Optimization theory of Hebbian/anti-Hebbian networks for PCA and whitening

Cengiz Pehlevan[1] and Dmitri B. Chklovskii[1]

Abstract— In analyzing information streamed by sensory organs, our brains face challenges similar to those solved in statistical signal processing. This suggests that biologically plausible implementations of online signal processing algorithms may model neural computation. Here, we focus on such workhorses of signal processing as Principal Component Analysis (PCA) and whitening which maximize information transmission in the presence of noise. We adopt the similarity matching framework, recently developed for principal subspace extraction, but modify the existing objective functions by adding a decorrelating term. From the modified objective functions, we derive online PCA and whitening algorithms which are implementable by neural networks with local learning rules, i.e. synaptic weight updates that depend on the activity of only pre- and postsynaptic neurons. Our theory offers a principled model of neural computations and makes testable predictions such as the dropout of underutilized neurons.

I. Introduction

Principal Component Analysis (PCA) plays an important role in statistical signal processing by denoising data, identifying important features, and simplifying further processing. Mathematically, PCA computes the eigenvectors corresponding to the top eigenvalues of the data covariance matrix and projects data onto them. PCA algorithms exist for both the offline setting, where a whole dataset is available to the algorithm from the outset, and the online setting, where input data samples are presented to the algorithm sequentially, one at a time, and the corresponding output is computed prior to the presentation of the next input [1], [2], [3]. Given this, we expect that PCA algorithms, especially in the online setting, model some aspects of biological neural computation.

The first principal component of streamed data can be computed by a highly simplified model of a *single* neuron. Suppose that each input data sample is represented by the activity vector of upstream neurons at a corresponding time point. By summing these activities with the weights of corresponding synapses a neuron projects each data sample onto the vector of synaptic weights and transmits the projection to downstream neurons via its output activity. If synaptic weights are updated after each data sample presentation according to the Oja learning rule, a neuron computes the top eigenvector of the covariance matrix and outputs the first principle component [4], [5]. Here, we ignore temporal correlations in activity and assume that the dataset is presented as a sequence of static "snapshots" streamed in an arbitrary order.

The Oja learning rule has two major attractions for modeling neural computation. First, it can be derived, along with the weighted summation of inputs, from a principled objective function, by alternating minimization of a sum of squared representation errors with respect to activity and synaptic weights [6]. Second, Oja learning is Hebbian, meaning that the weight update depends on the activity of only pre- and postsynaptic neurons, and hence biologically plausible.

In order to extract *multiple* principal components from streamed data, researchers attempted to construct networks of multiple neurons, the activity of each representing a different principal component. However, most attempts have given up one of the two attractions of the single-neuron Oja rule. Instead of deriving a local learning rule from a principled objective function some researchers have simply postulated it [7], [8], [9], [10], [11]. Others, by minimizing the representation error, or its variants, derived single-layer neural networks with biologically implausible features such as nonlocal learning rules [12] or synapses that take part in plasticity but not in neural dynamics [10].

Recently, we developed a novel theoretical framework, named similarity matching, that preserves both attractions of the single-neuron Oja rule in the multi-neuron case [13], [14], [15], [16]. Similarity matching postulates that similar inputs result in similar outputs and vice versa. Mathematically, pairwise similarities are quantified by the inner products of data vectors and matching is enforced by the classical multidimensional scaling (CMDS) cost function [17]. We formulated a family of optimization problems and solved them in both offline and online settings. Importantly, the derived online algorithms correspond to a family of biologically plausible neural networks with local, Hebbian and anti-Hebbian, learning rules.

However, strictly speaking, the existing similarity matching algorithms [13], [14], [15], [16], as well as many others [8], [6], do not perform PCA. Rather, they extract the principal subspace of the dataset, i.e. the space spanned by the eigenvectors corresponding to the top eigenvalues of the data covariance matrix, and project the data onto an arbitrary basis spanning this subspace (not necessarily the top eigenvectors *per se*). Yet, an algorithm to perform PCA is desirable because unlike other principal subspace projections its output is decorrelated.

Whitening, or equalization of variance across decorrelated output channels, is desirable because in the presence of Gaussian output noise and for limited output power, it achieves maximum information transmission [18], [19], [20].

[1] Cengiz Pehlevan and Dmitri B. Chklovskii are with the Simons Center for Data Analysis, 160 Fifth Ave, New York, NY 10010. cpehlevan,mitya@simonsfoundation.org

In neuroscience, the center-surround structure of retinal ganglion cell receptive fields is thought to implement whitening [21], [22].

In this paper, motivated by the optimal information transmission, we derive algorithms and networks for PCA and whitening in the similarity matching framework. The existing similarity matching objective functions [13], [16] do not necessarily perform PCA because they depend only on the Grammian of the output and hence are invariant to orthogonal rotations of the output. As PCA is unique among principal subspace projections in that it produces a decorrelated output, we break the symmetry of the objective functions by adding a decorrelating term favoring PCA.

We formulate and solve three optimization problems, each in online and offline settings. The solutions of the first and the second problems perform PCA of the input data. A common practice in PCA is to keep only a subset of principal components, containing the useful signal, for further processing. To this end, in the first problem, the number of output principle components is set by the smaller of the input and output numbers of channels. In the second problem, the number of output principle components is chosen adaptively, by hard-thresholding the eigenvalues of the data covariance matrix. The optimal solution of the third problem also chooses the number of output components adaptively by hard-thresholding but, in addition, whitens the output by equalizing the variance of orthogonal channels.

The paper is organized as follows. In Section II we formulate optimization problems in the offline setting and present their solutions. In Section III we derive corresponding online algorithms and demonstrate that they can be implemented by biologically plausible neural circuits. The performance of these online algorithms is evaluated numerically in Section IV. In Section V we predict that underutilized neurons drop out of the circuit. Section VI comments on decorrelating interneuron activities.

II. PCA AND WHITENING IN THE OFFLINE SETTING

In this Section, we introduce and solve three novel optimization problems in the offline setting:

$$\text{Offline setting}: \mathbf{Y} \leftarrow \arg\min_{\mathbf{Y}} L\left(\mathbf{X}, \mathbf{Y}\right), \qquad (1)$$

where the input, $\mathbf{X} = \left[\mathbf{x}_1, \ldots, \mathbf{x}_T\right]$ is an $n \times T$ matrix with T centered input data samples in $\mathbb{R}^n$ as its columns and the output, $\mathbf{Y} = \left[\mathbf{y}_1, \ldots, \mathbf{y}_T\right]$ is a $k \times T$ matrix with corresponding outputs in $\mathbb{R}^k$ as its columns. In this Section, we assume $T \geq n$ and $T \geq k$ for convenience, however our results could be generalized easily.

