PLATEAU IN MONOTONIC LINEAR INTERPOLATION — A "BIASED" VIEW OF LOSS LANDSCAPE FOR DEEP NETWORKS

Xiang Wang[†], **Annie N. Wang**[§], **Mo Zhou**[†], **Rong Ge**[†]
Department of Computer Science
Duke University, USA
[†]{xwang,mozhou,rongge}@cs.duke.edu,[§]annie.wang029@duke.edu

ABSTRACT

Monotonic linear interpolation (MLI) — on the line connecting a random initialization with the minimizer it converges to, the loss and accuracy are monotonic — is a phenomenon that is commonly observed in the training of neural networks. Such a phenomenon may seem to suggest that optimization of neural networks is easy. In this paper, we show that the MLI property is not necessarily related to the hardness of optimization problems, and empirical observations on MLI for deep neural networks depend heavily on the biases. In particular, we show that interpolating both weights and biases linearly leads to very different influences on the final output, and when different classes have different last-layer biases on a deep network, there will be a long plateau in both the loss and accuracy interpolation (which existing theory of MLI cannot explain). We also show how the last-layer biases for different classes can be different even on a perfectly balanced dataset using a simple model. Empirically we demonstrate that similar intuitions hold on practical networks and realistic datasets.

1 INTRODUCTION

Deep neural networks can often be optimized using simple gradient-based methods, despite the objectives being highly nonconvex. Intuitively, this suggests that the loss landscape must have nice properties that allow efficient optimization. To understand the properties of loss landscape, Goodfellow et al. (2014) studied the linear interpolation between a random initialization and the local minimum found after training. They observed that the loss interpolation curve is monotonic and approximately convex (see the MNIST curve in Figure 1) and concluded that these tasks are easy to optimize. However, other recent empirical observations, such as Frankle (2020) observed that for deep neural networks on more complicated datasets, both the loss and the error curves have a long plateau along the interpolation path, i.e., the loss and error remain high until close to the optimum (see the CIFAR-10 curve in Figure 1). Does the long plateau along the linear interpolation suggest these tasks are harder to optimize? Not necessarily, since the hardness of optimization problems does not need to be related to the shape of interpolation curves (see examples in Appendix A).

In this paper we give the first theory that explains the plateau in both loss and error interpolations. We attribute the plateau to simple reasons as the bias terms, the network initialization scale and the network depth, which may not necessarily be related to the difficulty of optimization.

Note that there are many different theories for the optimization of overparametrized neural networks, in particular the neural tangent kernel (NTK) analysis (Jacot et al., 2018; Du et al., 2018; Allen-Zhu et al., 2019; Arora et al., 2019) and mean-field analysis (Chizat & Bach, 2018; Mei et al., 2018). However they don't explain the plateau in both loss and error interpolations. For NTK regime, the network output is nearly linear in the parameters and the loss interpolation curve is monotonically decreasing and convex — no plateau in the loss interpolation. Mean-field regime often uses a smaller initialization on a homogeneous neural network (as considered in Chizat & Bach (2018); Mei et al. (2018)). In this case, the interpolated network output is basically a scaled version of the network output at the minimum and has same label predictions — no plateau in the error interpolation curve.

Figure 1: Loss interpolation curve and error interpolation curve for a four-layer fully-connected network (FCN4) on MNIST and for VGG16 on CIFAR-10.

1.1 OUR RESULTS

Our theoretical results consist of two parts. In the first part (see Section 3), we give a plausible explanation for the plateau in both the loss and error curves.

Claim 1 (informal). If a deep network has a relatively small initialization, and its last-layer biases are significantly different for different classes, then both the loss and error curves will have a plateau. The length of the plateau is longer for a deeper network.

We formalize this claim in Theorem 1. For intuition, consider a r-layer neural network that only has bias on the last layer, and consider Xavier initialization (Glorot & Bengio, 2010) which typically gives small output and zero bias. If we consider the interpolation point (with coefficient α for the minimum and $(1-\alpha)$ for the initialization), then the weight "signal" from the minimum scales as α^r (as it is the product of r layers) while the bias scales as α. As illustrated in Figure 2 (right), when r is large and there is a difference in biases, the bias will dominate, which creates a plateau in error. For the loss, one can also show that the weight signal is near 0 for small α and the network output is dominated by the biases and the loss cannot beat the random guessing at initialization. Note that this explanation for the plateau does not have any implication on the hardness of optimization.

However, why would the last-layer biases be different for different classes, especially in cases when the biases are initialized as zeros and all classes are balanced? In the second part (see Section 4), we focus on a simple model that we call a homogeneous-weight network. This is a two-layer network whose i-th output is $h(W_{i,:}, x)^r + b_i$, where $x \in \mathbb{R}^d$ is the network input, $W_{i,:} \in \mathbb{R}^d$ is the weight vector and $b_i \in \mathbb{R}$ is the bias (see Figure 2 (left)). Our simple model simulates a deep ReLU/linear network with bias on the output layer, in the sense that the signal is r-homogeneous while the bias is 1-homogeneous in the parameters. Under this model we can show that:

Claim 2 (informal). For the r-homogeneous-weight network on a simple balanced dataset, the class that is learned last has the largest bias.

Here, a class is learned when all the samples in this class get classified correctly with good confidence. We basically show that once a class gets learned, the bias associated with this class starts decreasing and eventually the class that is learned last has the largest bias. We formalize this claim in Theorem 2.

In Section 5, we verify these ideas empirically on fully-connected networks for MNIST (Deng, 2012), Fashion-MNIST (Xiao et al., 2017) and on VGG-16 (Simonyan & Zisserman, 2014) for CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009). We first show that if we train a neural network without using any bias, then the error curve has much shorter plateau or no plateau at all. Even for networks that are trained normally with biases, we design a homogeneous interpolation scheme for biases to make sure that both biases and weights are homogeneous. Such an interpolation indeed significantly shortens the plateau for the error. We also show that decreasing the initialization scale or increasing the network depth can produce a longer plateau in both the error and loss curves. Finally, we show that the bias is correlated with the ordering in which the classes are being learned for small datasets, which suggests that even though the model we consider in the convergence analysis is simple, it captures some of the behavior in practice.

Figure 2: (Left) Our r-homogeneous-weight model with $h_i(x) = \langle W_{i,:}, x \rangle^r + b_i$. (Right) The comparison between interpolated bias $(b_k^{(T)} - b_l^{(T)})$ and interpolated weights signal $\langle W_{i,:}^{[\lambda]}, x \rangle^r$.

1.2 Related works

There are two major lines of work studying interpolation between different points for neural networks, one on monotonic linear interpolation that interpolates the initial network and the learned network, and the other on mode connectivity that connects two learned networks.

Monotonic linear interpolation. Goodfellow et al. (2014) first studied the linear interpolation between the network at initialization and the network after training on MNIST. Frankle (2020) extended the experiments to modern networks on CIFAR-10 and ImageNet and found that though the loss/error is still monotonically non-increasing along the path, it remains high until close to the optimum. Lucas et al. (2021) showed that MLI holds when the network output curve along the interpolation path is close to linear (measured by Gaussian length). However, the Gaussian length can only be formally controlled in the NTK regime.

Mode connectivity. Mode connectivity considers the interpolation between two learned networks (modes) found by SGD. In general, a linear interpolation between two different local minima crosses regions of high loss (Goodfellow et al., 2014). Surprisingly, Draxler et al. (2018) and Garipov et al. (2018) observed that local minima found by SGD from different initializations can be connected via a piece-wise linear path of low loss. Frankle et al. (2020) and Fort et al. (2020) observed that local minima trained from the same initialization can also be connected using a linear path. Freeman & Bruna (2016); Venturi et al. (2018); Nguyen (2019; 2021); Kuditipudi et al. (2019); Shevchenko & Mondelli (2020); Nguyen et al. (2021) gave several theoretical explanations for this phenomenon.

2 Preliminaries

We first formally define the linear interpolation between the network at initialization and the network after training. Then we describe the notations that we will use in the paper.

Linear interpolation: Consider a network with parameters $\theta \in \mathbb{R}^p$. Suppose the network is initialized with parameters $\theta^{(0)}$ and it converges to $\theta^{(T)}$. A linear interpolation is constructed by setting the parameters $\theta^{[\lambda]} = (1-\lambda)\theta^{(0)} + \lambda\theta^{(T)}$ for $\lambda \in [0,1]$. The loss interpolation curve is defined as $\mathcal{L}_{\text{loss}}(\cdot) : [0,1] \to \mathbb{R}$ such that $\mathcal{L}_{\text{loss}}(\lambda)$ is the training loss of the network at $\theta^{[\lambda]}$. Similarly, the error interpolation curve is defined as $\mathcal{L}_{\text{error}}(\cdot) : [0,1] \to [0,1]$ with $\mathcal{L}_{\text{error}}(\lambda)$ as the training error of the network at $\theta^{[\lambda]}$. Here, the training error is simply the ratio of training samples that get classified incorrectly by the network.

Notations: We use $[k]$ to denote the set $\{1, 2, \ldots, k\}$. We use $\mathcal{N}(0, \sigma^2)$ to denote the Gaussian distribution with mean zero and variance σ^2. We use $\|\cdot\|$ to denote the ℓ_2 norm for a vector or the spectral norm for a matrix. For any non-zero vector v, we use $\bar{v}$ to denote $v/\|v\|$. We use $O(\cdot)$, $\Omega(\cdot)$, $\Theta(\cdot)$ to hide the dependency on constant factors and $\Omega(\cdot)$, $\tilde{\Theta}(\cdot)$ to hide the dependency on poly-logarithmic factors.

For any time t, we use $\theta^{(t)}, f^{(t)}$ to denote the parameters and the network at time t. For any $\lambda \in [0,1]$, we use $\theta^{[\lambda]}, f^{[\lambda]}$ to denote the λ interpolation point, which means $\theta^{[\lambda]} := (1-\lambda)\theta^{[0]} + \lambda\theta^{[T]}$ and $f^{[\lambda]}$ is the network with parameters $\theta^{[\lambda]}$.

3 PLATEAU FOR LOSS AND ERROR INTERPOLATIONS

We prove that the long plateau exists in the loss and error curves when the initialization is small and the network is deep on fully-connected networks. The detailed proof can be found in Appendix B.3.

We consider an r-layer fully-connected neural network with at least three. Given input $x \in \mathbb{R}^{n_0}$, the network output is

$$g(x) := V_r\,\sigma(V_{r-1}\,\sigma(\cdots\sigma(V_1 x)\cdots) + b, \tag{1}$$

where $V_i \in \mathbb{R}^{n_r \times n_{r-1}}$ for each layer $i \in [r]$ and $b \in \mathbb{R}^{n_r}$. Here the activation function $\sigma(\cdot)$ can be either identity function or ReLU function. The output layer width equals to the number of classes, i.e., $n_r = k$. We use $L(\{V_i\}, b)$ to denote the sum of cross entropy loss over all samples.

For the biases, we initialize them as zeros and assume after training there exists a gap between the largest bias and the second largest, which also holds empirically (see Figure 8). Note this bias gap is essential for the plateau in the error interpolation. If all the biases are equal in the trained network, the logits for different classes only differ by the weights signal and the interpolated network has same label predictions as the trained network.

Assumption 1 (Bias Gap). Choosing $i \in \arg\max_{i \in [k]} b_i^{(T)}$, we have $b_i^{(T)} - \max_{i \neq i} b_i^{(T)} > 0$. Without loss of generality, we assume that $b_k^{(T)} > \max_{i \in [k-1]} b_i^{(T)}$. We denote $\Delta_{\min} := b_k^{(T)} - \max_{i \in [k-1]} b_i^{(T)}$ and $\Delta_{\max} := b_k^{(T)} - \min_{i \in [k-1]} b_i^{(T)}$.

Then, we show both the loss and error interpolation curves have a long plateau in Theorem 1.

Theorem 1. Suppose the network is defined as in Equation (1) and suppose the weights satisfy $\|V_i^{(0)}\| \le \omega$, $\|V_i^{(T)}\| \le V_{\max}$ for all layers $i \in [r]$. On a k-class balanced dataset whose inputs have ℓ_2 norm at most 1, if Assumption 1 holds, for any $\omega > 0$, as long as $\omega < \min\{\frac{1}{r}, \frac{1}{r^2}, \frac{1}{2e}^{\frac{2}{r-2}}\}$, there exist $\alpha_1 = \frac{\omega}{\Delta_{\min}}$, $\alpha_2 = \frac{1}{1+\rho} = \frac{\Delta_{\min}}{2V_{\max}^r}^{\frac{1}{r-1}}$ and $\alpha_3 = \frac{1}{V_{\max}}^{1/r}$ such that

1. for all $\alpha \in [\alpha_1, \alpha_2]$, the error is $1 - 1/k$;
2. for all $\alpha \in [0, \alpha_3]$, we have $\log k - 2e^{-\frac{1}{N} L\left(\{V_i^{[\alpha]}\}, b^{[\alpha]}\right)} \le \log k + \Delta_{\max} + 2e^{-\Delta_{\max}}$, where N is the number of training examples.

The above theorem shows that for all $\alpha \in [\alpha_1, \alpha_2]$, the error remains at $1 - 1/k$ that is the same as random guessing. We skip the very short initial region $[0, \frac{\omega}{\Delta_{\min}}]$ since the bias is very small and the error can be unpredictable due to the randomness in initial weights. When initialization is small, this error plateau region is roughly $[0, (\frac{\Delta_{\min}}{2V_{\max}^r})^{\frac{1}{r-1}}]$. Empirically, $\frac{\Delta_{\min}}{2V_{\max}^r}$ is smaller than 1 and does not change much when depth increases. So the plateau becomes longer in a deeper network.

Intuitively, the plateau in error curve is there because for a small initialization, the output is close to $\alpha^r V_r^{(T)} V_{r-1}^{(T)} \cdots (V_1^{(T)} x) + b^{(T)}$. When α is not large enough, α^r is much smaller than α, so for every class $i \neq k$, the first term (signal part) cannot overcome the bias gap $(b_k^{(T)} - b_i^{(T)})$. This implies that all samples are predicted as class k and the error is $1 - 1/k$.

We also show that the average loss cannot be lower than $\log k - 2e^{-\Delta_{\max}}$ when $\alpha \le \frac{1}{V_{\max}}^{1/r}$. Note a small random initialization can achieve a loss of approximately $\log k$. Usually the bias gap $\Delta_{\max}$ in practice is not very large, so the loss curve remains nearly flat during this interpolation region. Again, the loss plateau is becoming longer when depth increases. This is because the weights signal remains near 0 for a larger range of α.

4 TRAINING DYNAMICS FOR CREATING A BIAS GAP

In this section, we explain how the gradient flow dynamics generates a bias gap on a balanced dataset by analyzing a simple model. Below, we first define the network model, training dataset and optimization procedure for our analysis.

r-homogeneous-weight network: We consider a two-layer and k-output neural network with activation function $\phi(z) := z^r$, where r is a positive constant that is at least three. As illustrated in Figure 2 (left), under input $x \in \mathbb{R}^d$, the i-th output $f_i(x)$ is $\langle W_{i,:}, x \rangle^r + b_i$, where the weight vector $W_{i,:} \in \mathbb{R}^d$ is the i-th row of weight matrix $W \in \mathbb{R}^{k \times d}$ and $b_i \in \mathbb{R}$ is the i-th entry of vector $b \in \mathbb{R}^k$. In output $f_i(x)$, we call $\langle W_{i,:}, x \rangle^r$ the signal since it is input-dependent and call b_i the bias.

Dataset: We consider a k-class balanced dataset, where k is a constant. We denote the whole dataset as S and denote the subset for each class $i \in [k]$ as S_i. Each subset S_i has exactly N/k samples and each sample $x \in \mathbb{R}^d$ is independently sampled as $x = v_i + \xi$, where the noise $\xi \sim \mathcal{N}(0, \frac{\sigma^2}{d} I)$. To differentiate the noise terms among different samples, we denote the noise associated with x sample as ξ_x. We assume all v_i's are orthonormal; without loss of generality, we assume $v_i = e_i$ for each class i. Here, we assume the orthogonal features to facilitate the convergence analysis beyond the NTK regime, following previous works (Allen-Zhu & Li, 2020; Ge et al., 2021).

Optimization: We initialize each entry in weight matrix W by independently sampling from Gaussian distribution $\mathcal{N}(0, \sigma^2)$ and then taking the absolute value.[1] Our analysis can be trivially generalized to standard Gaussian initialization (without taking absolute value) when r is an even integer. We initialize all bias terms as zeros. We use cross-entropy loss $L(W, b) = \sum_{i \in [k]} \sum_{x \in S_i} \log \frac{\exp(f_i(x))}{\sum_{j \in [k]} \exp(f_j(x))}$, and run gradient flow on $\frac{k}{N} L(W, b)$ for time T. Our analysis can also be extended to gradient descent with a small step size.

Next we show that running gradient flow from a small initialization can converge to a model with zero error and constant bias gap.

Theorem 2. Suppose the neural network, dataset and optimization procedure are as defined in Section 4. Suppose initialization scale $\sigma = o(1)$, noise level $\sigma_\xi = o(1)$, dimension $d = \Omega(1/\sigma^{2r-2})$ and number of samples $N = \Omega(1/\sigma^{r-1})$, with probability at least 0.99 in the initialization, there exists time $T = \Theta(\log(1/\sigma)/\sigma^{r-2})$ such that we have

1. zero error: for all different $i, j \in [k]$ and for all $x \in S_i$, $f_i^{(T)}(x) - f_j^{(T)}(x) \geq \Omega(1)$;
2. bias gap: $b_i^{(T)} - \max_{i \neq i^*} b_i^{(T)} \geq \Omega(1)$ with $i^* = \arg\max_{i \in [k]} b_i^{(T)}$.

Due to space limit, we only give a proof sketch here and leave the detailed proof in Appendix C. Since our dataset is perfectly balanced, it might seem surprising that gradient flow learns diverse biases. We can compute the time derivative on the bias as $\frac{db_i}{dt} = 1 - \frac{k}{N} \sum_{x \in S} u_i(x)$, where $u_i(x)$ is the softmax output for class i, that is $\frac{\exp(f_i(x))}{\sum_{i' \in [k]} \exp(f_{i'}(x))}$. At the beginning, all logits are small, we have $u_i(x) \approx 1/k$ and $\frac{db_i}{dt} \approx 0$. If all the samples are learned at the same time, we have $u_i(x) \approx 1, u_i(x') \approx 0$ for $x \in S_i, x' \in S \setminus S_i$, which again leads to $\frac{db_i}{dt} \approx 0$.

