Biological Cybernetics (2022) 116:557–568
https://doi.org/10.1007/s00422-022-00943-8

Biological
Cybernetics

ORIGINAL ARTICLE

Biologically plausible single-layer networks for nonnegative independent component analysis

David Lipshutz[1] · Cengiz Pehlevan[2] · Dmitri B. Chklovskii[1,3]

Received: 7 March 2022 / Accepted: 18 August 2022 / Published online: 7 September 2022

Abstract

An important problem in neuroscience is to understand how brains extract relevant signals from mixtures of unknown sources, i.e., perform blind source separation. To model how the brain performs this task, we seek a biologically plausible single-layer neural network implementation of a blind source separation algorithm. For biological plausibility, we require the network to satisfy the following three basic properties of neuronal circuits: (i) the network operates in the online setting; (ii) synaptic learning rules are local; and (iii) neuronal outputs are nonnegative. Closest is the work by Pehlevan et al. (Neural Comput 29:2925–2954, 2017), which considers nonnegative independent component analysis (NICA), a special case of blind source separation that assumes the mixture is a linear combination of uncorrelated, nonnegative sources. They derive an algorithm with a biologically plausible 2-layer network implementation. In this work, we improve upon their result by deriving 2 algorithms for NICA, each with a biologically plausible *single-layer* network implementation. The first algorithm maps onto a network with indirect lateral connections mediated by interneurons. The second algorithm maps onto a network with direct lateral connections and multi-compartmental output neurons.

Keywords Blind source separation · Nonnegative independent component analysis · Neural network · Local learning rules

1 Introduction

Brains effortlessly extract relevant signals from mixtures of unknown sources (Cherry 1953; Desimone and Duncan 1995; Hulse et al. 1997; Wilson and Mainen 2006; Narayan et al. 2007; Bee and Micheyl 2008; Shinn-Cunningham 2008; McDermott 2009; Bronkhorst 2015), an unsupervised signal processing problem known as blind source separation. A classic example in audition is the cocktail party problem, in which a listener tries to follow a single conversation in the presence of multiple background conversations. We seek a model of how brains perform blind source separation.

A special case of blind source separation is nonnegative independent component analysis (NICA), which assumes a generative model in which the mixture of stimuli is a linear combination of uncorrelated, nonnegative sources, i.e., $\mathbf{x} = \mathbf{A}\mathbf{s}$, where $\mathbf{s}$ denotes the nonnegative vector of source intensities, $\mathbf{A}$ is a mixing matrix and $\mathbf{x}$ denotes the vector of mixed stimuli. The goal of NICA is to infer the source vectors $\mathbf{s}$ from the mixture vectors $\mathbf{x}$. Both the linear additivity of stimuli and nonnegativity of the sources are reasonable assumptions in biological applications. For example, in olfaction, concentrations of odorants are both additive and nonnegative.

Plumbley (2002) showed that when the sources are well-grounded (i.e., they have nonzero probability of taking infinitesimally small values), NICA can be solved in 2 steps; see Fig. 1. In the first step, the mixture undergoes noncentered whitening; that is, the mixture is linearly transformed to have identity covariance matrix. The second step rotates the mixture until it lies in the nonnegative orthant. The result

Communicated by Benjamin Lindner.

✉ David Lipshutz
dlipshutz@flatironinstitute.org

[1] Center for Computational Neuroscience, Flatiron Institute, New York, USA

[2] John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, USA

[3] Neuroscience Institute, NYU Medical Center, New York, USA

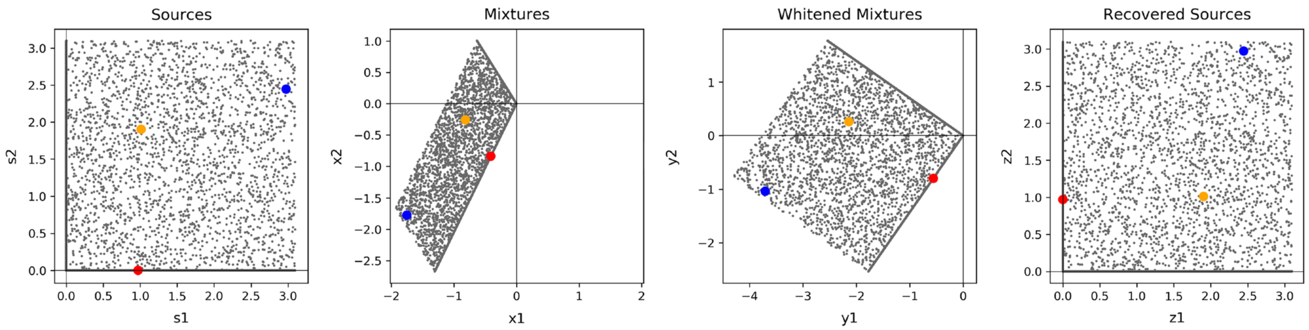

Fig. 1 Illustration of Plumbley's 2-step algorithm for NICA. The red, blue and oranges dots track three source vectors across the mixing, whitening and rotation steps. Our algorithms transform Mixtures into Recovered Sources in a single step implemented by single-layer neural networks

of these 2 steps must be a permutation of the original sources. This important observation led to a number of algorithms for implementing the rotation step (Plumbley 2003; Plumbley and Oja 2004; Oja and Plumbley 2004; Yuan and Oja 2004), many of which have neural network implementations.

Unfortunately, the above-mentioned networks do not offer a viable model of brain function because they do not satisfy one or more of the following three common requirements for biological plausibility (Pehlevan and Chklovskii 2019). First, the network operates in the online or streaming setting where it receives one input at a time and the output is computed before the next input arrives. Second, each synaptic update is local in the sense that it depends only on variables represented in the pre- and postsynaptic neurons. Third, the neuronal outputs are nonnegative.

Building on Plumbley's method, Pehlevan et al. (2017) proposed a 2-layer network for NICA, with each layer derived from a principled objective function. The first layer implements noncentered whitening and the second orthogonally rotates the whitened mixture. While their networks satisfy the requirements for biological plausibility, from a biological perspective, there are advantages to a single-layer network that economizes the number of neurons, which take up valuable resources such as space (Rivera-Alba et al. 2014) and metabolic energy (Laughlin and Sejnowski 2003). (See (Bahroun et al. 2021) for a recent example of a single-layer network with local learning rules for independent component analysis without the nonnegativity constraint).

In this work, we derive 2 NICA algorithms (Algorithms 1 and 2) that can be implemented in biologically plausible single-layer networks, which, respectively, require 2/3 and 1/3 as many neurons as the 2-layer network derived in (Pehlevan et al. 2017). The first algorithm maps onto a network with point neurons and indirect lateral connections mediated by interneurons, and the second algorithm maps onto a network with 2-compartmental neurons and direct lateral connections. To derive our algorithms, we adopt a normative approach which relies on the fact that the original sources can be expressed (up to permutation) as optimal solutions of single objective functions that combine the 2 objectives from (Pehlevan et al. 2017).

Notation. For integers p, q, let $\mathbb{R}^p$ denote p-dimensional Euclidean space, $\mathbb{R}^p_+$ denote the nonnegative orthant in $\mathbb{R}^p$, $\mathbb{R}^{p\times q}$ denote the set of $p \times q$ real-valued matrices and $\mathbb{R}^{p\times q}_+$ denote the subset of matrices with nonnegative entries. Let $\mathcal{S}^p_{++}$ denote the set of $p \times p$ positive definite matrices and let $\mathbf{I}_p$ denote the $p \times p$ identity matrix. We use boldface fonts to denote vectors and matrices and superscripts to denote the indices of a vector or matrix. For example, given a vector $\mathbf{v} \in \mathbb{R}^p$ and matrix $\mathbf{M} \in \mathbb{R}^{p\times q}$, we let v^i and M^{ij}, respectively, denote the i^{th} component of $\mathbf{v}$ and the $(i, j)^{\text{th}}$ element of $\mathbf{M}$, for $1 \le i \le p$ and $1 \le j \le q$.

Given T samples $\mathbf{h}_1, \ldots, \mathbf{h}_T$ of a time series, let

$$\langle \mathbf{h} \rangle := \frac{1}{T}\sum_{t=1}^{T} \mathbf{h}_t, \quad \mathbf{C}_{HH} := \frac{1}{T}\sum_{t=1}^{T}(\mathbf{h}_t - \langle \mathbf{h} \rangle)(\mathbf{h}_t - \langle \mathbf{h} \rangle)^\top,$$

respectively, denote the empirical mean and covariance of the time series. Let $\overline{\mathbf{h}}_t := \frac{1}{t}(\mathbf{h}_1 + \cdots + \mathbf{h}_t)$ denote the running sample mean. Given a data matrix $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_T]$, let $\delta\mathbf{H} := [\mathbf{h}_1 - \langle \mathbf{h} \rangle, \ldots, \mathbf{h}_T - \langle \mathbf{h} \rangle]$ denote the centered data matrix.

2 Review of prior work

In this section, we review Plumbley's analysis (Plumbley 2002) and the objective functions used by Pehlevan et al. (2017) to derive a 2-layer network for NICA. Let $d \ge 2$ and $\mathbf{s}_1, \ldots, \mathbf{s}_T \in \mathbb{R}^d_+$ be T samples of d-dimensional nonnegative source vectors whose components are uncorrelated. Since a constant factor multiplying a source can be absorbed into the associated column of the mixing matrix $\mathbf{A}$, we can assume, without loss of generality, that each component of the source vector has unit sample variance. In particular, $\mathbf{C}_{SS} = \mathbf{I}_d$. Let $k \ge d$, $\mathbf{A}$ be a full rank $k \times d$ mixing matrix and define the k-dimensional mixture vectors by $\mathbf{x}_t := \mathbf{A}\mathbf{s}_t$ for $t = 1, \ldots, T$.

2.1 Plumbley's NICA method

Plumbley (2002) proposed solving NICA in 2 steps: noncentered whitening followed by orthogonal transformation, which are depicted in Fig. 1.

