Unique Properties of Flat Minima in Deep Networks

Rotem Mulayoff [1] **Tomer Michaeli** [1]

Abstract

It is well known that (stochastic) gradient descent has an implicit bias towards flat minima. In deep neural network training, this mechanism serves to screen out minima. However, the precise effect that this has on the trained network is not yet fully understood. In this paper, we characterize the flat minima in linear neural networks trained with a quadratic loss. First, we show that linear ResNets with zero initialization necessarily converge to the flattest of all minima. We then prove that these minima correspond to nearly balanced networks whereby the gain from the input to any intermediate representation does not change drastically from one layer to the next. Finally, we show that consecutive layers in flat minima solutions are coupled. That is, one of the left singular vectors of each weight matrix, equals one of the right singular vectors of the next matrix. This forms a distinct path from input to output, that, as we show, is dedicated to the signal that experiences the largest gain end-to-end. Experiments indicate that these properties are characteristic of both linear and nonlinear models trained in practice.

1. Introduction

Optimization methods can have implicit biases towards certain solutions (Strand, 1974; Morgan & Bourlard, 1990; Neyshabur et al., 2014). In the context of deep network training, such biases have been shown to play key roles in shaping the properties of the learned model. For example, in binary classification of linearly separable data, among all linear separators that achieve the global minimum of the training loss, gradient descent (GD) converges to the maximum margin separator. This is true for shallow networks (Soudry et al., 2018), as well as for deep linear fully-connected models (Gunasekar et al., 2018b) and deep nonlinear networks with homogeneous activation functions (Lyu & Li, 2020). Implicit biases have been studied in many different context, including for linear convolutional networks (Gunasekar et al., 2018b), matrix factorization (Gunasekar et al., 2017), weight normalization (Wu et al., 2019), and with different loss functions (Gunasekar et al., 2018a).

Perhaps the simplest mechanism through which GD and stochastic GD (SGD) can screen out solutions, is their inability to stably converge to sharp minima (Jastrzębski et al., 2017; Wu et al., 2018; Simsekli et al., 2019). In fact, in some cases, GD can only converge to the flattest of all minima (see Section 2). However, interestingly, the effect that this has on the resulting trained model, is not yet fully understood. Keskar et al. (2016) suggested that flat minima tend to generalize better. This was somewhat supported by follow up works, showing that in SGD, larger step sizes and smaller batch sizes impose convergence to flatter minima, which indeed generalize better empirically (Jastrzębski et al., 2017; Hoffer et al., 2017; Masters & Luschi, 2018; Smith & Le, 2017). However, Dinh et al. (2017) showed that for networks with ReLU activations, a re-parametrization of the weights can make any minimum arbitrarily sharper (without affecting generalization). This suggests that minimum sharpness is not directly related to generalization, thus begging the question: What *does* the sharpness of the minimum affect?

Our goal in this paper is to unveil the properties of flat minima in deep neural networks. We specifically focus on linear models trained with a quadratic loss and define the sharpness of a minimum to be its maximal Hessian eigenvalue, which is the factor affecting stable convergence of GD and SGD. We start by showing that all minima become sharper as the network gets deeper. We discuss and illustrate the implications this has on the training process. We then move on to study the flattest minimum solutions. We prove that these networks possess a special structure, whereby the gain from the input to any intermediate layer is well behaved. Furthermore, consecutive layers in those solutions are coupled, forming a distinct path from input to output, which is dedicated to the signal that experiences the largest gain end-to-end. Interestingly, similar properties were recently shown to arise in deep linear networks for binary classification (Ji & Telgarsky, 2019). However, in our case of vector-valued regression, the behaviors turn out to be more complex. We

[1] Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel. Correspondence to: Rotem Mulayoff <rotem.mulayof@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

empirically illustrate that the properties we predict are also characteristic of nonlinear networks trained in practice.

2. Problem Setting and Motivation

Consider an m-layer linear network whose jth layer performs multiplication by $\boldsymbol{W}_j \in \mathbb{R}^{d_j \times d_{j-1}}$. The end-to-end function $f_{\boldsymbol{w}} : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_y}$ implemented by this network is

$$f_{\boldsymbol{w}}(x) = \boldsymbol{W}_m \boldsymbol{W}_{m-1} \cdots \boldsymbol{W}_1 x, \tag{1}$$

where we denoted $\boldsymbol{w} = \text{vec}\left([\boldsymbol{W}_1, \boldsymbol{W}_2, \dots, \boldsymbol{W}_m]\right) \in \mathbb{R}^N$. Here, $N = \sum_{j=1}^m d_j \times d_{j-1}$ and we use the convention that $d_0 = d_x$ and $d_m = d_y$. To ensure that the network can implement any linear function from $\mathbb{R}^{d_x}$ to $\mathbb{R}^{d_y}$, we assume that the dimensions of the internal representations are not smaller than those of the input or output, namely $\min_j\{d_j\} \geq \min\{d_x, d_y\}$.

We focus on the quadratic training loss

$$\ell(\boldsymbol{w}) = \hat{\mathbb{E}}\left[\left\|y - f_{\boldsymbol{w}}(x)\right\|^2\right], \tag{2}$$

where $\hat{\mathbb{E}}$ denotes empirical mean over paired examples $\{(x_i, y_i)\}_{i=1}^n$. Note that if the input x lies in a low dimensional subspace, (*e.g.* if the number of training examples n is smaller than the ambient dimension d_x), then there exist directions $\tilde{\boldsymbol{w}}$ in parameter space such that $\ell(\boldsymbol{w}) = \ell(\boldsymbol{w} + \alpha\tilde{\boldsymbol{w}})$ for every $\boldsymbol{w}$ and every $\alpha \in \mathbb{R}$. Minima that differ along these directions may correspond to different end-to-end functions, yet they have the exact same loss landscape around them. This implies that the sharpness of a minimum is indifferent to the end-to-end function in our setting, and in particular it is not associated with generalization. In our scenario, the sharpness criterion is only sensitive to *different implementations* of the same end-to-end function.

In light of this understanding, we assume that the empirical second-order moment matrix of x, denoted by $\hat{\boldsymbol{\Sigma}}_x$, is full rank. In this case, the end-to-end function minimizing the loss is unique and can be written as $f_{\boldsymbol{w}^*}(x) = \boldsymbol{T}x$, where

$$\boldsymbol{T} = \hat{\boldsymbol{\Sigma}}_{yx} \hat{\boldsymbol{\Sigma}}_x^{-1} \tag{3}$$

with $\hat{\boldsymbol{\Sigma}}_{yx}$ denoting the empirical cross second-order moment between y and x. Thus, the set of global minima of $\ell(\boldsymbol{w})$ is

$$\Omega = \left\{\boldsymbol{w} \in \mathbb{R}^N \ : \ \boldsymbol{W}_m \boldsymbol{W}_{m-1} \cdots \boldsymbol{W}_1 = \boldsymbol{T}\right\}. \tag{4}$$

Among all minima in Ω, GD and SGD can only stably converge to the flat ones (see App. I). Specifically, denote by $\boldsymbol{H}_{\boldsymbol{w}}$ the Hessian matrix of $\ell(\boldsymbol{w})$ at $\boldsymbol{w}$ and define the sharpness of a minimum point $\boldsymbol{w}^*$ to be $\lambda_{\max}\big(\boldsymbol{H}_{\boldsymbol{w}^*}\big)$. Then $\boldsymbol{w}^*$ is not stable for GD and SGD if its sharpness is larger than $2/\eta$, where η is the step-size (Wu et al., 2018). In other words, the larger the step size, the smaller the set of minima that are accessible by the optimizer. Particularly, when using the largest step size allowing convergence, we can only reach elements in the set of *flattest* global minima,

$$\Omega_0 = \underset{\boldsymbol{w} \in \Omega}{\text{argmin}} \ \lambda_{\max}\big(\boldsymbol{H}_{\boldsymbol{w}}\big). \tag{5}$$

Our goal in this paper is to unveil the properties of solutions in Ω_0. Our motivation for doing so goes beyond large step-size training. Indeed, in many cases convergence to a point in Ω_0 is guaranteed also with a small step-size. For example, we have the following result for gradient flow (GD with an infinitesimal step size) and for GD with a small step size (see proof in App. II).

