Journal of Machine Learning Research 23 (2022) 1-46 Submitted 8/21; Revised 5/22; Published 5/22

Principal Components Bias in Over-parameterized Linear Models, and its Manifestation in Deep Neural Networks

Guy Hacohen GUY.HACOHEN@MAIL.HUJI.AC.IL
The School of Computer Science and Engineering
Edmond and Lily Safra Center for Brain Sciences
The Hebrew University of Jerusalem
Jerusalem, Israel

Daphna Weinshall DAPHNA@MAIL.HUJI.AC.IL
The School of Computer Science and Engineering
The Hebrew University of Jerusalem
Jerusalem, Israel

Editor: Christoph Lampert

Abstract

Recent work suggests that convolutional neural networks of different architectures learn to classify images in the same order. To understand this phenomenon, we revisit the over-parametrized deep linear network model. Our analysis reveals that, when the hidden layers are wide enough, the convergence rate of this model's parameters is exponentially faster along the directions of the larger principal components of the data, at a rate governed by the corresponding singular values. We term this convergence pattern the *Principal Components bias (PC-bias)*. Empirically, we show how the *PC-bias* streamlines the order of learning of both linear and non-linear networks, more prominently at earlier stages of learning. We then compare our results to the simplicity bias, showing that both biases can be seen independently, and affect the order of learning in different ways. Finally, we discuss how the *PC-bias* may explain some benefits of early stopping and its connection to PCA, and why deep networks converge more slowly with random labels.

Keywords: Deep linear networks, Learning dynamics, PC-bias, Simplicity bias, Learning order.

1. Introduction

The dynamics of learning in deep neural networks is an intriguing subject, not yet sufficiently understood. Recent empirical studies of learning dynamics (Hacohen et al., 2020; Pliushch et al., 2021; Choshen et al., 2022) showed that neural networks memorize the training examples of natural datasets in a consistent order, and further impose a consistent order on the successful recognition of unseen examples. Below we call this effect *Learning Order Constancy* (LOC). Currently, the characteristics of visual data, which may explain this phenomenon, remain unclear. Surprisingly, this universal order persists despite the variability introduced into the training of different models and architectures.

To understand this phenomenon, we start by analyzing the deep linear network model (Saxe et al., 2014, 2019), defined by the concatenation of linear operators in a multi-class classification setting. Accordingly, in Section 3 we prove that the convergence of the weights of deep linear networks is governed by the eigendecomposition of the raw data, which is blind to the labels of the data, in a phenomenon we term *PC-bias*. These results are valid when the hidden layers are wide enough, a generalization of the known behavior of the single-layer convex linear model. In Section 4, we empirically show that this pattern of convergence is indeed observed in deep linear networks of rather a moderate width, validating the plausibility of our assumptions. We continue by showing that the *LOC-effect* in deep linear networks is determined solely by their *PC-bias*. We prove a similar (weaker) result for the non-linear two-layer ReLU model trained on a binary classification problem, introduced by Allen-Zhu et al. (2019).

In Section 5, we extend the study empirically to non-linear networks and investigate the relationship between the *PC-bias* and the *LOC-effect* in general deep networks. We first show that the order by which examples are learned by linear networks is highly correlated with the order induced by prevalent deep CNN models. We then show directly that the learning order of non-linear CNN models is affected by the principal components decomposition of the data. Moreover, the *LOC-effect* diminishes when the data is whitened, indicating a tight connection between the *PC-bias* and the *LOC-effect*.

Our results are reminiscent of another phenomenon, termed *Spectral bias* (see Section 2.2), which associates the learning dynamics of neural networks with the Fourier decomposition of functions in the hypothesis space. In Section 5.3 we investigate the relation between the *PC-bias*, *spectral bias*, and the *LOC-effect*. We find that the *LOC-effect* is very robust: (i) when we neutralize the *spectral bias* by using low complexity models such as deep linear networks, the effect is still observed; (ii) when we neutralize the *PC-bias* by using whitened data, the *LOC-effect* persists. We hypothesize that at the beginning of learning, the learning dynamics of neural models is governed by the eigendecomposition of the raw data. As learning proceeds, control of the dynamics slowly shifts to other factors.

The PC-bias has implications beyond the *LOC-effect*, as expanded in Section 6:

(i) Early stopping. It is often observed that when training deep networks with real data, the highest generalization accuracy is obtained before convergence. Consequently, early stopping is often prescribed to improve generalization. Following the commonly used assumption that in natural images the lowest principal components correspond to noise (Torralba and Oliva, 2003), our results predict the benefits of early stopping, and relate it to PCA. In Section 6 we investigate the relevance of this conclusion to real non-linear networks (see Basri et al., 2019; Li et al., 2020, for complementary accounts).

(ii) Slower convergence with random labels. Zhang et al. (2017) showed that neural networks can learn any label assignment. However, training with random label assignments is known to converge slower as compared to training with the original labels (Krueger et al., 2017). We report a similar phenomenon when training deep linear networks. Our analysis shows that when the principal eigenvectors are correlated with class identity, as is often the case in natural images, the loss decreases faster when given true label assignments as against random label assignments. In Section 6 we investigate this hypothesis empirically in linear and non-linear networks.

(iii) Weight initialization. Different weight initialization schemes have been proposed to stabilize the learning and minimize the hazard of "exploding gradients" (for example, Glorot and Bengio, 2010; He et al., 2015). Our analysis (see Appendix A) identifies a related variant, which eliminates the hazard when all the hidden layers are roughly of equal width. In the deep linear model, it can be proven that the proposed normalization variant in a sense minimizes repeated gradient amplification.

2. Scientific Background and Previous Work

Below we review related work, and discuss the relation of our work to this prior art.

2.1 Deep Over-Parametrized Neural Networks and Other Simple Models

A large body of work concerns the analysis of deep over-parameterized linear networks. While not a universal approximator, this model is nevertheless trained by minimizing a non-convex objective function with a multitude of equally valued global minima. The investigation of such networks is often employed to shed light on the learning dynamics when complex geometric landscapes are explored by GD (Fukumizu, 1998; Arora et al., 2018; Wu et al., 2019a; Du and Hu, 2019; Du et al., 2019; Hu et al., 2020b; Yun et al., 2021). Note that while such networks provably achieve a simple low-rank solution (Ji and Telgarsky, 2019; Du et al., 2018) when given linearly separable data, in our work this strict assumption is not needed.

Early on Baldi and Hornik (1989) characterized the optimization landscape of the over-parameterized linear model and its relation to PCA. More recent work suggests that with sufficient over-parameterization, this landscape is well behaved (Kawaguchi, 2016; Zhou and Liang, 2018) and all its local minima are global (Laurent and von Brecht, 2017). Deep linear networks are also used to study biases induced by architecture or optimization (Ji and Telgarsky, 2019; Wu et al., 2019b).

Several related studies investigated the dynamics of learning in over-parameterized models by way of spectral analysis. However, while we employ the eigenvectors of the data covariance matrix[1] $\Sigma_{XX} = XX^\top$, most of the behavior identified in these studies is driven by the eigenvectors of the cross-covariance matrix $\Sigma_{YX} = YX^\top$. This difference is crucial, as Σ_{XX} is blind to the class labels, unlike Σ_{YX}. In addition, quite a few of the studies reviewed below assumed a shallow 2-layer network, while our analysis accommodates over-parametrized networks of any depth, further showing that convergence depends on the eigenvectors of Σ_{XX} rather than Σ_{YX} without any additional assumption.

More specifically, Saxe et al. (2019) analyzed the evolution of learning dynamics in a shallow two-layered linear model, showing that convergence is guided by the eigenvectors of Σ_{YX}. This analysis assumed $\Sigma_{XX} = I_d$, thus obscuring the dependence of convergence on the raw data eigenvectors of Σ_{XX}, as all the eigenvalues are now identical by assumption. Likewise, Arora et al. (2018) also assumed that $\Sigma_{XX} = I_d$ while investigating continuous gradients of the deep linear model and allowing for a more general loss. Although there is some superficial resemblance between their pre-conditioning and our *gradient scale matrices* as defined below, they are defined quite differently and have different properties.

1. X and Y denote the matrices whose columns are the data points and one-hot label vectors respectively, see notations in Section 3.

Similarly to Saxe et al. (2019), Gidel et al. (2019) also investigated a shallow 2-layers linear model, but no longer assumed that $\Sigma_{XX} = I_d$. Instead, they assumed that Σ_{XX} and Σ_{YX} can be jointly decomposed, having the same eigenvectors. Under these conditions, they were able to show that convergence is guided by the eigenvectors of Σ_{YX}, which correspond to the eigenvectors of Σ_{XX} by the joint decomposition assumption. However, Σ_{XX} is now linked to the class labels by assumption, which is not the case in most datasets. Recently, Nguyen (2021) showed that the learning dynamics of two-layered non-linear auto-encoders converge faster along the eigenvectors of Σ_{XX}. While their results resemble ours, unlike us they focus on auto-encoders and assume that the data is sampled from a Gaussian distribution.

Relations between shallow and deep over-parameterized linear networks are often studied through the lens of gradient flow (Arora et al., 2018, 2019a). Arora et al. (2019a) derived an equation for the gradient-flow of over-parameterized deep linear networks, which was later expanded to shallow ReLU networks by Williams et al. (2019). Bah et al. (2019) showed that this gradient flow can be re-interpreted as a Riemannian gradient flow of matrices of some fixed ranked, hinting at a strong connection between the optimization process of shallow and deep linear networks, in accordance with our results.

Another line of related work was pioneered by Jacot et al. (2018), who showed how neural networks can be analyzed using kernel theory and introduced the Neural Tangent Kernel (NTK). In this framework, it was shown that convergence is fastest along the largest **kernel's** principal components. In the one-layer linear model, this result implies that convergence depends on the eigenvectors of Σ_{XX}, but to the best of our knowledge, it was not extended to the over-parameterized deep model analyzed here. While similar results to ours may be achieved in the future by using kernel theory, here we provide direct proof, thus bypassing possible limitations of the kernel theory (Chizat et al., 2019; Yehudai and Shamir, 2019). This direct proof further enables us to examine the behavior outside the limit of infinite width, and examine our assumptions using empirical methods.

2.2 Related Research Paradigms

Diverse empirical observations seem to support the hypothesis that neural networks start by learning a simple model, which then gains complexity as learning proceeds (Gunasekar et al., 2018; Soudry et al., 2018; Hu et al., 2020a; Kalimeris et al., 2019; Gissin et al., 2020; Heckel and Soltanolkotabi, 2020; Ulyanov et al., 2018; Pérez et al., 2019; Cao et al., 2021; Jin and Montúfar, 2020). This phenomenon is sometimes called *simplicity bias* (Dingle et al., 2018; Shah et al., 2020).

In a related line of work, Rahaman et al. (2019) empirically demonstrated that the complexity of classifiers learned by ReLU networks increases with time. Basri et al. (2019, 2020) showed theoretically, by way of analyzing elementary neural network models, that these models first fit the data with low-frequency functions, then gradually acquire higher frequencies to improve the fit. Nevertheless, the *spectral bias* and *PC-bias* are inherently different. Indeed, the eigendecomposition of raw images is closely related to the Fourier analysis of images as long as the statistical properties of images are (approximately) translation-invariant (Simoncelli and Olshausen, 2001; Torralba and Oliva, 2003). Still, the *PC-bias* is guided by spectral properties of the raw data and is therefore blind to class labels. In contrast, the *spectral bias*, as well as the related *frequency bias* that has been shown to characterize NTK

models (Basri et al., 2020), are all guided by spectral properties of the learned hypothesis, which crucially depend on label assignment.

Other related research paradigms include curriculum learning (Bengio et al., 2009; Hacohen and Weinshall, 2019; Weinshall and Amir, 2020), self-paced learning (Kumar et al., 2010; Tullis and Benjamin, 2011), hard data mining (Fu and Menzies, 2017), and active learning (Krogh and Vedelsby, 1994; Hacohen et al., 2022; Yehuda et al., 2022). In these frameworks, the research focus is on the order in which data is presented to the learner. Differently, our focus is on characterizing the order by which data is learned without additional guidance to this effect.

3. Theoretical Analysis

In this section, we analyze the over-parameterized deep linear networks model, in which the principal component bias fully governs the learning dynamics.

3.1 Notations and Definitions

Let $\mathbb{X} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ denote the training data, where $\boldsymbol{x}_i \in \mathbb{R}^q$ denotes the i-th data point and one-hot vector $\boldsymbol{y}_i \in \{0,1\}^K$ denotes its corresponding label. Let $\frac{1}{n_i}\boldsymbol{m}_i$ denote the centroid (mean) of class i with n_i points $\frac{1}{n_i}\boldsymbol{m}_i = \frac{1}{n_i}\sum_{j=1}^n \boldsymbol{x}_j \mathbb{1}_{[y_j=i]}$, and $M = [\boldsymbol{m}_1 \dots \boldsymbol{m}_K]^\top$ the matrix whose rows are vectors $\boldsymbol{m}_i^\top$. Finally, let X and Y denote the matrices whose i^{th} column is $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ respectively. $\Sigma_{XX} = XX^\top$ and $\Sigma_{YX} = YX^\top$ denote the covariance matrix of X and cross-covariance of X and Y respectively. We note that Σ_{XX} captures the structure of the data irrespective of class identity, and that $\Sigma_{YX} = M$.

3.1.1 Definitions

Definition 1 (Principal coordinate system) *The coordinate system obtained by rotating the data in $\mathbb{R}^q$ by an orthonormal matrix $U^\top$, where $SVD(\Sigma_{XX}) = UDU^\top$.*

In this system $\Sigma_{XX} = D$, a diagonal matrix whose elements are the singular values of $XX^\top$, arranged in decreasing order $d_1 \geq d_2 \geq \dots \geq d_q \geq 0$.

We analyze a deep linear network with L layers, whose parameters are computed by minimizing (using gradient descent) the following loss $L(X,Y)$

$$L(X,Y) = \frac{1}{2}\|\boldsymbol{W}X - Y\|_F^2, \qquad \boldsymbol{W} \coloneqq \prod_{l=L}^{1} W_l, \quad W_l \in \mathbb{R}^{m_l \times m_{l-1}}. \tag{1}$$

Above m_l denotes the number of neurons in layer l, where $m_0 = q$ and $m_L = K$.

Definition 2 (Compact representation) *Given a deep linear network $\boldsymbol{y} = W_L \cdot \dots \cdot W_1 \boldsymbol{x}$, its compact representation is denoted $\boldsymbol{W} \in \mathbb{R}^{K \times q}$ where $\boldsymbol{W} = \prod_{l=L}^{1} W_l$.*

Definition 3 (Gradient matrix) *For a deep linear network whose compact representation is $\boldsymbol{W}$, the gradient matrix of $L(X,Y)$ with respect to W is $G_r = \boldsymbol{W}\Sigma_{XX} - \Sigma_{YX}$.*

In the principal coordinate system, $G_r = \boldsymbol{W}D - M$.

Definition 4 (STD initialization) *Given a sequence of random matrices* $\{W_l\}_{l=1}^{L}$ *where* $W_l \in \mathbb{R}^{m_l \times m_{l-1}}$*, and all the elements in* W_l *are chosen i.i.d from a distribution with mean 0 and variance* σ_l^2*, define*

$$\sigma_l^2 = \frac{2}{m_{l-1}+m_l} \ \forall l \in [2 \dots L-1], \qquad \sigma_1^2 = \frac{1}{m_1}, \qquad \sigma_L^2 = \frac{1}{m_{L-1}}.$$

Def. 4 is a variant of the Glorot initialization (Glorot and Bengio, 2010). For more analytical and empirical results on *STD initialization*, refer to App. B.2 and App. C.1 respectively.

3.1.2 Asymptotic Analysis

Our analysis involves two asymptotic quantities: the learning rate (or gradient step size) μ, and the network's width $\mathbb{m}$ ($\mathbb{m}$ denotes the width of the smallest hidden layer). Accordingly, our analysis neglects terms of magnitude $O(\mu^2)$ and $O(\frac{1}{\mathbb{m}})$, and employs convergence in probability when $\mathbb{m} \to \infty$. In Fig. 1, we show the plausibility of these assumptions, as the predicted dynamics is seen to hold even when the width $\mathbb{m}$ is smaller than the input size, and the step size μ permits convergence in a relatively short time.

To simplify the presentation and to allow for modular analysis, we use the following conventions: (i) $O(\mu^2)$ and $O(\frac{1}{\mathbb{m}})$ denote terms which are upper bounded by $c\mu^2$ when μ is small and $c\frac{1}{\mathbb{m}}$ when $\mathbb{m}$ is large respectively, where $c \in \mathbb{R}$ denotes a fixed constant that does *not* depend on neither μ nor $\mathbb{m}$. (ii) Depending on the context, if matrices are involved then $O(\mu^2)$ and $O(\frac{1}{\mathbb{m}})$ denote full rank matrices with element-wise asymptotic quantities defined as above. (iii) The notation $\xrightarrow{p}$ denotes convergence in probability when $\mathbb{m} \to \infty$, where the probabilistic lower bound on width $\mathbb{m}$ does *not* depend on μ. When matrices are involved, convergence in probability occurs element-wise.

The treatment of the two asymptotic quantities μ and $\mathbb{m}$ is consolidated in the formal analysis of the network's dynamics, as detailed in the proof of Thm 15 in Appendix A.2, and the proofs of Thms 7-8 in Appendix B.2. Importantly, the analysis is valid at each time step t, where the only thing that changes with time is the magnitude of the constants governing the asymptotic behavior.

3.2 The Dynamics of Deep Over-Parametrized Linear Networks

Extending the analysis described in Du and Hu (2019), we derive here the temporal dynamics of $\boldsymbol{W}$, denoted $\boldsymbol{W}^{(t)}$, as it changes with each GD step.

Proposition 5 *Let* $G_r^{(t)}$ *(Def. 3) denote the gradient matrix at time* t*. Let* $B_l^{(t)}$ *and* $A_l^{(t)}$ *denote the gradient scale matrices, which are defined as follows*

$$B_l^{(t)} \coloneqq \Big(\prod_{j=l}^{1} W_j^{(t)}\Big)^\top \Big(\prod_{j=l}^{1} W_j^{(t)}\Big) \in \mathbb{R}^{q \times q}, \qquad A_l^{(t)} \coloneqq \Big(\prod_{j=L}^{l+1} W_j^{(t)}\Big)\Big(\prod_{j=L}^{l+1} W_j^{(t)}\Big)^\top \in \mathbb{R}^{K \times K}. \tag{2}$$

The compact representation $\boldsymbol{W}^{(t)}$ *obeys the following dynamics:*

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \mu \sum_{l=1}^{L} A_l^{(t)} \cdot G_r^{(t)} \cdot B_{l-1}^{(t)} + O(\mu^2). \tag{3}$$

The proof can be found in Appendix B.1. Note that $B_0^{(t)} = A_L^{(t)} = I$, $B_L^{(t)} = \boldsymbol{W}^{(t)^\top}\boldsymbol{W}^{(t)}$, and $A_0^{(t)} = \boldsymbol{W}^{(t)}\boldsymbol{W}^{(t)^\top}$.