Noncentered whitening is a linear transformation $\mathbf{y} := \mathbf{F}\mathbf{x}$ of the mixture, where $\mathbf{y} \in \mathbb{R}^d$ and $\mathbf{F}$ is a $d \times k$ whitening matrix such that $\mathbf{y}$ has identity covariance matrix, i.e., $\mathbf{C}_{YY} = \mathbf{I}_d$. The combined effect of source mixing and prewhitening steps, which is encoded in the $d \times d$ matrix $\mathbf{FA}$ (since $\mathbf{y} = \mathbf{F}\mathbf{x}$ and $\mathbf{x} = \mathbf{A}\mathbf{s}$), is an orthogonal transformation. To see this, we use the facts that $\mathbf{C}_{SS} = \mathbf{I}_d$, $\mathbf{y} = \mathbf{FA}\mathbf{s}$ and $\mathbf{C}_{YY} = \mathbf{I}_d$ to write

$$\begin{aligned}(\mathbf{FA})(\mathbf{FA})^\top &= (\mathbf{FA})\mathbf{C}_{SS}(\mathbf{FA})^\top \\ &= \frac{1}{T}\sum_{t=1}^{T}\mathbf{FA}(\mathbf{s}_t - \langle\mathbf{s}\rangle)(\mathbf{s}_t - \langle\mathbf{s}\rangle)^\top(\mathbf{FA})^\top \\ &= \frac{1}{T}\sum_{t=1}^{T}(\mathbf{y}_t - \langle\mathbf{y}\rangle)(\mathbf{y}_t - \langle\mathbf{y}\rangle)^\top = \mathbf{C}_{YY} = \mathbf{I}_d.\end{aligned}$$

In the second step, one looks for an orthogonal matrix $\mathbf{R}$ such that the transformation $\mathbf{z} := \mathbf{R}\mathbf{y}$ is nonnegative. For the solution to be unique up to a permutation, each source s^i must be well grounded; that is, $P(s^i < \epsilon) > 0$ for all $\epsilon > 0$. Then by (Plumbley 2002, Theorem 1), the vector $\mathbf{z}$ is equal to a permutation of the sources $\mathbf{s}$.

2.2 Similarity matching objectives for the 2-step algorithm

To obtain a biologically plausible network, Pehlevan et al. (2017) proposed novel mathematical formulations of the noncentered whitening and rotation steps. Here we recall the principled objective functions they use for each layer, which are closely related to the objective functions we use to derive our networks. To this end, define the $k \times T$ concatenated data matrix $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_T]$. In the first step, Pehlevan et al. (2017) optimize, with respect to the $d \times T$ matrix $\mathbf{Y} := [\mathbf{y}_1, \ldots, \mathbf{y}_T]$, the following objective:

$$\begin{aligned}&\underset{\mathbf{Y}\in\mathbb{R}^{d\times T}}{\arg\min}\; -\mathrm{Tr}(\delta\mathbf{Y}^\top\delta\mathbf{Y}\delta\mathbf{X}^\top\delta\mathbf{X}) \quad \text{subject to} \\ &\quad \delta\mathbf{Y}^\top\delta\mathbf{Y} \preceq T\mathbf{I}_T \quad \text{and} \quad \mathbf{Y} = \mathbf{F}\mathbf{X},\end{aligned} \tag{1}$$

for some $d \times k$ matrix $\mathbf{F}$, where we recall that $\delta\mathbf{Y} := [\mathbf{y}_1 - \langle\mathbf{y}\rangle, \ldots, \mathbf{y}_T - \langle\mathbf{y}\rangle]$ is the centered data matrix and the constraint enforces that the difference $T\mathbf{I}_T - \delta\mathbf{Y}^\top\delta\mathbf{Y}$ is positive semidefinite. As shown in (Pehlevan et al. 2017, Theorem 3), objective (1) is optimized when $\mathbf{Y}$ is a noncentered whitened transformation of $\mathbf{X}$. Note that the constraint $\mathbf{Y} = \mathbf{F}\mathbf{X}$ in equation (1) is to ensure that $\mathbf{Y}$ is a linear transformation of $\mathbf{X}$. Rather than optimizing over outputs $\mathbf{Y}$, we could alternatively optimize over whitening matrices $\mathbf{F}$; however, this formulation of the objective would not lead to an algorithm with a biologically plausible network implementation.

For the second step, Pehlevan et al. (2017) introduce the following nonnegative similarity matching (NSM) objective:

$$\underset{\mathbf{Z}\in\mathbb{R}_+^{d\times T}}{\arg\min}\; \|\mathbf{Z}^\top\mathbf{Z} - \mathbf{Y}^\top\mathbf{Y}\|_{\mathrm{Frob}}^2. \tag{2}$$

The objective minimizes the mismatch between similarities of the nonnegative outputs $\mathbf{Z}$ and the noncentered whitened mixtures $\mathbf{Y}$ (as measured by inner products). (Also see the work by Erdogan and Pehlevan (2020), who consider a related objective which allows for bounded, mixed-sign sources.) As shown in (Pehlevan et al. 2017), any orthogonal transformation of $\mathbf{Y}$ to the nonnegative orthant, which corresponds to a permutation of the original sources, is a solution of the NSM objective (2). However, it is challenging to establish uniqueness of solutions (i.e., if every solution of (2) corresponds to a permutation of the original sources). In general, verifying conditions for uniqueness is nontrivial and usually the verification is NP-complete (Donoho and Stodden 2003; Laurberg et al. 2008; Huang et al. 2013).

From objectives (1) and (2), Pehlevan et al. (2017) derive a 2-step algorithm for NICA that can be implemented in a 2-layer neural network that operates in the online setting, uses local learning rules, and whose rotation layer has nonnegative neuronal outputs. The first step of their algorithm requires at least $2d$ neurons and the second step requires d neurons, so their algorithm requires $3d$ neurons in total.

3 Combined objectives for NICA

We now modify objectives (1) and (2) to obtain 2 objectives for NICA, which will be the starting points for the derivations of our 2 online NICA algorithms with single-layer neural network implementations.

3.1 Adding a nonnegativity constraint to the noncentered whitening objective

We first modify the noncentered whitening objective (1). Note that the solution of objective (1) is not unique — left multiplying any solution $\mathbf{Y}$ by an orthogonal matrix $\mathbf{R}$ yields another noncentered whitened transformation of $\mathbf{X}$. In fact, the second step of Plumbley's method (Plumbley 2002) is to identify an orthogonal transformation $\mathbf{R}$ that results in a *nonnegative* whitened transformation $\mathbf{Z} = \mathbf{R}\mathbf{Y}$. Here, we combine the 2 objectives by adding a nonnegativity constraint to the noncentered whitening objective (1). In particular, we optimize $\mathbf{Y}$ over the set of nonnegative matrices, denoted

$\mathbb{R}_+^{d\times T}$, that are linear transformations of $\mathbf{X}$:

$$\begin{aligned}&\arg\min_{\mathbf{Y}\in\mathbb{R}_+^{d\times T}} -\mathrm{Tr}(\delta\mathbf{Y}^\top\delta\mathbf{Y}\delta\mathbf{X}^\top\delta\mathbf{X}) \quad \text{subject to}\\ &\delta\mathbf{Y}^\top\delta\mathbf{Y}\preceq T\mathbf{I}_T \text{and } \mathbf{Y}=\mathbf{F}\mathbf{X},\end{aligned} \tag{3}$$

for some $d\times k$ matrix $\mathbf{F}$, where the constraint $\mathbf{Y}=\mathbf{F}\mathbf{X}$ for some $\mathbf{F}\in\mathbb{R}^{d\times k}$ ensures that $\mathbf{Y}$ is a linear transformation of $\mathbf{X}$.

3.2 Adding a whitening matrix to the NSM objective

Next, we alter the NSM objective (2) by replacing the Gram matrix $\mathbf{Y}^\top\mathbf{Y}$ with terms that depend only on $\mathbf{X}$, which will avoid the need for the noncentered whitening step. Consider the eigendecomposition of the covariance matrix $\mathbf{C}_{XX}=\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$, where $\mathbf{U}$ is a $k\times d$ matrix with orthonormal column vectors and $\boldsymbol{\Lambda}$ is a $d\times d$ diagonal matrix whose diagonal entries are the nonzero eigenvalues of $\mathbf{C}_{XX}$. Then the whitening matrix $\mathbf{F}$ must be of the form $\mathbf{Q}\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{U}^\top$, where $\mathbf{Q}$ can be any $d\times d$ orthogonal matrix. Therefore,

$$\mathbf{Y}^\top\mathbf{Y}=\mathbf{X}^\top\mathbf{F}^\top\mathbf{F}\mathbf{X}=\mathbf{X}^\top\mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^\top\mathbf{X}=\mathbf{X}^\top\mathbf{C}_{XX}^{+}\mathbf{X},$$

where $\mathbf{C}_{XX}^{+}:=\mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^\top$ is the Moore-Penrose inverse of $\mathbf{C}_{XX}$. Substituting in for $\mathbf{Y}^\top\mathbf{Y}$ in the NSM objective (2) results in our second objective:

$$\arg\min_{\mathbf{Z}\in\mathbb{R}_+^{d\times T}}\|\mathbf{Z}^\top\mathbf{Z}-\mathbf{X}^\top\mathbf{C}_{XX}^{+}\mathbf{X}\|_{\mathrm{Frob}}^2. \tag{4}$$

4 Single-layer neural networks for NICA

Starting from objectives (3) and (4), we derive our 2 online NICA algorithms. The first algorithm maps onto a single-layer network with point neurons and *indirect* lateral connections. The second algorithm maps onto a single-layer network with 2-compartmental neurons and *direct* lateral connections.

