# Learning Overparameterized Neural Networks via Stochastic Gradient Descent on Structured Data

**Yuanzhi Li**
Computer Science Department
Stanford University
Stanford, CA 94305
yuanzhil@stanford.edu

**Yingyu Liang**
Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI 53706
yliang@cs.wisc.edu

## Abstract

Neural networks have many successful applications, while much less theoretical understanding has been gained. Towards bridging this gap, we study the problem of learning a two-layer overparameterized ReLU neural network for multi-class classification via stochastic gradient descent (SGD) from random initialization. In the overparameterized setting, when the data comes from mixtures of well-separated distributions, we prove that SGD learns a network with a small generalization error, albeit the network has enough capacity to fit arbitrary labels. Furthermore, the analysis provides interesting insights into several aspects of learning neural networks and can be verified based on empirical studies on synthetic data and on the MNIST dataset.

## 1 Introduction

Neural networks have achieved great success in many applications, but despite a recent increase of theoretical studies, much remains to be explained. For example, it is empirically observed that learning with stochastic gradient descent (SGD) in the overparameterized setting (i.e., learning a large network with number of parameters larger than the number of training data points) does not lead to overfitting [24, 31]. Some recent studies use the low complexity of the learned solution to explain the generalization, but usually do not explain how the SGD or its variants favors low complexity solutions (i.e., the inductive bias or implicit regularization) [3, 23]. It is also observed that overparameterization and proper random initialization can help the optimization [28, 12, 26, 18], but it is also not well understood why a particular initialization can improve learning. Moreover, most of the existing works trying to explain these phenomenons in general rely on unrealistic assumptions about the data distribution, such as Gaussian-ness and/or linear separability [32, 25, 10, 17, 7].

This paper thus proposes to study the problem of learning a two-layer overparameterized neural network using SGD for classification, on data with a more realistic structure. In particular, the data in each class is a mixture of several components, and components from different classes are well separated in distance (but the components in each class can be close to each other). This is motivated by practical data. For example, on the dataset MNIST [15], each class corresponds to a digit and can have several components corresponding to different writing styles of the digit, and an image in it is a small perturbation of one of the components. On the other hand, images that belong to the same component are closer to each other than to an image of another digit. Analysis in this setting can then help understand how the structure of the practical data affects the optimization and generalization.

In this setting, we prove that when the network is sufficiently overparameterized, SGD provably learns a network close to the random initialization and with a small generalization error. This result shows that in the overparameterized setting and when the data is well structured, though in principle

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

the network can overfit, SGD with random initialization introduces a strong inductive bias and leads to good generalization.

Our result also shows that the overparameterization requirement and the learning time depends on the parameters inherent to the structure of the data but not on the ambient dimension of the data. More importantly, the analysis to obtain the result also provides some interesting theoretical insights for various aspects of learning neural networks. It reveals that the success of learning crucially relies on overparameterization and random initialization. These two combined together lead to a tight coupling around the initialization between the SGD and another learning process that has a benign optimization landscape. This coupling, together with the structure of the data, allows SGD to find a solution that has a low generalization error, while still remains in the aforementioned neighborhood of the initialization. Our work makes a step towrads explaining how overparameterization and random initialization help optimization, and how the inductive bias and good generalization arise from the SGD dynamics on structured data. Some other more technical implications of our analysis will be discussed in later sections, such as the existence of a good solution close to the initialization, and the low-rankness of the weights learned. Complementary empirical studies on synthetic data and on the benchmark dataset MNIST provide positive support for the analysis and insights.

## 2 Related Work

**Generalization of neural networks.** Empirical studies show interesting phenomena about the generalization of neural networks: practical neural networks have the capacity to fit random labels of the training data, yet they still have good generalization when trained on practical data [24, 31, 2]. These networks are overparameterized in that they have more parameters than statistically necessary, and their good generalization cannot be explained by naïvely applying traditional theory. Several lines of work have proposed certain low complexity measures of the learned network and derived generalization bounds to better explain the phenomena. [3, 23, 21] proved spectrally-normalized margin-based generalization bounds, [9, 23] derived bounds from a PAC-Bayes approach, and [1, 33, 4] derived bounds from the compression point of view. They, in general, do not address why the low complexity arises. This paper takes a step towards this direction, though on two-layer networks and a simplified model of the data.

**Overparameterization and implicit regularization.** The training objectives of overparameterized networks in principle have many (approximate) global optima and some generalize better than the others [14, 8, 2], while empirical observations imply that the optimization process in practice prefers those with better generalization. It is then an interesting question how this implicit regularization or inductive bias arises from the optimization and the structure of the data. Recent studies are on SGD for different tasks, such as logistic regression [27] and matrix factorization [11, 19, 16]. More related to our work is [7], which studies the problem of learning a two-layer overparameterized network on linearly separable data and shows that SGD converges to a global optimum with good generalization. Our work studies the problem on data with a well clustered (and potentially not linearly separable) structure that we believe is closer to practical scenarios and thus can advance this line of research.

**Theoretical analysis of learning neural networks.** There also exists a large body of work that analyzes the optimization landscape of learning neural networks [13, 26, 30, 10, 25, 29, 6, 32, 17, 5]. They in general need to assume unrealistic assumptions about the data such as Gaussian-ness, and/or have strong assumptions about the network such as using only linear activation. They also do not study the implicit regularization by the optimization algorithms.

## 3 Problem Setup

In this work, a two-layer neural network with ReLU activation for $k$-classes classification is given by $f = (f_1, f_2, \cdots, f_k)$ such that for each $i \in [k]$:

$$f_i(x) = \sum_{r=1}^{m} a_{i,r} \mathbf{ReLU}(\langle w_r, x \rangle)$$

where $\{w_r \in \mathbb{R}^d\}$ are the weights for the $m$ neurons in the hidden layer, $\{a_{i,r} \in \mathbb{R}\}$ are the weights of the top layer, and $\mathbf{ReLU}(z) = \max\{0, z\}$.

**Assumptions about the data.** The data is generated from a distribution $\mathcal{D}$ as follows. There are $k \times l$ unknown distributions $\{\mathcal{D}_{i,j}\}_{i\in[k],j\in[l]}$ over $\mathbb{R}^d$ and probabilities $p_{i,j} \geq 0$ such that $\sum_{i,j} p_{i,j} = 1$. Each data point $(x, y)$ is i.i.d. generated by: (1) Sample $z \in [k] \times [l]$ such that $\Pr[z = (i, j)] = p_{i,j}$; (2) Set label $y = z[0]$, and sample $x$ from $\mathcal{D}_z$. Assume we sample $N$ points $\{(x_i, y_i)\}_{i=1}^N$.

Let us define the support of a distribution $\mathcal{D}$ with density $p$ over $\mathcal{R}^d$ as supp$(\mathcal{D}) = \{x : p(x) > 0\}$, the distance between two sets $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathcal{R}^d$ as dist$(\mathcal{S}_1, \mathcal{S}_2) = \min_{x\in\mathcal{S}_1,y\in\mathcal{S}_2}\{\|x - y\|_2\}$, and the diameter of a set $\mathcal{S}_1 \subseteq \mathcal{R}^d$ as diam$(\mathcal{S}_1) = \max_{x,y\in\mathcal{S}_1}\{\|x - y\|_2\}$. Then we are ready to make the assumptions about the data.

- **(A1)** (Separability) There exists $\delta > 0$ such that for every $i_1 \neq i_2 \in [k]$ and every $j_1, j_2 \in [l]$, dist$\left(\text{supp}(\mathcal{D}_{i_1,j_1}), \text{supp}(\mathcal{D}_{i_2,j_2})\right) \geq \delta$.Moreover, for every $i \in [k], j \in [l]$,[1] diam(supp$(\mathcal{D}_{i,j})) \leq \lambda\delta$, for $\lambda \leq 1/(8l)$.
- **(A2)** (Normalization) Any $x$ from the distribution has $\|x\|_2 = 1$.

A few remarks are worthy. Instead of having one distribution for one class, we allow an arbitrary $l \geq 1$ distributions in each class, which we believe is a better fit to the real data. For example, in MNIST, a class can be the number 1, and $l$ can be the different styles of writing 1 (1 or | or /).

Assumption **(A2)** is for simplicity, while **(A1)** is our key assumption. With $l \geq 1$ distributions inside each class, our assumption allows data that is not linearly separable, e.g., XOR type data in $\mathcal{R}^2$ where there are two classes, one consisting of two balls of diameter $1/10$ with centers $(0, 0)$ and $(2, 2)$ and the other consisting of two of the same diameter with centers $(0, 2)$ and $(2, 0)$. See Figure 3 in Appendix C for an illustration. Moreover, essentially the only assumption we have here is $\lambda = O(1/l)$. When $l = 1$, $\lambda = O(1)$, which is the minimal requirement on the order of $\lambda$ for the distribution to be efficiently learnable. Our work allows larger $l$, so that the data can be more complicated inside each class. In this case, we require the separation to also be higher. When we increase $l$ to refine the distributions inside each class, we should expect the diameters of each distribution become smaller as well. As long as the rate of diameter decreasing in each distribution is greater than the total number of distributions, then our assumption will hold.

**Assumptions about the learning process.** We will only learn the weight $w_r$ to simplify the analysis. Since the ReLU activation is positive homogeneous, the effect of overparameterization can still be studied, and a similar approach has been adopted in previous work [7]. So the network is also written as $y = f(x, w) = (f_1(x, w), \cdots, f_k(x, w))$ for $w = (w_1, \cdots, w_r)$.

We assume the learning is from a random initialization:

- **(A3)** (Random initialization) $w_r^{(0)} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, $a_{i,r} \sim \mathcal{N}(0, 1)$, with $\sigma = \frac{1}{m^{1/2}}$.

The learning process minimizes the cross entropy loss over the softmax, defined as:

$$L(w) = -\frac{1}{N}\sum_{s=1}^{N} \log o_{y_s}(x_s, w), \text{ where } o_y(x, w) = \frac{e^{f_y(x,w)}}{\sum_{i=1}^{k} e^{f_i(x,w)}}.$$

Let $L(w, x_s, y_s) = -\log o_{y_s}(x_s, w)$ denote the cross entropy loss for a particular point $(x_s, y_s)$.

We consider a minibatch SGD of batch size $B$, number of iterations $T = N/B$ and learning rate $\eta$ as the following process: Randomly divide the total training examples into $T$ batches, each of size $B$. Let the indices of the examples in the $t$-th batch be $\mathcal{B}_t$. At each iteration, the update is[2]

$$w_r^{(t+1)} = w_r^{(t)} - \eta\frac{1}{B}\sum_{s\in\mathcal{B}_t} \frac{\partial L(w^{(t)}, x_s, y_s)}{\partial w_r^{(t)}}, \forall r \in [m], \text{ where}$$

$$\frac{\partial L(w, x_s, y_s)}{\partial w_r} = \left(\sum_{i\neq y_s} a_{i,r}o_i(x_s, w) - \sum_{i\neq y_s} a_{y_s,r}o_i(x_s, w)\right) 1_{\langle w_r,x_s\rangle\geq 0}x_s. \tag{1}$$

[1] The assumption $1/(8l)$ can be made to $1/[(1 + \alpha)l]$ for any $\alpha > 0$ by paying a large polynomial in $1/\alpha$ in the sample complexity. We will not prove it in this paper because we would like to highlight the key factors.

[2] Strictly speaking, $L(w, x_s, y_s)$ does not have gradient everywhere due to the non-smoothness of ReLU. One can view $\frac{\partial L(w,x_s,y_s)}{\partial w_r}$ as a convenient notation for the right hand side of (1).

## 4 Main Result

For notation simplicity, for a target error $\varepsilon$ (to be specified later), with high probability (or w.h.p.) means with probability $1 - 1/\text{poly}(1/\delta, k, l, m, 1/\varepsilon)$ for a sufficiently large polynomial poly, and $\widetilde{O}$ hides factors of $\text{poly}(\log 1/\delta, \log k, \log l, \log m, \log 1/\varepsilon)$.

**Theorem 4.1.** *Suppose the assumptions **(A1)(A2)(A3)** are satisfied. Then for every $\varepsilon > 0$, there is $M = \text{poly}(k, l, 1/\delta, 1/\varepsilon)$ such that for every $m \geq M$, after doing a minibatch SGD with batch size $B = \text{poly}(k, l, 1/\delta, 1/\varepsilon, \log m)$ and learning rate $\eta = \frac{1}{m \cdot \text{poly}(k,l,1/\delta,1/\varepsilon,\log m)}$ for $T = \text{poly}(k, l, 1/\delta, 1/\varepsilon, \log m)$ iterations, with high probability:*

$$\Pr_{(x,y)\sim\mathcal{D}}\left[\forall j \in [k], j \neq y, f_y(x, w^{(T)}) > f_j(x, w^{(T)})\right] \geq 1 - \varepsilon.$$

Our theorem implies if the data satisfies our assumptions, and we parametrize the network properly, then we only need polynomial in $k, l, 1/\delta$ many samples to achieve a good prediction error. This error is measured directly on the true distribution $\mathcal{D}$, not merely on the input data used to train this network. Our result is also dimension free: There is no dependency on the underlying dimension $d$ of the data, the complexity is fully captured by $k, l, 1/\delta$. Moreover, no matter how much the network is overparameterized, it will only increase the total iterations by factors of $\log m$. So we can overparameterize by an *sub-exponential amount* without significantly increasing the complexity.