On the other hand, we can consider what happens if all samples in one class (e.g., class i) are learned before any sample in any other class (e.g., class j) is learned.[2] In this case we have

$$\frac{db_i}{dt} = 1 - \frac{k}{N} \sum_{x \in S_i} u_i(x) - \frac{k}{N} \sum_{x \in S \setminus S_i} u_i(x) \approx 1 - \frac{k}{N} \cdot \frac{N}{k} \cdot 1 - \frac{k}{N} \cdot \frac{N(k-1)}{k} \cdot \frac{1}{k} = -\frac{k-1}{k};$$

$$\frac{db_j}{dt} = 1 - \frac{k}{N} \sum_{x \in S_i} u_j(x) - \frac{k}{N} \sum_{x \in S \setminus S_i} u_j(x) \approx 1 - \frac{k}{N} \cdot \frac{N}{k} \cdot 0 - \frac{k}{N} \cdot \frac{N(k-1)}{k} \cdot \frac{1}{k} = \frac{1}{k};$$

where for any learned sample $x \in S_i$, we have $u_i(x) \approx 1, u_j(x) \approx 0$; for any not yet learned sample $x \in S \setminus S_i$, we have $u_i(x), u_j(x) \approx 1/k$. The above calculation shows that b_i starts to decrease and all the other bias terms increase. Generalizing this intuition, we show that b_{i^*} starts to decrease whenever class i^* is learned, and the class that is learned last will have the largest bias.

[1] In the Xavier initialization, each entry in weight matrix W is sampled from $\mathcal{N}(0, 1/d)$, so we can think of $\sigma^2 = 1/d$ that is small when input dimension d is large.

[2] This is indeed possible since all samples of one class only differ in the noise terms in our setting. In the analysis, we can show that the noise term has negligible contribution to the network output and all samples in one class are learned almost at the same time.

Figure 3: The training dynamics of W and b in a four-class example.

As the weights are initialized randomly, by standard anti-concentration, one can argue that there is a gap between $W_{i,i}^{(0)}$'s. Without loss of generality, we assume $W_{1,1}^{(0)} > W_{2,2}^{(0)} > \cdots > W_{k,k}^{(0)}$. The initial difference in the weights will lead to different classes being learned at different time. We show that by doing induction on the following hypothesis through training:

Proposition 1 (Induction Hypothesis) In the same setting of Theorem 2, with probability at least 0.99 in initialization, there exist time points $0 = : s_1 < t_1 < s_2 < t_2 < \cdots < s_{k-1} < t_{k-1} < s_k := T$ with $t_i - s_i = \Theta(\log(1/\eta)/\eta^{r-2})$ and $s_{i+1} - t_i = \Theta(1)$ for $i \in [k-1]$ such that for any $t \in [s_i, s_{i+1}]$,

1. (classes not yet learned) for any class j, $j' \geq i+1$, we have (1) $b_j^{(t)} \leq \max_{i' \in [k]} b_{i'}^{(t)} - \Theta(\eta^r)$, (2) $b_j^{(t)} \geq b_{j'}^{(t)} - \Theta(\eta^r)$ and (3) $W_{j,j}^{(t)} \leq \Theta(\eta)$;
2. (classes already learned) for any class $j \leq i-1$, we have (1) $b_j^{(t)} \geq \max_{i' \in [k]} b_{i'}^{(t)} - \Theta(1)$, (2) $f_j^{(t)}(x) \geq f_{i'}^{(t)}(x) + \Theta(1)$ for $i' \neq j$, $x \in S_j$ and (3) $W_{j,j}^{(t)} \geq \Theta(1)$;
3. (parameters movement) (1) for any $j \in [k]$, $W_{j,j}^{(0)} \leq W_{j,j}^{(t)}$; (2) for any distinct $j, j' \in [k]$, $0 \leq W_{j,j'}^{(t)} \leq \Theta(\eta)$ and (3) for any $j, j' \in [k]$ and any $x \in S_{j'}$, $\langle W_{j,:}^{(t)}, x \rangle \geq \min\{\Theta(\eta), W_{j,j'}^{(t)}\}$.

This proposition shows that gradient flow learns classes one by one, from class 1 to class k. More precisely, each class i is learned during time $[s_i, s_{i+1}]$: All the not yet learned classes $j \geq i+1$ have close to maximum biases and their weights $W_{j,j}^{(t)}$'s are small. All the already learned classes $j \leq i-1$ have small biases and large weights $W_{j,j}^{(t)}$'s. For the parameters movement, we know that all the diagonal entries $W_{j,j}^{(t)}$'s are larger than the initialization and all the off-diagonal entries $W_{j,j'}^{(t)}$'s are only $\Theta(\eta)$. The correlation between the weights and noise terms also remains small.

When learning class i during time $[s_i, s_{i+1}]$, the weight $W_{i,i}^{(t)}$ slowly grows to a small constant in $[s_i, t_i]$ and then quickly grows large in $[t_i, s_{i+1}]$. As a result, all $k \in S_i$ become classified correctly. During the same time, $b_i^{(t)}$ decreases and becomes smaller than the largest bias by at least a constant. At the end time $T = s_k$, although $W_{k,k}^{(T)}$ remains small, all $k \in S_k$ are also classified correctly because $b_k^{(T)}$ is the largest bias. See an illustration of this learning process in Figure 3.

Although we consider a simple neural network and data distribution, the analysis for the training dynamics is still non-trivial. There are three major challenges in our proof: (1) How to ensure that class $i+1$ is learned much later than class i? (2) For any class j that has not been learned, how to maintain that its bias is close to the maximum? (3) For any learned j, how to maintain the large bias gap from the top bias? Next, we give the proof ideas for these questions. Since all the off-diagonal entries and correlations with noise terms $W_{i,j}^{(t)}$ are negligible, in our proof we can essentially focus on the movement of $W_{i,i}^{(t)}$'s and $b_i^{(t)}$'s.

Lower bounding $s_{i+1} - s_i$. During time $[s_i, t_i]$, the dynamics of $W_{i,i}^{(t)}$ is similar as in the tensor power method (Allen-Zhu & Li, 2020; Ge et al., 2021). The initial gap between $W_{i,i}^{(0)}$ and $W_{j,j}^{(0)}$

ensures that when $W_{i,i}^{(t_i)}$ rises to a small constant, $W_{j,j}^{(t_i)}$ is still $O(\ )$ for all $j \quad i+1$. Then after constant times $s_{i+1} \quad t_i$, $W_{j,j}^{s_{i+1}}$ is still $O(\ )$ since the increasing rate of $W_{j,j}^{(t)}$ is merely $O(\ ^{r-1})$.

Bias for classes that are not learned. For $j \quad i+1$, we maintain that $b_j^{(t)} \quad \max_{i' \in [k]} b_{i'}^{(t)} \quad O(\ ^r)$. First, we use the below lemma to show biases for any two classes j, j' less than $i+1$ remain close.

Lemma 1 (Coupling Biases). Assuming $W_{j',j'}, W_{j,j} \quad O(\ )$ and $b_{j'}, b_j \quad \max_{i' \in [k]} b_{i'} \quad O(\ ^r)$, we have $b_{j'} \quad b_j > 0$ if $b_{j'} \quad b_j \quad ^r$, and $b_{j'} \quad b_j < 0$ if $b_{j'} \quad b_j \quad + \ ^r$ for some positive constant $$.

Second we show that any already learned or being learned class j' cannot have bias much larger than any class $j \quad i+1$ not yet learned.

Lemma 2 (Bias Gap Control I). For any different $j', j \in [k]$, if $W_{j',j'} \quad W_{j,j}$, $W_{j,j} \quad O(\ )$ and $b_{j'} \quad b_j \quad O(\ ^r)$, $b_j \quad \max_{i' \in [k]} b_{i'} \quad O(\ ^r)$, we have $b_{j'} \quad b_j < 0$.

Bias for learned classes. At time s_{j+1}, we can prove that $1 \quad u_j^{(s_{j+1})}(x) \quad C_1$ for all $x \in S_j$ and $b_j^{(s_{j+1})} \quad b_k^{(s_{j+1})} \quad C_2$. According to the below lemma, we can ensure that $b_j^{(t)} \quad b_k^{(t)} \quad C_2$ for any $t \quad s_{j+1}$.

Lemma 3 (Bias Gap Control II). There exist small positive constants C_1, C_2 such that for any $j \in [k \quad 1]$ and any $x \in S_j$, if $1 \quad u_j(x) \quad C_1$, $W_{k,k} \quad O(\ )$ and $b_j \quad b_k \quad C_2$, we have $b_j \quad b_k < \ (1)$.

4.1 Plateau and monotonicity for r-homogeneous weight network

Now assuming the network at initialization and after training satisfies the properties described in Theorem 2 and Proposition 1, we can prove a tighter bound on the plateau region and also show the monotonicity in error and loss curve. See the complete proofs in Appendix B.1 and Appendix B.2.

Same as in Assumption 1, we use $_i$ to denote the bias gap $b_k^{(T)} \quad b_i^{(T)}$ for $i \in [k \quad 1]$ and denote $_{\min} := \min_{i \in [k-1]} {}_i$ and $_{\max} = \max_{i \in [k-1]} {}_i$. For the weights, we denote $W_{\min} = \min_{i \in [k-1]} W_{i,i}^{(T)}$ and $W_{\max} = \max_{i \in [k]} W_{i,i}^{(T)}$. We denote $R_{\min} = \min_{i \in [k-1]} {}_i / [W_{i,i}^{(T)}]^r$, $R_{\max} = \max_{i \in [k-1]} {}_i / [W_{i,i}^{(T)}]^r$. Below, we show the plateau and monotonicity of loss and error interpolations in Theorem 3.

Theorem 3. Suppose the neural network, dataset and optimization procedure are as defined in Section 4. Suppose the network at initialization and after training satisfies the properties described in Theorem 2 and Proposition 1. For any $\in (0,1)$, suppose $\min((\ ^{1/r}), (R_{\min}^{\frac{1}{r-1}} \ ^{1/r}_{\min}), ((\frac{W_{\min}}{W_{\max}})^{\frac{2r}{r-2}}))$. There exist $_1 = \frac{}{_{\min}}$, $_2 = (\frac{1}{1+O(\ )}p\frac{}{})^{\frac{r}{r-1}} R_{\min}^{\frac{1}{r-1}}$, $_3 = \frac{^{1/r}}{W_{\max}}$ and $_4 = (1+O(\ ))^{\frac{1}{r-1}} \frac{R_{\max}}{r}^{\frac{1}{r-1}}$ such that

1. for all $\in [_1, _2]$, the error is $1 \quad 1/k$; for all $\in [_1, 1]$, the error is non-increasing;
2. for all $\in [0, _3]$, we have $\log k \quad e \quad \frac{1}{N} L(W^{[\ ]}, b^{[\ ]}) \quad \log k + \ _{\max} + e$; for all $\in [_4, 1]$, the loss is strictly monotonically decreasing.

The analysis for the plateau is very similar as in Theorem 1 since a r-homogeneous-weight network is similar to a depth-r fully connected neural network with only last-layer biases in the sense that the weights are r-homogeneous while the (last-layer) bias is 1-homogeneous. For the error plateau, we prove a tighter bound on the right boundary $_2$ than in Theorem 1. We also show the error is non-increasing for $\in [= _{\min}, 1]$ by arguing that once a sample is correctly classified at interpolated point $^0 = _{\min}$, it will remain so for any ^{0}. Similar as in Theorem 1, we can show that the loss is no smaller than $\log k \quad e$ when $^{1/r}$. To show the monotonicity of loss after $_4$, we show that $f_i^{[\ ]}(x) \quad f_j^{[\ ]}(x)$ is increasing in for $i \quad j$ and $x \in S_i$.

In summary, for $\in [_1, _2]$, the signal is smaller than the bias gap and the error remains at $1 \quad 1/k$. Before $_3$, the signal is very small and the loss remains large; after $_4$, the signal starts to overcome the bias gap and the loss is decreasing. See an illustration in Figure 2 (right).

5 EXPERIMENTS

In this section we empirically show that intuitions from our simple theoretical model can also be applied to more realistic datasets and architectures. First, we show that bias plays an important role in creating the plateau in the error interpolation, as predicted by Theorem 1. We then demonstrate the in uence of initialization size and network depth (also see Theorem 1). Finally, we show that similar to Proposition 1 the class that is learned last often has larger bias. Due to space constraint, we only show the results on MNIST and CIFAR-100 in this section, while similar results also hold on Fashion-MNIST and CIFAR-10 (see Appendix D).

Unless speci ed otherwise, we use a depth-10 and width-1024 fully-connected ReLU neural network (FCN10) for MNIST and use VGG-16 (without batch normalization) for CIFAR-100. We use Kaiming initialization (He et al., 2015) for the weights and set all bias terms as zeros. For FCN10 on MNIST, we use a small initialization by scaling the weights of each layer by $(0.01)^{1=10}$ so the output is scaled by 0.001. We train each network using SGD for 100 epochs. See more experiment settings in Appendix D.

We linearly interpolate using 50 evenly spaced points between the network at initialization and the network at the end of training. We evaluate error and loss on the train set. For each setting, we repeat the experiments three times from different random seeds and plot the mean and deviation.

Figure 4: Loss and error curves across networks with all bias, last bias and no bias.

Role of bias in creating plateau. We demonstrate the importance of bias using two experiments. In the rst experiment, we compare the loss/error interpolation curves between networks equipped with bias for all the layers (all bias), with bias only for the output layer (last bias), and with no bias at all (no bias). Figure 4 shows that networks with all bias and last bias have a much longer error plateau than networks without bias. Three bias settings have similar loss interpolation curves.

Figure 5: Loss and error curves across networks with normal and homogeneous interpolation on bias.

By our theory, the bias dominates the signal at the beginning of the interpolation because the bias term scales as while the signal scales as . In the second experiment, to correct this discrepancy, we interpolate the bias at the h-th layer (input is at the 0-th layer) as $b_h^{[\ ]} = (1\ \ )^h b_h^{(0)} + \ ^h b_h^{(T)} = \ ^h b_h^{(T)}$. We call this the homogeneous interpolation as now terms involving bias and weights all have ^{r} coef cients. We compare this with the normal interpolation that linearly interpolates the bias terms. Figure 5 shows that for networks with all bias or last bias, using homogeneous interpolation can signi cantly reduce the plateau in the error interpolation, but does not affect the loss interpolation.

Role of initialization scale and network depth. Our theory suggests that with a smaller initialization, the signal magnitude at the initial interpolation is smaller, which can create longer plateau in both loss interpolation and error interpolation. We compare networks under initialization scales $1; 0.1; 0.01$ and 0.001, where scale 1 corresponds to the standard Kaiming initialization. For other

Figure 6: Loss and error curves across networks with different initialization scales.

initialization ; we rescale each layer by the same factor so the output is rescaled by According to Figure 6, smaller initialization does create longer plateau in loss and error interpolation.

Figure 7: Loss and error curves across networks with different depths.

With a deeper network, the signal grows slower at the initial interpolation phase, which can potentially create a longer plateau in both loss interpolation and error interpolation. We compare FCN4, FCN6, FCN8, FCN10 on MNIST and compare VGG11, VGG13, VGG16, VGG19 on CIFAR-100. According to Figure 7, deeper networks do have longer plateau in loss and error interpolation.

Figure 8: Train loss for each class and bias term dynamics on 2-class MNIST and 3-class MNIST.

Bias learning dynamics. Our dynamics analysis in Section 4 shows that gradient descent can learn diverse biases on a balanced dataset by learning different classes at different time points. In particular, the last learned class should have the highest bias term. We verify this theory by studying FCN10 with only output bias on balanced 2-class or 3-class MNIST. To separate the learning of different classes, we compute the per-class loss by only considering the examples in that particular class. According to Figure 8, in the 2-class MNIST, number 1 is learned last and its bias is larger, which ts our theory. Also in the 3-class MNIST, class 2 is learned rst, class 3 the second and class 1 the last; for the learned bias, class 2 bias is smallest, class 3 bias in the middle and class 1 bias the highest.

6 CONCLUSION

Our theory suggests that the plateau in loss/error interpolation curves may be attributed to simple reasons, and it's unclear if these reasons are related to the dif culty/easiness of optimization. In our experiments although the training succeeds in all the settings, the loss and error interpolation curves can be easily manipulated by changing the initialization size, network depth and bias terms. Therefore, we believe one needs to look at structures more complicated than linear interpolation to understand why optimization succeeds for deep neural networks.

Though our theory requires a small initialization, we also observe plateau in CIFAR-100 with standard initialization, which suggests that the useful signal is still a high order term. We also observe that sometimes the ordering of the biases does not exactly follow the ordering of the learning. We believe this is partially due to the correlation between different-class features and offer a preliminary explanation in Appendix D.5. We leave the thorough study of these problems in the future work.

ACKNOWLEDGEMENT

This work is supported by NSF Award DMS-2031849, CCF-1845171 (CAREER), CCF-1934964 (Tripods) and a Sloan Research Fellowship.

REPRODUCIBILITY STATEMENT

For our theoretical results, we listed all the assumptions and stated the theorems in the main text and we left the complete proof for all the claims in the Appendix. For our experimental results, we de ned the detailed experiment settings in the Appendix and also uploaded the source code as supplementary material.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019.

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

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pp. 1309–1318. PMLR, 2018.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33:5850–5861, 2020.

Jonathan Frankle. Revisiting" qualitatively characterizing neural network optimization problems". *arXiv preprint arXiv:2012.06898*, 2020.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

C Daniel Freeman and Joan Bruna. Topology and geometry of half-recti ed network optimization. *arXiv preprint arXiv:1611.01540*, 2016.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

Rong Ge, Yunwei Ren, Xiang Wang, and Mo Zhou. Understanding de ation process in over-parametrized tensor decomposition. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014.

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

Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):1–39, 2013.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Rohith Kuditipudi, Xiang Wang, Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Rong Ge, and Sanjeev Arora. Explaining landscape connectivity of low-cost solutions for multilayer nets. *Advances in Neural Information Processing Systems*, 32, 2019.

James Lucas, Juhan Bae, Michael R Zhang, Stanislav Fort, Richard Zemel, and Roger Grosse. Analyzing monotonic linear interpolation in neural network loss landscapes. *arXiv preprint arXiv:2104.11044*, 2021.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

Quynh Nguyen. On connected sublevel sets in deep learning. In *International Conference on Machine Learning* pp. 4790–4799. PMLR, 2019.

Quynh Nguyen. A note on connectivity of sublevel sets in deep learning. *arXiv preprint arXiv:2101.08576*, 2021.

Quynh N Nguyen, Pierre Bréchet, and Marco Mondelli. When are solutions connected in deep networks? *Advances in Neural Information Processing Systems*, 34, 2021.

Alexander Shevchenko and Marco Mondelli. Landscape connectivity and dropout stability of sgd solutions for over-parameterized neural networks. In *International Conference on Machine Learning* pp. 8773–8784. PMLR, 2020.