4.1 Single-layer network with point neurons and indirect lateral connections

The derivation of our online algorithm starting from objective (3) closely follows the derivation of the whitening layer in the network derived in (Pehlevan et al. 2017). The main difference is that the neuronal outputs are constrained to be nonnegative. To begin, we introduce m-dimensional activity vectors $\mathbf{n}_1,\ldots,\mathbf{n}_T$, with $m\geq d$, which we concatenate into the data matrix $\mathbf{N}:=[\mathbf{n}_1,\ldots,\mathbf{n}_T]$ and recall that $\delta\mathbf{N}:=[\mathbf{n}_1-\langle\mathbf{n}\rangle,\ldots,\mathbf{n}_T-\langle\mathbf{n}\rangle]$ denotes the centered data matrix. We use the Gram matrix $\delta\mathbf{N}^\top\delta\mathbf{N}$ as a Lagrange multiplier to enforce the constraint $\delta\mathbf{Y}^\top\delta\mathbf{Y}\preceq T\mathbf{I}_T$ and normalize by T^2:

$$\begin{aligned}&\min_{\mathbf{Y}\in\mathbb{R}_+^{d\times T}}\max_{\mathbf{N}\in\mathbb{R}^{m\times T}}\frac{1}{T^2}\mathrm{Tr}\Big[-\delta\mathbf{Y}^\top\delta\mathbf{Y}\delta\mathbf{X}^\top\delta\mathbf{X}\\ &+\delta\mathbf{N}^\top\delta\mathbf{N}(\delta\mathbf{Y}^\top\delta\mathbf{Y}-T\mathbf{I}_T)\Big]\quad\text{subject to}\quad\mathbf{Y}=\mathbf{F}\mathbf{X}.\end{aligned}$$

In the above objective, the terms $\frac{1}{T}\delta\mathbf{Y}\delta\mathbf{X}^\top$ and $\frac{1}{T}\delta\mathbf{N}\delta\mathbf{Y}^\top$, respectively, encode sample covariances between the activity vectors $\mathbf{y}_t$ and $\mathbf{x}_t$, and between the activity vectors $\mathbf{n}_t$ and $\mathbf{y}_t$. These covariance matrices are a function of the entire dataset, so they cannot be computed in the online setting. Therefore, to derive an online algorithm, we encode the sample covariances in synaptic weight matrices $\mathbf{W}_{XY}$ and $\mathbf{W}_{YN}$ by substituting in with the Legendre transforms

$$\begin{aligned}&-\frac{1}{T^2}\mathrm{Tr}(\delta\mathbf{Y}^\top\delta\mathbf{Y}\delta\mathbf{X}^\top\delta\mathbf{X})\\ &=\min_{\mathbf{W}_{XY}\in\mathbb{R}^{d\times k}}\left\{-\frac{2}{T}\mathrm{Tr}(\delta\mathbf{Y}^\top\mathbf{W}_{XY}\delta\mathbf{X})+\mathrm{Tr}(\mathbf{W}_{XY}\mathbf{W}_{XY}^\top)\right\}\\ &\frac{1}{T^2}\mathrm{Tr}(\delta\mathbf{N}^\top\delta\mathbf{N}\delta\mathbf{Y}^\top\delta\mathbf{Y})\\ &=\max_{\mathbf{W}_{YN}\in\mathbb{R}^{d\times k}}\left\{\frac{2}{T}\mathrm{Tr}(\delta\mathbf{N}^\top\mathbf{W}_{YN}\delta\mathbf{Y})-\mathrm{Tr}(\mathbf{W}_{YN}\mathbf{W}_{YN}^\top)\right\}.\end{aligned}$$

The above equivalences can be readily justified by differentiating right-hand sides of the above equations with respect to $\mathbf{W}_{XY}$ and $\mathbf{W}_{YN}$ and noting the optima are achieved when $\mathbf{W}_{XY}=\frac{1}{T}\delta\mathbf{Y}\delta\mathbf{X}^\top$ and $\mathbf{W}_{YN}=\frac{1}{T}\delta\mathbf{N}\delta\mathbf{Y}^\top$. Substituting in with the Legendre transforms results in the following objective

$$\min_{\mathbf{Y}\in\mathbb{R}_+^{d\times T}}\max_{\mathbf{N}\in\mathbb{R}^{d\times T}}\min_{\mathbf{W}_{XY}\in\mathbb{R}^{d\times k}}\max_{\mathbf{W}_{YN}\in\mathbb{R}^{m\times d}}L_1(\mathbf{Y},\mathbf{N},\mathbf{W}_{XY},\mathbf{W}_{YN})$$

subject to $\mathbf{Y}=\mathbf{F}\mathbf{X}$,

where

$$\begin{aligned}L_1(\mathbf{Y},\mathbf{N},\mathbf{W}_{XY},\mathbf{W}_{YN}):=&\frac{1}{T}\mathrm{Tr}\Big(2\delta\mathbf{N}^\top\mathbf{W}_{YN}\delta\mathbf{Y}\\ &-2\delta\mathbf{Y}^\top\mathbf{W}_{XY}\delta\mathbf{X}-\delta\mathbf{N}^\top\delta\mathbf{N}\Big)\\ &-\mathrm{Tr}\Big(\mathbf{W}_{YN}\mathbf{W}_{YN}^\top)+\mathrm{Tr}(\mathbf{W}_{XY}\mathbf{W}_{XY}^\top\Big).\end{aligned}$$

Since L_1 is convex in $\mathbf{W}_{XY}$ (resp. $\mathbf{Y}$) and strongly concave in $\mathbf{N}$ (resp. $\mathbf{W}_{YN}$), L_1 satisfies the saddle point property with respect to $\mathbf{W}_{XY}$ and $\mathbf{N}$ (resp. $\mathbf{Y}$ and $\mathbf{W}_{YN}$), see appendix A for a definition of the of saddle point property, so we can interchange the order of optimization, as follows:

$$\min_{\mathbf{W}_{XY}\in\mathbb{R}^{d\times k}}\max_{\mathbf{W}_{YN}\in\mathbb{R}^{m\times d}}\min_{\mathbf{Y}\in\mathbb{R}_+^{d\times T}}\max_{\mathbf{N}\in\mathbb{R}^{d\times T}}$$

$$L_1(\mathbf{Y}, \mathbf{N}, \mathbf{W}_{XY}, \mathbf{W}_{YN}) \quad \text{subject to} \quad \mathbf{Y} = \mathbf{F}\mathbf{X}. \tag{5}$$

We first solve objective (5) in the offline setting. In general, optimizing over $(\mathbf{Y}, \mathbf{N})$ is challenging due to the constraint that $\mathbf{Y}$ be a nonnegative linear transformation of $\mathbf{X}$. In appendix B, we show that when the synaptic weights $\mathbf{W}_{XY}$ and $\mathbf{W}_{YN}$ are at their optimal values, we can optimize over $(\mathbf{Y}, \mathbf{N})$ by repeating the following projected gradient descent steps until convergence:

$$\begin{aligned} \mathbf{Y} &\leftarrow \left[\mathbf{Y} + \gamma\left(\mathbf{W}_{XY}\mathbf{X} - \mathbf{W}_{YN}^\top \mathbf{N}\right)\right]_+, \\ \mathbf{N} &\leftarrow \mathbf{N} + \gamma\left(\mathbf{W}_{YN}\mathbf{Y} - \mathbf{N}\right), \end{aligned} \tag{6}$$

where $\gamma > 0$ is a small step size and $[\cdot]_+$ denotes taking the positive part elementwise, which ensures the nonnegativity of $\mathbf{Y}$. In the case the synaptic weights $\mathbf{W}_{XY}$ and $\mathbf{W}_{YN}$ are not at their optimal values, we repeat the above projected gradient descent steps until convergence to obtain an approximation of the optimal $(\mathbf{Y}, \mathbf{N})$. We then perform a gradient descent-ascent step of the objective L_1 with respect to $\mathbf{W}_{XY}$ and $\mathbf{W}_{YN}$:

$$\mathbf{W}_{XY} \leftarrow \mathbf{W}_{XY} + \eta\left(\frac{1}{T}\delta\mathbf{Y}\delta\mathbf{X}^\top - \mathbf{W}_{XY}\right) \tag{7}$$

$$\mathbf{W}_{YN} \leftarrow \mathbf{W}_{YN} + \eta\left(\frac{1}{T}\delta\mathbf{N}\delta\mathbf{Y}^\top - \mathbf{W}_{YN}\right). \tag{8}$$

Here $\eta > 0$ is the learning rate for $\mathbf{W}_{XY}$ and $\mathbf{W}_{YN}$.

Next, we solve the objective (5) in the online setting. At each time step t, we approximate the optimization over $(\mathbf{y}_t, \mathbf{n}_t)$ by taking the following projected gradient descent steps until convergence:

$$\begin{aligned} \mathbf{y}_t &\leftarrow [\mathbf{y}_t + \gamma(\mathbf{W}_{XY}\mathbf{x}_t - \mathbf{W}_{NY}\mathbf{n}_t)]_+, \\ \mathbf{n}_t &\leftarrow \mathbf{n}_t + \gamma(\mathbf{W}_{YN}\mathbf{y}_t - \mathbf{n}_t), \end{aligned} \tag{9}$$

where we have defined $\mathbf{W}_{NY} := \mathbf{W}_{YN}^\top$. We then take stochastic gradient descent-ascent steps in $\mathbf{W}_{XY}$ and $\mathbf{W}_{YN}$ by replacing the averages in equations (7) and (8) with their online approximations:

$$\begin{aligned} \mathbf{W}_{XY} &\leftarrow \mathbf{W}_{XY} + \eta\left((\mathbf{y}_t - \overline{\mathbf{y}}_t)(\mathbf{x}_t - \overline{\mathbf{x}}_t)^\top - \mathbf{W}_{XY}\right) \\ \mathbf{W}_{YN} &\leftarrow \mathbf{W}_{YN} + \eta\left((\mathbf{n}_t - \overline{\mathbf{n}}_t)(\mathbf{y}_t - \overline{\mathbf{y}}_t)^\top - \mathbf{W}_{YN}\right) \\ \mathbf{W}_{NY} &\leftarrow \mathbf{W}_{NY} + \eta\left((\mathbf{y}_t - \overline{\mathbf{y}}_t)(\mathbf{n}_t - \overline{\mathbf{n}}_t)^\top - \mathbf{W}_{NY}\right). \end{aligned}$$

The symmetry of the updates for $\mathbf{W}_{NY}$ and $\mathbf{W}_{YN}$ ensures that $\mathbf{W}_{NY} = \mathbf{W}_{YN}^\top$ after each iteration provided the constraint holds at initialization; however, enforcing such a symmetric initialization may not be biologically plausible. In Appendix C, we show that we can relax this initialization constraint, which yields our first online NICA algorithm, Algorithm 1.