Furthermore, we can always treat each input example as an individual distribution, thus $\lambda$ is always zero. In this case, if we use batch size $B$ for $T$ iterations, we would have $l = N = BT$. Then our theorem indicate that as long as $m = \text{poly}(N, 1/\delta')$, where $\delta'$ is the minimal distance between each examples, we can actually fit arbitrary labels of the input data. However, since the total iteration only depends on $\log m$, when $m = \text{poly}(N, 1/\delta')$ but the input data is actually structured (with small $k, l$ and large $\delta$), then SGD can actually achieve a small generalization error, *even when* the network has enough capacity to fit arbitrary labels of the training examples (and can also be done by SGD). Thus, we prove that SGD has a strong inductive bias on structured data: Instead of finding a bad global optima that can fit arbitrary labels, it actually finds those with good generalization guarantees. This gives more thorough explanation to the empirical observations in [24, 31].

## 5 Intuition and Proof Sketch for A Simplified Case

To train a neural network with ReLU activations, there are two questions need to be addressed:

1 Why can SGD optimize the training loss? Or even finding a critical point? Since the underlying network is highly non-smooth, existing theorems do not give any finite convergence rate of SGD for training neural network with ReLUs activations.

2 Why can the trained network generalize? Even when the capacity is large enough to fit random labels of the input data? This is known as the inductive bias of SGD.

This work takes a step towards answering these two questions. We show that when the network is overparameterized, it becomes more "pseudo smooth", which makes it easir for SGD to minimize the training loss, and furthermore, it will not hurt the generalization error. Our proof is based on the following important observation:

> The more we overparameterize the network, the less likely the activation pattern for one neuron and one data point will change in a fixed number of iterations.

This observation allows us to couple the gradient of the true neural network with a "pseudo gradient" where the activation pattern for each data point and each neuron is fixed. That is, when computing the "pseudo gradient", for fixed $r, i$, whether the $r$-th hidden node is activated on the $i$-th data point $x_i$ will always be the same for different $t$. (But for fixed $t$, for different $r$ or $i$, the sign can be different.) We are able to prove that unless the generalization error is small, the "pseudo gradient" will always be large. Moreover, we show that the network is actually smooth thus SGD can minimize the loss.

We then show that when the number $m$ of hidden neurons increases, with a properly decreasing learning rate, the total number of iterations it takes to minimize the loss is roughly not changed.

However, the total number of iterations that we can couple the true gradient with the pseudo one increases. Thus, there is a polynomially large $m$ so that we can couple these two gradients until the network reaches a small generalization error.

### 5.1 A Simplified Case: No Variance

Here we illustrate the proof sketch for a simplified case and Appendix A provides the proof. The proof for the general case is provided in Appendix B. In the simplified case, we further assume:

- **(S)** (No variance) Each $\mathcal{D}_{a,b}$ is a single data point $(x_{a,b}, a)$, and also we are doing full batch gradient descent as opposite to the minibatch SGD.

Then we reload the loss notation as $L(w) = \sum_{a\in[k],b\in[l]} p_{a,b} L(w, x_{a,b}, a)$, and the gradient is

$$\frac{\partial L(w)}{\partial w_r} = \sum_{a\in[k],b\in[l]} p_{a,b} \left( \sum_{i\neq a} a_{i,r} o_i(x_{a,b}, w) - \sum_{i\neq a} a_{a,r} o_i(x_{a,b}, w) \right) 1_{\langle w_r, x_{a,b}\rangle \geq 0} x_{a,b}.$$

Following the intuition above, we define the pseudo gradient as

$$\frac{\tilde{\partial} L(w)}{\partial w_r} = \sum_{a\in[k],b\in[l]} p_{a,b} \left( \sum_{i\neq a} a_{i,r} o_i(x_{a,b}, w) - \sum_{i\neq a} a_{a,r} o_i(x_{a,b}, w) \right) 1_{\langle w_r^{(0)}, x_{a,b}\rangle \geq 0} x_{a,b},$$

where it uses $1_{\langle w_r^{(0)}, x_{a,b}\rangle \geq 0}$ instead of $1_{\langle w_r, x_{a,b}\rangle \geq 0}$ as in the true gradient. That is, the activation pattern is set to be that in the initialization. Intuitively, the pseudo gradient is similar to the gradient for a pseudo network $g$ (but not exactly the same), defined as $g_i(x, w) := \sum_{r=1}^{m} a_{i,r} \langle w_r, x\rangle 1_{\left\langle w_r^{(0)}, x\right\rangle \geq 0}$. Coupling the gradients is then similar to coupling the networks $f$ and $g$.

For simplicity, let $v_{a,a,b} := \sum_{i\neq a} o_i(x_{a,b}, w) = \frac{\sum_{i\neq a} e^{f_i(x_{a,b},w)}}{\sum_{i=1}^{k} e^{f_i(x_{a,b},w)}}$ and when $s \neq a$, $v_{s,a,b} := -o_s(x_{a,b}, w) = -\frac{e^{f_s(x_{a,b},w)}}{\sum_{i=1}^{k} e^{f_i(x_{a,b},w)}}$. Roughly, if $v_{a,a,b}$ is small, then $f_a(x_{a,b}, w)$ is relatively larger compared to the other $f_i(x_{a,b}, w)$, so the classification error is small.

We prove the following two main lemmas. The first says that at each iteration, the total number of hidden units whose gradient can be coupled with the pseudo one is quite large.

**Lemma 5.1** (Coupling)**.** *W.h.p. over the random initialization, for every $\tau > 0$, for every $t = \tilde{O}\left(\frac{\tau}{\eta}\right)$, we have that for at least $1 - \frac{e\tau kl}{\sigma}$ fraction of $r \in [m]$: $\frac{\partial L(w^{(t)})}{\partial w_r} = \frac{\tilde{\partial} L(w^{(t)})}{\partial w_r}$.*

The second lemma says that the pseudo gradient is large unless the error is small.

**Lemma 5.2.** *For $m = \tilde{\Omega}\left(\frac{k^3 l^2}{\delta}\right)$, for every $\{p_{a,b} v_{i,a,b}\}_{i,a\in[k],b\in[l]} \in [-v, v]$ (that depends on $w_r^{(0)}$, $a_{i,r}$, etc.) with $\max\{p_{a,b} v_{i,a,b}\}_{i,a\in[k],b\in[l]} = v$, there exists at least $\Omega(\frac{\delta}{kl})$ fraction of $r \in [m]$ such that $\left\| \frac{\tilde{\partial} L(w)}{\partial w_r} \right\|_2 = \tilde{\Omega}\left(\frac{v\delta}{kl}\right)$.*

We now illustrate how to use these two lemmas to show the convergence for a small enough learning rate $\eta$. For simplicity, let us assume that $kl/\delta = O(1)$ and $\varepsilon = o(1)$. Thus, by Lemma 5.2 we know that unless $v \leq \varepsilon$, there are $\Omega(1)$ fraction of $r$ such that $\left\|\tilde{\partial} L(w)/\partial w_r\right\|_2 = \Omega(\varepsilon)$. Moreover, by Lemma 5.1 we know that we can pick $\tau = \Theta(\sigma\varepsilon)$ so $e\tau/\sigma = \Theta(\varepsilon)$, which implies that there are $\Omega(1)$ fraction of $r$ such that $\|\partial L(w)/\partial w_r\|_2 = \Omega(\varepsilon)$ as well. For small enough learning rate $\eta$, doing one step of gradient descent will thus decrease $L(w)$ by $\Omega(\eta m \varepsilon^2)$, so it converges in $t = O\left(1/\eta m \varepsilon^2\right)$ iterations. In the end, we just need to make sure that $1/\eta m \varepsilon^2 \leq O(\tau/\eta) = \Theta(\sigma\varepsilon/\eta)$ so we can always apply the coupling Lemma 5.1. By $\sigma = \tilde{O}(1/m^{-1/2})$ we know that this is true as long as $m \geq \mathrm{poly}(1/\varepsilon)$. A small $v$ can be shown to lead to a small generalization error.

# 6 Discussion of Insights from the Analysis

Our analysis, though for learning two-layer networks on well structured data, also sheds some light upon learning neural networks in more general settings.

**Generalization.** Several lines of recent work explain the generalization phenomenon of overparameterized networks by low complexity of the learned networks, from the point views of spectrally-normalized margins [3, 23, 21], compression [1, 33, 4], and PAC-Bayes [9, 23].

Our analysis has partially explained how SGD (with proper random initialization) on structured data leads to the low complexity from the compression and PCA-Bayes point views. We have shown that in a neighborhood of the random initialization, w.h.p. the gradients are similar to those of another benign learning process, and thus SGD can reduce the error and reach a good solution while still in the neighborhood. The closeness to the initialization then means the weights (or more precisely the difference between the learned weights and the initialization) can be easily compressed. In fact, empirical observations have been made and connected to generalization in [22, 1]. Furthermore, [1] explicitly point out such a compression using a helper string (corresponding to the initialization in our setting). [1] also point out that the compression view can be regarded as a more explicit form of the PAC-Bayes view, and thus our intuition also applies to the latter.

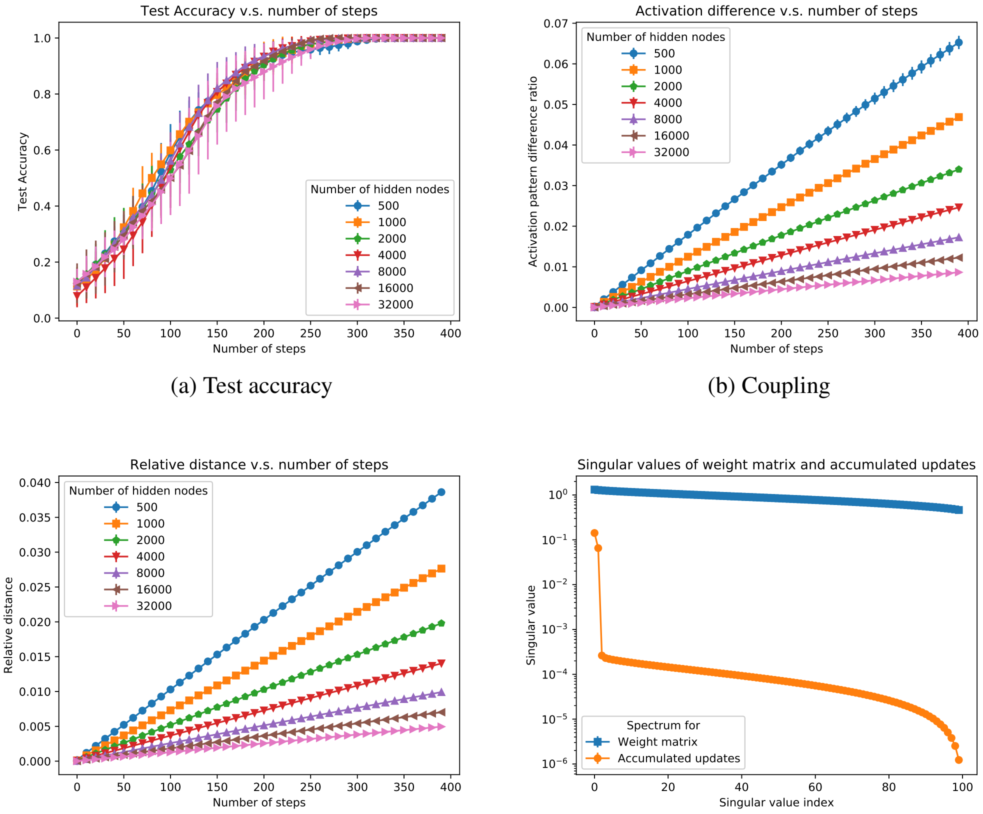
The existence of a solution of a small generalization error near the initialization is itself not obvious. Intuitively, on structured data, the updates are structured signals spread out across the weights of the hidden neurons. Then for prediction, the random initialized part in the weights has strong cancellation, while the structured signal part in the weights collectively affects the output. Therefore, the latter can be much smaller than the former while the network can still give accurate predictions. In other words, there can be a solution not far from the initialization with high probability.

Some insight is provided on the low rank of the weights. More precisely, when the data are well clustered around a few patterns, the accumulated updates (difference between the learned weights and the initialization) should be approximately low rank, which can be seen from checking the SGD updates. However, when the difference is small compared to the initialization, the spectrum of the final weight matrix is dominated by that of the initialization and thus will tend to closer to that of a random matrix. Again, such observations/intuitions have been made in the literature and connected to compression and generalization (e.g., [1]).

**Implicit regularization v.s. structure of the data.** Existing work has analyzed the implicit regularization of SGD on logistic regression [27], matrix factorization [11, 19, 16], and learning two-layer networks on linearly separable data [7]. Our setting and also the analysis techniques are novel compared to the existing work. One motivation to study on structured data is to understand the role of structured data play in the implicit regularization, i.e., the observation that the solution learned on less structured or even random data is further away from the initialization. Indeed, our analysis shows that when the network size is fixed (and sufficiently overparameterized), learning over poorly structured data (larger $k$ and $\ell$) needs more iterations and thus the solution can deviate more from the initialization and has higher complexity. An extreme and especially interesting case is when the network is overparameterized so that in principle it can fit the training data by viewing each point as a component while actually they come from structured distributions with small number of components. In this case, we can show that it still learns a network with a small generalization error; see the more technical discussion in Section 4.

We also note that our analysis is under the assumption that the network is sufficiently overparameterized, i.e., $m$ is a sufficiently large polynomial of $k$, $\ell$ and other related parameters measuring the structure of the data. There could be the case that $m$ is smaller than this polynomial but is more than sufficient to fit the data, i.e., the network is still overparameterized. Though in this case the analysis still provides useful insight, it does not fully apply; see our experiments with relatively small $m$. On the other hand, the empirical observations [24, 31] suggest that practical networks are highly overparameterized, so our intuition may still be helpful there.