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

Luca Venturi, Afonso S Bandeira, and Joan Bruna. Spurious valleys in two-layer neural network optimization landscapes. *arXiv preprint arXiv:1802.06384*, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

A EXAMPLES FOR THE DISCONNECTION BETWEEN LINEAR INTERPOLATION SHAPE AND OPTIMIZATION DIFFICULTY

We give two examples that illustrate the disconnection between the linear interpolation shape and the optimization dif culty. In Section A.1, we show a function that is NP-hard to optimize, but has a convex and monotonically decreasing loss interpolation. Then in Section A.2, we give a function that is easy to optimize, but has a non-monotonic loss interpolation.

A.1 HARD FUNCTION WITH CONVEX LOSS INTERPOLATION

For any symmetric third-order tensor $\mathcal{T} \in \mathbb{R}^{d \times d \times d}$, our goal is to minimize

$$f(x, z) = \mathcal{T}(x, x, x) + \|x\|^4 + z^4 \tag{2}$$

where $x \in \mathbb{R}^d$ and $z \in \mathbb{R}$.

It's known that nding the spectral norm of a symmetric third-order tensor (that is, $\max_{v \in \mathbb{R}^d, \|v\|=1} \mathcal{T}(v, v, v)$) is NP-hard (Hillar & Lim, 2013). We prove that minimizing $f(x, z)$ is also NP-hard by reducing the tensor spectral norm problem to it.

Proposition 2. Minimizing $f(x, z)$ as de ned in Eqn. 2 is NP-hard.

Proof. For any non-zero tensor $\mathcal{T}$, let (x^*, z^*) be one minimizer of $f(x, z)$; it's easy to verify that $\mathcal{T}(x^*, x^*, x^*) > 0$. We show that $u^* := x^*/\|x^*\|$ must be a solution to $\max_{v \in \mathbb{R}^d, \|v\|=1} \mathcal{T}(v, v, v)$.

For the sake of contradiction, assume there exists v^* with unit norm such that $\mathcal{T}(v^*, v^*, v^*) > \mathcal{T}(x^*, x^*, x^*)$. It's easy to verify that $f(\|x^*\| v^*, z^*) < f(x^*, z^*)$, which however contradicts the optimality of (x^*, z^*). □

Next, we prove that start from certain initialization, the loss along the linear interpolation path is convex and monotonically decreasing. Note that assuming the unit Frobenius norm of $\mathcal{T}$ does not hurt the NP-hardness of the problem. And our initialization is oblivious of the tensor $\mathcal{T}$.

Proposition 3. Assume $\|\mathcal{T}\|_F = 1$. Suppose we start from initialization (x_0, z_0) with $x_0 = 0$ and $|z_0| > \frac{\sqrt{3}}{4}$. Let (x^*, z^*) be a minimizer of $f(x, z)$ as de ned in Eqn. 2. We know the loss interpolation curve $\phi(\alpha) := f((1-\alpha)x_0 + \alpha x^*, (1-\alpha)z_0 + \alpha z^*)$ is convex and monotonically decreasing for $\alpha \in [0, 1]$.

Proof. We rst prove that at any minimizer (x^*, z^*), we must have $z^* \geq 0$. Otherwise, we can set z as zero to further decrease the loss. Starting from an initialization $(x^{(0)}, z^{(0)})$ with $x^{(0)} = 0$, we know at each interpolation point $x^{[\alpha]} = \alpha x^*, z^{[\alpha]} = (1-\alpha)z^{(0)}$. Therefore, we have

$$\begin{aligned}
\phi(\alpha) = f(x^{[\alpha]}, z^{[\alpha]}) &= \mathcal{T}(x^{[\alpha]}, x^{[\alpha]}, x^{[\alpha]}) + \|x^{[\alpha]}\|^4 + \left[z^{[\alpha]}\right]^4 \\
&= \mathcal{T}(\alpha x^*, \alpha x^*, \alpha x^*) + \alpha^4 \|x^*\|^4 + \left[(1-\alpha)z^{(0)}\right]^4 \\
&= \alpha^3 \mathcal{T}(x^*, x^*, x^*) + \alpha^4 \|x^*\|^4 + (1-\alpha)^4 \left[z^{(0)}\right]^4.
\end{aligned}$$

To prove the convexity of $\phi(\alpha)$ for $\alpha \in [0, 1]$, we only need to prove $\phi''(\alpha) > 0$ for $\alpha \in [0, 1]$. We have

$$\begin{aligned}
\phi''(\alpha) &= 6\alpha \mathcal{T}(x^*, x^*, x^*) + 12\alpha^2 \|x^*\|^4 + 12(1-\alpha)^2 \left[z^{(0)}\right]^4 \\
&= 6\alpha \|x^*\|^3 \mathcal{T}(x^*, x^*, x^*) + 12\alpha^2 \|x^*\|^4 + 12(1-\alpha)^2 \left[z^{(0)}\right]^4.
\end{aligned}$$

Since the formula for $\phi''(\alpha)$ involves both $\mathcal{T}(x^*, x^*, x^*)$ and $\|x^*\|$, we rst gure out the relation between these two quantities. Suppose $\mathcal{T}(x^*, x^*, x^*) = p > 0$, it's not hard to nd $\|x^*\|$ must be equal to $\frac{3p}{4}$. This is because $\alpha x^3 p + \alpha^2 x^4$ is minimized when $\|x^*\| = \frac{3p}{4}$. Next, we prove $\phi''(\alpha) > 0$ for $\alpha \in (2/3, 1]$ and $\alpha \in [0, 2/3]$ separetely.

When $\in (2/3, 1]$, we have

$$12\ ^2\|x\|^4 > 6\rho\ \|x\|^3 = 6\ \ \|x\|^3 T(x\ ; x\ ; x\ ).$$

Therefore, we know $^{00}(\ ) > 0$.

When $\in [0, 2/3]$, we know

$$12(1\ \ )^2\left[z^{(0)}\right]^4\ \ \frac{4}{3}\left[z^{(0)}\right]^4.$$

Since $\|T\|_F = 1$, we know $T(x\ ; x\ ; x\ )\ \ 1$ and $\|x\| \ \ 3/4$. Therefore, we have

$$6\ \|x\|^3 T(x\ ; x\ ; x\ )\ \ 6\ \frac{2}{3}\ \ \frac{3}{4}\ ^3\ \ 1 = \frac{27}{16}.$$

Then, we know that if $z^{(0)} > \frac{\sqrt{3}\sqrt{2}}{4}$, we have $^{00}(\ ) > 0$. □

A.2 Easy function with non-monotonic loss interpolation

In this section, we give an easy-to-optimize function that however has a non-monotonic loss interpolation curve. We consider the following loss function

$$f(x; y) = \begin{cases} 0 & \text{if } x = y = 0 \\ 1\ \ \frac{1}{3}\frac{y}{\sqrt{x^2+y^2}}\ \ x^2 + y^2\ ^2\ \ 2\sqrt{x^2+y^2} & \text{otherwise,} \end{cases} \tag{3}$$

where $x; y \in \mathbb{R}$. We can also re-parameterize $f(x; y)$ using angle $\in [0; 2\ )$ and length $r \in [0; 1\ )$ as $h(\ ; r\ ) = 1\ \ \frac{\sin(\ )}{3}\ \ r^4\ \ 2r^2$.

Next, we prove that starting from any non-zero point, gradient flow converges to the global minimizer.

Proposition 4. Starting from any non-zero initialization, gradient flow for $f(x; y)$ as defined in Eqn. 3 converges to the global minimizer $(0;\ \ 1)$.

Proof. We know the unique minimizer of $f(x; y)$ is $(0;\ \ 1)$ by considering its equivalent form $h(\ ; r\ )$. For $h(\ ; r\ ) = 1\ \ \frac{\sin(\ )}{3}\ \ r^4\ \ 2r^2$, we know $r^4\ \ 2r^2$ is minimized at $r = 1$ and $1\ \ \frac{\sin(\ )}{3}$ is maximized at $= \frac{3}{2}$.

Besides the minimizer $(0;\ \ 1)$, the other stationary point is $(0; 0)$. For any point $(x; y)$ different from $(0;\ \ 1)$ and $(0; 0)$; if $x^2 + y^2 \ \ 1$, the gradient along the radial direction is non-zero; if $\frac{y}{\sqrt{x^2+y^2}} \ \ 1$, the gradient along the tangent direction is non-zero. It's also easy to verify that starting from a non-zero point, gradient flow does not converge to $(0; 0)$, so it must converge to $(0;\ \ 1)$. □

It's also very easy to prove that gradient descent with appropriate step size converges to an ϵ-neighborhood of the global minimizer within $\mathrm{poly}(1/\epsilon)$ number of iterations. This is because the gradient is at least $\mathrm{poly}(\epsilon)$ for any non-zero point outside of the ϵ-neighborhood of the global minimizer. Starting from an initialization $(x; y)$ with $x^2 + y^2 = (1)$, the smoothness along the training is also bounded by a constant.

Next, we prove that starting from certain initialization,[3] the loss interpolation between the initialization and the global minimizer is non-monotonic. We prove this by identifying two points along the interpolation path such that the point closer to minimizer has a higher loss compared with the point further to the minimizer.

[3] Note the initialization condition in Prop. 5 is satisfied with constant probability for a reasonable initialization scheme. For example, if we uniformly sample $(x; y)$ from the set $S = \{(x; y) \in \mathbb{R}^2 | x^2 + y^2 \ \ R\}$ with $R \ \ 2$, the condition is satisfied with constant probability.

Proposition 5. Suppose we start from an initialization $(x_0; y_0) = (r \sin(\alpha); r\cos(\alpha))$ with $r \geq 1$ and $\alpha \in [\pi/3, 2\pi/3]$. Consider the loss interpolation curve $\ell(\rho) = f((1-\rho)x_0 + \rho x^*; (1-\rho)y_0 + \rho y^*)$ with $(x^*; y^*) = (0; 1)$ and $f(\cdot;\cdot)$ defined in Eqn. 3. We know there exist $0 < \rho_1 < \rho_2 < 1$ such that

$$\ell(\rho_2) - \ell(\rho_1) \geq \frac{5}{32}.$$

Proof. We prove for any $\alpha \in [\pi/3, 2\pi/3]$ and any $r \geq 1$, the loss interpolation between $(r\sin(\alpha); r\cos(\alpha))$ to $(0; 1)$ is non-monotonic. In particular, we show there are two points along the linear interpolation satisfying

$$f(\sin(\alpha/2)\cos(\alpha/2); \sin(\alpha/2)\sin(\alpha/2)) - f(\sin(\alpha); \cos(\alpha)) \geq 1/12,$$

where $(\sin(\alpha/2)\cos(\alpha/2); \sin(\alpha/2)\sin(\alpha/2))$ is the middle point between $(\sin(\alpha); \cos(\alpha))$ and $(0; 1)$.

Next, we separately upper bound $f(\sin(\alpha); \cos(\alpha))$ and lower bound $f(\sin(\alpha/2)\cos(\alpha/2); \sin(\alpha/2)\sin(\alpha/2))$. We have

$$\max_{\alpha \in [\pi/3, 2\pi/3]} f(\sin(\alpha); \cos(\alpha)) - f(0; 1) = \frac{2}{3}$$

and

$$\begin{aligned}
&\min_{\alpha \in [\pi/3, 2\pi/3]} f(\sin(\alpha/2)\cos(\alpha/2); \sin(\alpha/2)\sin(\alpha/2)) \\
&\quad - f(\sin(\alpha/6)\cos(\alpha/6); \sin(\alpha/6)\sin(\alpha/6)) \\
&= 1 + \frac{1}{2}\left(\frac{1}{3}\right)^4 - 2\left(\frac{1}{2}\right)^2 \\
&= \frac{49}{96}.
\end{aligned}$$

Therefore, we have $\ell(\sin(\alpha/2)\cos(\alpha/2); \sin(\alpha/2)\sin(\alpha/2)) - f(\sin(\alpha); \cos(\alpha)) \geq \frac{5}{32}$. □

B Proof for plateau and monotonicity

We first consider the r-homogeneous-weight model. We prove the plateau and monotonicity properties for the error interpolation (Theorem 4) in Section B.1. We then prove the plateau and monotonicity properties for the loss interpolation (Theorem 5) in Section B.2. Theorem 3 is a simple combination of Theorem 4 and Theorem 5. Finally, we give the plateau analysis for the fully-connected neural networks (Theorem 1) in Section B.3.

B.1 Error interpolation for r-homogeneous-weight model

Theorem 4 (Error Interpolation). Suppose the network at initialization and after training satisfy the properties described in Theorem 2 and Induction Hypothesis 1. Suppose $\min(O(1); O(R_{\min}^{\frac{1}{r-1}-1/r}); O((\frac{W_{\min}}{W_{\max}})^{\frac{2r}{r-2}}))$: There exist $\rho_1 = \frac{\cdot}{\lambda_{\min}}$ and $\rho_2 = (\frac{1}{1+O(\frac{\cdot}{\cdot})})^{\frac{r}{r-1}} R_{\min}^{\frac{1}{r-1}}$, such that

1. for all $\rho \in [\rho_1, \rho_2]$, the error is $\geq 1 - 1/k$;
2. for all $\rho \in [\rho_1, 1]$, the error is non-increasing.

Proof of Theorem 4. This theorem directly follows from Lemma 4 and Lemma 5.

Next, we separately prove the initial plateau in Lemma 4 and the monotonicity in Lemma 5.

Lemma 4 (Error Plateau) In the same setting as in Theorem 4, there exists $\rho_1 = \frac{\cdot}{\lambda_{\min}}$ and $\rho_2 = (\frac{1}{1+O(\frac{\cdot}{\cdot})})^{\frac{r}{r-1}} R_{\min}^{\frac{1}{r-1}}$, such that for any interpolation point with $\rho \in [\rho_1, \rho_2]$, the error is $\geq 1 - 1/k$. Moreover, we have $e_i^{[\rho]}(e_j) < f_k^{[\rho]}(e_j)$ for all $j \in [k]$ and all $i \neq k$.

In the proof of Lemma 4, we show that for interpolation point $x \in [\alpha_1, \alpha_2]$, the bias term dominates and all samples are classified as class k that has the largest bias.

Proof of Lemma 4. We only need to show that for all $x \in [\alpha_1, \alpha_2]$, we have

$$f_i^{[\ell]}(x) < f_k^{[\ell]}(x)$$

for all $x \in S$ and all $i \neq k$, which immediately implies the error is $1 - 1/K$. Without loss of generality, assume $x \in S_j$ where j may equal i or k.

For $x \in \left[\alpha_1, \frac{p-}{W_{\min}}\right]$. If $\alpha_1 = \frac{p-}{W_{\min}}$, we only need to consider the case when $x \in \left[\frac{p-}{W_{\min}}, \alpha_2\right]$. So here we assume $\frac{p-}{W_{\min}} < \alpha_2$. We can lower bound $f_k^{[\ell]}(x) - f_i^{[\ell]}(x)$ as

$$\begin{aligned} & f_k^{[\ell]}(x) - f_i^{[\ell]}(x) \\ = & \left\langle W_{k,:}^{[\ell]}, x^{\ell} \right\rangle + b_k^{[\ell]} - \left\langle W_{i,:}^{[\ell]}, x^{\ell} \right\rangle - b_i^{[\ell]} \\ = & \left\langle W_{k,:}^{[\ell]}, x^{\ell} \right\rangle + b_k^{[\ell]} - \left\langle W_{i,j}^{[\ell]} \pm O(\eta), x^{\ell} \right\rangle - b_i^{[\ell]} \\ \geq & \left\langle W_{i,j}^{(0)} + W_{i,j}^{(T)} \pm O(\eta), x^{\ell} \right\rangle, \end{aligned}$$

where the second equality uses $\left\langle W_{i,:}^{[\ell]}, x^{\ell} \right\rangle \pm O(\eta)$ and the inequality uses $\left\langle W_{k,:}^{[\ell]}, x^{\ell} \right\rangle \geq 0$.

To prove $f_k^{[\ell]}(x) - f_i^{[\ell]}(x) > 0$ for $x \in \left[\frac{}{W_{\min}}, \frac{p-}{W_{\min}}\right]$, we only need to prove $\frac{}{W_{\min}} \left\langle W_{i,j}^{(0)} + \frac{p-}{W_{\min}} W_{i,j}^{(T)} \pm O(\eta), x^{\ell} \right\rangle > 0$. Since $x \geq \alpha_1 \geq \frac{}{W_{\min}}$, we know $\frac{}{W_{\min}} x \geq 1$. Due to full accuracy, we know $\left\langle W_{i,:}^{(T)}, x^{\ell} \right\rangle \geq 1 = r$ for $x \in S_i$, which then implies $W_{i,i}^{(T)} \geq 1 = r$ because $x \in S_i$ (1) and $\left\langle W_{i,:}^{(T)}, x^{\ell} \right\rangle \leq O(\eta) + O(1)$. Since $W_{i,i}^{(T)} \geq 1 = r$ and $W_{i,j}^{(T)} \leq O(\eta)$ for $j \neq i$, so we have $W_{i,j}^{(T)} \leq W_{i,i}^{(T)} W_{\max}$ as long as $O(\eta) \leq 1 = r_{\min}$. So we can upper bound $\left\langle W_{i,j}^{(0)} + \frac{p-}{W_{\min}} W_{i,j}^{(T)} \pm O(\eta), x^{\ell} \right\rangle$ as follows,

$$\begin{aligned} & \left\langle W_{i,j}^{(0)} + \frac{p-}{W_{\min}} W_{i,j}^{(T)} \pm O(\eta), x^{\ell} \right\rangle \leq O(\eta) + \left(\frac{p- W_{\max}}{W_{\min}} \right)^{r} \\ & \leq \left(\frac{p- W_{\max}}{r W_{\min}} + \frac{p- W_{\max}}{W_{\min}} \right)^{r} \leq e^{r} \left(\frac{p- W_{\max}}{W_{\min}} \right)^{r}, \end{aligned}$$

where the second inequality assumes $O(\eta) \leq \left(\frac{p- W_{\max}}{W_{\min}} \right)^{r}$. Therefore, to prove $\frac{}{W_{\min}} \left\langle W_{i,j}^{(0)} + \frac{p-}{W_{\min}} W_{i,j}^{(T)} \pm O(\eta), x^{\ell} \right\rangle > 0$ we only need

$$e^{r} \left(\frac{p- W_{\max}}{W_{\min}} \right)^{r} > 0,$$

which holds as long as $< \left(\frac{1}{e} \frac{W_{\min}}{W_{\max}} \right)^{\frac{2}{r-2}}$.