Algorithm 1 Bio-NICA with interneurons

```
input mixtures $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$; parameters $\gamma$, $\eta$
initialize $\mathbf{W}_{XY}, \mathbf{W}_{YN}, \mathbf{W}_{NY}, \overline{\mathbf{x}}_0 = \mathbf{0}, \overline{\mathbf{y}}_0 = \mathbf{0}, \overline{\mathbf{n}}_0 = \mathbf{0}$
for $t = 1, 2, \ldots, T$ do
    $\mathbf{y}_t \leftarrow \mathbf{0}$
    $\mathbf{n}_t \leftarrow \mathbf{0}$
    repeat
        $\mathbf{y}_t \leftarrow [\mathbf{y}_t + \gamma(\mathbf{W}_{XY}\mathbf{x}_t - \mathbf{W}_{NY}\mathbf{n}_t)]_+$
        $\mathbf{n}_t \leftarrow \mathbf{n}_t + \gamma(\mathbf{W}_{YN}\mathbf{y}_t - \mathbf{n}_t)$
    until convergence
    $\overline{\mathbf{x}}_t \leftarrow \overline{\mathbf{x}}_{t-1} + \frac{1}{t}(\mathbf{x}_t - \overline{\mathbf{x}}_{t-1})$
    $\overline{\mathbf{y}}_t \leftarrow \overline{\mathbf{y}}_{t-1} + \frac{1}{t}(\mathbf{y}_t - \overline{\mathbf{y}}_{t-1})$
    $\overline{\mathbf{n}}_t \leftarrow \overline{\mathbf{n}}_{t-1} + \frac{1}{t}(\mathbf{n}_t - \overline{\mathbf{n}}_{t-1})$
    $\mathbf{W}_{XY} \leftarrow \mathbf{W}_{XY} + \eta((\mathbf{y}_t - \overline{\mathbf{y}}_t)(\mathbf{x}_t - \overline{\mathbf{x}}_t)^\top - \mathbf{W}_{XY})$
    $\mathbf{W}_{NY} \leftarrow \mathbf{W}_{NY} + \eta((\mathbf{y}_t - \overline{\mathbf{y}}_t)(\mathbf{n}_t - \overline{\mathbf{n}}_t)^\top - \mathbf{W}_{NY})$
    $\mathbf{W}_{YN} \leftarrow \mathbf{W}_{YN} + \eta((\mathbf{n}_t - \overline{\mathbf{n}}_t)(\mathbf{y}_t - \overline{\mathbf{y}}_t)^\top - \mathbf{W}_{YN})$
end for
```

Algorithm 1 can be implemented in a single-layer network with point neurons and indirect lateral connections mediated by interneurons, Fig. 2, so we refer to the algorithm as 'Bio-NICA with interneurons.' The network consists of k input neurons, d principal (output) neurons and m interneurons. Since $m \geq d$, Algorithm 1 requires a minimum of $2d$ neurons, which is 2/3 as many neurons as required by the 2-layer network in (Pehlevan et al. 2017). Feedforward synapses between the input and principal neurons encode the weight matrix $\mathbf{W}_{XY}$ and lateral synapses between the principal neurons (resp. interneurons) and the interneurons (resp. principal neurons) encode the weight matrix $\mathbf{W}_{YN}$ (resp. $\mathbf{W}_{NY}$). At each time step t, the k-dimensional mixture $\mathbf{x}_t$, which is represented by the k input neurons, is multiplied by the weight matrix $\mathbf{W}_{XY}$, which yields the d-dimensional projection $\mathbf{W}_{XY}\mathbf{x}_t$. This is followed by the fast recurrent dynamics in equation (9). The equilibrium values of $\mathbf{y}_t$ and $\mathbf{n}_t$, respectively, correspond to the nonnegative output of the principal neurons and the output of the interneurons.

We can write the elementwise synaptic updates as follows,

$$\begin{aligned} W_{XY}^{ij} &\leftarrow W_{XY}^{ij} + \eta\left((y_t^i - \overline{y}_t^i)(x_t^j - \overline{x}_t^j) - W_{XY}^{ij}\right), \\ &\quad 1 \leq i \leq d,\ 1 \leq j \leq k, \\ W_{NY}^{ij} &\leftarrow W_{NY}^{ij} + \eta\left((y_t^i - \overline{y}_t^i)(n_t^j - \overline{n}_t^j) - W_{NY}^{ij}\right), \\ &\quad 1 \leq i \leq d,\ 1 \leq j \leq m, \\ W_{YN}^{ij} &\leftarrow W_{YN}^{ij} + \eta\left((n_t^i - \overline{n}_t^i)(y_t^j - \overline{y}_t^j) - W_{YN}^{ij}\right), \\ &\quad 1 \leq i \leq m,\ 1 \leq j \leq d, \end{aligned}$$

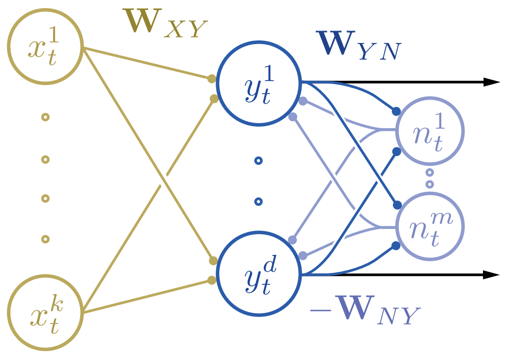

Fig. 2 Single-layer network with interneurons for implementing Algorithm 1

where we recall that $\overline{\mathbf{x}}_t$, $\overline{\mathbf{y}}_t$ and $\overline{\mathbf{n}}_t$ are the running means of $\mathbf{x}_t$, $\mathbf{y}_t$ and $\mathbf{n}_t$, respectively. We assume that each neuron stores the running mean of its activity. Biologically, these means could be represented at the pre- and postsynaptic terminals by slowly changing calcium concentrations. From the elementwise updates, we see that the update for each synapse is local in the sense that it only depends on variables that are represented in the pre- and postsynaptic neurons.

4.2 Single-layer network with 2-compartmental neurons and direct lateral connections

The derivation of the our online algorithm starting from objective (4) is closely related to the derivation of the single-layer networks with multi-compartmental neurons for solving generalized eigenvalue problems (Lipshutz et al. 2020, 2021). To begin, we expand the square, drop terms that do not depend on $\mathbf{Z}$, and normalize by T^2:

$$\min_{\mathbf{Z}\in\mathbb{R}_+^{d\times T}} \frac{1}{T^2}\operatorname{Tr}\left(-2\mathbf{Z}^\top\mathbf{Z}\mathbf{X}^\top\mathbf{C}_{XX}^+\mathbf{X}+\mathbf{Z}^\top\mathbf{Z}\mathbf{Z}^\top\mathbf{Z}\right). \tag{10}$$

Next, we introduce synaptic weight matrices $\mathbf{W}_{XZ}$ and $\mathbf{W}_{ZZ}$ in place of $\frac{1}{T}\mathbf{Z}\mathbf{X}^\top\mathbf{C}_{XX}^+$ and $\frac{1}{T}\mathbf{Z}\mathbf{Z}^\top$, respectively, by substituting in with the following Legendre transforms:

$$-\frac{1}{T^2}\operatorname{Tr}(\mathbf{Z}^\top\mathbf{Z}\mathbf{X}^\top\mathbf{C}_{XX}^+\mathbf{X})$$
$$=\min_{\mathbf{W}_{XZ}\in\mathbb{R}^{d\times k}}\left\{-\frac{2}{T}\operatorname{Tr}(\mathbf{Z}^\top\mathbf{W}_{XZ}\mathbf{X})+\operatorname{Tr}(\mathbf{W}_{XZ}\mathbf{C}_{XX}\mathbf{W}_{XZ}^\top)\right\}$$
$$\frac{1}{T^2}\operatorname{Tr}(\mathbf{Z}^\top\mathbf{Z}\mathbf{Z}^\top\mathbf{Z})$$
$$=\max_{\mathbf{W}_{ZZ}\in\mathcal{S}_{++}^d}\left\{\frac{2}{T}\operatorname{Tr}(\mathbf{Z}^\top\mathbf{W}_{ZZ}\mathbf{Z})-\operatorname{Tr}(\mathbf{W}_{ZZ}^2)\right\}.$$

The above equivalences can be seen by taking partial derivatives with respect to $\mathbf{W}_{XZ}$ (resp. $\mathbf{W}_{ZZ}$) and noting the minimum (resp. maximum) is achieved when $\mathbf{W}_{XZ}=\frac{1}{T}\mathbf{Z}\mathbf{X}^\top\mathbf{C}_{XX}^+$ (resp. $\mathbf{W}_{ZZ}=\frac{1}{T}\mathbf{Z}\mathbf{Z}^\top$). Substituting in with the Legendre transforms results in the minimax objective:

$$\min_{\mathbf{Z}\in\mathbb{R}_+^{d\times T}}\ \min_{\mathbf{W}_{XZ}\in\mathbb{R}^{d\times k}}\ \max_{\mathbf{W}_{ZZ}\in\mathcal{S}_{++}^d} L_2(\mathbf{Z},\mathbf{W}_{XZ},\mathbf{W}_{ZZ}), \tag{11}$$

where

$$L_2(\mathbf{Z},\mathbf{W}_{XZ},\mathbf{W}_{ZZ}) := \frac{2}{T}\operatorname{Tr}\left(\mathbf{Z}^\top\mathbf{W}_{ZZ}\mathbf{Z}-2\mathbf{Z}^\top\mathbf{W}_{XZ}\mathbf{X}\right)-$$
$$\operatorname{Tr}\left(\mathbf{W}_{ZZ}^2-2\mathbf{W}_{XZ}\mathbf{C}_{XX}\mathbf{W}_{XZ}^\top\right).$$