**Effect of random initialization.** Our analysis also shows how proper random initializations helps the optimization and consequently generalization. Essentially, this guarantees that w.h.p. for weights close to the initialization, many hidden ReLU units will have the same activation patterns (i.e., activated or not) as for the initializations, which means the gradients in the neighborhood look like those when the hidden units have fixed activation patterns. This allows SGD makes progress when



(a) Test accuracy

(b) Coupling

(c) Distance from the initialization

(d) Rank of accumulated updates ($y$-axis in log-scale)

Figure 1: Results on the synthetic data.

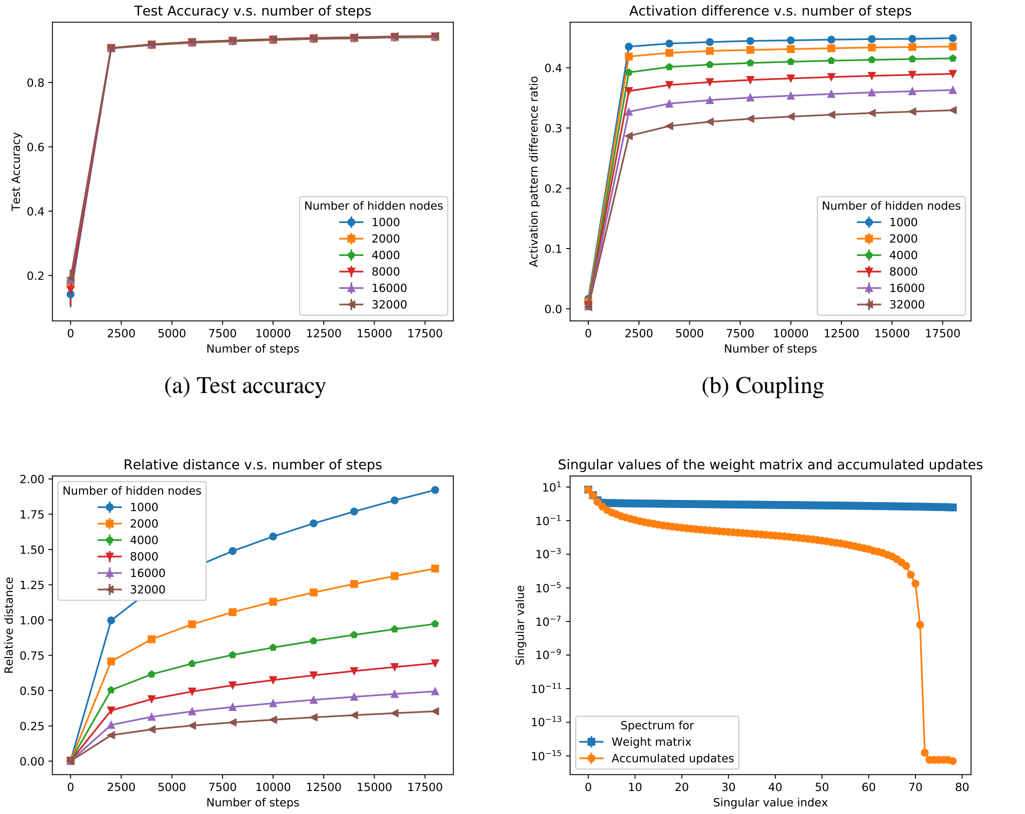
the loss is large, and eventually learns a good solution. We also note that it is essential to carefully set the scale of the initialization, which is a extensively studied topic [20, 28]. Our initialization has a scale related to the number of hidden units, which is particularly useful when the network size is varying, and thus can be of interest in such practical settings.

## 7 Experiments

This section aims at verifying some key implications: (1) the activation patterns of the hidden units couple with those at initialization; (2) The distance from the learned solution from the initialization is relatively small compared to the size of initialization; (3) The accumulated updates (i.e., the difference between the learned weight matrix and the initialization) have approximately low rank. These are indeed supported by the results on the synthetic and the MNIST data. Additional experiments are presented in Appendix D.

**Setup.** The synthetic data are of 1000 dimension and consist of $k = 10$ classes, each having $\ell = 2$ components. Each component is of equal probability $1/(kl)$, and is a Gaussian with covariance $\sigma^2/dI$ and its mean is i.i.d. sampled from a Gaussian distribution $\mathcal{N}(0, \sigma_0^2/d)$, where $\sigma = 1$ and $\sigma_0 = 5$. 1000 training data points and 1000 test data points are sampled.

The network structure and the learning process follow those in Section 3; the number of hidden units $m$ varies in the experiments, and the weights are initialized with $\mathcal{N}(0, 1/\sqrt{m})$. On the synthetic data, the SGD is run for $T = 400$ steps with batch size $B = 16$ and learning rate $\eta = 10/m$. On MNIST, the SGD is run for $T = 2 \times 10^4$ steps with batch size $B = 64$ and learning rate $\eta = 4 \times 10^2/m$.



(a) Test accuracy

(b) Coupling

(c) Distance from the initialization

(d) Rank of accumulated updates ($y$-axis in log-scale)

Figure 2: Results on the MNIST data.

Besides the test accuracy, we report three quantities corresponding to the three observations/implications to be verified. First, for coupling, we compute the fraction of hidden units whose activation pattern changed compared to the time at initialization. Here, the activation pattern is defined as $1$ if the input to the ReLU is positive and $0$ otherwise. Second, for distance, we compute the relative ratio $\|w^{(t)} - w^{(0)}\|_F / \|w^{(0)}\|_F$, where $w^{(t)}$ is the weight matrix at time $t$. Finally, for the rank of the accumulated updates, we plot the singular values of $w^{(T)} - w^{(0)}$ where $T$ is the final step. All experiments are repeated 5 times, and the mean and standard deviation are reported.

**Results.** Figure 1 shows the results on the synthetic data. The test accuracy quickly converges to $100\%$, which is even more significant with larger number of hidden units, showing that the overparameterization helps the optimization and generalization. Recall that our analysis shows that for a learning rate linearly decreasing with the number of hidden nodes $m$, the number of iterations to get the accuracy to achieve a desired accuracy should be roughly the same, which is also verified here. The activation pattern difference ratio is less than $0.1$, indicating a strong coupling. The relative distance is less than $0.1$, so the final solution is indeed close to the initialization. Finally, the top 20 singular values of the accumulated updates are much larger than the rest while the spectrum of the weight matrix do not have such structure, which is also consistent with our analysis.

Figure 2 shows the results on MNIST. The observation in general is similar to those on the synthetic data (though less significant), and also the observed trend become more evident with more overparameterization. Some additional results (e.g., varying the variance of the synthetic data) are provided in the appendix that also support our theory.

## 8 Conclusion

This work studied the problem of learning a two-layer overparameterized ReLU neural network via stochastic gradient descent (SGD) from random initialization, on data with structure inspired by practical datasets. While our work makes a step towards theoretical understanding of SGD for training neural networs, it is far from being conclusive. In particular, the real data could be separable with respect to different metric than $\ell_2$, or even a non-convex distance given by some manifold. We view this an important open direction.

## Acknowledgements

We would like to thank the anonymous reviewers of NeurIPS'18 and Jason Lee for helpful comments. This work was supported in part by FA9550-18-1-0166, NSF grants CCF-1527371, DMS-1317308, Simons Investigator Award, Simons Collaboration Grant, and ONR-N00014-16-1-2329. Yingyu Liang would also like to acknowledge that support for this research was provided by the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin Madison with funding from the Wisconsin Alumni Research Foundation.

## References

[1] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*, 2018.

[2] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. *arXiv preprint arXiv:1706.05394*, 2017.

[3] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6241–6250, 2017.

[4] Cenk Baykal, Lucas Liebenwein, Igor Gilitschenski, Dan Feldman, and Daniela Rus. Data-dependent coresets for compressing neural networks with applications to generalization bounds. *arXiv preprint arXiv:1804.05345*, 2018.

[5] Digvijay Boob and Guanghui Lan. Theoretical properties of the global optimizer of two layer neural network. *arXiv preprint arXiv:1710.11241*, 2017.

[6] Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. *arXiv preprint arXiv:1702.07966*, 2017.

[7] Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. Sgd learns over-parameterized networks that provably generalize on linearly separable data. *arXiv preprint arXiv:1710.10174*, 2017.

[8] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*, 2017.

[9] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

[10] Rong Ge, Jason D Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017.

[11] Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pages 6152–6160, 2017.

[12] Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

[13] Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, pages 586–594, 2016.

[14] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix recovery. *arXiv preprint arXiv:1712.09203*, 2017.

[17] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.

[18] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863, 2014.

[19] Cong Ma, Kaizheng Wang, Yuejie Chi, and Yuxin Chen. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion and blind deconvolution. *arXiv preprint arXiv:1711.10467*, 2017.

[20] James Martens. Deep learning via hessian-free optimization. In *ICML*, volume 27, pages 735–742, 2010.

[21] Cisse Moustapha, Bojanowski Piotr, Grave Edouard, Dauphin Yann, and Usunier Nicolas. Parseval networks: Improving robustness to adversarial examples. *arXiv preprint arXiv:1704.08847*, 2017.

[22] Vaishnavh Nagarajan and Zico Kolter. Generalization in deep networks: The role of distance from initialization. *NIPS workshop on Deep Learning: Bridging Theory and Practice*, 2017.

[23] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

[24] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[25] Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *arXiv preprint arXiv:1707.04926*, 2017.

[26] Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

[27] Daniel Soudry, Elad Hoffer, and Nathan Srebro. The implicit bias of gradient descent on separable data. *arXiv preprint arXiv:1710.10345*, 2017.

[28] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[29] Yuandong Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. *arXiv preprint arXiv:1703.00560*, 2017.

[30] Bo Xie, Yingyu Liang, and Le Song. Diversity leads to generalization in neural networks. *arXiv preprint Arxiv:1611.03131*, 2016.

[31] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[32] Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. *arXiv preprint arXiv:1706.03175*, 2017.

[33] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. Compressibility and generalization in large-scale deep learning. *arXiv preprint arXiv:1804.05862*, 2018.

# A Proofs for the Simplified Case

In the simplified case, we make the following simplifying assumption:

- **(S)** (No variance) Each $\mathcal{D}_{a,b}$ is a single data point $(x_{a,b}, a)$, and also we are doing full batch gradient descent as opposite to the minibatch SGD.

Recall that the loss is then $L(w) = \sum_{a\in[k],b\in[l]} p_{a,b} L(w, x_{a,b}, a)$. The gradient descent update on $w$ is given by

$$w_r^{(t+1)} = w_r^{(t)} - \eta \frac{\partial L(w^{(t)})}{\partial w_r^{(t)}},$$

and the gradient is

$$\frac{\partial L(w)}{\partial w_r} = \sum_{a\in[k],b\in[l]} p_{a,b} \left( \sum_{i\neq a} a_{i,r} o_i(x_{a,b}, w) - \sum_{i \neq a} a_{a,r} o_i(x_{a,b}, w) \right) 1_{\langle w_r, x_{a,b}\rangle \geq 0} x_{a,b},$$

where $o_y(x, w) = \frac{e^{f_y(x,w)}}{\sum_{i=1}^k e^{f_i(x,w)}}$. The pseudo gradient is defined as

$$\frac{\tilde{\partial} L(w)}{\partial w_r} = \sum_{a\in[k],b\in[l]} p_{a,b} \left( \sum_{i\neq a} a_{i,r} o_i(x_{a,b}, w) - \sum_{i \neq a} a_{a,r} o_i(x_{a,b}, w) \right) 1_{\langle w_r^{(0)}, x_{a,b}\rangle \geq 0} x_{a,b}.$$

Let us call

$$v_{s,a,b}(w) = \begin{cases} \frac{\sum_{i\neq a} e^{f_i(x_{a,b},w)}}{\sum_{i=1}^k e^{f_i(x_{a,b},w)}} & \text{if } s = a; \\ -\frac{e^{f_s(x_{a,b},w)}}{\sum_{i=1}^k e^{f_i(x_{a,b},w)}} & \text{otherwise.} \end{cases}$$

When clear from the context, we write $v_{s,a,b}(w)$ as $v_{s,a,b}$. Then we can simplify the above expression as:

$$\frac{\tilde{\partial} L(w)}{\partial w_r} = \sum_{a\in[k],b\in[l],i\in[k]} p_{a,b} a_{i,r} v_{i,a,b} 1_{\left\langle w_r^{(0)}, x_{a,b}\right\rangle \geq 0} x_{a,b}.$$

By definition, $v_{i,a,b}$'s satisfy:

1. $\forall a \in [k], b \in [l] : v_{a,a,b} \in [0, 1]$.
2. $\sum_{i=1}^k v_{i,a,b} = 0$.

Furthermore, $v_{a,a,b}$ indicates the "classification error". The smaller $v_{a,a,b}$ is, the smaller the classification error is.

In the following subsections, we first show that the gradient is coupled with the pseudo gradient, then show that if the classification error is large then the pseudo gradient is large, and finally prove the convergence.

## A.1 Coupling

We will show that $\partial L(w^{(t)})/\partial w_r$ is close to $\tilde{\partial} L(w^{(t)})/\partial w_r$ in the following sense:

**Lemma A.1** (Coupling, Lemma 5.1 restated)**.** *W.h.p. over the random initialization, for every $\tau > 0$, for every $t = \tilde{O}\left(\frac{\tau}{\eta}\right)$, we have that for at least $1 - \frac{e\tau k l}{\sigma}$ fraction of $r \in [m]$:*

$$\frac{\partial L(w^{(t)})}{\partial w_r} = \frac{\tilde{\partial} L(w^{(t)})}{\partial w_r}.$$

*Proof.* W.h.p. we know that every $|a_{i,r}| \le L = \tilde{O}(1)$. Thus, for every $r \in [m]$ and every $t \ge 0$, we have

$$\left\| \frac{\partial L(w^{(t)})}{\partial w_r} \right\|_2 \le L$$

which implies that $\left\| w_r^{(t)} - w_r^{(0)} \right\|_2 \le L\eta t$.

