INTERNEURONS ACCELERATE LEARNING DYNAMICS IN RECURRENT NEURAL NETWORKS FOR STATISTICAL ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Early sensory systems in the brain rapidly adapt to fluctuating input statistics, which requires recurrent communication between neurons. Mechanistically, such recurrent communication is often indirect and mediated by local interneurons. In this work, we explore the computational benefits of mediating recurrent communication via interneurons compared with direct recurrent connections. To this end, we consider two mathematically tractable recurrent linear neural networks that statistically whiten their inputs — one with direct recurrent connections and the other with interneurons that mediate recurrent communication. By analyzing the corresponding continuous synaptic dynamics and numerically simulating the networks, we show that the network with interneurons is more robust to initialization than the network with direct recurrent connections in the sense that the convergence time for the synaptic dynamics in the network with interneurons (resp. direct recurrent connections) scales logarithmically (resp. linearly) with the spectrum of their initialization. Our results suggest that interneurons are computationally useful for rapid adaptation to changing input statistics. Interestingly, the network with interneurons is an overparameterized solution of the whitening objective for the network with direct recurrent connections, so our results can be viewed as a recurrent linear neural network analogue of the implicit acceleration phenomenon observed in overparameterized feedforward linear neural networks.

1 INTRODUCTION

Efficient coding and redundancy reduction theories of neural coding hypothesize that early sensory systems decorrelate and normalize neural responses to sensory inputs (Barlow, 1961; Laughlin, 1989; Barlow & Földiák, 1989; Simoncelli & Olshausen, 2001; Carandini & Heeger, 2012; Westrick et al., 2016), operations that are closely related to statistical whitening of inputs. Since the input statistics are often in flux due to dynamic environments, this calls for early sensory systems that can rapidly adapt (Wark et al., 2007; Whitmire & Stanley, 2016). Decorrelating neural activities requires recurrent communication between neurons in a population, which is typically indirect and mediated by local interneurons (Christensen et al., 1993; Shepherd et al., 2004). Why do neuronal circuits for statistical adaptation mediate recurrent communication using interneurons, which take up valuable space and metabolic resources, rather than using direct recurrent connections?

A common explanation for why communication between neurons is mediated by interneurons is Dale's principle, which states that each neuron only produces one type of neurotransmitter with exclusively inhibitory or excitatory effects on all of its targets (Strata & Harvey, 1999). Therefore, an excitatory neuron cannot directly inhibit another neuron; rather, the inhibition must be mediated by an inhibitory interneuron. While Dale's principle provides a physiological constraint that explains why recurrent interactions are mediated by interneurons, we seek a computational principle that can account for using interneurons rather than direct recurrent connections. This perspective is useful for a couple of reasons. First, perhaps Dale's principle is not a hard constraint and biology could have found an efficient mechanism for a neuron to produce more than one type of neurotransmitter. In this case, a computational benefit of interneurons would provide a normative explanation for the existence of interneurons to mediate recurrent communication. Second, whitening is useful in statistical and machine learning methods (Hyvärinen & Oja, 2000; Krizhevsky, 2009; Ermolov

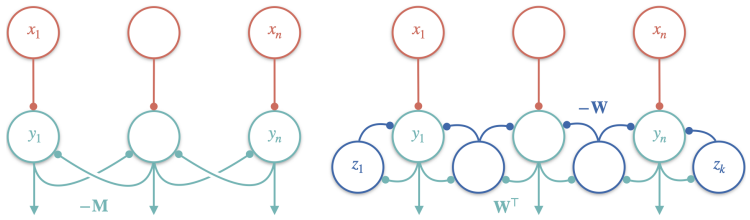


Figure 1: Recurrent neural networks for ZCA whitening with direct recurrent connections (left, Algorithm 1) and with interneurons (right, Algorithm 2).

et al., 2021), so our analysis is potentially relevant to the design of artificial neural networks and neuromorphic systems, which are not bound by Dale's principle.

In this work, to better understand the computational benefits of interneurons for statistical adaptation, we analyze the learning dynamics of two mathematically tractable recurrent neural networks that whiten their inputs using Hebbian/anti-Hebbian learning rules — one with direct recurrent connections and the other with indirect recurrent interactions mediated by interneurons, Figure 1. We show that the learning dynamics of the network with interneurons are more robust than the learning dynamics of the network with direct recurrent connections. In particular, we prove that the convergence time of the continuum limit of the network with direct lateral connections scales linearly with the spectrum of the initialization, whereas the convergence time of the continuum limit of the network with interneurons scales logarithmically with the spectrum of the initialization. We also numerically test the networks and, consistent with the theoretical results, find that the network with interneurons is more robust to initialization. Our results suggest that interneurons are computationally important for rapid adaptation to fluctuating input statistics.

Our analysis is closely related to analyses of learning dynamics in feedforward linear networks trained using backpropagation (Saxe et al., 2014; Arora et al., 2018; Saxe et al., 2019; Gidel et al., 2019; Tarmoun et al., 2021). The optimization problems for deep linear networks are overparameterizations of linear problems and this overparameterization can accelerate convergence of gradient descent or gradient flow optimization — a phenomenon referred to as *implicit acceleration*. Our results can be viewed as an analogous phenomenon for the gradient flows corresponding to recurrent linear networks trained using Hebbian/anti-Hebbian learning rules. In our setting, the network with interneurons is naturally viewed as an overparamterized solution of the whitening objective for the network with direct recurrent connections. In analogy with the feedforward setting, the interneurons can be viewed as a hidden layer that overparameterizes the optimization problem.

