Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks

Peter L. Bartlett [1] **David P. Helmbold** [2] **Philip M. Long** [3]

Abstract

We analyze algorithms for approximating a function $f(x) = \Phi x$ mapping $\Re^d$ to $\Re^d$ using deep linear neural networks, i.e. that learn a function h parameterized by matrices $\Theta_1, ..., \Theta_L$ and defined by $h(x) = \Theta_L \Theta_{L-1} ... \Theta_1 x$. We focus on algorithms that learn through gradient descent on the population quadratic loss in the case that the distribution over the inputs is isotropic. We provide polynomial bounds on the number of iterations for gradient descent to approximate the least squares matrix Φ, in the case where the initial hypothesis $\Theta_1 = ... = \Theta_L = I$ has excess loss bounded by a small enough constant. On the other hand, we show that gradient descent fails to converge for Φ whose distance from the identity is a larger constant, and we show that some forms of regularization toward the identity in each layer do not help. If Φ is symmetric positive definite, we show that an algorithm that initializes $\Theta_i = I$ learns an ϵ-approximation of f using a number of updates polynomial in L, the condition number of Φ, and $\log(d/\epsilon)$. In contrast, we show that if the least squares matrix Φ is symmetric and has a negative eigenvalue, then all members of a class of algorithms that perform gradient descent with identity initialization, and optionally regularize toward the identity in each layer, fail to converge. We analyze an algorithm for the case that Φ satisfies $u^\top \Phi u > 0$ for all u, but may not be symmetric. This algorithm uses two regularizers: one that maintains the invariant $u^\top \Theta_L \Theta_{L-1} ... \Theta_1 u > 0$ for all u, and another that "balances" $\Theta_1, ..., \Theta_L$ so that they have the same singular values.

[1]UC Berkeley [2]UC Santa Cruz [3]Google. Correspondence to: Peter L. Bartlett <peter@berkeley.edu>, David P. Helmbold <dph@soe.ucsc.edu>, Philip M. Long <plong@google.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

1. Introduction

Residual networks (He et al., 2016) are deep neural networks in which, roughly, subnetworks determine how a feature transformation should differ from the identity, rather than how it should differ from zero. After enabling the winning entry in the ILSVRC 2015 classification task, they have become established as a central idea in deep networks.

Hardt & Ma (2017) provided a theoretical analysis that shed light on residual networks. They showed that (a) any linear transformation with a positive determinant and a bounded condition number can be approximated by a "deep linear network" of the form $f(x) = \Theta_L \Theta_{L-1} ... \Theta_1 x$, where, for large L, each layer Θ_i is close to the identity, and (b) for networks that compose near-identity transformations this way, if the excess loss is large, then the gradient is steep. Bartlett et al. (2018a) extended both results to the nonlinear case, showing that any smooth, bi-Lipschitz map can be represented as a composition of near-identity functions, and that a suboptimal loss in a composition of near-identity functions implies that the functional gradient of the loss with respect to a function in the composition cannot be small. These results are interesting because they suggest that, in many cases, this non-convex objective may be efficiently optimized through gradient descent if the layers stay close to the identity, possibly with the help of a regularizer.

This paper describes and analyzes such algorithms for linear regression with d input variables and d response variables with respect to the quadratic loss, the same setting analyzed by Hardt and Ma. We abstract away sampling issues by analyzing an algorithm that performs gradient descent with respect to the population loss. We focus on the case that the distribution on the input patterns is isotropic. (The data may be transformed through a preprocessing step to satisfy this constraint.)

The traditional analysis of convex optimization algorithms (see Boyd & Vandenberghe, 2004) provides a bound in terms of the quality of the initial solution, together with bounds on the eigenvalues of the Hessian of the loss. For the non-convex problem of this paper, we show that if gradient descent starts at the identity in each layer, and if the excess

loss of that initial solution is bounded by a constant, then the Hessian remains well-conditioned enough throughout training for successful learning. Specifically, there is a constant c_0 such that, if the excess loss of the identity (over the least squares linear map) is at most c_0, then back-propagation initialized at the identity in each layer achieves loss within at most ϵ of optimal in time polynomial in $\log(1/\epsilon)$, d, and L (Section 3). On the other hand, we show that there is a constant c_1 and a least squares matrix Φ such that the identity has excess loss c_1 with respect to Φ, but backpropagation with identity initialization fails to learn Φ (Section 6).

We also show that if the least squares matrix Φ is symmetric positive definite then gradient descent with identity initialization achieves excess loss at most ϵ in a number of steps bounded by a polynomial in $\log(d/\epsilon)$, L and the condition number of Φ (Section 4).

In contrast, for any least squares matrix Φ that is symmetric but has a negative eigenvalue, we show that no such guarantee is possible for a wide variety of algorithms of this type: the excess loss is forever bounded below by the square of this negative eigenvalue. This holds for step-and-project algorithms, and also algorithms that initialize to the identity and regularize by early stopping or penalizing $\sum_i ||\Theta_i - I||_F^2$ (Section 6). Both this and the previous impossibility result can be proved using a least squares matrix Φ with a positive determinant and a good condition number. Recall that such Φ were proved by Hardt and Ma to have a good approximation as a product of near-identity matrices – we prove that gradient descent cannot learn them, even with the help of regularizers that reward near-identity representations.

In Section 5 we provide a convergence guarantee for a least squares matrix Φ that may not be symmetric, but satisfies the positivity condition $u^\top \Phi u > \gamma$ for some $\gamma > 0$ that appears in the bounds. We call such matrices γ-*positive*. Such Φ include rotations by acute angles. In this case, we consider an algorithm that regularizes in addition to a near-identity initialization. After the gradient update, the algorithm performs what we call *power projection*, projecting its hypothesis $\Theta_L \Theta_{L-1} ... \Theta_1$ onto the set of γ-positive matrices. Second, it "balances" $\Theta_1, ..., \Theta_L$ so that, informally, they contribute equally to $\Theta_L \Theta_{L-1} ... \Theta_1$. (See Section 5 for the details.) We view this regularizer as a theoretically tractable proxy for regularizers that promote positivity and balance between layers by adding penalties.

While, in practice, deep networks are non-linear, analysis of the linear case can provide a tractable way to gain insight through rigorous theoretical analysis (Saxe et al., 2013; Kawaguchi, 2016; Hardt & Ma, 2017). We might view back-propagation in the non-linear case as an approximation to a procedure that locally modifies the function computed by each layer in a manner that reduces the loss as fast as possible. If a non-linear network is obtained by composing transformations, each of which is chosen from a Hilbert space of functions (as in Daniely et al. (2016)), then a step in "function space" corresponds to a step in an (infinite-dimensional) linear space of functions.