Now, for every $\tau \ge 0$, we consider the set $\mathcal{H}$ such that

$$\mathcal{H} = \left\{ r \in [m] \mid \forall a \in [k], b \in [l] : \left| \langle w_r^{(0)}, x_{a,b} \rangle \right| \ge \tau \right\}.$$

For every $r \in \mathcal{H}$ and every $t \le \frac{\tau}{2L\eta}$, we know that for every $a \in [k], b \in [l]$:

$$\left| \left\langle w_r^{(t)} - w_r^{(0)}, x_{a,b} \right\rangle \right| \le L\eta t \le \frac{\tau}{2}$$

which implies that

$$1_{\left\langle w_r^{(0)}, x_{a,b} \right\rangle \ge 0} = 1_{\left\langle w_r^{(t)}, x_{a,b} \right\rangle \ge 0}.$$

This implies that $\frac{\partial L(w^{(t)})}{\partial w_r} = \frac{\tilde{\partial} L(w^{(t)})}{\partial w_r}$.

Now, we need to bound the size of $\mathcal{H}$. Since $\left\langle w_r^{(0)}, x_{a,b} \right\rangle \sim \mathcal{N}(0, \sigma^2)$, by standard property of Gaussian we directly have that for $|\mathcal{H}| \ge 1 - \frac{e\tau kl}{\sigma}$. □

### A.2 Error Large $\implies$ Gradient Large

The pseudo gradient can be rewritten as the following summation:

$$\frac{\tilde{\partial} L(w)}{\partial w_r} = \sum_{i \in [k]} a_{i,r} P_{i,r}$$

where

$$P_{i,r} = \sum_{a \in [k], b \in [l]} p_{a,b} v_{i,a,b} 1_{\left\langle w_r^{(0)}, x_{a,b} \right\rangle \ge 0} x_{a,b}.$$

We would like to show that if some $p_{a,b}v_{i,a,b}$ is large, a good fraction of $r \in [m]$ will have large pseudo gradient. Now, the first step is to show that for any fixed $\{p_{a,b}v_{i,a,b}\}$ (that does not depend on the random initialization $w_r^{(0)}$), with good probability (over the random choice of $w_r^{(0)}$) we have that $P_{i,r}$ is large; see Lemma A.2. Then we will take a union bound over an epsilon net on $\{p_{a,b}v_{i,a,b}\}$ to show that for every $\{p_{a,b}v_{,ia,b}\}$ (that can depend on $w_r^{(0)}$), at least a good fraction of of $P_{i,r}$ is large; See Lemma A.3.

**Lemma A.2** (The geometry of $\mathbf{ReLU}$)**.** *For any possible fixed* $\{p_{a,b}v_{1,a,b}\}_{a \in [k], b \in [l]} \in [-v, v]$ *such that* $p_{1,1}v_{1,1,1} = v$*, we have:*

$$\Pr\left[ \|P_{1,r}\|_2 = \tilde{\Omega}\left( \frac{v\delta}{kl} \right) \right] = \Omega\left( \frac{\delta}{kl} \right).$$

Clearly, without $\mathbf{ReLU}$, $P_{1,r}$ can be arbitrarily small if, say, $\forall b \in [l], v_{1,1,b} = v, p_{1,b} = p$ and $\sum_{b \in [l]} x_{1,b} = 0$. However, $\mathbf{ReLU}$ would prevent the cancellation of those $x_{1,b}$'s.

*Proof of Lemma A.2.* We will first prove that

$$h\left(w_r^{(0)}\right) = \sum_{a \in [k], b \in [l]} p_{a,b} v_{1,a,b} \mathbf{ReLU}\left( \left\langle w_r^{(0)}, x_{a,b} \right\rangle \right) = \langle P_{1,r}, w_r^{(0)} \rangle$$

is large with good probability.

Let us decompose $w_r^{(0)}$ into:

$$w_r^{(0)} = \alpha x_{1,1} + \beta$$

where $\beta \perp x_{1,1}$. For every $\tau \ge 0$, consider the event $\mathcal{E}_\tau$ defined as

1. $|\alpha| \leq \tau$, and
2. for all $a \in [k]\backslash[1], b \in [l]$: $|\langle \beta, x_{a,b} \rangle| \geq 4\tau$.

By the definition of initialization $w_r^{(0)}$, we know that:

$$\alpha \sim \mathcal{N}(0, \sigma^2)$$

and

$$\langle \beta, x_{a,b} \rangle \sim \mathcal{N}(0, (1 - \langle x_{a,b}, x_{1,1} \rangle^2)\sigma^2)$$

By assumption we know that for every $a \in [k]\backslash[1], b \in [l]$:

$$1 - \langle x_{a,b}, x_{1,1} \rangle^2 \geq \delta^2.$$

This implies that

$$\Pr\left[|\langle \beta, x_{a,b} \rangle| \leq 4\tau\right] \leq \frac{4e\tau}{\delta\sigma}.$$

Thus if we pick $\tau \leq \frac{\delta\sigma}{16ekl}$, taking a union bound we know that

$$\Pr\left[\forall a \in [k]\backslash[1], b \in [l] : |\langle \beta, x_{a,b} \rangle| \geq 4\tau\right] \geq \frac{1}{2}.$$

Moreover, since $\Pr[|\alpha| \leq \tau] \geq \frac{\tau}{e\sigma}$ and $\alpha$ is independent of $\beta$, we know that $\Pr[\mathcal{E}_\tau] \geq \frac{\tau}{16e^2\sigma}$.

The following proof will conditional on this event $\mathcal{E}_\tau$, and then treat $\beta$ as fixed and let $\alpha$ be the only random variable. In this way, we will have: for every $\alpha$ such that $|\alpha| \leq \tau$ and for every $a \in [k]\backslash[1], b \in [l]$, since $|\langle \beta, x_{a,b} \rangle| \geq 4\tau$ and $|\alpha\langle x_{1,1}, x_{a,b} \rangle| \leq \tau$,

$$\mathbf{ReLU}\left(\left\langle w_r^{(0)}, x_{a,b} \right\rangle\right) = \left(\alpha\langle x_{1,1}, x_{a,b} \rangle + \langle \beta, x_{a,b} \rangle\right) 1_{\langle \beta, x_{a,b} \rangle \geq 0}$$

which is a linear function of $\alpha$. With this information, we can rewrite $h\left(w_r^{(0)}\right)$ as:

$$\begin{aligned} h\left(w_r^{(0)}\right) = h(\alpha) &:= p_{1,1}v_{1,1,1}\mathbf{ReLU}(\alpha) \\ &+ \sum_{b \in [l]\backslash[1]} p_{1,b}v_{1,1,b}\mathbf{ReLU}\left(\alpha\langle x_{1,1}, x_{1,b} \rangle + \langle \beta, x_{a,b} \rangle\right) \\ &+ \mathsf{Linear}(\alpha) \end{aligned}$$

where $p_{1,b}v_{1,1,b} \geq 0$ and $\mathsf{Linear}(\alpha)$ is some linear function in $\alpha$. Thus, we know that

$$\phi(\alpha) := p_{1,1}v_{1,1,1}\mathbf{ReLU}(\alpha) + \sum_{b \in [l]\backslash[1]} p_{1,b}v_{1,1,b}\mathbf{ReLU}\left(\alpha\langle x_{1,1}, x_{1,b} \rangle\right)$$

is a convex function with $|\partial_{\max}\phi(0) - \partial_{\min}\phi(0)| \geq v$. Then applying Lemma A.5 gives

$$\Pr_{\alpha \sim U(-\tau,\tau)}\left[|\phi(\alpha) + \mathsf{Linear}(\alpha)| \geq \frac{v\tau}{128}\right] \geq \frac{1}{16}.$$

Since for $\tau \leq \frac{\delta\sigma}{16ekl}$, conditional on $\mathcal{E}_\tau$ the density $p(\alpha) \in \left[\frac{1}{e\tau}, \frac{e}{\tau}\right]$, which implies that

$$\Pr\left[h\left(w_r^{(0)}\right) \geq \frac{v\tau}{128} \mid \mathcal{E}_\tau\right] \geq \frac{1}{16e}.$$

Thus we have:

$$\Pr\left[h\left(w_r^{(0)}\right) \geq \frac{v\tau}{128}\right] \geq \Pr\left[h\left(w_r^{(0)}\right) \geq \frac{v\tau}{128} \mid \mathcal{E}_\tau\right]\Pr[\mathcal{E}_\tau] = \Omega\left(\frac{\tau}{\sigma}\right). \tag{2}$$

Now we can look at $P_{1,r}$. By the random initialization of $w_r^{(0)}$, and since by our assumption $v_{1,a,b}, x_{a,b}$ are not functions of $w_r^{(0)}$, a standard tail bound of Gaussian random variables shows that for every fixed $v_{1,a,b}$ and every $c > 10$:

$$\begin{aligned} \Pr\left[h\left(w_r^{(0)}\right) \geq 10c\sigma\|P_{1,r}\|_2\right] &= \Pr\left[\left\langle P_{1,r}, w_r^{(0)} \right\rangle \geq 10c\sigma\|P_{1,r}\|_2\right] \\ &\leq e^{-c^2}. \end{aligned}$$

Taking $c = 100\sqrt{\log \frac{kl}{\delta\sigma}}$ and putting together with inequality (2) with $\tau = \Theta\left(\frac{\delta\sigma}{kl}\right)$ complete the proof. □

Now, we can take an epsilon net and switch the order of the quantifiers in Lemma A.2 as shown in the following lemma.

**Lemma A.3** (Lemma 5.2 restated). *For* $m = \tilde{\Omega}\left(\frac{k^3 l^2}{\delta}\right)$*, for every* $\{p_{a,b}v_{i,a,b}\}_{i,a\in[k],b\in[l]} \in [-v, v]$ *(that depends on* $w_r^{(0)}, a_{i,r}$*, etc.) with* $\max\{p_{a,b}v_{i,a,b}\}_{i,a\in[k],b\in[l]} = v$*, there exists at least* $\Omega(\frac{\delta}{kl})$ *fraction of* $r \in [m]$ *such that*

$$\left\|\frac{\tilde{\partial} L(w)}{\partial w_r}\right\|_2 = \tilde{\Omega}\left(\frac{v\delta}{kl}\right).$$

This lemma implies that if the classification error is large, then many $w_r$ will have a large gradient.

*Proof of Lemma A.3.* We first consider fixed $\{p_{a,b}v_{i,a,b}\}_{i,a\in[k],b\in[l]} \in [-v, v]$. First of all, using the randomness of $a_{i,r}$ we know that with probability at least $1/e$,

$$\left\|\frac{\tilde{\partial} L(w)}{\partial w_r}\right\|_2 = \left\|\sum_{i=1}^{k} a_{i,r} P_{i,r}\right\|_2 \geq \|P_{1,r}\|_2.$$

Now, apply Lemma A.2 we know that

$$\Pr\left[\|P_{1,r}\|_2 = \tilde{\Omega}\left(\frac{v\delta}{kl}\right)\right] = \Omega\left(\frac{\delta}{kl}\right)$$

which implies that for fixed $\{p_{a,b}v_{i,a,b}\}_{i,a\in[k],b\in[l]} \in [-v, v]$ the probability that there are less than $O(\frac{\delta}{kl})$ of $r$ such that $\left\|\sum_{i=1}^{k} P_{i,r}\right\|_2$ is $\tilde{\Omega}\left(\frac{v\delta}{kl}\right)$ is no more than a value $p_{fix}$ given by:

$$p_{fix} \leq \exp\left\{-\Omega\left(\frac{\delta m}{kl}\right)\right\}.$$

Moreover, for every $\varepsilon > 0$, for two different $\{p_{a,b}v_{i,a,b}\}_{i,a\in[k],b\in[l]}, \{p_{a,b}v'_{i,a,b}\}_{i,a\in[k],b\in[l]} \in [-v, v]$ such that for all $i \in [k], a \in [k], b \in [l]$: $|p_{a,b}v_{i,a,b} - p_{a,b}v_{i,a,b}| \leq \varepsilon$. Moreover, since w.h.p. we know that every $|a_{i,r}| \leq L = \tilde{O}(1)$, it shows:

$$\left\|\sum_{a\in[k],b\in[l],i\in[k]} p_{a,b}a_{i,r}(v_{i,a,b} - v'_{i,a,b})1_{\left\langle w_r^{(0)}, x_{a,b}\right\rangle \geq 0} x_{a,b}\right\|_2 \leq L\varepsilon = \tilde{O}(\varepsilon)$$

which implies that we can take an $\ell_\infty$ $\varepsilon$-net over $\{p_{a,b}v_{i,a,b}\}_{i,a\in[k],b\in[l]} \in [-v, v]$ with $\varepsilon = \tilde{\Theta}\left(\frac{v\delta}{kl}\right)$. Thus, the probability that there exists $\{p_{a,b}v_{i,a,b}\}_{i,a\in[k],b\in[l]} \in [-v, v]$, such that there are no more than $O(\frac{\delta}{kl})$ fraction of $r \in [m]$with $\left\|\sum_{i=1}^{k} P_{i,r}\right\|_2 = \tilde{\Omega}\left(\frac{v\delta}{kl}\right)$ is no more than:

$$p \leq p_{fix}\left(\frac{v}{\varepsilon}\right)^{k^2 l} \leq \exp\left\{-\Omega\left(\frac{\delta m}{kl}\right) + k^2 l \log\frac{v}{\varepsilon}\right\}.$$

With $m = \tilde{\Omega}\left(\frac{k^3 l^2}{\delta}\right)$ we complete the proof. □

### A.3 Convergence

Having the lemmas, we can now prove the convergence:

**Lemma A.4** (Convergence)**.** *Let us denote* $\max\{p_{a,b}v_{i,a,b}^{(t)}\} = v^{(t)}$*. Then for a sufficiently small* $\eta$*, we have that for every* $T = \tilde{\Theta}\left(\frac{\sigma\delta}{kl\eta}\right)$*,*

$$\frac{1}{T}\sum_{t=1}^{T}\left(v^{(t)}\right)^2 = \tilde{O}\left(\frac{k^5l^5}{\delta^4\sigma m}\right).$$

By our choice of $\sigma = \tilde{O}\left(\frac{1}{m^{1/2}}\right)$, we know that

$$\frac{1}{T}\sum_{t=1}^{T}\left(v^{(t)}\right)^2 = \tilde{O}\left(\frac{k^5l^5}{\delta^4 m^{1/2}}\right)$$

Thus, this lemma shows that eventually $v^{(t)}$ will be small. However, we do not give any bound on how small the step size $\eta$ needs to be, and how a small $v^{(t)}$ leads to a small classification error. These are addressed in the proof of the general case in the next section, but here we are content with an eventually small $v^{(t)}$ for a sufficiently small $\eta$.