A. Similarity matching cost function for PCA

To formulate similarity matching mathematically, we minimize the summed squared differences between all pairwise similarities, the so-called CMDS cost function [17], [16]:

$$\min_{\mathbf{Y}} \left\|\mathbf{X}^\top\mathbf{X} - \mathbf{Y}^\top\mathbf{Y}\right\|_F^2. \qquad (2)$$

Optimal solutions of the CMDS cost function (2) are projections of the input dataset $\mathbf{X}$ onto its principal subspace

[17], [13]. Suppose the eigen-decomposition of $\mathbf{X}^\top\mathbf{X} = \mathbf{V}^X\mathbf{\Lambda}^X\mathbf{V}^{X^\top}$, where $\mathbf{\Lambda}^X = \text{diag}(\lambda_1^X, ..., \lambda_T^X)$ with $\lambda_1^X \geq ... \geq \lambda_T^X \geq 0$ are ordered eigenvalues of $\mathbf{X}^\top\mathbf{X}$. Then the following is a solution of (2):

$$\mathbf{Y} = \mathbf{U}_k\mathbf{\Lambda}_k^{Y\,1/2}\mathbf{V}_k^{X\top}, \qquad (3)$$

where $\mathbf{\Lambda}_k^Y$ is a $k \times k$ diagonal matrix whose non-zero diagonals are $\left\{\lambda_1^X, ..., \lambda_{\min(k,n)}^X\right\}$, $\mathbf{V}_k^X$ consists of the columns of $\mathbf{V}^X$ corresponding to the top k eigenvalues, $\mathbf{V}_k^X = \left[\mathbf{v}_1^X, \ldots, \mathbf{v}_k^X\right]$, and $\mathbf{U}_k$ is any $k \times k$ orthogonal matrix.

However, the solution of (2) is not unique: because the objective function depends on the output only via its Grammian, it is invariant to an orthogonal left-rotation of $\mathbf{Y}$. This degree of freedom corresponds to an arbitrary choice of an orthogonal matrix $\mathbf{U}_k$ in (3).

To eliminate this degree of freedom we take advantage of the fact that the only way to decorrelate the output, i.e. obtain diagonal output covariance matrix, is to compute *bona fide* principal components. Thus, we add to the objective the sum squared of the off-diagonal elements of the output covariance matrix, $\frac{1}{T}\mathbf{Y}\mathbf{Y}^\top = \frac{1}{T}\mathbf{U}_k\mathbf{\Lambda}_k^Y\mathbf{U}_k^\top$:

$$\min_{\mathbf{Y}} \left[\left\|\mathbf{X}^\top\mathbf{X} - \mathbf{Y}^\top\mathbf{Y}\right\|_F^2 + \gamma\left\|\text{off}\left(\mathbf{Y}\mathbf{Y}^\top\right)\right\|_F^2\right], \quad (4)$$

where $\gamma > 0$ and the operator $\text{off}()$ extracts off-diagonal elements of a matrix by setting diagonal elements to 0.

Eq. (4) defines an objective function for PCA, where data is projected onto top $\min(k, n)$ principal eigenvectors. Indeed, for correlated solutions (3) $\gamma\left\|\text{off}\left(\mathbf{Y}\mathbf{Y}^\top\right)\right\|_F^2$ is positive, thus resulting in suboptimal values of the objective. For uncorrelated solutions, $\gamma\left\|\text{off}\left(\mathbf{Y}\mathbf{Y}^\top\right)\right\|_F^2$ vanishes and therefore does not affect the value of the objective. Note that, if $k > n$ decorrelation implies that $k - n$ output channels are silent.

B. Objective function for adaptive PCA

One drawback of the PCA objective function (4) is that the number of output dimensions must be chosen prior to the presentation of the first data sample. In real-life situations, and, especially, neuroscience context, the input signal-to-noise ratio may not be known and the output dimensionality must adapt automatically. Such adaptive dimensionality reduction solution may be obtained by solving the following minimax problem [16]:

$$\min_{\mathbf{Y}}\max_{\mathbf{Z}} \left[\left\|\mathbf{X}^\top\mathbf{X} - \mathbf{Y}^\top\mathbf{Y}\right\|_F^2 - \left\|\mathbf{Y}^\top\mathbf{Y} - \mathbf{Z}^\top\mathbf{Z}\right\|_F^2\right.$$
$$\left. +2\alpha T\,\text{Tr}\left(\mathbf{Y}^\top\mathbf{Y}\right) - 2\alpha T\,\text{Tr}\left(\mathbf{Z}^\top\mathbf{Z}\right)\right], \quad (5)$$

where we introduced an auxiliary variable $\mathbf{Z} = \left[\mathbf{z}_1, \ldots, \mathbf{z}_T\right] \in \mathbb{R}^{l x T}$.

The number of output dimensions, $m = \text{rank}(\mathbf{Y})$, is determined by the trade off between the similarity matching and the regularization terms. Whereas higher rank reduces the matching error it adds to the regularizer, $\text{Tr}\left(\mathbf{Y}^\top\mathbf{Y}\right)$, because the regularizer is a nuclear norm of the Grammian, $\mathbf{Y}^\top\mathbf{Y}$, which is a convex relaxation of rank. Then, the number of output dimensions, m, is the number of eigenvalues of the
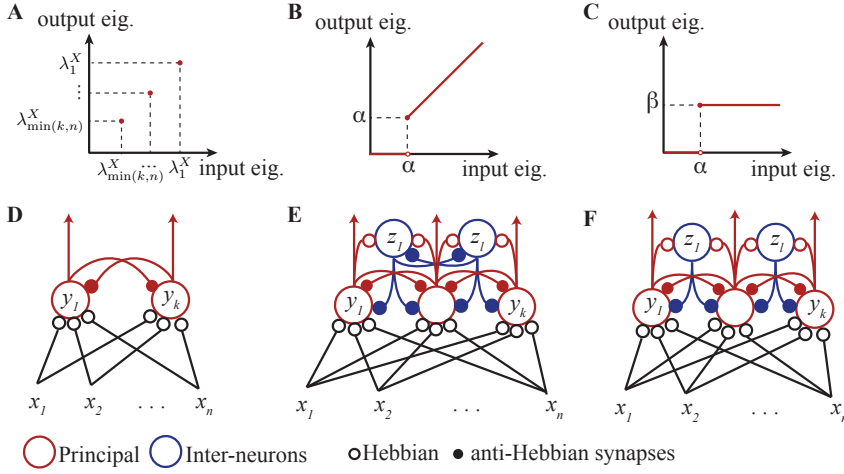
Fig. 1: Input-output functions of the three offline solutions and neural network implementations of the corresponding online algorithms. **A-C.** Input-output functions of covariance eigenvalues. **A.** Thresholding of top $\min(k, n)$ components. **B.** Adaptive hard-thresholding. **C.** Whitening after thresholding. **D-F.** Corresponding network architectures.

input data covariance matrix, $\mathbf{C} = \frac{1}{T}\mathbf{X}\mathbf{X}^\top$, greater than or equal to $\alpha > 0$. We assume that $k \geq m$ and $l \geq m$.

Objective (5) is solved by projecting the input dataset onto the m-dimensional principal subspace of input covariance [16]. Specifically, suppose the eigen-decomposition of $\mathbf{X}^\top\mathbf{X} = \mathbf{V}^X\mathbf{\Lambda}^X\mathbf{V}^{X^\top}$ where $\mathbf{\Lambda}^X = \mathrm{diag}(\lambda_1^X, ..., \lambda_T^X)$ with $\lambda_1^X \geq ... \geq \lambda_T^X \geq 0$ are ordered eigenvalues of $\mathbf{X}^\top\mathbf{X}$, as before. Then, the optimal $\mathbf{Y}$ and $\mathbf{Z}$ are:

$$\mathbf{Y} = \mathbf{U}_k \, \mathbf{HT}_k(\mathbf{\Lambda}^X, \alpha T)^{1/2} \, \mathbf{V}_k^{X^\top},$$
$$\mathbf{Z} = \mathbf{U}_l \, \mathbf{ST}_l(\mathbf{\Lambda}^X, \alpha T)^{1/2} \, \mathbf{V}_l^{X^\top}, \qquad (6)$$

where $\mathbf{HT}_k(\mathbf{\Lambda}^X, \alpha T) = \mathrm{diag}\left(\mathrm{HT}\left(\lambda_1^X, \alpha T\right), \dots, \mathrm{HT}\left(\lambda_k^X, \alpha T\right)\right)$, HT is the hard-thresholding function, $\mathrm{HT}(a, b) = a\Theta(a - b)$ with $\Theta()$ being the step function: $\Theta(a - b) = 1$ if $a \geq b$ and $\Theta(a - b) = 0$ if $a < b$, $\mathbf{ST}_l(\mathbf{\Lambda}^X, \alpha T) = \mathrm{diag}\left(\mathrm{ST}\left(\lambda_1^X, \alpha T\right), \dots, \mathrm{ST}\left(\lambda_l^X, \alpha T\right)\right)$, ST is the soft-thresholding function, $\mathrm{ST}(a, b) = \max(a - b, 0)$, $\mathbf{V}_p^X = \left[\mathbf{v}_1^X, \dots, \mathbf{v}_p^X\right]$ and $\mathbf{U}_p$ is any $p \times p$ orthogonal matrix.