Related work. The motivation for this work comes from the papers of Hardt & Ma (2017) and Bartlett et al. (2018a). Saxe et al. (2013) studied the dynamics of a continuous-time process obtained by taking the step size of backpropagation applied to deep linear neural networks to zero. Kawaguchi (2016) showed that deep linear neural networks have no suboptimal local minima. In the case that $L = 2$, the problem studied here has a similar structure as problems arising from low-rank approximation of matrices, especially as regards algorithms that approximate a matrix A by iteratively improving an approximation of the form UV. For an interesting survey on the rich literature on these algorithms, please see Ge et al. (2017a); successful algorithms have included a regularizer that promotes balance in the sizes of U and V. Taghvaei et al. (2017) studied the properties of critical points on the loss when learning deep linear neural networks in the presence of a weight decay regularizer; they studied networks that transform the input to the output through a process indexed by a continuous variable, instead of through discrete layers. Lee et al. (2016) showed that, given regularity conditions, for a random initialization, gradient descent converges to a local minimizer almost surely; while their paper yields useful insights, their regularity condition does not hold for our problem. Many papers have analyzed learning of neural networks with non-linearities. The papers most closely related to this work analyze algorithms based on gradient descent. Some of these (Andoni et al., 2014; Brutzkus & Globerson, 2017; Ge et al., 2017b; Li & Yuan, 2017; Zhong et al., 2017; Zhang et al., 2018; Brutzkus et al., 2018; Ge et al., 2018) analyze constant-depth networks. Daniely (2017) showed that stochastic gradient descent learns a subclass of functions computed by log-depth networks in polynomial time; this class includes constant-degree polynomials with polynomially bounded coefficients. Other theoretical treatments of neural network learning algorithms include Lee et al. (1996); Arora et al. (2014); Livni et al. (2014); Janzamin et al. (2015); Safran & Shamir (2016); Zhang et al. (2016); Nguyen & Hein (2017); Zhang et al. (2017); Orhan & Pitkow (2018), although these are less closely related.

Our three upper bound analyses combine a new upper bound on the operator norm of the Hessian of a deep linear network with the result of Hardt and Ma that gradients are lower bounded in terms of the loss for near-identity matrices. They otherwise have different outlines. The bound in terms of the loss of the initial solution proceeds by showing that the distance from each layer to the identity grows slowly enough that the loss is reduced before the layers stray

far enough to harm the conditioning of the Hessian. The bound for symmetric positive definite matrices proceeds by showing that, in this case, all of the layers are the same, and each of their eigenvalues converges to the Lth root of a corresponding eigenvalue of Φ. As mentioned above, the bound for γ-positive matrices Φ is for an algorithm that achieves favorable conditioning through regularization.

We expect that the theoretical analysis reported here will inform the design of practical algorithms for learning non-linear deep networks. One potential avenue for this arises from the fact that the leverage provided by regularizing toward the identity appears to already be provided by a weaker policy of promoting the property that the composition of layers is (potentially asymmetric) positive definite. Also, balancing singular values of the layers of the network aided our analysis; an analogous balancing of Jacobians associated with various layers may improve conditioning in practice in the non-linear case.

2. Preliminaries

2.1. Setting

For a joint distribution P with support contained in $\Re^d \times \Re^d$ and $g : \Re^d \to \Re^d$, define $\ell_P(g) = \mathbb{E}_{(X,Y)\sim P}(||g(X) - Y||^2/2)$. We focus on the case that, for (X, Y) drawn from P, the marginal on X is isotropic, with $\mathbb{E}XX^\top = I_d$. For convenience, we assume that $Y = \Phi X$ for $\Phi \in \Re^{d\times d}$. This assumption is without loss of generality: if Φ is the least squares matrix (so that f defined by $f(X) = \Phi X$ minimizes $\ell_P(f)$ among linear functions), for any linear g we have

$$\begin{aligned}\ell_P(g) &= \mathbb{E}\|g(X) - f(X)\|^2/2 + \mathbb{E}\|f(X) - Y\|^2/2 \\ &\quad + \mathbb{E}\left((g(X) - f(X))(f(X) - Y)\right) \\ &= \mathbb{E}\|g(X) - f(X)\|^2/2 + \mathbb{E}\|f(X) - Y\|^2/2 \\ &= \mathbb{E}\|g(X) - \Phi X)\|^2/2 + \mathbb{E}\|\Phi X - Y\|^2/2,\end{aligned}$$

since f is the projection of Y onto the set of linear functions of X. So assuming $Y = \Phi X$ corresponds to setting Φ as the least squares matrix and replacing the loss $\ell_P(g)$ by the excess loss

$$\mathbb{E}\|g(X)-\Phi X\|^2/2 = \mathbb{E}\|g(X)-Y\|^2/2-\mathbb{E}\|\Phi X-Y\|^2/2.$$

We study algorithms that learn linear mappings parameterized by deep networks. The network with L layers and parameters $\Theta = (\Theta_1, \ldots, \Theta_L)$ computes the parameterized function $f_\Theta(x) = \Theta_L\Theta_{L-1}\cdots\Theta_1 x$, where $x \in \Re^d$ and $\Theta_i \in \Re^{d\times d}$.

We use the notation $\Theta_{i:j} = \Theta_j\Theta_{j-1}\cdots\Theta_i$ for $i \le j$, so that we can write $f_\Theta(x) = \Theta_{1:L}x = \Theta_{i+1:L}\Theta_i\Theta_{1:i-1}x$.

When there is no possibility of confusion, we will sometimes refer to loss $\ell(f_\Theta)$ simply as $\ell(\Theta)$. Because the distribution of X is isotropic, $\ell(\Theta) = \frac{1}{2}||\Theta_{1:L} - \Phi||_F^2$ with respect to least squares matrix Φ. When Θ is produced by an iterative algorithm, will we also refer to loss of the tth iterate by $\ell(t)$.

Definition 1. *For $\gamma > 0$, a matrix $A \in \Re^{d\times d}$ is* γ-positive *if, for all unit length u, we have $u^\top Au > \gamma$.*

2.2. Tools and background

We use $||A||_F$ for the Frobenius norm of matrix A, $||A||_2$ for its operator norm, and $\sigma_{\min}(A)$ for its least singular value. For vector v, we use $||v||$ for its Euclidian norm.

For a matrix A and a matrix-valued function B, define $D_AB(A)$ to be the matrix with

$$(D_AB(A))_{i,j} = \frac{\partial \mathrm{vec}(B(A))_i}{\partial \mathrm{vec}(A)_j},$$

where $\mathrm{vec}(A)$ is the column vector constructed by stacking the columns of A. We use $T_{d,d}$ to denote the $d^2 \times d^2$ permutation matrix mapping $\mathrm{vec}(A)$ to $\mathrm{vec}(A^\top)$ for $A \in \Re^{d\times d}$. For $A \in \Re^{n\times m}$ and $B \in \Re^{p\times q}$, $A \otimes B$ denotes the Kronecker product, that is, the $np \times mq$ matrix of $n \times m$ blocks, with the i, jth block given by $A_{ij}B$.

We will need the gradient and Hessian of ℓ. (The gradient, which can be computed using backprop, is of course well known.) The proof is in the full paper (Bartlett et al., 2018b).