*Proof of Lemma A.4.* By Lemma A.3, we know that there are at least $\Omega\left(\frac{\delta}{kl}\right)$ fraction of $r \in [m]$ such that

$$\left\|\frac{\tilde{\partial} L(w^{(t)})}{\partial w_r}\right\|_2 = \tilde{\Omega}\left(\frac{v^{(t)}\delta}{kl}\right).$$

Now combine with Lemma A.1. If we pick $\tau = O\left(\frac{\sigma\delta}{k^2l^2}\right)$, then at least $\Omega\left(\frac{\delta}{kl}\right)$ fraction of $r \in [m]$ have

$$\left\|\frac{\partial L(w^{(t)})}{\partial w_r}\right\|_2 = \tilde{\Omega}\left(\frac{v^{(t)}\delta}{kl}\right).$$

Thus, for a sufficiently small $\eta$, we have:

$$L(w^{(t)}) - L(w^{(t+1)}) = \eta\tilde{\Omega}\left(\left(\frac{v^{(t)}\delta}{kl}\right)^2\frac{\delta m}{kl}\right).$$

By the property of the initialization, we know that $L(w^{(0)}) = \tilde{O}(1)$. This implies that for every $t = \tilde{O}\left(\frac{\tau}{\eta}\right) = \tilde{O}\left(\frac{\sigma\delta}{k^2l^2\eta}\right)$ we have:

$$\sum_{s=1}^{t}\left(v^{(s)}\right)^2 = \tilde{O}\left(\frac{k^3l^3}{\delta^3\eta m}\right).$$

Now, we can take $T = \tilde{\Theta}\left(\frac{\sigma\delta}{k^2l^2\eta}\right)$ to obtain

$$\frac{1}{T}\sum_{t=1}^{T}\left(v^{(t)}\right)^2 = \tilde{O}\left(\frac{k^5l^5}{\delta^4\sigma m}\right).$$

This completes the proof. □

### A.4 Technical Lemmas

The following lemma above non-smooth convex function v.s. linear function is needed in the proof.

**Lemma A.5.** *Let $\phi : \mathbb{R} \to \mathbb{R}$ be a convex function that is non-smooth at $0$. Let $\partial\phi(0)$ be the set of partial gradient of $\phi$ at $0$. Define*

$$\partial_{\max}\phi(0) = \max\{\partial\phi(0)\}, \quad \partial_{\min}\phi(0) = \min\{\partial\phi(0)\}.$$

*We have for every $\tau \geq 0$, for every linear function $l(\alpha)$:*

$$\int_{-\tau}^{\tau} |\phi(\alpha) - l(\alpha)| d\alpha \geq \frac{\tau^2(\partial_{\max}\phi(0) - \partial_{\min}\phi(0))}{8}.$$

*Moreover,*

$$\Pr_{\alpha\sim U(-\tau,\tau)}\left[|\phi(\alpha) - l(\alpha)| \geq \frac{\tau(\partial_{\max}\phi(0) - \partial_{\min}\phi(0))}{128}\right] \geq \frac{1}{16}.$$

*Proof of Lemma A.5.* Without loss of generality (up to subtracting a linear function on $\phi$), let us assume that $\phi(0) = 0$ and $l(\alpha) = -b$.

Moreover, denote $\rho = \partial_{\max}\phi(0) - \partial_{\min}\phi(0) \geq 0$, we know that at least one of the following is true:

1. $\partial_{\max}\phi(0) \geq \frac{\rho}{2}$,
2. $\partial_{\min}\phi(0) \leq -\frac{\rho}{2}$.

We shall give the proof for the case $\partial_{\max}\phi(0) \geq \frac{\rho}{2}$. The other case follows from replacing $\phi$ with $-\phi$.

Let us then consider the following two cases.

1. $b > 0$, in this case, by convexity of $\phi(\alpha)$ we have that $\forall\alpha > 0 : \phi(\alpha) > 0$. Thus,
$$\int_{-\tau}^{\tau} |\phi(\alpha) - l(\alpha)| d\alpha \geq \int_{0}^{\tau} \phi(\alpha) d\alpha \geq \frac{\rho}{4}\tau^2$$
2. $b < 0$, in this case, $\phi(\alpha)$ intersects with 0 at a point $\alpha_0 \geq 0$. Consider two cases:
   (a) $\alpha_0 \geq \frac{\tau}{2}$, then we have: $b \leq -\frac{\rho\tau}{4}$. Thus,
   $$\int_{-\tau}^{\tau} |\phi(\alpha) - l(\alpha)| d\alpha \geq \int_{0}^{\min\{\alpha_0,\tau\}} -\phi(\alpha) d\alpha \geq \frac{\rho}{8}\tau^2$$
   (b) $\alpha_0 \leq \frac{\tau}{2}$, then we have:
   $$\int_{-\tau}^{\tau} |\phi(\alpha) - l(\alpha)| d\alpha \geq \int_{\alpha_0}^{\tau} \phi(\alpha) d\alpha \geq \frac{\rho}{8}\tau^2$$

This completes the proof of the first claim. For the second claim, in case 1, we know that every $\alpha \in [\tau/2, \tau]$ would have $|\phi(\alpha) - l(\alpha)| \geq \frac{\tau\rho}{128}$. In case 2(a), every $\alpha \in [0, \alpha_0 - \tau/4]$ satisfies this claim. In case 2(b) we can take every $\alpha \in [\alpha_0 + \tau/4, \tau]$. This completes the proof. □

## B Proofs for the General Case

Recall that the loss is

$$L(w) = \frac{1}{N}\sum_{s=1}^{N} L(w, x_s, y_s)$$

where

$$L(w, x_s, y_s) = -\log o_{y_s}(x_s, w), \text{ where}$$
$$o_y(x, w) = \frac{e^{f_y(x,w)}}{\sum_{i=1}^{k} e^{f_i(x,w)}}.$$

We consider a minibatch SGD of batch size $B$, number of iterations $T = N/B$ and learning rate $\eta$ as the following process: Randomly divide the total training examples into $T$ batches, each of size $B$. Let the indices of the examples in the $t$-th batch be $\mathcal{B}_t$. The update rule is:

$$w_r^{(t+1)} = w_r^{(t)} - \eta \frac{1}{B} \sum_{s \in \mathcal{B}_t} \frac{\partial L(w^{(t)}, x_s, y_s)}{\partial w_r^{(t)}}, \forall r \in [m], \text{ where}$$

$$\frac{\partial L(w, x_s, y_s)}{\partial w_r} = \left( \sum_{i \neq y_s} a_{i,r} o_i(x_s, w) - \sum_{i \neq y_s} a_{y_s,r} o_i(x_s, w) \right) 1_{\langle w_r, x_s \rangle \geq 0} x_s.$$

The pseudo gradient on a point $(x_s, y_s)$ is defined as:

$$\frac{\tilde{\partial} L(w, x_s, y_s)}{\partial w_r} = \left( \sum_{i \neq y_s} a_{i,r} o_i(x_s, w) - \sum_{i \neq y_s} a_{y_s,r} o_i(x_s, w) \right) 1_{\langle w_r^{(0)}, x_s \rangle \geq 0} x_s.$$

The expected pseudo gradient is:

$$\frac{\tilde{\partial} L(w)}{\partial w_r} = \mathbb{E}_{(x_s, y_s)} \left[ \frac{\tilde{\partial} L(w, x_s, y_s)}{\partial w_r} \right].$$

In the following subsections, we first show that the gradient is coupled with the pseudo gradient, then show that if the classification error is large then the pseudo gradient is large, and finally prove the convergence.

### B.1 Coupling

We have the following lemma for coupling, analog to Lemma A.1.

**Lemma B.1** (Coupling)**.** *For every unit vector $x \in \mathbb{R}^d$, w.h.p. over the random initialization, for every $\tau > 0$, for every $t = \tilde{O}\left(\frac{\tau}{\eta}\right)$ we have that for at least $1 - \frac{10\tau}{\sigma}$ fraction of $r \in [m]$:*

$$\frac{\partial L(w^{(t)}, x, y)}{\partial w_r} = \frac{\tilde{\partial} L(w^{(t)}, x, y)}{\partial w_r} (\forall y \in [k]), \quad \text{and} \quad |\langle w_r^{(t)}, x \rangle| \geq \tau.$$

*Proof.* The proof follows that for Lemma A.1. □

### B.2 Expected Error Large $\implies$ Gradient Large

Following the same structure as before, we can write the expected pseudo gradient as:

$$\frac{\tilde{\partial} L(w)}{\partial \pi_r} = \sum_{i \in [k]} a_{i,r} P_{i,r}$$

where

$$P_{i,r} = \sum_{a \in [k], b \in [l]} p_{a,b} \mathbb{E}_{x_{a,b} \sim \mathcal{D}_{a,b}} \left[ v_{i,a,b}(x_{a,b}, w) 1_{\left\langle w_r^{(0)}, x_{a,b} \right\rangle \geq 0} x_{a,b} \right]$$

where $v_{s,a,b}(x_{a,b}, w)$ is defined as:

$$v_{s,a,b}(x_{a,b}, w) = \begin{cases} \frac{\sum_{i \neq a} e^{f_i(x_{a,b}, w)}}{\sum_{i=1}^k e^{f_i(x_{a,b}, w)}} & \text{if } s = a; \\ -\frac{e^{f_s(x_{a,b}, w)}}{\sum_{i=1}^k e^{f_i(x_{a,b}, w)}} & \text{otherwise.} \end{cases}$$

When clear from the context, we use $v_{s,a,b}(x_{a,b})$ for short. When the choice of $x_{a,b}$ is not important, we will also use $v_{s,a,b}$.

We would like to show that if some $\mathbb{E}[p_{a,b}v_{i,a,b}]$ is large, a good fraction of $r \in [m]$ will have large pseudo gradient. Now, the first step is to show that for any fixed $\{p_{a,b}v_{i,a,b}\}$ (that does not depend on the random initialization $w_r^{(0)}$), with good probability (over the random choice of $w_r^{(0)}$) we have that $P_{i,r}$ is large; see Lemma B.2. Then we will take a union bound over an epsilon net on $\{p_{a,b}v_{i,a,b}\}$ to show that for every $\{p_{a,b}v_{,ia,b}\}$ (that can depend on $w_r^{(0)}$), at least a good fraction of of $P_{i,r}$ is large; See Lemma B.3.

**Lemma B.2** (The geometry of $\mathbf{ReLU}$). *For any possible fixed set $\{p_{a,b}v_{1,a,b}\}$ (that does not depend on $w_r^{(0)}$) such that $\mathbb{E}[p_{1,1}v_{1,1,1}] = \max\{\mathbb{E}[p_{a,b}v_{1,a,b}]\}_{a\in[k],b\in[\ell]} = v$, we have:*

$$\Pr\left[\|P_{1,r}\|_2 = \tilde{\Omega}\left(\frac{v\delta}{kl}\right)\right] = \Omega\left(\frac{\delta}{kl}\right).$$

*Proof of Lemma B.2.* The proof is very similar to the proof of Lemma A.2.

We will actually prove that

$$h\left(w_r^{(0)}\right) = \sum_{a\in[k],b\in[l]} \mathbb{E}\left[p_{a,b}v_{1,a,b}\mathbf{ReLU}\left(\left\langle w_r^{(0)}, x_{a,b}\right\rangle\right)\right]$$

is large with good probability.