Since the objective L_2 is strongly convex in $\mathbf{W}_{XZ}$ and strongly concave in $\mathbf{W}_{ZZ}$, L_2 satisfies the saddle point property with respect to $\mathbf{Z}$ and $\mathbf{W}_{ZZ}$ (see appendix A), so we can interchange the order of optimization, as follows:

$$\min_{\mathbf{W}_{XZ}\in\mathbb{R}^{d\times k}}\ \max_{\mathbf{W}_{ZZ}\in\mathcal{S}_{++}^d}\ \min_{\mathbf{Z}\in\mathbb{R}_+^{d\times T}} L_2(\mathbf{Z},\mathbf{W}_{XZ},\mathbf{W}_{ZZ}). \tag{12}$$

We first solve the minimax objective (12) in the offline setting by minimizing L_2 over $\mathbf{Z}$ and then taking gradient descent-ascent steps in $\mathbf{W}_{XZ}$ and $\mathbf{W}_{ZZ}$. The minimization over $\mathbf{Z}$ can be approximated by repeating the following projected gradient descent steps until convergence:

$$\mathbf{Z}\leftarrow[\mathbf{Z}+\gamma(\mathbf{W}_{XZ}\mathbf{X}-\mathbf{W}_{ZZ}\mathbf{Z})]_+,$$

where $\gamma>0$ is a small step size. Next, having minimized over $\mathbf{Z}$, we perform a gradient descent-ascent step of the objective function L_2 with respect to $\mathbf{W}_{XZ}$ and $\mathbf{W}_{ZZ}$:

$$\mathbf{W}_{XZ}\leftarrow\mathbf{W}_{XZ}+2\eta\left(\frac{1}{T}\mathbf{Z}\mathbf{X}^\top-\mathbf{W}_{XZ}\mathbf{C}_{XX}\right), \tag{13}$$

$$\mathbf{W}_{ZZ}\leftarrow\mathbf{W}_{ZZ}+\frac{\eta}{\tau}\left(\frac{1}{T}\mathbf{Z}\mathbf{Z}^\top-\mathbf{W}_{ZZ}\right). \tag{14}$$

Here $\tau>0$ is the ratio between the learning rates for $\mathbf{W}_{XZ}$ and $\mathbf{W}_{ZZ}$, and $\eta\in(0,\tau)$ is the learning rate for $\mathbf{W}_{XZ}$. The upper bound $\eta<\tau$ ensures that $\mathbf{W}_{ZZ}$ remains positive definite given a positive definite initialization. To see this, note that if $\mathbf{W}_{ZZ}$ is positive definite and $0<\eta<\tau$, then the right-hand side of (14) is a strict convex combination of a positive definite matrix and a positive semidefinite matrix, so the right-hand side of (14) is positive definite. Therefore, given a positive definite initialization, $\mathbf{W}_{ZZ}$ remains positive definite.

To solve the minimax objective (12) in the online setting, we take stochastic gradient ascent-descent steps. At each time step t, analogous to the offline setting, we first minimize over the output $\mathbf{z}_t$ by repeating the following projected gradient descent steps until convergence:

$$\mathbf{z}_t\leftarrow[\mathbf{z}_t+\gamma(\mathbf{c}_t-\mathbf{W}_{ZZ}\mathbf{z}_t)]_+, \tag{15}$$

where we have defined the projection $\mathbf{c}_t := \mathbf{W}_{XZ}\mathbf{x}_t$. We then take stochastic gradient descent-ascent steps in $\mathbf{W}_{XZ}$ and $\mathbf{W}_{ZZ}$. To this end, we replace the averages $\frac{1}{T}\mathbf{Z}\mathbf{X}^\top$ and $\frac{1}{T}\mathbf{Z}\mathbf{Z}^\top$ in equations (13) and (14) with their respective online approximations $(\mathbf{z}_t - \bar{\mathbf{z}}_t)(\mathbf{x}_t - \bar{\mathbf{x}}_t)^\top$ and $(\mathbf{z}_t - \bar{\mathbf{z}}_t)(\mathbf{z}_t - \bar{\mathbf{z}}_t)^\top$. While we could approximate the matrix $\mathbf{W}_{XZ}\mathbf{C}_{XX}$ in the online setting with $\mathbf{W}_{XZ}(\mathbf{x}_t - \bar{\mathbf{x}}_t)(\mathbf{x}_t - \bar{\mathbf{x}}_t)^\top$, this does not lead to local learning rules. Instead, we observe that

$$\mathbf{W}_{XZ}\mathbf{C}_{XX} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{W}_{XZ}(\mathbf{x}_t - \langle\mathbf{x}\rangle)(\mathbf{x}_t - \langle\mathbf{x}\rangle)^\top$$

$$= \frac{1}{T}\sum_{t=1}^{T}(\mathbf{c}_t - \langle\mathbf{c}\rangle)(\mathbf{x}_t - \langle\mathbf{x}\rangle)^\top,$$

and replace $\mathbf{W}_{XZ}\mathbf{C}_{XX}$ with the online approximation $(\mathbf{c}_t - \bar{\mathbf{c}}_t)(\mathbf{x}_t - \bar{\mathbf{x}}_t)^\top$. This yields our second online algorithm for NICA, Algorithm 2.

Algorithm 2 Bio-NICA with 2-compartmental neurons

```
input mixtures $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$; parameters $\gamma, \eta, \tau$
initialize $\mathbf{W}_{XZ}, \mathbf{W}_{ZZ}, \bar{\mathbf{x}}_0 = \mathbf{0}, \bar{\mathbf{c}}_0 = \mathbf{0}$
for $t = 1, 2, \ldots, T$ do
    $\mathbf{c}_t \leftarrow \mathbf{W}_{XZ}\mathbf{x}_t$
    $\mathbf{z}_t \leftarrow \mathbf{0}$
    repeat
        $\mathbf{z}_t \leftarrow [\mathbf{z}_t + \gamma(\mathbf{c}_t - \mathbf{W}_{ZZ}\mathbf{z}_t)]_+$
    until convergence
    $\bar{\mathbf{x}}_t \leftarrow \bar{\mathbf{x}}_{t-1} + \frac{1}{t}(\mathbf{x}_t - \bar{\mathbf{x}}_{t-1})$
    $\bar{\mathbf{c}}_t \leftarrow \bar{\mathbf{c}}_{t-1} + \frac{1}{t}(\mathbf{c}_t - \bar{\mathbf{c}}_{t-1})$
    $\mathbf{W}_{XZ} \leftarrow \mathbf{W}_{XZ} + 2\eta(\mathbf{z}_t\mathbf{x}_t^\top - (\mathbf{c}_t - \bar{\mathbf{c}}_t)(\mathbf{x}_t - \bar{\mathbf{x}}_t)^\top)$
    $\mathbf{W}_{ZZ} \leftarrow \mathbf{W}_{ZZ} + \frac{\eta}{\tau}(\mathbf{z}_t\mathbf{z}_t^\top - \mathbf{W}_{ZZ})$
end for
```

Algorithm 2 can be implemented in a single-layer network with 2-compartmental neurons and direct lateral connections, Fig. 3, so we refer to the algorithm as 'Bio-NICA with 2-compartmental neurons.' The network consists of k input neurons and d output neurons, which is 1/3 as many neurons as required by the 2-layer network in (Pehlevan et al. 2017). Each output neuron has a dendritic compartment and a somatic compartment. Feedforward synapses between the input and output neurons encode the weight matrix $\mathbf{W}_{XZ}$ and recursive lateral synapses between the output neurons encode the weight matrix $-\mathbf{W}_{ZZ}$. At each time step t, the k-dimensional mixture $\mathbf{x}_t$, which is represented by the input neurons, is multiplied by the weight matrix $\mathbf{W}_{XZ}$, which is encoded by the feedforward synapses connecting the input neurons to the output neurons. This yields the d-dimensional projection $\mathbf{c}_t = \mathbf{W}_{XZ}\mathbf{x}_t$, which is computed in the dendritic compartments of the output neurons and then propagated to their somatic compartments. This is followed by the fast recurrent neural dynamics in equation (15). The equilibrium value of $\mathbf{z}_t$ corresponds to the nonnegative somatic activity of the output neurons.

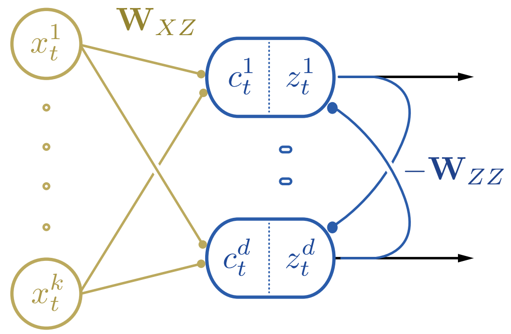

Fig. 3 Single-layer network with 2-compartmental neurons for implementing Algorithm 2

The elementwise synaptic updates are as follows,

$$W_{XZ}^{ij} \leftarrow W_{XZ}^{ij} + 2\eta\left(z_t^i x_t^j - (c_t^i - \bar{c}_t^i)(x_t^j - \bar{x}_t^j)\right),$$
$$1 \le i \le d,\ 1 \le j \le k,$$
$$W_{ZZ}^{ij} \leftarrow W_{ZZ}^{ij} + \frac{\eta}{\tau}\left(z_t^i z_t^j - W_{ZZ}^{ij}\right),$$
$$1 \le i, j \le d,$$

where we recall that $\bar{\mathbf{x}}_t$ and $\bar{\mathbf{c}}_t$ are the running means of $\mathbf{x}_t$ and $\mathbf{c}_t$, respectively. We assume that the input neurons and output neurons, respectively, store the running means $\bar{\mathbf{x}}_t$ and $\bar{\mathbf{c}}_t$. Thus, we see that the update for each synapse is local; that is, the update depends only on variables that are represented in the pre- and postsynaptic neurons.