Lemma 1. *Assume that $\hat{\boldsymbol{\Sigma}}_x = \boldsymbol{I}$, $d_y = d_x$, and that the weight matrices are all square and initialized to $\boldsymbol{I}$. Then:*

i. Gradient flow can only converge to a flattest minimum.

ii. If $\boldsymbol{T}$ is positive definite and its top singular value is $\sigma_{\max}(\boldsymbol{T})$, then GD with step size $\eta \leq \frac{1}{2m}\min\{1, (\sigma_{\max}(\boldsymbol{T}))^{-2(1-\frac{1}{m})}\}$ necessarily converges to a flattest minimum at a linear rate.

Note the relevance of this lemma to the practice of zero initialization for residual networks (ResNets) (Zhang et al., 2018). Indeed, linear networks with identity initialization can be viewed as linear ResNets with zero initialization.

3. Warm-Up: Scalar Networks

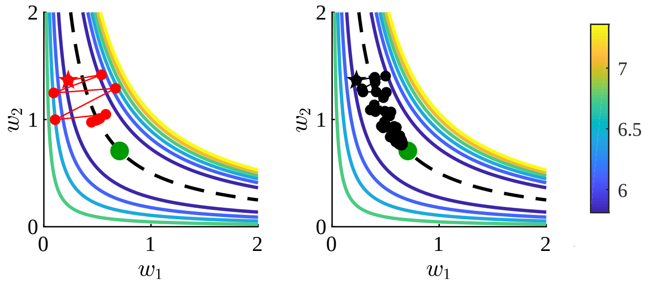
Before we present our main results, it is insightful to examine the simple case where the input, output and all intermediate representations, are scalars. In this case, the end-to-end function $f_{\boldsymbol{w}}(x)$ is given by

$$f_{\boldsymbol{w}}(x) = \prod_{j=1}^m w_j x, \tag{6}$$

where $\boldsymbol{w} = [w_1, w_2, \dots, w_m]^T \in \mathbb{R}^m$, and the quadratic loss is minimized when $f_{\boldsymbol{w}^*}(x) = \tau x$, with $\tau = \hat{\sigma}_{xy}/\hat{\sigma}_x^2$. Thus, the set of global minima is given by

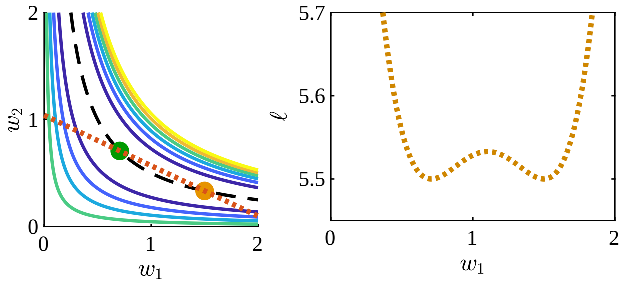
$$\Omega = \Big\{\boldsymbol{w} \in \mathbb{R}^m \ : \ \prod_{j=1}^m w_j = \tau\Big\}. \tag{7}$$

Observe that these global minima lie within connected valleys. For example, in the case of two layers, Ω corresponds to the hyperbola $w_2 = \tau/w_1$, shown in Fig. 1. Parts of these valleys are sharper than others, and as the theory predicts, GD indeed does not converge to a narrow part of the valley, even when initialized nearby such a global minimum.



(a) GD without momentum (b) GD with momentum

Figure 1. Level sets of the loss for a two-layer scalar network. The dashed line corresponds to the set of global minima Ω, and the green dot to the set of flattest minima Ω_0. When GD is initialized nearby a sharp minimum (star), it does not converge to that minimum, and rather traverses the valley of minima until reaching a flat enough point. This occurs both with and without momentum.



(a) Minima interpolation (b) Loss along the dashed line

Figure 2. A two-layer scalar network example for the misleading nature of minima interpolation. (a) We compute the loss along the (dashed) line connecting two global minima, one flattest (green) and one sharp (orange). (b) Despite having different sharpness in $\mathbb{R}^2$, their sharpness along this 1D cross section are the same. In this setting, this occurs for *any* choice of the non-flattest solution (orange point).

Direct computation (see App. III.1) shows that for $\boldsymbol{w} \in \Omega$,

$$\frac{\partial^2 \ell(\boldsymbol{w})}{\partial w_q \partial w_k} = \frac{2\hat{\sigma}_x^2 \tau^2}{w_k w_q}. \tag{8}$$

Therefore, letting $\boldsymbol{z} = [w_1^{-1}, w_2^{-1}, \ldots, w_m^{-1}]^T$, we can express the Hessian matrix at a global minimum as

$$\boldsymbol{H_w} = 2\hat{\sigma}_x^2 \tau^2 \boldsymbol{z}\boldsymbol{z}^T. \tag{9}$$

Evidently, the Hessian for scalar networks is a rank-one matrix whose (single) nonzero eigenvalue is

$$\lambda_{\max}\big(\boldsymbol{H_w}\big) = 2\hat{\sigma}_x^2 \tau^2 \|\boldsymbol{z}\|^2 = 2\hat{\sigma}_x^2 \tau^2 \sum_{j=1}^{m} \frac{1}{w_j^2}. \tag{10}$$

To determine the flattest minima, we need to seek for the weights that minimize $\lambda_{\max}\big(\boldsymbol{H_w}\big)$. This boils down to solving the constrained optimization problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^m} \sum_{j=1}^{m} \frac{1}{w_j^2} \qquad \text{s.t.} \qquad \prod_{j=1}^{m} w_j = \tau. \tag{11}$$

As we show in App. III.1, the minimum of this problem is attained when $|w_1| = |w_2| = \cdots = |w_m|$, so that the set of flattest minima is given by

$$\Omega_0 = \left\{ \boldsymbol{w} \ : \ |w_j| = |\tau|^{\frac{1}{m}}, \ \prod_{j=1}^{m} \operatorname{sgn}(w_j) = \operatorname{sgn}(\tau) \right\}. \tag{12}$$

Substituting $|w_j| = |\tau|^{1/m}$ into (10), we obtain that the sharpness of the flattest minima is given by

$$\min_{\boldsymbol{w} \in \boldsymbol{R}^m} \lambda_{\max}(\boldsymbol{H_w}) = 2m\hat{\sigma}_x^2 \tau^{2(1-\frac{1}{m})}. \tag{13}$$

Note that although there exist infinitely many global minima, there are far fewer flattest minima. Specifically, we see that for scalar networks, Ω_0 is a discrete set of cardinality 2^{m-1}. Geometrically speaking, within each connected valley of global minima, we have only one flattest minimum point. This property carries over to the vector case, in the sense that Ω_0 is always a set of measure zero within Ω.

This simple exercise of analyzing scalar networks already reveals several interesting properties of flat minima.