For $x \in \left[\frac{p-}{W_{\min}}, \right]^2$. Similar as above, we only need to show that $\left\langle W_{i,j}^{(0)} + W_{i,j}^{(T)} \pm O(\eta), x^{\ell} \right\rangle > 0$ for $i \neq k$ and $j \in [k]$. Since $W_{i,j}^{(0)} \geq -O(\eta)$ and $p-/W_{\min}$, we

have $W_{i,j}^{(0)} \le O(\eta^{-} W_{\min})$. Therefore, we have $W_{i,j}^{(0)} + W_{i,j}^{(T)} + O(\ ) \le 1 + O(\eta^{-}) \ W_{i,i}^{(T)}$. Therefore, we have

$$\Big[W_{i,j}^{(0)} + W_{i,j}^{(T)} + O(\ ) \Big]^r \le \Big[1 + O(\eta^{-}) \Big]^r \Big[W_{i,i}^{(T)} \Big]^r > 0,$$

where the last inequality assumes $\ _2 := \frac{1}{1+O(\eta^{-})} \left(\frac{r}{r-1} R_{\min}^{\frac{1}{r-1}} \right)$ where $R_{\min} = \min_{i \in [k \ 1]} \ _i / [W_{i,i}^{(T)}]^r$.

Next, we show that the error is non-increasing for $\in [\ _1; 1]$ by proving that once a sample is classified correctly it will remain so.

Lemma 5 (Error Monotonicity) In the same setting as in Theorem 4, there exists $= \frac{\ }{\ _{\min}}$ such that the error is non-increasing for $\in [\ _1; 1]$.

Proof of Lemma 5. We first show that sample k is correctly classified for the whole range $[\ _1; 1]$. Second, we show for any other sample once it become classified right it will remain so. Combining these two cases, we prove the monotonicity of the error rate.

Class k. We first show that every $x \in S_k$ is classified correctly for any $\in [\ _1; 1]$. According to Lemma 4, we know that

$$f_k^{[\ 1]}(x) > f_i^{[\ 1]}(x)$$

for any $i \neq k$. We only need to prove that $f_k^{[\ ]}(x) \ f_i^{[\ ]}(x)$ is increasing for $\in [\ _1; 1]$. Expanding $f_k^{[\ ]}(x) \ f_i^{[\ ]}(x)$, we have

$$\begin{aligned} & f_k^{[\ ]}(x) \ f_i^{[\ ]}(x) \\ &= (1 \ ) \left\langle W_{k,:}^{(0)}, x \right\rangle^r + \left\langle W_{k,:}^{(T)}, x \right\rangle^r (1 \ ) \left\langle W_{i,:}^{(0)}, x \right\rangle^r + \left\langle W_{i,:}^{(T)}, x \right\rangle^r + b_k^{(T)} \ b_i^{(T)}, \end{aligned}$$

which is increasing since $\left\langle W_{k,:}^{(T)}, x \right\rangle \ \left\langle W_{k,:}^{(0)}, x \right\rangle \ \left\langle W_{i,:}^{(T)}, x \right\rangle \ \left\langle W_{i,:}^{(0)}, x \right\rangle \ O(\ )$ and $b_k^{(T)} \ b_i^{(T)} > (1)$.

Other classes. For any class $i \neq k$, from Lemma 4, we know that it is classified incorrectly for $\in [\ _1; \ _2]$. We prove that once it become classified correctly at some $^0 \in (\ _2; 1]$, it remains so for $\in [\ ^0; 1]$.

We show that at $= \ ^0$, for any $x \in S_i$, if $f_i^{[\ ]}(x) > f_j^{[\ ]}(x)$ for all $j \neq i$, we have $\frac{@}{@} \left[f_i^{[\ ]}(x) \ f_j^{[\ ]}(x) \right] > 0$. Expanding $f_i^{[\ ]}(x) \ f_j^{[\ ]}(x)$, we have

$$\begin{aligned} & f_i^{[\ ]}(x) \ f_j^{[\ ]}(x) \\ &= \left[\left\langle W_{i,:}^{[\ ]}, x \right\rangle^r + b_i^{[\ ]} \right] \left[\left\langle W_{j,:}^{[\ ]}, x \right\rangle^r \ b_j^{[\ ]} \right] \\ &= \left\langle W_{i,:}^{[\ ]}, x \right\rangle^r \left\langle W_{j,:}^{[\ ]}, x \right\rangle^r \ b_j^{(T)} \ b_i^{(T)}. \end{aligned}$$

Since $f_i^{[\ ]}(x) \ f_j^{[\ ]}(x) > 0$, we have

$$\left\langle W_{i,:}^{[\ ]}, x \right\rangle^r > b_j^{(T)} \ b_i^{(T)},$$

where we use $\left\langle W_{j,:}^{[\ ]}, x \right\rangle > 0$. Computing $\frac{@}{@} \left[f_i^{[\ ]}(x) \ f_j^{[\ ]}(x) \right]$, we have

$$\begin{aligned} & \frac{@}{@} \left[f_i^{[\ ]}(x) \ f_j^{[\ ]}(x) \right] \\ &= \frac{@}{@} \Big[(1 \ ) \left\langle W_{i,:}^{(0)}, x \right\rangle^r + \left\langle W_{i,:}^{(T)}, x \right\rangle^r (1 \ ) \left\langle W_{j,:}^{(0)}, x \right\rangle^r + \left\langle W_{j,:}^{(T)}, x \right\rangle^r + b_i^{(T)} \ b_j^{(T)} \Big] \\ &= r \left[\left\langle W_{i,:}^{(0)}, x \right\rangle + \left\langle W_{i,:}^{(T)}, x \right\rangle \left\langle W_{i,:}^{(0)}, x \right\rangle \right]^{r \ 1} \left\langle W_{i,:}^{(T)}, x \right\rangle \left\langle W_{i,:}^{(0)}, x \right\rangle \ b_j^{(T)} \ b_i^{(T)} \ O(\ ^r); \end{aligned}$$

where the inequality uses $\langle W_{j,:}^{(0)}, x \rangle \geq \langle W_{j,:}^{(T)}, x \rangle - O(\epsilon)$.

If $b_j^{(T)} - b_i^{(T)} \geq 0$, we only need to prove

$$r \langle W_{i,:}^{(0)}, x \rangle^{r-1} \langle W_{i,:}^{(T)}, x \rangle - \langle W_{i,:}^{(0)}, x \rangle^{r-1} \langle W_{i,:}^{(T)}, x \rangle - \langle W_{i,:}^{(0)}, x \rangle^{r-1} \langle W_{i,:}^{(0)}, x \rangle - O(\epsilon^r) > 0,$$

which holds since $\langle W_{i,:}^{(T)}, x \rangle - \langle W_{i,:}^{(0)}, x \rangle \geq \langle W_{i,:}^{[t]}, x \rangle \geq (1)$.

If $b_j^{(T)} - b_i^{(T)} > 0$, we have

$$\begin{aligned}
&\frac{\partial}{\partial} f_i^{[t]}(x) - f_j^{[t]}(x) \\
&= r \langle W_{i,:}^{(0)}, x \rangle^{r-1} \langle W_{i,:}^{(T)}, x \rangle - \langle W_{i,:}^{(0)}, x \rangle^{r-1} \langle W_{i,:}^{(T)}, x \rangle - \langle W_{i,:}^{(0)}, x \rangle^{r-1} \langle W_{i,:}^{(0)}, x \rangle - b_j^{(T)} + b_i^{(T)} - O(\epsilon^r) \\
&> \frac{(1 - O(\epsilon^r)) r \langle W_{i,:}^{(T)}, x \rangle \langle W_{i,:}^{(0)}, x \rangle^{r-1}}{\langle W_{i,:}^{(0)}, x \rangle^{r-1} + \langle W_{i,:}^{(T)}, x \rangle^{r-1} - \langle W_{i,:}^{(0)}, x \rangle^{r-1}} - b_j^{(T)} + b_i^{(T)} - b_j^{(T)} + b_i^{(T)},
\end{aligned}$$

where the last inequality uses $\langle W_{i,:}^{[t]}, x \rangle > b_j^{(T)} - b_i^{(T)}$. Therefore, to prove $\frac{\partial}{\partial} f_i^{[t]}(e_i) - f_j^{[t]}(e_i) > 0$, we only need to prove $\frac{(1 - O(\epsilon^r)) r \langle W_{i,:}^{(T)}, x \rangle \langle W_{i,:}^{(0)}, x \rangle^{r-1}}{\langle W_{i,:}^{(0)}, x \rangle^{r-1} + \langle W_{i,:}^{(T)}, x \rangle^{r-1} - \langle W_{i,:}^{(0)}, x \rangle^{r-1}} \geq 1$.

We have

$$\frac{(1 - O(\epsilon^r)) r \langle W_{i,:}^{(T)}, x \rangle \langle W_{i,:}^{(0)}, x \rangle^{r-1}}{\langle W_{i,:}^{(0)}, x \rangle^{r-1} + \langle W_{i,:}^{(T)}, x \rangle^{r-1} - \langle W_{i,:}^{(0)}, x \rangle^{r-1}} \geq \frac{(1 - O(\epsilon^r)) r \langle W_{i,:}^{(T)}, x \rangle \langle W_{i,:}^{(0)}, x \rangle^{r-1}}{2 \langle W_{i,:}^{(T)}, x \rangle^{r-1}} \geq 1.$$

The first inequality requires $\langle W_{i,:}^{(0)}, x \rangle \leq \langle W_{i,:}^{(T)}, x \rangle$ and the second inequality uses $r \geq 3, (1 - O(\epsilon^r)) r \geq 3/2$. To prove $\langle W_{i,:}^{(0)}, x \rangle^{r-1} - \langle W_{i,:}^{(T)}, x \rangle^{r-1} \leq \langle W_{i,:}^{(0)}, x \rangle^{r-1}$, it's equivalent to show $\langle W_{i,:}^{(0)}, x \rangle \leq \frac{1}{1+\epsilon} \langle W_{i,:}^{(T)}, x \rangle$. Since $\epsilon_2 = \frac{1}{1+O(\sqrt{m})} \epsilon^{\frac{r}{r-1}} R_{\min}^{\frac{1}{r-1}}$, we can lower bound $\frac{1}{1+\epsilon}$ as follows,

$$\begin{aligned}
\frac{1}{1+\epsilon} &\geq \frac{1}{2} \frac{1}{1 + O(\sqrt{m})} \epsilon^{\frac{r}{r-1}} R_{\min}^{\frac{1}{r-1}} \\
&\geq \frac{1}{8} R_{\min}^{\frac{1}{r-1}},
\end{aligned}$$

where the first inequality uses $\epsilon \leq 1$ and the second inequality uses $1 + O(\sqrt{m}) \leq 2, r \geq 2$. So we have $\frac{1}{1+\epsilon} \langle W_{i,:}^{(T)}, x \rangle \geq \frac{1}{8} R_{\min}^{\frac{1}{r-1}} R_{\min}^{1/r} \geq R_{\min}^{\frac{1}{r-1}} R_{\min}^{1/r}$. Therefore, we only need $\epsilon \leq O\left(R_{\min}^{\frac{1}{r-1}} R_{\min}^{1/r}\right)$ to ensure that $\langle W_{i,:}^{(0)}, x \rangle \leq \langle W_{i,:}^{(T)}, x \rangle - \langle W_{i,:}^{(0)}, x \rangle$.

B.2 Loss interpolation for r-homogeneous weight model

In this section, we give a proof of Theorem 5.

Theorem 5 (Loss Interpolation) Suppose the network at initialization and after training satisfy the properties described in Theorem 2 and Induction Hypothesis 1. For any $\epsilon \in (0, 1)$, suppose $\epsilon \leq O(R_{\max}^{1/r})$; there exist $\epsilon_3 = \frac{1}{W_{\max}^{1/r}}$ and $\epsilon_4 = (1 + O(\epsilon)) \frac{1}{r} \frac{R_{\max}}{r}^{\frac{1}{r-1}}$ such that

1. for all $\epsilon \in [0, \epsilon_3]$, we have $\log \epsilon - \epsilon \leq \frac{1}{N} L(W^{[t]}, b^{[t]}) \leq \log \epsilon + \epsilon_{\max} + \epsilon$;
2. for all $\epsilon \in [\epsilon_4, 1]$, the loss is monotonically decreasing.

Proof of Theorem 5. This theorem directly follows from Lemma 6 and Lemma 7.

Next, we prove the initial loss plateau in Lemma 6 and the monotonicity in Lemma 7.

Lemma 6 (Loss Plateau) In the same setting as in Theorem 5, for any $\epsilon > 0$, there exists $\eta_3 = \frac{\epsilon^{1/r}}{W_{\max}}$ such that for all $\eta \in [0, \eta_3]$

$$N(\log k - \epsilon) \le L(W^{[T]}, b^{[T]}) \le N(\log k + \eta W_{\max} + \epsilon).$$

We show that for $\eta \in [0, \eta_3]$, the weights $W^{[T]}$ is negligible and the bias dominates, which then gives a lower bound and an upper bound of the loss.

Proof of Lemma 6. Since $\eta_3 = \frac{\epsilon^{1/r}}{W_{\max}}$ and $\eta = O(\epsilon^{1/r})$, we have

$$\langle W_{i,:}^{[T]}, x \rangle^r = \langle W_{i,:}^{(0)}, x \rangle^r + \left(\langle W_{i,:}^{(T)}, x \rangle^r - \langle W_{i,:}^{(0)}, x \rangle^r \right) \le \left(1 + \frac{1}{r}\right)^r \epsilon^r \le e\epsilon,$$

for all $i \in [k], x \in S$.

We can divide the dataset S into N/k disjoint subsets $\{P_l\}_{l=1}^{N/k}$ where each P_l contains exactly one sample from each class. Next, we bound the total loss of each subset P_l. Without loss of generality, let's consider subset P_1 and suppose $x^{(i)}$ is the i-th class sample in this subset. For convenience, we denote the total loss of samples in P_1 as $L_1(W^{[T]}, b^{[T]})$.

Lower bounding $L_1(W^{[T]}, b^{[T]})$: We have

$$L_1(W^{[T]}, b^{[T]}) = \sum_{i \in [k]} \log\left(\frac{\sum_{j \in [k]} \exp\left(f_j^{[T]}(x^{(i)})\right)}{\exp\left(f_i^{[T]}(x^{(i)})\right)}\right)$$
$$= \log\left(\prod_{i \in [k]} \frac{\sum_{j \in [k]} \exp\left(f_j^{[T]}(x^{(i)})\right)}{\exp\left(f_i^{[T]}(x^{(i)})\right)}\right)$$
$$\ge \log\left(\left(\frac{k}{\sum_{i \in [k]} \frac{\exp\left(f_i^{[T]}(x^{(i)})\right)}{\sum_{j \in [k]} \exp\left(f_j^{[T]}(x^{(i)})\right)}}\right)^k\right),$$

where the last inequality uses the HM-GM inequality. We can then upper bound $\sum_{i \in [k]} \frac{\exp\left(f_i^{[T]}(x^{(i)})\right)}{\sum_{j \in [k]} \exp\left(f_j^{[T]}(x^{(i)})\right)}$ as follows,

$$\sum_{i \in [k]} \frac{\exp\left(f_i^{[T]}(x^{(i)})\right)}{\sum_{j \in [k]} \exp\left(f_j^{[T]}(x^{(i)})\right)} = \sum_{i \in [k]} \frac{\exp\left(\langle W_{i,:}^{[T]}, x^{(i)} \rangle^r + b_i\right)}{\sum_{j \in [k]} \exp\left(\langle W_{j,:}^{[T]}, x^{(i)} \rangle^r + b_j\right)}$$
$$\le \sum_{i \in [k]} \frac{\exp(b_i + \epsilon)}{\sum_{j \in [k]} \exp(b_j)}$$
$$= \frac{\sum_{i \in [k]} \exp(b_i)}{\sum_{j \in [k]} \exp(b_j)} \exp(\epsilon)$$
$$= \exp(\epsilon).$$

Plugging back to the lower bound of $L_1(W^{[T]}, b^{[T]})$, we have

$$L_1(W^{[T]}, b^{[T]}) \ge k \log \frac{k}{\exp(\epsilon)} = k(\log k - \epsilon).$$

Upper bounding $L_1(W^{[\ ]}; b^{[\ ]})$: We have

$$L_1(W^{[\ ]}; b^{[\ ]}) = \sum_{i \in [k]} \log\left(\frac{\sum_{j \in [k]} \exp\left(f_j^{[\ ]}(x^{(i)})\right)}{\exp\left(f_i^{[\ ]}(x^{(i)})\right)}\right)$$
$$\sum_{i \in [k]} \log\left(\frac{\sum_{j \in [k]} \exp(b_j + \epsilon)}{\exp(b_i)}\right)$$
$$k \log(k \exp(\ _{max} + \epsilon))$$
$$k(\log k + \ _{max} + \epsilon)$$

The above analysis applies for every subset S, so we have

$$N(\log k \quad \epsilon) \quad L(W^{[\ ]}; b^{[\ ]}) \quad N(\log k + \ _{max} + \epsilon):$$

Next we show that when is reasonably large, we have $f_i^{[\ ]}(e_i) \quad f_j^{[\ ]}(e_i)$ increasing for all $\neq j$, which then implies that the loss is decreasing.

Lemma 7 (Loss Monotonicity) In the same setting as in Theorem 5, there exists $= (1 + O(\ ))\frac{1}{r-1}\left(\frac{R_{max}}{r}\right)^{\frac{1}{r-1}}$ such that the loss is monotonically decreasing for $\in [\ _4; 1]$.

Proof of Lemma 7. To prove that the loss is monotonically decreasing, we only need to show that for any $i \in [k]$ and any $x \in S_i$, $f_i^{[\ ]}(x) \quad f_j^{[\ ]}(x)$ is monotonically increasing for $\neq i$.

Same as in Lemma 5, it's easy to prove that for $x \in S_k$, $f_k(x) \quad f_j(x)$ with $j \neq k$ monotonically increases for $\in [0; 1]$. So we focus on other classes.