Similarly to the observation in the previous subsection, the solution of (5) is not unique because the objective function is invariant to orthogonal left-rotations of $\mathbf{Y}$. To obtain PCA as the unique optimal solution, as before, we add a term to the objective function that penalizes off-diagonal elements of the covariance matrix:

$$\min_{\mathbf{Y}} \max_{\mathbf{Z}} \left[\left\|\mathbf{X}^\top\mathbf{X} - \mathbf{Y}^\top\mathbf{Y}\right\|_F^2 - \left\|\mathbf{Y}^\top\mathbf{Y} - \mathbf{Z}^\top\mathbf{Z}\right\|_F^2 \right.$$
$$\left. +2\alpha T \, \mathrm{Tr}\left(\mathbf{Y}^\top\mathbf{Y}\right) - 2\alpha T \, \mathrm{Tr}\left(\mathbf{Z}^\top\mathbf{Z}\right) + \gamma \left\|\mathrm{off}\left(\mathbf{Y}\mathbf{Y}^\top\right)\right\|_F^2 \right], \qquad (7)$$

where $\gamma > 0$.

Eq. (4) defines an objective function for PCA, where data is projected onto top m principal eigenvectors. Indeed, for correlated solutions (6), $\gamma \left\|\mathrm{off}\left(\mathbf{Y}\mathbf{Y}^\top\right)\right\|_F^2$ is positive, thus resulting in suboptimal values of the objective. For uncorrelated solutions, $\gamma \left\|\mathrm{off}\left(\mathbf{Y}\mathbf{Y}^\top\right)\right\|_F^2$ vanishes and does not affect the value of the objective. If the number of eigenvalues of $\frac{1}{T}\mathbf{X}\mathbf{X}^\top$ greater than or equal to α is less

than the number of output channels, $m < k$, decorrelation implies that some output channels will be silent.

C. Objective function for whitening

Next we consider an objective function that leads to equalization of the ouptut covariance eigenvalues after thresholding [16]:

$$\min_{\mathbf{Y}} \max_{\mathbf{Z}} \mathrm{Tr}\left(-\mathbf{X}^\top\mathbf{X}\mathbf{Y}^\top\mathbf{Y} + \alpha T\mathbf{Y}^\top\mathbf{Y} \right.$$
$$\left. +\mathbf{Y}^\top\mathbf{Y}\mathbf{Z}^\top\mathbf{Z} - \beta T\mathbf{Z}^\top\mathbf{Z}\right), \qquad (8)$$

where $\alpha > 0$ controls the number of degrees of freedom in the output, m, and $\beta > 0$ sets magnitude of output eigenvalues. As before, we assume that $k \geq m$ and $l \geq m$.

Objective (8) is optimized by projecting the input dataset onto its principal subspace and equalizing the non-zero eigenvalues [16]. Specifically, suppose the eigen-decomposition of $\mathbf{X}^\top\mathbf{X} = \mathbf{V}^X\mathbf{\Lambda}^X\mathbf{V}^{X^\top}$, where $\mathbf{\Lambda}^X = \mathrm{diag}\left(\lambda_1^X, \dots, \lambda_T^X\right)$ with $\lambda_1^X \geq \dots \geq \lambda_T^X \geq 0$. Then, the optimal $\mathbf{Y}$ and $\mathbf{Z}$ are:

$$\mathbf{Y} = \mathbf{U}_k \, \sqrt{\beta T} \, \mathbf{\Theta}_k(\mathbf{\Lambda}^X, \alpha T) \, \mathbf{V}_k^{X^\top},$$
$$\mathbf{Z} = \mathbf{U}_l \, \mathbf{\Sigma}_l \, \mathbf{\Theta}_l(\mathbf{\Lambda}^X, \alpha T) \, \mathbf{V}_l^{X^\top}, \qquad (9)$$

where $\mathbf{\Sigma}_l = \mathrm{diag}\left(\sigma_1, \dots, \sigma_l\right)$ with σ_i arbitrary constants, $\mathbf{\Theta}_k(\mathbf{\Lambda}^X, \alpha T) = \mathrm{diag}\left(\Theta\left(\lambda_1^X - \alpha T\right), \dots, \Theta\left(\lambda_k^X - \alpha T\right)\right)$, $\mathbf{V}_p = \left[\mathbf{v}_1^X, \dots, \mathbf{v}_p^X\right]$ and $\mathbf{U}_p$ is any $p \times p$ orthogonal matrix. There are other optimal $\mathbf{Z}$, see [16] for full expressions.

As before, the solution of (8) is not unique. Even though eigenvalues are equalized, due to the freedom in choosing $\mathbf{U}_k$, the variances of output channels are not equal, generally. An exception is the case where $k = l = m$. Then $\mathbf{Y}$ is full-rank and $\frac{1}{T}\mathbf{Y}\mathbf{Y}^\top = \beta\mathbf{I}_k$, implying that the output is whitened. To obtain whitening as the unique optimal solution for general $k \geq m$ and $l \geq m$, following the arguments of the previous subsections, we add a term to the objective function that penalizes off-diagonal elements of the covariance matrix:

$$\min_{\mathbf{Y}} \max_{\mathbf{Z}} \left[\mathrm{Tr}\left(-\mathbf{X}^\top\mathbf{X}\mathbf{Y}^\top\mathbf{Y} + \alpha T\mathbf{Y}^\top\mathbf{Y} + \mathbf{Y}^\top\mathbf{Y}\mathbf{Z}^\top\mathbf{Z}\right.\right.$$
$$\left.\left. -\beta T\mathbf{Z}^\top\mathbf{Z}\right) + \frac{\gamma}{2}\left\|\mathrm{off}\left(\mathbf{Y}\mathbf{Y}^\top\right)\right\|_F^2\right], \quad (10)$$

where $\gamma > 0$.

Eq. (10) defines an objective function for whitening, which can be solved by projecting the input dataset onto top m principal eigenvectors with variance in each channel normalized to β. Indeed, for correlated solutions (9), $(\gamma/2) \left\| \text{off} \left(\mathbf{Y}\mathbf{Y}^\top \right) \right\|_F^2$ is positive, thus resulting in suboptimal values of the objective. For uncorrelated solutions, $(\gamma/2) \left\| \text{off} \left(\mathbf{Y}\mathbf{Y}^\top \right) \right\|_F^2$ vanishes and does not affect the value of the objective. Note that, if the number of eigenvalues of $\frac{1}{T}\mathbf{X}\mathbf{X}^\top$ greater than or equal to α, is less than the number of output channels, $m < k$, decorrelation implies that some output channels will be silent.