Gradient scale matrices. When the number of hidden layers is 0 ($L = 1$), both gradient scale matrices reduce to the identity matrix and the dynamics in (3) is reduced to the known result that $\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \mu G_r^{(t)}$. Recall, however, that our focus is the over-parameterized linear model with $L > 1$, in which the loss is not convex. Since the difference between the convex linear model and the over-parametrized deep model boils down to these matrices, our convergence analysis henceforth focuses on the dynamics of the *gradient scale matrices*. In accordance, we analyze the evolution of the *gradient scale matrices* as learning proceeds.

Theorem 6 *Let $\boldsymbol{W}$ denote the compact representation of a deep linear network, where $\mathbb{m} = \min(m_1, ..., m_{L-1})$ denotes the size of its smallest hidden layer and $\mathbb{m} \geq \{m_0, m_L\}$. At each layer l, assume weight initialization $W_l^{(0)}$ obtained by sampling from a distribution with mean 0 and variance σ_l^2, normalized as specified in Def. 4. Let $\left(B_l^{(t)}(\mathbb{m})\right)_{\mathbb{m}=1}^{\infty}$ and $\left(A_l^{(t)}(\mathbb{m})\right)_{\mathbb{m}=1}^{\infty}$ denote two sequences of gradient scale matrices as defined in (2), where the $\mathbb{m}^{th}$ element of each series corresponds to a network whose smallest hidden layer has $\mathbb{m}$ neurons. Let $\xrightarrow{p}$ denote element-wise convergence in probability as $\mathbb{m} \to \infty$. Then $\forall t, l$:*

$$\begin{aligned} &B_l^{(t)}(\mathbb{m}) \xrightarrow{p} I + O(\mu^2) \quad \forall l \in [1 \dots L-1], \qquad B_L^{(t)}(\mathbb{m}) \xrightarrow{p} \frac{K}{\mathbb{m}} I + O(\mu^2) \xrightarrow{p} O(\mu^2), \\ &\text{var}[B_l^{(t)}(\mathbb{m})] = O(\mu^2) + O\left(\frac{1}{\mathbb{m}}\right) \qquad \forall l, \end{aligned} \tag{4}$$

and

$$\begin{aligned} &A_l^{(t)}(\mathbb{m}) \xrightarrow{p} I + O(\mu^2) \quad \forall l \in [1 \dots L-1], \qquad A_0^{(t)}(\mathbb{m}) \xrightarrow{p} \frac{q}{\mathbb{m}} I + O(\mu^2) \xrightarrow{p} O(\mu^2), \\ &\text{var}[A_l^{(t)}(\mathbb{m})] = O(\mu^2) + O\left(\frac{1}{\mathbb{m}}\right) \qquad \forall l. \end{aligned} \tag{5}$$

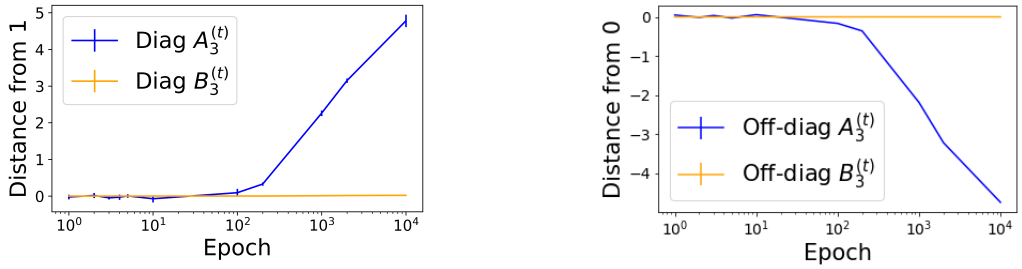
Proof sketch. The proof proceeds by induction on time t. To begin with, we recall that all the weight matrices $\{W_l^{(0)}\}_{l=1}^L$ are initialized by sampling from a distribution with mean 0 and variance $\sigma_l^2 = O(\frac{1}{\mathbb{m}})$. In Appendix A.1 we prove some statistical properties of such matrices, and their corresponding *gradient scale matrices* $B_l^{(0)}$ and $A_l^{(0)}$. This analysis shows that (4) and (5) are true at $t = 0$. To prove the induction step, we resort to results stated in Appendix A.2, where we analyze random matrices that are defined similarly to the gradient scale matrices, and whose dynamics is consistent with the update rule defined in (3). The main result can be used to show directly that if the theorem's assertion holds for $B_l^{(t)}$, it also holds for $B_l^{(t+1)}$. A complete proof and details can be found in Appendix B.2.

Relation to the one-layer linear model. From Thm 6, if the hidden layers are sufficiently wide (as measured by $\mathbb{m}$), we can assume that $B_l^{(t)}(\mathbb{m}) \approx I$ and $A_l^{(t)}(\mathbb{m}) \approx I$ $\forall l$. In this case, the dynamics of the over-parameterized model is identical to the dynamics of the convex linear model, $\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \mu G_r^{(t)}$. This is shown more formally below in Thm 7.

Note about convergence, and convergence rate. In Thm 6 we prove two asymptotic results: $B_l^{(t)}(\mathbb{m}) \xrightarrow{p} I$ and $A_l^{(t)}(\mathbb{m}) \xrightarrow{p} I$ up to small deviations of magnitude $O(\mu^2)$. This is true $\forall l \in [1 \dots L-1]$ and $\forall t$, irrespective of the convergence of the network. The only thing that changes with t is the magnitude of the constants governing this asymptotic behavior, hidden in the notations $O(\frac{1}{\mathbb{m}})$ and $O(\mu^2)$, whose magnitude increases with time. For a fixed network, there will come a time when the asymptotic analysis is longer applicable, and this may happen before or after the network has converged to its final solution.

Importantly, these constants are significantly different in the two sequences. Specifically, in (4) and (5) we see that the convergence of $B_L^{(t)}(\mathbb{m})$ is governed by $O\left(\frac{K}{\mathbb{m}}\right)$, while the convergence of $A_l^{(0)}(\mathbb{m})$ is governed by $O\left(\frac{q}{\mathbb{m}}\right)$. Typically $q \gg K$, as q denotes the dimension of the data space which can be fairly large, while K denotes the typically small number of classes. Considered in the context of the proof of Thm 6 by induction, we note that these constants are amplified in each iteration t. As a result, when $\mathbb{m}$ is large but fixed, $A_l^{(t)}(\mathbb{m})$ is expected to deviate from I sooner than $B_l^{(t)}(\mathbb{m})$ as the GD steps are reiterated.

Empirical validation. To understand the practical significance of the observations stated above, about the difference in convergence rate between the sequences $B_l^{(t)}(\mathbb{m})$ and $A_l^{(t)}(\mathbb{m})$, we resort to simulations whose results are shown in Fig. 1. These empirical results, recounting linear networks with 4 hidden layers of width 1024, clearly show that during a significant part of the training both *gradient scale matrices* remain approximately I. The difference between the convergence rate of $B_l^{(t)}$ and $A_l^{(t)}$ is seen later on, when $A_l^{(t)}$ starts to deviate from I shortly before the networks have reached their maximal test accuracy, while $B_l^{(t)}$ remains essentially the same throughout.



(a) Diagonal elements, layer 3

(b) Off-diagonal elements, layer 3

Figure 1: The dynamics of $B_3^{(t)}$ and $A_3^{(t)}$ when training 10 5-layered linear networks on the small mammals dataset (see App. D.4). (a) The empirical L_2-distance of the diagonal elements of $B_3^{(t)}$ and $A_3^{(t)}$ from[2] 1. (b) The empirical L_2-distance of the off-diagonal elements of $B_3^{(t)}$ and $A_3^{(t)}$ from 0. The networks reach maximal test accuracy in epoch $t = 100$, before the divergence of $A_3^{(t)}$. In these experiments $B_3^{(t)}$ never reaches the point of divergence. These result are typical of all the layers in the networks, see Fig. 13 in Appendix C.2.

2. More precisely, the distance is computed respectively to their analytical values of $\alpha_3^{(t)}$ and $\beta_3^{(t)}$, computed as $diag\left(A_3^{(t)} - \alpha_3^{(t)} I\right)$ and $diag\left(B_3^{(t)} - \beta_3^{(t)} I\right)$, where $\alpha_i^{(t)}, \beta_i^{(t)}$ are defined in Thm 10, §A.1.

3.3 Weight Evolution

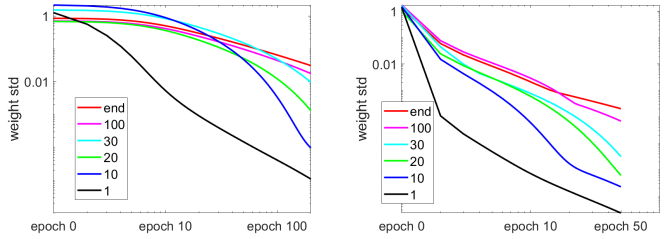
Next, we investigate the implications of Thm 6 regarding the evolution of the compact representation $\boldsymbol{W}$. In Thm 7, we show that at the beginning of training, when the assumption that both $B_l^{(t)} \approx I$ and $A_l^{(t)} \approx I$ is applicable, this evolution resembles the single-layer linear model and is governed by the eigendecomposition of the data. In Thm 8, we show that later on, when only $B_l^{(t)} \approx I$ is applicable and the evolution no longer resembles the single-layer linear model, it is still governed by the eigendecomposition of the data.

More specifically, let $\boldsymbol{w}_j^{opt}$ denote the j^{th} column of the optimal solution of (1), and $\boldsymbol{w}_j^{(0)}$ the j^{th} column of the initial weight matrix. Thm 7 states that if the hidden layers of the network are all wider than a certain fixed width $\hat{\mathrm{m}}$ and if the learning rate is slower than $\hat{\mu}$, then at the beginning of learning (time $t \in [0 \dots \hat{t}]$), the j^{th} column of the compact representation $\boldsymbol{W}^{(t)}$ is roughly the following linear combination of $\boldsymbol{w}_j^{(0)}$ and $\boldsymbol{w}_j^{opt}$:

$$\boldsymbol{w}_j^{(t+1)} \approx \lambda_j^t \boldsymbol{w}_j^{(0)} + [1 - \lambda_j^t] \boldsymbol{w}_j^{opt}, \qquad \lambda_j = 1 - \mu d_j L. \tag{6}$$

with high probability (at least $(1-\delta)$) and low error (at most ε). Constant λ_j depends on the j^{th} singular value of the data d_j and the number of layers L.

Result (6) is reminiscent of the well-understood dynamics of training the convex one-layer linear model. It is composed of two additive terms, revealing two parallel and separate processes: (i) The dependence on random initialization tends to 0 exponentially fast as a function of time t, when λ_j^t tends to 0. (ii) The optimal solution is reached exponentially fast as a function of time t, when $1 - \lambda_j^t$ tends to 1. In either case, convergence is fastest for the largest singular eigenvalue, or the first column of $\boldsymbol{W}$, and slowest for the smallest singular value. This behavior is visualized in Fig. 2a, which shows that the decrease in the variance of weight estimation between networks is indeed faster for larger singular values.



(a) 2-layer linear network (b) 2-layer ReLU network

Figure 2: Empirical evaluation of the dependence of convergence rate on the eigendecomposition in a *binary* classification problem. Here the classifier is a vector, denoted $\boldsymbol{w}$. Each line corresponds to a specific principal eigenvector, plotting in log-log scale the std of element w_i ($i = 1, 10, 20, 30, 100, 3072$, see legend) over 10 independently trained networks.

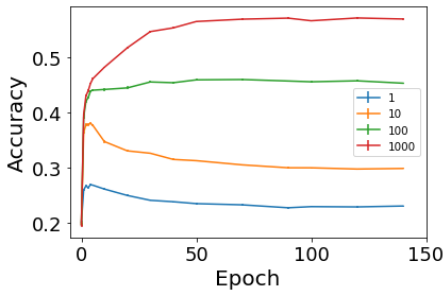


Figure 3: Mean accuracy of 10 st-VGG networks evaluated on test data projected using PCA to $\{1, 10, 100, 1000\}$ dimensions (see text for more details).

Formally, the theorem can be stated as follows (see proof in Appendix B.2):

Theorem 7 *Let $\boldsymbol{w}_j^{(t)}$ denote the j^{th} column of the compact representation matrix $\boldsymbol{W}^{(t)}$, and $\boldsymbol{w}_j^{opt}$ the j^{th} column of the optimal solution of (1). Assume that the data is rotated to its principal coordinate system, and let d_j denote the j^{th} singular value of the data. Then there exists $\hat{t}$ such that $\forall \delta, \varepsilon$ and $\forall t \leq \hat{t}$, $\exists \hat{\mathbb{m}}, \hat{\mu}$ such that $\forall \mu < \hat{\mu}, \mathbb{m} \geq \hat{\mathbb{m}}$*

$$Prob\left(\left\|\boldsymbol{w}_j^{(t+1)} - \left[\lambda_j^t \boldsymbol{w}_j^{(0)} + [1-\lambda_j^t]\boldsymbol{w}_j^{opt}\right]\right\| < \varepsilon\right) > (1-\delta), \qquad \lambda_j = 1 - \mu d_j L.$$

Proof sketch. We first prove by induction on time t that $\forall t$, $\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \mu L G_r^{(t)}$ with high probability and small error. We then shift to the principal coordinate system, and note that this is true separately for each column of matrix $\boldsymbol{W}$. The evolution of each column is then shown to be a telescopic series, whose solution is $\boldsymbol{w}_j^{(t)} = \lambda_j^t \boldsymbol{w}_j^{(0)} + [1-\lambda_j^t]\boldsymbol{w}_j^{opt}$ for $\lambda_j = 1 - \mu L d_j$, with high probability and small error.

Next we state Thm 8, which remains valid for longer (see footnote 3 below) as it does not depend on the convergence of $A_l^{(t)}$. This is the case discussed in Thm 6 and illustrated in Fig. 1, when $B_l^{(t)} \approx I$ remains approximately true while $A_l^{(t)} \neq I$. More specifically, let $A^{(t)} = \sum_{l=1}^{L} A_l^{(t)}$. Thm 8 asserts that if the hidden layers of the network are all wider than $\breve{\mathbb{m}}$ and if the learning rate is slower than $\breve{\mu}$, then at time $t \in [0 \dots \breve{t}]$,[3] the j^{th} column of the compact representation $\boldsymbol{W}^{(t)}$ is approximately the following

$$\boldsymbol{w}_j^{(t+1)} \approx \prod_{t'=1}^{t} (I - \mu d_j A^{(t')}) \boldsymbol{w}_j^{(0)} + \mu d_j \left[\sum_{t'=1}^{t} \prod_{t''=t'+1}^{t} (I - \mu d_j A^{(t'')}) A^{(t')}\right] \boldsymbol{w}_j^{opt},$$

with high probability (at least $(1-\delta)$) and low error (at most ε). Note that the j^{th} column of the compact representation $\boldsymbol{W}^{(t)}$ is still a linear combination of $\boldsymbol{w}_j^{(0)}$ and $\boldsymbol{w}_j^{opt}$, but now the coefficients are much more complex and depend on $A_l^{(t)}$.

Formally, the theorem can be stated as follows (see proof in Appendix B.2):

Theorem 8 *Let $\boldsymbol{w}_j^{(t)}$ denote the j^{th} column of the compact representation matrix $\boldsymbol{W}^{(t)}$, and $\boldsymbol{w}_j^{opt}$ the j^{th} column of the optimal solution of (1). Assume that the data is rotated to its principal coordinate system, and let d_j denote the j^{th} singular value of the data. Then there exists $\breve{t}$ such that $\forall \delta, \varepsilon$ and $\forall t \leq \breve{t}$, $\exists \breve{\mathbb{m}}, \breve{\mu}$ such that $\forall \mu < \breve{\mu}, \mathbb{m} \geq \breve{\mathbb{m}}$*

$$Prob\Bigg(\Bigg\|\boldsymbol{w}_j^{(t+1)} - \Bigg[\prod_{t'=1}^{t} (I - \mu d_j A^{(t')}) \boldsymbol{w}_j^{(0)} + \mu d_j \Big(\sum_{t'=1}^{t} \prod_{t''=t'+1}^{t} (I - \mu d_j A^{(t'')}) A^{(t')}\Big) \boldsymbol{w}_j^{opt}\Bigg]\Bigg\| < \varepsilon\Bigg) > (1-\delta).$$

Although the dynamics now depend on matrices $A_l^{(t)}$ as well, it is still the case that the convergence of each column is governed by its singular value d_j. This suggests that while the *PC-bias* is more pronounced at earlier stages of learning, its effect persists throughout.

3. Presumably, in comparison to Thm 7 and if we fix the network's width $\breve{m} = \hat{m}$, then $\breve{t} \gg \hat{t}$.

3.4 Adding ReLU Activation

We extend the results above to a relatively simple non-linear model suggested and analyzed by Allen-Zhu et al. (2019); Arora et al. (2019b); Basri et al. (2019), here trained on a classification task rather than a regression task. Specifically, it is a two-layer model with ReLU activation, where only the weights of the first layer are being learned. Similarly to (1), the optimization problem is defined as

$$W^* = \underset{W}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} \|f(\boldsymbol{x}_i) - \boldsymbol{y}_i\|^2, \qquad f(\boldsymbol{x}_i) = \boldsymbol{a}^\intercal \cdot \sigma(W\boldsymbol{x}_i), \quad \boldsymbol{a} \in \mathbb{R}^{\mathtt{m}},\ W \in \mathbb{R}^{\mathtt{m} \times d},$$

where $\mathtt{m}$ denotes the number of neurons in the hidden layer and $\boldsymbol{a}$ a fixed vector. We consider a binary classification problem with 2 classes, where $y_i = 1$ for $\boldsymbol{x}_i \in C_1$, and $y_i = -1$ for $\boldsymbol{x}_i \in C_2$. $\sigma(\cdot)$ denotes the ReLU activation function $\sigma(u) = \max(u, 0)$. Unlike deep linear networks, the analysis in this case requires two additional symmetry assumptions:

1. Each point $\boldsymbol{x}_i$ is drawn from a symmetric distribution $\mathcal{D}$ with density $f_{\mathcal{D}}(\mathbf{X})$, such that: $f_{\mathcal{D}}(\boldsymbol{x}_i) = f_{\mathcal{D}}(-\boldsymbol{x}_i)$.

2. W and $\boldsymbol{a}$ are initialized symmetrically so that $\boldsymbol{w}_{2i}^{(0)} = -\boldsymbol{w}_{2i-1}^{(0)}$ and $a_{2i} = -a_{2i-1}$ $\forall i \in [\frac{\mathtt{m}}{2}]$.

Theorem 9 *Retaining the assumptions stated above, at the beginning of the learning, the temporal dynamics of the model can be shown to obey the following update rule:*

$$W^{(t+1)} \approx W^{(t)} - \mu \frac{1}{2} \Big[(\boldsymbol{a}\boldsymbol{a}^\intercal) W^{(t)} \Sigma_{XX} - \tilde{M}^{(t)} \Big].$$

Above $\tilde{M}^{(t)}$ denotes the difference between the centroids of the 2 classes, computed in the half-space defined by $\boldsymbol{w}_r^{(t)} \cdot \boldsymbol{x} \geq 0$.

The complete proof can be found in App. B.3. Thm 9 is reminiscent of the single-layer linear model dynamics $\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \mu G_r^{(t)}$, and we may conclude that when it holds, using the principal coordinate system, the rate of convergence of the j-th column of $W^{(t)}$ is governed by the singular value d_j. Fig. 2b empirically demonstrated this result, showing the weight convergence of 10 ReLU models trained on the small mammals dataset (5 classes out of CIFAR-100, see App. D.4 for details), along different principal directions of the data.