For $i \neq k$, we show that $\frac{@}{@}\left(f_i^{[\ ]}(x) \quad f_j^{[\ ]}(x)\right) > 0$ for $x \in S_i$ when $\ _4$;

$$\begin{aligned}
&\frac{@}{@}\left(f_i^{[\ ]}(e_i) \quad f_j^{[\ ]}(e_i)\right) \\
&= \frac{@}{@}\left(\left\langle (1 \quad)W_{i,:}^{(0)} + W_{i,:}^{(T)}, x\right\rangle_r \quad \left\langle (1 \quad)W_{j,:}^{(0)} + W_{j,:}^{(T)}, x\right\rangle_r + b_i^{(T)} \quad b_j^{(T)}\right) \\
&\quad r\left\langle (1 \quad)W_{i,:}^{(0)} + W_{i,:}^{(T)}, x\right\rangle_{r\ 1}\left\langle W_{i,:}^{(T)}, x\right\rangle \quad \left\langle W_{i,:}^{(T)}, x\right\rangle\left\langle W_{i,:}^{(0)}, x\right\rangle \quad b_k^{(T)} \quad b_i^{(T)} \quad O(\ ^r) \\
&\quad r\ ^{r\ 1}\left\langle W_{i,:}^{(T)}, x\right\rangle_{r\ 1}\left\langle W_{i,:}^{(T)}, x\right\rangle \quad \left\langle W_{i,:}^{(T)}, x\right\rangle\left\langle W_{i,:}^{(0)}, x\right\rangle \quad b_k^{(T)} \quad b_i^{(T)} \quad O(\ ^r) \\
&\quad r\ ^{r\ 1}\left(1 \quad O\left(\frac{1}{\ ^{1=r}_{min}}\right)\right)\left\langle W_{i,:}^{(T)}, x\right\rangle_r \quad b_k^{(T)} \quad b_i^{(T)} \quad (1 + O(\ ^r)) \\
&> 0;
\end{aligned}$$

where the second last inequality uses $\left\langle W_{i,:}^{(0)}, x\right\rangle = \left\langle W_{i,:}^{(T)}, x\right\rangle \quad O\left(\ ^{1=r}_{min}\right)$. The last inequality requires

$$r\ ^{r\ 1} \quad (1 + O(\ ))\frac{b_k^{(T)} \quad b_i^{(T)}}{\left\langle W_{i,i}^{(T)}\right\rangle_r}$$

which is satisfied as long as $(1 + O(\ ))\frac{1}{r-1}\left(\frac{R_{max}}{r}\right)^{\frac{1}{r-1}}$ where $R_{max} = \max_{i \in [k\ 1]}\ _i/[W_{i,i}^{(T)}]^r$.

B.3 Plateau for deep fully-connected networks

In this section, we consider fully-connected neural networks as defined in Section 3 and prove that both the error and loss curves have plateau. We restate Theorem 1 as follows.

Theorem 1. Suppose the network is de ned as in Equation (1) and suppose the weights satisfy $V_i^{(0)} \quad ; \quad V_i^{(T)} \quad V_{max}$ for all layers $i \ 2 \ [r]$: On a k-class balanced dataset whose inputs have $`_2$ norm at most 1, if Assumption 1 holds, for any > 0; as long as $< \min \ \frac{1=r}{r}; \frac{1}{r^2}; \ \frac{1}{2e} \ ^{\frac{2}{r \ 2}}$,

there exist $_1 = \frac{}{_{min}}; \ _2 = \frac{1}{1+ \ p \ -} \ ^{\frac{r}{r \ 1}} \ \frac{_{min}}{2V_{max}^r} \ ^{\frac{1}{r \ 1}}$ and $_3 = \frac{1=r}{V_{max}}$ such that

1. for all $2 \ [\ _1; \ _2]$; the error is $1 \quad 1=k$;
2. for all $2 \ [0; \ _3]$; we have $\log k \quad 2e \ \frac{1}{N} L \ n \ V_i^{[\ ]} \ ; b^{[\ ]} \quad \log k + \quad _{max} + 2 e \ ;$ where N is the number of training examples.

Proof of Theorem 1. This theorem directly follows from Lemma 8 and Lemma 9.

We separately prove the plateau of error interpolation in Lemma 8 and the plateau of loss interpolation in Lemma 9. Then, Theorem 1 is simply a combination of Lemma 8 and Lemma 9. For convenience, we denote $h(x) := V_r \ (V_{r \ 1} \quad (V_1 x) \ )$ in the proof.

Lemma 8. In the setting of Theorem 1, there exist $_1 = \frac{}{_{min}}$ and $_2 = \frac{1}{1+ \ p \ -} \ ^{\frac{r}{r \ 1}} \ \frac{_{min}}{2V_{max}^r} \ ^{\frac{1}{r \ 1}}$ such that the error is $1 \quad 1=k$ for any interpolation point $2 \ [\ _1; \ _2]$:

Proof of Lemma 8. Recall that the network output under input x is $g(x) := V_r \ (V_{r \ 1} \quad (V_1 x) \ ) + b$. Similar as in the proof of Lemma 4, we only need to show that for all $2 \ [\ _1; \ _2]$; we have

$$g_i^{[\ ]}(x) < g_k^{[\ ]}(x)$$

for all $i \ 6 = k$ and all samples x, which immediately implies the error is $1 \quad 1=k$:

For $2 \ \left[0; \frac{p \ -}{V_{max}}\right]$. If $_1 = \frac{}{_{min}} \ \frac{p \ -}{V_{max}}$, we only need to consider the case when $2 \ \left[\frac{p \ -}{V_{max}}; \ _2\right]$: So here we assume $\frac{}{_{min}} < \frac{p \ -}{V_{max}}$: We can lower bound $g_k^{[\ ]}(x) \quad g_i^{[\ ]}(x)$ as

$$\begin{aligned} g_k^{[\ ]}(x) \quad g_i^{[\ ]}(x) = h_k^{[\ ]}(x) + b_k^{[\ ]} \quad h_i^{[\ ]}(x) \quad b_i^{[\ ]} & \\ \prod_{j \ 2 \ [r]} (1 \quad)V_j^{(0)} + V_j^{(T)} & \\ _{min} \quad 2 \quad & \\ _{min} \quad 2(\ + V_{max})^r \ ; & \end{aligned}$$

where the first inequality holds because $b_k^{[\ ]} \quad b_i^{[\ ]} \quad _{min}$ and $h_k^{[\ ]}(x) \ ; \ h_i^{[\ ]}(x) \quad \prod_{j \ 2 \ [r]} (1 \quad)V_j^{(0)} + V_j^{(T)}$: The second inequality uses $(1 \quad)V_j^{(0)} + V_j^{(T)} \quad (1 \quad) \ V_j^{(0)} + \quad V_j^{(T)} \quad + V_{max}$:

Since $2 \ \left[0; \frac{}{_{min}}; \frac{p \ -}{V_{max}}\right]$; we have

$$\begin{aligned} g_k^{[\ ]}(x) \quad g_i^{[\ ]}(x) \quad \frac{}{_{min}} \quad _{min} \quad 2 \quad + \frac{p \ -}{V_{max}} V_{max} \quad ; & \\ 2 \quad 1 + \frac{1}{r} \quad p \ - \ ^r \quad ; & \\ 2e^{ \ r=2} & \\ > 0; & \end{aligned}$$

where the second inequality assumes $1=r^2$ and the last inequality assumes $\leq \ \frac{1}{2e} \ ^{\frac{2}{r \ 2}}$:

For $\in \left[\frac{p-}{V_{max}}, {}_2\right]$. Similar as above, we only need to show that $_{min}\ 2(\ +\ V_{max})^r > 0$. Since $\frac{p-}{V_{max}}$; we have $^{p-}V_{max}$: Therefore, we have

$$_{min}\ 2(\ +\ V_{max})^r \quad _{min}\ 2\ \left(1+\frac{p-}{V_{max}}\right)^r > 0;$$

where the second inequality holds as long as ${}_2 := \frac{1}{2}\ ^{\frac{1}{r-1}}\ \frac{1}{1+p-}\ ^{\frac{r}{r-1}}\ \frac{min}{V_{max}^r}\ ^{\frac{1}{r-1}}$:

Next, we show that for $\in [0; \frac{1=r}{V_{max}}]$; the loss cannot decrease by much. Similar as in Lemma 6, we prove that the signal is very small and the logit is dominated by the bias term. This then gives a lower and upper bounds for the loss.

Lemma 9. In the setting of Theorem 1, there exists $= \frac{1=r}{V_{max}}$ such that for all $\in [0; {}_3]$

$$\log k \quad 2e \quad \frac{1}{N}L\left\{V_i^{[\ ]}; b^{[\ ]}\right\} \quad \log k + \quad _{max} + 2e;$$

where N is the number of samples.

Proof of Lemma 9. Since ${}_3 = \frac{1=r}{V_{max}}$ and $\frac{1=r}{r}$; we have

$$h^{[\ ]}(x) \quad (\ +\ V_{max})^r \quad e$$

for all input x:

Similar as in the proof of Lemma 6, we can show that

$$\log k \quad 2e \quad \frac{1}{N}L\left\{V_i^{[\ ]}; b^{[\ ]}\right\} \quad \log k + \quad _{max} + 2e;$$

where we have an additional factor of 2 before e because now the signal can be positive or negative. Here N is the number of samples.

C Proof of training dynamics

In this section, we give the complete proof of Theorem 2.

Theorem 2. Suppose the neural network, dataset and optimization procedure are as defined in Section 4. Suppose initialization scale (1), noise level $\tilde{(1)}$, dimension $d \tilde{(1 = {}^{2r-2})}$ and number of samples $N \tilde{(1 = {}^{r-1})}$, with probability at least 0.99 in the initialization, there exists time $T = (\log(1 =)= {}^{r-2})$ such that we have

1. zero error: for all different $i; j \in [k]$ and for all $x \in S_i$, $f_i^{(T)}(x) \quad f_j^{(T)}(x) + (1)$;
2. bias gap: $b_i^{(T)} \quad \max_{i \in i} b_i^{(T)} \quad (1)$ with $i = \arg\max_{i \in [k]} b_i^{(T)}$:

Proof of Theorem 2. This theorem directly follows from Proposition 1.

We consider the 2-homogeneous-weight network as defined in Section 4. Our simple model simulates a depth-r ReLU/linear network with bias on the output layer, in the sense that the weights signal is r-homogeneous while the bias is 1-homogeneous in the parameters.

Next, we prove Proposition 1 while leaving the proof of supporting lemmas into Section C.1. Through the proof of Proposition 1, we restate the lemmas when we use it for the convenience of readers.

Proposition 1 (Induction Hypothesis). In the same setting of Theorem 2, with probability at least 0.99 in initialization, there exist time points $0 =: s_1 < t_1 < s_2 < t_2 < \quad < s_{k-1} < t_{k-1} < s_k := T$ with $t_i \quad s_i = (\log(1 =)= {}^{r-2})$ and $s_{i+1} \quad t_i = (1)$ for $i \in [k-1]$ such that for any $t \in [s_i; s_{i+1}]$;

1. (classes not yet learned) for any class $j; j' \neq i+1$; we have (1) $b_j^{(t)} \quad \max_{i' \in [k]} b_{i'}^{(t)} \quad O(\ ^r)$, (2) $b_j^{(t)} \quad b_{j'}^{(t)} \quad O(\ ^r)$ and (3) $W_{j;j}^{(t)} \quad O(\ )$;

2. (classes already learned) for any class $j < i - 1$, we have (1) $b_j^{(t)} \le \max_{i' \in [k]} b_{i'}^{(t)} + (1)$, (2) $f_j^{(t)}(x) \le f_{i'}^{(t)}(x) + (1)$ for $i' \ne j, x \in S_j$ and (3) $W_{j,j}^{(t)} \le (1)$;

3. (parameters movement) (1) for any $j \in [k]$, $(\,) = W_{j,j}^{(0)} < W_{j,j}^{(t)}$; (2) for any distinct $j, j' \in [k]$, $0 < W_{j,j'}^{(t)} \le O(\,)$ and (3) for any $j, j' \in [k]$ and any $x \in S_{j'}$, $\langle W_{j,:}^{(t)}, x \rangle \le \min\{O(\,), W_{j,j'}^{(t)}\}$.

Proof of Proposition 1. Through the proof, we assume the conditions in Theorem 2 hold in all the lemmas without explicitly stated. At the initialization, we have the following properties with probability at least 0.99.

Lemma 10 (Initialization). With probability at least 0.99 in the initialization, we have

1. for all $j, j' \in [k]$, $W_{j,j'}^{(0)} = (\,)$;
2. for all distinct $j, j' \in [k]$, $W_{j,j}^{(0)} - W_{j',j'}^{(0)} = (\,)$;
3. for all $x \in S$, $\|x\| \le O(\,)$;
4. for all distinct $x, x' \in S$, $\langle x, x' \rangle \le O\left(\frac{\sqrt{\log(N)}}{p^{\bar{d}}}\right)$;
5. for all $j \in [k]$ and all $x \in S$, $\langle x, e_j \rangle \le \langle x, W_{j,:}^{(0)} \rangle \le O\left(\frac{\sqrt{\log(N)}}{p^{\bar{d}}}\right)$.

Without loss of generality, we assume $W_{1,1}^{(0)} > W_{2,2}^{(0)} > \cdots > W_{k,k}^{(0)}$.

It's not hard to verify that the induction hypothesis holds at the initialization[4]. For any $i \in [k-1]$, assuming the induction hypothesis holds for time $[0, s_i]$; now we prove that it continues to hold in $[s_i, s_{i+1}]$. Next, we first prove the first two properties in the Proposition 1 and leave the last one at the end.

The learning of $x \in S_i$ can be divided into four stages:

1. Stage 0: for $t \in [s_i, t_i]$ with $t_i - s_i = O(\log(1/\cdot)/\cdot^2)$: During this stage, $W_{i,i}^{(t)}$ grows to a small constant $\cdot_0$.
2. Stage 1: for $t \in [t_i, t_i^{(w)}]$ with $t_i^{(w)} - t_i = O(1)$: In this stage, $W_{i,i}^{(t)}$ grows from $\cdot_0$ to a large constant $\cdot_1$.
3. Stage 2: for $t \in [t_i^{(w)}, t_i^{(u)}]$ with $t_i^{(u)} - t_i^{(w)} = O(1)$: At the end of this stage, we have $\min_{x \in S_i} u_i^{(t)}(x) \ge 1 - \cdot_2$ for a small constant $\cdot_2$.
4. Stage 3: for $t \in [t_i^{(u)}, t_i^{(b)}]$ with $t_i^{(b)} - t_i^{(u)} = O(1)$, where $t_i^{(b)} = s_{i+1}$. During this stage, we have $b_i^{(t)} - b_k^{(t)}$ decreases to $\cdot_3$ with $\cdot_3$ a positive constant.

Next, we consider these four stages one by one.

Stage 0. We show that $W_{i,i}^{(t)}$ increases faster than $W_{i+1,i+1}^{(t)}$ so that $W_{i,i}^{(t)}$ reaches a constant while $W_{i+1,i+1}^{(t)}$ is still $O(\,)$. We use the following lemma to characterize the increasing rate of $W_{i+1,i+1}^{(t)}$ and $W_{i,i}^{(t)}$.

[4] We will maintain a stronger bound on $\langle W_{j,:}^{(t)}, x \rangle$ by proving $\langle W_{j,:}^{(t)}, x \rangle \le O(\sqrt{\log N})$, which implies $\langle W_{j,:}^{(t)}, x \rangle \le O(\,)$ as long as $\cdot \le O(1/\sqrt{\log N})$.

Lemma 11. For any $j \in [k]$, we have

$$\frac{\partial W_{j,j}^{(t)}}{\partial t} \geq -O\left(\frac{r^{-1}\sqrt{\log N}}{d}\right).$$

If $\min_{x \in S_j}(1 - u_j(x)) \geq \Omega(1)$, we further have

$$1 - O\left(\sqrt{\log N}\right)\frac{k}{N}\sum_{x \in S_j}(1 - u_j(x))\, r W_{j,j}^{r-1} \leq \frac{\partial W_{j,j}^{(t)}}{\partial t} \leq 1 + O\left(\sqrt{\log N}\right)\frac{k}{N}\sum_{x \in S_j}(1 - u_j(x))\, r W_{j,j}^{r-1}.$$

It's not hard to verify that $\min_{x \in S_i}\ 1 - u_i^{(t)}(x) \geq \Omega(1)$, $\min_{x \in S_{i+1}}\ 1 - u_{i+1}^{(t)}(x) \geq \Omega(1)$, so we have

$$\frac{\partial W_{i,i}^{(t)}}{\partial t} \geq 1 - O\left(\sqrt{\log N}\right)\frac{k}{N}\sum_{x \in S_i} 1 - u_i^{(t)}(x)\ r\left[W_{i,i}^{(t)}\right]^{r-1};$$

$$\frac{\partial W_{i+1,i+1}^{(t)}}{\partial t} \leq 1 + O\left(\sqrt{\log N}\right)\frac{k}{N}\sum_{x \in S_{i+1}} 1 - u_{i+1}^{(t)}(x)\ r\left[W_{i+1,i+1}^{(t)}\right]^{r-1}.$$

We can upper bound $1 - u_{i+1}^{(t)}(x)$ for any $x \in S_{i+1}$ as follows,

$$1 - u_{i+1}^{(t)}(x) = \frac{\sum_{i' \in [k]} \exp\left(f_{i'}^{(t)}(x) - \exp\left(f_{i+1}^{(t)}(x)\right)\right.}{\sum_{i' \in [k]} \exp\left(f_{i'}^{(t)}(x)\right)}$$
$$\frac{\sum_{i' \in [k]} \exp\left(b_{i'}^{(t)} - \exp\left(b_{i+1}^{(t)}\right)\right.}{\sum_{i' \in [k]} \exp\left(b_{i'}^{(t)}\right)}(1 + O(\varepsilon^r));$$

where the inequality uses $\left\langle W_{i',:}^{(t)}, x\right\rangle \geq -O(\varepsilon)$ for every $i' \in [k]$.

We can lower bound $1 - u_i^{(t)}(x)$ for $x \in S_i$ as follows,

$$1 - u_i^{(t)}(x) = \frac{\sum_{i' \in [k]} \exp\left(f_{i'}^{(t)}(x) - \exp\left(f_i^{(t)}(x)\right)\right.}{\sum_{i' \in [k]} \exp\left(f_{i'}^{(t)}(x)\right)}$$
$$\frac{\sum_{i' \in [k]} \exp\left(b_{i'}^{(t)} - \exp\left(b_i^{(t)}\right)\right.}{\sum_{i' \in [k]} \exp\left(b_{i'}^{(t)}\right)}(1 - O(\varepsilon_0^r))$$
$$\frac{\sum_{i' \in [k]} \exp\left(b_{i'}^{(t)} - \exp\left(b_{i+1}^{(t)}\right)\right.}{\sum_{i' \in [k]} \exp\left(b_{i'}^{(t)}\right)}(1 - O(\varepsilon_0^r) - O(\varepsilon^r)).$$

The first inequality uses $\left\langle W_{i',:}^{(t)}, x\right\rangle \geq -O(\varepsilon_0)$ for every $i' \in [k]$. The second inequality uses $b_i^{(t)} \geq b_{i+1}^{(t)} + O(\varepsilon^r)$, which is guaranteed by the following lemma.

Lemma 2 (Bias Gap Control I). For any different $j', j \in [k]$, if $W_{j',j'} - W_{j',j} \geq O(\varepsilon)$ and $b_{j'} - b_j \geq -O(\varepsilon^r)$, $b_j \geq \max_{i' \in [k]} b_{i'} - O(\varepsilon^r)$, we have $b_{j'} - b_j < 0$.