III. ONLINE LEARNING RULES FOR DECORRELATED OUTPUT

Unlike the offline setting where the whole input dataset is available before an output is computed, neurons compute output, $\mathbf{y}_T$, for each data sample presentation, $\mathbf{x}_T$, before the next data sample is presented and past outputs cannot be altered. Therefore, we formulate optimization problems in the online setting where optimization must be performed at every time step, T, on the objective which is a function of inputs and outputs up to time, T:

$$\text{Online setting} : \mathbf{y}_T \leftarrow \underset{\mathbf{y}_T}{\arg\min} \, L\left(\mathbf{X}, \mathbf{Y}\right). \quad (11)$$

In this Section, we solve the three optimization problems in the online setting and map the steps of the online algorithms onto the dynamics of neuronal activity and local learning rules for synaptic weights. Our derivations follow the methods described in detail [13], [16].

A. Online similarity matching for PCA

We start with an online version of the objective function (4):

$$\mathbf{y}_T \leftarrow \underset{\mathbf{y}_T}{\arg\min} \left[\left\| \mathbf{X}^\top \mathbf{X} - \mathbf{Y}^\top \mathbf{Y} \right\|_F^2 + \gamma \left\| \text{off} \left(\mathbf{Y}\mathbf{Y}^\top \right) \right\|_F^2 \right]. \quad (12)$$

By expanding the squared Frobenius norms and keeping only the terms that depend $\mathbf{y}_T$ we get:

$$\mathbf{y}_T \leftarrow \underset{\mathbf{y}_T}{\arg\min} \left[-4\mathbf{x}_T^\top \left(\sum_{t=1}^{T-1} \mathbf{x}_t \mathbf{y}_t^\top \right) \mathbf{y}_T \right.$$
$$+ 2\mathbf{y}_T^\top \left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{y}_t^\top + \gamma \, \text{off} \left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{y}_t^\top \right) \right) \mathbf{y}_T$$
$$\left. - 2\|\mathbf{x}_T\|^2 \|\mathbf{y}_T\|^2 + \|\mathbf{y}_T\|^4 + \gamma \mathbf{y}_T^\top \, \text{off} \left(\mathbf{y}_T \mathbf{y}_T^\top \right) \mathbf{y}_T \right]. \quad (13)$$

In the large-T limit, the first two terms grow linearly with T and dominate over the last three terms which can be dropped. The remaining objective is a positive definite quadratic form of $\mathbf{y}_T$ and the optimization problem is convex. At its minimum, the following holds:

$$\left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{y}_t^\top + \gamma \, \text{off} \left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{y}_t^\top \right) \right) \mathbf{y}_T = \left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{x}_t^\top \right) \mathbf{x}_T. \quad (14)$$

We could solve for $\mathbf{y}_T$ analytically via matrix inversion, however, to obtain a neurally plausible algorithm, we solve these equations by a weighted Jacobi iteration[1]:

$$\mathbf{y}_T \leftarrow (1 - \eta) \mathbf{y}_T + \eta \left(\mathbf{W}_T^{YX} \mathbf{x}_T - \mathbf{W}_T^{YY} \mathbf{y}_T \right). \quad (15)$$

where η is the weight parameter, and $\mathbf{W}_T^{YX}$ and $\mathbf{W}_T^{YY}$ are normalized input-output and output-output covariances,

$$W_{T,ij}^{YX} = \sum_{t=1}^{T-1} y_{t,i} x_{t,j} \Big/ \sum_{t=1}^{T-1} y_{t,i}^2,$$

$$W_{T,i,j\neq i}^{YY} = (1 + \gamma) \sum_{t=1}^{T-1} y_{t,i} y_{t,j} \Big/ \sum_{t=1}^{T-1} y_{t,i}^2, \quad W_{T,ii}^{YY} = 0. \quad (16)$$

Remarkably, iteration (15) can be implemented by neuronal dynamics in a single-layer network, Figure 1D. In this interpretation, $\mathbf{W}_T^{YX}$ and $\mathbf{W}_T^{YY}$ represent the weights of feedforward $(\mathbf{x}_t \rightarrow \mathbf{y}_t)$ and lateral $(\mathbf{y}_t \rightarrow \mathbf{y}_t)$ synaptic connections, respectively. Interestingly, although the optimization problems (2) and (12) are formulated only in terms of input and output activities, we recovered expressions naturally identified as feedforward and lateral synaptic weights.

At each data sample presentation, T, after the output $\mathbf{y}_T$ converges to a steady state, synaptic weights are updated according to (16). By rewriting (16) in a recursive form, we can eliminate the need to keep all past input and output in memory and obtain a fully online algorithm. To this end, let us define a scalar variable $D_{T,i}^Y$ representing cumulative activity of a neuron i up to time $T-1$,

$$D_{T,i}^Y = \sum_{t=1}^{T-1} y_{t,i}^2. \quad (17)$$

Then, synaptic weight updates are:

$$D_{T+1,i}^Y \leftarrow D_{T,i}^Y + y_{T,i}^2$$
$$W_{T+1,ij}^{YX} \leftarrow W_{T,ij}^{YX} + \left(y_{T,i} x_{T,j} - y_{T,i}^2 W_{T,ij}^{YX} \right) / D_{T+1,i}^Y$$
$$W_{T+1,i,j\neq i}^{YY} \leftarrow W_{T,ij}^{YY}$$
$$+ \left((1+\gamma) y_{T,i} y_{T,j} - y_{T,i}^2 W_{T,ij}^{YY} \right) / D_{T+1,i}^Y. \quad (18)$$

To summarize, equations (15) and (18) define a neural network algorithm that solves the optimization problem (12) for streaming data by alternating between two phases: neural activity dynamics and synaptic updates. After a data sample is presented at time T, the algorithm goes into the neuronal activity phase (15), where neuron activities are updated until convergence to a fixed point. Then, in the second phase of the algorithm, synaptic weights are updated, according to a local Hebbian rule (18) for feedforward connections, and according to a local anti-Hebbian rule (due to the $(-)$ sign in equation (15)) for lateral connections. Interestingly, these updates have the same form as the single-neuron Oja's rule [4], except that the learning rate is not a free parameter but is determined by the cumulative neuronal activity $1/D_{T+1,i}^Y$.

---

[1]See [13] for other possible iterative solutions and their convergence properties

A similar network was derived in [13] from the objective (12) without the decorrelating term, i.e. $\gamma = 0$. The addition of the decorrelating term did not spoil the locality of learning rules, nor did it change the network architecture. The only difference is the strengthening of lateral synaptic weights by a factor $(1 + \gamma)$ (16). Lateral connections implement competition between the output of neurons: without them, k output neurons would independently recover the first principal component [4]. Interestingly, the strengthening of lateral synapses is sufficient to decorrelate neuronal output and project the input to its principal eigenvectors, as opposed to an arbitrary basis in the principal subspace, as was the case in [13].