5 Numerical experiments

We evaluated Algorithms 1 and 2 on synthetic and real datasets and compare their performance to 2 state-of-the-art online NICA algorithms: Nonnegative PCA (Plumbley and Oja 2004) and 2-layer NSM (Pehlevan et al. 2017). Nonnegative PCA requires (noncentered) pre-whitened inputs, which we implemented offline. To quantify the performance of the algorithms, we use the mean-squared error,

$$\text{error}(t) = \frac{1}{td}\sum_{t'=1}^{t}|\mathbf{s}_{t'} - \mathbf{P}\mathbf{y}_{t'}|^2,$$

where $\mathbf{P}$ is the permutation matrix that minimizes the error at the final time point. For detailed descriptions of our implementations, see Appendix D. The evaluation code is available at https://github.com/flatironinstitute/bio-nica.

5.1 Mixture of sparse random uniform sources

We first compare the algorithms on a synthetic dataset generated by independent and identically distributed samples. Following Pehlevan et al. (2017), each source sample was set to zero with probability $1/2$ or sampled uniformly from the interval $(0, \sqrt{48/5})$ with probability $1/2$. We used random square mixing matrices whose elements were independent standard normal random variables. In Fig. 4, we plot the performance of each algorithm on mixtures of d-dimensional sources, for $d = 3, 5, 7, 10$.

5.2 Mixture of natural images

We apply the NICA algorithms to the problem of recovering images from their mixtures, see Fig. 5 (left). Three image patches of size 252 × 252 pixels were chosen from a set of images of natural scenes (Hyvärinen and Hoyer 2000) previously used in (Hyvärinen and Oja 2000; Plumbley and Oja 2004; Pehlevan et al. 2017). Each image is treated as one source, with the pixel intensities (shifted and scaled to be well-grounded and have unit variance) representing the $252^2 = 63504$ samples. The source vectors were multiplied by a random 3×3 mixing matrix to generate 3-dimensional mixtures, which were presented to the algorithms 5 times with a randomly permuted order in each presentation. In Fig. 5 (right), we show the performance of each algorithm. To generated the recovered images in Fig. 5 (left), we take the outputs of Algorithms 1 and 2 during the final presentation of the mixtures and invert the permutation that was applied to the mixtures in the final presentation.

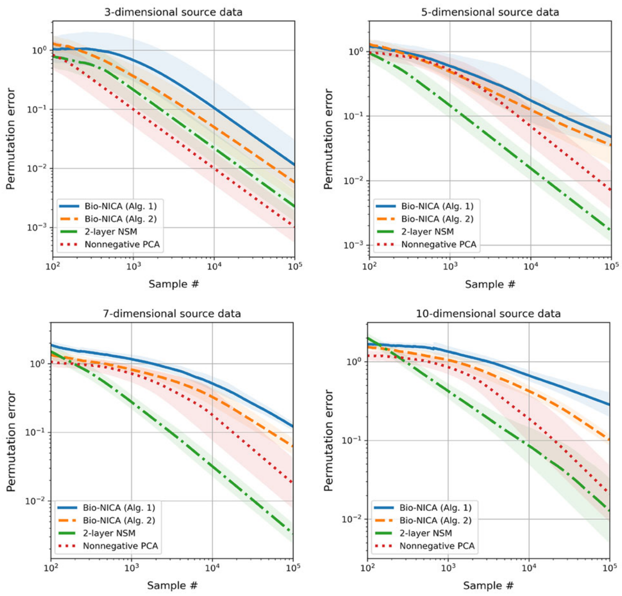

Fig. 4 Performance of algorithms when presented with mixtures of sparse random uniform sources, in terms of permutation error. The lines and shaded regions denote the means and 90% confidence intervals over 10 runs

6 Discussion

In this work, we derived 2 algorithms for NICA, each of which can be implemented by biologically plausible single-layer networks. Our networks, respectively, use 2/3 and 1/3 as many neurons as the 2-layer biologically plausible network derived in (Pehlevan et al. 2017). The wiring diagrams of both networks—feedforward connections with lateral recurrent connections or feedforward connections with indirect lateral recurrent connections mediated by interneurons—are common motifs in neural systems.

We speculate that such circuits are useful for understanding early odor adaptation in olfactory systems. As mentioned earlier, NICA is particularly relevant in olfaction because concentrations of odorants are both additive and nonnegative. Moreover, the vertebrate olfactory bulb has been shown to perform *pattern separation* (Friedrich and Laurent 2001; Gschwend et al. 2015); that is, distinct odors (i.e., sources) that activate overlapping sets of olfactory receptor neurons will activate non-overlapping sets of neurons in the olfactory bulb. Pattern separation is closely related to blind source separation, so the algorithms developed here may be useful for understanding pattern separation in the olfactory bulb.

Our numerical experiments suggest that Algorithm 1 is outperformed (in terms of convergence speed) by Algorithm 2. While Algorithm 2 converges faster than Algorithm 1, it requires tuning an extra hyperparameter τ. In addition, both our algorithms are outperformed by Nonnegative PCA and the 2-layer NSM network. However, direct comparison between our algorithms and the competing algorithms is not entirely fair because Nonnegative PCA requires prewhitened inputs and its neural network implementation does not use local learning rules, and the 2-layer NSM network requires more neurons. Our algorithms perform both the whitening and the rotation steps in a single layer, which leads to a trade-off in performance. Therefore, the ‘best’ algorithm for the application of interest will depend on the relative importance of convergence speed versus minimizing the total number of neurons.

Finally, we do not prove convergence guarantees for Algorithms 1 and 2. In general, establishing theoretical guarantees for gradient descent-ascent problems is challenging and is further complicated by the non-smoothness of the projected gradient descent steps in Algorithms 1 and 2.

Acknowledgements We are grateful to Lucy Reading-Ikkanda for creating Figs. 2 and 3. We thank Siavash Golkar, Johannes Friedrich, Tiberiu Tesileanu, Alex Genkin, Jason Moore and Yanis Bahroun for helpful comments and feedback on an earlier draft of this work. We especially thank Siavash Golkar for pointing out that, in Sect. 4.2,

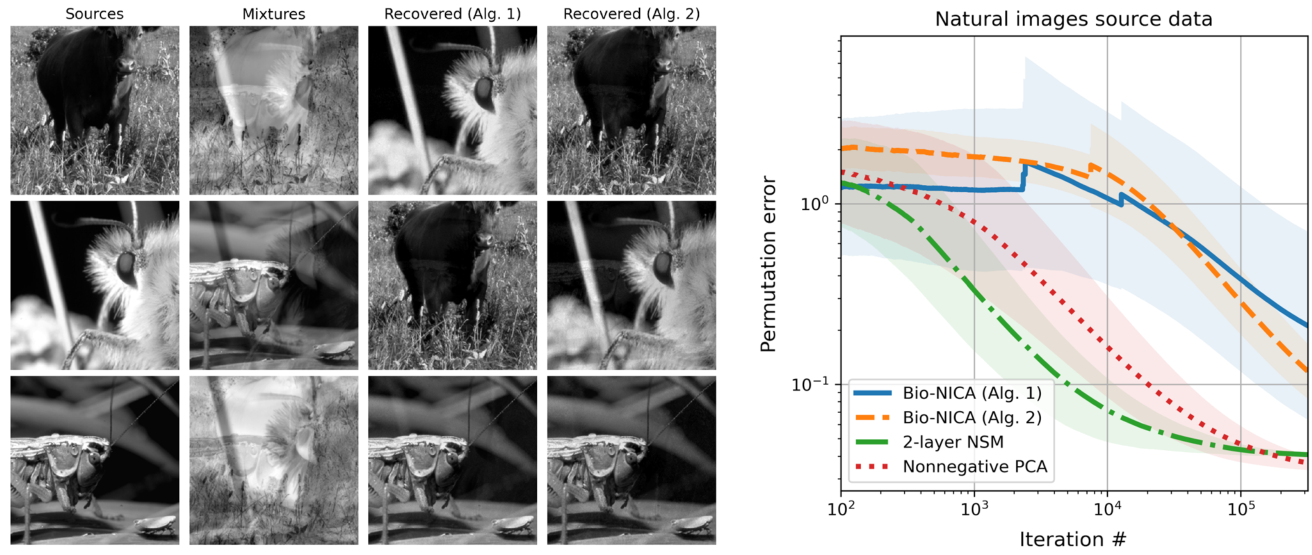

Fig. 5 Performance of algorithms when presented with mixtures of natural images. The left image shows the sources, mixtures, and recovered sources (from Algorithms 1 and 2). The right plot shows the performance of the algorithms in terms of permutation error. The lines and shaded regions denote the means and 90% confidence intervals over 10 runs

$\mathbf{W}_{XZ}(\mathbf{x}_t - \bar{\mathbf{x}}_t)(\mathbf{x}_t - \bar{\mathbf{x}}_t)$ is not equal to $(\mathbf{c}_t - \bar{\mathbf{c}}_t)(\mathbf{x}_t - \bar{\mathbf{x}}_t)$ due to the (suppressed) time-dependency of the weights $\mathbf{W}_{XZ}$. Finally, we thank the two referees for their careful reading of our paper and for their helpful comments.

A Saddle point property

Here we recall the following minmax property for a function that satisfies the saddle point property (Boyd and Vandenberghe 2004, section 5.4).