4. PC-Bias: Empirical Study

In this section, we first analyze deep linear networks, showing that the convergence rate is indeed governed by the principal singular values of the data, which demonstrates the plausibility of the assumptions made in Section 3. We continue by extending the scope of the investigation to non-linear neural networks, finding evidence for the *PC-bias* mostly in the earlier stages of learning.

4.1 Methodology

We say that a linear network is L-layered when it has $L-1$ hidden fully connected (FC) layers (without convolutional layers). In our empirical study, we relaxed some assumptions of the theoretical analysis, to increase the resemblance of the trained networks to networks in common use. Specifically, we changed the initialization to the commonly used Glorot initialization, replaced the L_2 loss with the cross-entropy loss, and employed SGD instead of the deterministic GD. As to be expected, the original assumptions yielded similar results (see §C.3.1). The results presented summarize experiments with networks of equal width across all hidden layers, fixing the moderate[4] value of $\mathtt{m} = 1024$ in order to assess the relevance of our asymptotic results obtained with $\mathtt{m} \to \infty$. Using a different width for each layer yielded similar qualitative results. Details regarding the hyper-parameters, architectures, and datasets can be found in §D.1, §D.3, and §D.4 respectively.

4.2 PC-bias in Deep Linear Networks

In this section, we train L-layered linear networks, then compute their compact representations $\boldsymbol{W}$ rotated to align with the canonical coordinate system (Def. 1). Note that each row $\boldsymbol{w}(r)$ in $\boldsymbol{W}$ essentially defines the one-vs-all separating hyper-plane corresponding to class r.

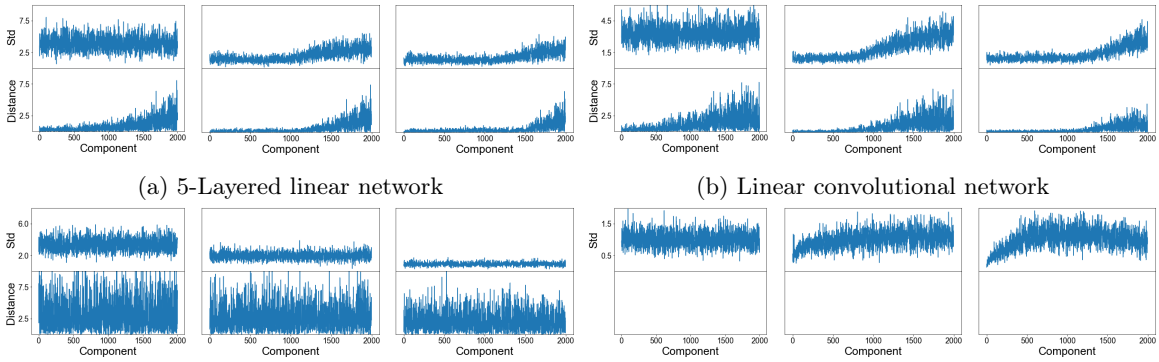
To examine both the variability between models and their convergence rate, we inspect $\boldsymbol{w}(r)$ at different time points during learning. The rate of convergence can be measured directly, by observing the changes in the weights of each element in $\boldsymbol{w}(r)$. These weight values[5] are compared with the corresponding optimal weights w_{opt}. The variability between models is measured by computing the standard deviation (std) of each row vector $\boldsymbol{w}(r)$ across N models, obtained with different random initializations.

We begin with linear networks. We trained 10 5-layered FC linear networks, and 10 linear st-VGG convolutional networks. When analyzing the compact representation of such networks we observe a similar behavior—weights corresponding to larger principal components converge faster to the optimal value, and their variability across models converges faster to 0 (Figs. 4a,4b). Thus, while the theoretical results are asymptotic, the *PC-bias* is empirically seen throughout the entire learning process of deep linear networks.

We note that in the principal coordinate system, the absolute values of the optimal solution are often higher in directions corresponding to lower principal components. Nevertheless, faster convergence in the directions corresponding to the higher principal components is seen even after this confounding factor is eliminated by normalization (see App. C.3.2).

Whitened data. The *PC-bias* is neutralized when the data is whitened, at which point Σ_{XX} is the scaled identity matrix. In Fig. 4c, we plot the results of the same experimental protocol while using a ZCA-whitened dataset. As predicted, the networks no longer show any bias towards any principal direction. Weights in all directions are scaled similarly, and the std over all models is the same in each epoch, irrespective of the principal direction. (Additional experiments show that this is *not* an artifact, due to the lack of uniqueness when deriving the principal eigenvectors of a white signal).

4. Note that for most image datasets, q (the input dimension) is much larger than 1024, and therefore $\frac{q}{\mathtt{m}} >> 1$. In this sense $\mathtt{m} = 1024$ is moderate, as one can no longer assume $A_l^{(t)}(\mathtt{m}) \approx I$.
5. We note that the weights tend to start larger for smaller principal components, see Fig. 4a left.



(a) 5-Layered linear network

(b) Linear convolutional network

(c) 5-Layered linear network, whitened data

(d) Non-linear convolutional network

Figure 4: Convergence of the compact representation along the principal directions in different epochs. The value of the X-axis corresponds to the index of a principal eigenvalue, from the most significant to the least significant. (a) 10 5-layered linear networks trained on the cats and dogs dataset. 3 plots are provided, corresponding to snapshots taken at different stages of learning: the beginning (epoch 1, left), intermediate stage (epoch 100, middle), and close to convergence (epoch 1500 right). Bottom panel: average distance of the weights in $\boldsymbol{w}(1)$ from the optimal linear classifier; top panel: respective std. (b) Similarly, for 10 linear st-VGG convolutional networks, trained on CIFAR-10 (epochs plotted: $1, 100, 500$). (c) Similarly, for 10 5-layered linear networks, trained on the cats and dogs dataset, with ZCA-whitening (epochs plotted: $1, 2, 1000$). (d) Similarly, for 10 **non-linear** st-VGG networks trained on the cats and dogs dataset. Here the distance to the optimal solution is not well defined and we therefore only show the std (epochs plotted: $1, 10, 100$). In all plots, the data dimension is 3072, while only the largest 2000 principal components are shown for clarity. Visualization of the values of λ_i can be found in Fig. 16.

4.3 PC-Bias in General CNNs

In this section, we investigate the manifestation of the *PC-bias* in non-linear deep convolutional networks. As we cannot directly track the learning dynamics separately in each principal direction of non-linear networks, we adopt two different evaluation mechanisms:

(i) Linear approximation. We considered several linear approximations, but since all of them showed the same qualitative behavior, we report results with the simplest one. Specifically, to obtain a linear approximation of a non-linear network, we ignore the non-linear layers of max-pooling and batch-normalization, and compute the compact representation as in Section 3 using only linear activations. We then align this matrix with the canonical coordinate system (Def. 1), and observe the evolution of the weights and their std across models along the principal directions during learning. Note that now the networks do not converge to the same compact representation, which is not unique. Nevertheless, we see that the *PC-bias* governs the weight dynamics to a noticeable extent.

We note that, in these networks, a large fraction of the lowest principal components hardly ever changes during learning. Nevertheless, the *PC-bias* affects the higher principal components, most notably at the beginning of training (see Fig. 4d). Thus weights corresponding to higher principal components converge faster, and the std across models of such weights decreases faster for higher principal components.

(ii) Projection to higher PC's. We created a modified *test-set*, by projecting each test example on the span of the first P principal components. This is equivalent to reducing the dimensionality of the test set to P using PCA. We trained an ensemble of N=100 st-VGG networks on the original training set of the small mammals dataset (see App. D.4), then evaluated these networks during training on 4 versions of the test-set, reduced to P=1,10,100,1000 dimensions respectively. Mean accuracy is plotted in Fig. 3. Similar results are obtained when training VGG-19 networks on CIFAR-10, see §C.4.

Taking a closer look at Fig. 3, we see that when evaluated on lower dimensionality test-data (P=1,10), the networks' accuracy peaks after a few epochs, at which point performance starts to decrease. This result suggests that the networks rely more heavily on these dimensions in the earlier phases of learning, and then continue to learn other things. In contrast, when evaluated on higher dimensionality test-data (P=100,1000), accuracy continues to rise, longer so for larger P. This suggests that significant learning of the additional dimensions continues in later stages of the learning.

5. PC-Bias: Learning Order Constancy

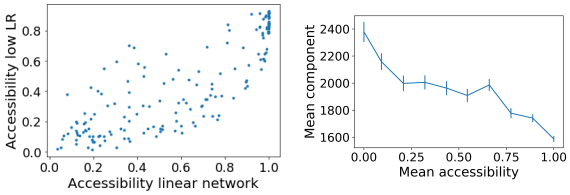
In this section, we show that the *PC-bias* is significantly correlated with the learning order of deep neural networks, and can therefore partially account for the *LOC-effect* described in Section 1. Following Hacohen et al. (2020), we measure the "speed of learning" of each example by computing its *accessibility* score. This score is computed per example, and characterizes how fast an ensemble of N networks memorizes it. Formally, $\textit{accessibility}(\boldsymbol{x}) = \mathbb{E}\left[\mathbb{1}(f_i^e(\boldsymbol{x}) = y(\boldsymbol{x}))\right]$, where $f_i^e(\boldsymbol{x})$ denotes the outcome of the i-th network trained over e epochs, and the mean is taken over networks and epochs. For the set of datapoints $\{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^n$, *Learning Order Constancy* is manifested by the high correlation between 2 instances of $\textit{accessibility}(\boldsymbol{x})$, each computed from a different ensemble.

PC-bias is shown to pertain to *LOC* in two ways: First, in Section 5.1 we show a high correlation between the learning order in deep linear and non-linear networks. Since the *PC-bias* fully accounts for *LOC* in deep linear networks, this suggests that it also accounts (at least partially) for the observed *LOC* in non-linear networks. Comparison with the *critical principal component* verifies this assertion. Second, we show in Section 5.2 that when the *PC-bias* is neutralized, *LOC* diminishes as well. In Section 5.3 we discuss the relationship between the *spectral bias*, *PC-bias* and the *LOC-effect*.

5.1 PC-Bias is Correlated with LOC

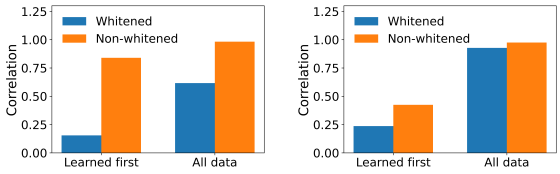
We first compare the order of learning of non-linear models and deep linear networks by computing the correlation between the *accessibility* scores of both models. This comparison reveals high correlation ($r = 0.85$, $p < 10^{-45}$), as seen in Fig. 5a. To investigate directly the connection between the *PC-bias* and *LOC*, we define the *critical principal component* of an example to be the first principal component P, such that a linear classifier trained on the original data can classify the example correctly when projected to P principal components. We trained N=100 st-VGG networks on the cats and dogs dataset (2 classes out of CIFAR-10, see §D.4 for details), and computed for each example its *accessibility* score and its *critical principal component*. In Fig. 5b we see strong negative correlation between the two scores

($p=-0.93$, $r<10^{-4}$), suggesting that the *PC-bias* affects the order of learning as measured by *accessibility*.



(a) (b)

Figure 5: (a) Correlation between the *accessibility* score of $N=100$ st-VGG networks trained with a low learning rate[6], and $N=100$ linear st-VGG networks, trained on small mammals (see §D.3, §D.4 for details). (b) The *critical principal component* score, plotted against the accessibility score of $N=100$ st-VGG networks trained on the cats and dogs dataset. *Accessibility* values were smoothed by a moving average filter of width 10. Error bars indicate standard error.



(a) Linear networks (b) Non-linear networks

Figure 6: *LOC* as measured with and without *PC-bias*. Each bar represents the correlation between the learning order of two collections of 10 networks trained on CIFAR-10. Orange bars represent natural images, in which the *PC-bias* is present, while blue bars represent whitened data, in which the *PC-bias* is neutralized. As *PC-bias* is more prominent earlier on, we compute these correlations using the entire data (right two bars), and using the subset of 20% "fastest learned" examples (left two bars).

5.2 Neutralizing the PC-bias Leads to Diminishing LOC

Whitening the data eliminates the *PC-bias* as shown in Fig. 4c, since all the singular values are now identical. Here we use this observation to further probe into the dependency of the *Learning Order Constancy* on the *PC-bias*. Starting with the linear case, we trained four ensembles of $N=10$ two-layered linear networks (following the same methodology as Section 4.1) on the cats and dogs dataset, two with and two without ZCA-whitening. We compute the *accessibility* score for each ensemble separately, and correlate the scores of the two ensembles in each test case. Each correlation captures the consistency of the *LOC-effect* for the respective condition. This correlation is expected to be very high for natural images. Low correlation implies that the *LOC-effect* is weak, as training the same network multiple times yields a different learning order.

Fig. 6a shows the results for deep linear networks. As expected, the correlation when using natural images is very high. However, when using whitened images, correlation plummets, indicating that the *LOC-effect* is highly dependent on the *PC-bias*. We note that the drop in the correlation is much higher when considering only the 20% "fastest learned" examples, suggesting that the *PC-bias* affects learning order more strongly at earlier stages of learning.

Fig. 6b shows the results when repeating this experiment with non-linear networks, training two collections of $N=10$ VGG-19 networks on CIFAR-10. We observe that neutralizing the *PC-bias* in this case affects *LOC* much less, suggesting that the *PC-bias* can only partially account for the *LOC-effect* in the non-linear case. Nevertheless, we note that at the beginning of learning, when the *PC-bias* is most pronounced, once again the drop is much larger and very significant (down by half), see left two bars of Fig. 6b.

6. As non-linear models achieve the accuracy of linear models within an epoch or 2, a low learning rate is used.

5.3 Spectral Bias, PC-bias, and LOC

The *spectral bias* (Rahaman et al., 2019) characterizes the dynamics of learning in neural networks differently, asserting that initially neural models can be described by low frequencies only. This may provide an alternative explanation to LOC. Recall that LOC is manifested in the consistency of the *accessibility* score across networks. To compare the *spectral bias* and *accessibility* score, we first need to estimate for each example whether it can be correctly classified by a low-frequency model. Accordingly, we define for each example a *discriminability* score—the percentage out of its k neighbors that share with it a class identity. Intuitively, an example has a low *discriminability* score when it is surrounded by examples from other classes, which would presumably force the learned boundary to incorporate high frequencies. In §C.5 we show that in the 2D case analyzed by Rahaman et al. (2019), this measure strongly correlates ($r{=}-0.8$, $p < 10^{-2}$) with the spectral bias.

We trained several networks (VGG-19 and st-VGG) on several real datasets, including small mammals, STL-10, CIFAR-10/100, and a subset of ImageNet-20 (see App. D.4). For each network and dataset, we compute the *accessibility* score as well as the *discriminability* of each example. The vector space, in which discriminability is evaluated, is either the raw data or the network's perceptual space (penultimate layer activation). The correlation between these scores is shown in Table 1.

Table 1: Correlation between *accessibility* and *discriminability*.

Dataset	Raw data	Penultimate
Small mammals	0.46	0.85
ImageNet 20	0.01	0.54
CIFAR-100	0.51	0.85
STL10	0.44	0.7

Using raw data, low correlation is still seen between the *accessibility* and *discriminability* scores when inspecting the smaller datasets (small mammals, CIFAR-100, and STL10). This correlation disappears when considering the larger ImageNet-20 dataset. It would appear that on its own, the *spectral bias* cannot adequately explain the *LOC-effect*. On the other hand, in the perceptual space, the correlation between *discriminability* and *accessibility* is quite significant for all datasets. Differently from the supposition of the spectral bias, it seems that networks learn a representation where the *spectral bias* is evident, but this bias does not necessarily govern its learning before the representation has been obtained.

Discussion. In the limit of infinite width, the spectral bias phenomenon in deep linear networks follows from the PC-bias. In function space, the training process of neural networks can be decomposed along different directions defined by the eigenfunctions of the neural tangent kernel, where each direction has its convergence rate and the rate is determined by the corresponding eigenvalue (Cao et al., 2021). Since the Neural Tangent Kernel for a deep linear network is proportional to Σ_{XX}, the spectral bias, in this case, corresponds to the statement that the right singular vectors of X are learned at rates corresponding to their

(squared) singular values. Looking at the spectral decomposition of $\boldsymbol{W}X$, we see that the PC Bias implies in function space that the right singular vectors of X are learned at rates corresponding to their singular values. Thus the PC Bias implies the Spectral Bias in this model.

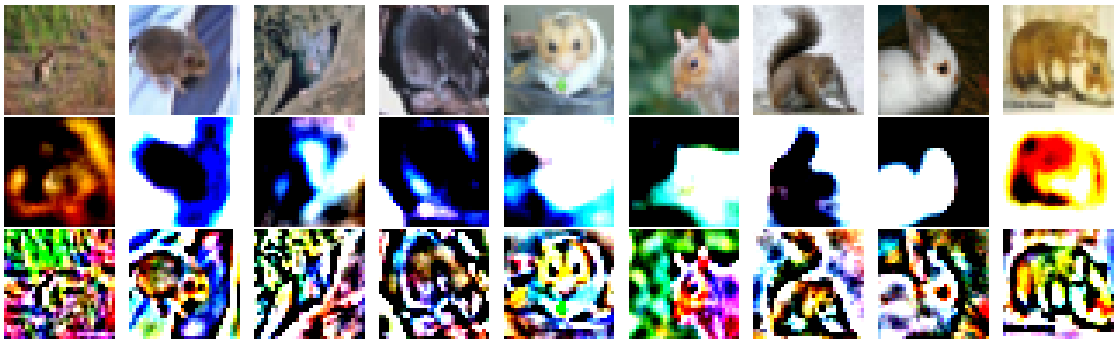
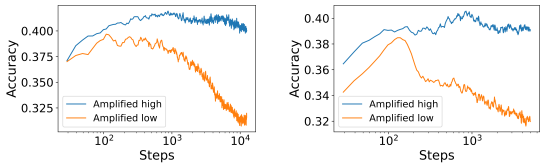


Figure 7: Visualization of the small mammals dataset, with amplification of 1.5% of its principal components by a factor of 10. Top: original data; middle: data amplified along the highest principal components; bottom: data amplified along the lowest principal components

6. PC-Bias: Further Implications

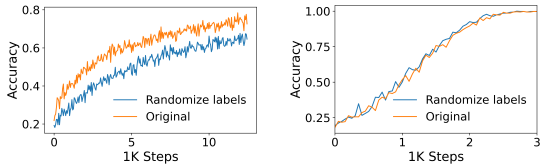
Early Stopping and the Generalization Gap. Considering natural images, it is often assumed that the least significant principal components of the data represent noise (Torralba and Oliva, 2003). In such cases, our analysis predicts that as noise dominates the components learned later in learning, early stopping is likely to be beneficial. To test this hypothesis directly, we manipulated CIFAR-10 to amplify the signal in either the 1.5% most significant (higher) or 1.5% least significant (lower) principal components (see examples in Fig. 7). Accuracy over the original test set, after training 10 st-VGG and linear st-VGG networks on these manipulated images, can be seen in Fig. 8. Both in linear and non-linear networks, early stopping is more beneficial when lower principal components are amplified, and significantly less so when higher components are amplified, as predicted by the *PC-bias*.