1. **Balancedness.** Note from (12) that the flattest minima correspond to networks, which are balanced in the sense that all their layers have the same weight magnitude. This property turns out to break in higher dimensions. However, as we will see, the flattest solutions are always at least *nearly* balanced, and they exhibit interesting coupling properties.

2. **Step-size and depth.** Observe from (13) that the sharpness of the flattest minima scales roughly linearly with the network's depth, m. Thus, the deeper the network, the smaller the maximal step-size that allows convergence. As we will see, this property persists in higher dimensions. Interestingly, although this behavior is known (Nar & Sastry, 2018), it has not been previously derived from minima sharpness considerations.

3. **Valley dimensions.** We saw that the Hessian at a global minimum is always rank-1. This implies that at every minimum point, $m-1$ orthogonal directions point into the valley, whereas only one direction points to an ascent slope. We will see that a similar phenomenon occurs also in higher dimensions.

Besides providing a glimpse into the nature of flat minima, the analysis of scalar networks also allows to assess the effectiveness of visualization methods. Particularly, it is common practice to visually compare the sharpness of two minima, $\boldsymbol{w}^{(1)}$ and $\boldsymbol{w}^{(2)}$, by plotting the loss along the line connecting them (Keskar et al., 2016; Jastrzębski et al., 2017). One expects that a flat minimum would appear flatter also along this 1D cross-section. However, our scalar network analysis reveals that this is typically incorrect. Let us first take a two-layer example. Figure 2 shows the loss along the line connecting a flattest minimum point $\boldsymbol{w}^{(1)}$ and a sharper one, $\boldsymbol{w}^{(2)}$. As can be seen, along this cross-section, both minima have the same sharpness. This is not a result of some particular choice of $\boldsymbol{w}^{(2)}$. It turns out that for two-layer scalar networks, the minimas' sharpness along this cross section are always the same, regardless of how sharp $\boldsymbol{w}^{(2)}$ is in practice. For deeper scalar networks, this is not always the case. However, this visualization is still frequently deceiving (see App. III.2).

Lemma 2. *Consider a scalar linear network. Let $\boldsymbol{w}^{(1)}$ be a flattest minimum and $\boldsymbol{w}^{(2)}$ be some other minimum that has the same sign pattern as $\boldsymbol{w}^{(1)}$. If the interpolation visualization shows that $\boldsymbol{w}^{(2)}$ is sharper than $\boldsymbol{w}^{(1)}$, then there exists another minimum, $\boldsymbol{w}^{(3)}$, which the visualization would show is rather flatter than $\boldsymbol{w}^{(1)}$.*

As we empirically show in Sec. 6, this phenomenon is common also in non-scalar networks with ReLU activations.

4. Main Results

We now move on to the general case of non-scalar deep linear networks. To simplify notations, we denote

$$\prod_{j=q}^{k} \boldsymbol{W}_j \triangleq \boldsymbol{W}_k \boldsymbol{W}_{k-1} \cdots \boldsymbol{W}_q, \tag{14}$$

where a product over an empty set ($q > k$) is defined to be the identity matrix $\boldsymbol{I}$. We make the following assumptions.

A1 The network has the capacity to implement any linear function from $\mathbb{R}^{d_x}$ to $\mathbb{R}^{d_y}$, namely $\min\{d_i\} \geq \min\{d_x, d_y\}$.

A2 The data is white, namely $\hat{\boldsymbol{\Sigma}}_x = \boldsymbol{I}$.

We begin by identifying the structure of the Hessian matrix at a global minimum (see App. IV).

Lemma 3 (Hessian structure)**.** *Assume A1. If $\boldsymbol{w} \in \Omega$, then*

$$\boldsymbol{H}_{\boldsymbol{w}} = 2\boldsymbol{\Phi}\boldsymbol{\Phi}^T, \tag{15}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1^T, \boldsymbol{\Phi}_2^T, \ldots, \boldsymbol{\Phi}_m^T]^T$, with

$$\boldsymbol{\Phi}_k = \left(\prod_{j=1}^{k-1} \boldsymbol{W}_j \hat{\boldsymbol{\Sigma}}_x^{\frac{1}{2}} \right) \otimes \left(\prod_{i=k+1}^{m} \boldsymbol{W}_i \right)^T. \tag{16}$$

Here $\otimes$ denotes the Kronecker product.

Note that $\boldsymbol{\Phi}_k$ is a $d_k d_{k-1} \times d_x d_y$ matrix. Therefore, $\boldsymbol{\Phi}$ has only $d_x d_y$ columns, while its number of rows is the total number of parameters in the net, $N = \sum_{k=1}^{m} d_k d_{k-1}$. This shows that for networks with more than one layer, the Hessian at a global minimum is always rank-deficient. For example, if $d_x = d_y \triangleq d$, then we have from Assumption A1 that $N \geq md^2$, so that at any global minimum point, only d^2 orthogonal directions point to a slope (the Hessian's rank), while the rest point into the valley of minima. In other words, the dimension of the valley is at least $(1 - 1/m)$ of the ambient dimension N.

In analogy with the scalar setting, we would now like to exploit Lemma 3 for analyzing the set of flattest minima, Ω_0. Unfortunately, here it is intractable to derive a closed form expression for $\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}})$ at an arbitrary minimum point. Yet, our key observation is that it is still possible to determine the minimal value of $\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}})$ over the set of global minima Ω, as well as its associated eigenvector. That is, we can deduce the sharpness of the flattest minima, without having an explicit expression for the sharpness of arbitrary minima. We elaborate on the proof technique in Sec. 5. Specifically, let $\sigma_{\max}(\boldsymbol{T})$ denote the top singular value of $\boldsymbol{T}$, and let $\boldsymbol{u}$ and $\boldsymbol{v}$ be its corresponding left and right singular vectors. Then we have the following.

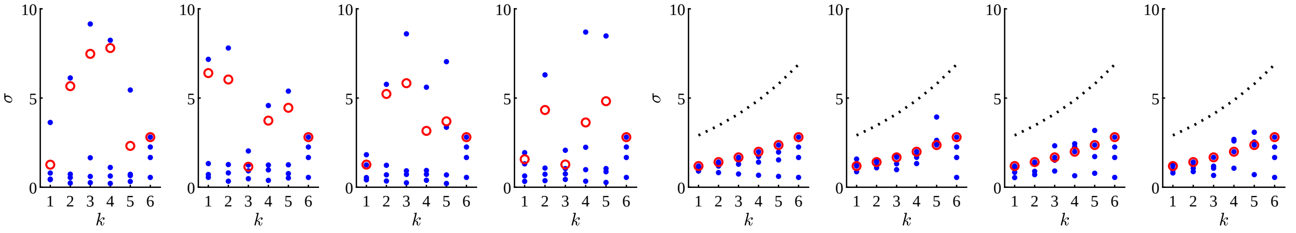
Theorem 1 (Sharpness of flattest minima)**.** *Assume A1 and A2. If $\boldsymbol{w} \in \Omega_0$ then*

$$\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}}) = 2m \times (\sigma_{\max}(\boldsymbol{T}))^{2(1-\frac{1}{m})}, \tag{17}$$

and the corresponding eigenvector is $\boldsymbol{b} = \boldsymbol{\Phi}\big(\boldsymbol{v} \otimes \boldsymbol{u}\big)$.

