The Impact of Neural Network Overparameterization on Gradient Confusion and Stochastic Gradient Descent

Karthik A. Sankararaman* [1] [2] Soham De* [3] Zheng Xu [2] W. Ronny Huang [2] Tom Goldstein [2]

Abstract

This paper studies how neural network architecture affects the speed of training. We introduce a simple concept called *gradient confusion* to help formally analyze this. When gradient confusion is high, stochastic gradients produced by different data samples may be negatively correlated, slowing down convergence. But when gradient confusion is low, data samples interact harmoniously, and training proceeds quickly. Through theoretical and experimental results, we demonstrate how the neural network architecture affects gradient confusion, and thus the efficiency of training. Our results show that, for popular initialization techniques, increasing the *width* of neural networks leads to *lower* gradient confusion, and thus faster model training. On the other hand, increasing the *depth* of neural networks has the opposite effect. Our results indicate that alternate initialization techniques or networks using both batch normalization and skip connections help reduce the training burden of very deep networks.

1. Introduction

Stochastic gradient descent (SGD) (Robbins & Monro, 1951) and its variants with momentum have become the standard optimization routine for neural networks due to their fast convergence and good generalization properties (Wilson et al., 2017; Sutskever et al., 2013; Smith et al., 2020). Yet the convergence behavior of SGD on neural networks still eludes full theoretical understanding. Furthermore, it is not well understood how design choices on neural network architecture affect training performance. In this paper, we make progress on these open questions.

Classical stochastic optimization theory predicts that the learning rate of SGD needs to decrease over time for convergence to be guaranteed to the minimizer of a convex function (Shamir & Zhang, 2013; Bertsekas, 2011). For strongly convex functions for example, such results show that a decreasing learning rate schedule of $O(1/k)$ is required to guarantee convergence to within ϵ-accuracy of the minimizer in $O(1/\epsilon)$ iterations, where k denotes the iteration number. Such decay schemes, however, typically lead to poor performance on standard neural network problems.

Neural networks operate in a regime where the number of parameters is much larger than the number of training data. In this "over-parameterized" regime, SGD seems to converge quickly with constant learning rates. Most neural network practitioners use a constant learning rate for the majority of training (with exponential decay only towards the end of training) without seeing the method stall (Krizhevsky et al., 2012; Simonyan & Zisserman, 2014; He et al., 2016; Zagoruyko & Komodakis, 2016). With constant learning rates, theoretical guarantees show that SGD converges quickly to a neighborhood of the minimizer, but then reaches a *noise floor* beyond which it stops converging; this noise floor depends on the learning rate and the variance of the gradients (Moulines & Bach, 2011; Needell et al., 2014). Recent results show that convergence without a noise floor is possible without decaying the learning rate, provided the model is strongly convex and overfitting occurs (Schmidt & Roux, 2013; Ma et al., 2017; Vaswani et al., 2018).

While these results do give important insights, they do not fully explain the dynamics of SGD on neural networks, and how they relate to over-parameterization. Furthermore, training performance is strongly influenced by network architecture. It is common knowledge among practitioners that, under standard Gaussian initialization techniques (Glorot & Bengio, 2010; He et al., 2015), deeper networks train slower (Bengio et al., 1994; Saxe et al., 2013). This has led to several innovations over the years to get deeper nets to train more easily, such as careful initialization strategies (Xiao et al., 2018), residual connections (He et al., 2016), and normalization schemes like batch normalization (Ioffe & Szegedy, 2015). Furthermore, there is evidence to indicate that wider networks are faster to train (Zagoruyko & Komodakis, 2016; Nguyen & Hein, 2017), and recent the-

*Equal contribution [1]Facebook. [2]University of Maryland, College Park. [3]DeepMind, London.. Correspondence to: Karthik A. Sankararaman <karthikabinavs@gmail.com>, Soham De <sohamde@google.com>.

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

oretical results suggest that the dynamics of SGD simplify considerably for very wide networks (Jacot et al., 2018; Lee et al., 2019). In this paper, we make progress on theoretically understanding these empirical observations and unifying existing theoretical results. To this end, we identify and analyze a condition that enables us to establish direct relationships between layer width, network depth, problem dimensionality, initialization schemes, and trainability and SGD dynamics for over-parameterized networks.

Our contributions. Typical neural networks are *over-parameterized* (*i.e.,* the number of parameters exceed the number of training points). In this paper, we ask how this over-parameterization, and more specifically the network architecture, affects the trainability of neural networks and the dynamics of SGD. Through extensive theoretical and experimental studies, we show how layer width, network depth, initialization schemes, and other architecture choices affect the dynamics. The following are our main contributions.[1]

- We identify a condition, termed *gradient confusion*, that impacts the convergence properties of SGD on over-parameterized models. We prove that high gradient confusion may lead to slower convergence, while convergence is accelerated (and could be faster than predicted by existing theory) if confusion is low, indicating a regime where constant learning rates work well in practice (sections 2 and 3). We use the gradient confusion condition to study the effect of various architecture choices on trainability and convergence.

- We study the effect of neural network architecture on gradient confusion at standard Gaussian initialization schemes (section 4), and prove (a) gradient confusion increases as the network depth increases, and (b) wider networks have lower gradient confusion. These indicate that deeper networks are more difficult to train and wider networks can improve trainability of networks. Directly analyzing the gradient confusion bound enables us to derive results on the effect of depth and width, without requiring restrictive assumptions like large layer widths (Du et al., 2018; Allen-Zhu et al., 2018). Our results hold for a large class of neural networks with different non-linear activations and loss-functions. In section 5, we present a more general result on the effect of depth on the trainability of networks without assuming the network is at initialization.

- We prove that for linear neural networks, gradient confusion is *independent of depth* when using orthogonal initialization schemes (section 6) (Saxe et al., 2013; Schoenholz et al., 2016). This indicates a way forward in developing techniques for training deeper models.

- We test our theoretical predictions using extensive experiments on wide residual networks (WRNs) (Zagoruyko & Komodakis, 2016), convolutional networks (CNNs) and multi-layer perceptrons (MLPs) for image classification tasks on CIFAR-10, CIFAR-100 and MNIST (section 7 and appendix A). We find that our theoretical results consistently hold across all our experiments. We further show that the combination of batch normalization and skip connections in residual networks help lower gradient confusion, thus indicating why SGD can efficiently train deep neural networks that employ such techniques.

2. Gradient confusion

Notations. We denote vectors in bold lower-case and matrices in bold upper-case. We use $(\mathbf{W})_{i,j}$ to indicate the (i, j) cell in matrix $\mathbf{W}$ and $(\mathbf{W})_i$ for the i^{th} row of matrix $\mathbf{W}$. $\|\mathbf{W}\|$ denotes the operator norm of $\mathbf{W}$. $[N]$ denotes $\{1, 2, \dots, N\}$ and $[N]_0$ denotes $\{0, 1, \dots, N\}$.

Preliminaries. Given N training points (specified by the corresponding loss functions $\{f_i\}_{i\in[N]}$), we use SGD to solve empirical risk minimization problems of the form,

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) := \min_{\mathbf{w}\in\mathbb{R}^d} \tfrac{1}{N} \textstyle\sum_{i=1}^{N} f_i(\mathbf{w}), \qquad (1)$$

using the following iterative update rule for T rounds:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha \nabla \tilde{f}_k(\mathbf{w}_k). \qquad (2)$$

Here α is the learning rate and $\tilde{f}_k$ is a function chosen uniformly at random from $\{f_i\}_{i\in[N]}$ at iteration $k \in [T]$. $\mathbf{w}^\star = \arg\min_{\mathbf{w}} F(\mathbf{w})$ denotes the optimal solution.

Gradient confusion. SGD works by iteratively selecting a random function $\tilde{f}_k$, and modifying the parameters to move in the direction of the negative gradient of $\tilde{f}_k$. It may happen that the selected gradient $\nabla \tilde{f}_k$ is negatively correlated with the gradient of another term ∇f_j. When the gradients of different mini-batches are negatively correlated, the objective terms disagree on which direction the parameters should move, and we say that there is *gradient confusion*.[2]

Definition 2.1. *A set of objective functions $\{f_i\}_{i\in[N]}$ has gradient confusion bound $\eta \geq 0$ if the pair-wise inner products between gradients satisfy, for a fixed $\mathbf{w} \in \mathbb{R}^d$,*

$$\langle \nabla f_i(\mathbf{w}), \nabla f_j(\mathbf{w}) \rangle \geq -\eta, \ \forall i \neq j \in [N]. \qquad (3)$$

Observations in simplified settings. SGD converges fast when gradient confusion is low along its path. To see why,

[1] To keep the main text of the paper concise, all proofs and several additional experimental results are delegated to the appendix.

[2] Gradient confusion is related to both gradient variance and gradient diversity (Yin et al., 2017), but with important differences, which we discuss in section 9. We also discuss alternate definitions of the gradient confusion condition in section 8.

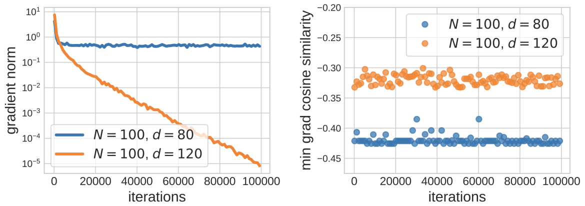


Figure 1. Linear regression on an over-parameterized ($d = 120$) and under-parameterized ($d = 80$) model with $N = 100$ samples generated randomly from a Gaussian, trained using SGD with mini-batch size 1. Plots are averaged over 3 independent runs. Gradient cosine similarities were calculated over all pairs of gradients.

consider the case of training a logistic regression model on a dataset with *orthogonal* vectors. We have $f_i(\mathbf{w}) = \mathcal{L}(y_i \mathbf{x}_i^\top \mathbf{w})$, where $\mathcal{L} : \mathbb{R} \to \mathbb{R}$ is the logistic loss, $\{\mathbf{x}_i\}_{i \in [N]}$ is a set of orthogonal training vectors, and $y_i \in \{-1, 1\}$ is the label for $\mathbf{x}_i$. We then have $\nabla f_i(\mathbf{w}) = \zeta_i \mathbf{x}_i$, where $\zeta_i = y_i \mathcal{L}'(y_i \cdot \mathbf{x}_i^\top \mathbf{w})$. Note that the gradient confusion is 0 since $\langle \nabla f_i(\mathbf{w}), \nabla f_j(\mathbf{w}) \rangle = \zeta_i \zeta_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle = 0, \forall i, j \in [N]$ and $i \neq j$. Thus, an update in the gradient direction f_i has *no* effect on the loss value of f_j for $i \neq j$. In this case, SGD decouples into (deterministic) gradient descent on each objective term separately, and we can expect to see the fast convergence rates attained by gradient descent.

Can we expect a problem to have low gradient confusion in practice? From the logistic regression problem, we have: $|\langle \nabla f_i(\mathbf{w}), \nabla f_j(\mathbf{w}) \rangle| = |\langle \mathbf{x}_i, \mathbf{x}_j \rangle| \cdot |\zeta_i \zeta_j|$. This inner product is expected to be small for all $\mathbf{w}$; the logistic loss satisfies $|\zeta_i \zeta_j| < 1$, and for fixed N the quantity $\max_{ij} |\langle \mathbf{x}_j, \mathbf{x}_i \rangle|$ is $O(1/\sqrt{d})$ whenever $\{\mathbf{x}_i\}$ are randomly sampled from a sphere (see lemma B.1 for the formal statement).[3] Thus, we would expect a random linear model to have nearly orthogonal gradients, when the number of parameters is "large" and the number of training data is "small", i.e., when the model is over-parameterized. This is further evidenced by a toy example in figure 1, where we show a slightly over-parameterized linear regression model can have much faster convergence rates, as well as lower gradient confusion. One can prove a similar result for problems that have random and low-rank Hessians, which suggests that one might expect gradient to be small near the minimizer for many standard neural nets (see appendix C for more discussion).

The above arguments are a bit simplistic, considering toy scenarios and ignoring issues like the effect of network structure. In the following sections, we rigorously analyze the effect of gradient confusion on the speed of convergence on non-convex problems, and the effect of width and depth of the neural network architecture on the gradient confusion.

3. SGD is fast when gradient confusion is low

Several prior papers have analyzed the convergence rates of constant learning rate SGD (Nedić & Bertsekas, 2001; Moulines & Bach, 2011; Needell et al., 2014). These results show that for strongly convex and Lipschitz smooth functions, SGD with a constant learning rate α converges *linearly* to a neighborhood of the minimizer. The noise floor it converges to depends on the learning rate α and the variance of the gradients at the minimizer, i.e., $\mathbb{E}_i \|\nabla f_i(\mathbf{w}^\star)\|^2$. To guarantee convergence to ϵ-accuracy in such a setting, the learning rate needs to be small, i.e., $\alpha = O(\epsilon)$, and the method requires $T = O(1/\epsilon)$ iterations. Some more recent results show convergence of constant learning rate SGD without a noise floor and without small step sizes for models that can completely fit the data (Schmidt & Roux, 2013; Ma et al., 2017; Vaswani et al., 2018).

Gradient confusion is related to these results. Cauchy-Schwarz inequality implies that if $\mathbb{E}_i \|\nabla f_i(\mathbf{w}^\star)\|^2 = O(\epsilon)$, then $\mathbb{E}_{i,j} |\langle \nabla f_i(\mathbf{w}^\star), \nabla f_j(\mathbf{w}^\star) \rangle| = O(\epsilon), \forall i, j$. Thus the gradient confusion at the minimizer is small when the variance of the gradients at the minimizer is small. Further note that when the variance of the gradients at the minimizer is $O(\epsilon)$, a direct application of the results in Moulines & Bach (2011) and Needell et al. (2014) shows that constant learning rate SGD has fast convergence to ϵ-accuracy in $T = O(\log(1/\epsilon))$ iterations, without the learning rate needing to be small. Generally however, bounded gradient confusion does not provide a bound on the variance of the gradients (see section 9). Thus, it is instructive to derive convergence bounds of SGD explicitly in terms of the gradient confusion to properly understand its effect.

We first consider functions satisfying the Polyak-Lojasiewicz (PL) inequality (Lojasiewicz, 1965), a condition related to, but weaker than, strong convexity, and used in recent work (Karimi et al., 2016; De et al., 2017). We provide bounds on the rate of convergence in terms of the optimality gap. We start with two standard assumptions.

(A1) $\{f_i\}_{i \in [N]}$ are *Lipschitz smooth*:
$f_i(\mathbf{w}') \leq f_i(\mathbf{w}) + \nabla f_i(\mathbf{w})^\top (\mathbf{w}' - \mathbf{w}) + \frac{L}{2} \|\mathbf{w}' - \mathbf{w}\|^2$.

(A2) $\{f_i\}_{i \in [N]}$ satisfy the *PL inequality*:
$\frac{1}{2} \|\nabla f_i(\mathbf{w})\|^2 \geq \mu(f_i(\mathbf{w}) - f_i^\star)$, $f_i^\star = \min_{\mathbf{w}} f_i(\mathbf{w})$.

We now state a convergence result of constant learning rate SGD in terms of the gradient confusion.

Theorem 3.1. *If the objective function satisfies (A1) and (A2), and has gradient confusion η, SGD converges linearly to a neighborhood of the minima of problem (1) as:*

$$\mathbb{E}[F(\mathbf{w}_T) - F^\star] \leq \rho^T (F(\mathbf{w}_0) - F^\star) + \frac{\alpha \eta}{1 - \rho},$$

where $\alpha < \frac{2}{NL}$, $\rho = 1 - \frac{2\mu}{N} \left(\alpha - \frac{NL\alpha^2}{2} \right)$, $F^\star = \min_{\mathbf{w}} F(\mathbf{w})$ and $\mathbf{w}_0$ is the initialized weights.

[3] Generally, this is true whenever $\mathbf{x}_i = \frac{1}{\sqrt{d}} \mathbf{y}_i$, where $\mathbf{y}_i$ is an isotropic random vector (Vershynin, 2018).

This result shows that SGD converges *linearly* to a neighborhood of a minimizer, and the size of this neighborhood depends on the level of gradient confusion. When the gradient confusion is small, i.e., $\eta = O(\epsilon)$, SGD has fast convergence to $O(\epsilon)$-accuracy in $T = O(\log(1/\epsilon))$ iterations, without requiring the learning rate to be vanishingly small. We now extend this to general smooth functions.

Theorem 3.2. *If the objective satisfies (A1) and has gradient confusion η, then SGD converges to a neighborhood of a stationary point of problem (1) as:*

$$\min_{k=1,\dots,T} \mathbb{E}\|\nabla F(\mathbf{w}_k)\|^2 \leq \frac{\rho(F(\mathbf{w}_1)-F^\star)}{T} + \rho\eta,$$

for $\alpha < \frac{2}{NL}$, $\rho = \frac{2N}{2-NL\alpha}$, and $F^\star = \min_{\mathbf{w}} F(\mathbf{w})$.

Thus, as long as $\eta = O(1/T)$, SGD has fast $O(1/T)$ convergence on smooth non-convex functions. Theorems 3.1 and 3.2 predict an initial phase of optimization with fast convergence to the neighborhood of a minimizer or a stationary point. This behavior is often observed when optimizing neural nets (Darken & Moody, 1992; Sutskever et al., 2013), where a constant learning rate reaches a high level of accuracy on the model. As we show in subsequent sections, this is expected since for neural networks typically used, the gradient confusion is expected to be low. See section 9 for more discussion on the above results and how they relate to previous work. We stress that our goal is not to study convergence rates per se, nor is it to prove state-of-the-art rate bounds for this class of problems. Rather, we show the direct effect that the gradient confusion bound has on the convergence rate and the noise floor for constant learning rate SGD. As we show in the following sections, this new perspective in terms of the gradient confusion helps us more directly understand how neural network architecture design affects SGD dynamics and why.