Slower Convergence with Random Labels. Deep neural models can learn any random label assignment to a given training set (Zhang et al., 2017). However, when trained on randomly labeled data, convergence appears to be much slower (Krueger et al., 2017). Assume, as before, that in natural images the lower principal components are dominated by noise. We argue that the *PC-bias* now predicts this empirical result, since learning randomly labeled examples requires a signal present in lower principal components. To test this hypothesis directly, we trained 10 two-layered linear networks (following the same methodology as in Section 4.1) using datasets of natural images. Indeed, these networks converge slower with random labels (see Fig. 9a). In Fig. 9b we repeat this experiment after having whitened the images, to neutralize the *PC-bias*. Now convergence rate is identical, whether the labels are original or shuffled. Clearly, in deep linear networks, the *PC-bias* gives a full account for this phenomenon.



(a) linear network (b) non-linear network

Figure 8: Comparing the accuracy trajectory when amplifying the highest (blue line) and lowest (orange line) principal components.

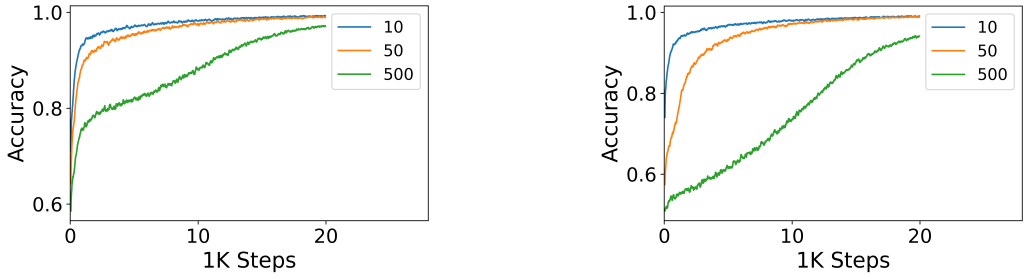


(a) original data (b) whitened data

Figure 9: Learning curves when using real and shuffled labels: 10 two-layered linear networks, (a) before and (b) after whitening.

To further check the relevance of this account to non-linear networks, we artificially generate datasets where only the first P principal components are discriminative, while the remaining components become noise by design. We constructed two such datasets: in one the labels are correlated with the original labels, while in the other they are not. Specifically, PCA is used to reduce the dimensionality of a two-class dataset to P, and the optimal linear separator in the reduced representation is computed. Next, all the labels of points that are incorrectly classified by the optimal linear separator are switched, so that the train and test sets are linearly separable by this separator. Note that the modified labels are still highly correlated with the original labels (for $P = 500$: $p = 0.82$, $r < 10^{-10}$). The second dataset is generated by repeating the process while starting from randomly shuffled labels. This dataset is likewise fully separable when projected to the first P components, but its labels are uncorrelated with the original labels (for $P = 500$: $p = 0.06$, $r < 10^{-10}$).

The mean training accuracy of 10 non-linear networks with P=10,50,500 is plotted in Fig. 10a (first dataset) and Fig. 10b (second dataset). In both cases, the lower P is (namely, only the first few principal components are discriminative), the faster the data is learned by the non-linear network. Whether the labels are real or shuffled makes little qualitative difference, as predicted by the *PC-bias*.



(a) Original labels (b) Shuffled labels

Figure 10: Learning curves of st-VGG networks trained on 3 datasets, which are linearly separable after projection to the highest P principal components (see legend).

7. Summary and Discussion

When trained with gradient descent, the convergence rate of the over-parameterized deep linear network model is provably governed by the eigendecomposition of the data. Specifically, we show that parameters corresponding to the most significant principal components converge faster than the least significant components. Empirical evidence is provided for the relevance of this result to more realistic non-linear networks. We term this effect *PC-bias*. This result provides a complementary account for some prevalent empirical observations, including the benefit of early stopping and the slower convergence rate with shuffled labels.

We use the *PC-bias* to explain the *Learning Order Constancy (LOC)*. Different empirical schemes are used to show that examples learned at earlier stages are more distinguishable by the data's higher principal components, which may indicate that networks' training relies more heavily on higher principal components early on. A causal link between the *PC-bias* and the *LOC-effect* is established, as the *LOC-effect* diminishes when the *PC-bias* is eliminated by whitening the images. Finally, we analyze these findings given a related phenomenon termed *spectral bias*. While the *PC-bias* may be more prominent early on, the *spectral bias* may be more important in later stages of learning.

Acknowledgments

We thank our two reviewers for the elaborated and insightful suggestions, which contributed to this work. This work was supported in part by a grant from the Israeli Ministry of Science and Technology, and by the Gatsby Charitable Foundations.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Adv. Neural Inform. Process. Syst.*, pages 6155–6166, 2019.

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

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019a.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332, 2019b.

Bubacarr Bah, Holger Rauhut, Ulrich Terstiege, and Michael Westdickenberg. Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers. *CoRR*, abs/1910.05505, 2019.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

Ronen Basri, David W. Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. In *Advances in Neural Information Processing Systems*, pages 4761–4771, 2019.

Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In *International Conference on Machine Learning*, pages 685–694. PMLR, 2020.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 2205–2211. ijcai.org, 2021. doi: 10.24963/ijcai.2021/304.

Lénaïc Chizat, Edouard Oyallon, and Francis R. Bach. On lazy training in differentiable programming. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 2933–2943, 2019.

Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. The grammar-learning trajectories of neural language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8281–8297. Association for Computational Linguistics, 2022.

Kamaludin Dingle, Chico Q Camargo, and Ard A Louis. Input–output maps are strongly biased towards simple outputs. *Nature communications*, 9(1):1–7, 2018.

Simon Du and Wei Hu. Width provably matters in optimization for deep linear neural networks. In *International Conference on Machine Learning*, pages 1655–1664. PMLR, 2019.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.

Simon S. Du, Wei Hu, and Jason D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 382–393, 2018.

Wei Fu and Tim Menzies. Easy over hard: A case study on deep learning. In *Proceedings of the 2017 11th joint meeting on foundations of software engineering*, pages 49–60, 2017.

Kenji Fukumizu. Effect of batch learning in multilayer neural networks. *Gen*, 1(04):1E–03, 1998.

Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32:3202–3211, 2019.

Daniel Gissin, Shai Shalev-Shwartz, and Amit Daniely. The implicit bias of depth: How incremental learning drives generalization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

Suriya Gunasekar, Jason D. Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9482–9491, 2018.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pages 2535–2544. PMLR, 2019.

Guy Hacohen, Leshem Choshen, and Daphna Weinshall. Let's agree to agree: Neural networks share classification order on real datasets. In *International Conference on Machine Learning*, pages 3950–3960. PMLR, 2020.

Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In *International Conference on Machine Learning*. PMLR, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Reinhard Heckel and Mahdi Soltanolkotabi. Denoising and regularization via exploiting the structural bias of convolutional generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Wei Hu, Lechao Xiao, Ben Adlam, and Jeffrey Pennington. The surprising simplicity of the early-time learning dynamics of neural networks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33*, 2020a.

Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. In *8th International Conference on Learning*

Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020b.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8580–8589, 2018.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Hui Jin and Guido Montúfar. Implicit bias of gradient descent for mean squared error regression with wide neural networks. *CoRR*, abs/2006.07356, 2020.

Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin L. Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. SGD on neural networks learns functions of increasing complexity. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3491–3501, 2019.

Kenji Kawaguchi. Deep learning without poor local minima. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 586–594, 2016.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, 7, 1994.

David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Maxinder S. Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, and Aaron C. Courville. Deep nets don't learn via memorization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

Thomas Laurent and James von Brecht. Deep linear neural networks with arbitrary loss: All local minima are global. *CoRR*, abs/1712.01473, 2017.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4313–4324. PMLR, 2020.

Phan-Minh Nguyen. Analysis of feature learning in weight-tied autoencoders via the mean field lens. *arXiv preprint arXiv:2102.08373*, 2021.

Guillermo Valle Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *7th International*

Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

Iuliia Pliushch, Martin Mundt, Nicolas Lupp, and Visvanathan Ramesh. When deep classifiers agree: Analyzing correlations between learning order and image statistics. *arXiv preprint arXiv:2105.08997*, 2021.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 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.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33*, 2020.

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

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: computation in neural systems*, 14(3):391–412, 2003.

Jonathan G Tullis and Aaron S Benjamin. On the effectiveness of self-paced learning. *Journal of memory and language*, 64(2):109–118, 2011.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

Daphna Weinshall and Dan Amir. Theory of curriculum learning, with convex loss functions. *Journal of Machine Learning Research*, 21(222):1–19, 2020.

Francis Williams, Matthew Trager, Daniele Panozzo, Cláudio T. Silva, Denis Zorin, and Joan Bruna. Gradient dynamics of shallow univariate relu networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8376–8385, 2019.

Lei Wu, Qingcan Wang, and Chao Ma. Global convergence of gradient descent for deep linear residual networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13368–13377, 2019a.

Yifan Wu, Barnabas Poczos, and Aarti Singh. Towards understanding the generalization bias of two layer convolutional linear classifiers with gradient descent. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1070–1078. PMLR, 2019b.

Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. *CoRR*, abs/2205.11320, 2022.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. *Advances in Neural Information Processing Systems*, 32: 6598–6608, 2019.

Chulhee Yun, Shankar Krishnan, and Hossein Mobahi. A unifying view on implicit bias in training linear neural networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Yi Zhou and Yingbin Liang. Critical points of linear neural networks: Analytical forms and landscape properties. In *Proc. Sixth International Conference on Learning Representations (ICLR)*, 2018.

Appendix

Appendix A. Random Matrices

A.1 Multiplication of Random Matrices

In this section, we present and prove some statistical properties of general random matrices and their multiplications. Let $\{Q_n \in \mathbb{R}^{m_n \times m_{n-1}}\}_{n=1}^{N}$ denote a set of random matrix whose elements are sampled iid from a distribution with mean 0 and variance σ_n^2, whose kurtosis is bounded by c, and whose support is compact where the norm of each element is bounded by

$c'\sigma_n^2$ for fixed constants c, c'. Let

$$\boldsymbol{Q}^l = \prod_{n=l}^{1} Q_n = Q_l \cdot \ldots \cdot Q_1, \qquad B^l = {\boldsymbol{Q}^l}^\top \boldsymbol{Q}^l \in \mathbb{R}^{m_0 \times m_0}, \tag{7}$$

$$\mathsf{Q}^l = \prod_{n=N}^{l+1} Q_n = Q_N \cdot \ldots \cdot Q_{l+1}, \qquad A^l = \mathsf{Q}^l {\mathsf{Q}^l}^\top \in \mathbb{R}^{m_N \times m_N}. \tag{8}$$

Theorem 10 *For random matrices A^l and B^l as defined in (7)-(8)*

$$\mathbb{E}(B^l) = \beta_l I, \qquad \beta_l = \prod_{n=1}^{l} m_n \sigma_n^2, \tag{9}$$

$$\mathbb{E}(A^l) = \alpha_l I, \qquad \alpha_l = \prod_{n=l+1}^{N} m_{n-1} \sigma_n^2. \tag{10}$$

Proof We only prove (9), as the proof of (10) is similar. To simplify the presentation, we use the following auxiliary notations: $V = Q_1$, $U = \prod_{n=l}^{2} Q_n \implies \boldsymbol{Q}^l = UV$.

The proof proceeds by induction on l.

- $l = 1$:

$$\mathbb{E}[B_{ij}^1] = \mathbb{E}[\sum_{k=1}^{m_1} V_{ki} V_{kj}] = \sum_{k=1}^{m_1} \mathbb{E}[V_{ki}]\mathbb{E}[V_{kj}] = 0 \qquad i \neq j,$$

$$\mathbb{E}[B_{ii}^1] = \mathbb{E}[\sum_{k=1}^{m_1} V_{ki} V_{ki}] = \sum_{k=1}^{m_1} \mathbb{E}[V_{ki}^2] = m_1 \sigma_1^2.$$

  Thus $\mathbb{E}(B^1) = \beta_1 I$.

- Assume that (9) holds for $l - 1$.

$$B_{ij}^l = \sum_k \boldsymbol{Q}_{ki}^l \boldsymbol{Q}_{kj}^l = \sum_k \sum_\nu U_{k\nu} V_{\nu i} \sum_\rho U_{k\rho} V_{\rho j},$$

  and therefore

$$\mathbb{E}[B_{ij}^l] = \sum_k \sum_\nu \sum_\rho \mathbb{E}[U_{k\nu} V_{\nu i} U_{k\rho} V_{\rho j}] = \sum_\nu \sum_\rho \mathbb{E}[V_{\nu i} V_{\rho j}] \sum_k \mathbb{E}[U_{k\nu} U_{k\rho}]. \tag{11}$$

  The last transition follows from the independence of U and V. From (11), where we denote $B' = U^\top U$

$$\mathbb{E}[B_{ij}^l] = \sum_\nu \mathbb{E}[V_{\nu i}] \sum_\rho \mathbb{E}[V_{\rho j}]\mathbb{E}[(U^\top U)_{\nu\rho}] = 0 \qquad i \neq j,$$

$$\mathbb{E}[B_{ii}^l] = \sum_\nu \sum_\rho \mathbb{E}[V_{\nu i} V_{\rho i}]\mathbb{E}[(U^\top U)_{\nu\rho}] = \sum_{\nu=1}^{m_1} \mathbb{E}[V_{\nu i}^2]\mathbb{E}[B'_{\nu\nu}] = m_1 \sigma_1^2 \prod_{n=2}^{l} m_n \sigma_n^2 = \beta_l.$$

  where the last transition above follows from the induction assumption applied to $B' = U^\top U$. Thus $\mathbb{E}(B^l) = \beta_l I$ and (9) follows.

■

Let $\mathbb{m}$ denote the width of the smallest hidden layer, $\mathbb{m} = \min(m_1, \ldots, m_{L-1})$, and assume that $\max(m_1, \ldots, m_{L-1}) \leq \mathbb{m} + \Delta m$ for a fixed constant Δm. m_0, m_L are likewise fixed constants (m_0 corresponds to the input dimension q and m_L corresponds to the number of classes K), while $\mathbb{m}$ is not bounded. Assume that the distribution of the random matrices $\{Q_n\}_{n=1}^L$ is normalized as specified in Def. 4, where specifically

$$\sigma_n^2 = \frac{2}{m_{n-1} + m_n} \ \forall n \in [2 \ldots L-1], \qquad \sigma_1^2 = \frac{1}{m_1}, \qquad \sigma_L^2 = \frac{1}{m_{L-1}}. \tag{12}$$

It follows that asymptotically, when $\mathbb{m} \to \infty$, we can write

$$\begin{aligned} m_n\sigma_n^2 &= 1 + O\left(\frac{1}{\mathbb{m}}\right) \quad n \in [1 \ldots L-1], \qquad m_L\sigma_L^2 = \frac{m_L}{\mathbb{m}} + O\left(\frac{1}{\mathbb{m}}\right), \\ m_{n-1}\sigma_n^2 &= 1 + O\left(\frac{1}{\mathbb{m}}\right) \quad n \in [2 \ldots L], \qquad m_0\sigma_1^2 = \frac{m_0}{\mathbb{m}} + O\left(\frac{1}{\mathbb{m}}\right). \end{aligned}$$

Corollary 11 *From Thm 10, using the initialization scheme specified in Def. 4 and (12)*

$$\begin{aligned} \mathbb{E}(B^l) &= I + O\left(\frac{1}{\mathbb{m}}\right) \quad \forall l \in [1 \ldots L-1], \qquad \mathbb{E}(B^L) = \frac{m_L}{\mathbb{m}} I + O\left(\frac{1}{\mathbb{m}}\right) = O\left(\frac{1}{\mathbb{m}}\right), \\ \mathbb{E}(A^l) &= I + O\left(\frac{1}{\mathbb{m}}\right) \quad \forall l \in [1 \ldots L-1], \qquad \mathbb{E}(A^0) = \frac{m_0}{\mathbb{m}} I + O\left(\frac{1}{\mathbb{m}}\right) = O\left(\frac{1}{\mathbb{m}}\right). \end{aligned}$$

Recall that in our asymptotic matrix notations, $O(\frac{1}{\mathbb{m}})$ is short-hand for a matrix which, for large enough $\mathbb{m}$, is upper bounded element-wise by $C\frac{1}{\mathbb{m}}$ where C denotes a fixed full rank matrix, and similarly $O(\mu^2)$ for small enough μ. When B^l is concerned $C \in \mathbb{R}^{m_0 \times m_0}$, and when A^l is concerned $C \in \mathbb{R}^{m_L \times m_L}$. Henceforth, for clarity and simplicity of notations, when we discuss asymptotic properties of functions of a certain matrix M, including its expected value $\mathbb{E}(M)$ or variance $\mathrm{var}(M)$, it is to be understood that the properties are considered element-wise.

Theorem 12 *Given random matrices A^l and B^l as defined in (7)-(8), and using the initialization scheme specified in Def. 4 and (12), we have*

$$\mathrm{var}(B^l) = O\left(\frac{1}{\mathbb{m}}\right), \qquad \mathrm{var}(A^l) = O\left(\frac{1}{\mathbb{m}}\right) \quad \forall l.$$

Proof Once again, we only provide a detailed proof for $\mathrm{var}(B^l)$, as the proof for $\mathrm{var}(A^l)$ is similar. From Corr 11, and since element-wise $Var(Z) = \mathbb{E}(Z^2) - [\mathbb{E}(Z)]^2$, it is sufficient to show that the following (somewhat stronger) assertion is valid:

$$\mathbb{E}[(B_{ij}^l)^2] = \begin{cases} O(\frac{1}{\mathbb{m}}) & i \neq j \\ 1 + O(\frac{1}{\mathbb{m}}) & i = j, l < L \\ O(\frac{1}{\mathbb{m}}) & i = j, l = L \end{cases}, \qquad \mathbb{E}[B_{ii}^l B_{jj}^l] = 1 + O\left(\frac{1}{\mathbb{m}}\right) \quad i \neq j \tag{13}$$

The proof proceeds by induction on l.