B. Online adaptive PCA

Next, we consider an online version of (5):

$$
\{\mathbf{y}_T, \mathbf{z}_T\} \leftarrow \arg\min_{\mathbf{y}_T} \arg\max_{\mathbf{z}_T} \left[\left\| \mathbf{X}^\top \mathbf{X} - \mathbf{Y}^\top \mathbf{Y} \right\|_F^2 \right.
$$
$$
\left. - \left\| \mathbf{Y}^\top \mathbf{Y} - \mathbf{Z}^\top \mathbf{Z} - \alpha T \mathbf{I}_T \right\|_F^2 + \gamma \left\| \mathrm{off}\left(\mathbf{Y}\mathbf{Y}^\top \right) \right\|_F^2 \right]. \quad (19)
$$

By expanding the norms and keeping only those terms that depend on $\mathbf{y}_T$ or $\mathbf{z}_T$ and taking the large-T limit, we get:

$$
\{\mathbf{y}_T, \mathbf{z}_T\} \leftarrow \arg\min_{\mathbf{y}_T} \arg\max_{\mathbf{z}_T} \left[-4\mathbf{x}_T^\top \left(\sum_{t=1}^{T-1} \mathbf{x}_t \mathbf{y}_t^\top \right) \mathbf{y}_T \right.
$$
$$
+ 2\mathbf{y}_T^\top \left(\gamma\, \mathrm{off}\left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{y}_t^\top \right) + \alpha T \mathbf{I}_k \right) \mathbf{y}_T
$$
$$
\left. + 4\mathbf{y}_T^\top \left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{z}_t^\top \right) \mathbf{z}_T - 2\mathbf{z}_T^\top \left(\sum_{t=1}^{T-1} \mathbf{z}_t \mathbf{z}_t^\top + \alpha T \mathbf{I}_l \right) \mathbf{z}_T \right]. \quad (20)
$$

This objective is a convex quadratic from in $\mathbf{y}_T$ and concave quadratic form in $\mathbf{z}_T$. The solution of this minimax problem is a 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 [23]:

$$
\left(\gamma\, \mathrm{off}\left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{y}_t^\top \right) + \alpha T \mathbf{I}_k \right) \mathbf{y}_T = \left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{x}_t^\top \right) \mathbf{x}_T
$$
$$
- \left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{z}_t^\top \right) \mathbf{z}_T,
$$
$$
\left(\sum_{t=1}^{T-1} \mathbf{z}_t \mathbf{z}_t^\top + \alpha T \mathbf{I}_l \right) \mathbf{z}_T = \left(\sum_{t=1}^{T-1} \mathbf{z}_t \mathbf{y}_t^\top \right) \mathbf{y}_T. \quad (21)
$$

We could solve these linear equations analytically, but to obtain a neurally plausible algorithm, we solve them using a weighted Jacobi iteration:

$$
\mathbf{y}_T \leftarrow (1 - \eta)\,\mathbf{y}_T + \eta \left(\mathbf{W}_T^{YX} \mathbf{x}_T - \mathbf{W}_T^{YZ} \mathbf{z}_T - \mathbf{W}_T^{YY} \mathbf{y}_T \right),
$$
$$
\mathbf{z}_T \leftarrow (1 - \eta)\,\mathbf{z}_T + \eta \left(\mathbf{W}_T^{ZY} \mathbf{y}_T - \mathbf{W}_T^{ZZ} \mathbf{z}_T \right), \quad (22)
$$

where η is the weight parameter and

$$
W_{T,ij}^{YX} = \frac{1}{\alpha T} \sum_{t=1}^{T-1} y_{t,i} x_{t,j}, \quad W_{T,ij}^{YZ} = \frac{1}{\alpha T} \sum_{t=1}^{T-1} y_{t,i} z_{t,j}
$$
$$
W_{T,i,j \neq i}^{YY} = \frac{\gamma}{\alpha T} \sum_{t=1}^{T-1} y_{t,i} y_{t,j}, \qquad W_{T,ii}^{YY} = 0,
$$
$$
W_{T,ij}^{ZY} = \sum_{t=1}^{T-1} z_{t,i} y_{t,j} \Big/ \left(\alpha T + \sum_{t=1}^{T-1} z_{t,i}^2 \right),
$$
$$
W_{T,i,j \neq i}^{ZZ} = \sum_{t=1}^{T-1} z_{t,i} z_{t,j} \Big/ \left(\alpha T + \sum_{t=1}^{T-1} z_{t,i}^2 \right), \quad W_{T,ii}^{ZZ} = 0. \quad (23)
$$

Iteration (22) can be implemented by neuronal dynamics in a single-layer two-population network, Figure 1E. In this interpretation, $\mathbf{y}_T$ represents the activities of output neurons, which we identify with principal neurons in neuroscience terminology. Similarly, $\mathbf{z}_T$ represents the activities of neurons which connect only within the layer, which we identify with interneurons in neuroscience terminology. Again, although the optimization problems (5) and (19) did not contain synaptic weights explicitly, we recovered expressions $\mathbf{W}_T^{YX}$, $\mathbf{W}_T^{YY}$, $\mathbf{W}_T^{ZY}$, $\mathbf{W}_T^{YZ}$ and $\mathbf{W}_T^{ZZ}$ that are naturally identified as the weights of synaptic connections in the network.

Finally, by rewriting (23) in a recursive form, we obtain a fully online algorithm:

$$
D_{T+1,i}^Y \leftarrow D_{T,i}^Y + \alpha, \qquad D_{T+1,i}^Z \leftarrow D_{T,i}^Z + \alpha + z_{T,i}^2
$$
$$
W_{T+1,ij}^{YX} \leftarrow W_{T,ij}^{YX} + \left(y_{T,i} x_{T,j} - \alpha W_{T,ij}^{YX} \right) / D_{T+1,i}^Y
$$
$$
W_{T+1,ij}^{YZ} \leftarrow W_{T,ij}^{YZ} + \left(y_{T,i} z_{T,j} - \alpha W_{T,ij}^{YZ} \right) / D_{T+1,i}^Y
$$
$$
W_{T+1,ij \neq i}^{YY} \leftarrow W_{T,ij}^{YY} + \left(\gamma y_{T,i} y_{T,j} - \alpha W_{T,ij}^{YY} \right) / D_{T+1,i}^Y
$$
$$
W_{T+1,ij}^{ZY} \leftarrow W_{T,ij}^{ZY}
$$
$$
+ \left(z_{T,i} y_{T,j} - \left(\alpha + z_{T,i}^2 \right) W_{T,ij}^{ZY} \right) / D_{T+1,i}^Z
$$
$$
W_{T+1,i,j \neq i}^{ZZ} \leftarrow W_{T,ij}^{ZZ}
$$
$$
+ \left(z_{T,i} z_{T,j} - \left(\alpha + z_{T,i}^2 \right) W_{T,ij}^{ZZ} \right) / D_{T+1,i}^Z. \quad (24)
$$

To summarize, equations (22) and (24) define a neural network algorithm that solves the optimization problem (19) for streaming data by alternating between two phases: neural activity dynamics and synaptic updates. After a data sample is presented at time T, the algorithm goes into the neuronal activity phase (22), where neuron activities are updated until convergence to a fixed point. Then, in the second phase of the algorithm, synaptic weights are updated, according to local Hebbian rules (24) for $\mathbf{W}_T^{YX}$ and $\mathbf{W}_T^{ZY}$ connections, and according to local anti-Hebbian rules for $\mathbf{W}_T^{YY}$, $\mathbf{W}_T^{YZ}$ and $\mathbf{W}_T^{ZZ}$ connections.

A similar network was derived in [16] from the objective (19) without the decorrelating term, i.e. $\gamma = 0$. The addition of the decorrelating term does not spoil the locality of learning rules, however, it changes the network architecture by adding anti-Hebbian lateral connections between principal neurons. These new lateral synapses decorrelate neuronal output, whereas in [16] the output was in general correlated.

In our discussion of the solutions to the offline objective (7), we observed that when the number of output channels is larger than the number of output eigenvalues, decorrelation forces extra channels to be silent. In the online case, synaptic weights to silent neurons will eventually decay to zero, as can be seen from inspecting (24).