This result asserts that the flattest minima become sharper as the number of layers increases (their sharpness grows approximately linearly with m for $m \gg 1$). Since a minimum point $\boldsymbol{w}^*$ is stable for GD if the step-size satisfies $\eta \leq 2/\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}^*})$, we conclude that the maximal step-size allowing convergence satisfies $\eta_{\max} \leq \frac{1}{m}(\sigma_{\max}(\boldsymbol{T}))^{-2(1-1/m)}$. In other words, the step-size should be taken to be smaller when training deeper models. As mentioned above, this result was also deduced by Nar & Sastry (2018), albeit from different considerations (without explicitly analyzing minima sharpness).

Next, we turn to analyze the flattest minima in terms of the gain that signals experience as they propagate through these networks. For general minimum points, the largest end-to-end gain is $\sigma_{\max}(\boldsymbol{T})$ (corresponding to the input $\boldsymbol{v}$), but the intermediate gain up to layer $k < m$ is unconstrained, as we can always multiply one weight matrix by α and another by $1/\alpha$ without affecting the end-to-end mapping. Flattest minima, however, have special structures. Two questions are thus in place regarding those solutions: (i) What gain does $\boldsymbol{v}$ experience up to layer k? (ii) What is the largest gain that *any signal* can experience up to layer k?



(a) Arbitrary global minima

(b) Flattest global minima

Figure 3. Intermediate gains versus layer number in deep linear networks. Here we visualize eight randomly chosen implementations of the same end-to-end function $\boldsymbol{T}$, where the dimension d is 4, and the number of layers m is always 6. For each depth k, the blue dots depict the singular values of the product of weight matrices from 1 to k, and the red circle corresponds to the gain of the top singular vector of $\boldsymbol{T}$. The black doted line corresponds the bound of Theorem 2(ii). (a) For arbitrary global minima, the intermediate gains can be high. (b) For flattest solutions, the maximal intermediate gain is well behaved, and $\boldsymbol{v}$ is a singular vector of all partial matrix products.

Theorem 2 (Intermediate gains)**.** *Assume A1 and A2. If $\boldsymbol{w} \in \Omega_0$ then for all k:*

i. *$\boldsymbol{v}$ is a right singular vector of $\prod_{j=1}^{k} \boldsymbol{W}_j$ with corresponding singular value $(\sigma_{\max}(\boldsymbol{T}))^{\frac{k}{m}}$.*

ii. $\sigma_{\max}(\prod_{j=1}^{k} \boldsymbol{W}_j) \leq \sqrt{m} \times (\sigma_{\max}(\boldsymbol{T}))^{\frac{k}{m}}$.

Similarly,

iii. *$\boldsymbol{u}$ is a left singular vector of $\prod_{j=k+1}^{m} \boldsymbol{W}_j$ with corresponding singular value $(\sigma_{\max}(\boldsymbol{T}))^{1-\frac{k}{m}}$.*

iv. $\sigma_{\max}(\prod_{j=k+1}^{m} \boldsymbol{W}_j) \leq \sqrt{m} \times (\sigma_{\max}(\boldsymbol{T}))^{1-\frac{k}{m}}$.

Figure 3 illustrates the theorem for six-layer linear networks designed to solve a linear regression problem involving synthetic data (see App. IX for details). Here $\boldsymbol{W}_j \in \mathbb{R}^{4\times 4}$ for all layers. The figure depicts eight randomly drawn global minima, four arbitrary and four flattest. The intermediate gain of $\boldsymbol{v}$, to which Theorem 2(i) refers, is marked by red circles. The bound of Theorem 2(ii) is shown as a dotted black line, and the singular values of the partial matrix products are marked by blue dots. As can be seen in Fig. 3(a), the intermediate gains in arbitrary global minimum solutions can be high. However, in the flattest solutions (Fig. 3(b)), the gain that $\boldsymbol{v}$ experiences varies gracefully along the net (as $(\sigma_{\max}(\boldsymbol{T}))^{k/m}$), and the maximal gain of any other signal (highest blue point) is never much larger. Finally, we see that $\boldsymbol{v}$ is indeed one of the singular vectors of the partial product matrix up to any depth (as the red circle coincides with one of the blue points in each layer).

In addition to the intermediate gains, it is of interest to analyze the individual weight matrices. It turns out that in the flattest solutions, the layers exhibit a sort of coupling associated with the signal $\boldsymbol{v}$. Specifically, we have the following.

Theorem 3 (Layer coupling)**.** *Assume A1 and A2. Denote $\boldsymbol{r}_k = \prod_{j=1}^{k-1} \boldsymbol{W}_j \boldsymbol{v}$, $\boldsymbol{q}_k = (\prod_{j=k+1}^{m} \boldsymbol{W}_j)^T \boldsymbol{u}$, and write $\bar{\boldsymbol{r}}_k = \boldsymbol{r}_k/\|\boldsymbol{r}_k\|$, $\bar{\boldsymbol{q}}_k = \boldsymbol{q}_k/\|\boldsymbol{q}_k\|$. If $\boldsymbol{w} \in \Omega_0$ then for all k:*

i. *$\bar{\boldsymbol{q}}_k$ and $\bar{\boldsymbol{r}}_k$ are a pair of left and right singular vectors of $\boldsymbol{W}_k$ with corresponding singular value $(\sigma_{\max}(\boldsymbol{T}))^{\frac{1}{m}}$.*

ii. *These vectors are coupled in the sense that $\bar{\boldsymbol{r}}_{k+1} = \bar{\boldsymbol{q}}_k$.*

Remark: From Theorem 2, $\|\boldsymbol{r}_k\| = (\sigma_{\max}(\boldsymbol{T}))^{(k-1)/m}$ and $\|\boldsymbol{q}_k\| = (\sigma_{\max}(\boldsymbol{T}))^{1-k/m}$.

Theorem 3 indicates that in flattest minimum networks, there forms a distinct path from input to output that is exclusively dedicated to the signal $\boldsymbol{v}$. Specifically, when such a network operates on $\boldsymbol{v}$, the input to each layer is a singular vector of that layer, with singular value $(\sigma_{\max}(\boldsymbol{T}))^{1/m}$. Note that this singular value is not necessarily the maximal one of each layer, but it must exist in all matrices. Now, since the input of each layer is a singular vector, so is its output. Therefore, we have that consecutive layers in the network have a singular vector in common, where a left singular vector of one matrix matches a right singular vector of the next.

We saw that if $\boldsymbol{w} \in \Omega_0$, then one of the singular values of each weight matrix must equal $(\sigma_{\max}(\boldsymbol{T}))^{1/m}$. One may wonder whether the other direction is also true. As we now show, if the singular value $(\sigma_{\max}(\boldsymbol{T}))^{1/m}$ not only exists, but is also the *largest* one of each matrix, then the network is necessarily a flattest minimum.

Theorem 4 (Sufficient condition)**.** *Assume A1 and A2. If a solution $\boldsymbol{w} \in \Omega$ satisfies $\sigma_{\max}(\boldsymbol{W}_k) = (\sigma_{\max}(\boldsymbol{T}))^{\frac{1}{m}}$ for all k, then necessarily $\boldsymbol{w} \in \Omega_0$.*

Observe that these cases are not rare, in the sense that they form a set of nonzero measure within Ω_0.

5. Proof Outline for Theorem 1

Our results in theorems 1-4 hinge on the ability to characterize the flattest minima without having an explicit expression for the top eigenvalue of the Hessian at an arbitrary minimum point. In this section we present an outline of the proof of Theorem 1, which illustrates how we go about this, and lays the basis for the proofs of the other theorems.