Lemma 1. *For $i < j$,*

$$\begin{aligned}D_{\Theta_i}\ell(f_\Theta) &= (\mathrm{vec}(I_d))^\top\left(\left(\Theta_{1:i-1}^\top \otimes (\Theta_{1:L}-\Phi)^\top\Theta_{i+1:L}\right)\right) \\ &= \mathrm{vec}(G)^\top,\end{aligned}$$

for the $d \times d$ matrix given by

$$G \stackrel{\text{def}}{=} \Theta_{i+1:L}^\top(\Theta_{1:L} - \Phi)\Theta_{1:i-1}^\top. \tag{1}$$

$$\begin{aligned}&D_{\Theta_j}D_{\Theta_i}\ell(f_\Theta) \\ &= (I_{d^2} \otimes (\mathrm{vec}(I_d))^\top)(I_d \otimes T_{d,d} \otimes I_d)\left(\mathrm{vec}(\Theta_{1:i-1}^\top) \otimes I_{d^2}\right) \\ &\quad ((\Theta_{i+1:L}^\top\Theta_{j+1:L} \otimes \Theta_{1:j-1}^\top)T_{d,d} \\ &\quad + (\Theta_{i+1:j-1}^\top \otimes (\Theta_{1:L} - \Phi)^\top\Theta_{j+1:L})).\end{aligned}$$

$$\begin{aligned}&D_{\Theta_i}D_{\Theta_i}\ell(f_\Theta) \\ &= (I_{d^2} \otimes (\mathrm{vec}(I_d))^\top)(I_d \otimes T_{d,d} \otimes I_d)\left(\mathrm{vec}(\Theta_{1:i-1}^\top) \otimes I_{d^2}\right) \\ &\quad \left(\Theta_{i+1:L}^\top\Theta_{i+1:L} \otimes \Theta_{1:i-1}^\top\right)T_{d,d}.\end{aligned}$$

3. Targets near the identity

In this section, we prove an upper bound for gradient descent in terms of the loss of the initial solution.

3.1. Procedure and upper bound

First, set $\Theta^{(0)} = (I, I, ..., I)$, and then iteratively update

$$\Theta_i^{(t+1)} = \Theta_i^{(t)} - \eta(\Theta_{i+1:L}^{(t)})^\top \left(\Theta_{1:L}^{(t)} - \Phi\right)(\Theta_{1:i-1}^{(t)})^\top.$$

Theorem 1. *There are positive constants c_1 and c_2 and polynomials p_1 and p_2 such that, if $\ell(\Theta_{1:L}^{(0)}) \leq c_1$, $L \geq c_2$, and $\eta \leq \frac{1}{p_1(L,d,||\Phi||_2)}$, then the above gradient descent procedure achieves $\ell(f_{\Theta^{(t)}}) \leq \epsilon$ within $t = p_2\left(\frac{1}{\eta}\right)\ln\left(\frac{\ell(0)}{\epsilon}\right)$ iterations.*

3.2. Proof of Theorem 1

The following lemma, which is implicit in the proof of Theorem 2.2 in (Hardt & Ma, 2017), shows that the gradient is steep if the loss is large and the singular values of the layers are not too small.

Lemma 2 (Hardt & Ma 2017)**.** *Let $\nabla_\Theta \ell(\Theta)$ be the gradient of $\ell(\Theta)$ with respect to any flattening of Θ. If, for all layers i, $\sigma_{\min}(\Theta_i) \geq 1 - a$, then $||\nabla_\Theta \ell(\Theta)||^2 \geq 4\ell(\Theta)L(1-a)^{2L}$.*

Next, we show that, if $\Theta^{(t)}$ and $\Theta^{(t+1)}$ are both close to the identity, then the gradient is not changing very fast between them, so that rapid progress continues to be made. We prove this through an upper bound on the operator norm of the Hessian that holds uniformly over members of a ball around the identity, which in turn can be obtained through a bound on the Frobenius norm. The proof is in the full paper (Bartlett et al., 2018b).

Lemma 3. *Choose an arbitrary Θ with $||\Theta_i||_2 \leq 1 + z$ for all i, and least squares matrix Φ with $||\Phi||_2 \leq (1+z)^L$. Let ∇^2 be the Hessian of $\ell(f_\Theta)$ with respect to an arbitrary flattening of the parameters of Θ. We have*

$$||\nabla^2||_F \leq 3Ld^5(1+z)^{2L}.$$

Armed with Lemmas 2 and 3, let us now analyze gradient descent. Very roughly, our strategy will be to show that the distance from the identity to the various layers grows slowly enough for the leverage from Lemmas 2 and 3 to enable successful learning. Let $\mathcal{R}(\Theta) = \max_i ||\Theta_i - I||_2$. From the update, we have

$$\begin{aligned}
&||\Theta_i^{(t+1)} - I||_2 \\
&\leq ||\Theta_i^{(t)} - I||_2 + \eta||(\Theta_{i+1:L}^{(t)})^\top \left(\Theta_{1:L}^{(t)} - \Phi\right)(\Theta_{1:i-1}^{(t)})^\top||_2 \\
&\leq ||\Theta_i^{(t)} - I||_2 + \eta(1 + \mathcal{R}(\Theta^{(t)}))^L||\Theta_{1:L}^{(t)} - \Phi||_2 \\
&\leq ||\Theta_i^{(t)} - I||_2 + \eta(1 + \mathcal{R}(\Theta^{(t)}))^L||\Theta_{1:L}^{(t)} - \Phi||_F.
\end{aligned}$$

If $\mathcal{R}(t) = \max_{s \leq t} \mathcal{R}(\Theta^{(s)})$ (so $\mathcal{R}(0) = 0$) and $\ell(t) = \frac{1}{2}||\Theta_{1:L}^{(t)} - \Phi||_F^2$, this implies

$$\mathcal{R}(t+1) \leq \mathcal{R}(t) + \eta(1 + \mathcal{R}(t))^L\sqrt{2\ell(t)}. \qquad (2)$$

By Lemma 3, for all Θ on the line segment from $\Theta^{(t)}$ to $\Theta^{(t+1)}$, we have

$$||\nabla_\Theta^2||_2 \leq ||\nabla_\Theta^2||_F \leq 3Ld^5 \max\{(1+\mathcal{R}(t+1)^{2L}, ||\Phi||_2^2\},$$

so that

$$\begin{aligned}
\ell(t+1) \leq \ell(t) - \eta||\nabla_{\Theta^{(t)}}||^2 + \\
\frac{3}{2}\eta^2 Ld^5 \max\{(1+\mathcal{R}(t+1))^{2L}, ||\Phi||_2^2\}||\nabla_{\Theta^{(t)}}||^2.
\end{aligned}$$

Thus, if we ensure

$$\eta \leq \frac{1}{3Ld^5(\max\{(1+\mathcal{R}(t+1))^{2L}, ||\Phi||_2^2\}}, \qquad (3)$$

we have $\ell(t+1) \leq \ell(t) - (\eta/2)||\nabla_{\Theta^{(t)}}||^2$, which, using Lemma 2, gives

$$\ell(t+1) \leq \left(1 - 2\eta L(1-\mathcal{R}(t))^{2L}\right)\ell(t). \qquad (4)$$

Pick any $c \geq 1$. Assume that $L \geq (4/3)\ln c = c_2$, $\ell(\Theta_{1:L}^{(0)}) \leq \frac{\ln(c)^2}{8c^{10}} = c_1$ and $\eta \leq \frac{1}{3Ld^5 \max\{c^4, ||\Phi||_2^2\}}$. We claim that, for all $t \geq 0$,

1. $\mathcal{R}(t) \leq \eta c\sqrt{2\ell(0)}\sum_{0 \leq s < t}\exp\left(-\frac{s\eta L}{c^4}\right)$
2. $\ell(t) \leq \left(\exp\left(-\frac{2t\eta L}{c^4}\right)\right)\ell(0)$.