Base case $l = 1$. Below Q stands for Q_1, to simplify index notations.

$$\mathbb{E}[(B^1_{ij})^2] = \mathbb{E}\Big[\sum_{\nu=1}^{m_1} Q_{\nu i}Q_{\nu j}\sum_{\rho=1}^{m_1} Q_{\rho i}Q_{\rho j}\Big] = \begin{cases} \sum_{\nu=1}^{m_1}\mathbb{E}[Q^2_{\nu i}]\mathbb{E}[Q^2_{\nu j}] = \frac{1}{m_1} & i \neq j \\ \sum_{\nu=1}^{m_1}\sum_{\substack{\rho=1\\ \rho\neq\nu}}^{m_1}\mathbb{E}[Q^2_{\nu i}]\mathbb{E}[Q^2_{\rho i}] + \sum_{\nu=1}^{m_1}\mathbb{E}[Q^4_{\nu i}] = 1 + O\big(\frac{1}{\mathbb{m}}\big) & i = j \end{cases}$$

Above we use the assumed fixed bound on the kurtosis of the distribution of Q.

$$\mathbb{E}[B^l_{ii}B^l_{jj}] = \mathbb{E}\Big[\sum_{\nu=1}^{m_1} Q_{\nu i}Q_{\nu i}\sum_{\rho=1}^{m_1} Q_{\rho j}Q_{\rho j}\Big] = \sum_{\nu=1}^{m_1}\sum_{\rho=1}^{m_1}\frac{1}{m_1}\frac{1}{m_1} = 1.$$

Induction step. Assume that (13) holds for $l-1$, and similarly to the above, let Q stand for Q_l to simplify index notations. Let $\frac{1}{\tilde{m}_l}$ denote the variance of Q_l as defined in (12).

$$\begin{aligned}\mathbb{E}[(B^l_{ij})^2] &= \mathbb{E}\Big[\sum_{\nu=1}^{m_l}\sum_{\rho=1}^{m_l} Q_{\nu i}B^{l-1}_{\nu\rho}Q_{\rho j}\sum_{\alpha=1}^{m_l}\sum_{\beta=1}^{m_l} Q_{\alpha i}B^{l-1}_{\alpha\beta}Q_{\beta j}\Big] \qquad \text{(by independence)} \\ &= \sum_{\nu=1}^{m_l}\sum_{\rho=1}^{m_l}\sum_{\alpha=1}^{m_l}\sum_{\beta=1}^{m_l}\mathbb{E}[Q_{\nu i}Q_{\rho j}Q_{\alpha i}Q_{\beta j}]\mathbb{E}[B^{l-1}_{\nu\rho}B^{l-1}_{\alpha\beta}].\end{aligned} \tag{14}$$

When $i \neq j$, using the independence of the elements of Q_l and the induction assumption

$$\begin{aligned}\mathbb{E}[(B^l_{ij})^2] &= \sum_{\substack{\nu=1\\ \alpha=\nu}}^{m_l}\sum_{\substack{\rho=1\\ \beta=\rho}}^{m_l}\mathbb{E}[Q^2_{\nu i}]\mathbb{E}[Q^2_{\rho j}]\mathbb{E}[(B^{l-1}_{\nu\rho})^2] \\ &= \sum_{\nu=1}^{m_l}\sum_{\substack{\rho=1\\ \rho\neq\nu}}^{m_l}\frac{1}{\tilde{m}_l}\frac{1}{\tilde{m}_l}\mathbb{E}[(B^{l-1}_{\nu\rho})^2] + \sum_{\substack{\nu=1\\ \rho=\nu}}^{m_l}\frac{1}{\tilde{m}_l}\frac{1}{\tilde{m}_l}\mathbb{E}[(B^{l-1}_{\nu\nu})^2] = O\Big(\frac{1}{\mathbb{m}}\Big) \quad \forall l.\end{aligned}$$

When $i = j$, we collect below all the terms in (14) which are not 0:

$$\begin{aligned}E[(B^l_{ii})^2] = &\sum_{\substack{\nu=1\\ \rho=\nu}}^{m_l}\sum_{\substack{\alpha=1\\ \beta=\alpha\\ \alpha\neq\nu}}^{m_l}\mathbb{E}[Q^2_{\nu i}]\mathbb{E}[Q^2_{\alpha i}]\mathbb{E}[B^{l-1}_{\nu\nu}B^{l-1}_{\alpha\alpha}] + \sum_{\substack{\nu=1\\ \alpha=\nu}}^{m_l}\sum_{\substack{\rho=1\\ \beta=\rho\\ \rho\neq\nu}}^{m_l}\mathbb{E}[Q^2_{\nu i}]\mathbb{E}[Q^2_{\rho i}]\mathbb{E}[(B^{l-1}_{\nu\rho})^2] \\ &+ \sum_{\substack{\nu=1\\ \beta=\nu}}^{m_l}\sum_{\substack{\rho=1\\ \alpha=\rho\\ \rho\neq\nu}}^{m_l}\mathbb{E}[Q^2_{\nu i}]\mathbb{E}[Q^2_{\rho i}]\mathbb{E}[(B^{l-1}_{\nu\rho})^2] + \sum_{\substack{\nu=1\\ \rho=\alpha=\beta=\nu}}^{m_l}\mathbb{E}[Q^4_{\nu i}]\mathbb{E}[(B^{l-1}_{\nu\nu})^2].\end{aligned}$$

From the induction assumption and since the kurtosis of Q is bounded by the assumption

$$\begin{aligned}E[(B^l_{ii})^2] &= \sum_{\nu=1}^{m_l}\sum_{\substack{\alpha=1\\ \alpha\neq\nu}}^{m_l}\mathbb{E}[Q^2_{\nu i}]\mathbb{E}[Q^2_{\alpha i}]\mathbb{E}[B^{l-1}_{\nu\nu}B^{l-1}_{\alpha\alpha}] + O\Big(\frac{1}{\mathbb{m}}\Big) \\ &= \sum_{\nu=1}^{m_l}\sum_{\substack{\alpha=1\\ \alpha\neq\nu}}^{m_l}\frac{1}{\tilde{m}_l}\frac{1}{\tilde{m}_l}\mathbb{E}[B^{l-1}_{\nu\nu}B^{l-1}_{\alpha\alpha}] + O\Big(\frac{1}{\mathbb{m}}\Big) = \begin{cases} 1 + O\big(\frac{1}{\mathbb{m}}\big) & l < L \\ O\big(\frac{1}{\mathbb{m}}\big) & l = L \end{cases}\end{aligned} \tag{15}$$

To justify the last transition, recall that the initialization scheme specified in Def. 4 implies that $m_l \frac{1}{\tilde{m}_l} = 1 + O(\frac{1}{\mathbb{m}})\ \forall l \in [1 \dots L-1]$. Making use once again of the induction assumption, it follows from (15) that now $E[(B^l_{ii})^2] = 1 + O(\frac{1}{\mathbb{m}})$. However, since m_L is fixed whereas $\frac{1}{\tilde{m}_L} = O(\frac{1}{\mathbb{m}})$, it follows that $E[(B^L_{ii})^2] = O(\frac{1}{\mathbb{m}})$.

A similar argument will show that $\mathbb{E}[B^l_{ii} B^l_{jj}] = 1 + O(\frac{1}{\mathbb{m}})$ when $i \neq j$. ■

Theorem 13 *Let $\{X(\mathbb{m})\}$ denote a sequence of random matrices where $\mathbb{E}[X(\mathbb{m})] = F + O(\frac{1}{\mathbb{m}})$ and $\mathrm{var}[X(\mathbb{m})] = O(\frac{1}{\mathbb{m}})$. Then $X(\mathbb{m}) \xrightarrow{p} F$, where $\xrightarrow{p}$ denotes element-wise convergence in probability as $\mathbb{m} \to \infty$.*

Proof For every element (i, j) of matrix $X(\mathbb{m})$, we need to show that $\forall \varepsilon, \delta > 0\ \exists \mathbb{m}' \in \mathbb{N}$, such that $\forall \mathbb{m} > \mathbb{m}'$

$$Prob\left(|X_{ij}(\mathbb{m}) - F_{ij}| > \varepsilon.\right) < \delta$$

Henceforth we use x, f as shorthand for $X_{ij}(\mathbb{m}), F_{ij}$ respectively. Since $\mathbb{E}[X(\mathbb{m})] = F + O(\frac{1}{\mathbb{m}})$ where by definition the asymptotic behavior occurs element-wise, it follows that $\forall \varepsilon > 0$ $\exists \mathbb{m}_1 \in \mathbb{N}$ such that $\forall \mathbb{m} > \mathbb{m}_1$ we have

$$|\mathbb{E}(x) - f| < \frac{\varepsilon}{2},$$

in which case

$$Prob\left(|x - f| > \varepsilon\right) \leq Prob\left(|x - \mathbb{E}(x)| > \frac{\varepsilon}{2}\right).$$

Since $\mathrm{var}[X(\mathbb{m})] = O(\frac{1}{\mathbb{m}})$, it follows that $\forall \varepsilon, \delta > 0,\ \exists \mathbb{m}_2 \in \mathbb{N} \quad \ni \ \forall \mathbb{m} > \mathbb{m}_2$

$$\mathrm{var}(x) < \frac{\varepsilon^2}{4}\delta,$$

from the above, and using Chebyshev's inequality

$$Prob\left(|x - f| > \varepsilon\right) < \frac{4\mathrm{var}(x)}{\varepsilon^2} < \delta,$$

$\forall \mathbb{m} > \mathbb{m}'$, where $\mathbb{m}' = \max\{\mathbb{m}_1, \mathbb{m}_2\}$.

■

Corollary 14 *Let $A^l(\mathbb{m})$ and $B^l(\mathbb{m})$ denote a sequence of random matrices as defined in (7)-(8), corresponding to multi-layer linear models for which $\mathbb{m} = \min(m_1, ..., m_{L-1})$. Then*

$$B^l(\mathbb{m}) \xrightarrow{p} I \quad \forall l \in [1 \dots L-1], \qquad B^L(\mathbb{m}) \xrightarrow{p} 0,$$
$$A^l(\mathbb{m}) \xrightarrow{p} I \quad \forall l \in [1 \dots L-1], \qquad A^{(0)}(\mathbb{m}) \xrightarrow{p} 0.$$

Proof This result follows from Corr 11, Thm 12, and Thm 13. ■

A.2 The Dynamics of Random Matrices

We now formalize a dynamical system, which captures the evolution of the weight matrices $\{W_l\}$ by gradient descent as seen in (29)-(30). Specifically, given random matrices as defined in (7)-(8), consider a dynamical system whereby $Q_j \to Q_j - \Delta Q_j \ \forall j$, and

$$\begin{aligned} \Delta Q_j &= \mu\Big(\prod_{n=L}^{j+1} Q_n\Big)^\top G_r\Big(\prod_{n=j-1}^{1} Q_n\Big)^\top = \mu[\mathbb{Q}^j]^\top G_r[\boldsymbol{Q}^{j-1}]^\top, \\ G_r &= \boldsymbol{Q}^L \Sigma_{XX} - \Sigma_{YX}. \end{aligned} \tag{16}$$

Denoting $\boldsymbol{Q}^l \to \boldsymbol{Q}^l - \Delta\boldsymbol{Q}^l,\ B^l \to B^l - \Delta B^l$ and applying the product rule

$$\Delta\boldsymbol{Q}^l = \sum_{j=1}^{l}\Big(\prod_{n=l}^{j+1} Q_n\Big)\Delta Q_j\Big(\prod_{n=j-1}^{1} Q_n\Big) = \sum_{j=1}^{l}\Big(\prod_{n=l}^{j+1} Q_n\Big)\Delta Q_j \boldsymbol{Q}^{j-1}, \tag{17}$$

$$\Delta B^l = [\Delta\boldsymbol{Q}^l{}^\top \boldsymbol{Q}^l + \boldsymbol{Q}^l{}^\top \Delta\boldsymbol{Q}^l]. \tag{18}$$

Recall that in our notations, $O(\mu^2)$ is short-hand for a matrix which, for small enough μ, is upper bounded element-wise by $C\mu^2$ where C denotes a *fixed* full rank matrix. Similarly, $O\big(\frac{1}{\mathrm{m}}\big)$ is short-hand for a matrix which, for large enough m, is upper bounded element-wise by $C\frac{1}{\mathrm{m}}$. Since $m_0 = m_q$ and $m_L = m_K$, when B^l is concerned $C \in \mathbb{R}^{q\times q}$, and when A^l is concerned $C \in \mathbb{R}^{K\times K}$. In addition, we will use the notation $O(\varepsilon)$ as short-hand for a matrix that is upper bounded element-wise by $C\varepsilon$, where C denotes a *fixed* full rank matrix.

Theorem 15 *Let $B^l(\mathrm{m}) = \boldsymbol{Q}^l(\mathrm{m})^\top \boldsymbol{Q}^l(\mathrm{m})$ denote a sequence of random matrices as defined in (7) for $\{Q_n\}_{n=1}^L$, whose dynamics is captured by (16)-(18). Assume that $B^l(\mathrm{m})$ is full rank $\forall l, \mathrm{m}$. If*

$$\begin{aligned} &B^l(\mathrm{m}) \xrightarrow{p} I + O(\mu^2) \qquad \forall l \in [1\dots L-1], \qquad\qquad B^L(\mathrm{m}) \xrightarrow{p} O(\mu^2), \\ &\mathrm{var}[B^l(\mathrm{m})] = O(\mu^2) + O\left(\frac{1}{\mathrm{m}}\right) \qquad \forall l, \\ &\mathbb{E}[\boldsymbol{Q}^L(\mathrm{m})] = O(\mu) + O\left(\frac{1}{\mathrm{m}}\right). \end{aligned}$$

then

$$\begin{aligned} &\Delta B^l(\mathrm{m}) \xrightarrow{p} O(\mu^2), \qquad\qquad \mathrm{var}[\Delta B^l(\mathrm{m})] = O(\mu^2) + O\left(\frac{1}{\mathrm{m}}\right) \qquad \forall l, \\ &\mathbb{E}[\Delta\boldsymbol{Q}^L(\mathrm{m})] = O(\mu) + O\left(\frac{1}{\mathrm{m}}\right). \end{aligned}$$

Proof $B^l(\mathrm{m}) \xrightarrow{p} I + O(\mu^2)$ and $B^L(\mathrm{m}) \xrightarrow{p} O(\mu^2)$ implies that $\forall \varepsilon, \delta > 0\ \exists \hat{\mathrm{m}} \in \mathbb{N}$, such that $\forall \mathrm{m} > \hat{\mathrm{m}}$ and with probability larger than $1-\delta$:

$$B^l(\mathrm{m}) = I + O(\mu^2) + O(\varepsilon) \quad \forall l \le L-1, \qquad\qquad B^L(\mathrm{m}) = O(\mu^2) + O(\varepsilon). \tag{19}$$

To evaluate ΔB^l from (18), we start from (17) to obtain

$$\boldsymbol{Q}^{l\top}\Delta\boldsymbol{Q}^l = \sum_{j=1}^{l} \boldsymbol{Q}^{l\top}\Big(\prod_{l}^{j+1} Q_n\Big)\Delta Q_j \boldsymbol{Q}^{j-1}. \tag{20}$$

We make use of the Moore–Penrose pseudo-inverse of matrices $\{\boldsymbol{Q}^l\}$, defined as

$$[\boldsymbol{Q}^l]^+ = (\boldsymbol{Q}^{l\top}\boldsymbol{Q}^l)^{-1}\boldsymbol{Q}^{l\top}.$$

This definition is valid since by assumption B^l is full rank $\forall l$, and is therefore invertible. Additionally

$$[\boldsymbol{Q}^l]^+ \cdot [(\boldsymbol{Q}^l)^\top]^+ = (\boldsymbol{Q}^{l\top}\boldsymbol{Q}^l)^{-1}\boldsymbol{Q}^{l\top}\cdot\boldsymbol{Q}^l(\boldsymbol{Q}^{l\top}\boldsymbol{Q}^l)^{-1} = (\boldsymbol{Q}^{l\top}\boldsymbol{Q}^l)^{-1} = (B^l)^{-1}. \tag{21}$$

We next use (7) and (21) to simplify T_j—the j^{th} term in the sum (20)

$$\begin{aligned}
T_j &= \mu\boldsymbol{Q}^{l\top}\left(\prod_{l}^{j+1} Q_n\right)[\mathbb{Q}^j]^\top G_r[\boldsymbol{Q}^{j-1}]^\top\boldsymbol{Q}^{j-1} \\
&= \mu\boldsymbol{Q}^{l\top}\left(\prod_{l}^{j+1} Q_n\right)\cdot\boldsymbol{Q}^j[\boldsymbol{Q}^j]^+\cdot[(\boldsymbol{Q}^j)^\top]^+[\boldsymbol{Q}^j]^\top\cdot[\mathbb{Q}^j]^\top G_r B^{j-1} && \text{(Lemma 19)} \\
&= \mu\boldsymbol{Q}^{l\top}\boldsymbol{Q}^l[\boldsymbol{Q}^j]^+[(\boldsymbol{Q}^j)^\top]^+\boldsymbol{Q}^L G_r B^{j-1} && (22) \\
&= \mu B^l[B^j]^{-1}\boldsymbol{Q}^{L\top}[\boldsymbol{Q}^L\Sigma_{XX} - \Sigma_{YX}]B^{j-1} && \text{(using (16), (21))} \\
&= \mu B^l[B^j]^{-1}[B^L\Sigma_{XX} - \boldsymbol{Q}^{L\top}\Sigma_{YX}]B^{j-1} \\
&= -\mu\boldsymbol{Q}^{L\top}\Sigma_{YX} + O(\mu^2) + O(\varepsilon). && \text{(using (19))}
\end{aligned}$$

By assumption $\mathbb{E}[\boldsymbol{Q}^L] = O(\mu) + O(\varepsilon)$, and therefore

$$\mathbb{E}(T_j) = O(\mu^2) + O(\varepsilon) \quad\Longrightarrow\quad \mathbb{E}[\boldsymbol{Q}^{l\top}\Delta\boldsymbol{Q}^l] = \sum_{j=1}^{l}\mathbb{E}(T_j) = O(\mu^2) + O(\varepsilon). \tag{23}$$

Finally, since $\Delta\boldsymbol{Q}^{l\top}\boldsymbol{Q}^l = [\boldsymbol{Q}^{l\top}\Delta\boldsymbol{Q}^l]^\top$, from (18) and (23)

$$\mathbb{E}[\Delta B^l] = \mathbb{E}[\Delta\boldsymbol{Q}^{l\top}\boldsymbol{Q}^l] + \mathbb{E}[\Delta\boldsymbol{Q}^{l\top}\boldsymbol{Q}^l]^\top = O(\mu^2) + O(\varepsilon). \tag{24}$$

To conclude the proof, we will show that $\forall\varepsilon', \delta' > 0\ \exists\hat{\mathtt{m}}' \in \mathbb{N}$, such that $\forall\mathtt{m} > \hat{\mathtt{m}}'$

$$Prob\left(|\Delta B^l - O(\mu^2)| > \varepsilon'\right) < \delta' \qquad \text{element-wise.} \tag{25}$$

Let b denote an element of matrix $\Delta B^l - O(\mu^2)$. Recall that (24) is true element-wise with probability $(1-\delta)$ $\forall\varepsilon, \delta$ and $\forall\mathtt{m} > \hat{\mathtt{m}}$. Therefore, $\forall\varepsilon', \delta' > 0$, we can choose ε, δ and the corresponding $\hat{\mathtt{m}}'$ such that

$$Prob\left(|\mathbb{E}(b)| < \frac{\varepsilon'}{2}\right) > (1 - \frac{\delta'}{2}) \quad \forall\mathtt{m} > \hat{\mathtt{m}}', \tag{26}$$

$$Prob\left(|b| > \varepsilon'\right) = Prob\left(|b| > \varepsilon',\ |\mathbb{E}(b)| < \frac{\varepsilon'}{2}\right) + Prob\left(|b| > \varepsilon',\ |\mathbb{E}(b)| \geq \frac{\varepsilon'}{2}\right)$$
$$\leq Prob\left(|b - \mathbb{E}(b)| > \frac{\varepsilon'}{2}\right) + Prob\left(|\mathbb{E}(b)| \geq \frac{\varepsilon'}{2}\right).$$

$\text{var}(\Delta B^l) = O(\mu^2) + O\left(\frac{1}{\mathtt{m}}\right)$ implies that $\forall \varepsilon', \delta' > 0\ \exists \hat{\mathtt{m}}'' \in \mathbb{N}$, such that $\forall \mathtt{m} > \hat{\mathtt{m}}''$

$$\text{var}(b) < \frac{\delta' \varepsilon'^2}{8}, \tag{27}$$

Using Chebychev inequality, (26) and (27), we finally have

$$Prob\left(|b| > \varepsilon'\right) \leq \frac{4\text{var}(b)}{\varepsilon'^2} + \frac{\delta'}{2} < \delta', \qquad \forall \mathtt{m} > \max\{\hat{\mathtt{m}}', \hat{\mathtt{m}}''\}.$$

We can now conclude that (25) is true element-wise.