Note from (1) that $\boldsymbol{w}$ is a concatenation of the vectorizations of the weights matrices $\{\boldsymbol{W}_j\}$. That is, denoting $\boldsymbol{w}_j = \text{vec}\,(\boldsymbol{W}_j)$, we have that $\boldsymbol{w} = [\boldsymbol{w}_1^T, \boldsymbol{w}_2^T, \dots, \boldsymbol{w}_m^T]^T$. Therefore, the Hessian has the following block structure,

$$\boldsymbol{H}_{\boldsymbol{w}} = \begin{bmatrix} \frac{\partial^2}{\partial \boldsymbol{w}_1 \partial \boldsymbol{w}_1} & \frac{\partial^2}{\partial \boldsymbol{w}_1 \partial \boldsymbol{w}_2} & \cdots & \frac{\partial^2}{\partial \boldsymbol{w}_1 \partial \boldsymbol{w}_m} \\ \frac{\partial^2}{\partial \boldsymbol{w}_2 \partial \boldsymbol{w}_1} & \frac{\partial^2}{\partial \boldsymbol{w}_2 \partial \boldsymbol{w}_2} & \cdots & \frac{\partial^2}{\partial \boldsymbol{w}_2 \partial \boldsymbol{w}_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial \boldsymbol{w}_m \partial \boldsymbol{w}_1} & \frac{\partial^2}{\partial \boldsymbol{w}_m \partial \boldsymbol{w}_2} & \cdots & \frac{\partial^2}{\partial \boldsymbol{w}_m \partial \boldsymbol{w}_m} \end{bmatrix} \ell(\boldsymbol{w}), \quad (18)$$

where we use denominator-layout notation. In App. IV we show that if $\boldsymbol{w} \in \Omega$, then the (i,j)th block is given by

$$\frac{\partial^2}{\partial \boldsymbol{w}_i \partial \boldsymbol{w}_j} \ell(\boldsymbol{w}) = 2 \boldsymbol{\Phi}_i \boldsymbol{\Phi}_j^T, \quad (19)$$

where $\boldsymbol{\Phi}_i$ is defined in (16). This implies that we can write $\boldsymbol{H}_{\boldsymbol{w}} = 2\boldsymbol{\Phi}\boldsymbol{\Phi}^T$, where $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1^T, \boldsymbol{\Phi}_2^T, \dots, \boldsymbol{\Phi}_m^T]^T$.

To study the maximal eigenvalue of the Hessian, we will be rather looking at the matrix $\hat{\boldsymbol{H}}_{\boldsymbol{w}} = 2\boldsymbol{\Phi}^T\boldsymbol{\Phi}$, whose nonzero eigenvalues coincide with those of $\boldsymbol{H}_{\boldsymbol{w}}$. Particularly,

$$\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}}) = \lambda_{\max}(\hat{\boldsymbol{H}}_{\boldsymbol{w}}) = \max_{\|\boldsymbol{b}\|=1} 2\|\boldsymbol{\Phi}\boldsymbol{b}\|^2. \quad (20)$$

Using the fact that $\|\boldsymbol{\Phi}\boldsymbol{b}\|^2 = \sum_{k=1}^m \|\boldsymbol{\Phi}_k \boldsymbol{b}\|^2$, together with properties of the Kronecker product (that appears in the definition of $\boldsymbol{\Phi}_k$), the right side of (20) can be written as

$$\max_{\|\boldsymbol{B}\|_{\mathrm{F}}=1} 2 \sum_{k=1}^m \left\| \Big(\prod_{i=k+1}^m \boldsymbol{W}_i \Big)^T \boldsymbol{B} \hat{\boldsymbol{\Sigma}}_x^{\frac{1}{2}} \Big(\prod_{j=1}^{k-1} \boldsymbol{W}_j \Big)^T \right\|_{\mathrm{F}}^2, \quad (21)$$

where $\boldsymbol{b} = \text{vec}\,(\boldsymbol{B})$ (see App. V.3). Obtaining a closed form solution to this optimization problem seems intractable. However, recall that our goal is merely to find the minimal value of $\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}})$ over $\boldsymbol{w} \in \Omega$. This corresponds to a minimax optimization problem over $\boldsymbol{w}$ and $\boldsymbol{B}$, where the minimum is taken over $\boldsymbol{w} \in \Omega$ and the maximum over $\boldsymbol{B} \in \{\boldsymbol{B} \in \mathbb{R}^{d_y \times d_x} : \|\boldsymbol{B}\|_{\mathrm{F}} = 1\}$.

Our solution approach consists of two steps. First, we bound the objective from below using an expression that is independent of $\boldsymbol{w}$. Then, we show that there exists a particular choice of $\boldsymbol{w} \in \Omega$ that achieves the lower bound. This proves that our bound is in fact the minimax value (*i.e.* the minimal value of $\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}})$ over Ω). To this end, we make use of the following lemma (see proof in App. V.1).

Lemma 4. *Let $\{\boldsymbol{\Psi}_k\}_{k=1}^m$ be a set of matrices such that $\boldsymbol{\Psi}_k \in \mathbb{R}^{d_k \times d_{k-1}}$, then*

$$\sum_{k=1}^m \|\boldsymbol{\Psi}_k\|_{\mathrm{F}}^2 \geq m \left(\left\| \prod_{k=1}^m \boldsymbol{\Psi}_k \right\|_2 \right)^{\frac{2}{m}}, \quad (22)$$

*where $\|\cdot\|_2$ is the matrix norm induced by the ℓ_2 vector norm (*i.e. *the maximal singular value of the argument).*

This lemma implies that the objective in (21) can be lower-bounded by

$$2m \left(\left\| \prod_{k=1}^m \Big(\prod_{i=k+1}^m \boldsymbol{W}_i \Big)^T \boldsymbol{B} \hat{\boldsymbol{\Sigma}}_x^{\frac{1}{2}} \Big(\prod_{j=1}^{k-1} \boldsymbol{W}_j \Big)^T \right\|_2 \right)^{\frac{2}{m}}. \quad (23)$$

Let us write out explicitly two consecutive terms of the outer product,

$$\underbrace{\Big(\prod_{i=q+2}^m \boldsymbol{W}_i \Big)^T \boldsymbol{B} \hat{\boldsymbol{\Sigma}}_x^{\frac{1}{2}} \Big(\prod_{j=1}^{q} \boldsymbol{W}_j \Big)^T}_{k=q+1} \underbrace{\Big(\prod_{i=q+1}^m \boldsymbol{W}_i \Big)^T \boldsymbol{B} \hat{\boldsymbol{\Sigma}}_x^{\frac{1}{2}} \Big(\prod_{j=1}^{q-1} \boldsymbol{W}_j \Big)^T}_{k=q} \quad (24)$$

It is easy to see that the product of the two terms in the middle equals $(\prod_{j=1}^m \boldsymbol{W}_j)^T$, which in turn equals $\boldsymbol{T}^T$ for global minima. Therefore, if we unwrap the entire outer product, we get $\boldsymbol{T}^T$ in between every two appearances of $\boldsymbol{B}\hat{\boldsymbol{\Sigma}}_x^{\frac{1}{2}}$, so that (23) reduces to

$$\nu(\boldsymbol{B}) \triangleq 2m \left\| \Big(\boldsymbol{B} \hat{\boldsymbol{\Sigma}}_x^{\frac{1}{2}} \boldsymbol{T}^T \Big)^{m-1} \boldsymbol{B} \hat{\boldsymbol{\Sigma}}_x^{\frac{1}{2}} \right\|_2^{\frac{2}{m}}. \quad (25)$$