4. Effect of neural network architecture at Gaussian initializations

To draw a connection between neural network architecture and training performance, we analyze gradient confusion for generic (i.e., random) model problems using methods from high-dimensional probability. In this section, we analyze the effect of neural network architecture at the beginning of training, when using standard Gaussian initialization techniques. Analyzing these models at initialization is important to understand which architectures are more easily trainable than others. Our results cover a wide range of scenarios compared to prior work, require minimal additional assumptions, and hold for a large family of neural networks with different non-linear activation functions and loss-functions. In particular, our results hold for fully connected networks (and can be extended to convolutional networks) with the square-loss and logistic-loss functions, and commonly used non-linear activations such as sigmoid, tanh and ReLU. We consider both the case where the input data is arbitrary but bounded (theorem 4.1, part 1), as well as where the input data is randomly drawn from the surface of a unit sphere (theorem 4.1, part 2).

Setting. We consider training data $\mathcal{D} = \{(\mathbf{x}_i, \mathcal{C}(\mathbf{x}_i))\}_{i\in[N]}$, with labeling function $\mathcal{C} : \mathbb{R}^d \to [-1, 1]$. For some of our results, we consider that the data points $\{\mathbf{x}_i\}$ are drawn uniformly at random from the surface of a d-dimensional unit sphere. The labeling function satisfies $|\mathcal{C}(\mathbf{x})| \leq 1$ and $\|\nabla_{\mathbf{x}}\mathcal{C}(\mathbf{x})\|_2 \leq 1$ for $\|\mathbf{x}\| \leq 1$. Note that this automatically holds for every model considered in this paper where the labeling function is *realizable* (i.e., where the model can express the labeling function using its parameters). More generally, this assumes a Lipschitz condition on the labels (i.e., the labels don't change too quickly with the inputs).

We consider two loss-functions: square-loss for regression and logistic loss for classification. The square-loss function is defined as $f_i(\mathbf{w}) = \frac{1}{2}(\mathcal{C}(\mathbf{x}_i) - g_{\mathbf{w}}(\mathbf{x}_i))^2$ and the logistic function is defined as $f_i(\mathbf{w}) = \log(1 + \exp(-\mathcal{C}(\mathbf{x}_i)g_{\mathbf{w}}(\mathbf{x}_i)))$. Here, $g_{\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}$ denotes the parameterized function we fit to the training data and $f_i(\mathbf{w})$ denotes the loss-function of hypothesis $g_{\mathbf{w}}$ on data point $\mathbf{x}_i$.

Let $\mathbf{W}_0 \in \mathbb{R}^{\ell_1\times d}$ and $\{\mathbf{W}_p\}_{p\in[\beta]}$ where $\mathbf{W}_p \in \mathbb{R}^{\ell_p\times\ell_{p-1}}$ are weight matrices. Let $\mathbf{W}$ denote the tuple $(\mathbf{W}_p)_{p\in[\beta]_0}$. Define $\ell := \max_{p\in[\beta]} \ell_p$ to be the *width* and β to be the *depth* of the network. Then, the model $g_{\mathbf{W}}$ is defined as

$$g_{\mathbf{W}}(\mathbf{x}) := \sigma(\mathbf{W}_\beta\sigma(\mathbf{W}_{\beta-1}\dots\sigma(\mathbf{W}_1\sigma(\mathbf{W}_0\mathbf{x}))\dots)),$$

where σ denotes the non-linear activation function applied point-wise to its arguments. We assume that the activation is given by a function $\sigma(x)$ with the following properties.

- **(P1) Boundedness:** $|\sigma(x)| \leq 1$ for $x \in [-1, 1]$.
- **(P2) Bounded differentials:** Let $\sigma'(x)$ and $\sigma''(x)$ denote the first and second sub-differentials respectively. Then, $|\sigma'(x)| \leq 1$ and $|\sigma''(x)| \leq 1$ for all $x \in [-1, 1]$.

When $\|\mathbf{x}\| \leq 1$, activation functions such as *sigmoid*, *tanh*, *softmax* and *ReLU* satisfy these requirements.

Furthermore, in this section, we consider the following Gaussian weight initialization strategy.

Strategy 4.1. $\mathbf{W}_0 \in \mathbb{R}^{\ell\times d}$ *has independent* $\mathcal{N}(0, \frac{1}{d})$ *entries. For every $p \in [\beta]$, the weights $\mathbf{W}_p \in \mathbb{R}^{\ell_p\times\ell_{p-1}}$ have independent* $\mathcal{N}\left(0, \frac{1}{\kappa\ell_{p-1}}\right)$ *entries for some constant $\kappa > 0$.*

This initialization strategy with different settings of κ are used almost universally for neural networks (Glorot & Bengio, 2010; LeCun et al., 2012; He et al., 2015). For instance, typically $\kappa = \frac{1}{2}$ when ReLU activations are used, and $\kappa = 1$ when tanh activations are used.

Main result. The following theorem shows how the width $\ell := \max_{p \in [\beta]} \ell_p$ and the depth β affect the gradient confusion condition at standard initializations. We show that *as width increases (for fixed depth) or depth decreases (for fixed width) the probability that the gradient confusion bound (equation 3) holds increases*. Thus, as the depth increases (with fixed width), training a model becomes harder, while as the width increases (with fixed depth), training a model becomes easier. Furthermore, note that this result also implies that training very deep *linear* neural networks (with identity activation functions) with standard Gaussian initializations is hard. Throughout the paper, we define the parameter $\zeta_0 := 2\sqrt{\beta}$. See the appendix (Lemma D.1) for a more careful definition of this quantity.

Theorem 4.1. *Let $\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_\beta$ be weight matrices chosen according to strategy 4.1. There exists fixed constants $c_1, c_2 > 0$ such that we have the following.*

1. *Consider a fixed but arbitrary dataset $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ with $\|\mathbf{x}_i\| \leq 1$ for every $i \in [N]$. For $\eta > 4$, the gradient confusion bound in equation 3 holds with probability at least*

$$1 - \beta \exp\left(-c_1\kappa^2\ell^2\right) - N^2 \exp\left(\frac{-c\ell^2\beta(\eta-4)^2}{64\zeta_0^4(\beta+2)^4}\right).$$

2. *If the dataset $\{\mathbf{x}_i\}_{i \in [N]}$ is such that each $\mathbf{x}_i$ is an i.i.d. sample from the surface of d-dimensional unit sphere, then for every $\eta > 0$ the gradient confusion bound in equation 3 holds with probability at least*

$$1 - \beta \exp\left(-c_1\kappa^2\ell^2\right) - N^2 \exp\left(\frac{-c_2(\ell d+\ell^2\beta)\eta^2}{16\zeta_0^4(\beta+2)^4}\right).$$

Theorem 4.1 shows that under popular Gaussian initializations used, training becomes harder as networks get deeper. The result however also shows a way forward: layer width improves the trainability of deep networks. Other related work supports this showing that when the layers are infinitely wide, the learning dynamics of gradient descent simplifies considerably (Jacot et al., 2018; Lee et al., 2019). Hanin & Rolnick (2018) also suggest that the width should increase linearly with depth in a neural network to help dynamics at the beginning of training. In section 7 and appendix A, we show substantial empirical evidence that, given a sufficiently deep network, increasing the layer width often helps in lowering gradient confusion and speeding up convergence for a range of models.

5. A more general result on the effect of depth

While our results in section 4 hold at standard initialization schemes, in this section we derive a more general version of the result. In particular, we assume the setting where the data is drawn uniformly at random from a unit sphere and the weights lie in a ball around a local minimizer. Our results hold for both fully connected networks and convolutional networks with the square-loss and logistic-loss functions, and commonly-used non-linear activations such as sigmoid, tanh, softmax and ReLU.

We consider the same setup as in the previous section, and assume additionally that the data points $\{\mathbf{x}_i\}$ are drawn uniformly from the surface of a d-dimensional unit sphere. Additionally, instead of studying the network at initialization, we make the following assumption on the weights.

Assumption 1 (Small Weights)**.** *We assume that the operator norm of the weight matrices $\{\mathbf{W}_i\}_{i \in [\beta]_0}$ are bounded above by 1, i.e., for every $i \in [\beta]_0$ we have $\|\mathbf{W}_i\| \leq 1$.*

The operator norm of the weight matrices $\|\mathbf{W}\|$ being close to 1 is important for the trainability of neural networks, as it ensures that the input signal is passed through the network without exploding or shrinking across layers (Glorot & Bengio, 2010). Proving non-vacuous bounds in case of such blow-ups in magnitude of the signal or the gradient is not possible in general, and thus, we consider this restricted class of weights. Most standard neural networks are trained using *weight decay* regularizers of the form $\sum_i \|W_i\|_F^2$. This biases the weights to be small when training neural networks in practice. See appendix F for further discussion on the small weights assumption.

We now present a more general version of theorem 4.1.

Theorem 5.1. *Let $\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_\beta$ satisfy assumption 1. For some fixed constant $c > 0$, the gradient confusion bound (equation 3) holds with probability at least*

$$1 - N^2 \exp\left(\frac{-cd\eta^2}{16\zeta_0^4(\beta+2)^4}\right).$$

Theorem 5.1 shows that (for fixed dimension d and number of samples N) when the depth β decreases, the probability that the gradient confusion bound in equation 3 holds increases, and vice versa. Thus, our results indicate that in the general case when the weights are small, increasing the network depth will typically lead to slower model training.

Note that on assuming $\|\mathbf{W}\| \leq 1$ for each weight matrix $\mathbf{W}$, the dependence of gradient confusion on the width goes away. To see why this, consider an example where each weight matrix in the neural network has exactly one non-zero element, which is set to 1. The operator norm of each such weight matrix is 1, but the forward or backward propagated signals would not depend on the width.

Note that the convergence rate results of SGD in section 3 assume that the gradient confusion bound holds at every point along the path of SGD. On the other hand, theorem

5.1 shows concentration bounds for the gradient confusion at a fixed weight $\mathbf{W}$. Thus, to make the above result more relevant for the convergence of SGD on neural networks, we now make the concentration bound in theorem 5.1 *uniform over all weights inside a ball* $\mathcal{B}_r$ *of radius* r.

Corollary 5.1. *Select a point* $\mathbf{W} = (\mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_\beta)$*, satisfying assumption 1. Consider a ball* $\mathcal{B}_r$ *centered at* $\mathbf{W}$ *of radius* $r > 0$*. If the data* $\{\mathbf{x}_i\}_{i\in[N]}$ *are sampled uniformly from a unit sphere, then the gradient confusion bound in equation 3 holds uniformly at all points* $\mathbf{W}' \in \mathcal{B}_r$ *with probability at least*

$$\begin{cases} 1 - N^2 \exp\left(-\frac{cd\eta^2}{64\zeta_0^4(\beta+2)^4}\right), & \text{if } r \le \eta/4\zeta_0^2, \\ 1 - N^2 \exp\left(-\frac{cd\eta^2}{64\zeta_0^4(\beta+2)^4} + \frac{8d\zeta_0^2 r}{\eta}\right), & \text{otherwise.} \end{cases}$$

Corollary 5.1 shows that the probability that the gradient confusion bound holds decreases with increasing depth, for all weights in a ball around the minimizer.[4] This explains why, in the general case, training very deep models might always be hard. This raises the question why most deep neural networks used in practice are so efficiently trained using SGD. While careful Gaussian initialization strategies prevent vanishing or exploding gradients, these strategies still suffer from high gradient confusion for very deep networks unless the width is also increased with the depth, as we show in section 4. Practitioners over the years, however, have achieved state-of-the-art results by making networks deeper, without necessarily making networks wider. Thus, in section 7, we empirically study how popular techniques used in these models like skip connections and batch normalization affect gradient confusion. We find that these techniques drastically lower gradient confusion, making deep networks significantly easier to train. Furthermore, in the next section, we show how deep linear nets are trainable when used with orthogonal initialization techniques, indicating a way forward for training deeper models.

6. Gradient confusion is independent of depth for orthogonal initializations

In this section, we show that for deep linear neural networks, gradient confusion is independent of depth when the weight matrices are initialized as orthogonal matrices.[5] Consider the following linear neural network:

$$g_{\mathbf{W}}(\mathbf{x}) := \gamma \mathbf{W}_\beta \cdot \mathbf{W}_{\beta-1} \cdot \ldots \cdot \mathbf{W}_1 \cdot \mathbf{x}, \tag{4}$$

where the rescaling parameter $\gamma = \frac{1}{\sqrt{2\beta}}$, and assume we use the squared loss function. Then we have the following.

Theorem 6.1. *Let* $\{\mathbf{W}_i\}_{i\in[\beta]}$ *be arbitrary orthogonal matrices that satisfy assumption 1. Let the dataset* $\{\mathbf{x}_i\}_{i\in[N]}$ *be such that each* $\mathbf{x}_i$ *is an i.i.d. sample from the surface of* d*-dimensional unit sphere. Consider the linear neural network in equation 4 that minimizes the empirical square loss function. For some fixed constant* $c > 0$*, the gradient confusion bound (equation 3) holds with probability at least*

$$1 - N^2 \exp\left(-cd\eta^2\right).$$

From Theorem 6.1, we see that the probability does not depend on the depth β or maximum width ℓ. Thus, trainability does not get worse with depth when using orthogonal initializations. This result matches previous theoretical and empirical results showing the efficiency of orthogonal initialization techniques for training very deep linear or tanh networks (Saxe et al., 2013; Schoenholz et al., 2016; Xiao et al., 2018). However, orthogonal initializations are not compatable with non-linear activation functions like sigmoids or ReLUs, which limit their use in practice. Nonetheless, this result suggests a promising direction in developing techniques for training deeper models.

7. Experimental results

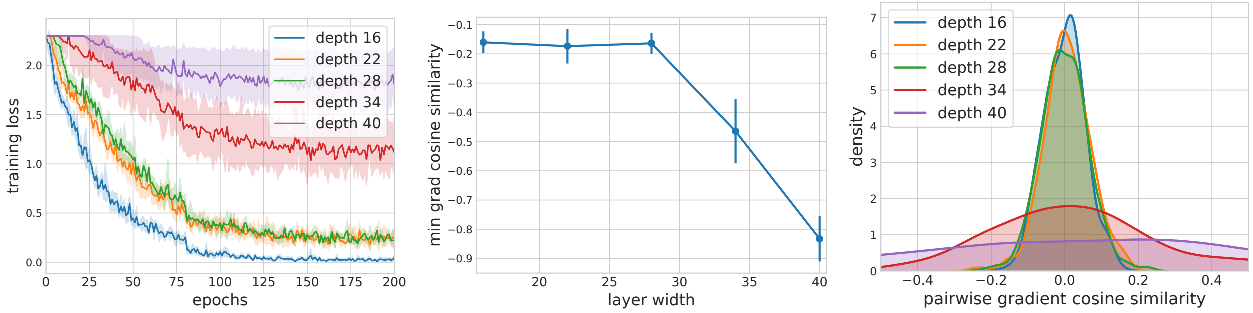
To test our theoretical results and to probe why standard neural networks are efficiently trained with SGD, we now present experimental results showing the effect of the neural network architecture on the convergence of SGD and gradient confusion. It is worth noting that theorems 3.1 and 3.2 indicate that we would expect the effect of gradient confusion to be most prominent closer to the end of training.

We performed experiments on wide residual networks (WRNs) (Zagoruyko & Komodakis, 2016), convolutional networks (CNNs) and multi-layer perceptrons (MLPs) for image classification tasks on CIFAR-10, CIFAR-100 and MNIST. We present results for CNNs on CIFAR-10 in this section, and present all other results in appendix A. We use CNN-β-ℓ to denote WRNs that have no skip connections or batch normalization, with a depth β and width factor ℓ.[6] We turned off dropout and weight decay for all our experiments. We used SGD as the optimizer without any momentum. Following Zagoruyko & Komodakis (2016), we ran all experiments for 200 epochs with minibatches of size 128, and reduced the initial learning rate by a factor of 10 at epochs 80 and 160. We used the MSRA initializer (He et al., 2015) for the weights as is standard for this model, and used the same preprocessing steps for the CIFAR-10 images as described in Zagoruyko & Komodakis (2016). We ran each experiment 5 times, and we show the standard deviation across runs in our plots. We tuned the optimal initial

[4] The above results automatically hold for convolutional networks, since a convolution operation on $\mathbf{x}$ can be represented as a matrix multiplication $\mathbf{Ux}$ for an appropriate Toeplitz matrix $\mathbf{U}$.

[5] An orthogonal matrix $\mathbf{A}$ satisfies $\mathbf{A}^T \cdot \mathbf{A} = \mathbf{A} \cdot \mathbf{A}^T = \mathbf{I}$.

[6] The width factor denotes the number of filters relative to the original ResNet model (Zagoruyko & Komodakis, 2016).



(a) (b) (c)

Figure 2. The effect of network depth with CNN-β-2 on CIFAR-10 for depths $\beta = 16$, 22, 28, 34 and 40. Plots show the (a) convergence curves for SGD, (b) minimum of pairwise gradient cosine similarities at the end of training, and the (c) kernel density estimate of the pairwise gradient cosine similarities at the end of training (over all independent runs).

learning rate for each model over a logarithmically-spaced grid and selected the run that achieved the lowest training loss value. To measure gradient confusion, at the end of every training epoch, we sampled 100 pairs of mini-batches each of size 128 (the same size as the training batch). We calculated gradients on each mini-batch, and then computed pairwise cosine similarities. See appendix A.2 for more details on the experimental setup and architectures used.

Effect of depth. To test our theoretical results, we consider CNNs with a fixed width factor of 2 and varying network depth. From figure 2, we see that our theoretical results are backed by the experiments: increasing depth slows down convergence, and increases gradient confusion. We also notice that with increasing depth, the density of pairwise gradient cosine similarities concentrates less sharply around 0, which makes the network harder to train.

Effect of width. We now consider CNNs with a fixed depth of 16 and varying width factors. From figure 3, we see that increasing width results in faster convergence and lower gradient confusion. We further see that gradient cosine similarities concentrate around 0 with growing width, indicating that SGD decouples across the training samples with growing width. Note that the smallest network considered (CNN-16-2) is still over-parameterized and achieves a high level of performance (see appendix A.3).