C. Online whitening

Finally, we consider the following minimax problem in the online setting:

$$\{\mathbf{y}_T, \mathbf{z}_T\} \leftarrow \arg\min_{\mathbf{y}_T} \arg\max_{\mathbf{z}_T} \mathrm{Tr}\left(-\mathbf{X}^\top \mathbf{X} \mathbf{Y}^\top \mathbf{Y} + \alpha T \mathbf{Y}^\top \mathbf{Y}\right.$$
$$\left. + \mathbf{Y}^\top \mathbf{Y} \mathbf{Z}^\top \mathbf{Z} - \beta T \mathbf{Z}^\top \mathbf{Z}\right) + \frac{\gamma}{2} \left\| \mathrm{off}\left(\mathbf{Y}\mathbf{Y}^\top\right)\right\|_F^2. \tag{25}$$

By keeping only those terms that depend on $\mathbf{y}_T$ or $\mathbf{z}_T$ and taking the large-T limit, we get:

$$\{\mathbf{y}_T, \mathbf{z}_T\} \leftarrow \arg\min_{\mathbf{y}_T} \arg\max_{\mathbf{z}_T} \left[-2\mathbf{x}_T^\top \left(\sum_{t=1}^{T-1} \mathbf{x}_t \mathbf{y}_t^\top\right) \mathbf{y}_T \right.$$
$$+ \mathbf{y}_T^\top \left(\gamma\,\mathrm{off}\left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{y}_t^\top\right) + \alpha T \mathbf{I}_k\right) \mathbf{y}_T$$
$$\left. + 2\mathbf{y}_T^\top \left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{z}_t^\top\right) \mathbf{z}_T - \beta T \mathbf{z}_T^\top \mathbf{z}_T \right]. \tag{26}$$

As before, this objective is convex in $\mathbf{y}_T$ and concave in $\mathbf{z}_T$. The solution of this minimax problem is a saddle-point of the objective function:

$$\left(\gamma\,\mathrm{off}\left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{y}_t^\top\right) + \alpha T \mathbf{I}_k\right) \mathbf{y}_T = \left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{x}_t^\top\right) \mathbf{x}_T$$
$$- \left(\sum_{t=1}^{T-1} \mathbf{y}_t \mathbf{z}_t^\top\right) \mathbf{z}_T,$$
$$\beta T \mathbf{z}_T = \left(\sum_{t=1}^{T-1} \mathbf{z}_t \mathbf{y}_t^\top\right) \mathbf{y}_T. \tag{27}$$

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

$$\mathbf{y}_T \leftarrow (1-\eta)\,\mathbf{y}_T + \eta\left(\mathbf{W}_T^{YX}\mathbf{x}_T - \mathbf{W}_T^{YZ}\mathbf{z}_T - \mathbf{W}_T^{YY}\mathbf{y}_T\right),$$
$$\mathbf{z}_T \leftarrow (1-\eta)\,\mathbf{z}_T + \eta\mathbf{W}_T^{ZY}\mathbf{y}_T, \tag{28}$$

where,

$$W_{T,ij}^{YX} = \frac{1}{\alpha T}\sum_{t=1}^{T-1} y_{t,i} x_{t,j}, \quad W_{T,ij}^{YZ} = \frac{1}{\alpha T}\sum_{t=1}^{T-1} y_{t,i} z_{t,j},$$

$$W_{T,i,j\neq i}^{YY} = \frac{\gamma}{\alpha T}\sum_{t=1}^{T-1} y_{t,i} y_{t,j}, \qquad W_{T,ii}^{YY} = 0,$$

$$W_{T,ij}^{ZY} = \frac{1}{\beta T}\sum_{t=1}^{T-1} z_{t,i} y_{t,j}. \tag{29}$$

Again, iteration (28) can be implemented by neuronal dynamics in a single-layer two-population network, Figure 1F, where $\mathbf{y}_T$ represents the activity of principal neurons

and $\mathbf{z}_T$ represents the activities of interneurons. Once again, although the optimization problems (8) and (25) did not contain synaptic weights explicitly, we recovered expressions $\mathbf{W}_T^{YX}$, $\mathbf{W}_T^{YY}$, $\mathbf{W}_T^{ZY}$ and $\mathbf{W}_T^{YZ}$ which are naturally identified as the weights of synaptic connections in the network. Note that, unlike in (22), interneurons do not synapse with each other.

Finally, by rewriting (29) in a recursive form, we obtain a fully online algorithm:

$$D_{T+1,i}^Y \leftarrow D_{T,i}^Y + \alpha, \qquad D_{T+1,i}^Z \leftarrow D_{T,i}^Z + \beta$$
$$W_{T+1,ij}^{YX} \leftarrow W_{T,ij}^{YX} + \left(y_{T,i} x_{T,j} - \alpha W_{T,ij}^{YX}\right)/D_{T+1,i}^Y$$
$$W_{T+1,ij}^{YZ} \leftarrow W_{T,ij}^{YZ} + \left(y_{T,i} z_{T,j} - \alpha W_{T,ij}^{YZ}\right)/D_{T+1,i}^Y$$
$$W_{T+1,i,j\neq i}^{YY} \leftarrow W_{T,ij}^{YY} + \left(\gamma y_{T,i} y_{T,j} - \alpha W_{T,ij}^{YY}\right)/D_{T+1,i}^Y$$
$$W_{T+1,ij}^{ZY} \leftarrow W_{T,ij}^{ZY} + \left(z_{T,i} y_{T,j} - \beta W_{T,ij}^{ZY}\right)/D_{T+1,i}^Z. \tag{30}$$

To summarize, equations (28) and (30) define a neural network algorithm that solves the optimization problem (25) for streaming data by alternating between two phases: neural activity dynamics and synaptic updates. After a data sample is presented at time T, the algorithm goes into the neuronal activity phase (28), where neuron activities are updated until convergence to a fixed point. Then, in the second phase of the algorithm, synaptic weights are updated, according to local Hebbian rules (30) for $\mathbf{W}_T^{YX}$ and $\mathbf{W}_T^{ZY}$ connections, and according to local anti-Hebbian rules for $\mathbf{W}_T^{YY}$ and $\mathbf{W}_T^{YZ}$ connections.

A similar network was derived in [16] from the cost (25) without the decorrelating term, i.e. $\gamma = 0$. The addition of the decorrelating term does not spoil the locality of learning rules, however, it changes the network architecture by adding anti-Hebbian lateral connections between principal neurons. These new lateral synapses decorrelate neuronal output, whereas in [16] output was decorrelated only if the output was full rank, i.e. dimensionality of principal neural activity was the same as the number of principal neurons.

In our discussion of the solution to the offline whitening objective, (10), we observed that when the number of output channels is larger than the number of output eigenvalues, decorrelation forces extra channels to be silent. In the online case, synaptic weights to silent neurons will eventually decay to zero, as can be seen by inspecting (30).

IV. Numerical experiments

In this section, we evaluate the performance of the proposed algorithms on a synthetic dataset, which is generated by an $n = 64$ dimensional colored Gaussian distribution with a specified covariance matrix. The top 4 eigenvalues are $\lambda_{1..4} = \{7, 6, 5, 4\}$ and the remaining $\lambda_{5..60}$ are sampled 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 $\alpha = 1$, $\gamma = \{0, 0.5, 1\}$, and, for the whitening algorithm, we choose $\beta = 2$. In the $T \to \infty$ limit, the optimal non-zero offline eigenvalues are $\{7, 6, 5, 4\}$ for PCA and $\{2, 2, 2, 2\}$ for whitening. In all simulated networks, the number of principal

neurons, $k = 10$, and, for adaptive PCA and whitening algorithms, the number of interneurons, $l = 10$. Synaptic weight matrices were initialized randomly, and synaptic update learning rates, $1/D_{0,i}^Y$ and $1/D_{0,i}^Z$ were initialized to 0.01. Network dynamics is run with a weight $\eta = 0.1$ until the relative change in $\mathbf{y}_T$ and $\mathbf{z}_T$ in one cycle is $< 10^{-5}$.