The base case holds as $\mathcal{R}(0) = 0$ and $\ell(0) = \ell(0)$.

Before starting the inductive step, notice that for any $t \geq 0$,

$$\begin{aligned}
&\eta c\sqrt{2\ell(0)}\sum_{0 \leq s < t}\exp\left(-\frac{s\eta L}{c^4}\right) \\
&\leq \eta c\sqrt{2\ell(0)} \times \frac{1}{1 - \exp\left(-\frac{\eta L}{c^4}\right)} \\
&\leq \eta c\sqrt{2\ell(0)} \times \frac{2c^4}{\eta L} \quad (\text{since } \tfrac{\eta L}{c^4} \leq 1) \\
&= \frac{2c^5\sqrt{2\ell(0)}}{L} \leq \frac{\ln c}{L} \leq 3/4
\end{aligned}$$

where the last two inequalities follow from the constraints on $\ell(0)$ and L.

Using (2),

$$\begin{aligned}
\mathcal{R}(t+1) &\leq \mathcal{R}(t) + \eta(1+\mathcal{R}(t))^L\sqrt{2\ell(t)} \\
&\leq \mathcal{R}(t) + \eta\left(1 + \frac{\ln c}{L}\right)^L\sqrt{2\ell(t)} \\
&\leq \mathcal{R}(t) + \eta c\sqrt{2\ell(t)} \\
&\leq \mathcal{R}(t) + \eta c\sqrt{2\ell(0)}\exp\left(-\frac{t\eta L}{c^4}\right) \\
&\leq \eta c\sqrt{2\ell(0)}\sum_{0 \leq s < t+1}\exp\left(-\frac{s\eta L}{c^4}\right).
\end{aligned}$$

Since $\mathcal{R}(t+1) \leq \frac{\ln c}{L}$, the choice of η satisfies (3), so

$$\ell(t+1) \leq \left(1 - 2\eta L(1-\mathcal{R}(t))^{2L}\right)\ell(t).$$

Now consider $(1-\mathcal{R}(t))^{2L}$:

$$\begin{aligned}
\ln\left((1-\mathcal{R}(t))^{2L}\right) &= 2L\ln(1-\mathcal{R}(t)) \\
&\geq 2L(-2\mathcal{R}(t)) \qquad \text{since } \mathcal{R}(t) \in [0, 3/4] \\
&\geq 2L\left(-2\frac{\ln c}{L}\right) \qquad \text{since } \mathcal{R}(t) \leq \frac{\ln c}{L} \\
(1-\mathcal{R}(t))^{2L} &\geq 1/c^4.
\end{aligned}$$

Using this in the bound on $\ell(t+1)$:

$$\begin{aligned}
\ell(t+1) &\leq \left(1 - 2\eta L(1-\mathcal{R}(t))^{2L}\right)\ell(t) \\
&\leq \left(1 - \frac{2\eta L}{c^4}\right)\ell(t) \\
&\leq \left(\exp\left(-\frac{2\eta L}{c^4}\right)\right)\left(\exp\left(-\frac{2t\eta L}{c^4}\right)\right)\ell(0) \\
&= \left(\exp\left(-\frac{2(t+1)\eta L}{c^4}\right)\right)\ell(0).
\end{aligned}$$

Solving $\ell(0)\exp\left(-\frac{2t\eta L}{c^4}\right) \leq \epsilon$ for t and recalling that $\eta < 1/c^4$ completes the proof of the theorem.

4. Symmetric positive definite Φ

In this section, we analyze the procedure of Section 3.1 when the least squares matrix Φ is symmetric and positive definite.

Theorem 2. *There is an absolute positive constant c_3 such that, if Φ is symmetric and γ-positive with $0 < \gamma < 1$, and $L \geq c_3 \ln(||\Phi||_2/\gamma)$, then for all $\eta \leq \frac{1}{L(1+||\Phi||_2^2)}$, gradient descent achieves $\ell(f_{\Theta^{(t)}}) \leq \epsilon$ in $\mathrm{poly}(L, ||\Phi||_2/\gamma, 1/\eta)\log(d/\epsilon)$ iterations.*

Note that a symmetric matrix is γ-positive when its minimum eigenvalue is at least γ.

4.1. Proof of Theorem 2

Let Φ be a symmetric, real, γ-positive matrix with $\gamma > 0$, and let $\Theta^{(0)}, \Theta^{(1)}, ...$ be the iterates of gradient descent with a step size $0 < \eta \leq \frac{1}{L(1+||\Phi||_2^2)}$.

Definition 2. *Symmetric matrices $\mathcal{A} \subseteq \Re^{d\times d}$ are* commuting normal matrices *if there is a single unitary matrix U such that for all $A \in \mathcal{A}$, $U^\top AU$ is diagonal.*

We will use the following well-known facts about commuting normal matrices.

Lemma 4 (Horn & Johnson 2013)**.** *If $\mathcal{A} \subseteq \Re^{d\times d}$ is a set of symmetric commuting normal matrices and $A, B \in \mathcal{A}$, the following hold:*

- *$AB = BA$;*
- *for all scalars α and β, $\mathcal{A} \cup \{\alpha A + \beta B, AB\}$ are commuting normal;*
- *there is a unitary matrix U such that $U^\top AU$ and $U^\top BU$ are real and diagonal;*
- *the multiset of singular values of A is the same as the multiset of magnitudes of its eigenvalues;*
- *$||A-I||_2$ is the largest value of $|z-1|$ for an eigenvalue z of A.*

Lemma 5. *The matrices $\{\Phi\} \cup \{\Theta_i^{(t)} : i \in \{1, ..., L\}, t \in \mathbb{Z}^+\}$ are commuting normal. For all t, $\Theta_1^{(t)} = ... = \Theta_L^{(t)}$.*

Proof. The proof is by induction. The base case follows from the fact that Φ and I are commuting normal.

For the induction step, the fact that

$$\begin{aligned}
&\{\Phi\} \cup \{\Theta_i^{(s)} : i \in \{1, ..., L\}, s \leq t\} \\
&\cup \{\Theta_i^{(s+1)} : i \in \{1, ..., L\}, s \leq t\}
\end{aligned}$$

are commuting normal follows from Lemma 4. The update formula now reveals that $\Theta_1^{(t+1)} = ... = \Theta_L^{(t+1)}$. □

Now we are ready to analyze the dynamics of the learning process. Let $\Phi = U^\top D^L U$ be a diagonalization of Φ. Let $\Gamma = \max\{1, ||\Phi||_2\}$. We next describe a sense in which gradient descent learns each eigenvalue independently.