2 Statistical Whitening

Let $\mathbf{x}_1, \ldots, \mathbf{x}_T$ be a sequence of n-dimensional centered inputs, for some $n \geq 1$, with positive definite (empirical) covariance matrix $\mathbf{C}_{xx} := \frac{1}{T}\mathbf{X}\mathbf{X}^\top$, where $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_T]$ is the $n \times T$ data matrix of concatenated inputs. Whitening is a linear transformation of the inputs to have identity covariance matrix. In particular, an $n \times n$ matrix $\mathbf{F}$ is a *whitening matrix* if the outputs $\mathbf{y}_t := \mathbf{F}\mathbf{x}_t$, for $t = 1, \ldots, T$, have identity covariance matrix; that is, $\mathbf{C}_{yy} := \frac{1}{T}\mathbf{Y}\mathbf{Y}^\top = \mathbf{I}_n$, where $\mathbf{Y} := [\mathbf{y}_1, \ldots, \mathbf{y}_T] = \mathbf{F}\mathbf{X}$ is the data matrix of concatenated outputs and $\mathbf{I}_n$ denotes the $n \times n$ identity matrix.

Whitening is not a unique transformation. Let $\mathbf{X} = \mathbf{U}_x(\sqrt{T}\mathbf{S}_x)\mathbf{V}_x^\top$ be the singular value decomposition of $\mathbf{X}$, where $\mathbf{U}_x$ is an $n \times n$ orthogonal matrix, $\sqrt{T}\mathbf{S}_x = \operatorname{diag}(\sqrt{T}\lambda_1, \ldots, \sqrt{T}\lambda_n)$ is the $n \times n$ diagonal matrix of singular values and $\mathbf{V}_x$ is an $T \times n$ matrix with orthonormal column vectors. Then every whitening matrix $\mathbf{F}$ is of the form $\mathbf{F} = \mathbf{Q}\mathbf{S}_x^{-1}\mathbf{U}_x^\top$, where $\mathbf{Q}$ is an arbitrary $n \times n$ orthogonal matrix (Kessy et al., 2018).

We focus on ZCA whitening (Bell & Sejnowski, 1997), also referred to as Mahalanobis whitening, which corresponds to the case $\mathbf{Q} = \mathbf{U}_x$, so that

$$\mathbf{F}_{\text{ZCA}} = \mathbf{U}_x\mathbf{S}_x^{-1}\mathbf{U}_x^\top = \mathbf{C}_{xx}^{-1/2}, \qquad \mathbf{Y}_{\text{ZCA}} := \mathbf{F}_{\text{ZCA}}\mathbf{X} = \mathbf{U}_x\mathbf{V}_x^\top.$$

ZCA whitening is a common choice of whitening transformation used in statistical analysis and machine learning (Krizhevsky, 2009; Huang et al., 2018; Ermolov et al., 2021) that minimizes the mean-square error between $\mathbf{X}$ and $\mathbf{Y}$ (Eldar & Oppenheim, 2003). In other words, $\mathbf{Y}_{\text{ZCA}}$ is the unique solution to the objective

$$\underset{\mathbf{Y}\in\mathbb{R}^{n\times T}}{\arg\min}\,\frac{1}{T}\|\mathbf{Y}-\mathbf{X}\|^2_{\text{Frob}} \qquad \text{subject to} \qquad \frac{1}{T}\mathbf{Y}\mathbf{Y}^\top=\mathbf{I}_n. \tag{1}$$

The goal of this work is a theoretical analysis of 2 recurrent neural networks that perform ZCA whitening in the online or streaming setting using Hebbian/anti-Hebbian learning rules. The derivations of the 2 neural networks, which we include here for completeness, are closely related to derivations of PCA networks carried out in (Pehlevan & Chklovskii, 2015; Pehlevan et al., 2018). Our main theoretical and numerical results are presented in sections 6 and 7, respectively.

3 Objectives for deriving ZCA whitening networks

In this section, we rewrite the ZCA whitening objective 1 to obtain 2 objectives that will be the starting points for deriving our 2 recurrent whitening networks.

3.1 Objective for the network with direct recurrent connections

To derive a network with direct recurrent connections, we rewrite objective 1 to include the synaptic weight matrix of direct recurrent connections $-\mathbf{M}$. Expanding the square in equation 1, substituting in with the constraint $\mathbf{Y}\mathbf{Y}^\top=T\mathbf{I}_n$, dropping the terms that do not depend on $\mathbf{Y}$ and flipping the sign yields the following objective in terms of the neural activities data matrix $\mathbf{Y}$:

$$\max_{\mathbf{Y}\in\mathbb{R}^{n\times T}}\frac{2}{T}\operatorname{Tr}(\mathbf{Y}\mathbf{X}^\top) \qquad \text{subject to} \qquad \frac{1}{T}\mathbf{Y}\mathbf{Y}^\top=\mathbf{I}_n.$$

Next, we introduce the positive definite matrix $\mathbf{M}$ as a Lagrange multiplier to enforce $\mathbf{Y}\mathbf{Y}^\top\preceq T\mathbf{I}_n$:

$$\max_{\mathbf{Y}\in\mathbb{R}^{n\times T}}\min_{\mathbf{M}\in\mathcal{S}^n_{++}} f(\mathbf{M},\mathbf{Y}), \qquad f(\mathbf{M},\mathbf{Y}):=\frac{2}{T}\operatorname{Tr}(\mathbf{Y}\mathbf{X}^\top)-\frac{1}{T}\operatorname{Tr}(\mathbf{M}(\mathbf{Y}\mathbf{Y}^\top-T\mathbf{I}_n)),$$

where $\mathcal{S}^n_{++}$ denotes the set of positive definite $n\times n$ matrices. The maximization over $\mathbf{Y}$ ensures that the upper bound $\mathbf{Y}\mathbf{Y}^\top\preceq T\mathbf{I}_n$ is in fact saturated. Since $f(\mathbf{M},\mathbf{Y})$ is linear in $\mathbf{M}$ and strongly concave in $\mathbf{Y}$, it satisfies the saddle point property (see Appendix A), so we can interchange the order of optimization. This results in the minimax objective

$$\min_{\mathbf{M}\in\mathcal{S}^n_{++}}\max_{\mathbf{Y}\in\mathbb{R}^{n\times T}} f(\mathbf{M},\mathbf{Y}), \tag{2}$$

which is a function of neural activities $\mathbf{Y}$ and synaptic weights $\mathbf{M}$, and will be the starting point for our derivation of a whitening network with direct recurrent connections.