We will only provide a sketch of the proof for $\text{var}[\Delta B^l(\mathtt{m})]$ and $\mathbb{E}[\Delta \boldsymbol{Q}^L(\mathtt{m})]$, as a detailed proof follows very similar steps, and principles, to the proof presented above. To analyze the variance of $\Delta B^l(\mathtt{m})$, we start from (22) and observe that $\Delta B^l(\mathtt{m}) = -l\mu {\boldsymbol{Q}^L}^\top \Sigma_{YX} + O(\mu^2) + O(\varepsilon)$, from which (and the theorem's assumptions) it can be shown that $\text{var}[\Delta B^l(\mathtt{m})] = O(\mu^2) + O\left(\frac{1}{\mathtt{m}}\right)$. Similarly to (22), we can derive that $\mathbb{E}[\Delta \boldsymbol{Q}^l(\mathtt{m})] = -\mu L G_r + O(\mu^2) + O\left(\frac{1}{\mathtt{m}}\right)$, from which it follows that $\mathbb{E}[\Delta \boldsymbol{Q}^L(\mathtt{m})] = O(\mu) + O\left(\frac{1}{\mathtt{m}}\right)$. ■

Theorem 16 *Let $A^l(\mathtt{m}) = \boldsymbol{Q}^l(\mathtt{m}) \boldsymbol{Q}^l(\mathtt{m})^\top$ denote a sequence of random matrices as defined in (8) but where $A^{(0)}(\mathtt{m}) = \boldsymbol{Q}^0 \Sigma_{XX} {\boldsymbol{Q}^0}^\top$, whose dynamics is captured by (16). Assume that $A^l(\mathtt{m})$ is full rank $\forall l, \mathtt{m}$. If*

$$A^l(\mathtt{m}) \xrightarrow{p} I + O(\mu^2)\ \ \forall l \in [1 \dots L-1], \qquad A^0(\mathtt{m}) \xrightarrow{p} O(\mu^2)$$
$$\text{var}[A^l(\mathtt{m})] = O(\mu^2) + O\left(\frac{1}{\mathtt{m}}\right) \qquad \forall l$$
$$\mathbb{E}[\boldsymbol{Q}^0(\mathtt{m})] = O(\mu) + O\left(\frac{1}{\mathtt{m}}\right),$$

then

$$\Delta A^l(\mathtt{m}) \xrightarrow{p} O(\mu^2), \qquad \text{var}[\Delta A^l(\mathtt{m})] = O(\mu^2) + O\left(\frac{1}{\mathtt{m}}\right), \qquad \forall l$$
$$\mathbb{E}[\Delta \boldsymbol{Q}^0(\mathtt{m})] = O(\mu) + O\left(\frac{1}{\mathtt{m}}\right).$$

The proof is mostly similar to Thm 15.

A.3 Some Useful Lemmas

Lemma 17 *Given function $G(W) = \frac{1}{2}\|UWVX - Y\|_F^2$, its derivative is the following*

$$\frac{dG(W)}{dW} = U^\top UWVX(VX)^\top - U^\top Y(VX)^\top = U^\top [UWV\Sigma_{XX} - \Sigma_{YX}] V^\top.$$

Lemma 18 *Assume $\boldsymbol{Q} = \prod_{n=N}^{1} Q_n$, where $Q_n \in \mathbb{R}^{m_n \times m_{n-1}}$ denotes a random matrix whose elements are sampled iid from a distribution with mean 0 and variance σ_n^2 $\forall i, j$, initialized as defined in (12). Then*

$$\mathbb{E}[\boldsymbol{Q}_{ij}] = 0, \qquad \mathrm{var}[\boldsymbol{Q}_{ij}] = O\left(\frac{1}{\mathbb{m}}\right). \tag{28}$$

Proof By induction on N. Clearly for $N = 1$:

$$\mathbb{E}[\boldsymbol{Q}_{ij}] = \mathbb{E}[(Q_1)_{ij}] = 0, \qquad \mathrm{var}[\boldsymbol{Q}_{ij}] = \mathrm{var}[(Q_1)_{ij}] = \sigma_1^2.$$

Assume that (28) holds for $N-1$. Let $V = \prod_{n=N-1}^{1} Q_n$, $U = Q_N$. It follows that

$$\mathbb{E}[\boldsymbol{Q}_{ij}] = \mathbb{E}[(UV)_{ij}] = \sum_k \mathbb{E}[U_{ik}V_{kj}] = \sum_k \mathbb{E}[U_{ik}]\mathbb{E}[V_{kj}] = 0.$$

where the last transition follows from the independence of U and V. In a similar manner

$$\begin{aligned}\mathrm{var}[\boldsymbol{Q}_{ij}] = \mathbb{E}[\boldsymbol{Q}_{ij}^2] = \mathbb{E}[(\sum_k U_{ik}V_{kj})^2] = \mathbb{E}[\sum_k U_{ik}V_{kj} \sum_l U_{il}V_{lj}] = \sum_k \mathbb{E}[U_{ik}^2]\mathbb{E}[V_{kj}]^2 \\ = m_{N-1}\sigma_N^2 \frac{1}{m_{N-1}} \prod_{n=1}^{N-1} m_n \cdot \sigma_n^2 = \frac{1}{m_N} \prod_{n=1}^{N} m_n \cdot \sigma_n^2.\end{aligned}$$

With the initialization scheme defined in (12), $\mathrm{var}(\boldsymbol{Q}_{ij}) = O(\frac{1}{\mathbb{m}})$. ■

Lemma 19 *Let $C \in \mathbb{R}^{k \times m}$ and $V \in \mathbb{R}^{m \times k}$ both of rank k, where $k < m$. Then*

$$CV = I \implies C = CVV^{+}.$$

Proof By definition $C = V^{+}$, hence

$$C = V^{+}VC = CVV^{+}.$$

■

Appendix B. Supplementary Proofs

B.1 Deep Linear networks

Here we prove Prop. 5 as defined in Section 3.2.

Proposition 5. *Let $G_r^{(t)}$ (Def. 3) denote the gradient matrix at time t. Let $A_l^{(t)}$ and $B_l^{(t)}$ denote the gradient scale matrices, which are defined in (2). The compact representation $\boldsymbol{W}^{(t)}$ obeys the following dynamics:*

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \mu \sum_{l=1}^{L} A_l^{(t)} \cdot G_r^{(t)} \cdot B_{l-1}^{(t)} + O(\mu^2).$$

Proof At time t, the gradient step $\Delta W_l^{(t)}$ of layer l is defined by differentiating $L(\mathbb{X})$ with respect to $W_l^{(t)}$. Henceforth we omit index t for clarity. First, we rewrite $L(\mathbb{X})$ as follows:

$$L(\mathbb{X}; W_l) = \frac{1}{2}\| \left(\prod_{j=L}^{l+1} W_j\right) W_l \left(\prod_{j=l-1}^{1} W_j\right) X - Y\|_F^2.$$

Differentiating $L(\mathbb{X}; W_l)$ to obtain the gradient $\Delta W_l = \frac{\partial L(\mathbb{X};W_l)}{\partial W_l}$, using Lemma 17 above, we get

$$\Delta W_l = \left(\prod_{j=L}^{l+1} W_j\right)^\top [\boldsymbol{W}\Sigma_{XX} - \Sigma_{YX}]\left(\prod_{l-1}^{1} W_j\right)^\top. \tag{29}$$

Finally,

$$\Delta \boldsymbol{W} = \prod_{l=L}^{1}(W_l - \mu\Delta W_l) - \prod_{l=L}^{1} W_l = -\mu \sum_{l=1}^{L} \left(\prod_{n=L}^{l+1} W_n\right) \Delta W_l \left(\prod_{n=l-1}^{1} W_n\right) + O(\mu^2). \tag{30}$$

Substituting ΔW_l and G_r (as specified in Def. 3) into the above completes the proof. ■

B.2 Weight Evolution

The proofs in this section assume that the norm of the data matrix X is bounded, that the norms of the initial random weight matrices $W_l^{(0)}$ are bounded $\forall l, \mathbb{m}$, and that the norms of the gradient scale matrices are also bounded $\forall l, \mathbb{m}$ in the relevant time range $t \leq \bar{t}$. The last assertion follows from our initial assumption, that the support of the distribution of the weight matrices is compact, and additionally that the norm of each element is bounded by $c\sigma_n^2$ for fixed a constant c.[7]

We start by proving Thm 6, which is stated in Section 3.2.

Theorem 6. *Let $\boldsymbol{W}$ denote the compact representation of a deep linear network, where $\mathbb{m} = \min(m_1, ..., m_{L-1})$ denotes the size of its smallest hidden layer and $\mathbb{m} \geq \{m_0, m_L\}$. At each layer l, assume weight initialization $W_l^{(0)}$ obtained by sampling from a distribution with mean 0 and variance σ_l^2, normalized as specified in Def. 4. Let $\left(B_l^{(t)}(\mathbb{m})\right)_{\mathbb{m}=1}^{\infty}$ and $\left(A_l^{(t)}(\mathbb{m})\right)_{\mathbb{m}=1}^{\infty}$ denote two sequences of gradient scale matrices as defined in (2), where the $\mathbb{m}^{th}$ element of each series corresponds to a network whose smallest hidden layer has $\mathbb{m}$ neurons. Let $\xrightarrow{p}$ denote element-wise convergence in probability as $\mathbb{m} \to \infty$. Then $\forall t, l$:*

$$B_l^{(t)}(\mathbb{m}) \xrightarrow{p} I + O(\mu^2) \quad \forall l \in [1 \dots L-1], \qquad B_L^{(t)}(\mathbb{m}) \xrightarrow{p} \frac{m_L}{\mathbb{m}} I + O(\mu^2) \xrightarrow{p} O(\mu^2),$$

$$\text{var}[B_l^{(t)}(\mathbb{m})] = O(\mu^2) + O\left(\frac{1}{\mathbb{m}}\right) \qquad \forall l,$$

7. These assumptions can be relaxed to having bounds with high probability, which follow from the analysis of random matrices in Appendix A, but for simplicity we assume fixed bounds.

and

$$A_l^{(t)}(\mathbb{m}) \xrightarrow{p} I + O(\mu^2) \;\; \forall l \in [1 \dots L-1], \qquad A_0^{(t)}(\mathbb{m}) \xrightarrow{p} \frac{m_0}{\mathbb{m}} I + O(\mu^2) \xrightarrow{p} O(\mu^2),$$
$$\text{var}[A_l^{(t)}(\mathbb{m})] = O(\mu^2) + O\left(\frac{1}{\mathbb{m}}\right) \qquad \forall l.$$

Proof We will only work out the detailed proof of the first assertion concerning $B_l^{(t)}(\mathbb{m})$, as the proof of the second assertion is similar. Specifically, we prove by induction on time t a stronger claim:

$$\begin{aligned} &B_l^{(t)}(\mathbb{m}) \xrightarrow{p} I + O(\mu^2) \;\; \forall l \in [1 \dots L-1], \qquad B_L^{(t)}(\mathbb{m}) \xrightarrow{p} O(\mu^2) \\ &\text{var}[B_l^{(t)}(\mathbb{m})] = O(\mu^2) + O\left(\frac{1}{\mathbb{m}}\right) \qquad \forall l \\ &\mathbb{E}[\boldsymbol{W}^{(t)}(\mathbb{m})] = O(\mu) + O\left(\frac{1}{\mathbb{m}}\right). \end{aligned} \tag{31}$$

For the corresponding series $\{\boldsymbol{W}(\mathbb{m})\}$.

Base case $t = 0$**.** From (2)

$$B_l^{(0)} \coloneqq \Big(\prod_{j=l}^{1} W_j^{(0)}\Big)^\top \Big(\prod_{j=l}^{1} W_j^{(0)}\Big).$$

Recall that all weight matrices $\{W_l^{(0)}\}_{l=1}^L$ are initialized by sampling from a distribution with mean 0 and variance $\sigma_l^2 = O(\frac{1}{\mathbb{m}})$. In Appendix A.1 we prove some statistical properties of such matrices, and their corresponding *gradient scale matrices* $A_l^{(0)}$ and $B_l^{(0)}$. This analysis culminates in Corr 11 and Thm 12, which can now be used directly to infer that

$$\mathbb{E}(B_l^{(0)}) = I + O\left(\frac{1}{\mathbb{m}}\right) \;\; \forall l \in [1 \dots L-1], \qquad \mathbb{E}(B_L^{(0)}) = O\left(\frac{1}{\mathbb{m}}\right)$$
$$\text{var}[B_l^{(0)}(\mathbb{m})] = O\left(\frac{1}{\mathbb{m}}\right) \forall l.$$

Finally, Thm 13 further proves that such matrices also satisfy

$$B_l^{(0)}(\mathbb{m}) \xrightarrow{p} I \;\; \forall l \in [1 \dots L-1], \qquad B_L^{(0)}(\mathbb{m}) \xrightarrow{p} 0.$$

Similarly, from Lemma 18 we have that $\mathbb{E}[\boldsymbol{W}^{(0)}(\mathbb{m})] = 0 \;\forall \mathbb{m}$, and we therefore conclude that the assertion in (31) is true at $t = 0$.

Induction step. In Appendix A.2 we analyze random matrices which are defined similarly to the gradient scale matrices, and whose dynamics correspond to the update rule defined in (3). The main result is stated in Thm 15, which can be used directly now to show that if assertion (31) holds for $B_l^{(t)}$ and $\boldsymbol{W}^{(t)}(\mathbb{m})$, it also holds for $B_l^{(t+1)}$ and $\boldsymbol{W}^{(t+1)}(\mathbb{m})$.

First, let us verify that the conditions of Thm 15 are met. By construction, the dynamics captured by (16)-(18) describes the dynamics of $\{W_l\}$, a result that follows from the proof of

Prop. 5, where specifically (29)-(30) imply (16)-(17). In addition, as by assumption $\mathbb{m} \geq m_0$ and $m_0 = q$, $B_l^{(t)}(\mathbb{m}) \in \mathbb{R}^{q\times q}$ is full rank $\forall l, \mathbb{m}$ with probability 1. We may therefore use Thm 15 to conclude, based on the induction assumption, that

$$\Delta B_l^{(t)}(\mathbb{m}) \xrightarrow{p} O(\mu^2), \quad \text{var}[\Delta B_l^{(t)}(\mathbb{m})] = O(\mu^2) + O\left(\frac{1}{\mathbb{m}}\right) \qquad \forall l$$
$$\mathbb{E}[\Delta \boldsymbol{W}^{(t)}(\mathbb{m})] = O(\mu) + O\left(\frac{1}{\mathbb{m}}\right).$$

Noting that $B_l^{(t+1)}(\mathbb{m}) = B_l^{(t)}(\mathbb{m}) + \Delta B^{(t)}(\mathbb{m})\ \forall l$ and $\boldsymbol{W}^{(t+1)}(\mathbb{m}) = \boldsymbol{W}^{(t)}(\mathbb{m}) + \Delta \boldsymbol{W}_l^{(t)}(\mathbb{m})$, the assertion in (31) follows for all t. ■

We proceed to prove Thm 7, which is stated in Section 3.3.

Theorem 7. *Let $\boldsymbol{w}_j^{(t)}$ denote the j^{th} column of the compact representation matrix $\boldsymbol{W}^{(t)}$, and $\boldsymbol{w}_j^{opt}$ the j^{th} column of the optimal solution of (1). Assume that the data is rotated to its principal coordinate system, and let d_j denote the j^{th} singular value of the data. Then there exists $\hat{t}$ such that $\forall \delta, \varepsilon$ and $\forall t \leq \hat{t}$, $\exists \hat{\mathbb{m}}, \hat{\mu}$ such that $\forall \mu < \hat{\mu}, \mathbb{m} \geq \hat{\mathbb{m}}$*

$$Prob\left(\left\|\boldsymbol{w}_j^{(t+1)} - \left[\lambda_j^t \boldsymbol{w}_j^{(0)} + [1-\lambda_j^t]\boldsymbol{w}_j^{opt}\right]\right\| < \varepsilon\right) > (1-\delta), \qquad \lambda_j = 1 - \mu d_j L.$$

Proof First let us derive a bound on the total gradient magnitude $\|G_r^{(t)}\|$. Since by assumption the norm of the data $\|X\|$ and the norms of $W_l^{(0)}\ \forall l$ are bounded, it follows that the loss at time $t = 0$ is also bounded. Let U_1 denote this bound, and U_2 denote the data bound $\|X\|^2 \leq U_2$. Let $L^{(t)}(\mathbb{X})$ denote the loss at time t. Since the loss is decreasing, it follows that $L^{(t)}(X, Y) \leq U_1\ \forall t$, and therefore

$$\|G_r^{(t)}\|^2 = \|[\boldsymbol{W}^{(t)}X - Y]X^\top\|^2 \leq 2L^{(t)}(\mathbb{X})\|X\|^2 \leq 2U_1U_2.$$

We now introduce the notation $B_l^{(t)} = I + O(\mu^2) + \Delta_{Bl}^{(t)}$ and $A_l^{(t)} = I + O(\mu^2) + \Delta_{Al}^{(t)}$. Let U_3 denote a tight bound so that $\{\|\Delta_{Al}^{(t)}\|^2, \|\Delta_{Bl}^{(t)}\|^2\} \leq U_3\ \forall l, t \leq \bar{t}$, where we further assume that $U_3 \leq 1$ (this last assumption will be justified later). Starting from (3)

$$\begin{aligned}
\boldsymbol{W}^{(t+1)} =& \boldsymbol{W}^{(t)} - \mu \sum_{l=1}^{L} A_l^{(t)} \cdot G_r^{(t)} \cdot B_{l-1}^{(t)} + O(\mu^2) \\
=& \boldsymbol{W}^{(t)} - \mu \sum_{l=1}^{L} [I + O(\mu^2) + \Delta_{Al}^{(t)}]\ G_r^{(t)}\ [I + O(\mu^2) + \Delta_{Bl}^{(t)}] + O(\mu^2) \qquad (32) \\
=& \boldsymbol{W}^{(t)} - \mu \left[LG_r^{(t)} + \left(\sum_{l=1}^{L} \Delta_{Al}^{(t)}\right) G_r^{(t)} + G_r^{(t)} \sum_{l=1}^{L} \Delta_{Bl}^{(t)} + \sum_{l=1}^{L} \Delta_{Al}^{(t)}\ G_r^{(t)}\ \Delta_{Bl}^{(t)} \right] + O(\mu^2).
\end{aligned}$$

The last transition is valid because the norms of $G_r^{(t)}, \Delta_{Al}^{(t)}, \Delta_{Bl}^{(t)}$ are bounded. It follows that

$$\|\boldsymbol{W}^{(t+1)} - [\boldsymbol{W}^{(t)} - \mu L G_r^{(t)}]\|^2 \leq \mu L 2 U_1 U_2 [U_3 + U_3 + U_3^2] + O(\mu^2) \leq \mu 6 L U_1 U_2 U_3 + O(\mu^2).$$