Theorem 1 *Let $V \subseteq \mathbb{R}^n$, $W \subseteq \mathbb{R}^m$ and $f : V \times W \to \mathbb{R}$. Suppose f satisfies the saddle point property; that is, there exists $(\mathbf{a}^*, \mathbf{b}^*) \in V \times W$ such that*

$$f(\mathbf{a}^*, \mathbf{b}) \leq f(\mathbf{a}^*, \mathbf{b}^*) \leq f(\mathbf{a}, \mathbf{b}^*),$$
$$\text{for all } (\mathbf{a}, \mathbf{b}) \in V \times W.$$

Then

$$\min_{\mathbf{a}\in V} \max_{\mathbf{b}\in W} f(\mathbf{a}, \mathbf{b}) = \max_{\mathbf{b}\in W} \min_{\mathbf{a}\in V} f(\mathbf{a}, \mathbf{b}) = f(\mathbf{a}^*, \mathbf{b}^*).$$

B Optimization over neural activity matrices (Y, N) in the derivation of Algorithm 1

In this section, we show that when $\mathbf{W}_{XY}$ and $\mathbf{W}_{YN}$ are at their optimal values, the optimal neural activity matrices $(\mathbf{Y}, \mathbf{N})$ can be approximated via projected gradient descent. We first compute that optimal values of $\mathbf{W}_{XY}$ and $\mathbf{W}_{YN}$.

Lemma 1 *Suppose $(\mathbf{W}^*_{XY}, \mathbf{W}^*_{YN}, \mathbf{Y}^*, \mathbf{N}^*)$ is an optimal solution of objective (5). Then*

$$\mathbf{W}^*_{XY} = \mathbf{P}\mathbf{A}^\top, \qquad \mathbf{W}^{*,\top}_{YN}\mathbf{W}^*_{YN} = \mathbf{P}\mathbf{A}^\top\mathbf{A}\mathbf{P}^\top,$$

for some permutation matrix $\mathbf{P}$.

Proof From (Pehlevan et al. 2017, Theorem 3), we know that every solution of the objective

$$\begin{aligned}&\underset{\mathbf{Y}\in\mathbb{R}^{d\times T}}{\arg\min} \; -\operatorname{Tr}(\delta\mathbf{Y}^\top\delta\mathbf{Y}\delta\mathbf{X}^\top\delta\mathbf{X})\\ &\quad\text{subject to} \quad \delta\mathbf{Y}^\top\delta\mathbf{Y} \preceq T\mathbf{I}_T \quad \text{and} \quad \mathbf{Y} = \mathbf{F}\mathbf{X},\end{aligned} \tag{16}$$

is of the form $\mathbf{Y} = \mathbf{F}\mathbf{X}$, where $\mathbf{F}$ is a whitening matrix. In particular, since $\mathbf{Y} = \mathbf{F}\mathbf{A}\mathbf{S}$ and $\mathbf{S}$ also has identity covariance matrix, $\mathbf{Y}$ is an orthogonal transformation of $\mathbf{S}$. Furthermore, since $\mathbf{S}$ is well grounded, by (Plumbley 2002, Theorem 1), $\mathbf{Y}$ is nonnegative if and only if $\mathbf{F}\mathbf{A}$ is a permutation matrix. Therefore, every solution $\mathbf{Y}^*$ of the objective

$$\begin{aligned}&\underset{\mathbf{Y}\in\mathbb{R}_+^{d\times T}}{\arg\min} \; -\operatorname{Tr}(\delta\mathbf{Y}^\top\delta\mathbf{Y}\delta\mathbf{X}^\top\delta\mathbf{X})\\ &\quad\text{subject to} \quad \delta\mathbf{Y}^\top\delta\mathbf{Y} \preceq T\mathbf{I}_T \quad \text{and} \quad \mathbf{Y} = \mathbf{F}\mathbf{X},\end{aligned} \tag{17}$$

is of the form $\mathbf{Y}^* = \mathbf{P}\mathbf{X}$ for some permutation matrix $\mathbf{P}$. In addition, differentiating the expression

$$-\operatorname{Tr}(\delta\mathbf{Y}^\top\delta\mathbf{Y}\delta\mathbf{X}^\top\delta\mathbf{X} + \delta\mathbf{N}^\top\delta\mathbf{N}(\delta\mathbf{Y}^\top\delta\mathbf{Y} - T\mathbf{I}_T)), \tag{18}$$

with respect to $\delta\mathbf{Y}$ and setting the derivative equal to zero, we see that the at the optimal value, $\delta\mathbf{N}^{*,\top}\delta\mathbf{N}^{*} = \delta\mathbf{X}^{\top}\delta\mathbf{X} = \delta\mathbf{S}^{\top}\mathbf{A}^{\top}\mathbf{A}\delta\mathbf{S}$.

Differentiating L_1 with respect to $\mathbf{W}_{XY}$ and $\mathbf{W}_{NY}$, we see that the optimal values for the synaptic weight matrices are achieved at $\mathbf{W}_{XY} = \frac{1}{T}\delta\mathbf{Y}\delta\mathbf{X}^{\top}$ and $\mathbf{W}_{YN} = \frac{1}{T}\delta\mathbf{N}\delta\mathbf{Y}^{\top}$. Thus,

$$\mathbf{W}^{*}_{XY} = \frac{1}{T}\delta\mathbf{Y}^{*}\delta\mathbf{X}^{\top} = \frac{1}{T}\mathbf{P}\delta\mathbf{S}\delta\mathbf{S}^{\top}\mathbf{A}^{\top} = \mathbf{P}\mathbf{A}^{\top},$$

and

$$\begin{aligned}\mathbf{W}^{*,\top}_{YN}\mathbf{W}^{*}_{YN} &= \frac{1}{T^2}\delta\mathbf{Y}\delta\mathbf{N}^{*,\top}\delta\mathbf{N}^{*}\delta\mathbf{Y}^{\top}\\ &= \frac{1}{T^2}\mathbf{P}\delta\mathbf{S}\delta\mathbf{S}^{\top}\mathbf{A}^{\top}\mathbf{A}\delta\mathbf{S}\delta\mathbf{S}^{\top}\mathbf{P}^{\top} = \mathbf{P}\mathbf{A}^{\top}\mathbf{A}\mathbf{P}^{\top}.\end{aligned}$$

□

Next, we show that when $\mathbf{W}_{XY}$ and $\mathbf{W}_{YN}$ are at their optimal values, the optimal $(\mathbf{Y}^{*}, \mathbf{N}^{*})$ can be approximated by running the projected gradient dynamics in Eq. (6).

Lemma 2 *Suppose* $\mathbf{W}_{XY} = \mathbf{P}\mathbf{A}^{\top}$ *and* $\mathbf{W}_{YN}^{\top}\mathbf{W}_{YN} = \mathbf{P}\mathbf{A}^{\top}\mathbf{A}\mathbf{P}^{\top}$ *for some permutation matrix* $\mathbf{P}$. *Then*

$$\mathbf{Y}^{*} = (\mathbf{W}_{YN}^{\top}\mathbf{W}_{YN})^{-1}\mathbf{W}_{XY}\mathbf{X} = \mathbf{P}\mathbf{S}, \quad \mathbf{N}^{*} = \mathbf{W}_{YN}\mathbf{Y}^{*}. \quad (19)$$

is a solution of the minmax problem

$$\begin{aligned}&\min_{\mathbf{Y}\in\mathbb{R}_{+}^{d\times T}}\max_{\mathbf{N}\in\mathbb{R}^{m\times T}}\frac{2}{T}\operatorname{Tr}\Big(\delta\mathbf{N}^{\top}\mathbf{W}_{YN}\delta\mathbf{Y}\\ &\quad-\delta\mathbf{Y}^{\top}\mathbf{W}_{XY}\delta\mathbf{X}-\delta\mathbf{N}^{\top}\delta\mathbf{N}\Big)\\ &\text{s.t.}\quad \mathbf{Y} = \mathbf{F}\mathbf{X}. \qquad (20)\end{aligned}$$

In particular, $(\mathbf{Y}^{*}, \mathbf{N}^{*})$ *is the unique solution of the minmax problem*

$$\min_{\mathbf{Y}\in\mathbb{R}_{+}^{d\times T}}\max_{\mathbf{N}\in\mathbb{R}^{m\times T}}\frac{2}{T}\operatorname{Tr}\left(\mathbf{N}^{\top}\mathbf{W}_{YN}\mathbf{Y}-\mathbf{Y}^{\top}\mathbf{W}_{XY}\mathbf{X}-\mathbf{N}^{\top}\mathbf{N}\right), \quad (21)$$

which can be approximated by running the projected gradient dynamics in Eq. (6).

Proof We first relax the condition that $\mathbf{Y}$ be a nonnegative linear transformation of $\mathbf{X}$ and consider the minmax problem

$$\begin{aligned}&\min_{\mathbf{Y}\in\mathbb{R}^{d\times T}}\max_{\mathbf{N}\in\mathbb{R}^{d\times T}}\frac{2}{T}\operatorname{Tr}\Big(\delta\mathbf{N}^{\top}\mathbf{W}_{YN}\delta\mathbf{Y}\\ &\quad-\delta\mathbf{Y}^{\top}\mathbf{W}_{XY}\delta\mathbf{X}-\delta\mathbf{N}^{\top}\delta\mathbf{N}\Big).\end{aligned}$$

After differentiating with respect to $\delta\mathbf{Y}$ and $\delta\mathbf{N}$, we see that this objective is optimized when the centered matrices $\delta\mathbf{Y}$ and $\delta\mathbf{N}$ are given by

$$\delta\mathbf{Y} = (\mathbf{W}_{YN}^{\top}\mathbf{W}_{YN})^{-1}\mathbf{W}_{XY}\delta\mathbf{X}, \qquad \delta\mathbf{N} = \mathbf{W}_{YN}\delta\mathbf{Y}.$$

Next, we see that the above relations for the centered matrices hold when $\mathbf{Y}$ and $\mathbf{N}$ are given by Eq. (19), where we have used the fact that $\mathbf{W}_{XY} = \mathbf{P}\mathbf{A}^{\top}$ and $\mathbf{W}_{YN}^{\top}\mathbf{W}_{YN} = \mathbf{P}\mathbf{A}^{\top}\mathbf{A}\mathbf{P}^{\top}$. Note that $\mathbf{Y}$ is a linear transformation of $\mathbf{X}$ and $\mathbf{Y}$ is nonnegative since it is a permutation of the nonnegative sources. It follows that $(\mathbf{Y}, \mathbf{N})$ is also a solution to the *constrained* minmax problem (20). Finally, differentiating the objective in Eq. (21) with respect to $\mathbf{Y}$ and $\mathbf{N}$, we see that the optimal $\mathbf{Y}$ and $\mathbf{N}$ are again given by Eq. (19). □

C Decoupling the interneuron synapses

The NICA algorithm derived in Sect. 4.1 requires the interneuron-to-output neuron synaptic weight matrix $\mathbf{W}_{NY}$ to be the transpose of the output neuron-to-interneuron synaptic weight matrix $\mathbf{W}_{YN}$. Enforcing this symmetry via a centralized mechanism is not biologically plausible and is commonly referred to as the weight transport problem.