3.2 Overparameterized objective for the network with interneurons

In order to derive a network with $k\geq n$ interneurons, whose activities are encoded in the $k\times T$ data matrix $\mathbf{Z}=[\mathbf{z}_1,\dots,\mathbf{z}_T]$, we replace the matrix $\mathbf{M}$ in equation 2 with the overparameterized matrix product $\mathbf{W}\mathbf{W}^\top$, where $\mathbf{W}$ is an $n\times k$ matrix, which yields the minimax objective

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times k}}\max_{\mathbf{Y}\in\mathbb{R}^{n\times T}} f(\mathbf{W}\mathbf{W}^\top,\mathbf{Y}).$$

The matrix $-\mathbf{W}$ (resp. $\mathbf{W}^\top$) will correspond to the synaptic weights between the interneurons (resp. principal neurons) and principal neurons (resp. interneurons). Finally, in order to arrive at an objective that also includes the interneuron activities $\mathbf{Z}$ we substitute in with the Legendre transform:

$$-\operatorname{Tr}(\mathbf{W}\mathbf{W}^\top\mathbf{Y}\mathbf{Y}^\top)=\min_{\mathbf{Z}\in\mathbb{R}^{k\times T}}\left\{-2\operatorname{Tr}(\mathbf{Y}^\top\mathbf{W}\mathbf{Z})+\operatorname{Tr}(\mathbf{Z}\mathbf{Z}^\top)\right\}.$$

This results in the objective

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times k}}\max_{\mathbf{Y}\in\mathbb{R}^{n\times T}}\min_{\mathbf{Z}\in\mathbb{R}^{k\times T}} g(\mathbf{W},\mathbf{Y},\mathbf{Z}), \tag{3}$$

where

$$g(\mathbf{W}, \mathbf{Y}, \mathbf{Z}) := \frac{2}{T}\operatorname{Tr}(\mathbf{Y}\mathbf{X}^\top) - \frac{2}{T}\operatorname{Tr}(\mathbf{Y}^\top\mathbf{W}\mathbf{Z}) + \frac{1}{T}\operatorname{Tr}(\mathbf{Z}\mathbf{Z}^\top) + \operatorname{Tr}(\mathbf{W}\mathbf{W}^\top),$$

which is a function of neural activities $\mathbf{Y}, \mathbf{Z}$ and synaptic weights $\mathbf{W}$, and will be the starting point for the derivation of a whitening network with interneurons.

4 Derivation of a network with direct recurrent connections

Starting from the whitening objective 2, we derive an online algorithm that maps onto a network with direct recurrent connections.

4.1 Offline algorithm

To derive our algorithms, we assume the recurrent neural dynamics operate at a much faster timescale than the synaptic updates. In the offline setting, at each iteration, we first maximize $f(\mathbf{M}, \mathbf{Y})$ with respect to the neural activities $\mathbf{Y}$ by taking gradient ascent steps until convergence:

$$\mathbf{Y} \leftarrow \mathbf{Y} + \gamma(\mathbf{X} - \mathbf{M}\mathbf{Y}) \qquad \Rightarrow \qquad \mathbf{Y} = \mathbf{M}^{-1}\mathbf{X},$$

where $\gamma > 0$ is a small constant. After the neural activities converge, we minimize $f(\mathbf{M}, \mathbf{Y})$ with respect to $\mathbf{M}$ by taking a gradient descent step with respect to $\mathbf{M}$:

$$\mathbf{M} \leftarrow \mathbf{M} + \eta\left(\frac{1}{T}\mathbf{Y}\mathbf{Y}^\top - \mathbf{I}_n\right). \tag{4}$$

4.2 Online algorithm

In the online setting, at each time step t, we only have access to the input $\mathbf{x}_t$. In this case, we first optimize $f(\mathbf{M}, \mathbf{Y})$ with respect to the neural activities vector $\mathbf{y}_t$ by running the fast neural dynamics until convergence:

$$\mathbf{y}_t \leftarrow \mathbf{y}_t + \gamma(\mathbf{x}_t - \mathbf{M}\mathbf{y}_t) \qquad \Rightarrow \qquad \mathbf{y}_t = \mathbf{M}^{-1}\mathbf{x}_t.$$

After the neural activities converge, the synaptic weight matrix $\mathbf{M}$ is updated by taking a stochastic gradient descent step:

$$\mathbf{M} \leftarrow \mathbf{M} + \eta(\mathbf{y}_t\mathbf{y}_t^\top - \mathbf{I}_n).$$

This results in Algorithm 1, which can be implemented in a network with direct recurrent connections $-\mathbf{M}$, Figure 1 (left).

Algorithm 1: A whitening network with direct recurrent connections

input centered inputs $\{\mathbf{x}_t\}$; parameters γ, η
initialize positive definite $\mathbf{M}$
for $t = 1, \ldots, T$ **do**
  repeat
    $\mathbf{y}_t \leftarrow \mathbf{y}_t + \gamma(\mathbf{x}_t - \mathbf{M}\mathbf{y}_t)$
  until convergence
  $\mathbf{M} \leftarrow \mathbf{M} + \eta(\mathbf{y}_t\mathbf{y}_t^\top - \mathbf{I}_n)$
end for

5 Derivation of a network with interneurons

In this section, starting from the overeparameterized whitening objective 3, we derive an online algorithm that maps onto a network with interneurons.