Effect of batch normalization and skip connections. Almost all state-of-the-art neural networks currently contain both skip connections and normalization layers. To help understand why such neural networks are so efficiently trained using SGD with constant learning rates, we test the effect of adding skip connections and batch normalization to CNNs of fixed width and varying depth. Figure 4 shows that adding skip connections or batch normalization individually help in training deeper models, but these models still suffer from worsening results and increasing gradient confusion as the network gets deeper. When these techniques are used together, the model has relatively low gradient confusion even for very deep networks, significantly improving trainability of deep models. Note that our observations are consistent with prior work (De & Smith, 2020; Yang et al., 2019).

8. Alternate definitions of gradient confusion

Note that the gradient confusion bound η in equation 3 is defined for the worst-case gradient inner product. However, all the results in this paper can be trivially extended to using a bound on the average gradient inner product of the form:

$$\textstyle\sum_{i,j=1}^{N} \langle \nabla f_i(\mathbf{w}), \nabla f_j(\mathbf{w}) \rangle / N^2 \geq -\eta.$$

In this case, all theoretical results would remain the same up to constants. We can also define a normalized variant of the gradient confusion condition:

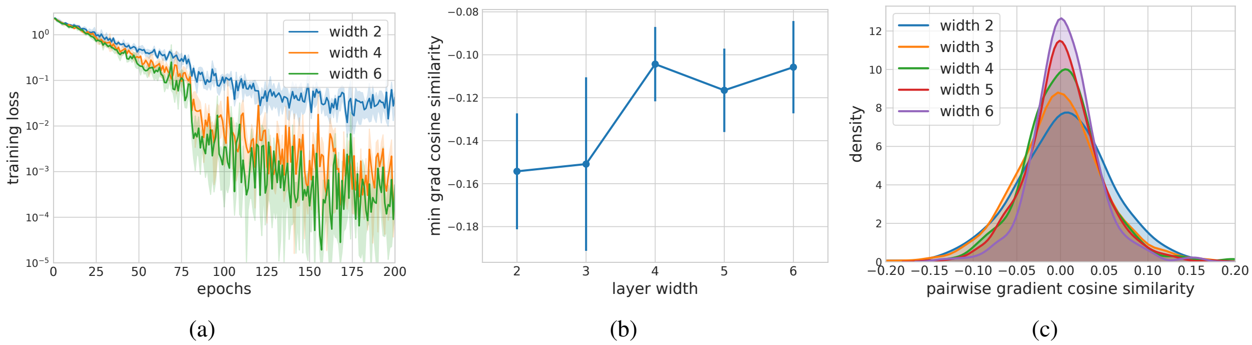
$$\langle \nabla f_i(\mathbf{w}), \nabla f_j(\mathbf{w}) \rangle / (\|\nabla f_i(\mathbf{w})\| \|\nabla f_j(\mathbf{w})\|) \geq -\eta.$$

This condition inherently makes an additional assumption that the norm of the stochastic gradients, $\|\nabla f_i(\mathbf{w})\|$, is bounded, and thus the gradient variance is also bounded (see discussion in section 9). Thus, while all our theoretical results would qualitatively remain the same under this condition, we can prove tighter versions of our current results.

Finally, note that gradient confusion condition in equation 3 is applicable even when the stochastic gradients are averaged over minibatches of size B. The variance of the gradient inner product scales down as $1/B^2$ in this case, and thus η is expected to decrease as B grows.

9. Related work

The gradient confusion bound and our theoretical results have interesting connections to prior work. In this section, we briefly discuss some of these connections.

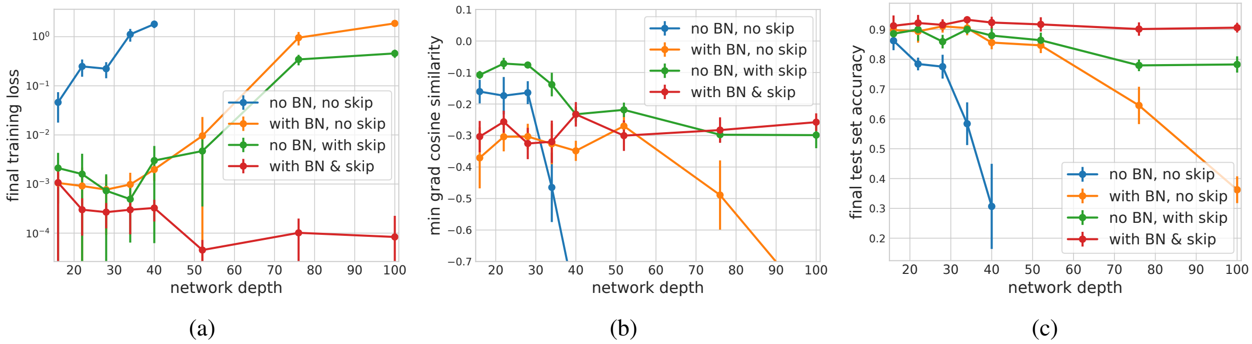


(a)

(b)

(c)

Figure 3. The effect of width with CNN-16-ℓ on CIFAR-10 for width factors $\ell = 2$, 3, 4, 5 and 6. Plots show the (a) convergence curves of SGD (for cleaner figures, we plot results for width factors 2, 4 and 6 here), (b) minimum of pairwise gradient cosine similarities at the end of training, and the (c) kernel density estimate of the pairwise gradient cosine similarities at the end of training (over all independent runs).



(a)

(b)

(c)

Figure 4. The effect of adding skip connections and batch normalization to CNN-β-2 on CIFAR-10 for depths $\beta = 16$, 22, 28, 34, 40, 52, 76 and 100. Plots show the (a) optimal training losses, (b) minimum pairwise gradient cosine similarities, and the (c) test set accuracies at the end of training.

Connections to the gradient variance: If we assume bounded gradient variance $\mathbb{E}_i\|\nabla f_i(\mathbf{w}) - \nabla F(\mathbf{w})\|^2 \leq \sigma^2$, we can bound the gradient confusion parameter η in terms of other quantities. For example, suppose the true gradient $\nabla F(\mathbf{w}) = \nabla f_1(\mathbf{w})/2 + \nabla f_2(\mathbf{w})/2$. Then we can write: $|\langle \nabla f_1(\mathbf{w}), \nabla f_2(\mathbf{w})\rangle| \leq \sigma^2 + \|\nabla F(\mathbf{w})\|^2$. However, in general one cannot bound the gradient variance in terms of the gradient confusion parameter. As a counter-example, consider a problem with the following distribution on the gradients: $\frac{1}{1-p}$ samples with gradient $\frac{1}{\epsilon}$ and $\frac{1}{p}$ samples with gradient ϵ, where $p = \epsilon \to 0$. In this case, the gradients are positive, so gradient confusion $\eta = 0$. The mean of the gradients is given by $1+\epsilon(1-\epsilon)$, which remains bounded as $\epsilon \to 0$. On the other hand, the variance (and thus the squared norm of the stochastic gradients) is unbounded ($O(1/\epsilon)$ as $\epsilon \to 0$). A consequence of this is that in theorems 3.1 and 3.2, the "noise term" (i.e., the second term in the RHS of the convergence bounds) does not depend on the learning rate in the general case. If gradients have unbounded variance, lowering the learning rate does not reduce the variance of the SGD updates, and thus does not reduce the noise term.

Connections to gradient diversity: Gradient diversity (Yin et al., 2017) also measures the degree to which individual gradients at different data samples are different from each other. However, the gradient diversity measure gets larger as the individual gradients become orthogonal to each other, and further increases as the gradients start pointing in opposite directions. On the other hand, gradient confusion between two individual gradients is zero unless the inner product between them is negative. As we show in this paper, this has important implications when we study the convergence of SGD in the over-parameterized setting: increased width makes gradients more orthogonal to each other improving trainability, while increased depth result in gradients pointing in opposite directions making networks harder to train. Thus, we view our papers to be complementary, providing insights about different issues (large batch distributed training vs. small batch convergence).

Related work on the impact of network architecture: Balduzzi et al. (2017) studied neural networks with ReLU activations at Gaussian initializations, and showed that gradients become increasingly negatively correlated with depth.

Hanin (2018) showed that the variance of gradients in fully connected networks with ReLU activations is exponential in the sum of the reciprocals of the hidden layer widths at Gaussian initializations. In a follow-up work, Hanin & Rolnick (2018) showed that this sum of the reciprocals of the hidden layer widths determines the variance of the sizes of the activations at each layer. When this sum of reciprocals is too large, early training dynamics are very slow, suggesting the difficulties of starting training on deeper networks, as well as the benefits of increased width.

Other work on SGD convergence: There has recently been a lot of interest in analyzing conditions under which SGD converges to global minimizers of over-parameterized linear and non-linear neural networks. Arora et al. (2018) shows SGD converges linearly to global minimizers for linear neural networks under certain conditions. Du et al. (2018); Allen-Zhu et al. (2018); Zou et al. (2018); Brutzkus et al. (2017) also show convergence to global minimizers of SGD for non-linear networks. This paper complements these recent results by studying how low gradient confusion contributes to SGD's success on over-parameterized neural networks used in practice.

10. Discussion

In this paper, we study how neural network architecture affects the trainability of networks and the dynamics of SGD. To rigorously analyze this, we introduce a concept called gradient confusion, and show that when gradient confusion is low, SGD has fast convergence. We show at standard Gaussian initializations, increasing layer width leads to lower gradient confusion, making the model easier to train. In contrast, increasing depth results in higher gradient confusion, making models harder to train. These results indicate that increasing the layer width with the network depth is important to maintain trainability of the neural network. This is supported by other recent work that suggest that the width should increase linearly with depth in a Gaussian-initialized neural network to help dynamics early in training (Hanin, 2018; Hanin & Rolnick, 2018).

Many previous results have shown how deeper models are more efficient at modeling higher complexity function classes than wider models, and thus depth is essential for the success of neural networks (Eldan & Shamir, 2016; Telgarsky, 2016). Indeed, practitioners over the years have achieved state-of-the-art results on various tasks by making networks deeper, without necessarily making networks wider. We thus study techniques that enable us to train deep models without requiring us to increase the width with depth. Most state-of-the-art neural networks currently contain both skip connections and normalization layers. We thus, empirically study the effect of introducing batch normalization and skip connections to a neural network. We show that the combination of batch normalization and skip connections lower gradient confusion and help train very deep models, explaining why many neural networks used in practice are so efficiently trained. Furthermore, we show how orthogonal initialization techniques provide a promising direction for improving the trainability of very deep networks.

Our results provide a number of important insights that can be used for neural network model design. We demonstrate that the gradient confusion condition could be useful as a measure of trainability of networks, and thus could potentially be used to develop algorithms for more efficient training. Additionally, the correlation between gradient confusion and the test set accuracies shown in appendix A suggest that an interesting topic for future work would be to investigate the connection between gradient confusion and generalization (Fort et al., 2019). Our results also suggest the importance of further work on orthogonal initialization schemes for neural networks with non-linear activations that make training very deep models possible.

Acknowledgements

The authors thank Brendan OâĂŹDonoghue, Aleksandar Botev, James Martens, Sudha Rao, and Samuel L. Smith for helpful discussions and for reviewing earlier versions of this manuscript. This paper was supported by the ONR MURI program, AFOSR MURI Program, and the National Science Foundation DMS directorate.

References

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018.

Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question? *arXiv preprint arXiv:1702.08591*, 2017.

Bengio, Y., Simard, P., and Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Bertsekas, D. P. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

Boucheron, S., Lugosi, G., and Massart, P. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Brutzkus, A., Globerson, A., Malach, E., and Shalev-Shwartz, S. Sgd learns over-parameterized networks that provably generalize on linearly separable data. *arXiv preprint arXiv:1710.10174*, 2017.

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. *arXiv preprint arXiv:1611.01838*, 2016.

Cooper, Y. The loss landscape of overparameterized neural networks. *arXiv preprint arXiv:1804.10200*, 2018.

Darken, C. and Moody, J. Towards faster stochastic gradient search. In *Advances in neural information processing systems*, pp. 1009–1016, 1992.

De, S. and Smith, S. L. Batch normalization biases residual blocks towards the identity function in deep networks. *arXiv preprint arXiv:2002.10444*, 2020.

De, S., Yadav, A., Jacobs, D., and Goldstein, T. Automated inference with adaptive batches. In *Artificial Intelligence and Statistics*, pp. 1504–1513, 2017.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Eldan, R. and Shamir, O. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, pp. 907–940, 2016.

Fort, S., Nowak, P. K., and Narayanan, S. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*, 2019.

Ghorbani, B., Krishnan, S., and Xiao, Y. An investigation into neural net optimization via hessian eigenvalue density. *arXiv preprint arXiv:1901.10159*, 2019.

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

Hanin, B. Which neural net architectures give rise to exploding and vanishing gradients? In *Advances in Neural Information Processing Systems*, pp. 582–591, 2018.

Hanin, B. and Rolnick, D. How to start training: The effect of initialization and architecture. In *Advances in Neural Information Processing Systems*, pp. 571–581, 2018.

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.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8580–8589, 2018.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer, 2016.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–48. Springer, 2012.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

Lojasiewicz, S. Ensembles semi-analytiques. *Lectures Notes IHES (Bures-sur-Yvette)*, 1965.

Ma, S., Bassily, R., and Belkin, M. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning. *arXiv preprint arXiv:1712.06559*, 2017.

Martens, J. *Second-order optimization for neural networks*. University of Toronto (Canada), 2016.

Milman, V. D. and Schechtman, G. *Asymptotic Theory of Finite Dimensional Normed Spaces*. Springer-Verlag, Berlin, Heidelberg, 1986. ISBN 0-387-16769-2.

Moulines, E. and Bach, F. R. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pp. 451–459, 2011.

Nagarajan, V. and Kolter, J. Z. Generalization in deep networks: The role of distance from initialization. *arXiv preprint arXiv:1901.01672*, 2019.

Nedić, A. and Bertsekas, D. Convergence rate of incremental subgradient algorithms. In *Stochastic optimization: algorithms and applications*, pp. 223–264. Springer, 2001.

Needell, D., Ward, R., and Srebro, N. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in Neural Information Processing Systems*, pp. 1017–1025, 2014.

Nesterov, Y. *Lectures on convex optimization*, volume 137. Springer, 2018.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. Towards understanding the role of overparametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

Nguyen, Q. and Hein, M. The loss surface of deep and wide neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2603–2612. JMLR. org, 2017.

Oymak, S. and Soltanolkotabi, M. Overparameterized nonlinear learning: Gradient descent takes the shortest path? *arXiv preprint arXiv:1812.10004*, 2018.

Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

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.

Schmidt, M. and Roux, N. L. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep information propagation. *arXiv preprint arXiv:1611.01232*, 2016.

Sedghi, H., Gupta, V., and Long, P. M. The singular values of convolutional layers. *arXiv preprint arXiv:1805.10408*, 2018.

Shamir, O. and Zhang, T. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pp. 71–79, 2013.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Smith, S. L., Elsen, E., and De, S. On the generalization benefit of noise in stochastic gradient descent. *arXiv preprint arXiv:2006.15081*, 2020.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147, 2013.

Tao, T. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012.

Telgarsky, M. Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485*, 2016.

Vaswani, S., Bach, F., and Schmidt, M. Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron. *arXiv preprint arXiv:1810.07288*, 2018.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4151–4161, 2017.

Wu, L., Zhu, Z., et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S. S., and Pennington, J. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. *arXiv preprint arXiv:1806.05393*, 2018.

Yang, G., Pennington, J., Rao, V., Sohl-Dickstein, J., and Schoenholz, S. S. A mean field theory of batch normalization. *arXiv preprint arXiv:1902.08129*, 2019.

Yin, D., Pananjady, A., Lam, M., Papailiopoulos, D., Ramchandran, K., and Bartlett, P. Gradient diversity: a key ingredient for scalable distributed learning. *arXiv preprint arXiv:1706.05699*, 2017.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018.

Appendix

We first briefly outline the different sections in the appendix.

- In appendix A, we provide details of our experimental setup, and provide additional empirical results on fully connected networks, convolutional networks and residual networks with the MNIST, CIFAR-10 and CIFAR-100 datasets.
- In appendix B, we state and prove a lemma on the near orthogonality of random vectors, which we refer to in the main text. This result is often attributed to Milman & Schechtman (1986).
- In appendix C, we provide some intuition on why many standard over-parameterized neural networks with low-rank Hessians might have low gradient confusion for a large set of weights near the minimizer.
- In appendix D, we provide the proofs of the theorems presented in the main section. In appendix D.1, we provide proofs of theorems 3.1 and 3.2. In appendix D.2, we provide the proof of lemma D.1, which we refer to in the main text. In appendix D.3, we provide proofs of theorem 5.1 and corollary 5.1. In appendix D.4, we provide the proof of theorem 4.1. In appendix D.5, we provide the proof of theorem 6.1.
- In appendix E, we briefly describe a few lemmas that we require in our analysis.
- In appendix F, we discuss the small weights assumption (assumption 1), which is required for theorem 5.1, corollary 5.1 and theorem 6.1 in the main text.

A. Additional experimental results

In this section, we present more details about our experimental setup, as well as, additional experimental results on a range of models (MLPs, CNNs and Wide ResNets) and a range of datasets (MNIST, CIFAR-10, CIFAR-100).

A.1. MLPs on MNIST

To further test the main claims in the paper, we performed additional experiments on an image classification problem on the MNIST dataset using fully connected neural networks. We iterated over neural networks of varying depth and width, and considered both the identity activation function (i.e., linear neural networks) and the tanh activation function. We also considered two different weight initializations that are popularly used and appropriate for these activation functions:

- The Glorot normal initializer (Glorot & Bengio, 2010) with weights initialized by sampling from the distribution $\mathcal{N}\big(0, 2/(\text{fan-in} + \text{fan-out})\big)$, where fan-in denotes the number of input units in the weight matrix, and fan-out denotes the number of output units in the weight matrix.
- The LeCun normal initializer (LeCun et al., 2012) with weights initialized by sampling from the distribution $\mathcal{N}\big(0, 1/\text{fan-in}\big)$.