Let us denote $x^*_{a,b} = \frac{\mathbb{E}_{x_{a,b}\sim\mathcal{D}_{a,b}}[x_{a,b}]}{\|\mathbb{E}_{x_{a,b}\sim\mathcal{D}_{a,b}}[x_{a,b}]\|_2}$. Thus, we can decompose $w_r^{(0)}$ into:

$$w_r^{(0)} = \alpha x^*_{1,1} + \beta$$

where $\beta \perp x^*_{1,1}$. For every $\tau \geq 0$, consider the event $\mathcal{E}_\tau$ defined as

1. $|\alpha| \leq \tau$.
2. $$\sum_{a\in[k]\backslash[1],b\in[l]} |p_{a,b}v_{1,a,b}| 1_{|\langle\beta, x^*_{a,b}\rangle|\leq 4\tau} \leq \frac{v}{3}.$$

By the definition of initialization $w_r^{(0)}$, we know that:

$$\alpha \sim \mathcal{N}(0, \sigma^2)$$

and

$$\langle\beta, x^*_{a,b}\rangle \sim \mathcal{N}(0, (1 - \langle x^*_{a,b}, x^*_{1,1}\rangle^2)\sigma^2).$$

By assumption we can simply calculate that for every $a \in [k]\backslash[1], b \in [l]$: $1 - \langle x^*_{a,b}, x^*_{1,1}\rangle^2 \geq \delta^2$. This implies that

$$\mathbb{E}\left[1_{|\langle\beta, x^*_{a,b}\rangle|\leq 4\tau}\right] \leq \frac{4\tau}{\delta\sigma}.$$

Thus,

$$\sum_{a\in[k]\backslash[1],b\in[l]} \mathbb{E}\left[|p_{a,b}v_{1,a,b}| 1_{|\langle\beta, x^*_{a,b}\rangle|\leq 4\tau}\right] \leq \frac{4\tau}{\delta\sigma} vl.$$

With $\tau = \frac{\sigma\delta}{12l}$, we know that $\Pr[\mathcal{E}_\tau] = \Omega\left(\frac{\tau}{\sigma}\right)$. The following proof will conditional on this event $\mathcal{E}_\tau$, and then treat $\beta$ as fixed and let $\alpha$ be the only random variable. In this way, for every $\alpha$ such that $|\alpha| \leq \tau$ and for every $a \in [k]\backslash[1], b \in [l]$:

$$\begin{aligned}\left\langle w_r^{(0)}, x_{a,b}\right\rangle &= \alpha\langle x^*_{1,1}, x_{a,b}\rangle + \langle\beta, x_{a,b}\rangle \\ &= \alpha\langle x^*_{1,1}, x^*_{a,b}\rangle + \langle\beta, x^*_{a,b}\rangle + \langle w_r^{(0)}, x_{a,b} - x^*_{a,b}\rangle.\end{aligned}$$

With $|\alpha\langle x^*_{1,1}, x^*_{a,b}\rangle| \leq \tau$, and since $\mathbb{E}[\langle w_r^{(0)}, x_{a,b} - x^*_{a,b}\rangle] \leq \frac{3}{2}\sigma\lambda\delta < 2\tau$, we know that if $\left|\left\langle \beta, x^*_{a,b}\right\rangle\right| \geq 4\tau$, then

$$\mathbf{ReLU}\left(\left\langle w_r^{(0)}, x_{a,b}\right\rangle\right) = \left(\alpha\langle x^*_{1,1}, x_{a,b}\rangle + \langle \beta, x_{a,b}\rangle\right) 1_{\langle \beta, x^*_{a,b}\rangle \geq 0}$$

is a linear function for $\alpha \in [-\tau, \tau]$ with probability $\geq 2/3$.

With this information, we can rewrite $h\left(w_r^{(0)}\right)$ as:

$$\begin{aligned} h\left(w_r^{(0)}\right) = h(\alpha) := \mathbb{E}\left[p_{1,1}v_{1,1,1}\mathbf{ReLU}\left(\alpha\langle x_{1,1}, x^*_{1,1}\rangle + \langle \beta, x^*_{1,1} - x_{1,1}\rangle\right)\right] \\ + \sum_{b \geq 2} \mathbb{E}\left[p_{1,b}v_{1,1,b}\mathbf{ReLU}\left(\langle \alpha x^*_{1,1} + \beta, x_{a,b}\rangle\right)\right] + l(\alpha). \end{aligned}$$

where $l(\alpha)$ is a convex function with $\partial_{\max} l(\tau) - \partial_{\max} l(-\tau) \leq v/3$.

This time, we know that w.h.p. $\langle \beta, x^*_{1,1} - x_{1,1}\rangle = \tilde{O}(\sigma\lambda\delta) \leq \tau/4$. This implies that for function $\phi$ defined as

$$\begin{aligned} \phi(\alpha) : = \mathbb{E}\left[p_{1,1}v_{1,1,1}\mathbf{ReLU}\left(\alpha\langle x_{1,1}, x^*_{1,1}\rangle + \langle \beta, x^*_{1,1} - x_{1,1}\rangle\right)\right] \\ + \sum_{b \geq 2} \mathbb{E}\left[p_{1,b}v_{1,1,b}\mathbf{ReLU}\left(\langle \alpha x^*_{1,1} + \beta, x_{a,b}\rangle\right)\right], \end{aligned}$$

We will have $\partial_{\max}\phi(\tau/2) - \partial_{\max}\phi(-\tau/2) \geq v/2$. Now apply Lemma B.5, we can conclude from the same proof of Lemma A.2. □

Now we can take the union bound to switch the order of quantifiers. However, we cannot do a naive union bound since there are infinitely many $x_{a,b}$. Instead, we will use a sampling trick to prove the following Lemma:

**Lemma B.3.** *For every $v > 0$, for $m = \tilde{\Omega}\left(\left(\frac{kl}{v\delta}\right)^4\right)$, for every possible $\{p_{a,b}v_{i,a,b}\}$ (that depend on $a_{i,r}, w_r^{(0)}$, etc.) such that $\max\{\mathbb{E}[p_{a,b}v_{i,a,b}]\}_{i,a \in [k], b \in [l]} = v$, there exists at least $\Omega\left(\frac{\delta}{kl}\right)$ fraction of $r \in [m]$ such that*

$$\left\|\frac{\tilde{\partial} L(w)}{\partial w_r}\right\|_2 = \tilde{\Omega}\left(\frac{v\delta}{kl}\right).$$

This lemma implies that if the classification error is large, then many $w_r$'s have a large pseudo gradient.

*Proof of Lemma A.3.* We first pick $S$ samples $\mathcal{S} = \{x^{(s)}_{a,b}\}$, with $p_{a,b}S$ many from distribution $\mathcal{D}_{a,b}$, and with the corresponding value function $v^{(s)}_{i,a,b}$. Since each $v^{(s)}_{i,a,b} \in [-1, 1]$, we know that w.h.p., for every $i \in [k], a \in [k], b \in [l]$:

$$\left|\mathbb{E}[p_{a,b}v_{i,a,b}] - \frac{1}{p_{a,b}S}\sum_s p_{a,b}v^{(s)}_{i,a,b}\right| = \tilde{O}\left(\frac{1}{\sqrt{p_{a,b}S}}\right).$$

This implies that as long as $S = \tilde{\Omega}\left(\frac{1}{v^2}\right)$, we will have that

$$\max_{i \in [k], a \in [k], b \in [l]} \left\{\frac{1}{p_{a,b}S}\sum_s p_{a,b}v^{(s)}_{i,a,b}\right\} \in \left[\frac{1}{2}v, \frac{3}{2}v\right].$$

Thus, following the same proof as in Lemma A.3, but this time applying a union bound over $v^{(s)}_{i,a,b}$, we know that as long as $m = \tilde{\Omega}\left(\frac{Sk^2 l}{\delta}\right)$, w.h.p. for every possible choices of $v^{(s)}_{i,a,b}$, there are at least

$\Omega\left(\frac{\delta}{kl}\right)$ fraction of $r \in [m]$ such that

$$\left\|\frac{1}{S}\sum_{x_{a,b}\in\mathcal{S}}\frac{\tilde{\partial} L(w, x_{a,b}, a)}{\partial w_r}\right\|_2 = \tilde{\Omega}\left(\frac{v\delta}{kl}\right).$$

Now we consider the difference between the sample gradient and the expected gradient. Since $\left\|\frac{\tilde{\partial} L(w,x,y)}{\partial w_r}\right\|_2 \le \tilde{O}(1)$, by standard concentration bound we know that w.h.p. for every $r \in [m]$,

$$\left\|\frac{1}{S}\sum_{x_{a,b}\in\mathcal{S}}\frac{\tilde{\partial} L(w, x_{a,b}, a)}{\partial w_r} - \frac{\tilde{\partial} L(w)}{\partial w_r}\right\|_2 = \tilde{O}\left(\frac{1}{\sqrt{S}}\right).$$

This implies that as long as $S = \tilde{\Omega}\left(\left(\frac{kl}{v\delta}\right)^2\right)$, such $r \in [m]$ also have:

$$\left\|\frac{\tilde{\partial} L(w)}{\partial w_r}\right\|_2 = \tilde{\Omega}\left(\frac{v\delta}{kl}\right)$$

which completes the proof. □

### B.3 Convergence

We now show the following important lemma about convergence.

**Lemma B.4** (Convergence)**.** *Denote* $\max\{\mathbb{E}[p_{a,b} v_{i,a,b}(x_{a,b}, w^{(t)})]\}_{i,a\in[k],b\in[\ell]} = v^{(t)} = v$*, and let* $\gamma = \Omega\left(\frac{\delta}{kl}\right)$*. Then for a sufficiently small* $\eta = \tilde{O}\left(\frac{\gamma}{m}\left(\frac{v\delta}{kl}\right)^2\right)$*, if we run SGD with a batch size at least* $B_t = \tilde{\Omega}\left(\left(\frac{kl}{v\delta}\right)^4 \frac{1}{\gamma^2}\right)$ *and* $t = \tilde{O}\left(\left(\frac{v\delta}{kl}\right)^2 \frac{\sigma\gamma}{\eta}\right)$*, then w.h.p.,*

$$L(w^{(t)}) - L(w^{(t+1)}) = \eta\gamma m\tilde{\Omega}\left(\left(\frac{v\delta}{kl}\right)^2\right).$$

*Proof of Lemma B.4.* We know that for at least $\gamma$ fraction of $r \in [m]$ such that

$$\left\|\frac{\tilde{\partial} L(w^{(t)})}{\partial w_r}\right\|_2 = \tilde{\Omega}\left(\frac{v\delta}{kl}\right).$$

Note that w.h.p. over the random initialization, for every $(x, y)$, $\left\|\frac{\tilde{\partial} L(w^{(t)},x,y)}{\partial w_r}\right\|_2 \le \tilde{O}(1)$. By Hoeffding concentration, this implies that for a randomly sampled batch $\mathcal{B}_t = \{(x_1, y_1), \cdots, (x_{B_t}, y_{B_t})\}$ of size $B_t$, we have that w.h.p. over $\mathcal{B}_t$,

$$\left\|\frac{1}{B_t}\sum_{i=1}^{B_t}\frac{\tilde{\partial} L(w^{(t)}, x_i, y_i)}{\partial w_r}\right\| = \tilde{\Omega}\left(\frac{v\delta}{kl}\right) - O\left(\frac{L}{\sqrt{B_t}}\right) = \tilde{\Omega}\left(\frac{v\delta}{kl}\right).$$

On the other hand, according to Lemma B.1 with $\tau = \frac{\sigma\gamma}{100B_t}$, we know that w.h.p. over the random initialization, for *every* $x_i$ in $\mathcal{B}_t$, we have: for at least $1 - \gamma/(2B_t)$ fraction of $r \in [m]$, $\frac{\partial L(w^{(t)},x_i,y_i)}{\partial w_r} = \frac{\tilde{\partial} L(w^{(t)},x_i,y_i)}{\partial w_r}$. This implies that for at least $\gamma/2$ fraction of $r \in [m]$ such that for *every* $x_i$ in $\mathcal{B}_t$ we have $\frac{\partial L(w^{(t)},x_i)}{\partial w_r} = \frac{\tilde{\partial} L(w^{(t)},x_i)}{\partial w_r}$. Let us denote the set of these $r$ as set $\mathcal{R}$. Then for every $r \in \mathcal{R}$:

$$\left\|\frac{1}{B_t}\sum_{i=1}^{B_t}\frac{\partial L(w^{(t)}, x_i, y_i)}{\partial w_r}\right\| = \tilde{\Omega}\left(\frac{v\delta}{kl}\right).$$

For every $r \in [m]$, let us denote $\tilde{\nabla}_{t,r} = \frac{1}{B_t}\sum_{i=1}^{B_t} \frac{\partial L(w^{(t)}, x_i, y_i)}{\partial w_r}$, and $\nabla_{t,r} = \frac{\partial L(w^{(t)})}{\partial w_r}$. Then similarly as above, since $\left\| \frac{\partial L(w^{(t)}, x, y)}{\partial w_r} \right\|_2 \le \tilde{O}(1)$, by Hoeffding concentration, we have

$$\begin{aligned} \|\nabla_{t,r} - \tilde{\nabla}_{t,r}\|_2 &= \tilde{O}\left(\frac{1}{\sqrt{B_t}}\right), \\ \|\nabla_{t,r}\|_2 &= \tilde{\Omega}\left(\frac{v\delta}{kl}\right) - \tilde{O}\left(\frac{1}{\sqrt{B_t}}\right) = \tilde{\Omega}\left(\frac{v\delta}{kl}\right). \end{aligned}$$

Now we consider the non-smooth gradient descent. Consider a newly sampled point $(x', y')$, and let us denote

$$\tilde{\nabla}'_{t,r} = \frac{\partial L(w^{(t)}, x', y')}{\partial w_r}.$$

By Lemma B.1, we know that w.h.p. over the random initialization, at least $1 - \frac{10\tau}{\sigma}$ fraction of $r$ satisfies $\langle w_r, x' \rangle \ge \tau$. Let us denote the set of these $r$'s as $\mathcal{S}_r$. We know that on these sets, the function is $\tilde{O}(1)$ smooth and $\tilde{O}(1)$ Lipschitz smooth. By Lemma B.6,

$$\begin{aligned} \Delta_t &:= L(w^{(t)} - \eta\tilde{\nabla}_t, x', y') - L(w^{(t)}, x', y') \\ &\le -\eta \sum_{r \in \mathcal{S}_r} \langle \tilde{\nabla}_{t,r}, \tilde{\nabla}'_{t,r} \rangle + \sum_{r \in [m] \setminus \mathcal{S}_r} \tilde{O}(\eta) + \tilde{O}\left(\eta^2 m^2\right) \\ &\le -\eta \sum_{r \in [m]} \langle \tilde{\nabla}_{t,r}, \tilde{\nabla}'_{t,r} \rangle + \tilde{O}\left(\frac{\eta\tau m}{\sigma}\right) + \tilde{O}\left(\eta^2 m^2\right). \end{aligned} \tag{3}$$

Let $G_1$ denote the event that (3) holds.