5.1 OFFLINE ALGORITHM

We first optimize $g(\mathbf{W}, \mathbf{Y}, \mathbf{Z})$ in the offline setting by alternating optimizing over the neural activities data matrices $(\mathbf{Y}, \mathbf{Z})$ and taking a gradient descent step with respect to synaptic weight matrix $\mathbf{W}$. We first optimize $g(\mathbf{W}, \mathbf{Y}, \mathbf{Z})$ with respect to the neural activities data matrices $(\mathbf{Y}, \mathbf{Z})$ by taking gradient steps:

$$\mathbf{Y} \leftarrow \mathbf{Y} + \gamma(\mathbf{X} - \mathbf{W}\mathbf{Z}), \qquad \mathbf{Z} \leftarrow \mathbf{Z} + \gamma(\mathbf{W}^\top \mathbf{Y} - \mathbf{Z}).$$

which eventually converge to $\mathbf{Y} = (\mathbf{W}\mathbf{W}^\top)^{-1}\mathbf{X}$ and $\mathbf{Z} = \mathbf{W}^\top(\mathbf{W}\mathbf{W}^\top)^{-1}\mathbf{X}$. After the neural activities $(\mathbf{Y}, \mathbf{Z})$ converge, we take a gradient descent step with respect to $\mathbf{W}$:

$$\mathbf{W} \leftarrow \mathbf{W} + \eta\left(\frac{1}{T}\mathbf{Y}\mathbf{Z}^\top - \mathbf{W}\right). \tag{5}$$

5.2 ONLINE ALGORITHM

In the online setting, at each time step t, we first optimize $g(\mathbf{W}, \mathbf{Y}, \mathbf{Z})$ with respect to the neural activities vectors $(\mathbf{y}_t, \mathbf{z}_t)$ by running the gradient ascent-descent neural dynamics:

$$\mathbf{y}_t \leftarrow \mathbf{y}_t + \gamma(\mathbf{x}_t - \mathbf{W}\mathbf{z}_t), \qquad \mathbf{z}_t \leftarrow \mathbf{z}_t + \gamma(\mathbf{W}^\top \mathbf{y}_t - \mathbf{z}_t),$$

which eventually converge to $\mathbf{y}_t = (\mathbf{W}\mathbf{W}^\top)^{-1}\mathbf{x}_t$ and $\mathbf{z}_t = \mathbf{W}^\top(\mathbf{W}\mathbf{W}^\top)^{-1}\mathbf{x}_t$. After convergence of the neural activities, the matrix $\mathbf{W}$ is updated by taking a stochastic gradient descent step:

$$\mathbf{W} \leftarrow \mathbf{W} + \eta(\mathbf{y}_t\mathbf{z}_t^\top - \mathbf{W}).$$

This results in Algorithm 2, which can be implemented in a recurrent neural network with recurrent interactions mediated by interneurons, Figure 1 (right).

Algorithm 2: A whitening network with interneurons

input centered inputs $\{\mathbf{x}_t\}$; parameters γ, η
initialize full rank $\mathbf{W}$
for $t = 1, \ldots, T$ **do**
  repeat
    $\mathbf{y}_t \leftarrow \mathbf{y}_t + \gamma(\mathbf{x}_t - \mathbf{W}\mathbf{z}_t)$
    $\mathbf{z}_t \leftarrow \mathbf{z}_t + \gamma(\mathbf{W}^\top \mathbf{y}_t - \mathbf{z}_t)$
  until convergence
  $\mathbf{W} \leftarrow \mathbf{W} + \eta(\mathbf{y}_t\mathbf{z}_t^\top - \mathbf{W})$
end for

6 ANALYSES OF CONTINUOUS SYNAPTIC DYNAMICS

In this section, we present our main theoretical results on convergence of the continuous synaptic dynamics for $\mathbf{M}$ and $\mathbf{W}$ that correspond to the updates in Algorithms 1 and 2, respectively.

6.1 GRADIENT DESCENT ALGORITHMS

We first show that the offline updates in equations equation 4 and equation 5 are naturally viewed as gradient descent algorithms for minimizing objectives $U(\mathbf{M})$ and $V(\mathbf{W})$, respectively, where

$$U(\mathbf{M}) := \max_{\mathbf{Y} \in \mathbb{R}^{n \times T}} f(\mathbf{M}, \mathbf{Y}) = \operatorname{Tr}\left(\mathbf{M}^{-1}\mathbf{C}_{xx} - \mathbf{M}\right)$$

and $V(\mathbf{W}) = U(\mathbf{W}\mathbf{W}^\top)$ is the overparameterization of $U(\mathbf{M})$:

$$V(\mathbf{W}) := \max_{\mathbf{Y} \in \mathbb{R}^{n \times T}} \min_{\mathbf{Z} \in \mathbb{R}^{k \times T}} g(\mathbf{W}, \mathbf{Y}, \mathbf{Z}) = \operatorname{Tr}\left((\mathbf{W}\mathbf{W}^\top)^{-1}\mathbf{C}_{xx} - \mathbf{W}\mathbf{W}^\top\right).$$

In particular, substituting into the offline update equation 4 with the optimal $\mathbf{Y} = \mathbf{M}^{-1}\mathbf{X}$, we see that the offline update for $\mathbf{M}$ is

$$\mathbf{M} \leftarrow \mathbf{M} + \eta\left(\mathbf{M}^{-1}\mathbf{C}_{xx}\mathbf{M}^{-1} - \mathbf{I}_n\right) = \mathbf{M} - \eta\nabla U(\mathbf{M}).$$

Substituting into the offline update equation 5 with the optimal $\mathbf{Y} = (\mathbf{W}\mathbf{W}^\top)^{-1}\mathbf{X}$ and $\mathbf{Z} = \mathbf{W}^\top\mathbf{Y}$, we see that the offline update for $\mathbf{W}$ is

$$\mathbf{W} \leftarrow \mathbf{W} + \eta\left((\mathbf{W}\mathbf{W}^\top)^{-1}\mathbf{C}_{xx}(\mathbf{W}\mathbf{W}^\top)^{-1}\mathbf{W} - \mathbf{W}\right) = \mathbf{W} - \eta\nabla V(\mathbf{W}).$$

Similarly, Algorithms 1 and 2 are naturally viewed as stochastic gradient descent methods for minimizing $U(\mathbf{M})$ and $V(\mathbf{W})$, respectively.