We considered the simplified case where all hidden layers have the same width ℓ. Thus, the first weight matrix $\mathbf{W}_0 \in \mathbb{R}^{\ell \times d}$, where $d = 784$ for the 28×28-sized images of MNIST; all intermediate weight matrices $\{\mathbf{W}_p\}_{p \in [\beta-1]} \in \mathbb{R}^{\ell \times \ell}$; and the final layer $\mathbf{W}_\beta \in \mathbb{R}^{10 \times \ell}$ for the 10 image classes in MNIST. We added biases to each layer, which we initialized to 0. We used softmax cross entropy as the loss function. We use MLP-β-ℓ to denote this fully connected network of depth β and width ℓ. We used the standard train-valid-test splits of 40000-10000-10000 for MNIST.

This relatively simple model gave us the ability to iterate over a large number of combinations of network architectures of varying width and depth, and different activation functions and weight initializations. Linear neural networks are an efficient way to directly understand the effect of changing depth and width without increasing model complexity over linear regression. Thus, we considered both linear and non-linear neural networks in our experiments.

We used SGD with constant learning rates for training with a mini-batch size of 128 and trained each model for 40000 iterations (more than 100 epochs). The constant learning rate α was tuned over a logarithmically-spaced grid:

$$\alpha \in \{10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}.$$

We ran each experiment 10 times (making sure at least 8 of them ran till completion), and picked the learning rate that achieved the lowest training loss value on average at the end of training. Our grid search was such that the optimal learning rate never occurred at one of the extreme values tested.

To measure gradient confusion at the end training, we sampled 1000 pairs of mini-batches each of size 128 (the same size as the training batch size). We calculated gradients on each of these pairs of mini-batches, and then calculated the cosine similarity between them. To measure the worse-case gradient confusion, we computed the lowest gradient cosine similarity among all pairs. We explored the effect of changing depth and changing width on the different activation functions and weight initializations. We plot the final training loss achieved for each model and the minimum gradient cosine similarities calculated over the 1000 pairs of gradients at the end of training. For each point, we plot both the mean and the standard deviation over the 10 independent runs.

The effect of depth. We first present results showing the effect of network depth. We considered a fixed width of $\ell = 100$, and varied the depth of the neural network, on the log scale, as:

$$\beta \in \{3, 10, 30, 100, 300, 1000\}.$$

Figure 5 shows results on neural networks with identity and tanh activation functions for the two weight initializations considered (Glorot normal and LeCun normal). Similar to the experimental results in section 7, and matching our theoretical results in sections 4 and 5, we notice the consistent trend of gradient confusion increasing with increasing depth. This makes the networks harder to train with increasing depth, and this is evidenced by an increase in the final training loss value. By depth $\beta = 1000$, the increased gradient confusion effectively makes the network untrainable when using tanh non-linearities.

The effect of width. We explored the effect of width by varying the width of the neural network while keeping the depth fixed at $\beta = 300$. We chose a very deep model, which is essentially untrainable for small widths (with standard initialization techniques) and helps better illustrate the effects of increasing width. We varied the width of the network, again on the log scale, as:

$$\ell \in \{10, 30, 100, 300, 1000\}.$$

Crucially, note that the smallest network considered here, MLP-300-10, still has more than 50000 parameters (i.e., more than the number of training samples), and the network with width $\ell = 30$ has almost three times the number of parameters as the high-performing MLP-3-100 network considered in the previous section. Figure 6 show results on linear neural networks and neural networks with tanh activations for both the Glorot normal and LeCun normal initializations. As in the experimental results of section 7, we see the consistent trend of gradient confusion decreasing with increasing width. Thus, wider networks become easier to train and improve the final training loss value. We further see that when the width is too small ($\ell = 30$), the gradient confusion becomes drastically high and the network becomes completely untrainable.

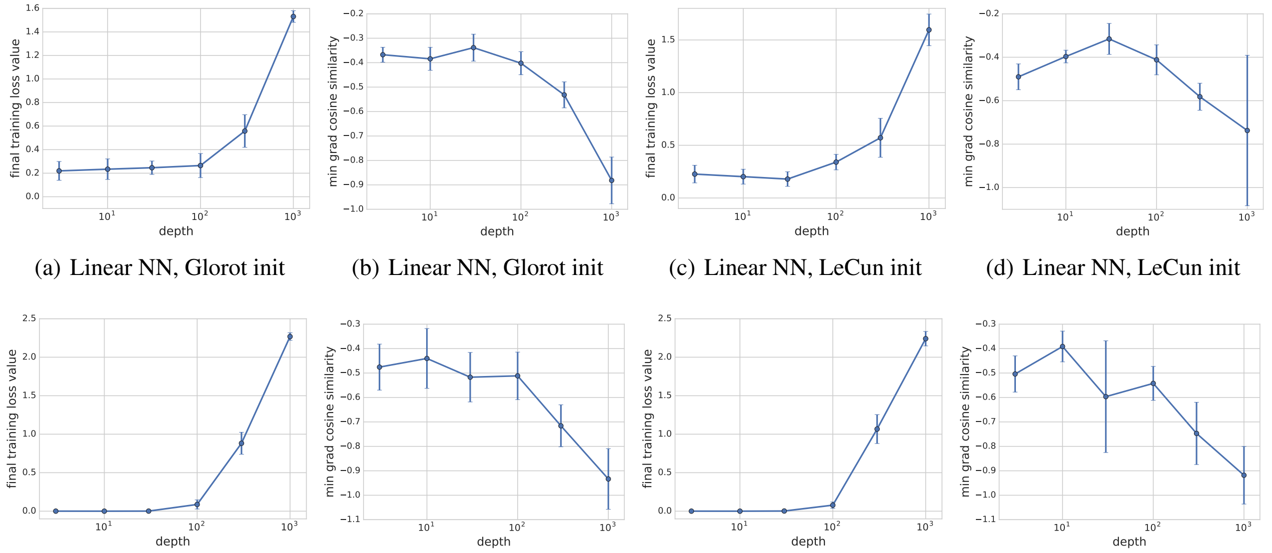
A.2. Additional experimental details for CNNs and WRNs

In this section, we review the details of our setup for the image classification experiments on CNNs and WRNs on the CIFAR-10 and CIFAR-100 datasets.

WIDE RESIDUAL NETWORKS

The Wide ResNet (WRN) architecture (Zagoruyko & Komodakis, 2016) for CIFAR datasets is a stack of three groups of residual blocks. There is a downsampling layer between two blocks, and the number of channels (width of a convolutional layer) is doubled after downsampling. In the three groups, the width of convolutional layers is $\{16\ell, 32\ell, 64\ell\}$, respectively. Each group contains β_r residual blocks, and each residual block contains two 3×3 convolutional layers equipped with ReLU activation, batch normalization and dropout. There is a 3×3 convolutional layer with 16 channels before the three groups of residual blocks. And there is a global average pooling, a fully-connected layer and a softmax layer after the three groups. The depth of WRN is $\beta = 6\beta_r + 4$.

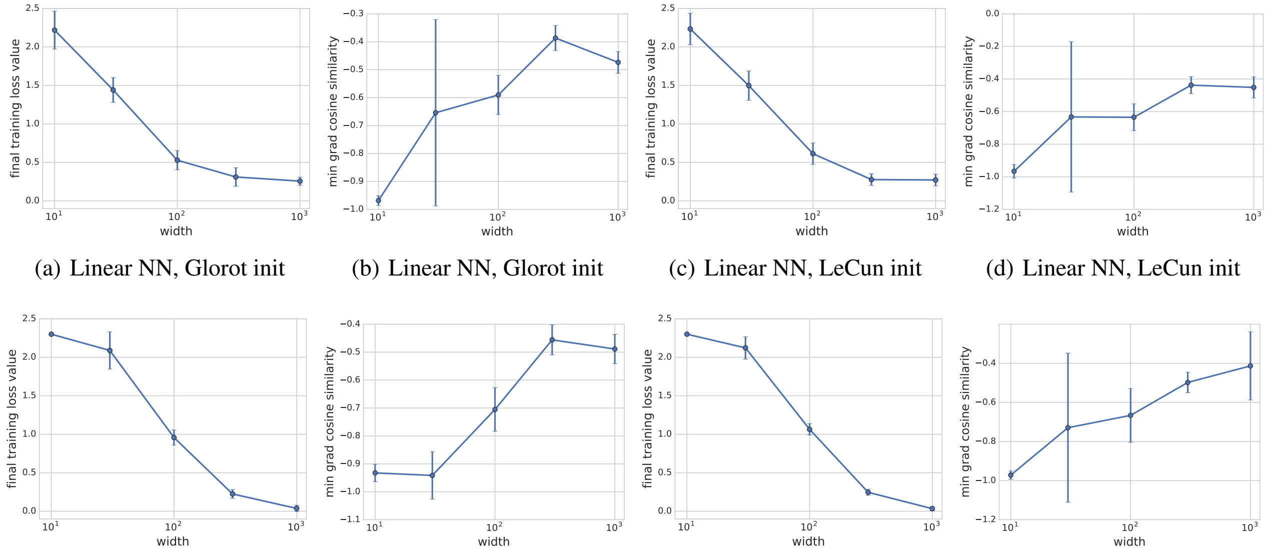
For our experiments, we turned off dropout. Unless otherwise specified, we also turned off batch normalization. We added biases to the convolutional layers when not using batch normalization to maintain model expressivity. We used the MSRA initializer (He et al., 2015) for the weights as is standard for this model, and used the same preprocessing steps for the CIFAR images as described in Zagoruyko & Komodakis (2016). This preprocessing step involves normalizing the images and doing data augmentation (Zagoruyko & Komodakis, 2016). We denote this network as WRN-β-ℓ, where β represents the depth and ℓ represents the width factor of the network.



(a) Linear NN, Glorot init (b) Linear NN, Glorot init (c) Linear NN, LeCun init (d) Linear NN, LeCun init

(e) Tanh NN, Glorot init (f) Tanh NN, Glorot init (g) Tanh NN, LeCun init (h) Tanh NN, LeCun init

Figure 5. Effect of varying depth on MLP-β-100.



(a) Linear NN, Glorot init (b) Linear NN, Glorot init (c) Linear NN, LeCun init (d) Linear NN, LeCun init

(e) Tanh NN, Glorot init (f) Tanh NN, Glorot init (g) Tanh NN, LeCun init (h) Tanh NN, LeCun init

Figure 6. Effect of varying width on MLP-300-ℓ.

To study the effect of depth, we considered WRNs with width factor $\ell = 2$ and depth varying as:

$$\beta \in \{16, 22, 28, 34, 40, 52, 76, 100\}.$$

For cleaner figures, we sometimes plot a subset of these results: $\beta \in \{16, 28, 40, 52, 76, 100\}$. To study the effect of width, we considered WRNs with depth $\beta = 16$ and width factor varying as:

$$\ell \in \{2, 3, 4, 5, 6\}.$$

CONVOLUTIONAL NEURAL NETWORKS

The WRN architecture contains skip connections that, as we show, help in training deep networks. To consider VGG-like convolutional networks, we consider a family of networks where we remove the skip connections from WRNs. Following the WRN convention, we denote these networks as CNN-β-ℓ, where β denotes the depth and ℓ denotes the width factor.

To study the effect of depth, we considered CNNs with width factor $\ell = 2$ and depth varying as:

$$\beta \in \{16, 22, 28, 34, 40\}.$$

To study the effect of width, we considered CNNs with depth $\beta = 16$ and width factor varying as:

$$\ell \in \{2, 3, 4, 5, 6\}.$$

HYPERPARAMETER TUNING AND OTHER DETAILS

We used SGD as the optimizer without any momentum. Following Zagoruyko & Komodakis (2016), we ran all experiments for 200 epochs with minibatches of size 128, and reduced the initial learning rate by a factor of 10 at epochs 80 and 160. We turned off weight decay for all our experiments.

We ran each individual experiment 5 times. We ignored any runs that were unable to decrease the loss from its initial value. We also made sure at least 4 out of the 5 independent runs ran till completion. When the learning rate is close to the threshold at which training is still possible, some runs may converge, while others may fail to converge. Thus, these checks ensure that we pick a learning rate that converges reliably in most cases on each problem. We show the standard deviation across runs in our plots.

We tuned the optimal initial learning rate for each model over a logarithmically-spaced grid:

$$\alpha \in \{10^1, 3\times 10^0, 10^0, 3\times 10^{-1}, 10^{-1}, 3\times 10^{-2}, 10^{-2}, 3\times 10^{-3}, 10^{-3}, 3\times 10^{-4}, 10^{-4}, 3\times 10^{-5}\},$$

and selected the run that achieved the lowest final training loss value (averaged over the independent runs). Our grid search was such that the optimal learning rate never occurred at one of the extreme values tested. We used the standard train-valid-test splits of 40000-10000-10000 for CIFAR-10 and CIFAR-100.

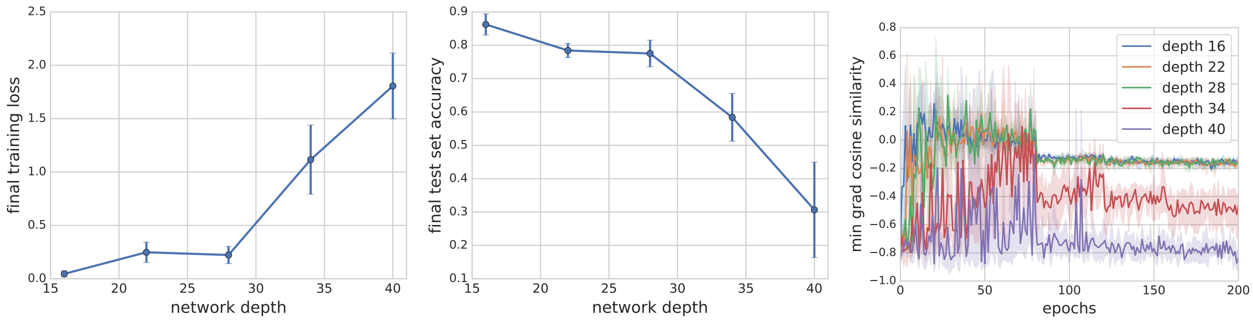
To measure gradient confusion, at the end of every training epoch, we sampled 100 pairs of mini-batches each of size 128 (the same size as the training batch size). We calculated gradients on each mini-batch, and then computed pairwise cosine similarities. To measure the worse-case gradient confusion, we computed the lowest gradient cosine similarity among all pairs. We also show the kernel density estimation of the pairwise gradient cosine similarities of the 100 minibatches sampled at the end of training (after 200 epochs), to see the concentration of the distribution. To do this, we combine together the 100 samples for each independent run and then perform kernel density estimation with a gaussian kernel on this data.

A.3. Additional plots for CIFAR-10 on CNNs

In section 7, we showed results for image classification using CNNs on CIFAR-10. In this section, we show some additional plots for this experiment. Figure 7 shows the effect of changing the depth, while figure 8 shows the effect of changing the width factor of the CNN. We see that the final training loss and test set accuracy values show the same trends as in section 7: deeper networks are harder to train, while wider networks are easier to train. As mentioned previously, theorems 3.1 and 3.2 indicate that we would expect the effect of gradient confusion to be more prominent near the end of training. From the plots we see that deeper networks have higher gradient confusion close to minimum, while wider networks have lower gradient confusion close to the minimum.

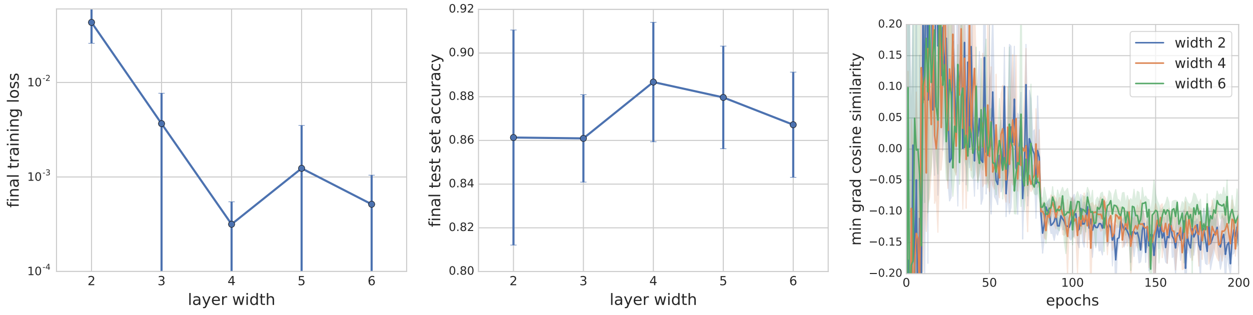
A.4. CIFAR-100 on CNNs

We now consider image classifications tasks with CNNs on the CIFAR-100 dataset. Figure 9 shows the effect of varying depth, while figure 10 shows the effect of varying width. We notice the same trends as in our results with CNNs on CIFAR-10. Interestingly, from the width results in figure 10, we see that while there is no perceptible change to the minimum pairwise gradient cosine similarity, the distribution still sharply concentrates around 0 with increasing width. Thus more gradients become orthogonal to each other with increasing width, implying that SGD on very wide networks becomes closer to decoupling over the data samples.



(a) (b) (c)

Figure 7. The effect of network depth with CNN-β-2 on CIFAR-10. The plots show the (a) final training loss values at the end of training, (b) final test set accuracy values at the end of training, and (c) the minimum of pairwise gradient cosine similarities during training.



(a) (b) (c)

Figure 8. The effect of width with CNN-16-ℓ on CIFAR-10. The plots show the (a) final training loss values at the end of training, (b) final test set accuracy values at the end of training, and the (c) minimum of pairwise gradient cosine similarities during training.