Thus $\exists\hat{\mu}$ such that $\forall\mu<\hat{\mu}$

$$\|\boldsymbol{W}^{(t+1)}-[\boldsymbol{W}^{(t)}-\mu LG_r^{(t)}]\|^2\leq\hat{\mu}12LU_1U_2U_3. \tag{33}$$

From Thm 6, $\forall t$ and $\forall l\leq L-1$, $B_l^{(t)}(\mathbb{m})\xrightarrow{p}I+O(\mu^2)$ and $A_l^{(t)}(\mathbb{m})\xrightarrow{p}I+O(\mu^2)$. In other words, if we fix $\hat{t}$ and consider all the iterations leading to $\hat{t}$, then $\forall\varepsilon,\delta>0$ $\exists\hat{\mathbb{m}}\in\mathbb{N}$, such that $\forall l,\mathbb{m}>\hat{\mathbb{m}},t\leq\hat{t}$

$$Prob\left(\left|B_l^{(t)}(\mathbb{m})-[I+O(\mu^2)]\right|^2\leq\min\left\{\frac{\varepsilon}{\hat{\mu}12LU_1U_2},1\right\}\right)>(1-\delta)\qquad\text{element-wise}$$

$$Prob\left(\left|A_l^{(t)}(\mathbb{m})-[I+O(\mu^2)]\right|^2\leq\min\left\{\frac{\varepsilon}{\hat{\mu}12LU_1U_2},1\right\}\right)>(1-\delta)\qquad\text{element-wise}$$

$$\implies\qquad Prob\left(U_3\leq\min\left\{\frac{\varepsilon}{\hat{\mu}12LU_1U_2},1\right\}\right)>(1-\delta).\quad(U_3\text{ is tight})$$

Finally, from (33) and $\forall\mu<\hat{\mu}$

$$Prob\left(\|\boldsymbol{W}^{(t+1)}-[\boldsymbol{W}^{(t)}-\mu LG_r^{(t)}]\|^2\leq\varepsilon\right)>(1-\delta). \tag{34}$$

Next, we evaluate

$$\tilde{\boldsymbol{W}}^{(t+1)}=\boldsymbol{W}^{(t)}-\mu LG_r^{(t)}. \tag{35}$$

We first shift to the principal coordinate system defined in Def 1. In this representation $G_r^{(t)}=W^{(t)}D-M$, where $D=diag(\{d_j\}_{j=1}^q)$ is a diagonal matrix whose elements are the principal eigenvalues of the data $\{d_j\}_{j=1}^q$, arranged in decreasing order. Since D is diagonal, (35) can be written separately for each column of $\tilde{\boldsymbol{W}}^{(t)}$, and we get for each column $\boldsymbol{w}_j^{(t)}$

$$\tilde{\boldsymbol{w}}_j^{(t+1)}=\boldsymbol{w}_j^{(t)}-\mu L[\boldsymbol{w}_j^{(t)}d_j-\boldsymbol{m}_j],\qquad j\in K.$$

This is a telescoping series, whose solution is

$$\begin{aligned}\tilde{\boldsymbol{w}}_j^{(t+1)}&=(1-\mu Ld_j)\boldsymbol{w}_j^{(t)}+\mu L\boldsymbol{m}_j=\dots\\&=(1-\mu Ld_j)^t\boldsymbol{w}_j^{(0)}+\mu Ld_j\left[\sum_{\nu=1}^{t}(1-\mu Ld_j)^{\nu-1}\right]\frac{\boldsymbol{m}_j}{d_j}\\&=\lambda_j^t\boldsymbol{w}_j^{(0)}+[1-\lambda_j^t]\frac{\boldsymbol{m}_j}{d_j},\qquad\lambda_j=1-\mu Ld_j.\end{aligned}$$

As $\boldsymbol{w}_j^{opt}=\frac{\boldsymbol{m}_j}{d_j}$ and using (34), the assertion in the theorem follows.

■

We conclude by proving Thm 8. To this end we introduce another notation, $A^{(t)}=\sum_{l=1}^{L}A_l^{(t)}$, and let U_4 denote the bound on A_l where $\|A_l^{(t)}\|^2\leq U_4\ \ \forall l,t\leq\bar{t}$.

Theorem 8. *Let $\boldsymbol{w}_j^{(t)}$ denote the j^{th} column of the compact representation matrix $\boldsymbol{W}^{(t)}$, and $\boldsymbol{w}_j^{opt}$ the j^{th} column of the optimal solution of (1). Assume that the data is rotated to its*

principal coordinate system, and let d_j denote the j^{th} singular value of the data. Then there exists $\breve{t}$ such that $\forall \delta, \varepsilon$ and $\forall t \le \breve{t}$, $\exists \breve{\mathrm{m}}, \breve{\mu}$ such that $\forall \mu < \breve{\mu}, \mathrm{m} \ge \breve{\mathrm{m}}$

$$Prob\Bigg(\Big\|\boldsymbol{w}_j^{(t+1)} - \Big[\prod_{t'=1}^{t}(I-\mu d_j A^{(t')})\boldsymbol{w}_j^{(0)} + \mu d_j\Big(\sum_{t'=1}^{t}\prod_{t''=t'+1}^{t}(I-\mu d_j A^{(t'')})A^{(t')}\Big)\boldsymbol{w}_j^{opt}\Big]\Big\| < \varepsilon\Bigg) > (1-\delta).$$

Proof We use the same bound notations as in the proof of Thm 7, but where only $\|\Delta_{Bl}^{(t)}\|^2 \le U_3 \ \forall l, t \le \bar{t}$ a tight bound, $U_3 \le 1$. Similarly to (32),

$$\begin{aligned} & \boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \mu\left[\Big(\sum_{l=1}^{L} A_l^{(t)}\Big)G_r^{(t)} + \sum_{l=1}^{L} A_l^{(t)}\ G_r^{(t)}\ \Delta_{Bl}^{(t)}\right] + O(\mu^2) \\ \implies \quad & \|\boldsymbol{W}^{(t+1)} - [\boldsymbol{W}^{(t)} - \mu A^{(t)} G_r^{(t)}]\|^2 \le \mu 2LU_1U_2U_4 + O(\mu^2). \end{aligned}$$

From Thm 6, $\forall \varepsilon, \delta > 0\ \exists \breve{\mathrm{m}}, \breve{t}, \breve{\mu}$, such that $\forall \mathrm{m} \ge \breve{\mathrm{m}}, t \le \breve{t}, \mu \le \breve{\mu}$

$$\begin{aligned} & Prob\left(U_3 \le \min\left\{\frac{\varepsilon}{\breve{\mu} 4LU_1U_2U_4}, 1\right\}\right) > (1-\delta) \\ \implies \quad & Prob\left(\|\boldsymbol{W}^{(t+1)} - [\boldsymbol{W}^{(t)} - \mu A^{(t)} G_r^{(t)}]\|^2 \le \varepsilon\right) > (1-\delta). \end{aligned} \tag{36}$$

As before, we evaluate

$$\tilde{\boldsymbol{W}}^{(t+1)} = \boldsymbol{W}^{(t)} - \mu A^{(t)} G_r^{(t)}.$$

In the principal coordinate system this expression is separable by columns, thus

$$\tilde{\boldsymbol{w}}_j^{(t+1)} = \boldsymbol{w}_j^{(t)} - \mu \sum_{l=1}^{L} A_l^{(t)}(d_j \boldsymbol{w}_j^{(t)} - \boldsymbol{m}_j), \qquad j \in [K].$$

Once again we have a telescoping series, whose solution is

$$\tilde{\boldsymbol{w}}_j^{(t+1)} = \prod_{t'=1}^{t}\big(I-\mu d_j A^{(t')}\big)\boldsymbol{w}_j^{(0)} + \mu d_j\Big[\sum_{t'=1}^{t}\prod_{t''=t'+1}^{t}\big(I-\mu d_j A^{(t'')}\big)A^{(t')}\Big]\frac{\boldsymbol{m}_j}{d_j}.$$

As $\boldsymbol{w}_j^{opt} = \frac{\boldsymbol{m}_j}{d_j}$ and using (36), the assertion in the theorem follows. ■

B.3 Adding Non-Linear ReLU Activation

In this section, we analyze the two-layer model with ReLU activation, where only the weights of the first layer are being learned (Arora et al., 2019b). Similarly to (1), the loss is defined as

$$W^* = \underset{W}{\operatorname{argmin}} \frac{1}{2}\sum_{i=1}^{n}\|f(\boldsymbol{x}_i) - \boldsymbol{y}_i\|^2, \qquad f(\boldsymbol{x}_i) = \boldsymbol{a}^\intercal \cdot \sigma(W\boldsymbol{x}_i), \quad \boldsymbol{a} \in \mathbb{R}^{\mathrm{m}},\ W \in \mathbb{R}^{\mathrm{m}\times d}.$$

where $\mathbb{m}$ denotes the number of neurons in the hidden layer and $\boldsymbol{a}$ a fixed vector. We consider a binary classification problem with 2 classes, where $y_i = 1$ for $\boldsymbol{x}_i \in C_1$, and $y_i = -1$ for $\boldsymbol{x}_i \in C_2$. $\sigma(\cdot)$ denotes the ReLU activation function applied element-wise to vectors, where $\sigma(u) = u$ if $u \geq 0$, and 0 otherwise.

At time t, each gradient step is defined by differentiating the loss with respect to W. Due to the non-linear nature of the activation function $\sigma(\cdot)$, we separately[8] differentiate each row of W, denoted $\boldsymbol{w}_r$ where $r \in [\mathbb{m}]$, as follows:

$$\begin{aligned}
\boldsymbol{w}_r^{(t+1)} - \boldsymbol{w}_r^{(t)} &= -\mu \frac{\partial L(\mathbb{X})}{\partial \boldsymbol{w}_r}\Big|_{\boldsymbol{w}_r=\boldsymbol{w}_r^{(t)}} = -\mu \sum_{i=1}^{n} \Big[\boldsymbol{a}^\intercal \cdot \sigma(W^{(t)}\boldsymbol{x}_i) - y_i\Big] \frac{\partial f(\boldsymbol{x}_i)}{\partial \boldsymbol{w}_r}\Big|_{\boldsymbol{w}_r=\boldsymbol{w}_r^{(t)}} \\
&= -\mu \sum_{i=1}^{n} \Big[\sum_{j=1}^{\mathbb{m}} a_j \sigma(\boldsymbol{w}_j^{(t)} \cdot \boldsymbol{x}_i) - y_i\Big] a_r \boldsymbol{x}_i^\intercal \mathbb{1}_{\boldsymbol{w}_r^{(t)}}(\boldsymbol{x}_i) \\
&= -\mu a_r \sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{w}_r^{(t)}}(\boldsymbol{x}_i)\Big[\Psi^{(t)}(\boldsymbol{x}_i) \cdot \boldsymbol{x}_i - y_i\Big] \boldsymbol{x}_i^\intercal, \qquad \text{where } \Psi^{(t)}(\boldsymbol{x}_i) = \sum_{j=1}^{\mathbb{m}} a_j \boldsymbol{w}_j^{(t)} \mathbb{1}_{\boldsymbol{w}_j^{(t)}}(\boldsymbol{x}_i).
\end{aligned}$$

Above $\mathbb{1}_{\boldsymbol{w}_r^{(t)}}(\boldsymbol{x}_i)$ denotes the indicator function that equals 1 when $\boldsymbol{w}_r^{(t)} \cdot \boldsymbol{x}_i \geq 0$, and 0 otherwise.

To proceed, we make two assumptions:

1. Each point $\boldsymbol{x}_i$ is drawn from a symmetric distribution $\mathcal{D}$ with density $f_{\mathcal{D}}(\mathbf{X})$, such that: $f_{\mathcal{D}}(\boldsymbol{x}_i) = f_{\mathcal{D}}(-\boldsymbol{x}_i)$.
2. W and $\boldsymbol{a}$ are initialized so that $\boldsymbol{w}_{2i}^{(0)} = -\boldsymbol{w}_{2i-1}^{(0)}$ and $a_{2i} = -a_{2i-1}$ $\forall i \in [\frac{\mathbb{m}}{2}]$.

Theorem 9. Retaining the assumptions stated above, at the beginning of the learning, the temporal dynamics of the model can be shown to obey the following update rule:

$$W^{(t+1)} \approx W^{(t)} - \mu \frac{1}{2}\Big[(\boldsymbol{a}\boldsymbol{a}^\intercal) W^{(t)} \Sigma_{XX} - \tilde{M}^{(t)}\Big].$$

Above $\tilde{M}^{(t)}$ denotes the difference between the centroids of the 2 classes, computed in the half-space defined by $\boldsymbol{w}_r^{(t)} \cdot \boldsymbol{x} \geq 0$.

Proof It follows from Assumption 2 that at the beginning of training $\mathbb{1}_{\boldsymbol{w}_{2j}^{(0)}}(\boldsymbol{x}_i) + \mathbb{1}_{\boldsymbol{w}_{2j-1}^{(0)}}(\boldsymbol{x}_i) = 1$, $\forall \boldsymbol{x}_i$ such that $\boldsymbol{w}_{2j-1}\boldsymbol{x}_i \neq \boldsymbol{w}_{2j}\boldsymbol{x}_i \neq 0$, and $\forall j \in [\frac{\mathbb{m}}{2}]$. Consequently

$$\Psi^{(0)}(\boldsymbol{x}_i) = \sum_{j=1}^{\mathbb{m}} a_j \boldsymbol{w}_j^{(0)} \mathbb{1}_{\boldsymbol{w}_j^{(t)}}(\boldsymbol{x}_i) = \frac{1}{2} \sum_{j=1}^{\mathbb{m}} a_j \boldsymbol{w}_j^{(0)} = \frac{1}{2} \boldsymbol{a}^\intercal W^{(0)}.$$

$\forall \boldsymbol{x}_i$ such that $\boldsymbol{w}_{2j-1}\boldsymbol{x}_i \neq \boldsymbol{w}_{2j}\boldsymbol{x}_i \neq 0$. Finally

$$\boldsymbol{w}_r^{(1)} - \boldsymbol{w}_r^{(0)} = -\mu a_r \big[\frac{1}{2} \boldsymbol{a}^\intercal W^{(0)} \sum_{\substack{i=1 \\ \boldsymbol{w}_r^{(0)}\boldsymbol{x}_i \geq 0}}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\intercal - \sum_{\substack{i=1 \\ \boldsymbol{w}_r^{(0)}\boldsymbol{x}_i \geq 0}}^{n} y_i \boldsymbol{x}_i^\intercal\big].$$

8. Since the ReLU function is not everywhere differentiable, the following may be considered the definition of the update rule.

Next, we note that Assumption 1 implies

$$\mathbb{E}[\sum_{\substack{i=1\\ \boldsymbol{w}\cdot\boldsymbol{x}_i\geq 0}}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal]=\frac{1}{2}\mathbb{E}[\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal]=\frac{1}{2}\mathbb{E}[\Sigma_{XX}].$$

for any vector $\boldsymbol{w}$. Thus, if the sample size n is large enough, at the beginning of training we expect to see

$$\boldsymbol{w}_r^{(t+1)}-\boldsymbol{w}_r^{(t)}\approx-\mu\frac{a_r}{2}[\boldsymbol{a}^\intercal W^{(t)}\Sigma_{XX}-\tilde{\boldsymbol{m}}_r^{(t)}],\quad\forall r.$$

where row vector $\tilde{\boldsymbol{m}}_r^{(t)}$ denotes the vector difference between the centroids of classes C_1 and C_2, computed in the half-space defined by $\boldsymbol{w}_r^{(t)}\cdot\boldsymbol{x}\geq 0$. Finally (for small t)

$$W^{(t+1)}-W^{(t)}\approx-\mu\frac{1}{2}\Big[(\boldsymbol{a}\boldsymbol{a}^\intercal)W^{(t)}\Sigma_{XX}-\tilde{M}^{(t)}\Big].$$

where $\tilde{M}^{(t)}$ denotes the matrix whose r-th row is $a_r\tilde{\boldsymbol{m}}_r^{(t)}$. This equation is reminiscent of the single-layer linear model dynamics $\boldsymbol{W}^{(t+1)}=\boldsymbol{W}^{(t)}-\mu G_r^{(t)}$, and we may conclude that when it holds and using the principal coordinate system, the rate of convergence of the j-th column of $W^{(t)}$ is governed by the singular value d_j . ■

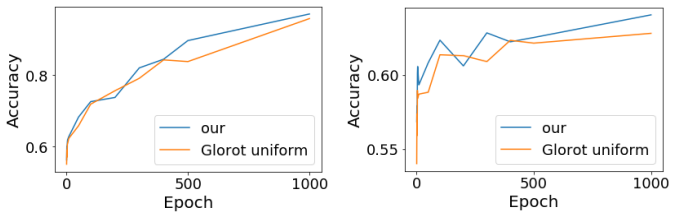
Appendix C. Additional Empirical Results

C.1 Weight Initialization

We evaluate empirically the weight initialization scheme from Def. 4 and (12). When compared to Glorot uniform initialization, the only difference between the two schemes lies in how the first and last layers are scaled. Thus, to highlight the difference between the methods, we analyze a fully connected linear network with a single hidden layer, whose dimension (the number of hidden neurons) is much larger than the input and output dimensions. We trained N=10 such networks on a binary classification problem, once with the initialization suggested in (12), and again with Glorot uniform initialization. While both initialization schemes achieve the same final accuracy upon convergence, our proposed initialization variant converges faster on both train and test datasets (see Fig. 11).

C.2 Divergence of Gradient Scale Matrices

We now discuss some additional results, complementary to Section 3.3. In Fig. 1, we visualize how the gradient scale matrices $B_l^{(t)}(\mathbb{m})$ remain approximately equal to I much longer than $A_l^{(t)}(\mathbb{m})$, for a specific layer in a 5-layered model. In Fig. 13 we show a similar trend in all 5 layers of the network. Note that in the input and output layers, $B_l^{(t)}(\mathbb{m})$ and $A_l^{(t)}(\mathbb{m})$ are I by definition, and hence their convergence is self-evident.



(a) train (b) test

Figure 11: Learning curves of a fully connected linear network with one hidden layer, trained on the dogs and cats dataset, and initialized by either Glorot uniform initialization (orange), or the initialization proposed in (12) (blue).