A common approach for studying stochastic gradient descent algorithms is to analyze the corresponding continuous *gradient flows* (Saxe et al., 2014; 2019; Tarmoun et al., 2021), which are more mathematically tractable and are useful approximations of the average behavior of the stochastic gradient descent dynamics when the step size is small. In the remainder of this section, we analyze and compare the continuous gradient flows associated with Algorithms 1 and 2.

To further facilitate the analysis, we also consider so called 'spectral initializations' that commute with $\mathbf{C}_{xx}$ (Saxe et al., 2014; 2019; Gidel et al., 2019; Tarmoun et al., 2021). Specificially, we say that $\mathbf{A}_0 \in \mathcal{S}^n_{++}$ is a *spectral initialization* if $\mathbf{A}_0 = \mathbf{U}_x\text{diag}(\sigma_1(0), \ldots, \sigma_n(0))\mathbf{U}_x^\top$, where we recall that $\mathbf{U}_x$ is the $n \times n$ orthogonal matrix whose column vectors are the eigenvectors of $\mathbf{C}_{xx}$. To characterize the convergence rates of the gradient flows, we define the Lyapunov function

$$\ell(\mathbf{A}) := \|\mathbf{C}_{xx} - \mathbf{A}^2\|_{\text{Frob}}, \qquad \mathbf{A} \in \mathcal{S}^n_{++}.$$

6.2 Gradient flow analysis of Algorithm 1

The gradient flow of $U(\mathbf{M})$ is given by

$$\frac{d\mathbf{M}(t)}{dt} = -\nabla U(\mathbf{M}(t)) = \mathbf{M}(t)^{-1}\mathbf{C}_{xx}\mathbf{M}(t)^{-1} - \mathbf{I}_n. \tag{6}$$

To analyze solutions of $\mathbf{M}(t)$, we focus on spectral intializations.

Lemma 1. *Suppose $\mathbf{M}_0$ is a spectral initialization. Then the solution $\mathbf{M}(t)$ of the ODE 6 is of the form $\mathbf{M}(t) = \mathbf{U}_x diag(\sigma_1(t), \ldots, \sigma_n(t))\mathbf{U}_x^\top$ where $\sigma_i(t)$, $i = 1, \ldots, n$, are the solutions of the ODE*

$$\frac{d\sigma_i(t)}{dt} = \frac{\lambda_i^2}{\sigma_i(t)^2} - 1, \qquad i = 1, \ldots, n. \tag{7}$$

Consequently,

$$\frac{d}{dt}(\sigma_i(t)^2 - \lambda_i^2)^2 = -\frac{4}{\sigma_i(t)}(\sigma_i(t)^2 - \lambda_i^2)^2, \qquad i = 1, \ldots, n. \tag{8}$$

Proof. Let $\sigma_1(t), \ldots, \sigma_n(t)$ be solutions of the ODE 7 and define $\mathbf{M}(t) := \mathbf{U}_x\text{diag}(\sigma_1(t), \ldots, \sigma_n(t))\mathbf{U}_x^\top$. Then

$$\frac{d\mathbf{M}(t)}{dt} = \mathbf{U}_x\text{diag}\left(\frac{\lambda_1^2}{\sigma_1(t)^2}, \ldots, \frac{\lambda_n^2}{\sigma_n(t)^2}\right)\mathbf{U}_x^\top - \mathbf{I}_n = \mathbf{M}(t)^{-1}\mathbf{C}_{xx}\mathbf{M}(t)^{-1} - \mathbf{I}_n.$$

In particular, we see that $\mathbf{M}(t)$ must be the unique solution of the ODE 6, where uniqueness of solutions follows because the right-hand-side of equation 6 is locally Lipschitz continuous on its domain of definition. Equation 8 then follows from the ODE 7 and the chain rule. □

From Lemma 1, we see that for a spectral initialization with $\sigma_i(0) \leq \lambda_i$, the dynamics of $\sigma_i(t)$ satisfy

$$\frac{d}{dt}(\sigma_i(t)^2 - \lambda_i^2)^2 \leq -\frac{4}{\lambda_i}(\sigma_i(t)^2 - \lambda_i^2)^2.$$

It follows that $\sigma_i(t)^2$ converges to λ_i^2 exponentially with convergence rate greater than $\frac{2}{\sqrt{\lambda_i}}$. On the other hand, suppose $\sigma_i(0) \gg \lambda_i$. From equation 7, we see that while $\sigma_i(t) \gg \lambda_i$, $\sigma_i(t)$ decays at approximately unit rate; that is,

$$\frac{d\sigma_i(t)}{dt} \approx -1. \tag{9}$$

Therefore, the time for $\sigma_i(t)$ to converge to λ_i grows linearly with $\sigma_i(0)$. We make these statements precise in the following proposition.