According to Lemma 10, we know there exists constant $C > 1$ such that $W_{i,i}^{(0)} \geq C W_{i+1,i+1}^{(0)}$. Choose constant S such that $S^{\frac{1}{r-2}} = \sqrt{C}$ and $W_{i,i}^{(0)} \geq S^{\frac{1}{r-2}}\sqrt{C} W_{i+1,i+1}^{(0)}$. Choosing ε_0 as small constants and $\Omega(1/\sqrt{\log N}) \leq \varepsilon \leq O(1)$, we have

$$1 + O\left(\sqrt{\log N}\right)\max_{x \in S_{i+1}} 1 - u_{i+1}^{(t)}(x) \leq S\left(1 - O\left(\sqrt{\log N}\right)\min_{x \in S_i} 1 - u_i^{(t)}(x)\right).$$

We can also lower bound $1 - u_i^{(t)}(x)$ for $x \in S_i$ by a constant,

$$1 - u_i^{(t)}(x) \geq \frac{\sum_{i' \in [k]} \exp(b_{i'}^{(t)}) - \exp(b_i^{(t)})}{\sum_{i' \in [k]} \exp(b_{i'}^{(t)})}(1 - O(\epsilon_0)) \geq \frac{\exp(b_{i+1}^{(t)})}{\sum_{i' \in [k]} \exp(b_{i'}^{(t)})}(1 - O(\epsilon_0)) \geq \Omega(1);$$

where the last inequality holds because ϵ_0 is a small constant and $b_{i+1}^{(t)} \geq \max_{i' \in [k]} b_{i'}^{(t)} - O(\epsilon^r)$.

Lemma 12 (Adapted from Lemma C.19 in Allen-Zhu & Li (2020)). Let $r \geq 3$ be a constant and let $\{W_{i,i}^{(t)}, W_{j,j}^{(t)}\}_{t \geq 0}$ be two positive sequences updated as

$$\frac{\partial W_{i,i}^{(t)}}{\partial t} \geq C_t \left[W_{i,i}^{(t)}\right]^{r-1} \text{ for some } C_t = \Omega(1);$$
$$\frac{\partial W_{j,j}^{(t)}}{\partial t} \leq S C_t \left[W_{j,j}^{(t)}\right]^{r-1} \text{ for some } S = \Omega(1).$$

Suppose $W_{i,i}^{(0)} \geq W_{j,j}^{(0)} S^{\frac{1}{r-2}}(1 + \Omega(1))$; then we must have for every $A = O(1)$, let t_i be the first time such that $W_{i,i}^{(t_i)} \geq A$, then

$$W_{j,j}^{(t_i)} \leq O(W_{j,j}^{(0)}).$$

Then, according to Lemma 12, we know that there exists $\tau = O(\log(1/\epsilon)/\epsilon^{r-2})$ such that $W_{i,i}^{(t_i)} = \epsilon_0$ and $W_{i+1,i+1}^{(t_i)} \leq O(\epsilon)$. By similar argument, we also know $W_{j,j}^{(t_i)} \leq O(\epsilon)$ for any $j \geq i+1$.

Stage 1. In this stage, we show that $W_{i,i}^{(t)}$ grows to a large constant ϵ_1 within constant time. Since $W_{i,i}^{(t)} \geq \epsilon_1$ and $b_{i,i}^{(t)} \geq b_{i+1,i+1}^{(t)} + O(\epsilon^r)$, we have

$$1 - u_i^{(t)}(x) \geq \Omega(1);$$

for all $x \in S_i$. This further implies,

$$\frac{\partial W_{i,i}^{(t)}}{\partial t} \geq 1 - O(\sqrt{\log N}) \frac{k}{N} \sum_{x \in S_i} 1 - u_i^{(t)}(x) \geq r\left[W_{i,i}^{(t)}\right]^{r-1} \geq \Omega(1);$$

where the inequality also uses $W_{i,i}^{(t)} \geq \epsilon_0$. Since the increasing rate is at least a constant, we know $W_{i,i}^{(t)}$ grows to ϵ_1 in constant time. For any $j \geq i+1$, since the increasing rate of $W_{j,j}^{(t)}$ is merely $O(\epsilon^{r-1})$, we know $W_{j,j}^{(t)}$ remains as $O(\epsilon)$ through Stage 1.

Stage 2. In this stage, we prove that $u_i^{(t)}(x)$ for any $x \in S_i$ grows to $1 - \epsilon_2$ with ϵ_2 a small constant. We use the following lemma to characterize the increasing rate of $f_i^{(t)}(x) - f_j^{(t)}(x)$.

Lemma 13. For any $x \in S_i$ and any $j \neq i$, if $1 - u_i(x) \geq \Omega(1)$, we have

$$\frac{\partial}{\partial t}(f_i(x) - f_j(x)) \geq (W_{i,i}^{2r-2}) - O(1).$$

Since $u_i^{(t)}(x) \leq 1 - \epsilon_2$, we know $1 - u_i^{(t)}(x) \geq \Omega(1)$. For any $j \neq i$, we have

$$\frac{\partial}{\partial t}\left(f_i^{(t)}(x) - f_j^{(t)}(x)\right) \geq \left[W_{i,i}^{(t)}\right]^{2r-2} - O(1) \geq \Omega(1);$$

where the last inequality holds because $W_{i,i}^{(t)} \geq \rho_1$ with ρ_1 a large enough constant.

The next lemma guarantees that at the beginning of Stage 2, we have $b_j^{(t)} \leq O(1)$, which then implies $f_i^{(t)}(x) - f_j^{(t)}(x) \leq O(1)$.

Lemma 14 (Bias Gap Control III). For any different $i, j \in [k]$, if $W_{i,i} \leq O(1)$, $W_{j,j} \leq O(\rho)$ and $b_i - b_j \leq O(1)$, we have

$$b_i - b_j > 0.$$

Let C be a constant such that $f_i^{(t)}(x) - f_j^{(t)}(x) \leq C$ for every $j \neq i$ implies $u_i(x) \geq 1 - \rho_2$. Since at the at the beginning of Stage 2, we have $f_i^{(t)}(x) - f_j^{(t)}(x) \leq O(1)$, within constant time, we have $f_i^{(t)}(x) - f_j^{(t)}(x) \leq C$ for every $j \neq i$ and $u_i(x) \geq 1 - \rho_2$.

Lemma 15 (Accuracy Monotonicity). Given any positive constant C_2, there exists positive constant C_1 such that for all different $i, j \in [k]$, as long as $W_{i,i} \geq C_1$ and $f_i(x) - f_j(x) \leq C_2$ for any $x \in S_i$, we have $\frac{\partial(f_i(x) - f_j(x))}{\partial t} > 0$.

According to Lemma 15, by choosing large enough ρ_1, we can ensure that $f_i^{(t)}(e_i) - f_j^{(t)}(e_i) \leq C$ and $u_i(e_i) \geq 1 - \rho_2$ throughout the training. Note that once $W_{i,i}^{(t)}$ rises to ρ_1, it will stay at least $\rho_1 - O(\rho)$ throughout the training, according to the gradient lower bound in Lemma 11.

Stage 3. In this stage, we prove that within constant time we have $b_i^{(t)} - b_k^{(t)} \leq \rho_3$. The following lemma shows that $b_i^{(t)} - b_k^{(t)}$ decreases in at least a constant rate.

Lemma 3 (Bias Gap Control II). There exist small positive constants C_1, C_2 such that for any $j \in [k-1]$ and any $x \in S_j$, if $1 - u_j(x) \leq C_1$, $W_{k,k} \leq O(\rho)$ and $b_j - b_k \leq C_2$, we have $b_j - b_k < -\Omega(1)$.

Choosing $\rho_2 = C_1$, $\rho_3 = C_2$ where C_1, C_2 are from Lemma 3, so we know that $b_i^{(t)} - b_k^{(t)}$ decreases at a constant rate until $b_i^{(t)} - b_k^{(t)} \leq \rho_3$. At time $t_i^{(u)}$, we know that $b_i^{t_i^{(u)}} - b_k^{t_i^{(u)}} \leq O(\rho^r)$. So within constant time, we have $b_i^{s_{i+1}} - b_k^{s_{i+1}} = \rho_3$. By Lemma 3, we also know that for any $t \geq s_{i+1}$, we have $b_i^{(t)} - b_k^{(t)} \leq \rho_3$.

The following lemma shows that $b_k^{(t)}$ is close to the maximum bias.

Lemma 1 (Coupling Biases). Assuming $W_{j_0,j_0} \geq \rho$, $W_{j,j} \leq O(\rho)$ and $b_{j_0} - b_j \geq \max_{i_0 \in [k]} b_{i_0} - O(\rho^r)$, we have $b_{j_0} - b_j > 0$ if $b_{j_0} - b_j \leq \rho^r$, and $b_{j_0} - b_j < 0$ if $b_{j_0} - b_j \leq -\rho^r$ for some positive constant ρ.

Combining Lemma 1 and Lemma 2, we know that throughout the training $\max_{k \in [k]} b_{i_0}^{(t)} - O(\rho^r)$. Therefore, we have $b_i^{(t)} \geq \max_{i_0 \in [k]} b_{i_0}^{(t)} - \rho_3$ for $t \geq s_{i+1}$.

Finally, let's bound the movement of different parameters.

Monotonicity of diagonal terms: For $j \in [k-1]$, according to Lemma 11 we know $W_{j,j}^{(t)}$ can only start decreasing when it exceeds a large constant and can only decrease by $O(\rho)$ throughout the algorithm. By choosing $\rho \leq O(1)$, we can ensure that $W_{j,j}^{(0)} < W_{j,j}^{(t)}$ for any t. For $W_{k,k}^{(t)}$, we know it monotonically increases since we always have $u_k^{(t)}(x) \leq 1 - \rho_2$ for $x \in S_k$. This is because $W_{k,k}^{(t)}$ remains as small as $O(\rho)$ through the algorithm and $b_k^{(t)} - b_{k-1}^{(t)} \leq O(1)$.

Bounding non-diagonal terms: We use the following lemma to prove that $W_{j,j_0}^{(t)} < W_{j,j_0}^{(0)} + O(\rho)$ for $j \neq j_0$.

Lemma 16. For any $j \neq j_0$, we have $|W_{j,j_0}| \leq O(\rho)$. Furthermore, there exists absolute constant $\rho > 0$ such that if $0 < W_{j,j_0} < \frac{\rho}{\log^{\frac{1}{r-2}}(1/\rho)}$, we have $|W_{j,j_0}| \leq \frac{\rho}{2\log^{\frac{1}{r-2}}(1/\rho)}$.

The first property in Lemma 16 guarantees that the increasing rate is so small that that the total increase within T time is only $O(\ )$, which then implies that $W_{j;j}^{(t)}{}_0 \quad O(\ )$ through the training. The second property in Lemma 16 guarantees that once $W_{j;j}^{(t)}{}_0$ falls below $\frac{}{\log^{\frac{1}{r-2}}(1=\ )}$; its decreasing rate is so small that that the total decrease within T time is only $\frac{}{2\log^{\frac{1}{r-2}}(1=\ )}$, which then implies that $W_{j;j}^{(t)}{}_0 > \Theta(\ )$ through the training.

Bounding noise correlations: The following lemma shows that the total change of $\langle W_{j;:}^{(t)}; \ _x \rangle$ within T time is only $O(\sqrt{\log N}\ )$: Since at initialization, we know $\langle W_{j;:}^{(0)}; \ _x \rangle \quad O(\sqrt{\log N}\ )$; we conclude that $\langle W_{j;:}^{(t)}; \ _x \rangle \quad O(\sqrt{\log N}\ )$ throughout the training. Since $W_{j;j}^{(t)}{}_0 \quad \Theta(\ )$; as long as $\quad \Theta(1)$; we also have $\langle W_{j;:}^{(t)}; \ _x \rangle \quad W_{j;j}{}_0$ for $x \ 2 \ S_j{}_0$:

Lemma 17. *For every $j \ 2 \ [k]$ and every $x \ 2 \ S$; we have*

$$\langle W_{j;:}; \ _x \rangle \quad T \quad O\left(\sqrt{\log N}\right)$$

C.1 Proof of lemmas

Lemma 10 (Initialization). *With probability at least 0.99 in the initialization, we have*

1. *for all $j; j^0 \ 2 \ [k]$, $W_{j;j}^{(0)}{}_0 = (\ )$;*
2. *for all distinct $j; j^0 \ 2 \ [k]$, $W_{j;j}^{(0)} \quad W_{j^0;j^0}^{(0)}{}_0 = (\ )$;*
3. *for all $x \ 2 \ S; k \ _x k \quad O(\ )$;*
4. *for all distinct $x; x^0 \ 2 \ S$; $\ _x; \ _{x^0} \quad O\left(\frac{\sqrt{\log(N)}}{p\ d}\right)$:*
5. *for all $j \ 2 \ [k]$ and all $x \ 2 \ S$; $\ _x; e_j \ ; \ _x; W_{j;:}^{(0)} \quad O\left(\frac{\sqrt{\log(N)}}{p\ d}\right)$:*

Without loss of generality, we assume $W_{1;1}^{(0)} > W_{2;2}^{(0)} > \quad > W_{k;k}^{(0)}$:

Proof of Lemma 10. Recall that each $W_{j;j}^{(0)}{}_0$ is independently sampled from $\mathcal{N}(0; \ ^2)$ before taking the absolute value. By standard Gaussian concentration inequality, we know for any $j; j^0 2 \ [k]$; with probability at least $1 \quad \frac{1}{1000k^2}$;

$$W_{j;j}^{(0)}{}_0 \quad O(\ ):$$

By anti-concentration inequality of Gaussian polynomials, we know for any $j; j^0 2 \ [k]$; with probability at least $1 \quad \frac{1}{1000k^2}$;

$$W_{j;j}^{(0)}{}_0 \quad (\ ):$$

Also by anti-concentration inequality of Gaussian polynomials, we know for any distinct $j; j^0 2 \ [k]$; with probability at least $1 \quad \frac{1}{1000k^2}$;

$$\left[W_{j;j}^{(0)}\right]^2 \quad \left[W_{j^0;j^0}^{(0)}{}_0\right]^2 \quad (\ ^2);$$

which implies $W_{j;j}^{(0)} \quad W_{j^0;j^0}^{(0)}{}_0 \quad (\ )$ assuming $W_{j;j}^{(0)}; W_{j^0;j^0}^{(0)}{}_0 = (\ )$:

By the norm concentration of random vectors with independent Gaussian entries, for each $x \in S$, we have with probability at least $1 - \frac{1}{1000N^2}$,

$$\|x\| = O(\sqrt{d})$$

as long as $d = O(\log N)$.

By the concentration of standard Gaussian variable, for any distinct $x, x' \in S$, we have with probability at least $1 - \frac{1}{1000N^2}$,

$$\langle x, x' \rangle = O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right).$$

Similarly, for any x and any e_j, we have with probability at least $1 - \frac{1}{1000kN}$,

$$\langle x, e_j \rangle = O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right);$$

for any x and any $W_{j,:}^{(0)}$, we have with probability at least $1 - \frac{1}{1000kN}$,

$$\left\langle x, W_{j,:}^{(0)} \right\rangle = O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right);$$

Taking a union bound over all these events, we know with probability at least 0.99 in the initialization, we have

1. for all $j, j' \in [k]$, $W_{i,j}^{(0)} = (\quad)$;
2. for all distinct $j, j' \in [k]$, $W_{j,j}^{(0)} \quad W_{j',j'}^{(0)} = (\quad)$;
3. for all $x \in S$, $\|x\| = O(\sqrt{})$;
4. for all distinct $x, x' \in S$, $\langle x, x' \rangle = O\left(\frac{\sqrt{\log(N)}}{\sqrt{d}}\right)$;
5. for all $j \in [k]$ and all $x \in S$, $\langle e_j, x \rangle \quad \left\langle x, W_{j,:}^{(0)} \right\rangle = O\left(\frac{\sqrt{\log(N)}}{\sqrt{d}}\right)$.

Lemma 12 (Adapted from Lemma C.19 in Allen-Zhu & Li (2020)) Let $r \quad 3$ be a constant and let $\{W_{i,i}^{(t)}, W_{j,j}^{(t)}\}_{t \quad 0}$ be two positive sequences updated as

$$\frac{\partial W_{i,i}^{(t)}}{\partial t} \quad C_t \left[W_{i,i}^{(t)}\right]^{r-1} \text{ for some } C_t = (1);$$
$$\frac{\partial W_{j,j}^{(t)}}{\partial t} \quad S C_t \left[W_{j,j}^{(t)}\right]^{r-1} \text{ for some } S = (1);$$

Suppose $W_{i,i}^{(0)} \quad W_{j,j}^{(0)} \quad S^{\frac{1}{r-2}}(1 + (1))$; then we must have for every $A = O(1)$, let t_i be the first time such that $W_{i,i}^{(t_i)} \quad A$, then

$$W_{j,j}^{(t_i)} \quad O(W_{j,j}^{(0)}).$$

Proof of Lemma 12. This lemma directly follows from Lemma C.19 in Allen-Zhu & Li (2020) by taking the continuous time limit and setting k as a constant.

Lemma 1 (Coupling Biases) Assuming $W_{j',j'}, W_{j,j} \quad O(\quad)$ and $b_{j'}, b_j \quad \max_{i' \in [k]} b_{i'} \quad O(\quad^r)$, we have $b_{j'} \quad b_j > 0$ if $b_{j'} \quad b_j \quad \quad^r$, and $b_{j'} \quad b_j < 0$ if $b_{j'} \quad b_j \quad + \quad^r$ for some positive constant $$:

Proof of Lemma 1. Let's first write down the time derivative of b_{j0}:

$$\dot{b}_{j0} = -\frac{k}{N}\sum_{x\in S} u_{j0}(x)$$
$$= -\frac{k}{N}\sum_{x\in S} \frac{\exp(\langle w_{j0,:}, x\rangle^r + b_{j0})}{\sum_{i0\in[k]}\exp(\langle w_{i0,:}, x\rangle^r + b_{i0})}$$

For any $x \in S$, we can bound $\frac{\exp(\langle w_{j0,:}, x\rangle^r + b_{j0})}{\sum_{i0\in[k]}\exp(\langle w_{i0,:}, x\rangle^r + b_{i0})}$ as follows,

$$\frac{\exp(\langle w_{j0,:}, x\rangle^r + b_{j0})}{\sum_{i0\in[k]}\exp(\langle w_{i0,:}, x\rangle^r + b_{i0})} \le \frac{\exp(b_{j0})}{\sum_{i0\in[k]}\exp(\langle w_{i0,:}, x\rangle^r + b_{i0})} \le O(\epsilon^r),$$

where we used $|\langle w_{j0,:}, x\rangle| \le O(\epsilon) + O(\sqrt{\log N}) \le O(\epsilon)$ assuming $1/\epsilon \ge \sqrt{\log N}$. The similar bound also holds for $\frac{\exp(\langle w_{j,:}, x\rangle^r + b_j)}{\sum_{i0\in[k]}\exp(\langle w_{i0,:}, x\rangle^r + b_{i0})}$.