Note that w.h.p. over the random initialization, $|L(w^{(t)}, x', y')| = \tilde{O}(L\eta tmk) = \tilde{O}(m)$, and $\|\tilde{\nabla}_{t,r,i}\| \le \tilde{O}(1)$, $\|\tilde{\nabla}'_{t,r}\| \le \tilde{O}(1)$ for all $(x_i, y_i)$'s and $(x', y')$. Let $G_0$ denote this event.

Then we have $P[\neg G_0]$ and $P[\neg G_1]$ bounded by $1/\text{poly}(k, l, m, 1/\delta, 1/\epsilon)$. Conditioned on $G_0$, we have $\nabla_{t,r} = \mathbb{E}_{(x',y')}\left[\tilde{\nabla}'_{t,r} | G_0\right]$ and $L(w^{(t)}) - L(w^{(t+1)}) = \mathbb{E}_{(x',y')}\left[\Delta_t | G_0\right]$ where the expectation is over $(x', y')$. Now we have

$$\nabla_{t,r} = \mathbb{E}_{(x',y')}\left[\tilde{\nabla}'_{t,r} | G_0, G_1\right] P[G_1|G_0] + \mathbb{E}_{(x',y')}\left[\tilde{\nabla}'_{t,r} | G_0, \neg G_1\right] P[\neg G_1 | G_0].$$

So

$$\left\| \nabla_{t,r} - \mathbb{E}_{(x',y')}\left[\tilde{\nabla}'_{t,r} | G_0, G_1\right] \right\|_2 = \frac{1}{\text{poly}(k, l, m, 1/\delta, 1/\epsilon)}.$$

Then

$$\begin{aligned} L(w^{(t)}) - L(w^{(t+1)}) &= \mathbb{E}_{(x',y')}\left[\Delta_t | G_0, G_1\right] P[G_1|G_0] + \mathbb{E}_{(x',y')}\left[\Delta_t | G_0, \neg G_1\right] P[\neg G_1|G_0] \\ &\ge \frac{\eta}{2} \sum_{r \in [m]} \langle \tilde{\nabla}_{t,r}, \mathbb{E}_{(x',y')}\left[\tilde{\nabla}'_{t,r} | G_0, G_1\right] \rangle - \tilde{O}\left(\eta^2 m^2\right) - \tilde{O}\left(\frac{\eta\tau m}{\sigma}\right) \\ &\quad - \frac{\tilde{O}(m)}{\text{poly}(k, l, m, 1/\delta, 1/\epsilon)} \\ &\ge \frac{\eta}{2} \sum_{r \in [m]} \langle \tilde{\nabla}_{t,r}, \nabla_{t,r} \rangle - \tilde{O}\left(\eta^2 m^2\right) - \tilde{O}\left(\frac{\eta\tau m}{\sigma}\right) \\ &\quad - \frac{\tilde{O}(m)}{\text{poly}(k, l, m, 1/\delta, 1/\epsilon)} - \frac{\tilde{O}(\eta m)}{\text{poly}(k, l, m, 1/\delta, 1/\epsilon)} \\ &\ge \frac{\eta}{2} \sum_{r \in [m]} \langle \tilde{\nabla}_{t,r}, \nabla_{t,r} \rangle - \tilde{O}\left(\eta^2 m^2\right) - \tilde{O}\left(\frac{\eta\tau m}{\sigma}\right). \end{aligned}$$

Note that $\tilde{\nabla}_{t,r}$ concentrates around $\nabla_{t,r}$. This leads to w.h.p. when $\eta = \tilde{O}\left(\frac{\gamma}{m}\left(\frac{v\delta}{kl}\right)^2\right)$, $\tau = \tilde{O}\left(\gamma\left(\frac{v\delta}{kl}\right)^2\sigma\right)$, and $B_t = \tilde{\Omega}\left(\left(\frac{kl}{v\delta}\right)^4\frac{1}{\gamma^2}\right)$,

$$\begin{aligned}
L(w^{(t)}) - L(w^{(t+1)}) &\geq \sum_{r=1}^{m}\frac{\eta}{2}\|\tilde{\nabla}_{t,r}\|_2^2 - \tilde{O}(\eta^2 m^2) - \tilde{O}\left(\frac{\eta\tau m}{\sigma}\right) - \eta\tilde{O}\left(\frac{m}{\sqrt{B_t}}\right)\\
&\geq \eta\gamma m\tilde{\Omega}\left(\left(\frac{v\delta}{kl}\right)^2\right) - \tilde{O}(\eta^2 m^2) - \tilde{O}\left(\frac{\eta\tau m}{\sigma}\right) - \eta\tilde{O}\left(\frac{m}{\sqrt{B_t}}\right)\\
&\geq \eta\gamma m\tilde{\Omega}\left(\left(\frac{v\delta}{kl}\right)^2\right).
\end{aligned}$$

This completes the proof. □

Now we can prove the main theorem.

**Theorem 4.1.** *Suppose the assumptions **(A1)(A2)(A3)** are satisfied. Then for every $\varepsilon > 0$, there is $M = poly(k, l, 1/\delta, 1/\varepsilon)$ such that for every $m \geq M$, after doing a minibatch SGD with batch size $B = poly(k, l, 1/\delta, 1/\varepsilon, \log m)$ and learning rate $\eta = \frac{1}{m \cdot poly(k,l,1/\delta,1/\varepsilon,\log m)}$ for $T = poly(k, l, 1/\delta, 1/\varepsilon, \log m)$ iterations, with high probability:*

$$\Pr_{(x,y)\sim\mathcal{D}}\left[\forall j \in [k], j \neq y, f_y(x, w^{(T)}) > f_j(x, w^{(T)})\right] \geq 1 - \varepsilon.$$

*Proof of Theorem 4.1.* Let $v_{i,a,b}^{(t)}$ denote $v_{i,a,b}(x_{a,b}, w^{(t)})$.

First, we will show that if $\Pr_{(x,y)\sim\mathcal{D}}\left[\forall j \in [k], j \neq y, f_y(x, w^{(t)}) > f_j(x, w^{(t)})\right] \leq 1 - \varepsilon$, there must be one $a, b$ such that $\mathbb{E}[v_{i,a,b}^{(t)}] \geq \varepsilon^2$. Let us denote $\max\{\mathbb{E}[p_{a,b}v_{i,a,b}^{(t)}]\} = v^{(t)} = v$. For a particular $a \in [k], b \in [l]$, for any $x_{a,b}$ from $\mathcal{D}_{a,b}$, by definition,

$$v_{a,a,b}(x_{a,b}, w^{(t)}) = 1 - \frac{e^{f_a(x_{a,b},w^{(t)})}}{\sum_{i=1}^{k} e^{f_i(x_{a,b},w^{(t)})}}.$$

Then for every $\varepsilon \leq \frac{1}{e}$, if $v_{a,a,b}^{(t)}(x_{a,b}, w^{(t)}) \leq \varepsilon$, then

$$\forall i \in [k], i \neq a : f_a(x_{a,b}, w^{(t)}) \geq f_i(x_{a,b}, w^{(t)}) + 1,$$

which implies that the prediction is correct. So if $\mathbb{E}[v_{a,a,b}^{(t)}] \leq \varepsilon^2$, then there are at most $\varepsilon$ fraction of $x_{a,b}$ such that $f_a(x_{a,b}, w^{(t)}) \leq f_i(x_{a,b}, w^{(t)})$ for some $i \neq a$. In other words, if $\Pr_{(x,y)\sim\mathcal{D}}\left[\forall j \in [k], j \neq y, f_y(x, w^{(t)}) > f_j(x, w^{(t)})\right] \leq 1 - \varepsilon$, there must be some $i, a, b$ such that $\mathbb{E}[v_{i,a,b}^{(t)}] \geq \varepsilon^2$.

Now, consider two cases:

1. $p_{a,b} \leq \frac{\varepsilon}{2kl}$. For all such $a, b$, even if all the predictions are wrong, it will only increase the total error by $\varepsilon/2$ so the other half $\varepsilon/2$ error must come from other $p_{a,b}$.

2. $p_{a,b} \geq \frac{\varepsilon}{2kl}$, which means that $\mathbb{E}[p_{a,b}v_{i,a,b}^{(t)}] \geq \frac{\varepsilon}{2kl}\mathbb{E}[v_{i,a,b}^{(t)}] \geq \frac{\varepsilon^3}{8kl}$. Thus, $\max\{\mathbb{E}[p_{a,b}v_{i,a,b}^{(t)}]\} = v^{(t)} = v \geq \frac{\varepsilon^3}{8kl}$.

Therefore, to prove the theorem, it suffices to show that $v^{(t)}$ will be smaller than $\frac{\varepsilon^3}{8kl}$ after a proper amount of iterations. Suppose $v^{(t)} \geq \frac{\varepsilon^3}{8kl}$, then by Lemma B.4, as long as

$$t = \tilde{O}\left(\frac{\sigma}{\eta}\frac{\delta^3\varepsilon^6}{k^5\ell^5}\right), \tag{4}$$

we have:

$$L(w^{(t)}) - L(w^{(t+1)}) \geq \tilde{O}\left(\eta m \frac{\delta^3 \varepsilon^6}{k^5 \ell^5}\right).$$

Note that by the random initialization, originally for each $f_i$ we have: for every unit vector $x \in \mathbb{R}^d$, $\langle w_r^{(0)}, x \rangle \sim \mathcal{N}(0, \sigma^2)$. Thus, with $\sigma = \frac{1}{\sqrt{m}}$ and $a_{i,r} \sim \mathcal{N}(0,1)$, an elementary calculation shows that w.h.p.,

$$|f_i(x, w^{(0)})| = \left| \sum_{r \in [m]} a_{i,r} \mathbf{ReLU}(\langle w_r^{(0)}, x \rangle) \right| = \tilde{O}(1).$$

Thus, $L(w^{(0)}) = \tilde{O}(1)$. Since $L(w) \geq 0$, we know that $L(w^{(t)}) - L(w^{(t+1)}) \geq \tilde{O}\left(\eta m \frac{\delta^3 \varepsilon^6}{k^5 \ell^5}\right)$ can happen for at most

$$\tilde{O}\left(\frac{1}{\eta m} \frac{k^5 \ell^5}{\delta^3 \varepsilon^6}\right)$$

iterations. By our choice of $\eta$, we know that $\eta m = \tilde{O}\left(\frac{\delta^3 \varepsilon^6}{k^5 \ell^5}\right)$, so we need at most $T = \tilde{O}\left(\frac{k^{10} \ell^{10}}{\delta^6 \varepsilon^{12}}\right)$ iterations.

To this end, we just need

$$\frac{\sigma}{\eta} \frac{\delta^3 \varepsilon^6}{k^5 \ell^5} = \tilde{\Omega}\left(\frac{1}{\eta m} \frac{k^5 \ell^5}{\delta^3 \varepsilon^6}\right)$$

to make sure (4) holds so that we can keep the coupling before convergence. This is true as long as $m = \tilde{\Omega}\left(\frac{k^{20} \ell^{20}}{\delta^{12} \varepsilon^{24}}\right)$. □

### B.4 Technical Lemmas

The following lemma above non-smooth convex function v.s. linear function is needed in the proof.

**Lemma B.5.** *Let $\phi : \mathbb{R} \to \mathbb{R}$ be a convex function. Let $\partial \phi(x)$ be the set of partial gradient of $\phi$ at $x$. Define*

$$\partial_{\max} \phi(x) = \max\{\partial \phi(x)\}, \quad \partial_{\min} \phi(x) = \min\{\partial \phi(x)\}.$$

*We have that for every $\tau \geq 0$, for every convex function $l(\alpha)$, let $\gamma = (\partial_{\max} \phi(\tau/2) - \partial_{\min} \phi(-\tau/2)) - (\partial_{\max} l(\tau) - \partial_{\min} l(-\tau))$, then*

$$\int_{-\tau}^{\tau} |\phi(\alpha) - l(\alpha)| d\alpha \geq \frac{\tau^2 \gamma}{32}$$

*and*

$$\Pr_{a \sim U(-\tau, \tau)}\left[|\phi(\alpha) - l(\alpha)| \geq \frac{\tau \gamma}{512}\right] \geq \frac{1}{64}.$$

*Proof.* Without loss of generality, we can assume that either $\partial_{\max} l(\tau)$ and $\partial_{\max} \phi(\tau/2) \geq \gamma/2$ , or $\partial_{\min} l(-\tau) = 0$ and $\partial_{\min} \phi(-\tau/2) \leq -\gamma/2$. The lemma can be proved using the same argument as in Lemma A.5. □

We also need the following lemma regarding the gradient descent on non-smooth function.