We characterize the performance of our algorithms using three different metrics. The first metric, eigenvalue error, measures the deviation of the eigenvalues of the output covariance $\frac{1}{T}\mathbf{Y}\mathbf{Y}^\top$ at time T from their optimal offline values, $10\log_{10}\sum_{i=1}^T(\bar{\lambda}_{T,i}^Y - \bar{\lambda}_{\text{offline},i}^Y)^2$ dB. Here $\bar{\lambda}_{T,i}^Y$ is the ith eigenvalue of $\frac{1}{T}\mathbf{Y}\mathbf{Y}^\top$ and $\bar{\lambda}_{\text{offline},i}^Y$ is its optimal value. For all three algorithms, the eigenvalue error decreases with time, Figure 2. Note, however, that adding the decorrelating term, i.e. increasing γ leads to a slower decrease of the eigenvalue error.

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, $\mathbf{x}_T$, to outputs, $\mathbf{y}_T = \mathbf{F}_T\mathbf{x}_T$ at the fixed points of the neural dynamics stages of the three algorithms(15), (22), (28). For PCA $\mathbf{F}_T = (\mathbf{I}_k + \mathbf{W}_T^{YY})^{-1}\mathbf{W}_T^{YX}$, for adaptive PCA $\mathbf{F}_T = (\mathbf{I}_k + \mathbf{W}_T^{YY} + \mathbf{W}_T^{YZ}(\mathbf{I}_l + \mathbf{W}_T^{ZZ})^{-1}\mathbf{W}_T^{ZY})^{-1}\mathbf{W}_T^{YX}$, and for whitening $\mathbf{F}_T = (\mathbf{I}_k + \mathbf{W}_T^{YY} + \mathbf{W}_T^{YZ}\mathbf{W}_T^{ZY})^{-1}\mathbf{W}_T^{YX}$. Then, at each T, the subspace error is $10\log_{10}\left\|\mathbf{F}_{4,T}\mathbf{F}_{4,T}^\top - \mathbf{V}_{4,T}^X\mathbf{V}_{4,T}^{X}{}^\top\right\|_F^2$ dB, where $\mathbf{F}_{4,T}$ is an $n \times 4$ matrix whose columns are the top 4 right singular vectors of $\mathbf{F}_T$, $\mathbf{F}_{4,T}\mathbf{F}_{4,T}^\top$ is the projection matrix to the subspace spanned by these singular vectors, $\mathbf{V}_{4,T}^X$ is an $n \times 4$ matrix whose columns are the principal eigenvectors of the input covariance matrix $\mathbf{C}$, $\mathbf{V}_{4,T}^X\mathbf{V}_{4,T}^{X}{}^\top$ is the projection matrix to the principal subspace. Figure 2 shows that subspace error decreases quickly with time for all algorithms, however, increasing γ leads to a loss of performance for the adaptive PCA and whitening algorithms.

The third metric, decorrelation error, represents correlations among output channels: $10\log_{10}\left\|\frac{1}{T}\text{off}(\mathbf{Y}_T\mathbf{Y}_T^\top)\right\|_F^2$ dB, Figure 2. For $\gamma > 0$ output channels decorrelate with the rate increasing with γ.

For $\gamma = 0$, the observed output correlation approaches that for a random projection onto the principal subspace (horizontal dashed black lines in Figure 2). The decorrelation errors for random projections are averaged over a set of 100000 randomly generated $k \times k$ covariance matrices. Each instance of such covariance matrix is constructed from the eigenvalue decomposition, $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where diagonals of $\mathbf{\Lambda}$ contain optimal offline eigenvalues in $T \to \infty$ limit, and $k \times k$ orthogonal matrices $\mathbf{U}$ are randomly sampled under the Haar measure.

V. DROPOUT OF UNDERUTILIZED NEURONS

A decorrelation of principal neuron activities in adaptive PCA and whitening circuits makes an interesting prediction. If the number of principal neurons is greater than the typical
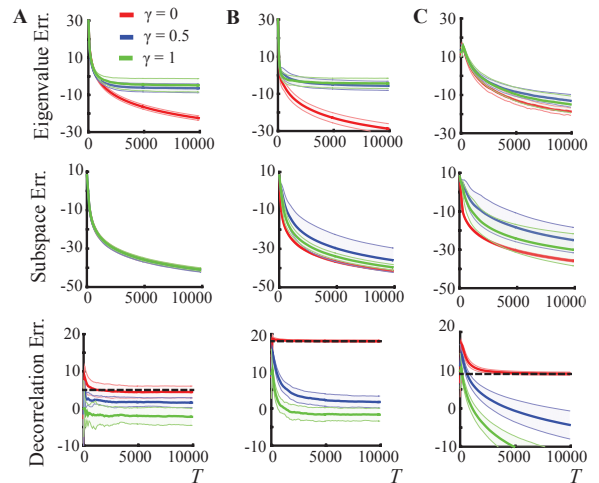

Fig. 2: Performance of the three similarity matching neural networks - PCA (**A**), adaptive PCA (**B**), and whitening (**C**) - as a function of the number of synthetic data sample presentations (see text). Top: eigenvalue error; middle: subspace error; bottom: decorrelation error (for definitions see text). Means (solid lines) and STDs over 20 runs (shades) of the metrics are shown for three different decorrelation parameters: $\gamma = 0$, or no decorrelation (red), $\gamma = 0.5$ (green), $\gamma = 1$ (blue). Horizontal dashed black lines (bottom row) show the correlation error for random covariance matrices (see text).

number of output eigenvalues then the extra neurons are typically silent. Because the weight of synapses onto the extra neurons is proportional to their activities (23), (29) these synapses will be weak or non-existent. This suggests that the extra neurons disconnect or drop out of the circuit and, in a biological system, may be disposed off. An example of this phenomenon for the adaptive PCA network is shown in Figure 3. Note that our use of the term "dropout" is different from random and intermittent silencing of neurons to regularize learning in deep artificial neural networks [24].

A reverse process may also take place. If the number of principal neurons is less than the typical number of eigenvalues exceeding the threshold, in a biological system, extra neurons may be added to the circuit via neurogenesis.

How does the PCA network behave if the input covariance matrix has few non-zero eigenvalues, $m < \min(k, n)$? As above, the $k - m$ principal are silent in the steady state after each data presentation. However, if the weights of synapses onto these principal neurons are initialized randomly, they do not decay to zero according to (18). Therefore, these silent neurons are active during the initial iterations of the dynamics stage (15). Furthermore, the learning rates of the silent neurons stay high and they can become active if the input covariance acquires a new non-zero eigenvalue.