Lemma 6. *For each t, there is a real diagonal matrix $\hat{D}^{(t)}$ such that, for all i, $\Theta_i^{(t)} = U^\top \hat{D}^{(t)} U$ and*

$$\hat{D}^{(t+1)} = \hat{D}^{(t)} - \eta(\hat{D}^{(t)})^{L-1}((\hat{D}^{(t)})^L - D^L). \tag{5}$$

Proof. Lemma 5 implies that there is a single real U such that $\Theta_i^{(t)} = U^\top \hat{D}^{(t)} U$ for all i. Applying Lemma 1, recalling that $\Theta_1^{(t)} = ... = \Theta_L^{(t)}$, and applying the fact that $\Theta_i^{(t)}$ and Φ commute, we get

$$\Theta_i^{(t+1)} = \Theta_i^{(t)} - \eta(\Theta_i^{(t)})^{L-1}\left((\Theta_i^{(t)})^L - \Phi\right).$$

Replacing each matrix by its diagonalization, we get

$$\begin{aligned}
&U^\top \hat{D}^{(t+1)} U \\
&= U^\top \hat{D}^{(t)} U \\
&\quad - \eta(U^\top(\hat{D}^{(t)})^{L-1}U)\left(U^\top(\hat{D}^{(t)})^L U - U^\top D^L U\right) \\
&= U^\top \hat{D}^{(t)} U - \eta U^\top(\hat{D}^{(t)})^{L-1}\left((\hat{D}^{(t)})^L - D^L\right)U,
\end{aligned}$$

and left-multiplying by U and right-multiplying by $U^\top$ gives (5). □

We will now analyze the convergence of each $\hat{D}_{kk}^{(t)}$ to D_{kk} separately. Let us focus for now on an arbitrary single index k, let $\lambda = D_{kk}$ and $\hat{\lambda}^{(t)} = \hat{D}_{kk}^{(t)}$.

Recalling that $||\Phi||_2 \leq \Gamma$, we have $\gamma^{1/L} \leq \lambda \leq \Gamma^{1/L}$. Also, $\Gamma^{1/L} = e^{\frac{1}{L}\ln\Gamma} \leq e^{1/a} \leq 1 + 2/a$ whenever $a \geq 1$ and $L \geq a\ln\Gamma$. Similarly, $\gamma^{1/L} \geq 1 - a$ whenever $L \geq a\ln(1/\gamma)$. Thus, there are absolute constants c_3 and c_4 such that $|1-\lambda| \leq \frac{c_4\ln(\Gamma/\gamma)}{L} < 1$ for all $L \geq c_3\ln(\Gamma/\gamma)$.

We claim that, for all t, $\hat{\lambda}^{(t)}$ lies between 1 and λ inclusive, so that $|\hat{\lambda}^{(t)} - \lambda| \leq \frac{c_4\ln(\Gamma/\gamma)}{L}$. The base case holds because $\hat{\lambda}^{(t)} = 1$ and $|1-\lambda| \leq \frac{c_4\ln(\Gamma/\gamma)}{L}$. Now let us work on the induction step. Applying (5) together with Lemma 1, we get

$$\hat{\lambda}^{(t+1)} = \hat{\lambda}^{(t)} + \eta(\hat{\lambda}^{(t)})^{L-1}(\lambda^L - (\hat{\lambda}^{(t)})^L). \tag{6}$$

By the induction hypothesis, we just need to show that $\text{sign}(\hat{\lambda}^{(t+1)} - \hat{\lambda}^{(t)}) = \text{sign}(\lambda - \hat{\lambda}^{(t)})$ and $|\hat{\lambda}^{(t+1)} - \hat{\lambda}^{(t)}| \leq |\lambda - \hat{\lambda}^{(t)}|$ (i.e., the step is in the correct direction, and does not "overshoot"). First, to see that the step is in the right direction, note that $\lambda^L \geq (\hat{\lambda}^{(t)})^L$ if and only if $\lambda \geq (\hat{\lambda}^{(t)})$, and the inductive hypothesis implies that $\hat{\lambda}^{(t)}$, and therefore $(\hat{\lambda}^{(t)})^{L-1}$, is non-negative. To show that $|\hat{\lambda}^{(t+1)} - \hat{\lambda}^{(t)}| \leq |\lambda - \hat{\lambda}^{(t)}|$, it suffices to show that $\eta(\hat{\lambda}^{(t)})^{L-1}\left|\lambda^L - (\hat{\lambda}^{(t)})^L)\right| \leq |\lambda - \hat{\lambda}^{(t)}|$, which, in turn would be implied by $\eta \leq \left|\frac{1}{(\hat{\lambda}^{(t)})^{L-1}\left(\sum_{i=0}^{L-1}(\hat{\lambda}^{(t)})^i\lambda^{L-1-i}\right)}\right|$ (since $\lambda^L - (\hat{\lambda}^{(t)})^L = (\lambda - \hat{\lambda}^{(t)})\sum_{i=0}^{L-1}(\hat{\lambda}^{(t)})^i\lambda^{L-1-i}$), which follows from the inductive hypothesis and $\eta \leq \frac{1}{L\Gamma^2}$.

We have proved that each $\hat{\lambda}^{(t)}$ lies between λ and 1, so that $|1 - \hat{\lambda}^{(t)}| \leq |1-\lambda| \leq c_4\ln(\Gamma/\gamma)$.

Now, since the step is in the right direction, and does not overshoot,

$$\begin{aligned}
&|\hat{\lambda}^{(t+1)} - \lambda| \\
&\leq |\hat{\lambda}^{(t)} - \lambda| - \eta(\hat{\lambda}^{(t)})^{L-1}|\lambda^L - (\hat{\lambda}^{(t)})^L| \\
&\leq |\hat{\lambda}^{(t)} - \lambda|\left(1 - \eta(\hat{\lambda}^{(t)})^{L-1}\left(\sum_{i=0}^{L-1}(\hat{\lambda}^{(t)})^i\lambda^{L-1-i}\right)\right) \\
&\leq |\hat{\lambda}^{(t)} - \lambda|\left(1 - \eta L\gamma^2\right),
\end{aligned}$$

since the fact that $\hat{\lambda}^{(t)}$ lies between 1 and λ implies that $\hat{\lambda}^{(t)} \geq \gamma^{1/L}$. Thus, $|\hat{\lambda}^{(t)} - \lambda| \leq \left(1 - \eta L\gamma^2\right)^t c_4\ln(\Gamma/\gamma)$. This implies that, for any $\epsilon \in (0,1)$, for any absolute constant c_5, there is a constant c_6 such that, after $c_6\frac{1}{\eta L\gamma^2}\ln\left(\frac{dL\ln\Gamma}{\gamma\epsilon}\right)$ steps, we have $|\hat{\lambda}^{(t)} - \lambda| \leq \frac{c_5\gamma\sqrt{\epsilon}}{L\Gamma\sqrt{d}}$. Writing $r = \hat{\lambda}^{(t)} - \lambda$, this implies, if c_5 is small enough, that

$$\begin{aligned}
&((\hat{\lambda}^{(t)})^L - \lambda^L)^2 = ((\lambda+r)^L - \lambda^L)^2 \leq \Gamma^2\left(\left(1+\frac{r}{\lambda}\right)^L - 1\right)^2 \\
&\leq \Gamma^2\left(\frac{2c_5rL}{\lambda}\right)^2 \leq \Gamma^2\left(\frac{2c_5rL}{\gamma}\right)^2 \leq \frac{\epsilon}{d}.
\end{aligned}$$

Thus, after $O\left(\frac{1}{\eta L\gamma^2}\ln\left(\frac{dL\ln\Gamma}{\gamma\epsilon}\right)\right)$ steps, $(D_{kk} - \hat{D}_{kk}^{(t)})^2 \leq \epsilon/d$ for all k, and therefore $\ell(\Theta^{(t)}) \leq \epsilon$, completing the proof.

5. Asymmetric positive definite matrices

We have seen that if the least squares matrix is symmetric, γ-positivity is sufficient for convergence of gradient descent. We shall see in Section 6 that positivity is also necessary for a broad family of gradient-based algorithms to converge to the optimal solution when the least squares matrix is symmetric. Thus, in the symmetric case, positivity characterizes the success of gradient methods. In this section, we show that positivity suffices for the convergence of a gradient method even without the assumption that the least squares matrix is symmetric.

Note that the set of γ-positive (but not necessarily symmetric) matrices includes both rotations by an acute angle and "partial reflections" of the form $ax + b\,\text{refl}(x)$ where $\text{refl}(\cdot)$ is a length-preserving reflection and $0 \leq |b| < a$. Since $\left(u^\top Au\right)^\top = u^\top A^\top u$, a matrix A is γ-positive if and only if $u^\top(A + A^\top)u \geq 2\gamma$ for all unit length u, i.e. $A + A^\top$ is positive definite with eigenvalues at least 2γ.

5.1. Balanced factorizations

The algorithm analyzed in this section uses a construction that is new, as far as we know, that we call a *balanced factorization*. This factorization may be of independent interest.

Recall that a *polar decomposition* of a matrix A consists of a unitary matrix R and a positive semidefinite matrix P such that $A = RP$. The *principal Lth root* of a complex number whose expression in polar coordinates is $re^{\theta i}$ is $r^{1/L}e^{\theta i/L}$. The *principal Lth root* of a matrix A is the matrix B such that $B^L = A$, and each eigenvalue of B is the principal Lth root of the corresponding eigenvalue of A.

Definition 3. *If A be a matrix with polar decomposition RP, then A has the* balanced factorization $A = A_1, ..., A_L$ *where for each i,*

$$A_i = R^{1/L}P_i, \text{ with } P_i = R^{(L-i)/L}P^{1/L}R^{-(L-i)/L},$$

and each of the Lth roots is the principal Lth root.

The motivation for balanced factorization is as follows. We want each factor to do a $1/L$ fraction of the total amount of rotation, and a $1/L$ fraction of the total amount of scaling. However, the scaling done by the ith factor should be done in directions that take account of the partial rotations done by the other factors. The following is the key property of the balanced factorization; its proof is in the full paper (Bartlett et al., 2018b).

Lemma 7. *If $\sigma_1, ..., \sigma_d$ are the singular values of A, and $A_1, ..., A_L$ is a balanced factorization of A, then the following hold: (a) $A = \prod_{i=1}^{L} A_i$; (b) for each $i \in \{1, ..., L\}$, $\sigma_1^{1/L}, ..., \sigma_d^{1/L}$ are the singular values of A_i.*

5.2. Procedure and upper bound

The following is the *power projection algorithm*. It has a positivity parameter $\gamma > 0$, and uses $\mathcal{H} = \{A : \forall u \text{ s.t. } ||u|| = 1,\ u^\top A u \geq \gamma\}$ as its "hypothesis space". First, it initializes $\Theta_i^{(0)} = \gamma^{1/L} I$ for all $i \in \{1, ..., L\}$. Then, for each t, it does the following.

- **Gradient Step.** For each $i \in \{1, ..., L\}$, update:
$$\Theta_i^{(t+1/2)} = \Theta_i^{(t)} - \eta (\Theta_{i+1:L}^{(t)})^\top \left(\Theta_{1:L}^{(t)} - \Phi\right) (\Theta_{1:i-1}^{(t)})^\top.$$
- **Power Project.** Compute the projection $\Psi^{(t+1/2)}$ (w.r.t. the Frobenius norm) of $\Theta_{1:L}^{(t+1/2)}$ onto $\mathcal{H}$.
- **Factor.** Let $\Theta_1^{(t+1)}, ..., \Theta_L^{(t+1)}$ be the balanced factorization of $\Psi^{(t+1/2)}$, so that $\Psi^{(t+1/2)} = \Theta_{1:L}^{(t+1)}$.

Theorem 3. *For any Φ such that $u^\top \Phi u > \gamma$ for all unit-length u, the power projection algorithm produces $\Theta^{(t)}$ with $\ell(\Theta^{(t)}) \leq \epsilon$ in $\mathrm{poly}(d, ||\Phi||_F, \frac{1}{\gamma}) \log(1/\epsilon)$ iterations.*

5.3. Proof of Theorem 3

Lemma 8. *For all t, $\Theta_{1:L}^{(t)} \in \mathcal{H}$.*

Proof. $\Theta_{1:L}^{(0)} = \gamma I \in \mathcal{H}$, and, for all t, $\Psi^{(t+1/2)}$ is obtained by projection onto $\mathcal{H}$, and $\Theta_{1:L}^{(t+1)} = \Psi^{(t+1/2)}$. □

Definition 4. *The exponential of a matrix A is $\exp(A) \stackrel{\text{def}}{=} \sum_{k=0}^{\infty} \frac{1}{k!} A^k$, and B is a logarithm of A if $A = \exp(B)$.*

Lemma 9 (Culver 1966)**.** *A real matrix has a real logarithm if and only if it is invertible and each Jordan block belonging to a negative eigenvalue occurs an even number of times.*

Lemma 10. *For all t, $\Theta_{1:L}^{(t)}$ has a real Lth root.*

Proof. Since $\Theta_{1:L}^{(t)} \in \mathcal{H}$ implies $u^\top \Theta_{1:L}^{(t)} u > 0$ for all u, $\Theta_{1:L}^{(t)}$ does not have a negative eigenvalue and is invertible. By Lemma 9, $\Theta_{1:L}^{(t)}$ has a real logarithm. Thus, its real Lth root can be constructed via $\exp(\log(\Theta_{1:L}^{(t)})/L)$. □

The preceding lemma implies that the algorithm is well-defined, since all of the required roots can be calculated.