Proposition 1. *Suppose* $\mathbf{M}_0$ *is a spectral initialization and let* $\mathbf{M}(t)$ *denote the solution of the ODE 6 starting from* $\mathbf{M}_0$*. If* $\sigma_i \leq \lambda_i$ *for all* $i = 1, \ldots, n$*, then for* $\epsilon < \ell(\mathbf{M}_0)$*,*

$$\min\{t \geq 0 : \ell(\mathbf{M}(t)) \leq \epsilon\} \leq \frac{\sqrt{\lambda_{max}}}{2} \log\left(\ell(\mathbf{M}_0)\epsilon^{-1}\right), \tag{10}$$

where $\lambda_{max} := \max_i \lambda_i$*. On the other hand, if* $\sigma_i > \lambda_i$ *for some* $i = 1, \ldots, n$*, then for* $\epsilon < \sigma_i^2 - \lambda_i^2$*,*

$$\min\{t \geq 0 : \ell(\mathbf{M}(t)) \leq \epsilon\} \geq \sigma_i - \sqrt{\lambda_i^2 + \epsilon}. \tag{11}$$

Proof. Suppose $\sigma_i(0) \leq \lambda_i$ for $i = 1, \ldots, n$. By equation 8, $\sigma_i(t) \leq \lambda_i$ for all $t \geq 0$ and so

$$(\sigma_i(t)^2 - \lambda_i^2)^2 \leq (\sigma_i(0)^2 - \lambda_i^2)^2 \exp\left(-\frac{4t}{\sqrt{\lambda_i}}\right), \qquad i = 1, \ldots, n.$$

Therefore,

$$\ell(\mathbf{M}(t))^2 = \sum_{i=1}^{m} (\sigma_i(t)^2 - \lambda_i^2)^2 \leq \ell(\mathbf{M}_0)^2 \exp\left(-\frac{4t}{\sqrt{\lambda_{\max}}}\right).$$

It follows that equation 10 holds. Now suppose $\sigma_i(0) \geq \lambda_i$ for some $i = 1, \ldots, n$. By Lemma 1,

$$\sigma_i(t)^2 - \lambda_i^2 \geq (\sigma_i(0) - t)^2 - \lambda_i^2, \qquad t \in [0, \sigma_i(0) - \lambda_i].$$

It follows that equation 11 holds. □

6.3 Gradient flow analysis of Algorithm 2

The gradient flow of the overparameterized cost $V(\mathbf{W})$ is given by

$$\frac{d\mathbf{W}(t)}{dt} = -\nabla V(\mathbf{W}) = \left(\mathbf{W}(t)\mathbf{W}(t)^\top\right)^{-1} \mathbf{C}_{xx} \left(\mathbf{W}(t)\mathbf{W}(t)^\top\right)^{-1} \mathbf{W}(t) - \mathbf{W}(t). \tag{12}$$

Given a solution $\mathbf{W}(t)$ do the ODE 12, define $\mathbf{A}(t) := \mathbf{W}(t)\mathbf{W}(t)^\top$. Then

$$\frac{d\mathbf{A}(t)}{dt} = \mathbf{A}(t)^{-1}\mathbf{C}_{xx} + \mathbf{C}_{xx}\mathbf{A}(t)^{-1} - 2\mathbf{A}(t). \tag{13}$$

Note that the dynamics of $\mathbf{A}(t)$ do not depend on $k \geq n$. In the following proposition, we show that $\ell(\mathbf{A}(t))$ converges to zero exponentially for any initialization $\mathbf{W}_0$.

Proposition 2. *Let* $\mathbf{W}(t)$ *denote the solution of the ODE 12 starting from any* $\mathbf{W}_0 \in \mathbb{R}^{n \times k}$*. Let* $\epsilon < \ell(\mathbf{W}_0\mathbf{W}_0^\top)$*. Suppose* $\mathbf{W}_0\mathbf{W}_0^\top$ *is a spectral initialization. Then*

$$\min\{t \geq 0 : \ell(\mathbf{W}(t)\mathbf{W}(t)^\top) \leq \epsilon\} = \frac{1}{4} \log\left(\ell(\mathbf{W}_0\mathbf{W}_0^\top)\epsilon^{-1}\right). \tag{14}$$

For general initializations $\mathbf{W}_0\mathbf{W}_0^\top$*,*

$$\min\{t \geq 0 : \ell(\mathbf{W}(t)\mathbf{W}(t)^\top) \leq \epsilon\} \leq \frac{1}{2} \log\left(\ell(\mathbf{W}_0\mathbf{W}_0^\top)\epsilon^{-1}\right). \tag{15}$$

Proof. By the chain rule and equation 13, for all $t \geq 0$,

$$\begin{aligned}
\frac{d\ell(\mathbf{A}(t))^2}{dt} &= 4 \operatorname{Tr}\left(\left(\mathbf{A}(t)^2 - \mathbf{C}_{xx}\right)\mathbf{A}(t)\frac{d\mathbf{A}(t)}{dt}\right) \\
&= -4 \operatorname{Tr}\left(\mathbf{A}(t)^2 - \mathbf{C}_{xx}\right)^2 - 4 \operatorname{Tr}\left((\mathbf{A}(t)^2 - \mathbf{C}_{xx})(\mathbf{A}(t)^2 - \mathbf{A}(t)\mathbf{C}_{xx}\mathbf{A}(t)^{-1})\right) \\
&= -4\ell(\mathbf{A}(t))^2 - 4\|\mathbf{A}(t)^{-1/2}(\mathbf{A}(t)^2 - \mathbf{C}_{xx})\mathbf{A}(t)^{1/2}\|_{\text{Frob}}^2.
\end{aligned}$$

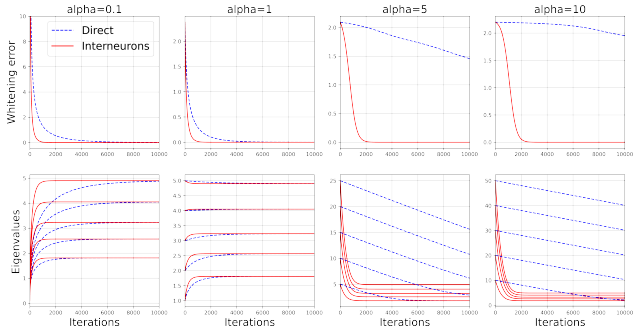
Suppose $\mathbf{W}_0\mathbf{W}_0^\top$ is a spectral initialization. Then $\mathbf{A}(t)$ commutes with $\mathbf{C}_{xx}$ and so

$$\frac{d\ell(\mathbf{A}(t))^2}{dt} = -8\ell(\mathbf{A}(t))^2 \qquad \Rightarrow \qquad \ell(\mathbf{A}(t)) = \ell(\mathbf{A}(0)) \exp(-4t).$$