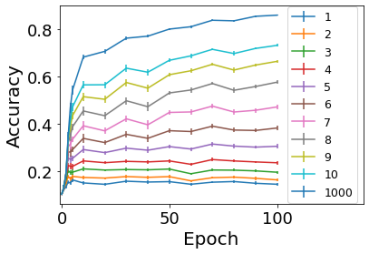
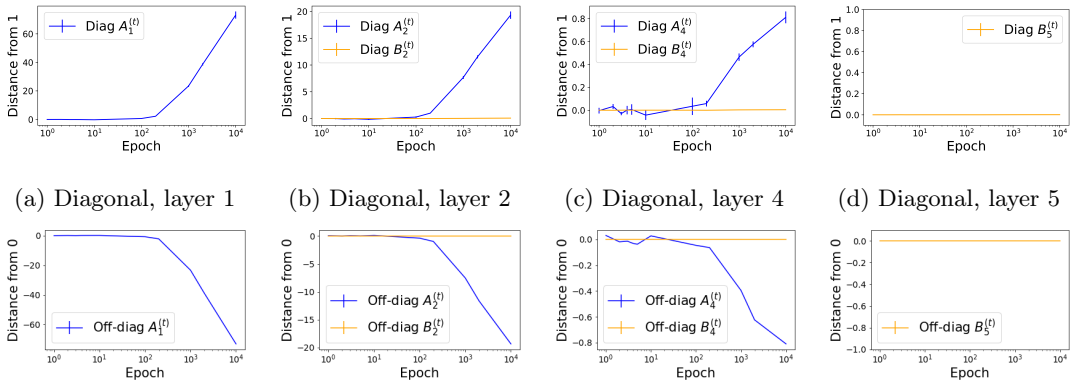


Figure 12: Evaluations on test-sets projected to the first P principal components, for different values of P (see legend) of 10 VGG-19 models trained on CIFAR-10



(a) Diagonal, layer 1 (b) Diagonal, layer 2 (c) Diagonal, layer 4 (d) Diagonal, layer 5

(e) Off-diagonal, layer 1 (f) Off-diagonal, layer 2 (g) Off-diagonal, layer 4 (h) Off-diagonal, layer 5

Figure 13: Similar to Fig. 1, showing the dynamics of $A_l^{(t)}$ and $B_l^{(t)}$ in additional layers, when training 10 5-layered linear networks on the small mammals dataset. (a-d) The empirical L_2-distance of the diagonal elements of $A_3^{(t)}$ and $B_3^{(t)}$ from their analytical value of $\alpha_l^{(t)}$ and $\beta_l^{(t)}$—$diag\left(A_l^{(t)} - \alpha_l^{(t)} I\right)$ and $diag\left(B_l^{(t)} - \beta_l^{(t)} I\right)$—respectively. (e-h) The empirical L_2-distance of the off-diagonal elements of $A_l^{(t)}$ and $B_l^{(t)}$ from 0. The networks reach their maximal test accuracy in epoch $t = 100$, before the divergence of $A_l^{(t)}$.

C.3 Empirical Validation of the PC-bias on Original Model

C.3.1 Models with L_2 Loss

In Section 4, we relaxed some of the theoretical assumptions while pursuing the empirical investigation, in order to match more commonly used models. Specifically, we changed the initialization to the commonly used Glorot initialization, replaced the L_2 loss with the cross-entropy loss, and employed SGD instead of the deterministic GD. As can be expected from the theoretical results, without this relaxation the PC-bias becomes even more pronounced. To support this claim, we repeated the experiments of Section 4.2 and Fig. 4 with the assumptions of the theoretical analysis, showing the results in Fig. 14.

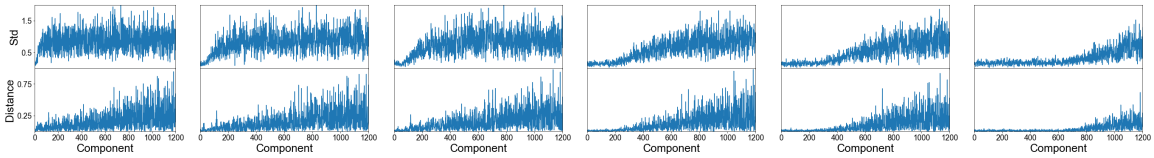


Figure 14: Same as Fig. 4, using the L_2 loss and the initialization scheme proposed in Def. 4. Epochs plotted: $1, 5, 10, 50, 100, 500$.

C.3.2 Distance of Convergence in Each Principal Direction

In Figs. 4, 14, we see that weights in directions corresponding to larger principal components converge faster, in the sense that the std across different repetitions drops to zero faster along these directions. However, the optimal solution in each direction is drastically different. Directions corresponding to larger principal components tend to have lower values in their optimal solution. This could serve as a possible explanation for the convergence phenomenon—it might be possible that in all directions the distance of the current solution drops at the same speed, but as directions corresponding to higher principal values start nearer to the optimal solution, they seem to converge faster.

To test this hypothesis, we repeated the experiment from the previous section (Fig. 14), and computed the distance in each direction from the optimal solution, normalized by the maximal difference achieved in each direction. We plot these results in Fig. 15. As suggested by our theory, we see that the normalized distance in each direction also drops faster in directions corresponding to larger principal components.

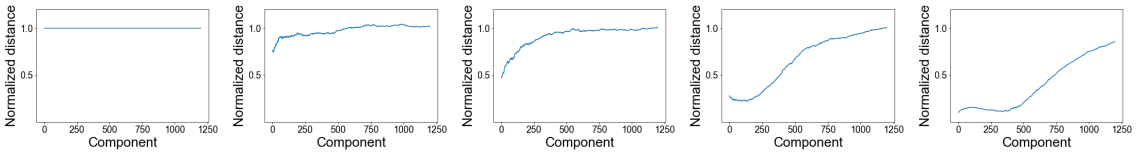


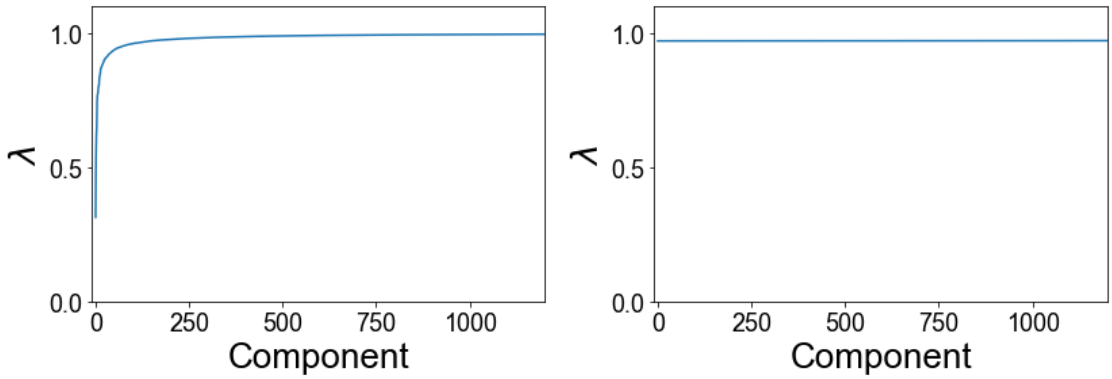
Figure 15: Rate of convergence along the principal directions in different epochs. The value of the X-axis corresponds to the index of a principal eigenvalue, from the most significant to the least significant. The value of the Y-axis is the L_2 distance between the mean value of the weights and the optimal solution, normalized by the maximum distance across all epochs. The results are plotted for epochs: $1, 2, 10, 50, 200$, for 10 5-layered linear networks trained on the cats and dogs dataset with the L_2 loss.

C.4 Projection to Higher PC's

In Section 4.3 we described an evaluation methodology, based on the creation of a modified *test-set* by projecting each test example on the span of the first P principal components. We repeat this experiment with VGG-19 networks on CIFAR-10, and plot the results in Fig. 12.

C.5 Spectral Bias

The *spectral bias*, discussed in Section 5.3, can also induce a similar learning order in different networks. To support the discussion in Section 5.3, we analyze the relation between the *spectral bias* and *accessibility*, in order to clarify its relation to the *Learning Order Constancy*



(a) λ_j for Fig. 4a

(b) λ_j for Fig. 4c

Figure 16: In the same experimental setup as in Fig. 4, we show the evolution of λ_j with epochs.

and the *PC-bias* (§C.5.2) . First, however, we expand the scope of the empirical evidence for this effect to the classification scenario and real image data (§C.5.1).

C.5.1 Spectral Bias in Classification

Rahaman et al. (2019) showed that when regressing a 2D function by a neural network, the model seems to approximate the lower frequencies of the function before its higher frequencies. Here we extend this empirical observation to the classification framework. Thus, given frequencies $\kappa = (\kappa_1, \kappa_2, ..., \kappa_m)$ with corresponding phases $\boldsymbol{\varphi} = (\varphi_1, \varphi_2, ..., \varphi_m)$, we consider the mapping $\lambda : [-1, 1] \to \mathbb{R}$ given by

$$\lambda(z) = \sum_{i=1}^{m} \sin(2\pi\kappa_i z + \varphi_i) := \sum_{i=1}^{m} freq_i(z). \tag{37}$$

Above κ is strictly monotonically increasing, while $\boldsymbol{\varphi}$ is sampled uniformly.

The classification rule is defined by $\lambda(z) \lessgtr 0$. We created a binary dataset whose points are fully separated by $\lambda(z)$, henceforth called the *frequency dataset* (see the visualization in Fig. 18 and details in §D.4). When training on this dataset, we observe that the frequency of the corresponding separator increases as learning proceeds, in agreement with the results of Rahaman et al. (2019).

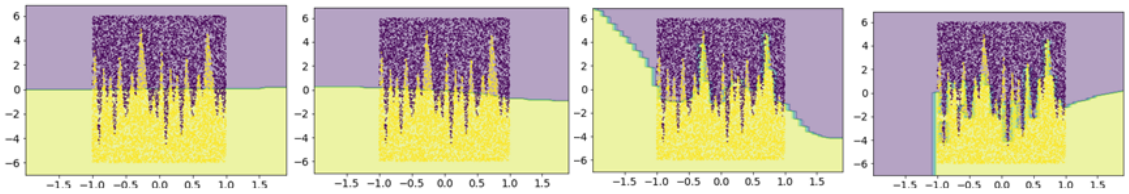


Figure 17: Visualization of the separator learned by st-VGG when trained on the frequency dataset, as captured in advancing epochs (from left to right): 1, 100, 1000, 10000. Each point represents a training example (yellow for one class and purple for the other). The background color represents the classification that the network predicts for points in that region.

To visualize the decision boundary of an st-VGG network trained on this dataset as it evolves with time, we trained N=100 st-VGG networks. Since the data lies in $\mathbb{R}^2$, we can visualize it and the corresponding network's inter-class boundary at each epoch as shown in Fig. 17. We can see that the decision boundary incorporates low frequencies at the beginning of the learning, adding the higher frequencies only later on. The same qualitative results are achieved with other instances of st-VGG as well. We note that while the decision functions are very similar in the region where the training data is, at points outside this region they differ drastically across networks.

C.5.2 Spectral Bias: Relation to Accessibility

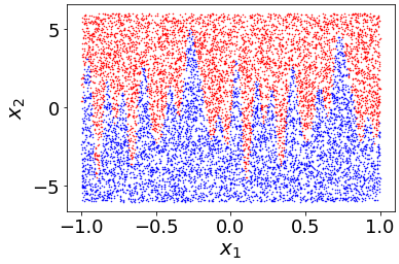
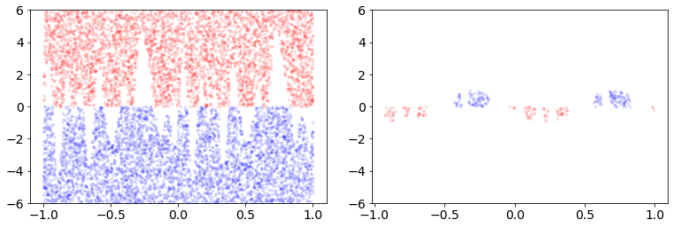


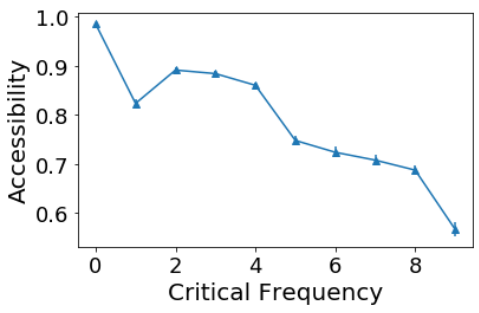
Figure 18: Visualization of the classification dataset used to extend Rahaman et al. (2019) to a classification framework.



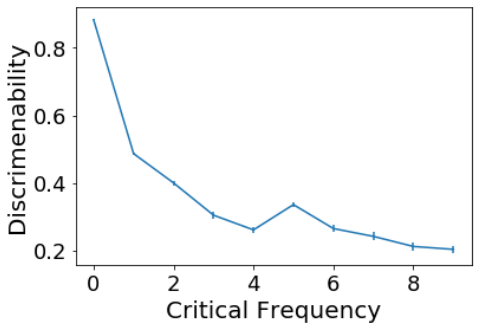
(a) 2 classes freq 0 (b) 2 classes freq 1

Figure 19: Visualization of the *critical frequency*, showing all the points in the 2D-frequency dataset with *critical frequency* of (a) 0, and (b) 1.

To connect between the learning order, which is defined over examples, and the Fourier analysis of classifiers, we define for each example its *critical frequency*, which characterizes the smallest number of frequencies needed to correctly classify the example. To illustrate, consider the *frequency dataset* defined above. Here, the *critical frequency* is defined as the smallest $j \in [m]$ such that $\lambda_j(z) = \sum_{i=1}^{j} freq_i(z)$ classifies the example correctly (see Fig. 19).



(a)



(b)

Figure 20: (a) Correlation between *critical frequency* and *accessibility* score in the 2D-frequency dataset. (b) Correlation between *discriminability* and *critical frequency* in the 2D-frequency dataset.

In this binary classification task, we observe a strong connection between the order of learning and the *critical frequency*. Specifically, we trained N=100 st-VGG networks on

the *frequency dataset*, and correlated the *accessibility* scores with the *critical frequency* of the examples (see Fig. 20a). We see a strong negative correlation ($r = -0.93$, $p < 10^{-2}$), suggesting that examples whose *critical frequency* is high are learned last by the networks.

To see the effect of the *spectral bias* in real classification tasks and extend the above analysis to natural images, we need to define a score that captures the notion of *critical frequency*. To this end, we define the *discriminability* measure of an example as the percentage out of its k neighbors that share the same class as the example. Intuitively, an example has a low *discriminability* score when it is surrounded by examples from other classes, which forces the learned boundary to incorporate high frequencies. In Fig. 20b we plot the correlation between the *discriminability* and the *critical frequency* for the 2D frequency dataset. The high correlation (r=-0.8, $p < 10^{-2}$) indicates that *discriminability* indeed captures the notion of *critical frequency*.

Discussion. We find that the *spectral bias* is connected to the LOC effect. Examples that can be classified correctly by a low-frequency function are also learned faster by the neural network. However, as the definition of such frequency is not trivial in higher dimensions, it is yet unclear if this result can be extended to image classification.

Appendix D. Methodology

D.1 Implementation Details and Hyperparameters

The results reported in Section 6represent the mean performance of 100 st-VGG and linear st-VGG networks, trained on the small mammals dataset. The results reported in Section 6 represent the mean performance of 10 two-layered fully connected linear networks trained over the cats and dogs dataset. The results in Fig. 10 represent the mean performance of the 100 st-VGG network trained on the small mammals dataset. In every experimental setup, the network's hyper-parameters were coarsely grid-searched to achieve good performance over a validation set, for a fair comparison. Other hyper-parameters exhibit similar results.

D.2 Generalization Gap

In Section 6 we discuss the evaluation of networks on datasets with amplified principal components. Examples of these images are shown in Fig. 7: the top row shows examples of the original images, the middle row shows what happens to each image when its 1.5% most significant principal components are amplified, and the bottom row shows what happens when its 1.5% least significant principal components are amplified. Amplification involved a factor of 10, which is significantly smaller than the ratio between the values of the first and last principal components of the data. After amplification, all the images were re-normalized to have 0 mean and std 1 in every channel as customary.

D.3 Architectures

st-VGG. A stripped version of VGG which we used in many of the experiments. It is a convolutional neural network, containing 8 convolutional layers with 32, 32, 64, 64, 128, 128, 256, 256 filters respectively. The first 6 layers have filters of size 3×3, and the last 2 layers have filters of size 2×2. Every other layer is followed by a 2×2 max-pooling layer and a

0.25 dropout layer. After the convolutional layers, the units are flattened, and there is a fully connected layer with 512 units followed by a 0.5 dropout. The batch size we used was 100. The output layer is a fully-connected layer with output units matching the number of classes in the dataset, followed by a softmax layer. We trained the network using the SGD optimizer, with cross-entropy loss. When training st-VGG, we used a learning rate of 0.05.

Linear st-VGG. A linear version of the st-VGG network. In linear st-VGG, we change the activation function to the identity function, and replace max-pooling with average pooling with a similar stride. Similar to the non-linear case, a cross-entropy loss is being used. The network does not contain a softmax layer.

Linear fully connected network. An L-layered fully connected network. Each layer contains 1024 weights, initialized with Glorot uniform initialization. 0.5 dropout is used before the output layer. Networks are trained with an SGD optimizer, without momentum or L_2 regularization. Unless stated otherwise, a cross-entropy loss is being used. The network does not contain a softmax layer.

D.4 Datasets

In all the experiments and all the datasets, the data was always normalized to have 0 mean and std 1, in each channel separately.

Small Mammals. The small mammals dataset used in our experiments is the relevant super-class of the CIFAR-100 dataset. It contains 2500 train images divided into 5 classes equally, and 500 test images. Each image is of size $32 \times 32 \times 3$. This dataset was chosen due to its small size.

Cats and Dogs. The cats and dogs dataset is a subset of CIFAR-10. It uses only the 2 relevant classes, to create a binary problem. Each image is of size $32 \times 32 \times 3$. The dataset is divided to 20000 train images (10000 per class) and 2000 test images (1000 per class). This dataset is used when a binary problem is required.

ImageNet-20. The ImageNet-20 dataset (Russakovsky et al., 2015) is a subset of ImageNet containing 20 classes. This data resembles ImageNet in terms of image resolution and data variability, but contains a smaller number of examples to reduce computation time. The dataset contains 26000 train images (1300 per class) and 1000 test images (50 per class). The choice of the 20 classes was arbitrary, and contained the following classes: boa constrictor, jellyfish, American lobster, little blue heron, Shih-Tzu, scotch terrier, Chesapeake Bay retriever, komondor, snow leopard, tiger, long-horned beetle, warthog, cab, holster, remote control, toilet seat, pretzel, fig, burrito and toilet tissue.

Frequency dataset A binary 2D dataset, is used in Section 5.3, to examine the effects of spectral bias in classification. The data is define by the mapping $\lambda : [-1, 1] \to \mathbb{R}$ given in (37) by

$$\lambda(z) = \sum_{i=1}^{m} \sin(2\pi\kappa_i z + \varphi_i) := \sum_{i=1}^{m} freq_i(z),$$

with frequencies $\kappa = (\kappa_1, \kappa_2, ..., \kappa_m)$ and corresponding phases $\boldsymbol{\varphi} = (\varphi_1, \varphi_2, ..., \varphi_m)$. The classification rule is defined by $\lambda(z) \lessgtr 0$.

In our experiments, we chose $m = 10$, with frequencies $\kappa_1 = 0, \kappa_2 = 1, \kappa_3 = 2, ..., \kappa_{10} = 9$. Other choices of m yielded similar qualitative results. The phases were chosen randomly between 0 and 2π, and were set to be: $\varphi_1 = 0$, $\varphi_2 = 3.46$, $\varphi_3 = 5.08$, $\varphi_4 = 0.45$, $\varphi_5 = 2.10$, $\varphi_6 = 1.4$, $\varphi_7 = 5.36$, $\varphi_8 = 0.85$, $\varphi_9 = 5.9$, $\varphi_{10} = 5.16$. As the first frequency is $\kappa_1 = 0$, the choice of φ_0 does not matter, and is set to 0. The dataset contained 10000 training points, and 1000 test points, all uniformly distributed in the first dimension between -1 and 1 and in the second dimension between -2π and 2π. The labels were set to be either 0 or 1, in order to achieve perfect separation with the classification rule $\lambda(z)$.