If $b_{j0} - b_j \ge \epsilon^r$, we can now upper bound $\dot{b}_{j0} - \dot{b}_j$ as follows,

$$\dot{b}_{j0} - \dot{b}_j \le -\frac{k}{N}\sum_{x\in S} \frac{\exp(b_j) - \exp(b_{j0})}{\sum_{i0\in[k]}\exp(\langle w_{i0,:}, x\rangle^r + b_{i0})} + O(\epsilon^r)$$
$$\le -\frac{k}{N}\sum_{x\in S_j\cap S_{j0}} \frac{\exp(b_j) - \exp(b_{j0})}{\sum_{i0\in[k]}\exp(\langle w_{i0,:}, x\rangle^r + b_{i0})} + O(\epsilon^r)$$
$$\le (-\epsilon^r)\frac{k}{N}\sum_{x\in S_j\cap S_{j0}} \frac{\exp(b_j)}{\sum_{i0\in[k]}\exp(\langle w_{i0,:}, x\rangle^r + b_{i0})} + O(\epsilon^r)$$

When $x \in S_j \setminus S_{j0}$, we can lower bound $\frac{\exp(b_j)}{\sum_{i0\in[k]}\exp(\langle w_{i0,:}, x\rangle^r + b_{i0})}$ as follows,

$$\frac{\exp(b_j)}{\sum_{i0\in[k]}\exp(\langle w_{i0,:}, x\rangle^r + b_{i0})} = \frac{\exp(b_j)}{\sum_{i0\in[k]}\exp(b_{i0})\exp(\langle w_{i0,:}, x\rangle^r)}$$
$$\ge \frac{\exp(b_j)}{\sum_{i0\in[k]}\exp(b_{i0})} \cdot \frac{1}{1+O(\epsilon^r)}$$
$$\ge (1-\delta);$$

where the first inequality uses $|\langle w_{i0,:}, x\rangle| \le \delta$ and the second inequality assumes $\max_{i0\in[k]} b_{i0} \le O(\epsilon^r)$ and δ is at most some small constant.

Therefore, if $b_{j0} - b_j \ge \epsilon^r$, we have

$$\dot{b}_{j0} - \dot{b}_j \le (-\epsilon^r)(1-\delta) + O(\epsilon^r) < 0,$$

where the second inequality chooses δ as a large enough constant. Similarly, we can prove that if $b_{j0} - b_j \le -\epsilon^r$, we have

$$\dot{b}_{j0} - \dot{b}_j \ge (-\epsilon^r)(1+\delta) - O(\epsilon^r) > 0.$$

Lemma 2 (Bias Gap Control II). For any different $j0, j \in [k]$, if $\|w_{j0,:} - w_{j,:}\| \le O(\epsilon)$ and $b_{j0} - b_j \le O(\epsilon^r)$, $b_j \ge \max_{i0\in[k]} b_{i0} - O(\epsilon^r)$, we have $\dot{b}_{j0} - \dot{b}_j < 0$.

Proof of Lemma 2. We can write down $\dot{b}_{j0} - \dot{b}_j$ as follows,

$$\dot{b}_{j0} - \dot{b}_j = \left(1 - \frac{k}{N}\sum_{x\in S} u_{j0}(x)\right) - \left(1 - \frac{k}{N}\sum_{x\in S} u_j(x)\right)$$
$$= \frac{k}{N}\sum_{x\in S_{j0}} (u_j(x) - u_{j0}(x)) + \frac{k}{N}\sum_{x\in S\setminus S_{j0}} (u_j(x) - u_{j0}(x)).$$

We first prove that for any $x \in S_{j'}$, we have $u_j(x) - u_{j'}(x) < 0$. We can upper bound $f_j(x)$ and lower bound $f_{j'}(x)$ as follows,

$$f_j(x) = \langle W_{j,:}, x \rangle^r + b_j < O(\epsilon^r) + b_j$$
$$f_{j'}(x) = \langle W_{j',:}, x \rangle^r + b_{j'} > b_{j'}.$$

The bound on $f_j(x)$ holds because $\langle W_{j,:}, x \rangle = W_{j,j'} + \langle W_{j,:}, x \rangle < O(\epsilon) + O(\epsilon\sqrt{\log N}) < O(\epsilon)$. The bound on $f_{j'}(x)$ holds because $\langle W_{j',:}, x \rangle = W_{j',j'} + \langle W_{j',:}, x \rangle > -O(\epsilon) - O(\epsilon\sqrt{\log N}) > -O(\epsilon)$. With the above two bounds, we know that $u_j(x) - u_{j'}(x) < 0$ as long as $b_{j'} - b_j < O(\epsilon^r)$.

Same as in the proof of Lemma 1, for each $x \in S \cap S_{j'}$, we can bound $u_{j'}(x), u_j(x)$ as follows,

$$\frac{\exp(b_{j'})}{\sum_{i' \in [k]} \exp(f_{i'}(x))} - O(\epsilon^r) < u_{j'}(x) < \frac{\exp(b_{j'})}{\sum_{i' \in [k]} \exp(f_{i'}(x))} + O(\epsilon^r);$$
$$\frac{\exp(b_j)}{\sum_{i' \in [k]} \exp(f_{i'}(x))} - O(\epsilon^r) < u_j(x) < \frac{\exp(b_j)}{\sum_{i' \in [k]} \exp(f_{i'}(x))} + O(\epsilon^r).$$

Therefore, if $b_{j'} - b_j > \epsilon^r$, we can further upper bound $b_{j'} - b_j$ as follows,

$$\begin{aligned} b_{j'} - b_j &< \frac{k}{N} \sum_{x \in S \cap S_{j'}} (u_j(x) - u_{j'}(x)) \\ &< \frac{k}{N} \sum_{x \in S \cap S_{j'}} \frac{\exp(b_j) - \exp(b_{j'})}{\sum_{i' \in [k]} \exp(f_{i'}(x))} + O(\epsilon^r) \\ &< \frac{k}{N} \sum_{x \in S_j} \frac{\exp(b_j) - \exp(b_{j'})}{\sum_{i' \in [k]} \exp(f_{i'}(x))} + O(\epsilon^r) \\ &< (-\epsilon^r) \frac{k}{N} \sum_{x \in S_j} \frac{\exp(b_j)}{\sum_{i' \in [k]} \exp(f_{i'}(x))} + O(\epsilon^r). \end{aligned}$$

Similar as in Lemma 1, we can show that $\frac{\exp(b_j)}{\sum_{i' \in [k]} \exp(f_{i'}(x))} \geq \Omega(1)$ due to $W_{j,j} \geq O(\epsilon)$ and $b_j \geq \max_{i' \in [k]} b_{i'} - O(\epsilon^r)$. So, finally we have

$$b_{j'} - b_j < (-\epsilon^r) + O(\epsilon^r) < 0,$$

where the last inequality chooses ϵ as a large enough constant.

Lemma 15 (Accuracy Monotonicity.) Given any positive constant C_2, there exists positive constant C_1 such that for all different $i, j \in [k]$, as long as $W_{i,i} \geq C_1$ and $f_i(x) - f_j(x) \leq C_2$ for any $x \in S_i$, we have $\frac{\partial(f_i(x) - f_j(x))}{\partial t} > 0$.

Proof of Lemma 15. Since $f_i(x) - f_j(x) \leq C_2$, we know $1 - u_i(x) \geq \Omega(1)$. This immediately implies $\min_{x' \in S_i}(1 - u_i(x')) \geq \Omega(1)$ since $|u_i(x) - u_i(x')| \leq O(\epsilon)$. According to Lemma 13, we can bound $\frac{\partial(f_i(e_i) - f_j(e_i))}{\partial t}$ as follows,

$$\frac{\partial(f_i(x) - f_j(x))}{\partial t} \geq \Omega(W_{i,i}^{2r-2}) \geq \Omega(1) > 0$$

where the second inequality holds because we set C_1 with C_1 a large enough constant.

Lemma 3 (Bias Gap Control II.) There exist small positive constants C_1, C_2 such that for any $j \in [k-1]$ and any $x \in S_j$, if $1 - u_j(x) \leq C_1$, $W_{k,k} \leq O(\epsilon)$ and $b_j - b_k \leq C_2$, we have $b_j - b_k < -\Omega(1)$.

Proof of Lemma 3. Since $1 - u_j(x) \leq C_1$ for some $x \in S_j$, we know $1 - u_j(x') \leq C_1 + O(\epsilon)$ for every $x' \in S_j$. We can write down $b_j - b_k$ as follows,

$$\begin{aligned} b_j - b_k &= \left(1 - \frac{k}{N} \sum_{x' \in S} u_j(x')\right) - \left(1 - \frac{k}{N} \sum_{x' \in S} u_k(x')\right) \\ &= \frac{k}{N} \sum_{x' \in S_j} (u_k(x') - u_j(x')) + \frac{k}{N} \sum_{x' \in S \cap S_j} (u_k(x') - u_j(x')). \end{aligned}$$

First, we upper bound $u_k(x') - u_j(x')$ for every $x' \in S_j$ as follows,

$$u_k(x') - u_j(x') \le 1 + u_j(x') - u_j(x') = 1 + 2(1 - u_j(x')) \le 2C_1 + 1 + O(\epsilon).$$

Same as in the proof of Lemma 1, for each $x' \in S \cap S_j$, we can bound $u_j(x'), u_k(x')$ as follows,

$$\frac{\exp(b_j)}{\sum_{i' \in [k]} \exp(f_{i'}(x'))} - O(\epsilon) \le u_j(x') \le \frac{\exp(b_j)}{\sum_{i' \in [k]} \exp(f_{i'}(x'))} + O(\epsilon);$$
$$\frac{\exp(b_k)}{\sum_{i' \in [k]} \exp(f_{i'}(x'))} - O(\epsilon) \le u_k(x') \le \frac{\exp(b_k)}{\sum_{i' \in [k]} \exp(f_{i'}(x'))} + O(\epsilon).$$

Therefore, we can upper bound $u_k(x') - u_j(x')$ as follows,

$$\begin{aligned} u_k(x') - u_j(x') &\le \frac{\exp(b_k) - \exp(b_j)}{\sum_{i' \in [k]} \exp(f_{i'}(x'))} + O(\epsilon) \\ &= \frac{\exp(b_j)}{\sum_{i' \in [k]} \exp(f_{i'}(x'))} (\exp(b_k - b_j) - 1) + O(\epsilon) \\ &\le O(C_2) + O(\epsilon); \end{aligned}$$

where the last inequality uses $b_k - b_j \le C_2$.

Above all, we can upper bound $b_j - b_k$ as follows,

$$\begin{aligned} b_j - b_k &\le 1 + 2C_1 + O(C_2) + O(\epsilon) \\ &< (1); \end{aligned}$$

where the second inequality holds as long as C_1, C_2, ϵ are at most some small constants.

Lemma 14 (Bias Gap Control III). For any different $i, j \in [k]$, if $W_{i,i} \le O(1)$, $W_{j,j} \ge O(\epsilon)$ and $b_i - b_j \le O(1)$, we have

$$b_i - b_j > 0.$$

Proof of Lemma 14. We can write down $b_i - b_j$ as follows,

$$\begin{aligned} b_i - b_j &= \left(1 - \frac{k}{N} \sum_{x \in S} u_i(x)\right) - \left(1 - \frac{k}{N} \sum_{x \in S} u_j(x)\right) \\ &= \frac{k}{N} \sum_{x \in S} (u_j(x) - u_i(x)) \end{aligned}$$

Next, we lower bound $u_j(x) - u_i(x)$ for every $x \in S$,

$$\begin{aligned} u_j(x) - u_i(x) = &\frac{\exp(\langle W_{j,:}, x_i \rangle + b_j) - \exp(\langle W_{i,:}, x_i \rangle + b_i)}{\sum_{i' \in [k]} f_{i'}(x)} \\ &\frac{\exp(O(\epsilon) + b_j) - \exp(O(1) + b_i)}{\sum_{i' \in [k]} f_{i'}(x)} \end{aligned}$$

So as long as $b_j - b_i > O(1)$, we have $u_j(x) - u_i(x) > 0$ for all $x \in S$, which then implies $b_i - b_j > 0$.

Lemma 17. For every $j \in [k]$ and every $x \in S$, we have

$$\langle W_{j,:}, x \rangle \le O\left(\sqrt{\log N}\right)$$

Proof of Lemma 17. For each $j \in [k]$, we have

$$W_{j,:} = \frac{k}{N} \left(\sum_{x' \in S_j} (1 - u_j(x')) \cdot \langle W_{j,:}, x' \rangle \cdot x' - \sum_{x' \in S \setminus S_j} u_j(x') \cdot \langle W_{j,:}, x' \rangle \cdot x' \right)$$

and

$$\left\langle W_{j,:}, x \right\rangle = \frac{k}{N}\left(\sum_{x' \in S_j} (1 - u_j(x')) \, r \langle W_{j,:}, x' \rangle^{r-1} x', x + \sum_{x' \in S \setminus S_j} u_j(x') r \langle W_{j,:}, x' \rangle^{r-1} x', x\right)$$

We know that $\langle x', x \rangle \leq O(\sqrt{\log N}/\sqrt{d})$. For any $x' \in S_j$, we have $\langle x', x \rangle \leq O((\sqrt{\log N}/\sqrt{d}) + \sqrt{\log N}/\sqrt{d}) \leq O(\sqrt{\log N}/\sqrt{d})$ as long as < 1.

According to Lemma 18, we know that for $x' \in S_j$, we have $(1 - u_j(x')) \langle W_{j,:}, x' \rangle^{r-1} \leq O(1)$.

For $x' \in S \setminus S_i$, we have $u_i(x') \langle W_{j,:}, x' \rangle^{r-1} \leq O(^{r-1})$ since $|\langle W_{j,:}, x' \rangle| \leq O(\cdot) + O(\sqrt{\log N}) \leq O(\cdot)$ assuming $\leq 1/\sqrt{\log N}$.

Therefore, we can bound $\left\langle W_{j,:}, x \right\rangle$ as follows,

$$\left\langle W_{j,:}, x \right\rangle \leq O\left(\frac{1}{N} + \frac{\sqrt{\log N}}{\sqrt{d}}\right)$$

Since $T \geq O(\log(1/\epsilon)/^{r-2})$, $N \geq \log(1/\epsilon)/^{r-1}$ and $d \geq \log^2(1/\epsilon)/^{2r-2}$, we know

$$\left\langle W_{j,:}, x \right\rangle \leq T \leq O(\sqrt{\log N}).$$

Lemma 18. For any $i \in [k]$ and $x \in S_i$, if $(1 - u_i(x)) \langle W_{i,:}, x \rangle^{r-1} \leq (1)$, we have

$$\frac{d}{dt}\left((1 - u_i(x)) \langle W_{i,:}, x \rangle^{r-1}\right) < 0.$$

Proof of Lemma 18. We can write $1 - u_i(x)$ as $\frac{\sum_{j \in [k], j \neq i} \exp(f_j(x))}{\sum_{j \in [k], j \neq i} \exp(f_j(x)) + \exp(-f_i(x))}$. Next, we prove that for any $j' \neq i$, we have

$$\frac{d}{dt}\left(\frac{\exp(f_{j'}(x))}{\sum_{j \in [k], j \neq i} \exp(f_j(x)) + \exp(-f_i(x))} \langle W_{i,:}, x \rangle^{r-1}\right) < 0.$$

This derivative can be written the sum of two terms:

$$\begin{aligned}
&\frac{d}{dt}\left(\frac{\exp(f_{j'}(x))}{\sum_{j \in [k], j \neq i} \exp(f_j(x)) + \exp(-f_i(x))} \langle W_{i,:}, x \rangle^{r-1}\right) \\
&= \sum_{j \in [k], j \neq i} \frac{1}{\exp(f_j(x) - f_{j'}(x)) + \exp(-f_i(x) - f_{j'}(x))} \frac{d}{dt} \langle W_{i,:}, x \rangle^{r-1} \\
&\quad + \frac{d}{dt}\left(\sum_{j \in [k], j \neq i} \frac{1}{\exp(f_j(x) - f_{j'}(x)) + \exp(-f_i(x) - f_{j'}(x))}\right) \langle W_{i,:}, x \rangle^{r-1}.
\end{aligned}$$

For the first term, we have

$$\begin{aligned}
&\sum_{j \in [k], j \neq i} \frac{1}{\exp(f_j(x) - f_{j'}(x)) + \exp(-f_i(x) - f_{j'}(x))} \frac{d}{dt} \langle W_{i,:}, x \rangle^{r-1} \\
&= \sum_{j \in [k], j \neq i} \frac{1}{\exp(f_j(x) - f_{j'}(x)) + \exp(-f_i(x) - f_{j'}(x))} (r-1) \langle W_{i,:}, x \rangle^{r-2} \left\langle W_{i,:}, x \right\rangle \\
&\quad \frac{1}{\exp(f_i(x) - f_{j'}(x))} (r-1) \langle W_{i,:}, x \rangle^{r-2} \left\langle W_{i,:}, x \right\rangle.
\end{aligned}$$

For the second term, we have

$$\frac{d}{dt}\left(\frac{1}{\sum_{j\in[k],j\neq i}\exp(f_j(x)-f_{j^*}(x))+\exp(f_i(x)-f_{j^*}(x))}\right)\langle W_{i,:},x\rangle^{r-1}$$
$$=-\frac{\sum_{j\in[k],j\neq i}\exp(f_j(x)-f_{j^*}(x))\,\langle\nabla f_j(x)-\nabla f_{j^*}(x),-\nabla f_{j^*}(x)\rangle+\exp(f_i(x)-f_{j^*}(x))\,\langle\nabla f_i(x)-\nabla f_{j^*}(x),-\nabla f_{j^*}(x)\rangle}{\left(\sum_{j\in[k],j\neq i}\exp(f_j(x)-f_{j^*}(x))+\exp(f_i(x)-f_{j^*}(x))\right)^2}\langle W_{i,:},x\rangle^{r-1}$$
$$\leq\frac{1}{2}\,r\langle W_{i,:},x\rangle^{r-1}\left\langle W_{i,:},x\right\rangle\frac{\exp(f_i(x)-f_{j^*}(x))}{\exp(f_i(x)-f_{j^*}(x))}\langle W_{i,:},x\rangle^{r-1},$$

where the last inequality uses $\|\nabla f_j(x)\|_2 \leq (1)$, $\|\nabla f_{j^*}(x)\|_2 \leq O(1)$ and $\langle\nabla f_{j^*}(x), r\langle W_{i,:},x\rangle^{r-1}W_{i,:}\rangle \leq O(1)\cdot(1)$.