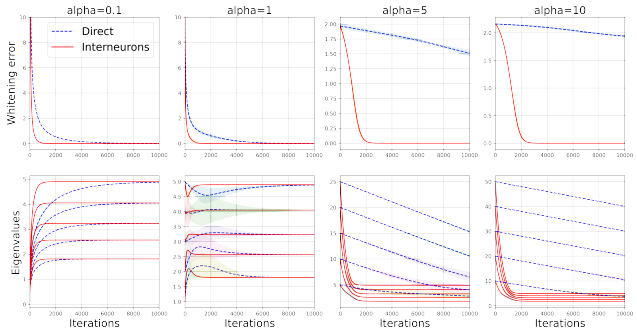
Thus, equation 14 holds. For general initializations,

$$\frac{d\ell(\mathbf{A}(t))^2}{dt} \leq -4\ell(\mathbf{A}(t))^2 \qquad \Rightarrow \qquad \ell(\mathbf{A}(t)) \leq \ell(\mathbf{A}(0)) \exp(-2t).$$

Therefore, inequality equation 15 holds. □



(a) Spectral initializations



(b) Nonspectral initializations

Figure 2: Comparison of whitening error and eigenvalue evolution for the offline algorithms with (a) spectral initializations and (b) nonspectral initializaitons. For the nonspectral initializations, the lines and shaded regions respectively denote the means and 95% confidence intervals over 10 runs.

7 Numerical Experiments

In this section, we numerically test the offline and online algorithms on synthetic datasets.

$$\text{Whitening error}(t) = \|\mathbf{A}_t^{-1}\mathbf{C}_{xx}\mathbf{A}_t^{-1} - \mathbf{I}_n\|_{\text{Frob}},$$

and compare the evolution of the eigenvalues for $\mathbf{A}_t = \mathbf{M}_t, \mathbf{W}_t\mathbf{W}_t^\top$, where the subscript t indicates the value of matrices after the tth iterate. To quantify the convergence time, we define

$$\text{Convergence time} := \min\{t \geq 1 : \text{Whitening error}(t) < 0.1\}.$$

7.1 Offline Algorithms

Let $n = 5$, $k = 10$ and $T = 10^5$. We generate a data matrix $\mathbf{X}$ with i.i.d. entries $x_{t,i}$ chosen uniformly from the interval $(0, 12)$. The eigenvalues of $\mathbf{C}_{xx}$ are $\lambda_1^2 = 24.01$, $\lambda_2^2 = 16.42$, $\lambda_3^2 = 10.45$, $\lambda_4^2 = 6.59$, $\lambda_5^2 = 3.28$. We initialize $\mathbf{W}_0 = \mathbf{Q}\sqrt{\alpha\mathbf{\Sigma}}\mathbf{P}^\top$, where $\alpha > 0$, $\mathbf{\Sigma} = \text{diag}(5, 4, 3, 2, 1)$, $\mathbf{Q}$ is a random $n \times n$ orthogonal matrix and $\mathbf{P}$ is a random $k \times n$ orthogonal matrix with orthonormal column vectors. We set $\mathbf{M}_0 = \mathbf{W}_0\mathbf{W}_0^\top = \alpha\mathbf{Q}\mathbf{\Sigma}^2\mathbf{Q}^\top$. We consider both spectral initializations (i.e., $\mathbf{Q} = \mathbf{U}_x$) and nonspectral initializations. We use step size $\eta = 10^{-3}$.

We first consider spectral (i.e., $\mathbf{Q} = \mathbf{U}_x$) and nonspectral initializations (i.e., $\mathbf{Q}$ is a random orthogonal matrix). The results of running the offline algorithms for $\alpha = 0.1, 1, 5, 10$ are shown in Figure 2. Consistent with Propositions 1 and 2, the network with direct recurrent connections convergences slowly for 'large' α, whereas the network with interneurons converges exponentially for all α. Furthermore, as predicted by equation 9, when the eigenvalues of $\mathbf{M}$ are 'large', i.e., $\sigma_i \gg \lambda_i$, they decay linearly. Interestingly, contrast to the case of spectral initializations, the evolution of the eigenvalues $\sigma_i(t)$ is not always monotone for nonspectral initializations (see the bottom row of Figure 2b).

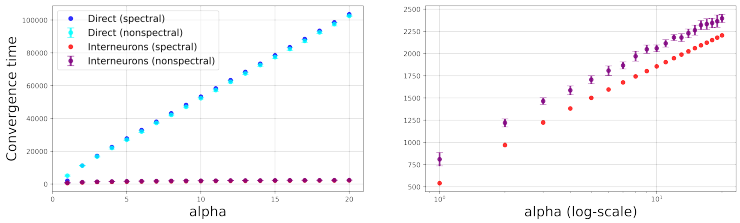


Figure 3: Comparison of convergence times for the offline algorithms with both spectral and nonspectral initializations, as functions of $\alpha = 1, 2, \ldots, 20$. For the nonspectral initializations, the dots and error bars respectively denote the means and 95% confidence intervals over 10 runs.

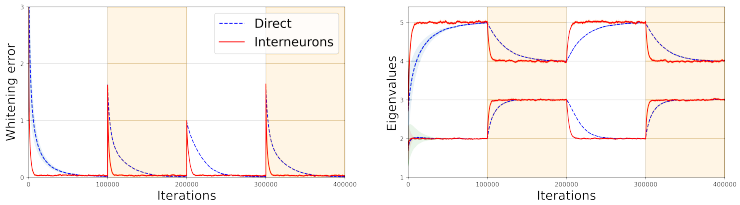


Figure 4: Comparison of whitening error and eigenvalue evolution for the online algorithms on a dataset with switching distributions (white vs. light orange backgrounds). The lines and shaded regions respectively denote the means and 95% confidence intervals over 10 runs.