A.5. Image classification with WRNs on CIFAR-10 and CIFAR-100

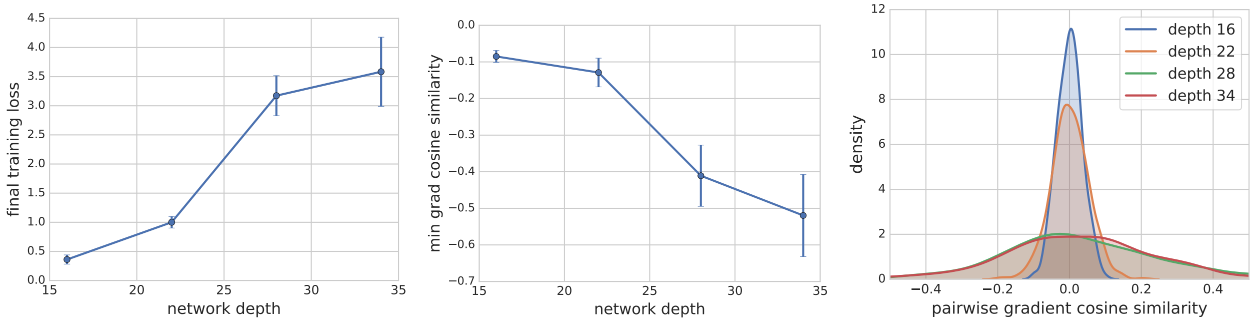
We now show results for image classification problems using wide residual networks (WRNs) on CIFAR-10 and CIFAR-100. The WRNs we consider do not have any batch normalization. Later we show results on the effect of adding batch normalization to these networks.

Figures 11 and 12 show results on the effect of depth using WRNs on CIFAR-10 and CIFAR-100 respectively. We again see the consistent trend of deeper networks having higher gradient confusion, making them harder to train. We further see that increasing depth results in the pairwise gradient cosine similarities concentrating less around 0.

Figures 13 and 14 show results on the effect of width using WRNs on CIFAR-10 and CIFAR-100 respectively. We see that increasing width typically lowers gradient confusion and helps the network achieve lower loss values. The pairwise gradient cosine similarities also typically concentrate around 0 with higher width. We also notice from these figures that in some cases, increasing width might lead to diminishing returns, i.e., the benefits of increased width diminish after a certain point, as one would expect.

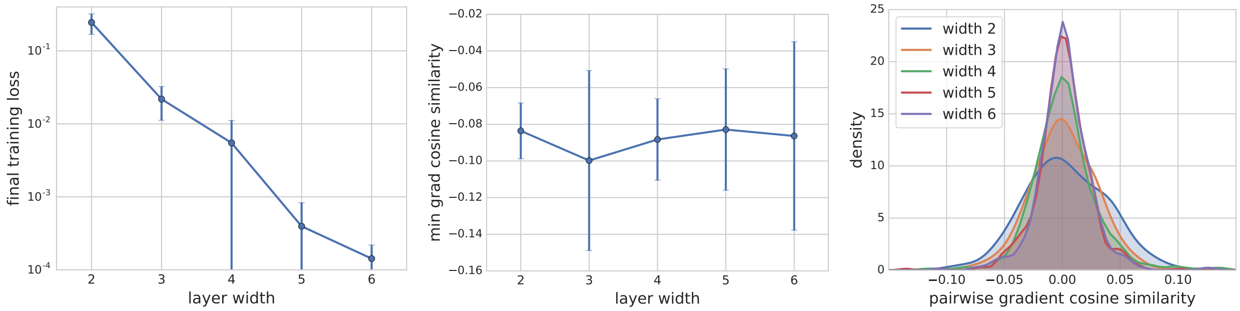
A.6. Effect of batch normalization and skip connections

In section 7 we showed results on the effect of adding batch normalization and skip connections to CNNs and WRNs on an image classification task on CIFAR-10. In this section, we present similar results for image classification on CIFAR-100. Similar to section 7, figure 15 shows that adding skip connections or batch normalization individually help in training deeper models, but these models still suffer from worsening results and increasing gradient confusion as the network gets deeper. Both these techniques together keep the gradient confusion relatively low even for very deep networks, significantly improving trainability of deep models.



(a) (b) (c)

Figure 9. The effect of network depth with CNN-β-2 on CIFAR-100. The plots show the (a) training loss values at the end of training, (b) minimum of pairwise gradient cosine similarities at the end of training, and the (c) kernel density estimate of the pairwise gradient cosine similarities at the end of training.



(a) (b) (c)

Figure 10. The effect of width with CNN-16-ℓ on CIFAR-100. The plots show the (a) training loss values at the end of training, (b) minimum of pairwise gradient cosine similarities at the end of training, and the (c) kernel density estimate of the pairwise gradient cosine similarities at the end of training.

B. Near orthogonality of random vectors

For completeness, we state and prove below a lemma on the near orthogonality of random vectors. This result is often attributed to Milman & Schechtman (1986).

Lemma B.1 (Near orthogonality of random vectors)**.** For vectors $\{\mathbf{x_i}\}_{i\in[N]}$ drawn uniformly from a unit sphere in d dimensions, and $\nu > 0$,

$$\Pr\left[\exists i, j \ \ |\mathbf{x}_i^\top \mathbf{x}_j| > \nu\right] \leq N^2 \sqrt{\tfrac{\pi}{8}} \exp\left(-\tfrac{d-1}{2}\nu^2\right).$$

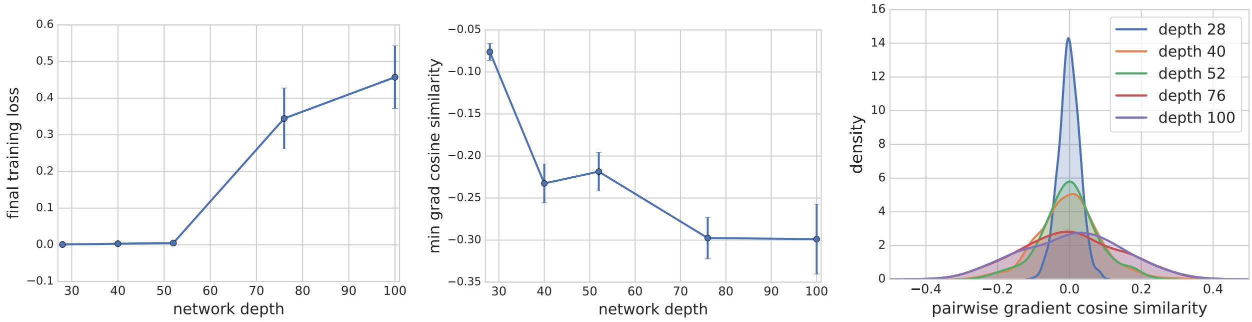
Proof. Given a fixed vector $\mathbf{x}$, a uniform random vector $\mathbf{y}$ satisfies $|\mathbf{x}^\top \mathbf{y}| \geq \nu$ only if $\mathbf{y}$ lies in one of two spherical caps: one centered at $\mathbf{x}$ and the other at $-\mathbf{x}$, and both with angular radius $\cos^{-1}(\nu) \leq \frac{\pi}{2} - \nu$. A simple result often attributed to Milman & Schechtman (1986) bounds the probability of lying in either of these caps as

$$\Pr[|\mathbf{x}^\top \mathbf{y}| \geq \nu] \leq \sqrt{\frac{\pi}{2}} \exp\left(-\frac{d-1}{2}\nu^2\right). \tag{5}$$

Because of rotational symmetry, the bound (5) holds if both $\mathbf{x}$ and $\mathbf{y}$ are chosen uniformly at random.

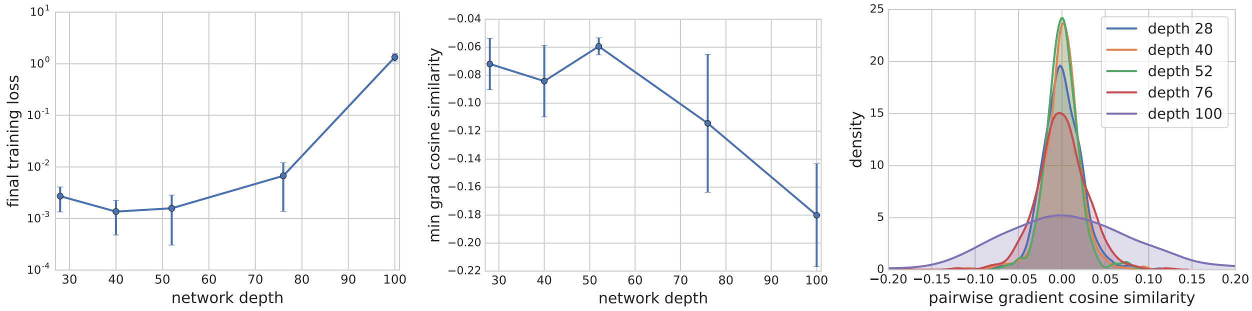
We next apply a union bound to control the probability that $|\mathbf{x}_i^\top \mathbf{x}_j| \geq \nu$ for some pair (i, j). There are fewer than $N^2/2$ such pairs, and so the probability of this condition is

$$\Pr[|\mathbf{x}_i^\top \mathbf{x}_j| \geq \nu, \text{for some } i, j] \leq \frac{N^2}{2} \sqrt{\frac{\pi}{2}} \exp\left(-\frac{d-1}{2}\nu^2\right). \qquad \square$$



(a) (b) (c)

Figure 11. The effect of depth with WRN-β-2 (no batch normalization) on CIFAR-10. The plots show the (a) training loss values at the end of training, (b) minimum of pairwise gradient cosine similarities at the end of training, and the (c) kernel density estimate of the pairwise gradient cosine similarities at the end of training.



(a) (b) (c)

Figure 12. The effect of depth with WRN-β-2 (no batch normalization) on CIFAR-100. The plots show the (a) training loss values at the end of training, (b) minimum of pairwise gradient cosine similarities at the end of training, and the (c) kernel density estimate of the pairwise gradient cosine similarities at the end of training.

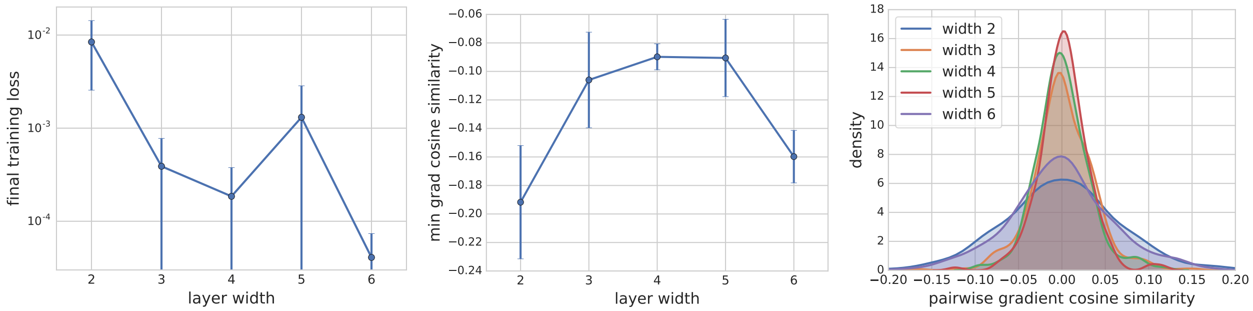
C. Low-rank Hessians lead to low gradient confusion

In this section, we show that low-rank random Hessians result in low gradient confusion. For clarity in presentation, suppose each f_i has a minimizer at the origin (the same argument can be easily extended to the more general case). Suppose also that there is a Lipschitz constant for the Hessian of each function f_i that satisfies $\|\mathbf{H}_i(\mathbf{w}) - \mathbf{H}_i(\mathbf{w}')\| \leq L_H \|\mathbf{w} - \mathbf{w}'\|$ (note that this is a standard optimization assumption (Nesterov, 2018), with evidence that it is applicable for neural networks (Martens, 2016)). Then $\nabla f_i(\mathbf{w}) = \mathbf{H}_i \mathbf{w} + \mathbf{e}$, where $\mathbf{e}$ is an error term bounded as: $\|\mathbf{e}\| \leq \frac{1}{2} L_H \|\mathbf{w}\|^2$, and we use the shorthand $\mathbf{H}_i$ to denote $\mathbf{H}_i(\mathbf{0})$. Then we have:

$$\begin{aligned}|\langle \nabla f_i(\mathbf{w}), \nabla f_j(\mathbf{w})\rangle| &= |\langle \mathbf{H}_i \mathbf{w}, \mathbf{H}_j \mathbf{w}\rangle| + \langle \mathbf{e}, \mathbf{H}_i \mathbf{w} + \mathbf{H}_j \mathbf{w}\rangle + \|\mathbf{e}\|^2 \\ &\leq \|\mathbf{w}\|^2 \|\mathbf{H}_i\| \|\mathbf{H}_j\| + \|\mathbf{e}\| \|\mathbf{w}\| (\|\mathbf{H}_i\| + \|\mathbf{H}_j\|) + \|\mathbf{e}\|^2 \\ &\leq \|\mathbf{w}\|^2 \|\mathbf{H}_i\| \|\mathbf{H}_j\| + \frac{1}{2} L_H \|\mathbf{w}\|^3 (\|\mathbf{H}_i\| + \|\mathbf{H}_j\|) + \frac{1}{4} L_H^2 \|\mathbf{w}\|^4.\end{aligned}$$

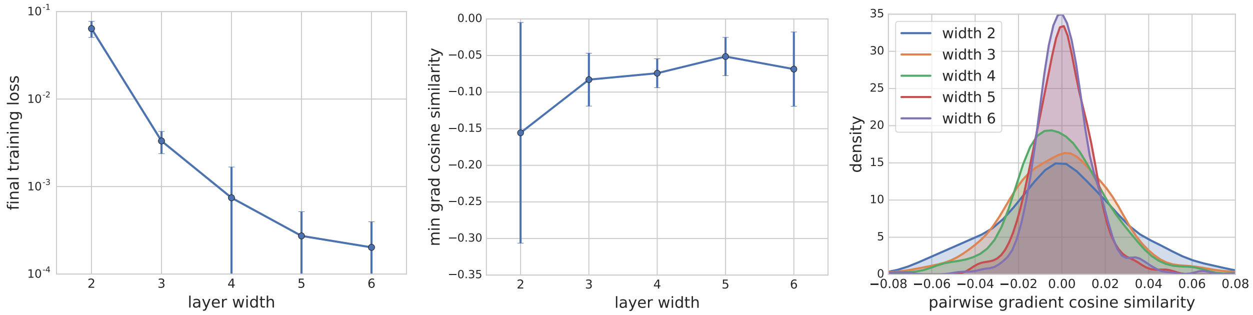
If the Hessians are sufficiently random and low-rank (e.g., of the form $\mathbf{H}_i = \mathbf{a}_i \mathbf{a}_i^\top$ where $\mathbf{a}_i \in \mathbb{R}^{N \times r}$ are randomly sampled from a unit sphere), then one would expect the terms in this expression to be small for all $\mathbf{w}$ within a neighborhood of the minimizer.

There is evidence that the Hessian at the minimizer is very low rank for many standard over-parameterized neural network models (Sagun et al., 2017; Cooper, 2018; Chaudhari et al., 2016; Wu et al., 2017; Ghorbani et al., 2019). While a bit non-rigorous, the above result nonetheless suggests that for many standard neural network models, the gradient confusion might be small for a large class of weights near the minimizer.



(a) (b) (c)

Figure 13. The effect of width with WRN-16-ℓ (no batch normalization) on CIFAR-10. The plots show the (a) training loss values at the end of training, (b) minimum of pairwise gradient cosine similarities at the end of training, and the (c) kernel density estimate of the pairwise gradient cosine similarities at the end of training.



(a) (b) (c)

Figure 14. The effect of width with WRN-16-ℓ (no batch normalization) on CIFAR-100. The plots show the (a) training loss values at the end of training, (b) minimum of pairwise gradient cosine similarities at the end of training, and the (c) kernel density estimate of the pairwise gradient cosine similarities at the end of training.

D. Missing proofs

D.1. Proofs of theorems 3.1 and 3.2

This section presents proofs for the convergence theorems of SGD presented in section 3, under the assumption of low gradient confusion. For clarity of presentation, we re-state each theorem before its proof.

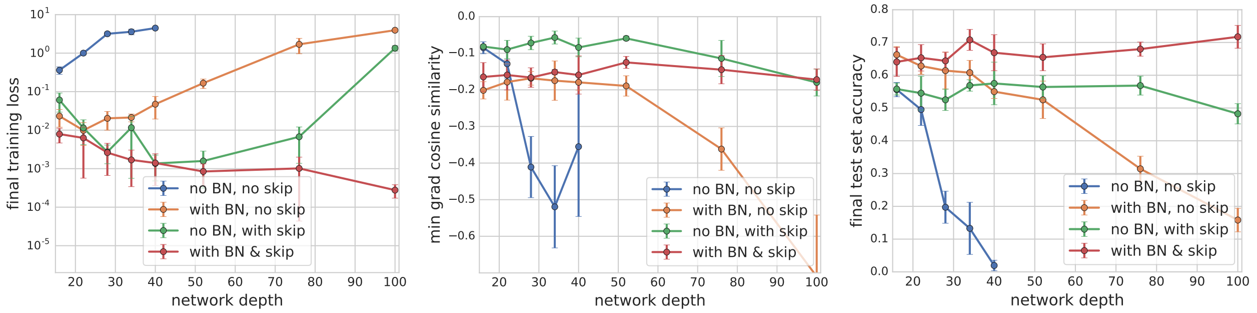
Theorem 3.1. *If the objective function satisfies (A1) and (A2), and has gradient confusion η, SGD converges linearly to a neighborhood of the minima of problem* (1) *as:*

$$\mathbb{E}[F(\mathbf{w}_T) - F^\star] \leq \rho^T (F(\mathbf{w}_0) - F^\star) + \tfrac{\alpha\eta}{1-\rho},$$

where $\alpha < \frac{2}{NL}$, $\rho = 1 - \frac{2\mu}{N}\big(\alpha - \frac{NL\alpha^2}{2}\big)$, $F^\star = \min_{\mathbf{w}} F(\mathbf{w})$ and $\mathbf{w}_0$ is the initialized weights.

Proof. Let $\tilde{i} \in [N]$ denote the index of the realized function $\tilde{f}_k$ in the uniform sampling from $\{f_i\}_{i\in[N]}$ at step k. From assumption (A1), we have

$$\begin{aligned}
F(\mathbf{w}_{k+1}) &\leq F(\mathbf{w}_k) + \langle \nabla F(\mathbf{w}_k), \, \mathbf{w}_{k+1} - \mathbf{w}_k \rangle + \frac{L}{2}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 \\
&= F(\mathbf{w}_k) - \alpha \langle \nabla F(\mathbf{w}_k), \, \nabla \tilde{f}_k(\mathbf{w}_k) \rangle + \frac{L\alpha^2}{2}\|\nabla \tilde{f}_k(\mathbf{w}_k)\|^2 \\
&= F(\mathbf{w}_k) - \Big(\frac{\alpha}{N} - \frac{L\alpha^2}{2}\Big)\|\nabla \tilde{f}_k(\mathbf{w}_k)\|^2 - \frac{\alpha}{N}\sum_{\forall i : i \neq \tilde{i}} \langle \nabla f_i(\mathbf{w}_k), \, \nabla \tilde{f}_k(\mathbf{w}_k) \rangle
\end{aligned}$$



(a)

(b)

(c)

Figure 15. The effect of adding skip connections and batch normalization to CNN-β-2 on CIFAR-100. Plots show the (a) training loss, (b) minimum pairwise gradient cosine similarities, and the (c) test accuracies at the end of training.

$$
\begin{aligned}
&\leq F(\mathbf{w}_k) - \Big(\frac{\alpha}{N} - \frac{L\alpha^2}{2}\Big)\|\nabla \tilde{f}_k(\mathbf{w}_k)\|^2 + \frac{\alpha(N-1)\eta}{N}, \\
&\leq F(\mathbf{w}_k) - \Big(\frac{\alpha}{N} - \frac{L\alpha^2}{2}\Big)\|\nabla \tilde{f}_k(\mathbf{w}_k)\|^2 + \alpha\eta,
\end{aligned}
$$

where the second-last inequality follows from definition 2.1. Let the learning rate $\alpha < 2/NL$. Then, using assumption (A2) and subtracting by $F^\star = \min_{\mathbf{w}} F(\mathbf{w})$ on both sides, we get

$$
F(\mathbf{w}_{k+1}) - F^\star \leq F(\mathbf{w}_k) - F^\star - 2\mu\Big(\frac{\alpha}{N} - \frac{L\alpha^2}{2}\Big)(\tilde{f}_k(\mathbf{w}_k) - \tilde{f}_k^\star) + \alpha\eta,
$$

where $\tilde{f}_k^\star = \min_{\mathbf{w}} \tilde{f}_k(\mathbf{w})$. It is easy to see that by definition we have, $\mathbb{E}_i[f_i^\star] \leq F^\star$. Moreover, from assumption that $\alpha < \frac{2}{NL}$, it implies that $\Big(\frac{\alpha}{N} - \frac{L\alpha^2}{2}\Big) > 0$. Therefore, taking expectation on both sides we get,

$$
\mathbb{E}[F(\mathbf{w}_{k+1}) - F^\star] \leq \Big(1 - \frac{2\mu\alpha}{N} + \mu L\alpha^2\Big)\mathbb{E}[F(\mathbf{w}_k) - F^\star] + \alpha\eta.
$$

Writing $\rho = 1 - \frac{2\mu\alpha}{N} + \mu L\alpha^2$, and unrolling the iterations, we get

$$
\begin{aligned}
\mathbb{E}[F(\mathbf{w}_{k+1}) - F^\star] &\leq \rho^{k+1}(F(\mathbf{w}_0) - F^\star) + \sum_{i=0}^{k} \rho^i \alpha\eta \\
&\leq \rho^{k+1}(F(\mathbf{w}_0) - F^\star) + \sum_{i=0}^{\infty} \rho^i \alpha\eta \\
&= \rho^{k+1}(F(\mathbf{w}_0) - F^\star) + \frac{\alpha\eta}{1-\rho}.
\end{aligned}
$$

□

Theorem 3.2. *If the objective satisfies (A1) and has gradient confusion η, then SGD converges to a neighborhood of a stationary point of problem* (1) *as:*

$$
\min_{k=1,\ldots,T} \mathbb{E}\|\nabla F(\mathbf{w}_k)\|^2 \leq \tfrac{\rho(F(\mathbf{w}_1) - F^\star)}{T} + \rho\eta,
$$

for $\alpha < \frac{2}{NL}$, $\rho = \frac{2N}{2-NL\alpha}$, *and* $F^\star = \min_{\mathbf{w}} F(\mathbf{w})$.

Proof. From theorem 3.1, we have:

$$
F(\mathbf{w}_{k+1}) \leq F(\mathbf{w}_k) - \Big(\frac{\alpha}{N} - \frac{L\alpha^2}{2}\Big)\|\nabla \tilde{f}_k(\mathbf{w}_k)\|^2 + \alpha\eta. \tag{6}
$$

Now we know that:

$$\mathbb{E}\|\nabla \tilde{f}_k(\mathbf{w}_k)\|^2 = \mathbb{E}\|\nabla \tilde{f}_k(\mathbf{w}_k) - \nabla F(\mathbf{w}_k)\|^2 + \mathbb{E}\|\nabla F(\mathbf{w}_k)\|^2 \geq \mathbb{E}\|\nabla F(\mathbf{w}_k)\|^2.$$

Thus, taking expectation and assuming the step size $\alpha < 2/(NL)$, we can rewrite equation 6 as:

$$\mathbb{E}\|\nabla F(\mathbf{w}_k)\|^2 \leq \frac{2N}{2\alpha - NL\alpha^2}\mathbb{E}[F(\mathbf{w}_k) - F(\mathbf{w}_{k+1})] + \frac{2N\eta}{2 - NL\alpha}.$$

Taking an average over T iterations, and using $F^\star = \min_{\mathbf{w}} F(\mathbf{w})$, we get:

$$\min_{k=1,\dots,T} \mathbb{E}\|\nabla F(\mathbf{w}_k)\|^2 \leq \frac{1}{T}\sum_{k=1}^{T} \mathbb{E}\|\nabla F(\mathbf{w}_k)\|^2 \leq \frac{2N}{2\alpha - NL\alpha^2}\frac{F(\mathbf{w}_1) - F^\star}{T} + \frac{2N\eta}{2 - NL\alpha}. \qquad \square$$

D.2. Proof of lemma D.1

Lemma D.1. *Consider the set of loss-functions $\{f_i(\mathbf{W})\}_{i\in[N]}$ where all f_i are either the square-loss function or the logistic-loss function. Recall that $f_i(\mathbf{W}) := f(\mathbf{W}, \mathbf{x}_i)$. Consider a feed-forward neural network as defined in equation 4 whose weights $\mathbf{W}$ satisfy assumption 1. Consider the gradient $\nabla_{\mathbf{W}} f_i(\mathbf{W})$ of each function f_i. From definition we have that $\nabla_{\mathbf{W}} f_i(\mathbf{W}) = \zeta_{\mathbf{x}_i}(\mathbf{W})\nabla_{\mathbf{W}} g_{\mathbf{W}}(\mathbf{x}_i)$, where we define $\zeta_{\mathbf{x}_i}(\mathbf{W}) = \partial f_i(\mathbf{W})/\partial g_{\mathbf{W}}$. Then we have the following properties.*

1. *When $\|\mathbf{x}\| \leq 1$ for every $p \in [\beta]$ we have $\|\nabla_{\mathbf{W}_p} g_{\mathbf{W}}(\mathbf{x}_i)\| \leq 1$.*
2. *There exists $0 < \zeta_0 \leq 2\sqrt{\beta}$, such that $|\zeta_{\mathbf{x}_i}(\mathbf{W})| \leq 2$, $\|\nabla_{\mathbf{x}_i}\zeta_{\mathbf{x}_i}(\mathbf{W})\|_2 \leq \zeta_0$, $\|\nabla_{\mathbf{W}}\zeta_{\mathbf{x}_i}(\mathbf{W})\|_2 \leq \zeta_0$.*

Proof. The first property is a direct consequence of assumption 1 and property (P2) of the activation function.

Let $\mathbf{W}$ denote the tuple $(\mathbf{W}_p)_{p\in[\beta]_0}$. Consider $|\zeta_{\mathbf{x}_i}(\mathbf{W})| = |\partial f_i(\mathbf{W})/\partial g_{\mathbf{W}}|$. In the case of square-loss function this evaluates to $|g_{\mathbf{W}}(\mathbf{x}) - \mathcal{C}(\mathbf{x})| \leq 2$. In case of logistic regression, this evaluates to $|\frac{-1}{1+\exp(\mathcal{C}(\mathbf{x}_i)g_{\mathbf{W}}(\mathbf{x}_i))}| \leq 1$. Now we consider $\|\nabla_{\mathbf{x}_i}\zeta_{\mathbf{x}_i}(\mathbf{W})\|$. Consider the squared loss function. We then have the following.

$$\begin{aligned}\|\nabla_{\mathbf{x}_i}\zeta_{\mathbf{x}_i}(\mathbf{W})\| &= \|\nabla_{\mathbf{x}_i} f'(\mathbf{W})\| \\ &= \|\nabla_{\mathbf{x}_i} g_{\mathbf{W}}(\mathbf{x}_i) - \mathcal{C}(\mathbf{x}_i)\| \\ &\leq \|\nabla_{\mathbf{x}_i} g_{\mathbf{W}}(\mathbf{x}_i)\| + 1.\end{aligned}$$

Likewise, consider the logistic-loss function. We then have the following.

$$\begin{aligned}\|\nabla_{\mathbf{x}_i}\zeta_{\mathbf{x}_i}(\mathbf{W})\| &\leq \left\|\frac{\mathcal{C}(\mathbf{x}_i)^2}{(1+\exp(\mathcal{C}(\mathbf{x}_i)g_{\mathbf{W}}(\mathbf{x}_i)))^2}\exp(\mathcal{C}(\mathbf{x}_i)g_{\mathbf{W}}(\mathbf{x}_i))\right\| \|\nabla_{\mathbf{x}_i} g_{\mathbf{W}}(\mathbf{x}_i)\| \\ &\leq \|\nabla_{\mathbf{x}_i} g_{\mathbf{W}}(\mathbf{x}_i)\|.\end{aligned}$$

Thus, it suffices to bound $\|\nabla_{\mathbf{x}_i} g_{\mathbf{W}}(\mathbf{x}_i)\|$. Using assumption 1 and the properties (P1), (P2) of σ, this can be upper-bounded by 1.

Consider $\nabla_{\mathbf{W}_p}\zeta_{\mathbf{x}_i}(\mathbf{W})$ for some layer index $p \in [\beta]_0$. We will show that $\|\nabla_{\mathbf{W}_p}\zeta_{\mathbf{x}_i}(\mathbf{W})\|_2 \leq 2$. Then it immediately follows that $\|\nabla_{\mathbf{W}}\zeta_{\mathbf{x}_i}(\mathbf{W})\|_2 \leq 2\sqrt{\beta}$. In the case of a squared loss function. We have the following.

$$\begin{aligned}\|\nabla_{\mathbf{W}_p}\zeta_{\mathbf{x}_i}(\mathbf{W})\| &= \|\nabla_{\mathbf{W}_p} f'(\mathbf{W})\| \\ &= \|\nabla_{\mathbf{W}_p} g_{\mathbf{W}}(\mathbf{x}_i) - \mathcal{C}(\mathbf{x}_i)\| \\ &\leq \|\nabla_{\mathbf{W}_p} g_{\mathbf{W}}(\mathbf{x}_i)\| + 1.\end{aligned}$$

Likewise, consider the logistic-loss function. We then have the following.

$$\begin{aligned}\|\nabla_{\mathbf{W}_p}\zeta_{\mathbf{x}_i}(\mathbf{W})\| &\leq \left\|\frac{\mathcal{C}(\mathbf{x}_i)^2}{(1+\exp(\mathcal{C}(\mathbf{x}_i)g_{\mathbf{W}}(\mathbf{x}_i)))^2}\exp(\mathcal{C}(\mathbf{x}_i)g_{\mathbf{W}}(\mathbf{x}_i))\right\| \|\nabla_{\mathbf{W}_p} g_{\mathbf{W}}(\mathbf{x}_i)\| \\ &\leq \|\nabla_{\mathbf{W}_p} g_{\mathbf{W}}(\mathbf{x}_i)\|.\end{aligned}$$

Since $\|\nabla_{\mathbf{W}_p} g_{\mathbf{W}}(\mathbf{x}_i)\| \leq 1$, we have that $\|\nabla_{\mathbf{W}_p}\zeta_{\mathbf{x}_i}(\mathbf{W})\| \leq 2$ in both the cases. Thus, $\zeta_0 = 2\sqrt{\beta}$. $\square$

D.3. Proofs of theorem 5.1 and corollary 5.1

In this section, we will present the proofs of theorem 5.1 and corollary 5.1.

Theorem 5.1. *Let $\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_\beta$ satisfy assumption 1. For some fixed constant $c > 0$, the gradient confusion bound (equation 3) holds with probability at least*

$$1 - N^2 \exp\left(\frac{-cd\eta^2}{16\zeta_0^4(\beta+2)^4}\right).$$

Proof. We show two key properties, namely bounded gradient and non negative expectation. We will then use both these properties to complete the proof.

Bounded gradient. For every $i \in [n]$ define $\zeta_{\mathbf{x}_i}(\mathbf{W}) := f'(\mathbf{W})$. For every $p \in [\beta]$ define $\mathbf{H}_p$ as follows.

$$\mathbf{H}_p(\mathbf{x}) := \sigma(\mathbf{W}_p \cdot \sigma(\mathbf{W}_{p-1} \cdot \sigma(\ldots \cdot \sigma(\mathbf{W}_0 \cdot \mathbf{x}) \ldots).$$

Fix an $i \in [N]$. Then we have the following recurrence

$$\begin{aligned} &g_\beta(\mathbf{x}_i) := \sigma'(H_\beta(\mathbf{x}_i)) \\ &\mathbf{g}_p(\mathbf{x}_i) := (\mathbf{W}_{p+1}^\top \cdot \mathbf{g}_{p+1}(\mathbf{x}_i)) \cdot \mathrm{Diag}(\sigma'(\mathbf{H}_p(\mathbf{x}_i))) \qquad \forall p \in \{0, 1, \ldots, \beta - 1\}. \end{aligned}$$

Then the gradients can be written in terms of the above quantities as follows.

$$\nabla_{\mathbf{W}_p} f_i(\mathbf{W}) = \mathbf{g}_p(\mathbf{x}_i) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i)^\top \qquad \forall p \in [\beta]_0.$$

We can write, the gradient confusion denote by $h_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j)$, as follows.

$$\zeta_{\mathbf{x}_i}(\mathbf{W})\zeta_{\mathbf{x}_j}(\mathbf{W}) \left(\sum_{p \in [\beta]_0} \mathrm{Tr}[\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i)^\top] \right). \tag{7}$$

We will now bound $\|\nabla_{(\mathbf{x}_i, \mathbf{x}_j)} h_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j)\|_2$. Consider $\nabla_{\mathbf{x}_i} h_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j)$. This can be written as follows.

$$\begin{aligned} &(\nabla_{\mathbf{x}_i}\zeta_{\mathbf{x}_i}(\mathbf{W}))\zeta_{\mathbf{x}_j}(\mathbf{W}) \left(\sum_{p \in [\beta]_0} \mathrm{Tr}[\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i)^\top] \right) + \\ &\qquad \zeta_{\mathbf{x}_i}(\mathbf{W})\zeta_{\mathbf{x}_j}(\mathbf{W}) \sum_{p \in [\beta]_0} \left[\nabla_{\mathbf{x}_i} \left(\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i) \right) \right]^\top. \end{aligned} \tag{8}$$

Observe that each of the entries in the diagonal matrix $\mathrm{Diag}(\sigma'(\mathbf{H}_p(\mathbf{x}_i)))$ is at most 1. Thus, we have that $\| \mathrm{Diag}(\sigma'(\mathbf{H}_p(\mathbf{x}_i)))\| \leq 1$.

We have the following relationship.

$$\begin{aligned} &\|g_\beta(\mathbf{x}_i)\| \leq 1 \\ &\|\mathbf{g}_p(\mathbf{x}_i)\| \leq \|\mathbf{W}_{p+1}^\top\| \|\mathbf{g}_{p+1}(\mathbf{x}_i))\| \| \mathrm{Diag}(\sigma'(\mathbf{H}_p(\mathbf{x}_i)))\| \leq 1 \qquad \forall p \in \{0, 1, \ldots, \beta - 1\}. \end{aligned}$$

Moreover we have,

$$\| \mathrm{Tr}[\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i)^\top]\| \leq \|\mathbf{H}_{p-1}(\mathbf{x}_i)\| \|\mathbf{g}_p(\mathbf{x}_i)^\top\| \|\mathbf{g}_p(\mathbf{x}_j)\| \|\mathbf{H}_{p-1}(\mathbf{x}_i)^\top\| \leq 1.$$

Consider $\|\nabla_{\mathbf{x}_i} \left(\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i) \right) \|$ for every $p \in [\beta]_0$.