Lemma 11. *$\mathcal{H}$ is convex.*

Proof. Suppose A and B are in H and $\lambda \in (0, 1)$. We have
$$u^\top(\lambda A + (1-\lambda)B)u = \lambda u^\top A u + (1-\lambda) u^\top B u \geq \gamma.$$
□

Lemma 12. *For all $A \in \mathcal{H}$, $\sigma_{\min}(A) \geq \gamma$.*

Proof. Let u and v be singular vectors such that $uAv^\top = \sigma_{\min}(A)$.
$$\gamma \leq v^\top A v = \sigma_{\min}(A) v^\top u \leq \sigma_{\min}(A).$$
□

Lemma 13. *For all t, $\sigma_{\min}(\Theta_i^{(t)}) \geq \gamma^{1/L}$.*

Proof. First, $\sigma_{\min}(\Theta_i^{(0)}) = \gamma^{1/L} \geq \gamma^{1/L}$.

Now consider $t > 0$. Since $\Psi^{(t-1/2)}$ was projected into $\mathcal{H}$, we have $\sigma_{\min}(\Psi^{(t-1/2)}) \geq \gamma$. Lemma 7 then completes the proof. □

Define $U(t) = \max\left\{\max_{s \leq t} \max_i ||\Theta_i^{(s)}||_2, ||\Phi||_2^{1/L}\right\}$, $B(t) = \min_{s \leq t} \min_i \sigma_{\min}(\Theta_i^{(s)})$, and recall that $\ell(t) = ||\Theta_{1:L}^{(t)} - \Phi||_F^2$.

Arguing as in the initial portion of Section 3.2, as long as
$$\eta \leq \frac{1}{3Ld^5 U(t)^{2L}} \tag{7}$$
we have $\ell(t + 1/2) \leq \left(1 - \eta L B(t)^{2L}\right) \ell(t)$ (see Equation 4). Lemma 13 gives $B(t) \geq \gamma^{1/L}$, so $\ell(t + 1/2) \leq \left(1 - \eta L \gamma^2\right) \ell(t)$. Since $\Psi^{(t+1/2)}$ is the projection of $\Theta_{1:L}^{(t+1/2)}$ onto a convex set $\mathcal{H}$ that contains Φ, and $\Theta_{1:L}^{(t+1)} = \Psi^{(t+1/2)}$, (7) implies
$$\ell(t+1) \leq \ell(t+1/2) \leq \left(1 - \eta L \gamma^2\right) \ell(t). \tag{8}$$

Next, we prove an upper bound on U.

Lemma 14. *For all t, $U(t) \leq \left(\sqrt{\ell(t)} + ||\Phi||_F\right)^{1/L}$.*

Proof. Recall that $\ell(t) = ||\Theta_{1:L}^{(t)} - \Phi||_F^2$. By the triangle inequality, we have $||\Theta_{1:L}^{(t)}||_F \leq \sqrt{\ell(t)} + ||\Phi||_F$. Thus $||\Theta_{1:L}^{(t)}||_2 \leq \sqrt{\ell(t)} + ||\Phi||_F$. By Lemma 7, for all i, we have $||\Theta_i^{(t)}||_2 \leq \left(\sqrt{\ell(t)} + ||\Phi||_F\right)^{1/L}$. Since $||\Phi||_2 \leq ||\Phi||_F$, this completes the proof. □

Note that the triangle inequality implies that $\ell(0) \leq ||\Theta_{1:L}^{(0)}||_F^2 + ||\Phi||_F^2 \leq \gamma^2 d + ||\Phi||_F^2$. Since $\sigma_{\min}(\Phi) \geq \gamma$, we have $||\Phi||_F^2 \geq \gamma^2 d$, so $\ell(t) \leq 2||\Phi||_F^2$ and $U(t) \leq (3||\Phi||_2)^{1/L}$. Now, if we set $\eta = \frac{1}{cLd^5||\Phi||_F^2}$, for a large enough absolute constant c, then (7) is satisfied, so that (8) gives $\ell(t+1) \leq \left(1 - \frac{\gamma^2}{cd^5||\Phi||_F^2}\right)\ell(t)$ and the power projection algorithm achieves $\ell(t+1) \leq \epsilon$ after

$$O\left(\frac{d^5||\Phi||_F^2}{\gamma^2}\log\left(\frac{\ell(0)}{\epsilon}\right)\right)$$
$$= O\left(\frac{d^5||\Phi||_F^2}{\gamma^2}\log\left(\frac{||\Phi||_F^2}{\epsilon}\right)\right)$$

updates.

6. Failure

In this section, we show that positive definite Φ are necessary for several gradient descent algorithms with different kinds of regularization to minimize the loss. One family of algorithms that we will analyze is parameterized by a function ψ mapping the number of inputs d and the number of layers L to a radius $\psi(d, L)$, step sizes η_t and initialization parameter $\gamma \geq 0$. In particular, a *ψ-step-and-project* algorithm is any instantiation of the following algorithmic template.

Initialize each $\Theta_i^{(0)} = \gamma^{1/L} I$ for some $\gamma \geq 0$ and iterate:

- **Gradient Step.** For each $i \in \{1, ..., L\}$, update:

$$\Theta_i^{(t+1/2)} = \Theta_i^{(t)} - \eta_t(\Theta_{i+1:L}^{(t)})^\top \left(\Theta_{1:L}^{(t)} - \Phi\right)(\Theta_{1:i-1}^{(t)})^\top.$$

- **Project.** Set each Θ_i^{t+1} to the projection of $\Theta_i^{t+1/2}$ onto $\{A : ||A - I||_2 \leq \psi(d, L)\}$.

We will also show that *Penalty Regularized Gradient Descent* which uses gradient descent with any step sizes η_t on the regularized objective $\ell(\Theta) + \frac{\kappa}{2}\sum_i ||I - \Theta||_F^2$ also fails to minimize the loss.

Both results use the simple observation that when $\Theta_{1:L}$ and Φ are mutually diagonalizable then

$$||\Theta_{1:L} - \Phi||_F^2 = ||U^\top \hat{D} U - U^\top D U||_F^2 = \sum_{j=1}^{d}(\hat{D}_{jj} - D_{jj})^2,$$

where the D_{ii} are the eigenvalues of Φ.

Theorem 4. *If the least squares matrix Φ is symmetric then Penalty Regularized Gradient Descent produces hypotheses $\Theta_{1:L}^{(t)}$ that are commuting normal with Φ.*

In addition, if Φ has a negative eigenvalue $-\lambda$ and L is even, then $\ell(\Theta^{(t)}) \geq \lambda^2/2$ for all t.

Proof. For all t, Penalty Regularized Gradient Descent produces $\Theta_i^{(t+1)} = (1-\kappa)\Theta_i^{(t)} + \kappa I - \eta_t(\Theta_{i+1:L}^{(t)})^\top \left(\Theta_{1:L}^{(t)} - \Phi\right)(\Theta_{1:i-1}^{(t)})^\top$. Thus, by induction, the $\Theta_i^{(t)}$ are matrix polynomials of Φ, and therefore they are all commuting normal. As in Lemmas 5 and 6 each $\Theta_i^{(t)}$ is the same $U^\top \tilde{D}^{(t)} U$ and $\Theta_{1:L}^{(t)} = U^\top (\tilde{D}^{(t)})^L U$. Since L is even, each $(\tilde{D}^{(t)})_{jj}^L \geq 0$, so $\ell(\Theta^{(t)}) = \frac{1}{2}||\Theta_{1:L}^{(t)} - \Phi||_F^2 \geq \lambda^2/2$. □

To analyze step-and-project algorithms, it is helpful to first characterize the project step. The proof is in the full paper (Bartlett et al., 2018b) (see also (Lefkimmiatis et al., 2013)).