VI. DECORRELATION OF INTERNEURONS

Optimal downstream information transmission by principal neurons in adaptive PCA and whitening circuits does not require decorrelation of interneuron activities. Yet, interneuron decorrelation is easily achieved by adding a decorrelating regularizer $-\rho\left\|\text{off}(\mathbf{Z}\mathbf{Z}^\top)\right\|_F^2$, where $\rho > 0$, to adaptive PCA (5) and whitening (8) objectives. From the modified objectives one can derive corresponding online algorithms
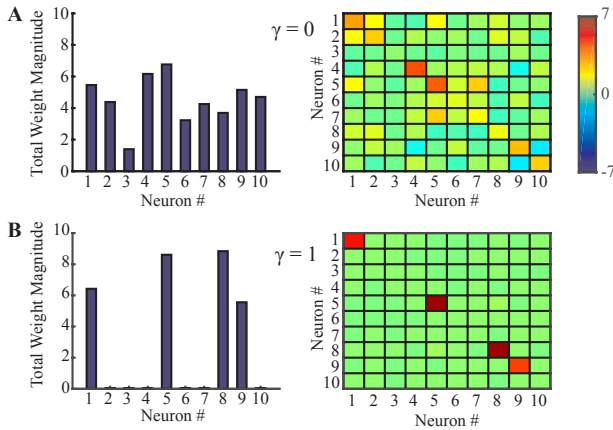
Fig. 3: Dropout of underutilized neurons in the adaptive PCA network. Simulations of the adaptive PCA network without the decorrelation term, $\gamma = 0$, demonstrating that all neurons are active and correlated (**A**) and with the decorrelation term, $\gamma = 1$, demonstrating the silencing of extra neurons whose synaptic weights decay to zero (**B**). Left: Summed squared weights of synapses onto principal neurons at $T = 10000$, defined for the ith neuron as $\sqrt{\sum_{j=1}^{n}(W_{ij}^{YX})^2 + \sum_{j=1}^{l}(W_{ij}^{YZ})^2 + \sum_{j=1,j\neq i}^{k}(W_{ij}^{YY})^2}$. Right: Output covariance matrices for principal neurons, $\frac{1}{T}\mathbf{Y}_T\mathbf{Y}_T^\top$, at $T = 10000$. Input data statistics and parameters same as in Section IV.

following the derivations presented in Section III. As before, the steps of these algorithms can be mapped onto the activity of single-layer two-population neural networks with neurally plausible learning rules. In comparison with the corresponding networks presented in Section III, the modified adaptive PCA network has stronger lateral connections between interneurons and the modified whitening network adds lateral connections between interneurons.

We note that, previously, interneurons have been added to single-layer circuits for dimensionality reduction [25] and sparse dictionary learning [26], [27]. In addition, for sparse dictionary learning, two-layer circuits with local learning rules have been proposed [28], [29], [30]. However, none of these models included interneuron activities as dynamical variables in objective functions as was done here and in [16].

VII. CONCLUSION

We developed an optimization theory for PCA and whitening neural networks by adding a decorrelating term to the existing objective functions for projecting data onto principal subspace [13], [16] and deriving, from such objective functions, online algorithms that map onto neural networks with local learning rules. Our theory predicts the dropout of underutilized neurons, due to the decay of their synaptic weights.

ACKNOWLEDGMENT

REFERENCES

[1] K. Crammer, "Online tracking of linear subspaces," in *Learning Theory*. Springer, 2006, pp. 438–452.

[2] R. Arora, A. Cotter, K. Livescu, and N. Srebro, "Stochastic optimization for pca and pls," in *Allerton Conf. Communication, Control, and Computing*. IEEE, 2012, pp. 861–868.

[3] J. Goes, T. Zhang, R. Arora, and G. Lerman, "Robust stochastic principal component analysis," in *Proc. 17th Int. Conf. Artificial Intelligence and Statistics*, 2014, pp. 266–274.

[4] E. Oja, "Simplified neuron model as a principal component analyzer," *J Math Biol*, vol. 15, no. 3, pp. 267–273, 1982.

[5] T. Hu, Z. Towfic, C. Pehlevan, A. Genkin, and D. B. Chklovskii, "A neuron as a signal processing device," in *Asilomar Conf. Signals, Systems and Computers*. IEEE, 2013, pp. 362–366.

[6] K. Diamantaras and S. Kung, *Principal component neural networks: theory and applications*. John Wiley & Sons, Inc., 1996.

[7] T. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Net*, vol. 2, pp. 459–473, 1989.

[8] P. Földiak, "Adaptive network for optimal linear feature extraction," in *Int. Joint Conf. Neural Networks*. IEEE, 1989, pp. 401–405.

[9] J. Rubner and P. Tavan, "A self-organizing network for principal-component analysis," *EPL*, vol. 10, no. 7, p. 693, 1989.

[10] T. Leen, "Dynamics of learning in recurrent feature-discovery networks," *Advances in Neural Information Processing Systems*, 1990.

[11] S. Kung and K. Diamantaras, "A neural network learning algorithm for adaptive principal component extraction (apex)," in *Int. Conf. Acoustics, Speech, and Signal Processing*. IEEE, 1990, pp. 861–864.

[12] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Net*, vol. 5, no. 6, pp. 927–935, 1992.

[13] C. Pehlevan, T. Hu, and D. B. Chklovskii, "A hebbian/anti-hebbian neural network for linear subspace learning: A derivation from multi-dimensional scaling of streaming data," *Neural Comput*, vol. 27, pp. 1461–1495, 2015.

[14] C. Pehlevan and D. B. Chklovskii, "A hebbian/anti-hebbian network derived from online non-negative matrix factorization can cluster and discover sparse features," in *Asilomar Conf. Signals, Systems and Computers*. IEEE, 2014, pp. 769–775.

[15] T. Hu, C. Pehlevan, and D. B. Chklovskii, "A hebbian/anti-hebbian network for online sparse dictionary learning derived from symmetric matrix factorization," in *Asilomar Conference on Signals, Systems and Computers*. IEEE, 2014, pp. 613–619.

[16] C. Pehlevan and D. B. Chklovskii, "A normative theory of adaptive dimensionality reduction in neural networks," *Advances in Neural Information Processing Systems*, 2015.

[17] K. Mardia, J. Kent, and J. Bibby, *Multivariate analysis*. Academic press, 1980.

[18] R. Linsker, "Self-organization in a perceptual network," *Computer*, vol. 21, no. 3, pp. 105–117, 1988.

[19] M. Plumbley, "A hebbian/anti-hebbian network which optimizes information capacity by orthonormalizing the principal subspace," in *Proc. 3rd Int. Conf. Artificial Neural Networks*, 1993, pp. 86–90.

[20] ——, "Information processing in negative feedback neural networks," *Network: Comp Neural*, vol. 7, no. 2, pp. 301–305, 1996.

[21] J. Atick and A. Redlich, "What does the retina know about natural scenes?" *Neural comput*, vol. 4, no. 2, pp. 196–210, 1992.

[22] A. Hyvärinen, J. Hurri, and P. O. Hoyer, *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*. Springer, 2009, vol. 39.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J Mach Learn Res*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] M. Plumbley, "A subspace network that determines its own output dimension," Technical Report, August 1994.

[26] P. D. King, J. Zylberberg, and M. R. DeWeese, "Inhibitory interneurons decorrelate excitatory cells to drive sparse code formation in a spiking model of v1," *J Neurosci*, vol. 33, pp. 5475–5485, 2013.

[27] M. Zhu and C. Rozell, "Modeling inhibitory interneurons in efficient sensory coding models," *PLoS Comput Biol*, vol. 11, no. 7, p. e1004353, 2015.

[28] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision Res*, vol. 37, no. 23, pp. 3311–3325, 1997.

[29] A. A. Koulakov and D. Rinberg, "Sparse incomplete representations: a potential role of olfactory granule cells," *Neuron*, vol. 72, pp. 124–136, 2011.

[30] S. Druckmann, T. Hu, and D. B. Chklovskii, "A mechanistic model of early sensory processing based on subtracting sparse representations," in *Advances in Neural Information Processing Systems*, 2012, pp. 1979–1987.