This can be upper-bounded by,

$$\|\nabla_{\mathbf{x}_i} \mathbf{H}_{p-1}(\mathbf{x}_i)\| \|\mathbf{g}_p(\mathbf{x}_i)^\top\| \|\mathbf{g}_p(\mathbf{x}_j)\| \|\mathbf{H}_{p-1}(\mathbf{x}_i)\| + \|\mathbf{H}_{p-1}(\mathbf{x}_i)\| \|\nabla_{\mathbf{x}_i} \mathbf{g}_p(\mathbf{x}_i)^\top\| \|\mathbf{g}_p(\mathbf{x}_j)\| \|\mathbf{H}_{p-1}(\mathbf{x}_i)\|.$$

Note that $\nabla_{\mathbf{x}_i}\mathbf{H}_{p-1}(\mathbf{x}_i) = \mathbf{g}_1(\mathbf{x}_i)\cdot \operatorname{Diag}(\sigma'(\mathbf{W}_0\cdot\mathbf{x}_i))\cdot \mathbf{W}_0^\top \cdot \mathbf{g}_p(\mathbf{x}_i)^\top$. Thus, $\|\nabla_{\mathbf{x}_i}\mathbf{H}_{p-1}(\mathbf{x}_i)\| \le 1$. We will now show that $\|\nabla_{\mathbf{x}_i}\mathbf{g}_p(\mathbf{x}_i)\| \le \beta - p + 1$. We prove this inductively. Consider the base case when $p = \beta$.

$$\|\nabla_{\mathbf{x}_i}\mathbf{g}_\beta(\mathbf{x}_i)\| = \|\nabla_{\mathbf{x}_i}\sigma'(\mathbf{H}_\beta(\mathbf{x}_i))\| \le 1 = \beta - \beta + 1.$$

Now, the inductive step.

$$\|\nabla_{\mathbf{x}_i}\mathbf{g}_p(\mathbf{x}_i)\| \le \|\nabla_{\mathbf{x}_i}\mathbf{g}_{p+1}(\mathbf{x}_i)\| + \|\nabla_{\mathbf{x}_i}\operatorname{Diag}(\sigma'(\mathbf{H}_p(\mathbf{x}_i)))\| \le \beta - p \le \beta - p + 1.$$

Thus, using equation 8 and the above arguments, we obtain, $\|\nabla_{\mathbf{x}_i}h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)\|_2 \le \zeta_0^2(\beta+1) + \zeta_0^2(\beta+1)(\beta+2) \le 2\zeta_0^2(\beta+2)^2$ and thus, $\|\nabla_{(\mathbf{x}_i,\mathbf{x}_j)}h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)\|_2 \le 4\zeta_0^2(\beta+2)^2$.

Non-negative expectation.

$$\begin{aligned}\mathbb{E}_{\mathbf{x}_i,\mathbf{x}_j}[h(\mathbf{x}_i,\mathbf{x}_j)] =& \mathbb{E}_{\mathbf{x}_i,\mathbf{x}_j}[\langle \nabla f_i(\mathbf{W}), \nabla f_j(\mathbf{W})\rangle] \\ &= \langle \mathbb{E}_{\mathbf{x}_i}[\nabla f_i(\mathbf{W})], \mathbb{E}_{\mathbf{x}_j}[\nabla f_j(\mathbf{W})]\rangle \\ &= \|\mathbb{E}_{\mathbf{x}_i}[\nabla f_i(\mathbf{W})]\|^2 \ge 0. \end{aligned} \tag{9}$$

We have used the fact that $\nabla f_i(\mathbf{W})$ and $\nabla f_j(\mathbf{W})$ are identically distributed and independent.

Concentration of Measure. We combine the two properties as follows. From **Non-negative Expectation** property and equation 26, we have that

$$\Pr[h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j) \le -\eta] \le \Pr[h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j) \le \mathbb{E}_{(\mathbf{x}_i,\mathbf{x}_j)}[h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)] - \eta] \le \exp\left(\frac{-cd\eta^2}{16\zeta_0^4(\beta+2)^4}\right). \tag{10}$$

To obtain the probability that *some* value of $h_{\mathbf{w}}(\nabla_{\mathbf{w}} f_i, \nabla_{\mathbf{w}} f_j)$ lies below $-\eta$, we use a union bound. There are $N(N-1)/2 < N^2/2$ possible pairs of data points to consider, and so this probability is bounded above by $N^2\exp\left(\frac{-cd\eta^2}{16\zeta_0^4(\beta+2)^4}\right)$. □

D.3.1. Proof of corollary 5.1

Before we prove corollary 5.1 we first prove the following helper lemma.

Lemma D.2. *Suppose* $\max_{\mathbf{W}}\|\nabla_{\mathbf{W}} f_i(\mathbf{W})\| \le M$, *and both* $\nabla_{\mathbf{W}} f_i(\mathbf{w})$ *and* $\nabla_{\mathbf{W}} f_j(\mathbf{W})$ *are Lipschitz in* $\mathbf{W}$ *with constant* L. *Then* $h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)$ *is Lipschitz in* $\mathbf{W}$ *with constant* $2LM$.

Proof. We view $\mathbf{W}$ as flattened vector. We now prove the above result for these two vectors. For two vectors $\mathbf{w}, \mathbf{w}'$,

$$\begin{aligned}&|h_{\mathbf{w}}(\mathbf{x}_i,\mathbf{x}_j) - h_{\mathbf{w}'}(\mathbf{x}_i,\mathbf{x}_j)| \\ &= |\langle \nabla_{\mathbf{w}} f_i(\mathbf{w}), \nabla_{\mathbf{w}} f_j(\mathbf{w})\rangle - \langle \nabla_{\mathbf{w}'} f_i(\mathbf{w}'), \nabla_{\mathbf{w}'} f_j(\mathbf{w}')\rangle| \\ &= |\langle \nabla_{\mathbf{w}} f_i(\mathbf{w}) - \nabla_{\mathbf{w}'} f_i(\mathbf{w}') + \nabla_{\mathbf{w}'} f_i(\mathbf{w}'), \nabla_{\mathbf{w}} f_j(\mathbf{w})\rangle \\ &\quad - \langle \nabla_{\mathbf{w}'} f_i(\mathbf{w}'), \nabla_{\mathbf{w}'} f_j(\mathbf{w}') - \nabla_{\mathbf{w}} f_j(\mathbf{w}) + \nabla_{\mathbf{w}} f_j(\mathbf{w})\rangle| \\ &= |\langle \nabla_{\mathbf{w}} f_i(\mathbf{w}) - \nabla_{\mathbf{w}'} f_i(\mathbf{w}'), \nabla_{\mathbf{w}} f_j(\mathbf{w})\rangle - \langle \nabla_{\mathbf{w}'} f_i(\mathbf{w}'), \nabla_{\mathbf{w}'} f_j(\mathbf{w}') - \nabla_{\mathbf{w}} f_j(\mathbf{w})\rangle| \\ &\le |\langle \nabla_{\mathbf{w}} f_i(\mathbf{w}) - \nabla_{\mathbf{w}'} f_i(\mathbf{w}'), \nabla_{\mathbf{w}} f_j(\mathbf{w})\rangle| + |\langle \nabla_{\mathbf{w}'} f_i(\mathbf{w}'), \nabla_{\mathbf{w}'} f_j(\mathbf{w}') - \nabla_{\mathbf{w}} f_j(\mathbf{w})\rangle| \\ &\le \|\nabla_{\mathbf{w}} f_i(\mathbf{w}) - \nabla_{\mathbf{w}'} f_i(\mathbf{w}')\|\|\nabla_{\mathbf{w}} f_j(\mathbf{w})\| + \|\nabla_{\mathbf{w}'} f_i(\mathbf{w}')\|\|\nabla_{\mathbf{w}'} f_j(\mathbf{w}') - \nabla_{\mathbf{w}} f_j(\mathbf{w})\| \\ &\le L\|\mathbf{w} - \mathbf{w}'\|\|\nabla_{\mathbf{w}} f_j(\mathbf{w})\| + \|\nabla_{\mathbf{w}'} f_i(\mathbf{w}')\|L\|\mathbf{w}' - \mathbf{w}\| \\ &\le 2LM\|\mathbf{w} - \mathbf{w}'\|. \end{aligned}$$

Here the first inequality uses the triangle inequality, the second inequality uses the Cauchy-Schwartz inequality, and the third and fourth inequalities use the assumptions that $\nabla_{\mathbf{w}} f_i(\mathbf{w})$ and $\nabla_{\mathbf{w}} f_j(\mathbf{w})$ are Lipschitz in $\mathbf{w}$ and have bounded norm. □

We are now ready to prove the corollary, which we restate here. The proof uses a standard "epsilon-net" argument; we identify a fine net of points within the ball $\mathcal{B}_r$. If the gradient confusion is small at every point in this discrete set, and the gradient confusion varies slowly enough with $\mathbf{W}$, when we can guarantee small gradient confusion at every point in $\mathcal{B}_r$.

Corollary 5.1. *Select a point $\mathbf{W} = (\mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_\beta)$, satisfying assumption 1. Consider a ball $\mathcal{B}_r$ centered at $\mathbf{W}$ of radius $r > 0$. If the data $\{\mathbf{x}_i\}_{i\in[N]}$ are sampled uniformly from a unit sphere, then the gradient confusion bound in equation 3 holds uniformly at all points $\mathbf{W}' \in \mathcal{B}_r$ with probability at least*

$$\begin{cases} 1 - N^2 \exp\left(-\frac{cd\eta^2}{64\zeta_0^4(\beta+2)^4}\right), & \textit{if } r \le \eta/4\zeta_0^2, \\ 1 - N^2 \exp\left(-\frac{cd\eta^2}{64\zeta_0^4(\beta+2)^4} + \frac{8d\zeta_0^2 r}{\eta}\right), & \textit{otherwise}. \end{cases}$$

Proof. Define the function $h^+(\mathbf{W}) = \max_{ij} h_\mathbf{W}(\mathbf{x}_i, \mathbf{x}_j)$. Our goal is to find conditions under which $h^+(\mathbf{W}) > -\eta$ for all $\mathbf{W}$ in a large set. To derive such conditions, we will need a Lipschitz constant for $h^+(\mathbf{W})$, which is no larger than the maximal Lipschitz constant of $h_\mathbf{W}(\mathbf{x}_i, \mathbf{x}_j)$ for all i, j. We have that $\|\nabla_\mathbf{W} f_i\| = \|\zeta_{\mathbf{x}_i}(\mathbf{W})\mathbf{x}_i\| \le \zeta_0$. Now we need to get a $\mathbf{W}$-Lipschitz constants for $\nabla_{\mathbf{x}_i} f_i = \zeta_{\mathbf{x}_i}(\mathbf{W})\mathbf{x}_i$. By lemma D.1, we have $\|\nabla_\mathbf{W}(\zeta_{\mathbf{x}_i}(\mathbf{W})\mathbf{x}_i)\| = \|(\nabla_\mathbf{W}\zeta_{\mathbf{x}_i}(\mathbf{W}))\mathbf{x}_i\| \le \zeta_0$. Using lemma D.2, we see that $2\zeta_0^2$ is a Lipschitz constant for $h_\mathbf{W}(\mathbf{x}_i, \mathbf{x}_j)$, and thus also $h^+(\mathbf{W})$.

Now, consider a minimizer $\mathbf{W}$ of the objective, and a ball $\mathcal{B}_r$ around this point of radius r. Define the constant $\epsilon = \frac{\eta}{4\zeta_0^2}$, and create an ϵ-net of points $\mathcal{N}_\epsilon = \{\mathbf{W}_i\}$ inside the ball. This net is sufficiently dense that any $\mathbf{W}' \in \mathcal{B}_r$ is at most ϵ units away from some $\mathbf{W}_i \in \mathcal{N}_\epsilon$. Furthermore, because $h^+(\mathbf{W})$ is Lipschitz in $\mathbf{W}$, $|h^+(\mathbf{W}') - h^+(\mathbf{W}_i)| \le 2\zeta_0^2\epsilon = \eta/2$.

We now know the following: if we can guarantee that

$$h^+(\mathbf{W}_i) \ge -\eta/2, \text{ for all } \mathbf{W}_i \in \mathcal{N}_\epsilon, \tag{11}$$

then we also know that $h^+(\mathbf{W}') \ge -\eta$ for all $\mathbf{W}' \in \mathcal{B}_r$. For this reason, we prove the result by bounding the probability that (11) holds. It is known that $\mathcal{N}_\epsilon$ can be constructed so that $|\mathcal{N}_\epsilon| \le (2r/\epsilon + 1)^d = (8\zeta_0^2 r/\eta + 1)^d$ (see Vershynin (2018), corollary 4.1.13). Theorem 5.1 provides a bound on the probability that each individual point in the net satisfies condition (11). Using a union bound, we see that all points in the net satisfy this condition with probability at least

$$1 - N^2\left(\frac{8\zeta_0^2 r}{\eta} + 1\right)^d \exp\left(-\frac{cd(\eta/2)^2}{16\zeta_0^4}\right) \tag{12}$$

$$= 1 - N^2 \exp(d\log(8\zeta_0^2 r/\eta + 1)) \exp\left(-\frac{cd\eta^2}{64\zeta_0^4}\right) \tag{13}$$

$$\ge 1 - N^2 \exp(8d\zeta_0^2 r/\eta) \exp\left(-\frac{cd\eta^2}{64\zeta_0^4}\right) \tag{14}$$

$$= 1 - N^2 \exp\left(-\frac{cd\eta^2}{64\zeta_0^4} + \frac{8d\zeta_0^2 r}{\eta}\right). \tag{15}$$

Finally, note that, if $r < \epsilon$, then we can form a net with $|\mathcal{N}_\epsilon| = 1$. In this case, the probability of satisfying (11) is at least

$$1 - N^2 \exp\left(-\frac{cd(\eta/2)^2}{64\zeta_0^4}\right). \qquad \square$$

D.4. Proof of theorem 4.1

Theorem 4.1. *Let $\mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_\beta$ be weight matrices chosen according to strategy 4.1. There exists fixed constants $c_1, c_2 > 0$ such that we have the following.*

1. *Consider a fixed but arbitrary dataset $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ with $\|\mathbf{x}_i\| \le 1$ for every $i \in [N]$. For $\eta > 4$, the gradient confusion bound in equation 3 holds with probability at least*

$$1 - \beta\exp\left(-c_1\kappa^2\ell^2\right) - N^2\exp\left(\frac{-c\ell^2\beta(\eta-4)^2}{64\zeta_0^4(\beta+2)^4}\right).$$

2. *If the dataset $\{\mathbf{x}_i\}_{i\in[N]}$ is such that each $\mathbf{x}_i$ is an i.i.d. sample from the surface of d-dimensional unit sphere, then for every $\eta > 0$ the gradient confusion bound in equation 3 holds with probability at least*

$$1 - \beta\exp\left(-c_1\kappa^2\ell^2\right) - N^2\exp\left(\frac{-c_2(\ell d + \ell^2\beta)\eta^2}{16\zeta_0^4(\beta+2)^4}\right).$$

Both parts in theorem 4.1 depend on the following argument. From theorem 2.3.8 and Proposition 2.3.10 in Tao (2012) with appropriate scaling[7], we have for every $p = 1, \dots, \beta$ we have that the matrix norm $\|\mathbf{W}_p\| \leq 1$ with probability at least $1 - \beta \exp\left(-c_1 \kappa^2 \ell^2\right)$ and $\|\mathbf{W}_0\| \leq 1$ with probability at least $1 - \exp\left(-c_1 \kappa^2 d^2\right)$ when the weight matrices are initialized according to strategy 4.1. Thus, conditioning on this event it implies that these matrices satisfy assumption 1. The proof strategy is similar to that of theorem 5.1. We will first show that the gradient of the function $h(.,.)$ as defined in equation (7) *with respect to the weights* is bounded. Note that in part (1) the random variable is the set of weight matrices $\{W_p\}_{p \in [\beta]}$. Thus, the dimension used to invoke theorem E.1 is at most $\ell^2 \beta$. In part (2) along with the weights, the data $\mathbf{x} \in \mathbb{R}^d$ is also random. Thus, the dimension used to invoke theorem E.1 is at most $\ell d + \ell^2 \beta$. Combining this with theorem E.1, the bound on the gradient of $h(.,.)$ and taking a union bound, we get the respective parts of the theorem. Thus, all it remains to prove is the bound on the gradient of the function $h(.,.)$ as defined in equation (7) *with respect to the weights* conditioning on the event that $\|\mathbf{W}_p\| \leq 1$ for every $p \in \{0, 1, \dots, \beta\}$.