Here, we show that the symmetry of the 2 weights asymptotically follows from the learning rules in Algorithm 1, even when the symmetry does not hold at initialization. Let $\mathbf{W}_{NY,0}$ and $\mathbf{W}_{YN,0}$ denote the initial values of $\mathbf{W}_{NY}$ and $\mathbf{W}_{YN}$. Then, in view of the updates rules in Algorithm 1, the difference $\mathbf{W}_{NY} - \mathbf{W}_{YN}^{\top}$ after t updates is given by

$$\mathbf{W}_{NY} - \mathbf{W}_{YN}^{\top} = (1-\eta)^{t}\,(\mathbf{W}_{NY,0} - \mathbf{W}_{YN,0}^{\top}).$$

In particular, the difference decays exponentially.

D Details of numerical experiments

The simulations were performed on an Apple iMac with a 2.8 GHz Quad-Core Intel Core i7 processor. For each of the algorithms that we implement, we use a time-dependent learning rate of the form:

$$\eta_t = \frac{\eta_0}{1+\gamma t}. \qquad (22)$$

To choose the parameters, we perform a grid search over $\eta_0 \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ and over $\gamma \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$. In Table 1 we report the best performing hyperparameters we found for each algorithm. We now detail our implementation of each algorithm.

Table 1 Optimal hyperparameters used for Algorithm 1, Algorithm 2, 2-layer NSM, and Nonnegative PCA

	Alg. 1 (η_0, γ)	Alg. 2 (η_0, γ, τ)	2-layer NSM (η_0, γ)	NPCA (η_0, γ)
$d = 3$	$(10^{-2}, 10^{-3})$	$(10^{-1}, 10^{-2}, 0.8)$	$(10^{-1}, 10^{-7})$	$(10^{-2}, 10^{-5})$
$d = 5$	$(10^{-2}, 10^{-5})$	$(10^{-2}, 10^{-2}, 0.05)$	$(10^{-1}, 10^{-6})$	$(10^{-1}, 10^{-2})$
$d = 7$	$(10^{-2}, 10^{-4})$	$(10^{-3}, 10^{-4}, 0.05)$	$(10^{-1}, 10^{-6})$	$(10^{-2}, 10^{-6})$
$d = 10$	$(10^{-2}, 10^{-3})$	$(10^{-3}, 10^{-4}, 0.03)$	$(10^{-1}, 10^{-6})$	$(10^{-2}, 10^{-5})$
Images	$(10^{-3}, 10^{-6})$	$(10^{-2}, 10^{-4}, 0.5)$	$(10^{-1}, 10^{-6})$	$(10^{-3}, 10^{-5})$

1. **Bio-NICA with interneurons (Algorithm 1):** The neural outputs were computed using the quadratic convex optimization function `solve_qp` from the Python package `quadprog`. After each iteration, we checked if any output neuron had not been active up until that iteration. If so, we flipped the sign of its feedforward inputs. In addition, if the norm of one of the row vectors of $\mathbf{W}_{XY}$ fell below 0.1, we would replace the row vector with a random vector to avoid the row vector becoming degenerate, and if a singular value of $\mathbf{W}_{XY}$, $\mathbf{W}_{YN}$ or $\mathbf{W}_{NY}$ fell below 0.01, we replaced the singular value with 1 (we checked every 100 iterations).
2. **Bio-NICA with 2-compartmental neurons (Algorithm 2):** The neural outputs were computed using the quadratic convex optimization function `solve_qp` from the Python package `quadprog`. We used the time-dependent learning rate of Eq. (22) and included $\tau \in \{0.01, 0.03, 0.05, 0.08, 0.1, 0.3, 0.5, 0.8, 1, 3\}$ in the grid search to find the best performance. After each iteration, we checked if any output neuron had not been active up until that iteration. If so, we flipped the sign of its feedforward inputs. In addition, if an eigenvalue of $\mathbf{W}_{ZZ}$ fell below 0.01, we replaced the eigenvalue with 1 to prevent $\mathbf{W}_{ZZ}$ from becoming degenerate (we checked every 100 iterations).
3. **2-layer NSM:** We implemented the algorithm in Pehlevan et al. (2017) with time-dependent learning rates. For the whitening layer, we used the optimal time-dependent learning rate reported in Pehlevan et al. (2017): $\zeta_t = 0.01/(1 + 0.01t)$. For the NSM layer, we used the time-dependent learning rate of Eq. (22). To compute the neuronal outputs, we used the quadratic convex optimization function `solve_qp` from the Python package `quadprog`. After each iteration, we checked if any output neuron had not been active up until that iteration. If so, we flipped the sign of its feedforward inputs.
4. **Nonnegative PCA (NPCA):** We use the online version given in Plumbley and Oja (2004). The algorithm assumes the inputs are noncentered and whitened. We performed the noncentered whitening offline. After each iteration, we checked if any output neuron had not been active up until that iteration. If so, we flipped the sign of its feedforward inputs.

References

Bahroun Y, Chklovskii D, Sengupta A (2021) A normative and biologically plausible algorithm for independent component analysis. Adv Neural Inf Process Syst 34:7368–7384

Bee MA, Micheyl C (2008) The cocktail party problem: what is it? How can it be solved? And why should animal behaviorists study it? J Comp Psychol 122(3):235

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge

Bronkhorst AW (2015) The cocktail-party problem revisited: early processing and selection of multi-talker speech. Atten Percept Psychophys 77(5):1465–1487

Colin CE (1953) Some experiments on the recognition of speech, with one and with two ears. J Acoust Soc Am 25(5):975–979

Desimone R, Duncan J (1995) Neural mechanisms of selective visual attention. Ann Rev Neurosci 18(1):193–222

Donoho D, Stodden V (2003) When does non-negative matrix factorization give a correct decomposition into parts? Advances in neural information processing systems, 16

Erdogan AT, Pehlevan C (2020) Blind bounded source separation using neural networks with local learning rules. In ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, pp 3812–3816

Friedrich RW, Gilles L (2001) Dynamic optimization of odor representations by slow temporal patterning of mitral cell activity. Science 291(5505):889–894

Gschwend O, Abraham NM, Lagier S, Begnaud F, Rodriguez I, Carleton A (2015) Neuronal pattern separation in the olfactory bulb improves odor discrimination learning. Nat Neurosci 18(10):1474–1482

Huang K, Sidiropoulos ND, Swami A (2013) Non-negative matrix factorization revisited: uniqueness and algorithm for symmetric decomposition. IEEE Trans Signal Process 62(1):211–224

Hulse SH, MacDougall-Shackleton SA, Wisniewski AB (1997) Auditory scene analysis by songbirds: stream segregation of birdsong by European starlings (Sturnus vulgaris). J Comp Psychol 111(1):3

Hyvärinen A, Hoyer P (2000) Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. Neural Comput 12(7):1705–1720

Hyvärinen A, Oja E (2000) Independent component analysis: algorithms and applications. Neural Netw 13(4–5):411–430

Laughlin SB, Sejnowski TJ (2003) Communication in neuronal networks. Science 301(5641):1870–1874

Laurberg H Christensen MG, Plumbley MD, Hansen LK, and Jensen SH (2008). On the uniqueness of NMF. Computational intelligence and neuroscience, Theorems on positive data, p 2008

Lipshutz D, Windolf C, Golkar S, Chklovskii DB (2020) A biologically plausible neural network for slow feature analysis. Adv Neural Inf Process Syst 33:14986–14996

Lipshutz D, Bahroun Y, Golkar S, Sengupta AM, Chklovskii DB (2021) A biologically plausible neural network for multichannel canonical correlation analysis. Neural Comput 33(9):2309–2352

McDermott JH (2009) The cocktail party problem. Current Biol 19(22):R1024–R1027

Narayan R, Best V, Ozmeral E, McClaine E, Dent M, Shinn-Cunningham B, Sen K (2007) Cortical interference effects in the cocktail party problem. Nat Neurosci 10(12):1601–1607

Oja E, Plumbley M (2004) Blind separation of positive sources by globally convergent gradient search. Neural Comput 16(9):1811–1825

Pehlevan C, Chklovskii DB (2019) Neuroscience-inspired online unsupervised learning algorithms: artificial neural networks. IEEE Signal Process Mag 36(6):88–96

Pehlevan C, Mohan S, Chklovskii DB (2017) Blind nonnegative source separation using biological neural networks. Neural Comput 29(11):2925–2954

Plumbley M (2002) Conditions for nonnegative independent component analysis. IEEE Signal Process Lett 9(6):177–180

Plumbley MD (2003) Algorithms for nonnegative independent component analysis. IEEE Trans Neural Netw 14(3):534–543

Plumbley MD, Oja E (2004) A“nonnegative PCA”algorithm for independent component analysis. IEEE Trans Neural Netw 15(1):66–76

Rivera-Alba M, Hanchuan P, de Polavieja GG, Chklovskii DB (2014) Wiring economy can account for cell body placement across species and brain areas. Current Biol 24(3):R109–R110

Shinn-Cunningham BG (2008) Object-based auditory and visual attention. Trends Cognit Sci 12(5):182

Wilson RI, Mainen ZF (2006) Early events in olfactory processing. Ann Rev Neurosci 29:163–201

Yuan Z, Oja E (2004) A fastICA algorithm for non-negative independent component analysis. In international conference on independent component analysis and signal separation, Springer, pp 1–8

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.