Combining the bounds on both terms, as long as $\langle W_{i,:},x\rangle$ is larger than certain constant (which is guaranteed by $(1-u_i(x))\langle W_{i,:},x\rangle^{r-1} \geq (1)$), we know $\frac{d}{dt}(1-u_i(x))\langle W_{i,:},x\rangle^{r-1} < 0$.

Lemma 16. For any $j \neq j^*$, we have $\|W_{j,0}\| \leq O(\ )$. Furthermore, there exists absolute constant >0 such that if $0 < \|W_{j,0}\| < \frac{}{\log^{\frac{1}{r-2}}(1/\ )}$, we have $\|W_{j,0}\| \leq \frac{}{2\log^{\frac{1}{r-2}}(1/\ )}$.

Proof of Lemma 16. We can write down the derivative of $W_{j,0}$ as follows,

$$-W_{j,0} = \frac{k}{N}\sum_{x\in S_j}(1-u_j(x))\,r\langle W_{j,:},x\rangle^{r-1}h_{e_{j^*},x}$$
$$-\frac{k}{N}\sum_{x\in S_{j^*}}u_j(x)\,r\langle W_{j,:},x\rangle^{r-1}h_{e_{j^*},x}$$
$$-\frac{k}{N}\sum_{x\in S\setminus(S_j\cup S_{j^*})}u_j(x)\,r\langle W_{j,:},x\rangle^{r-1}h_{e_{j^*},x}$$
$$= O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right) - O\left(\|W_{j,0}\|\right) - O\left(\sqrt{\log N}\right)^{r-1} - O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right)^{r-1}.$$

The bound on the first term relies on $(1-u_j(x))\langle W_{j,:},x\rangle^{r-1} \leq O(1)$ and $h_{e_{j^*},x} = O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right)$ for $x \in S_j$; where $(1-u_j(x))\langle W_{j,:},x\rangle^{r-1} \leq O(1)$ is guaranteed by Lemma 18. The bound on the second term uses $\langle W_{j,:},x\rangle = \|W_{j,0}\| \cdot O\left(\sqrt{\log N}\right)$ and $h_{e_{j^*},x} = 1 \cdot O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right)$ for $x \in S_{j^*}$. The bound on the third term uses $\langle W_{j,:},x\rangle = O(\ )$ and $h_{e_{j^*},x} = O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right)$ for $x \in S\setminus(S_j\cup S_{j^*})$.

To prove the upper bound of the derivative, we have

$$-\|W_{j,0}\| \leq O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right)$$

where we use $\|W_{j,0}\| \geq O\left(\sqrt{\log N}\right)$ 0. Since $T = O(\log(1/\ )/\ ^{r-2})$, we have

$$T\|W_{j,0}\| \leq O(\ );$$

as long as $d \geq O\left(\frac{\log N\log^2(1/\ )}{2r\ ^{r-2}}\right)$.

We show that there exists absolute constant > 0 such that if $0 < W_{j;j}^{(0)} < \frac{}{\log^{\frac{1}{r-2}}(1/)}$; we have

$TW_{j;j}^{(0)} \quad \frac{}{2\log^{\frac{1}{r-2}}(1/)}$; which holds as long as $W_{j;j}^{(0)} \quad O\left(\frac{}{\log^{\frac{r-1}{r-2}}(1/)}\right)^{r-1}$: We have

$$\begin{aligned} W_{j;j}^{(0)} &= O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right) \quad O\left(W_{j;j}^{(0)} \quad O\left(\sqrt{\log N}\right)^{r-1}\right) \\ &\quad O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right) \quad O\left(\frac{}{\log^{\frac{r-1}{r-2}}(1/)}\right)^{r-1} \\ &\quad O\left(\frac{}{\log^{\frac{r-1}{r-2}}(1/)}\right)^{r-1} \\ &\quad O\left(\frac{}{\log^{\frac{r-1}{r-2}}(1/)}\right)^{r-1} : \end{aligned}$$

The first inequality assumes $O\left(\frac{}{\sqrt{\log N}\log^{\frac{1}{r-2}}(1/)}\right)$: The second inequality assumes $O\left(\frac{\log N \log^{\frac{2r-2}{r-2}}(1/)}{2r-2\ 2r-2}\right)$: The third inequality chooses as a small enough constant.

Lemma 11. For any $j \in [k]$; we have

$$\frac{@W_{j;j}^{(t)}}{@t} \quad O\left(\frac{}{\sqrt{d}}\right)^{r-1}\sqrt{\log N} :$$

If $\min_{x\in S_j}(1 \quad u_j(x)) \quad (1)$, we further have

$$1 \quad O\left(\sqrt{\log N}\right)\frac{k}{N}\sum_{x\in S_j}(1 \quad u_j(x))\,rW_{j;j}^{r-1} \quad \frac{@W_{j;j}^{(t)}}{@t} \quad 1 + O\left(\sqrt{\log N}\right)\frac{k}{N}\sum_{x\in S_j}(1 \quad u_j(x))\,rW_{j;j}^{r-1}:$$

Proof of Lemma 11. We have

$$\begin{aligned} W_{j;j} &= \frac{k}{N}\sum_{x\in S_j}(1 \quad u_j(x))\,r\,h W_{j;:};x i^{r-1}hx;e_j i \quad \frac{k}{N}\sum_{x\in S\setminus S_j} u_j(x)r\,h W_{j;:};x i^{r-1}hx;e_j i \\ &= \frac{k}{N}\sum_{x\in S_j}(1 \quad u_j(x))\,r \quad W_{j;j} \quad O\left(\sqrt{\log N}\right)^{r-1}\left(1 \quad O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right)\right) \\ &\quad \frac{k}{N}\sum_{x\in S\setminus S_j} u_j(x)r \quad O(\ ) \quad O\left(\sqrt{\log N}\right)^{r-1} \quad O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right) \\ &= 1 \quad O\left(\sqrt{\log N}\right)\frac{k}{N}\sum_{x\in S_j}(1 \quad u_j(x))\,rW_{j;j}^{r-1} \quad O\left(\frac{}{\sqrt{d}}\right)^{r-1}\sqrt{\log N} ; \end{aligned}$$

where the second equality uses $jhW_{j;:};\ _x ij \quad O\left(\sqrt{\log N}\right); jh\ _x;e_j ij \quad O\left(\frac{\sqrt{\log N}}{\sqrt{d}}\right)$ and $W_{j;j}^{(0)} \quad O(\ )$ for $j \in j^0$.

Therefore, if $\min_{x\in S_j}(1 \quad u_j(x)) \quad (1)$, we know

$$1 \quad O\left(\sqrt{\log N}\right)\frac{k}{N}\sum_{x\in S_j}(1 \quad u_j(x))\,rW_{j;j}^{r-1} \quad W_{j;j} \quad 1 + O\left(\sqrt{\log N}\right)\frac{k}{N}\sum_{x\in S_j}(1 \quad u_j(x))\,rW_{j;j}^{r-1}:$$

And we always have

$$W_{j;j} \quad O\left(\frac{}{\sqrt{d}}\right)^{r-1}\sqrt{\log N} :$$

Lemma 13. For any $x \in S_i$ and any $j \neq i$, if $1 - u_i(x) \geq (1)$, we have

$$\frac{\partial}{\partial t}(f_i(x) - f_j(x)) \geq (W_{i,i}^{2r-2}) - O(1).$$

Proof of Lemma 13. Recall that $f_i(x) = \langle W_{i,:}, x\rangle^r + b_i$, so we have

$$\begin{aligned} f_i'(x) &= r\langle W_{i,:}, x\rangle^{r-1}\langle W_{i,:}', x\rangle + b_i' \\ &= r W_{i,i}\sqrt{\log N}^{r-1} W_{i,i}' + \langle W_{i,:}', x\rangle + b_i' \\ &\geq (W_{i,i}^{2r-2}) - O(1), \end{aligned}$$

where in the last inequality we used $W_{i,i} \geq (W_{i,i}^{r-1}) \geq (\ ^r)$ and $\langle W_{i,:}', x\rangle \geq -\frac{\sqrt{\log N}^{r-1}}{\log(1/\ )} = O(\ ^{r-1})$.

We also have

$$\begin{aligned} f_j'(x) &= r\langle W_{j,:}, x\rangle^{r-1}\langle W_{j,:}', x\rangle + b_j' \\ &\leq r W_{j,i}\sqrt{\log N}^{r-1} W_{j,i}' + \langle W_{j,:}', x\rangle + b_j' \\ &\leq O(1), \end{aligned}$$

where we used $|W_{j,i}| \leq O(\ )$, $W_{j,i}' \leq O(\ ^{r-1})$ and $\langle W_{j,:}', x\rangle \leq O(\ ^{r-1})$.

Therefore, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}(f_i(x) - f_j(x)) \geq (W_{i,i}^{2r-2}) - O(1).$$

D Additional experiments

In this section, we describe the detailed setting of our experiments and also include additional experiment results.

MNIST & Fashion-MNIST. Unless specified otherwise, we use a depth-10 and width-1024 fully-connected ReLU neural network (FCN10) for MNIST and Fashion-MNIST. We use Kaiming initialization for the weights and set all bias terms as zero. We use a small initialization by scaling the weights of each layer by $(0.001)^{1/h}$ so the output is scaled by 0.001, where h is the network depth. We train the network using SGD with learning rate 0.01 and momentum 0.9 for 100 epochs.

CIFAR-10 & CIFAR-100 We use VGG-16 (without batch normalization) for CIFAR-10 and CIFAR-100. We use Kaiming initialization for the weights and set all bias terms as zero. We run SGD with momentum 0.9 and weight decay $1e-4$ for 100 epochs. For the learning rate, we start from 0.01 and reduce it by a factor of 0.1 at the 60-th epoch and 90-th epoch.

We linearly interpolate using 50 evenly spaced points between the network at initialization and the network at the end of training. We evaluate error and loss on the train set. For each setting, we repeat the experiments three times from different random seeds and plot the mean and deviation.

Note in Figure 1, to contrast the convex curve and plateau curve, we have used FCN4 with standard initialization on MNIST, and VGG-16 with 0.001 initialization on CIFAR-10.

Our code is based on the implementation from Lucas et al. (2021). Each trial of our experiment can be finished on an Nvidia Tesla P100 within one hour.

D.1 ALL BIAS V.S. LAST BIAS V.S. NO BIAS

Figure 9: Comparison between networks with all bias, last bias and no bias on Fashion-MNIST and CIFAR-10.

Figure 9 shows that on both Fashion-MNIST and CIFAR-10, having bias on the last layer or on all layers can create longer plateau in error curve, while does not signi cantly affect the loss curve.

D.2 NORMAL INTERPOLATION V.S. HOMOGENEOUS INTERPOLATION

Figure 10: Comparison between networks with normal interpolation and homogeneous interpolation on bias on Fashion-MNIST and CIFAR-10.

Figure 10 shows that on both Fashion-MNIST and CIFAR-10, applying homogeneous interpolation on biases can signi cantly reduce the plateau on error interpolation curve.

D.3 DIFFERENT INITIALIZATIONS

Figure 11: Comparison between networks with different initialization scales on MNIST and CIFAR-100 with last bias.

Figure 12: Comparison between networks with different initialization scales on Fashion-MNIST and CIFAR-10 with all bias.

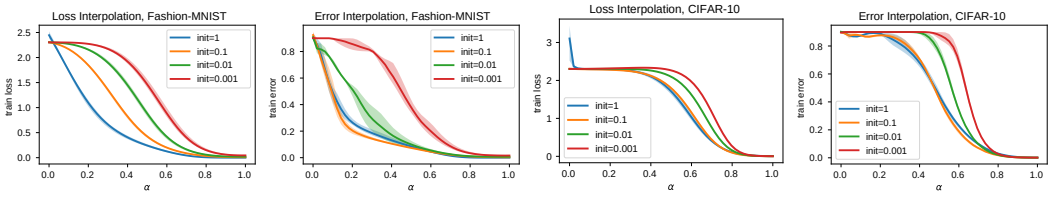

Figure 13: Comparison between networks with different initialization scales on Fashion-MNIST and CIFAR-10 with last bias.

Smaller initialization creates longer plateau in both error and loss curves. See Figure 11 for MNIST, CIFAR-100 with last bias; see Figure 12 for Fashion-MNIST, CIFAR-10 with all bias; see Figure 13 for Fashion-MNIST, CIFAR-10 with last bias.

D.4 Different depths

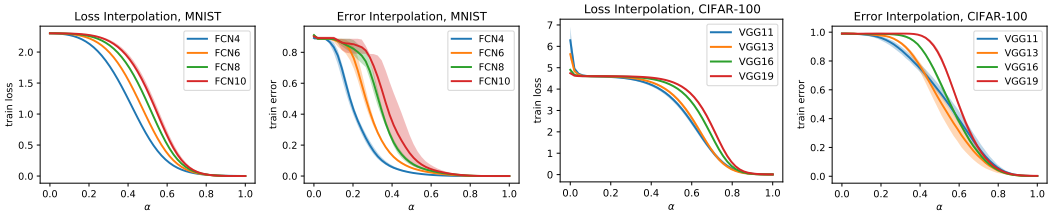

Figure 14: Comparison between networks with different depth on MNIST and CIFAR-100 with last bias.

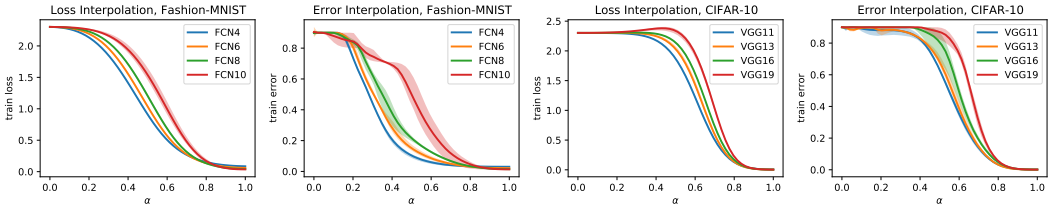

Figure 15: Comparison between networks with different depth on Fashion-MNIST and CIFAR-10 with all bias. We use 0.001 initialization scale for VGG-16 on CIFAR-10.

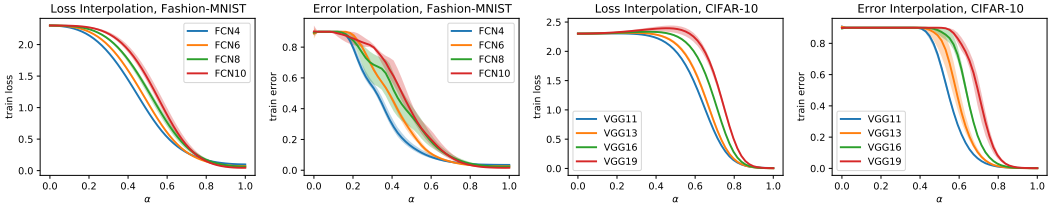

Figure 16: Comparison between networks with different depth on Fashion-MNIST and CIFAR-10 with last bias. We use 0.001 initialization scale for VGG-16 on CIFAR-10.

Deeper networks create longer plateau in both error and loss curves. See Figure 14 for MNIST, CIFAR-100 with last bias; see Figure 15 for Fashion-MNIST, CIFAR-10 with all bias; see Figure 16 for Fashion-MNIST, CIFAR-10 with last bias.

D.5 BIAS DYNAMICS

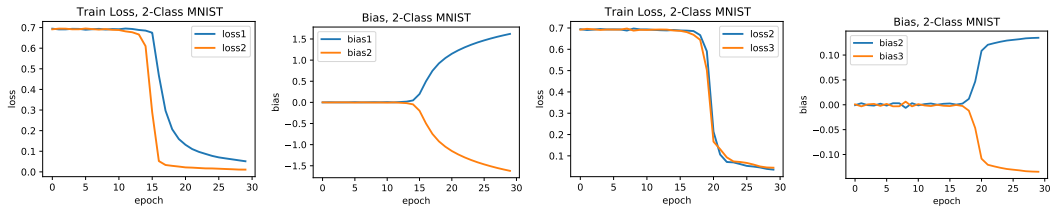

Figure 17: Train loss for each class and bias term dynamics on MNIST $f1; 2g$ and MNIST $f2; 3g$.

In Figure 17, we give two more examples on two-class MNIST in which the later learned class has larger bias.

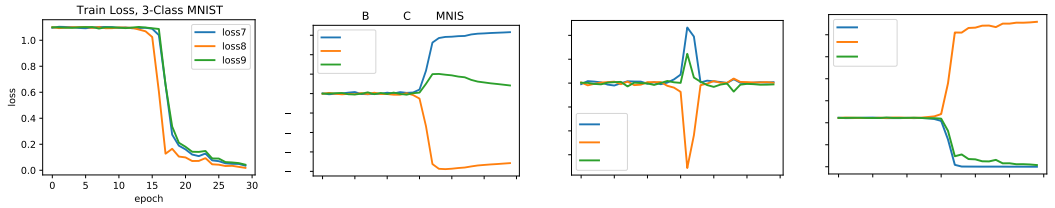

Figure 18: Train loss for each class and bias term dynamics on MNIST $f7; 8; 9g$.

In Figure 18, although class 9 is learned last, class 7 gets the largest bias after training. Let S be the set of all samples for number 7,8,9 and let $S_7; S_9; S_9$ be the set of samples for each class. For convenience, we use $u_{i;j}$ to denote $\frac{1}{|S|} \sum_{x \in S_j} u_i(x)$; where $u_i(x)$ is the softmax output for class i under input x: Then, we can write down the derivative on three bias terms:

$$b_7 = \frac{1}{3} \quad u_{7;7} \quad u_{7;8} \quad u_{7;9}$$

$$b_8 = \frac{1}{3} \quad u_{8;7} \quad u_{8;8} \quad u_{8;9}$$

$$b_9 = \frac{1}{3} \quad u_{9;7} \quad u_{9;8} \quad u_{9;9}:$$

According to the per-class loss, we know that $\sum_{x \in S_7} \log(u_7(x)) < \sum_{x \in S_9} \log(u_9(x))$; which intuitively implies that $\sum_{x \in S_7} u_7(x) > \sum_{x \in S_9} u_9(x)$ that is $u_{7;7} > u_{9;9}$. This tends to drive b_7 smaller than b_9: However, because $u_{9;8} > u_{7;8}$; we actually have $b_9 < b_7$: So eventually b_9 becomes smaller than b_7: Intuitively, class 9 is more correlated with class 8; so $u_{9;8} > u_{7;8}$: