 10
 102 103 104
 T

Figure 2: Performance of the three neural networks: soft-thresholding (A), hard-thresholding (B), equalization after thresholding (C). Top: eigenvalue error, bottom: subspace error as a function of data presentations. Solid lines - means and shades - stds over 10 runs. Red - principal, blue inter-neurons. Dashed lines - best-fit power laws. For metric definitions see text. time T , the algorithm, first, iterates (24) by the dynamics of neuronal activities until convergence and, second, updates synaptic weights according to local anti-Hebbian (for synapses from interneurons) and Hebbian (25) (for all other synapses) rules. While an algorithm similar to (24), (25), but with predetermined learning rates, was previously given in [15, 14], it has not been derived from an optimization

problem. Plumbley's convergence analysis of his algorithm [14] suggests that at the fixed point of synaptic updates, the interneuron activity is also a projection onto the principal subspace. This result is a special case of our offline solution, (9), supported by the online numerical simulations (next Section).

4

Numerical simulations

Here, we evaluate the performance of the three online algorithms on a synthetic dataset, which is generated by an $n = 64$ dimensional colored Gaussian process with a specified covariance matrix. In this covariance matrix, the eigenvalues, $\lambda = \{5, 4, 3, 2\}$ and the remaining 55.60 are chosen uniformly from the interval $[0, 0.5]$. Correlations are introduced in the covariance matrix by generating random orthonormal eigenvectors. For all three algorithms, we choose $\eta = 1$ and, for the equalizing algorithm, we choose $\eta = 1$. In all simulated networks, the number of principal neurons, $k = 20$, and, for the hard-thresholding and the equalizing algorithms, the number of interneurons, $l = 5$. Synaptic weight matrices were initialized randomly, and synaptic update learning rates, $\eta \sum 1/D_{0,i}$ and $1/D_{0,i}$ were initialized to 0.1 . Network dynamics is run with a weight $\eta = 0.1$ until the relative change in y^T and z^T in one cycle is $\leq 10^{-5}$. To quantify the performance of these algorithms, we use two different metrics. The first metric, eigenvalue error, measures the deviation of output covariance eigenvalues from their optimal offline values given in Theorems 1, 2 and 3. The eigenvalue error at time T is calculated by summing squared differences between the eigenvalues of $T^T Y Y^T$ or $T^T Z Z^T$, and their optimal offline values at time T . The second metric, subspace error, quantifies the deviation of the learned subspace from the true principal subspace. To form such metric, at each T , we calculate the linear transformation that maps inputs, x^T , to outputs, $y^T = F Y^T X x^T$ and $z^T = F Z^T X x^T$, at the fixed points of the neural dynamics stages ((13), (19), (24)) of the three algorithms. Exact expressions for these matrices for all algorithms are given in the Supplementary Material. Then, at each T , the deviation

$\|X X^T - F_m^T F_{l,m}^T U_m^T U_m^T F\|_F$, where F_m^T is an $n \times m$ matrix whose columns are the top m right singular vectors of F^T , $F_m^T F_{l,m}^T$ is the projection matrix to the subspace spanned by these X singular vectors, U_m^T is an $n \times m$ matrix whose columns are the principal eigenvectors of the input X , covariance matrix C at time T , $U X_{l,m}^T U_m^T$ is the projection matrix to the principal subspace. 7

Further numerical simulations comparing the performance of the soft-thresholding algorithm with $\eta = 0$ with other neural principal subspace algorithms can be found in [24].

5

Discussion and conclusions

We developed a normative approach for dimensionality reduction by formulating three novel optimization problems, the solutions of which project the input onto its principal subspace, and rescale the data by i) soft-thresholding, ii) hard-thresholding, iii) equalization after thresholding of the input eigenvalues. Remarkably we found that these optimization problems can be solved online

using biologically plausible neural circuits. The dimensionality of neural activity is the number of either input covariance eigenvalues above the threshold, m , (if $m \leq k$) or output neurons, k (if $k > m$). The former case is ubiquitous in the analysis of experimental recordings, for a review see [31]. Interestingly, the division of neurons into two populations, principal and interneurons, in the last two models has natural parallels in biological neural networks. In biology, principal neurons and interneurons usually are excitatory and inhibitory respectively. However, we cannot make such an assignment in our theory, because the signs of neural activities, x^T and y^T , and, hence, the signs of synaptic weights, W , are unconstrained. Previously, interneurons were included into neural circuits [32], [33] outside of the normative approach. Similarity matching in the offline setting has been used to analyze experimentally recorded neuron activity lending support to our proposal. Semantically similar stimuli result in similar neural activity patterns in human (fMRI) and monkey (electrophysiology) IT cortices [34, 35]. In addition, [36] computed similarities among visual stimuli by matching them with the similarity among corresponding retinal activity patterns (using an information theoretic metric). We see several possible extensions to the algorithms presented here: 1) Our online objective functions may be optimized by alternative algorithms, such as gradient descent, which map onto different circuit architectures and learning rules. Interestingly, gradient descent-ascent on convex-concave objectives has been previously related to the dynamics of principal and interneurons [37]. 2) Inputs coming from a non-stationary distribution (with time-varying covariance matrix) can be processed by algorithms derived from the objective functions where contributions from older data points are “forgotten”, or “discounted”. Such discounting results in higher learning rates in the corresponding online algorithms, even at large T , giving them the ability to respond to variations in data statistics [24, 4]. Hence, the output dimensionality can track the number of input dimensions whose eigenvalues exceed the threshold. 3) In general, the output of our algorithms is not decorrelated. Such decorrelation can be achieved by including a correlation-penalizing term in our objective functions [38]. 4) Choosing the threshold parameter τ requires an a priori knowledge of input statistics. A better solution, to be presented elsewhere, would be to let the network adjust such threshold adaptively, e.g. by filtering out all the eigenmodes with power below the mean eigenmode power. 5) Here, we focused on dimensionality reduction using only spatial, as opposed to the spatio-temporal, correlation structure. We thank L. Greengard, A. Sengupta, A. Grinshpan, S. Wright, A. Barnett and E. Pnevmatikakis.

2 References

- [1] David H Hubel. Eye, brain, and vision. Scientific American Library/Scientific American Books, 1995.
- [2] E Oja. Simplified neuron model as a principal component analyzer. *J Math Biol*, 15(3):267–273, 1982.
- [3] KI Diamantaras and SY Kung. Principal component neural networks: theory and applications.

John Wiley & Sons, Inc., 1996. [4] B Yang. Projection approximation subspace tracking. *IEEE Trans. Signal Process.*, 43(1):95?107, 1995. [5] T Hu, ZJ Towfic, C Pehlevan, A Genkin, and DB Chklovskii. A neuron as a signal processing device. In *Asilomar Conference on Signals, Systems and Computers*, pages 362?366. IEEE, 2013. [6] E Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5(6):927? 935, 1992. [7] R Arora, A Cotter, K Livescu, and N Srebro. Stochastic optimization for pca and pls. In *Allerton Conf. on Communication, Control, and Computing*, pages 861?868. IEEE, 2012. [8] J Goes, T Zhang, R Arora, and G Lerman. Robust stochastic principal component analysis. In *Proc. 17th Int. Conf. on Artificial Intelligence and Statistics*, pages 266?274, 2014.

8

[9] Todd K Leen. Dynamics of learning in recurrent feature-discovery networks. *NIPS*, 3, 1990. [10] P F?oldiak. Adaptive network for optimal linear feature extraction. In *Int. Joint Conf. on Neural Networks*, pages 401?405. IEEE, 1989. [11] TD Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*, 2(6):459?473, 1989. [12] J Rubner and P Tavan. A self-organizing network for principal-component analysis. *EPL*, 10:693, 1989. [13] MD Plumbley. A hebbian/anti-hebbian network which optimizes information capacity by orthonormalizing the principal subspace. In *Proc. 3rd Int. Conf. on Artificial Neural Networks*, pages 86?90, 1993. [14] MD Plumbley. A subspace network that determines its own output dimension. *Tech. Rep.*, 1994. [15] MD Plumbley. Information processing in negative feedback neural networks. *Network-Comp Neural*, 7(2):301?305, 1996. [16] P Vertechi, W Brendel, and CK Machens. Unsupervised learning of an efficient short-term memory network. In *NIPS*, pages 3653?3661, 2014. [17] BA Olshausen and DJ Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Res*, 37(23):3311?3325, 1997. [18] AA Koulakov and D Rinberg. Sparse incomplete representations: a potential role of olfactory granule cells. *Neuron*, 72(1):124?136, 2011. [19] S Druckmann, T Hu, and DB Chklovskii. A mechanistic model of early sensory processing based on subtracting sparse representations. In *NIPS*, pages 1979?1987, 2012. [20] AL Fairhall, GD Lewen, W Bialek, and RRR van Steveninck. Efficiency and ambiguity in an adaptive neural code. *Nature*, 412(6849):787?792, 2001. [21] SE Palmer, O Marre, MJ Berry, and W Bialek. Predictive information in a sensory population. *PNAS*, 112(22):6908?6913, 2015. [22] E Doi, JL Gauthier, GD Field, J Shlens, et al. Efficient coding of spatial information in the primate retina. *J Neurosci*, 32(46):16256?16264, 2012. [23] R Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105?117, 1988. [24] C Pehlevan, T Hu, and DB Chklovskii. A hebbian/anti-hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data. *Neural Comput*, 27:1461?1495, 2015. [25] G Young and AS Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19?22, 1938. [26] WS Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401?419, 1952. [27] HG Barrow and JML Budd. Automatic gain control by a basic neural circuit. *Artificial Neural Networks*,

2:433?436, 1992. [28] EJ Candès and B Recht. Exact matrix completion via convex optimization. *Found Comput Math*, 9(6):717?772, 2009. [29] J Mairal, F Bach, J Ponce, and G Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11:19?60, 2010. [30] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. [31] P Gao and S Ganguli. On simplicity and complexity in the brave new world of large-scale neuroscience. *Curr Opin Neurobiol*, 32:148?155, 2015. [32] M Zhu and CJ Rozell. Modeling inhibitory interneurons in efficient sensory coding models. *PLoS Comput Biol*, 11(7):e1004353, 2015. [33] PD King, J Zylberberg, and MR DeWeese. Inhibitory interneurons decorrelate excitatory cells to drive sparse code formation in a spiking model of v1. *J Neurosci*, 33(13):5475?5485, 2013. [34] N Kriegeskorte, M Mur, DA Ruff, R Kiani, et al. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60(6):1126?1141, 2008. [35] R Kiani, H Esteky, K Mirpour, and K Tanaka. Object category structure in response patterns of neuronal population in monkey inferior temporal cortex. *J Neurophysiol*, 97(6):4296?4309, 2007. [36] G Tkaczik, E Granot-Atedgi, R Segev, and E Schneidman. Retinal metric: a stimulus distance measure derived from population neural responses. *PRL*, 110(5):058104, 2013. [37] HS Seung, TJ Richardson, JC Lagarias, and JJ Hopfield. excitatory-inhibitory networks. *NIPS*, 10:329?335, 1998.

Minimax and hamiltonian dynamics of

[38] C Pehlevan and DB Chklovskii. Optimization theory of hebbian/anti-hebbian networks for pca and whitening. In *Allerton Conf. on Communication, Control, and Computing*, 2015.