We obtain the following analogue of equation (8).

$$
\begin{aligned}
&(\nabla_{\mathbf{W}} \zeta_{\mathbf{x}_i}(\mathbf{W})) \zeta_{\mathbf{x}_j}(\mathbf{W}) \left(\sum_{p \in [\beta]_0} \mathrm{Tr}[\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i)^\top] \right) + \\
&\qquad (\nabla_{\mathbf{W}} \zeta_{\mathbf{x}_j}(\mathbf{W})) \zeta_{\mathbf{x}_i}(\mathbf{W}) \left(\sum_{p \in [\beta]_0} \mathrm{Tr}[\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i)^\top] \right) + \\
&\qquad\qquad \zeta_{\mathbf{x}_i}(\mathbf{W}) \zeta_{\mathbf{x}_j}(\mathbf{W}) \sum_{p \in [\beta]_0} \left[\nabla_{\mathbf{W}} \left(\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i) \right) \right]^\top . \qquad (16)
\end{aligned}
$$

As in the case of the proof for theorem 5.1, we will upper-bound the ℓ_2-norm of the above expression. In particular, we show the following.

$$
\left\| (\nabla_{\mathbf{W}} \zeta_{\mathbf{x}_i}(\mathbf{W})) \zeta_{\mathbf{x}_j}(\mathbf{W}) \left(\sum_{p \in [\beta]_0} \mathrm{Tr}[\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i)^\top] \right) \right\|_2 \leq 2\zeta_0^2 (\beta + 2)^2. \qquad (17)
$$

$$
\left\| (\nabla_{\mathbf{W}} \zeta_{\mathbf{x}_j}(\mathbf{W})) \zeta_{\mathbf{x}_i}(\mathbf{W}) \left(\sum_{p \in [\beta]_0} \mathrm{Tr}[\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i)^\top] \right) \right\|_2 \leq 2\zeta_0^2 (\beta + 2)^2. \qquad (18)
$$

$$
\left\| \zeta_{\mathbf{x}_i}(\mathbf{W}) \zeta_{\mathbf{x}_j}(\mathbf{W}) \sum_{p \in [\beta]_0} \left[\nabla_{\mathbf{W}} \left(\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i) \right) \right]^\top \right\|_2 \leq 4\zeta_0^2 (\beta + 2)^2. \qquad (19)
$$

Equations (17) and 18 follow from the the fact that $\|(\nabla_{\mathbf{W}} \zeta_{\mathbf{x}_i}(\mathbf{W}))\|_2 \leq \zeta_0$ and the arguments in the proof for theorem 5.1. We will now show the proof sketch for equation (19). For every $p \in [\beta]_0$, consider $\|\nabla_{\mathbf{W}} \left(\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i) \right)\|$. Using the symmetry between $\mathbf{x}_i$ and $\mathbf{x}_j$, the expression can be upper-bounded by,

$$
2\|\nabla_{\mathbf{W}} \mathbf{H}_{p-1}(\mathbf{x}_i)\| \|\mathbf{g}_p(\mathbf{x}_i)^\top\| \|\mathbf{g}_p(\mathbf{x}_j)\| \|\mathbf{H}_{p-1}(\mathbf{x}_i)\| + 2\|\mathbf{H}_{p-1}(\mathbf{x}_i)\| \|\nabla_{\mathbf{W}} \mathbf{g}_p(\mathbf{x}_i)^\top\| \|\mathbf{g}_p(\mathbf{x}_j)\| \|\mathbf{H}_{p-1}(\mathbf{x}_i)\|.
$$

As before we can use an inductive argument to find the upper-bound and thus, we obtain the following which implies equation (19).

$$
\|\nabla_{\mathbf{W}} \left(\mathbf{H}_{p-1}(\mathbf{x}_i) \cdot \mathbf{g}_p(\mathbf{x}_i)^\top \cdot \mathbf{g}_p(\mathbf{x}_j) \cdot \mathbf{H}_{p-1}(\mathbf{x}_i) \right)\| \leq 4(\beta + 2)^2.
$$

Next, we show that the expected value can be lower-bounded by -4 as in the case of theorem 4.1 above. Combining these two gives us the desired result. Consider $\mathbb{E}_{\mathbf{W}}[h_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j)]$. We compute this expectation iteratively as follows.

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{W}}[h_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j)] \\
&= \mathbb{E}_{\mathbf{W}_0}[\mathbb{E}_{\mathbf{W}_1}[\dots \mathbb{E}_{\mathbf{W}_\beta}[h_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j)]
\end{aligned}
$$

[7] In particular, each entry has to be scaled by $\frac{1}{\ell}$ for matrices $\{\mathbf{W}_p\}_{p \in [\beta]}$ and $\frac{1}{d}$ for the matrix $\mathbf{W}_0$.

$$\geq -4\mathbb{E}_{\mathbf{W}_0}\left[\mathbb{E}_{\mathbf{W}_1}\left[\ldots\mathbb{E}_{\mathbf{W}_\beta}\left[\sum_{p\in[\beta]_0}\mathrm{Tr}(\mathbf{H}_{p-1}(\mathbf{x}_i)\cdot\mathbf{g}_p(\mathbf{x}_i)^\top\cdot\mathbf{g}_p(\mathbf{x}_j)\cdot\mathbf{H}_{p-1}(\mathbf{x}_i)^\top)\right]\right]\right].$$

The inequality combines equation 7 with Lemma D.1. We now prove the following inequality.

$$\mathbb{E}_{\mathbf{W}_0}\left[\mathbb{E}_{\mathbf{W}_1}\left[\ldots\mathbb{E}_{\mathbf{W}_\beta}\left[\sum_{p\in[\beta]_0}\mathrm{Tr}(\mathbf{H}_{p-1}(\mathbf{x}_i)\cdot\mathbf{g}_p(\mathbf{x}_i)^\top\cdot\mathbf{g}_p(\mathbf{x}_j)\cdot\mathbf{H}_{p-1}(\mathbf{x}_i)^\top)\right]\right]\right]\leq 1. \tag{20}$$

Consider the inner-most expectation. Note that the only random variable is $\mathbf{W}_\beta$. Moreover, the term inside the trace is *scalar*. Note that the activation function σ satisfies $|\sigma'(x)|\leq 1$. Using the linearity of expectation, the LHS in equation (20) can be upper-bounded by the following.

$$\mathbb{E}_{\mathbf{W}_0}\Big[\mathbb{E}_{\mathbf{W}_1}\Big[\ldots\mathbb{E}_{\mathbf{W}_{\beta-1}}\Big[\mathrm{Tr}(\mathbf{H}_{\beta-1}(\mathbf{x}_i)\cdot\mathbf{H}_{\beta-1}(\mathbf{x}_i)^\top)\Big]\Big]\Big] \tag{21}$$

$$+\,\mathbb{E}_{\mathbf{W}_0}\Big[\mathbb{E}_{\mathbf{W}_1}\Big[\ldots\mathbb{E}_{\mathbf{W}_\beta}\Big[\sum_{p\in[\beta]_0\setminus\{\beta\}}\mathrm{Tr}(\mathbf{H}_{p-1}(\mathbf{x}_i)\cdot\mathbf{g}_p(\mathbf{x}_i)^\top\cdot\mathbf{g}_p(\mathbf{x}_j)\cdot\mathbf{H}_{p-1}(\mathbf{x}_i)^\top)\Big]\Big]\Big]. \tag{22}$$

The first sum in the above expression can be upper-bounded by 1, since $|\sigma(x)|\leq 1$. We will now show that the second sum is 0. Consider the inner-most expectation. The weights $\mathbf{W}_\beta$ appears only in the expression $\mathbf{g}_p(\mathbf{x}_i)^\top\cdot\mathbf{g}_p(\mathbf{x}_j)$. Moreover, note that every entry in $\mathbf{W}_\beta$ is an i.i.d. normal random variable with mean 0. Thus, the second summand simplifies to,

$$\mathbb{E}_{\mathbf{W}_0}\Big[\mathbb{E}_{\mathbf{W}_1}\Big[\ldots\mathbb{E}_{\mathbf{W}_{\beta-1}}\Big[\sum_{p\in[\beta]_0\setminus\{\beta,\beta-1\}}\mathrm{Tr}(\mathbf{H}_{p-1}(\mathbf{x}_i)\cdot\mathbf{g}_p(\mathbf{x}_i)^\top\cdot\mathbf{g}_p(\mathbf{x}_j)\cdot\mathbf{H}_{p-1}(\mathbf{x}_i)^\top)\Big]\Big]\Big].$$

Applying the above argument repeatedly we obtain that the second summand (equation (22)) is 0.

Thus, we obtain the inequality in equation (20) which implies that $\mathbb{E}_{\mathbf{W}}[h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)]\geq -4$.

D.5. Proof of Theorem 6.1

In this section, we prove Theorem 6.1. The proof follows similar to those in previous sub-sections; we prove a bound on the gradient of the gradient inner-product and show that the expectation is non-negative. Combining these two with an argument similar to equation 10 we get the theorem.

Note that the dataset is obtained by considering i.i.d. samples from a d-dimensional unit sphere. Thus, the lower-bound on the expectation (*i.e.,* non-negative expectation of the gradient inner-product) follows from equation 9. Thus, it remains to prove an upper-bound on the norm of the gradient of the gradient inner-product term.

Throughout this proof, we will use $g(\mathbf{x})$ as a short-hand to denote $g_{\mathbf{W}}(\mathbf{x})$. Consider the gradient $\nabla_{\mathbf{W}}g(\mathbf{x})$. The the i^{th} component of this can be written as follows.

$$[\nabla_{\mathbf{W}}g(\mathbf{x})]_i=\gamma^2\zeta_{\mathbf{x}}(\mathbf{W})\left(\mathbf{W}_\beta^T\cdot\ldots\mathbf{W}_{i+1}^T\cdot\mathbf{x}^T\cdot\mathbf{W}_1^T\cdot\ldots\mathbf{W}_{i-1}^T\right). \tag{23}$$

Now consider, the gradient inner-product $h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)$. We want to upper-bound the quantity $\|\nabla_{(\mathbf{x}_i,\mathbf{x}_j)}h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)\|$. From symmetry, this can be upper-bounded by $2\|\nabla_{\mathbf{x}_i}h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)\|$. Consider the k^{th} coordinate of $\nabla_{\mathbf{x}_i}h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)$. Using equation 23, the assumption that $\{\mathbf{W}_i\}_{i\in[\beta]}$ are orthogonal matrices and taking the gradient, this can be written as,

$$[\nabla_{\mathbf{x}_i}h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)]_k=\gamma^2\zeta_{\mathbf{x}_i}(\mathbf{W})\mathbf{x}_j+\alpha^2\left(\mathbf{W}_\beta^T\cdot\ldots\mathbf{W}_{i+1}^T\cdot\mathbf{x}^T\cdot\mathbf{W}_1^T\cdot\ldots\mathbf{W}_{i-1}^T\right)\left(\nabla_{\mathbf{x}_i}\zeta_{\mathbf{x}_i}(\mathbf{W})\right). \tag{24}$$

Combining assumption 1 with equation 24 we have that $\|\nabla_{\mathbf{x}_i}h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)\|$ is at most $2\gamma^2\beta\|\mathbf{x}_j\|\leq 2\gamma^2\beta$. For the definition of the scaling factor $\gamma=\frac{1}{\sqrt{2\beta}}$, we have that $2\gamma^2\beta=1$. Thus, $\|\nabla_{(\mathbf{x}_i,\mathbf{x}_j)}h_{\mathbf{W}}(\mathbf{x}_i,\mathbf{x}_j)\|\leq 2$.

E. Technical lemmas

We will briefly describe some technical lemmas we require in our analysis. The following Chernoff-style concentration bound is proved in Chapter 5 of Vershynin (2018).

Lemma E.1 (Concentration of Lipshitz function over a sphere)**.** *Let $\mathbf{x} \in \mathbb{R}^d$ be sampled uniformly from the surface of a d-dimensional sphere. Consider a Lipshitz function $\ell : \mathbb{R}^d \to \mathbb{R}$ which is differentiable everywhere. Let $\|\nabla \ell\|_2$ denote* $\sup_{\mathbf{x} \in \mathbb{R}^d} \|\nabla \ell(\mathbf{x})\|_2$*. Then for any $t \geq 0$ and some fixed constant $c \geq 0$, we have the following.*

$$\Pr\left[\left|\ell(\mathbf{x}) - \mathbb{E}[\ell(\mathbf{x})]\right| \geq t\right] \leq 2\exp\left(-\frac{cdt^2}{\rho^2}\right), \tag{25}$$

where $\rho \geq \|\nabla \ell\|_2$.

We will rely on the following generalization of lemma E.1. We would like to point out that the underlying metric is the Euclidean metric and thus we use the $\|.\|_2$-norm.

Corollary E.1. *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ be two mutually independent vectors sampled uniformly from the surface of a d-dimensional sphere. Consider a Lipshitz function $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ which is differentiable everywhere. Let $\|\nabla \ell\|_2$ denote* $\sup_{(\mathbf{x},\mathbf{y}) \in \mathbb{R}^d \times \mathbb{R}^d} \|\nabla \ell(\mathbf{x}, \mathbf{y})\|_2$*. Then for any $t \geq 0$ and some fixed constant $c \geq 0$, we have the following.*

$$\Pr\left[\left|\ell(\mathbf{x}, \mathbf{y}) - \mathbb{E}[\ell(\mathbf{x}, \mathbf{y})]\right| \geq t\right] \leq 2\exp\left(-\frac{cdt^2}{\rho^2}\right), \tag{26}$$

where $\rho \geq \|\nabla \ell\|_2$.

Proof. This corollary can be derived from lemma E.1 as follows. Note that for every fixed $\tilde{\mathbf{y}} \in \mathbb{R}^d$, equation 25 holds. Additionally, we have that the vectors $\mathbf{x}$ and $\mathbf{y}$ are mutually independent. Hence we can write the LHS of equation 26 as the following.

$$\int_{(\tilde{\mathbf{y}})_1 = -\infty}^{(\tilde{\mathbf{y}})_1 = \infty} \cdots \int_{(\tilde{\mathbf{y}})_d = -\infty}^{(\tilde{\mathbf{y}})_d = \infty} \Pr\left[\left|\ell(\mathbf{x}, \mathbf{y}) - \mathbb{E}[\ell(\mathbf{x}, \mathbf{y})]\right| \geq t \;\middle|\; \mathbf{y} = \tilde{\mathbf{y}}\right] \phi(\tilde{\mathbf{y}}) d(\tilde{\mathbf{y}})_1 \ldots d(\tilde{\mathbf{y}})_d.$$

Here $\phi(\tilde{\mathbf{y}})$ refers to the pdf of the distribution of $\mathbf{y}$. From independence, the inner term in the integral evaluates to $\Pr\left[\left|\ell(\mathbf{x}, \tilde{\mathbf{y}}) - \mathbb{E}[\ell(\mathbf{x}, \tilde{\mathbf{y}})]\right| \geq t\right]$. We know this is less than or equal to $2\exp\left(-\frac{cdt^2}{\|\nabla \ell\|_2^2}\right)$. Therefore, the integral can be upper bounded by the following.

$$\int_{(\tilde{\mathbf{y}})_1 = -\infty}^{(\tilde{\mathbf{y}})_1 = \infty} \cdots \int_{(\tilde{\mathbf{y}})_d = -\infty}^{(\tilde{\mathbf{y}})_d = \infty} 2\exp\left(-\frac{cdt^2}{\|\nabla \ell\|_2^2}\right) \phi(\tilde{\mathbf{y}}) d(\tilde{\mathbf{y}})_1 \ldots d(\tilde{\mathbf{y}})_d.$$

Since $\phi(\tilde{\mathbf{y}})$ is a valid pdf, we get the required equation 26. □

Additionally, we will use the following facts about a normalized Gaussian random variable.

Lemma E.2. *For a normalized Gaussian $\mathbf{x}$ (*i.e.*, an $\mathbf{x}$ sampled uniformly from the surface of a unit d-dimensional sphere) the following statements are true.*

1. $\forall p \in [d]$ *we have that* $\mathbb{E}[(\mathbf{x})_p] = 0$.
2. $\forall p \in [d]$ *we have that* $\mathbb{E}[(\mathbf{x})_p^2] = 1/d$.

Proof. Part (1) can be proved by observing that the normalized Gaussian random variable is spherically symmetric about the origin. In other words, for every $p \in [d]$ the vectors $(x_1, x_2, \ldots, x_p, \ldots, x_d)$ and $(x_1, x_2, \ldots, -x_p, \ldots, x_d)$ are identically distributed. Hence $\mathbb{E}[x_p] = \mathbb{E}[-x_p]$ which implies that $\mathbb{E}[x_p] = 0$.

Part (2) can be proved by observing that for any $p, p' \in [d]$, x_p and $x_{p'}$ are identically distributed. Fix any $p \in [d]$. We have that $\sum_{p' \in [d]} \mathbb{E}[x_{p'}^2] = d \times \mathbb{E}[x_p^2]$. Note that we have

$$\sum_{p' \in [d]} \mathbb{E}[x_{p'}^2] = \int_{(\mathbf{x})_1 = -\infty}^{(\mathbf{x})_1 = \infty} \cdots \int_{(\mathbf{x})_d = -\infty}^{(\mathbf{x})_d = \infty} \frac{\sum_{p' \in [d]} x_{p'}^2}{\sum_{p'' \in [d]} x_{p''}^2} \phi(\mathbf{x}) d(\mathbf{x})_1 \ldots d(\mathbf{x})_d = 1.$$

Therefore $\mathbb{E}[x_p^2] = 1/d$. □

We use the following well-known Gaussian concentration inequality in our proofs (*e.g.,* Chapter 5 in Boucheron et al. (2013)).

Lemma E.3 (Gaussian Concentration)**.** *Let* $\mathbf{x} = (x_1, x_2, \dots, x_d)$ *be* i.i.d. $\mathcal{N}(0, \nu^2)$ *random variables. Consider a Lipshitz function* $\ell : \mathbb{R}^d \to \mathbb{R}$ *which is differentiable everywhere. Let* $\|\nabla \ell\|_2$ *denote* $\sup_{\mathbf{x} \in \mathbb{R}^d} \|\nabla \ell(\mathbf{x})\|_2$*. Then for any* $t \geq 0$*, we have the following.*

$$\Pr\left[\left|\ell(\mathbf{x}) - \mathbb{E}[\ell(\mathbf{x})]\right| \geq t\right] \leq 2\exp\left(-\frac{t^2}{2\nu^2\rho^2}\right), \tag{27}$$

where $\rho \geq \|\nabla \ell\|_2$*.*

F. Additional discussion of the small weights assumption (assumption 1)

Without the small-weights assumption, the signal propagated forward or the gradients $\nabla_{\mathbf{W}} f_i$ could potentially blow up in magnitude, making the network untrainable. Proving non-vacuous bounds in case of such blow-ups in magnitude of the signal or the gradient is not possible in general, and thus, we assume this restricted class of weights.

Note that the small-weights assumption is not just a theoretical concern, but also usually holds in practice. Neural networks are often trained with *weight decay* regularizers of the form $\sum_i \|W_i\|_F^2$, which keep the weights small during optimization. The operator norm of convolutional layers have also recently been used as an effective regularizer for image classification tasks by Sedghi et al. (2018).

In the proof of theorem 4.1 we showed that assumption 1 holds with high probability at standard Gaussian initializations used in practice. While, in general, there is no reason to believe that such a small-weights assumption would continue to hold during optimization without explicit regularizers like weight decay, some recent work has shown evidence that the weights do not move too far away during training from the random initialization point for overparameterized neural networks (Neyshabur et al., 2018; Dziugaite & Roy, 2017; Nagarajan & Kolter, 2019; Zou et al., 2018; Allen-Zhu et al., 2018; Du et al., 2018; Oymak & Soltanolkotabi, 2018). It is worth noting though that all these results have been shown under some restrictive assumptions, such as the width requiring to be much larger than generally used by practitioners.