Lemma 15. *Let X be a symmetric matrix and let $U^\top D U$ be its diagonalization.*

For $a > 0$, let Y be the Frobenius norm projection of X onto $\mathcal{B}_a = \{A : A \text{ is symmetric psd and } ||A - I||_2 \leq a\}$. Then $Y = U^\top \tilde{D} U$ where $\tilde{D}$ is obtained from D by projecting all of its diagonal elements onto $[1 - a, 1 + a]$.

Thus $\{X, Y\}$ are symmetric commuting normal matrices.

Theorem 5. *If the least squares matrix Φ is symmetric then ψ-step-and-project algorithms produce hypotheses $\Theta_{1:L}^{(t)}$ that are commuting normal with Φ.*

In addition, if Φ has a negative eigenvalue $-\lambda$ and either L is even or $\psi(L, d) \leq 1$, then $\ell(\Theta^{(t)}) \geq \lambda^2/2$ for all t.

Proof. As in Lemmas 5 and 6, the $\Theta_i^{(t+1/2)}$ are identical and mutually diagonalizable with Φ. Lemma 15 shows that this is preserved by the projection step. Thus there is a real diagonal $\tilde{D}^{(t)}$ such that each $\Theta_i^{(t)} = U^\top D_i^{(t)} U$, so $\Theta_{1:L}^{(t)} = U^\top (\tilde{D}^{(t)})^L U$.

When L is even, each $(\tilde{D}^{(t)})^L)_{j,j} \geq 0$. When $\psi(d, L) \leq 1$ then the projection ensures that the elements of $\tilde{D}^{(t)}$ are non-negative, and thus each $(\tilde{D}^{(t)})^L)_{j,j} \geq 0$. In either case, $\ell(\Theta^{(t)}) = \frac{1}{2}||\Theta_{1:L}^{(t)} - \Phi||_F^2 \geq \lambda^2/2$. □

One choice of Φ that satisfies the requirements of Theorems 4 and 5 is $\Phi = \text{diag}(-\lambda, 1, 1, ..., 1)$. For constant λ, the loss of $\Theta^{(0)} = (I, I, ..., I)$ is a constant for this target. Another choice is $\Phi = \text{diag}(-\lambda, -\lambda, 1, 1, ..., 1)$, which has a positive determinant.

Our proof of failure to minimize the loss exploits the fact that the layers are initialized to multiples of the identity. Since the training process is a continuous function of the initial solution, this implies that any convergence to a good solution will be very slow if the initializations are sufficiently close to the identity.

Acknowledgements

We thank Yair Carmon, Roy Frostig, Vineet Gupta, Moritz Hardt, Tomer Koren, Hanie Sedghi, Yoram Singer and Kunal Talwar for valuable conversations. PB gratefully acknowledges the support of the NSF through grant IIS-1619362.

References

Andoni, A., Panigrahy, R., Valiant, G., and Zhang, L. Learning polynomials with neural networks. In *ICML*, 2014.

Arora, S., Bhaskara, A., Ge, R., and Ma, T. Provable bounds for learning some deep representations. In *International Conference on Machine Learning*, pp. 584–592, 2014.

Bartlett, P. L., Evans, S. N., and Long, P. M. Representing smooth functions as compositions of near-identity functions with implications for deep network optimization, 2018a. URL http://arxiv.org/abs/1804.05012.

Bartlett, P. L., Helmbold, D. P., and Long, P. M. Gradient descent efficiently learns positive definite deep linear residual networks, 2018b. URL https://arxiv.org/abs/1802.06093.

Boyd, S. P. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. URL http://www.stanford.edu/~boyd/cvxbook.html.

Brutzkus, A. and Globerson, A. Globally optimal gradient descent for a convnet with gaussian inputs. In *ICML*, 2017.

Brutzkus, A., Globerson, A., Malach, E., and Shalev-Shwartz, S. SGD learns over-parameterized networks that provably generalize on linearly separable data. *ICLR*, 2018.

Culver, W. J. On the existence and uniqueness of the real logarithm of a matrix. *Proceedings of the American Mathematical Society*, 17(5):1146–1151, 1966.

Daniely, A. SGD learns the conjugate kernel class of the network. *NIPS*, 2017.

Daniely, A., Frostig, R., and Singer, Y. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *NIPS*, 2016.

Ge, R., Jin, C., and Zheng, Y. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. *arXiv preprint arXiv:1704.00708*, 2017a.

Ge, R., Lee, J. D., and Ma, T. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017b.

Ge, R., Lee, J. D., and Ma, T. Learning one-hidden-layer neural networks with landscape design. *ICLR*, 2018.

Hardt, M. and Ma, T. Identity matters in deep learning. *ICLR*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Horn, R. A. and Johnson, C. R. *Matrix analysis*. Cambridge University Press, 2013. Second edition.

Janzamin, M., Sedghi, H., and Anandkumar, A. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

Kawaguchi, K. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, pp. 586–594, 2016.

Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient descent only converges to minimizers. In *COLT*, 2016.

Lee, W. S., Bartlett, P. L., and Williamson, R. C. Efficient agnostic learning of neural networks with bounded fan-in. *IEEE Transactions on Information Theory*, 42(6): 2118–2132, 1996.

Lefkimmiatis, S., Ward, J. P., and Unser, M. Hessian schatten-norm regularization for linear inverse problems. *IEEE transactions on image processing*, 22(5):1873–1888, 2013.

Li, Y. and Yuan, Y. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pp. 597–607, 2017.

Livni, R., Shalev-Shwartz, S., and Shamir, O. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pp. 855–863, 2014.

Nguyen, Q. and Hein, M. The loss surface of deep and wide neural networks. In *ICML*, pp. 2603–2612, 2017.

Orhan, A. E. and Pitkow, X. Skip connections eliminate singularities. *ICLR*, 2018.

Safran, I. and Shamir, O. On the quality of the initial basin in overspecified neural networks. In *International Conference on Machine Learning*, pp. 774–782, 2016.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Taghvaei, A., Kim, J. W., and Mehta, P. How regularization affects the critical points in linear networks. In *NIPS*, 2017.

Zhang, Q., Panigrahy, R., and Sachdeva, S. Electron-proton dynamics in deep learning. *ITCS*, 2018.

Zhang, Y., Lee, J. D., and Jordan, M. I. l1-regularized neural networks are improperly learnable in polynomial time. In *International Conference on Machine Learning*, pp. 993–1001, 2016.

Zhang, Y., Lee, J., Wainwright, M., and Jordan, M. On the learnability of fully-connected neural networks. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 83–91, 2017.

Zhong, K., Song, Z., Jain, P., Bartlett, P. L., and Dhillon, I. S. Recovery guarantees for one-hidden-layer neural networks. In *ICML*, pp. 4140–4149, 2017.