In Figure 3, we plot the convergence times of the offline algorithms with spectral and nonspectral initializations, for $\alpha = 1, 2, \ldots, 20$. Consistent with our analysis of the gradient flows, the convergence times for network with direct lateral connections (resp. interneurons) grows linearly (resp. logarithmically) with α.

7.2 Online Algorithms

To test our online algorithms, we generate a synthetic dataset with samples from 2 distributions. We first set $n = 2$, $k = 4$ and fixed covariance matrices $\mathbf{C}_A = \mathbf{U}_A \mathrm{diag}(4, 25)\mathbf{U}_A^\top$ and $\mathbf{C}_B = \mathbf{U}_B \mathrm{diag}(9, 16)\mathbf{U}_B^\top$, where $\mathbf{U}_A$, $\mathbf{U}_B$ are random 2×2 rotation matrices. We set $T = 10^5$ and generated a $2 \times 4T$ dataset $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_{4T}]$ with i.i.d. samples $\mathbf{x}_1, \ldots, \mathbf{x}_T, \mathbf{x}_{2T+1}, \ldots, \mathbf{x}_{3T}$ (resp. $\mathbf{x}_{T+1}, \ldots, \mathbf{x}_{2T}, \mathbf{x}_{3T+1}, \ldots, \mathbf{x}_{4T}$) drawn according to a mean zero Gaussian distribution with covariance matrix $\mathbf{C}_A$ (resp. $\mathbf{C}_B$). We evaluated our online algorithms with constant learning rates $\eta = 10^{-4}$ and initializations $\mathbf{M}_0 = \mathbf{W}_0\mathbf{W}_0^\top$, where $\mathbf{W}_0 = \mathbf{Q}\mathrm{diag}(\sigma_1, \sigma_2)\mathbf{P}^\top$, $\mathbf{Q}$ is a random 2×2 rotation matrix, $\mathbf{P}$ is a random 4×2 matrix with orthonormal column vectors and σ_1, σ_2 are independent random variables chosen uniformly from the interval $(1, 1.5)$. The results are shown in Figure 4. Consistent with our theoretical analyses, we see that the network with interneurons adapts to changing distributions faster than the network with direct recurrent connections.

8 Conclusion

We analyzed the gradient flow dynamics of a 2 recurrent neural networks for ZCA whitening — one with direct recurrent connections and one with interneurons. For spectral initializations we can analytically estimate the convergence time for both gradient flows. For nonspectral initializations, we show numerically that the convergence times are close to the spectral initializations with the same initial spectrum. Both our theoretical and numerical results show that the recurrent neural network with interneurons is more robust to initialization than the recurrent neural network with direct recurrent connections.

An interesting question is whether including interneurons in other classes of recurrent neural networks also accelerates the learning dynamics of those networks. We hypothesize that interneurons accelerate learning dynamics when the objective for the network with interneurons can be viewed as an overparameterization of the objective for the recurrent neural network with direct connections.

REFERENCES

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pp. 244–253. PMLR, 2018.

Horace Barlow and Peter Földiák. *Adaptation and Decorrelation in the Cortex*, pp. 54–72. Addison-Wesley Longman Publishing Co., Inc., USA, 1989. ISBN 020118348X.

Horace B Barlow. Possible principles underlying the transformation of sensory messages. *Sensory Communication*, 1(01), 1961.

Anthony J Bell and Terrence J Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Matteo Carandini and David J Heeger. Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13(1):51–62, 2012.

TA Christensen, BR Waldrop, ID Harrow, and JG Hildebrand. Local interneurons and information processing in the olfactory glomeruli of the moth manduca sexta. *Journal of Comparative Physiology A*, 173(4):385–399, 1993.

Yonina C Eldar and Alan V Oppenheim. MMSE whitening and subspace whitening. *IEEE Transactions on Information Theory*, 49(7):1846–1851, 2003.

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pp. 3015–3024. PMLR, 2021.

Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 791–800, 2018.

Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009.

Simon B Laughlin. The role of sensory adaptation in the retina. *Journal of Experimental Biology*, 146(1):39–62, 1989.

Cengiz Pehlevan and Dmitri B Chklovskii. A normative theory of adaptive dimensionality reduction in neural networks. *Advances in Neural Information Processing Systems*, 28, 2015.

Cengiz Pehlevan, Anirvan M Sengupta, and Dmitri B Chklovskii. Why do similarity matching objectives lead to Hebbian/anti-Hebbian networks? *Neural Computation*, 30(1):84–124, 2018.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116 (23):11537–11546, 2019.

Gordon M Shepherd, Chen Wei R, and Charles A Greer. Olfactory bulb. In Gordon M Shepherd (ed.), *The Synaptic Organization of the Brain*, chapter 5, pp. 165–216. Oxford University Press, 2004.

Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(1):1193–1216, 2001.

Piergiorgio Strata and Robin Harvey. Dale's principle. *Brain Research Bulletin*, 50(5):349–350, 1999.

Salma Tarmoun, Guilherme Franca, Benjamin D Haeffele, and Rene Vidal. Understanding the dynamics of gradient flow in overparameterized linear models. In *International Conference on Machine Learning*, pp. 10153–10161. PMLR, 2021.

Barry Wark, Brian Nils Lundstrom, and Adrienne Fairhall. Sensory adaptation. *Current Opinion in Neurobiology*, 17(4):423–429, 2007.

Zachary M Westrick, David J Heeger, and Michael S Landy. Pattern adaptation and normalization reweighting. *Journal of Neuroscience*, 36(38):9805–9816, 2016.

Clarissa J Whitmire and Garrett B Stanley. Rapid sensory adaptation redux: a circuit perspective. *Neuron*, 92(2):298–315, 2016.

A SADDLE POINT PROPERTY

Here we recall the following minimax property for a function that satisfies the saddle point property (Boyd & 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*

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

Then

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