To recap, we have that if $\boldsymbol{w}$ is a global minimum, then $\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}}) \geq \max \nu(\boldsymbol{B})$ s.t. $\|\boldsymbol{B}\|_{\mathrm{F}} = 1$. For the special case $\hat{\boldsymbol{\Sigma}}_x = \boldsymbol{I}$ (Assumption A2), we show in App. V.2 that

$$\max_{\|\boldsymbol{B}\|_{\mathrm{F}}=1} \nu(\boldsymbol{B}) = 2m \left(\sigma_{\max}(\boldsymbol{T}) \right)^{2(1-\frac{1}{m})}. \quad (26)$$

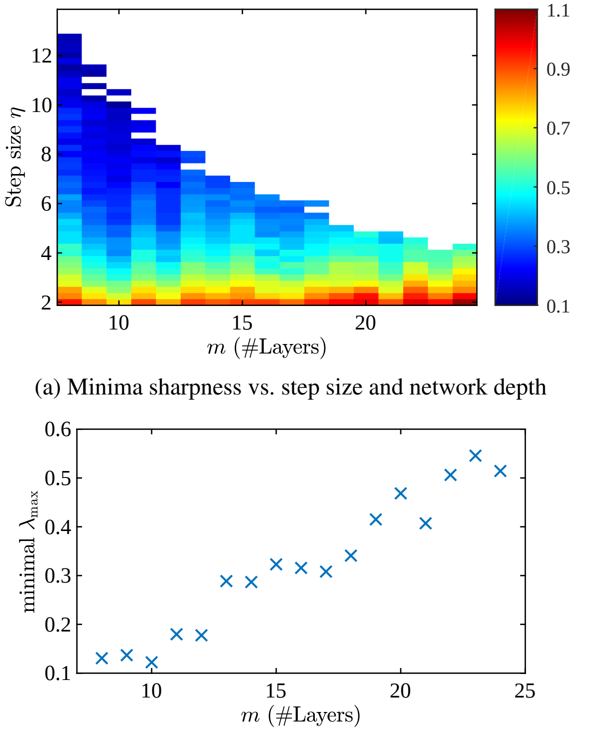
We have thus obtained a lower-bound on $\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}})$, which is independent of $\boldsymbol{w}$.

We now determine a particular solution achieving the bound. Denote the SVD of $\boldsymbol{T}$ by $\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$ and let $\boldsymbol{w}^* \in \Omega$ be

$$\boldsymbol{W}_m^* = \boldsymbol{U}\boldsymbol{S}_m^{\frac{1}{m}}, \quad \boldsymbol{W}_j^* = \boldsymbol{S}_j^{\frac{1}{m}}, \quad \boldsymbol{W}_1^* = \boldsymbol{S}_1^{\frac{1}{m}} \boldsymbol{V}^T. \quad (27)$$

Here we slightly abuse the notation $\boldsymbol{S}_j^{1/m}$ to denote a $d_j \times d_{j-1}$ diagonal matrix whose kth diagonal entry is $(\sigma_k(\boldsymbol{T}))^{1/m}$, the kth largest singular value of $\boldsymbol{T}$. Note that for this particular solution, all the weight matrices have the same set of nonzero singular values, which are precisely the mth roots of the singular values of $\boldsymbol{T}$. As we show in App. V.3, for this solution it is rather easy to compute the Hessian's top eigenvalue, which turns out to equal

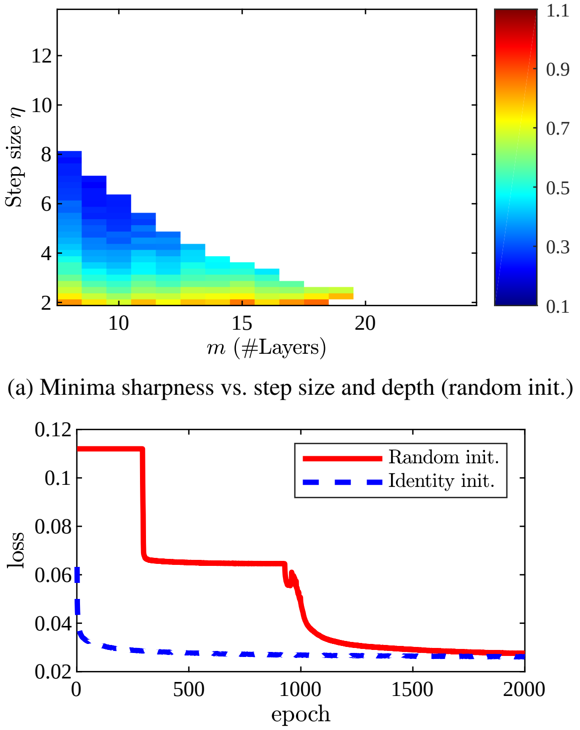
$$\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}^*}) = 2m \left(\sigma_{\max}(\boldsymbol{T}) \right)^{2(1-\frac{1}{m})}. \quad (28)$$



(a) Minima sharpness vs. step size and network depth

(b) Sharpness of the flattest minima vs. network depth

Figure 4. Sharpness of minima obtained with identity initialization in fully connected ReLU networks trained to denoise MNIST digits. (a) The color of each tile corresponds to the sharpness of the minimum to which SGD converged for a particular step size and network depth m. White tiles correspond to non-converged trainings. We can see that larger step sizes lead to flatter minima, and that the maximal step size allowing convergence behaves as $1/m$. (b) Here, we see that the sharpness of the flattest minimum (bluest tile) increases roughly linearly with m, as Theorem 1 predicts.



(a) Minima sharpness vs. step size and depth (random init.)

(b) Training plot

Figure 5. Sharpness of minima obtained with random initialization. (a) As opposed to identity initilization (Fig. 4), here the maximal step sizes allowing convergence are smaller, and the minima to which SGD converges are sharper. This aligns with the prediction of Lemma 1. (b) We plot the progression of the training loss with random and identity initializations, for an 18 layer network with step size $\eta = 2.25$. The graph demonstrates that SGD converges faster when initialized at identity.

Since $\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}^*})$ achieves the lower-bound (26), this bound must be the minimal value of $\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}})$. We have thus established that

$$\min_{\boldsymbol{w}\in\Omega} \lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}}) = 2m\left(\sigma_{\max}(\boldsymbol{T})\right)^{2(1-\frac{1}{m})}, \tag{29}$$

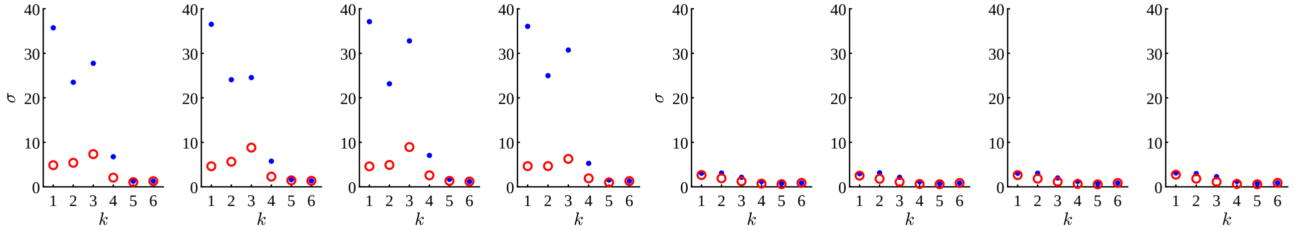
which completes the proof for $\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}})$. The proof for the corresponding eigenvector can be found in App. V.4.

Two comments are in place. First, note that as a byproduct, we obtained that the solution (27) is a flattest global minimum. This is actually a special case of Theorem 4, which applies also to non-diagonal weight matrices, and to matrices whose singular values do not all coincide with the mth roots of the singular values of $\boldsymbol{T}$. Namely, according to Theorem 4, merely requiring that $\sigma_{\max}(\boldsymbol{W}_k) = (\sigma_{\max}(\boldsymbol{T}))^{1/m}$ for all k, already guarantees that a minimum is flattest. Second, although we focused on the case $\hat{\boldsymbol{\Sigma}}_x = \boldsymbol{I}$, we conjecture that $\min_{\boldsymbol{w}\in\Omega} \lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}}) = \max_{\|\boldsymbol{B}\|_{\mathrm{F}}=1} \nu(\boldsymbol{B})$ also for arbitrary $\hat{\boldsymbol{\Sigma}}_x$. However, in the general setting, there is no closed form solution for the maximization over $\boldsymbol{B}$, so that its study seems to allow no further insight.

6. Experiments with Nonlinear Networks

Our theoretical results apply to linear networks. Yet, as we now empirically illustrate, they also nicely capture the behavior of nonlinear networks. To show this, we trained fully connected networks with ReLU activation functions to denoise images of handwritten digits. We used the MNIST dataset (LeCun, 1998) and simulated zero-mean white Gaussian noise of standard deviation 1.25, where the pixel range of the clean images was $[0, 1]$. The input, output, and all intermediate representations had 784 dimensions, so that the total number of parameters was over $600,000 \times m$ for an m-layer network. We minimized the quadratic loss using SGD without momentum.

We start by demonstrating Theorem 1, which asserts that all minima become sharper as the depth of the network in-



(a) Sharp minima (Adam)

(b) Flat minima (SGD with large step-size)

Figure 6. Intermediate gains versus layer number for six-layer fully connected ReLU networks, trained to denoise MNIST digits. The maximal gain from input to each layer is marked by a blue dot (analogous to the highest blue dot in Fig. 3). The red circles correspond to the intermediate gain of the signal that experiences the maximal end-to-end amplification. (a) The gains in the sharp minimum solutions reached by Adam, are large. (b) The gains in the flat minimum solutions found by SGD, are significantly more balanced.

creases. Figure 4 visualizes the result of training networks of different depths using varying step sizes. For each configuration, we measured the top eigenvalue of the Hessian using the power method. Thus, each tile in Fig. 4(a) corresponds to a different trained network, where $\lambda_{\max}(\boldsymbol{H}_{\boldsymbol{w}})$ is color-coded and white tiles correspond to non-converged runs. We can see that larger step sizes indeed lead to flatter minima (*i.e.* more bluish tiles). Also, notice that the maximal step size allowing convergence behaves as $1/m$, in accordance with Theorem 1. Now, for each network depth, we took the lowest measured sharpness, and plotted it against the number of layers in Fig. 4(b). Here we can see that the flattest minima indeed get sharper as the network gets deeper. Particularly, the behavior is roughly linear, as (17) predicts.

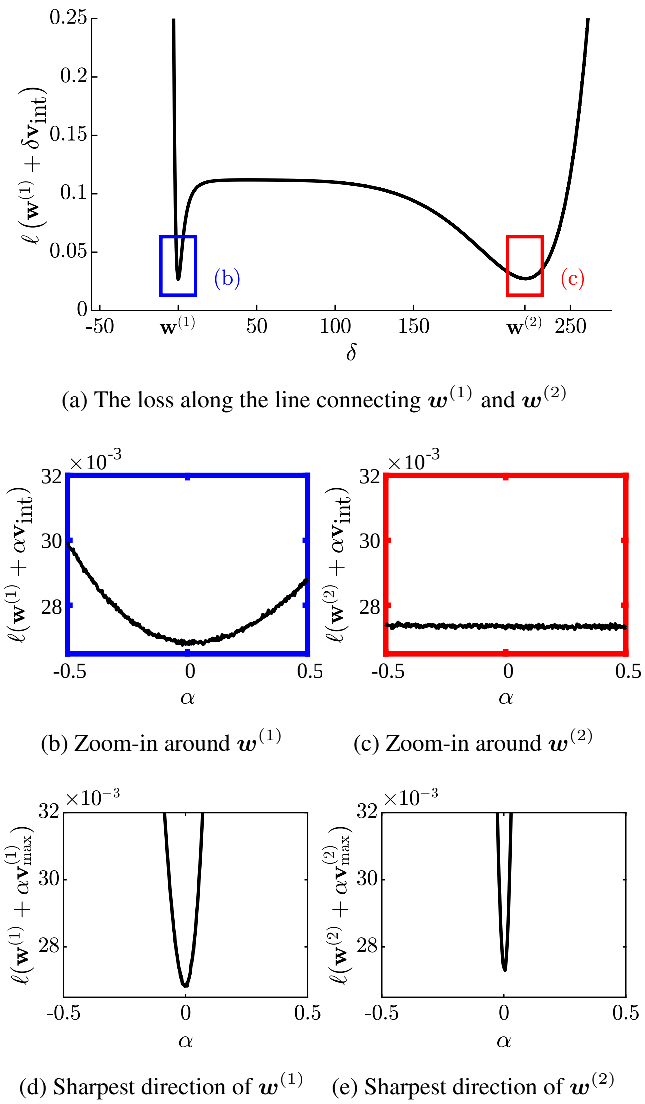
In the experiment above, we used identity initialization, as Lemma 1 suggests this should lead to a flat minimum. To verify that this is indeed the case, we repeated the experiment with the initialization of He et al. (2015). As can be seen in Fig. 5(a), in this case SGD indeed converges to sharper minima, and cannot accommodate large step sizes. To further compare these initializations, we plot the loss function during training in Fig. 5(b). We can see that with identity initialization, the loss rapidly convergence already at an early stage, whereas with random initialization it decreases only at later iterations. This shows that identity initialization indeed leads to flatter minima, as Lemma 1 predicts, and that SGD converges faster to flat solutions.

Next, we demonstrate Theorem 2. For the purpose of comparing the properties of sharp and flat solutions, we trained a six layer network for the same denoising problem as above, using two different optimization methods: (i) SGD with a large step size and moderate batch size, a configuration that is known to converge to flat minima (Keskar et al., 2016); (ii) Adam (Kingma & Ba, 2014) with a small step size, which can converge to sharp minima (Wu et al., 2018). We ran each method with four different random initializations, and calculated the top eigenvalue of the Hessian at the minimum it converged to. We verified this eigenvalue was indeed significantly smaller (roughly 6×) for the minima found by SGD. Now, for each network, we estimated the maximal gain that any signal can experience up to each layer. We did so by optimizing the input so as to maximize the norm of the intermediate signal, where we started from 100 different random initializations and chose the maximum over all runs. These gains are analogous to the top singular values of the partial matrix products in the linear setting. As can be seen in Fig. 6, these gains (blue dots) tend to be high in the sharp solutions, and quite restrained in the flat ones. A similar behavior is seen for the intermediate gains of the signal that experiences the largest end-to-end amplification (red circles). These are analogous to the intermediate gains of the vector $\boldsymbol{v}$ in the linear setting. These behaviors are in accordance with points (ii) and (i) of Theorem 2, respectively.

Finally, we illustrate Lemma 2. For each of the 16 pairs of flat and sharp minima, we evaluated the loss along the line connecting them. The result for one pair is shown in Fig. 7(a) (all 16 pairs showed the same behavior). As can be seen, the flat minimum appears to be sharper than the sharp one along this 1D cross-section. To appreciate how distorted this image is, we also plot in figures 7(d) and 7(e) the loss along the sharpest cross-section of each minimum, which visualizes its true sharpness. This illustration confirms that the interpolation visualization is frequently deceiving also for nonlinear networks in high-dimensional settings.

7. Related Work

Notions of sharpness Many works studied flat minima in neural networks, especially in relation to generalization. These minima are thought to represent simple models, which are less expected to overfit. However, there is no single definition for minimum sharpness. Hochreiter & Schmidhuber (1997) defined it as the size of the connected region around the minimum where the training loss remains low. Chaudhari et al. (2019) used local entropy as a measure of sharpness. And Keskar et al. (2016) characterized sharpness



(a) The loss along the line connecting $\boldsymbol{w}^{(1)}$ and $\boldsymbol{w}^{(2)}$

(b) Zoom-in around $\boldsymbol{w}^{(1)}$

(c) Zoom-in around $\boldsymbol{w}^{(2)}$

(d) Sharpest direction of $\boldsymbol{w}^{(1)}$

(e) Sharpest direction of $\boldsymbol{w}^{(2)}$

Figure 7. One dimensional cross-sections of the loss landscape. (a) The loss along the line connecting $\boldsymbol{w}^{(1)}$ and $\boldsymbol{w}^{(2)}$, the solutions obtained by SGD and Adam, respectively. Here, the direction vector is $\boldsymbol{v}_{\rm int}=(\boldsymbol{w}^{(2)}-\boldsymbol{w}^{(1)})/\|\boldsymbol{w}^{(2)}-\boldsymbol{w}^{(1)}\|$. Along this cross-section, $\boldsymbol{w}^{(1)}$ appears to be sharper than $\boldsymbol{w}^{(2)}$. (b), (c) Close-ups on $\boldsymbol{w}^{(1)}$ and $\boldsymbol{w}^{(2)}$, respectively. (d), (e) Cross-sections corresponding to the sharpest direction of $\boldsymbol{w}^{(1)}$ and $\boldsymbol{w}^{(2)}$, respectively, which show that $\boldsymbol{w}^{(1)}$ is in fact flatter. Here, $\boldsymbol{v}^{(i)}_{\rm max}$ is the top eigenvector of the Hessian matrix at $\boldsymbol{w}^{(i)}$.

using the eigenvalues of the Hessian, and proposed an approximation using the maximal loss in an ϵ-neighborhood of the minimum. These notions of sharpness were devised with the purpose of correlating with generalization, although Neyshabur et al. (2017) and Dinh et al. (2017) argued they often do not suffice for predicting the generalization of solutions. In contrast to these papers, the definition we studied is associated with the stability of the optimizer at the minimum. Thus, whether correlated with generalization or not for nonlinear nets, it is most relevant for the solutions found in practice by SGD.

Balancedness and alignment We showed that GD and SGD tend to converge to solutions that are balanced and aligned. Similar results were studied in different contexts. For example, it has been shown that gradient flow maintains the differences between the squared norms of the layers, both in linear networks (Arora et al., 2018) and in nonlinear models with homogeneous activation functions (Du et al., 2018). Notice, however, that as opposed to our analysis, these results break for GD with a large step size, as the authors indicated in their work. Interestingly, while our results apply to linear models trained with a quadratic loss, similar phenomena occur in deep linear networks for binary classification trained with a monotonic loss. Specifically, Ji & Telgarsky (2019) showed that in those cases gradient flow aligns the layers in the sense that the normalized matrices asymptotically equal their rank-1 approximations. Additionally, they showed that adjacent rank-1 approximations have a singular vector in common, where a left singular vector of one layer asymptotically matches a right singular vector of the next. Nevertheless, note that networks with vector-valued outputs trained for regression, as we analyzed here, exhibit richer and more complex behaviors than models with scalar outputs trained for binary classification.

Visualization of minima sharpness It is fairly common to compare the sharpness of two minima by plotting the loss along the line connecting them (Keskar et al., 2016; Jastrzębski et al., 2017). Yet, we are not the first to discuss the limitations of this common practice. For example, Li et al. (2018) argued that this may depict a misleading picture due to unnormalized weights. Instead, they offered to plot the loss on a randomly chosen 2D cross-section, where the perturbation is normalized with respect to the weights. Here, we gave a concrete example along with a proof that the interpolation visualization is deceiving surprisingly often.

8. Conclusion

Gradient descent methods have a bias towards flat minima. In this work, we proved that for linear networks trained with a quadratic loss, these solutions possess unique properties. For example, in flat minima networks, the signal $\boldsymbol{v}$ that experiences the largest gain end-to-end, is amplified as moderately as possible by each layer. Moreover, no other signal can experience a significantly larger gain than $\boldsymbol{v}$ up to an intermediate layer. Finally, these solutions exhibit a coupling between the layers, which forms a distinct path for $\boldsymbol{v}$ from input to output. While our theoretical results apply to linear networks, our experiments show that these properties are also characteristic of nonlinear networks trained in practice.

Acknowledgments

This research was supported in part by the Technion Ollendorff Minerva Center.

References

Arora, S., Cohen, N., and Hazan, E. E. On the optimization of deep networks: Implicit acceleration by overparameterization. In *35th International Conference on Machine Learning, ICML 2018*, pp. 372–389. International Machine Learning Society (IMLS), 2018.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-SGD: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1019–1028. JMLR. org, 2017.

Du, S. S., Hu, W., and Lee, J. D. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, pp. 384–395, 2018.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 6151–6159, 2017.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*, 2018a.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 9461–9471, 2018b.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, pp. 1731–1741, 2017.

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. Three factors influencing minima in SGD. *arXiv preprint arXiv:1711.04623*, 2017.

Ji, Z. and Telgarsky, M. J. Gradient descent aligns the layers of deep linear networks. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

LeCun, Y. The MNIST database of handwritten digits. 1998.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pp. 6389–6399, 2018.

Lyu, K. and Li, J. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020.

Masters, D. and Luschi, C. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

Morgan, N. and Bourlard, H. Generalization and parameter estimation in feedforward nets: Some experiments. In *Advances in neural information processing systems*, pp. 630–637, 1990.

Nar, K. and Sastry, S. Step size matters in deep learning. In *Advances in Neural Information Processing Systems*, pp. 3436–3444, 2018.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pp. 5947–5956, 2017.

Simsekli, U., Sagun, L., and Gurbuzbalaban, M. A tail-index analysis of stochastic gradient noise in deep neural networks. *arXiv preprint arXiv:1901.06053*, 2019.

Smith, S. L. and Le, Q. V. A bayesian perspective on generalization and stochastic gradient descent. *arXiv preprint arXiv:1710.06451*, 2017.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Strand, O. N. Theory and methods related to the singular-function expansion and landweber's iteration for integral equations of the first kind. *SIAM Journal on Numerical Analysis*, 11(4):798–825, 1974.

Wu, L., Ma, C., and Weinan, E. How SGD selects the global minima in over-parameterized learning: A dynamical stability perspective. In *Advances in Neural Information Processing Systems*, pp. 8279–8288, 2018.

Wu, X., Dobriban, E., Ren, T., Wu, S., Li, Z., Gunasekar, S., Ward, R., and Liu, Q. Implicit regularization of normalization methods. *arXiv preprint arXiv:1911.07956*, 2019.

Zhang, H., Dauphin, Y. N., and Ma, T. Fixup initialization: Residual learning without normalization. In *International Conference on Learning Representations*, 2018.