**Lemma B.6.** *Suppose for every $i \in [m]$, $g_i : \mathbb{R}^d \to \mathbb{R}$ is a L-Lipschitz smooth function. Moreover, suppose for an $r \in [m]$, for all $i \in [m-r]$ we have that $g_i$ is also L-smooth. Suppose $g : \mathbb{R} \to \mathbb{R}$ is L-smooth and L-Lipschitz smooth, and let $f(w)$ denote $g(\sum_{i \in [m]} g_i(w_i))$. Then for every $w, \delta \in \mathbb{R}^{dm}$ with $\|\delta_i\|_2 \leq p$ we have:*

$$g\left(\sum_{i \in [m]} g_i(w_i + \delta_i)\right) - g\left(\sum_{i \in [m]} g_i(w_i)\right) \leq \sum_{i \in [m-r]} \left\langle \frac{\partial f(w)}{\partial w_i}, \delta_i \right\rangle + L^3 m^2 p^2 + L^2 r p.$$

*Proof of Lemma B.6.* The proof of this lemma follows directly from

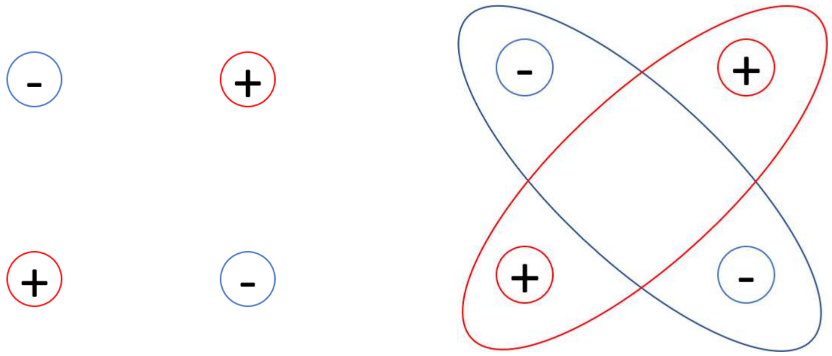
$$
\begin{aligned}
& g\left(\sum_{i\in[m]} g_i(w_i+\delta_i)\right) - g\left(\sum_{i\in[m]} g_i(w_i)\right) \\
& \leq g\left(\sum_{i\in[m-r]} g_i(w_i+\delta_i) + \sum_{i>m-r} g_i(w_i)\right) - g\left(\sum_{i\in[m]} g_i(w_i)\right) \\
& \quad + L\left|\sum_{i>m-r} g_i(w_i) - \sum_{i>m-r} g_i(w_i+\delta_i)\right| \\
& \leq g\left(\sum_{i\in[m-r]} g_i(w_i+\delta_i) + \sum_{i>m-r} g_i(w_i)\right) - g\left(\sum_{i\in[m]} g_i(w_i)\right) + L^2pr \\
& \leq \left\langle \nabla g\left(\sum_{i\in[m]} g_i(w_i)\right), \sum_{i\in[m-r]} g_i(w_i+\delta_i) - \sum_{i\in[m-r]} g_i(w_i)\right\rangle \\
& \quad + \frac{L}{2}\left\|\sum_{i\in[m-r]} g_i(w_i+\delta_i) - \sum_{i\in[m-r]} g_i(w_i)\right\|^2 + L^2pr \\
& \leq \left\langle \nabla g\left(\sum_{i\in[m]} g_i(w_i)\right), \sum_{i\in[m-r]} g_i(w_i+\delta_i) - \sum_{i\in[m-r]} g_i(w_i)\right\rangle + L^3m^2p^2 + L^2pr \\
& \leq \sum_{i\in[m-r]} \left\langle \frac{\partial f(w)}{\partial w_i}, \delta_i \right\rangle + L^3m^2p^2 + L^2pr
\end{aligned}
$$

where the last line follows from the chain rule and Lipschitz smoothness, and the last to second line follows from

$$
\left|\sum_{i\in[m-r]} g_i(w_i+\delta_i) - \sum_{i\in[m-r]} g_i(w_i)\right| \leq Lpm.
$$

This completes the proof. □

# C Illustration of the Separability Assumption



(a) Each class as two components

(b) Each class as one component

Figure 3: Illustration of the separability assumption. The data lie in $\mathcal{R}^2$ and are from two classes $-$ and $+$. The $+$ class contains points uniformly over two balls of diameter $1/10$ with centers $(0, 0)$ and $(2, 2)$, and the $-$ class contains points uniformly over two balls of the same diameter with centers $(0, 2)$ and $(2, 0)$. (a) We can view each ball in each class as one component, then the data will satisfy the separability assumption with $\ell = 2$. (b) We can also view each class as just one component, but the data will not satisfy the separability assumption with $\ell = 1$.

Recall the separability assumption introduced in Section 3:

**(A1)** (Separability) There exists $\delta > 0$ such that for every $i_1 \neq i_2 \in [k]$ and every $j_1, j_2 \in [l]$,

$$\text{dist}\left(\text{supp}(\mathcal{D}_{i_1,j_1}), \text{supp}(\mathcal{D}_{i_2,j_2})\right) \geq \delta.$$

Moreover, for every $i \in [k], j \in [l]$,

$$\text{diam}(\text{supp}(\mathcal{D}_{i,j})) \leq \lambda\delta, \text{ for } \lambda \leq 1/(8l).$$

In this assumption, each class can contain multiple components when $\ell \geq 2$. This allows more flexibility and also allows non-linearly separable data. See Figure 3 for such an example. The data lie in $\mathcal{R}^2$ and are from two classes $-$ and $+$. The $+$ class contains points uniformly over two balls of diameter $1/10$ with centers $(0, 0)$ and $(2, 2)$, and the $-$ class contains points uniformly over two balls of the same diameter with centers $(0, 2)$ and $(2, 0)$. As illustrated in Figure 3(a), the data satisfy the separability assumption with $\ell = 2$: each ball in each class is viewed as one component, then the distance between any two points in one component is at most $1/10$ while the distance between any two points from different components will be at least $19/10$. However, as illustrated in Figure 3(b), the data do not satisfy the separability assumption with $\ell = 1$, by viewing each class as just one component. This demonstrates that allowing $\ell \geq 2$ leads to more flexibility. Furthermore, the data are clearly not linearly separable, showing that the assumption captures nonlinear structures of practical data better than linear separability.

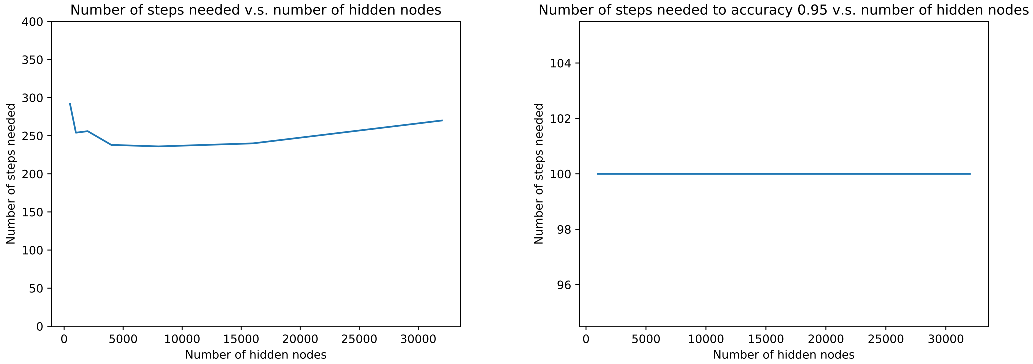
# D Additional Experimental Results

Here we provide some additional experimental results.

## D.1 Statistics When Achieving A Small Error v.s. Number of Hidden Nodes

Recall that our analysis that for a learning rate decreasing with the number of hidden nodes $m$, the number of iterations to get the accuracy roughly remain the same. A more direct way to check is to plot the number of steps to achieve the accuracy for different $m$. As shown in Figure 4, the number of steps roughly match what our theory predicts.

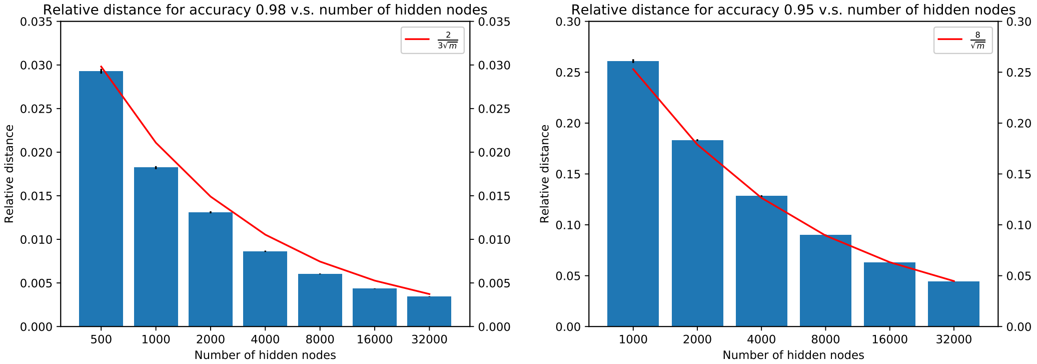
Furthermore, Figure 5 shows the relative distances when achieving the desired accuracies. It is observed that the distances scale roughly as $O(1/\sqrt{m})$. In particular, they closely match $2/3\sqrt{m}$ on the synthetic data and $8/\sqrt{m}$ on MNIST (the red lines in the figures), where $m$ is the number of hidden nodes. Explanations are left for future work.



(a) On synthetic data

(b) on MNIST

Figure 4: Number of steps to achieve $98\%$ on the synthetic data and $95\%$ test accuracy on MNIST for different values of number of hidden nodes. They are roughly the same for different number of hidden nodes.



(a) On synthetic data

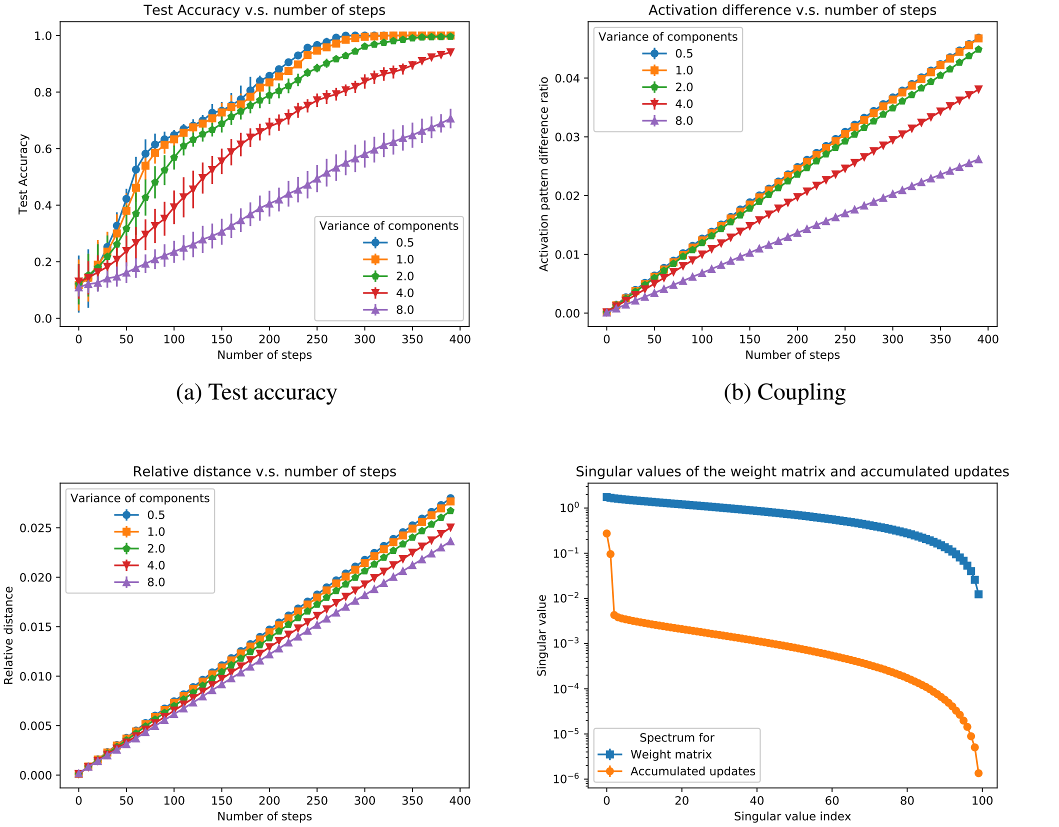
(b) on MNIST

Figure 5: Relative distances when achieving $98\%$ on the synthetic data and $95\%$ test accuracy on MNIST for different values of number of hidden nodes. They closely match $2/3\sqrt{m}$ on the synthetic data and $8/\sqrt{m}$ on MNIST (the red lines), where $m$ is the number of hidden nodes.

### D.2 Synthetic Data with Larger Variances

Here we test the effect of the in-component variance on the learning process. First recall that the synthetic data are of 1000 dimension and consist of $k = 10$ classes, each having $\ell = 2$ components. Each component is of equal probability $1/(kl)$, and is a Gaussian with covariance $\sigma/\sqrt{d}I$ and its mean is i.i.d. sampled from a Gaussian distribution $\mathcal{N}(0, \sigma_0/\sqrt{d})$. 1000 training data points and 1000 test data points are sampled. Here we fix $\sigma_0 = 5$ and vary $\sigma$ and plot the test accuracy, the coupling, the distance across different time steps, and the spectrum of the final solution.

Figure 6 shows that the test accuracy decreases with increasing variance $\sigma$, and it takes longer time to get a good solution. On the other hand, an increasing variance does not change the trends for activation patterns, distance, and the rank of the weight matrix. This is possibly due to that the signal in the updates remain small with increasing variances, while the noise in the updates act similarly as the randomness in the weights.



(a) Test accuracy

(b) Coupling

(c) Distance from the initialization

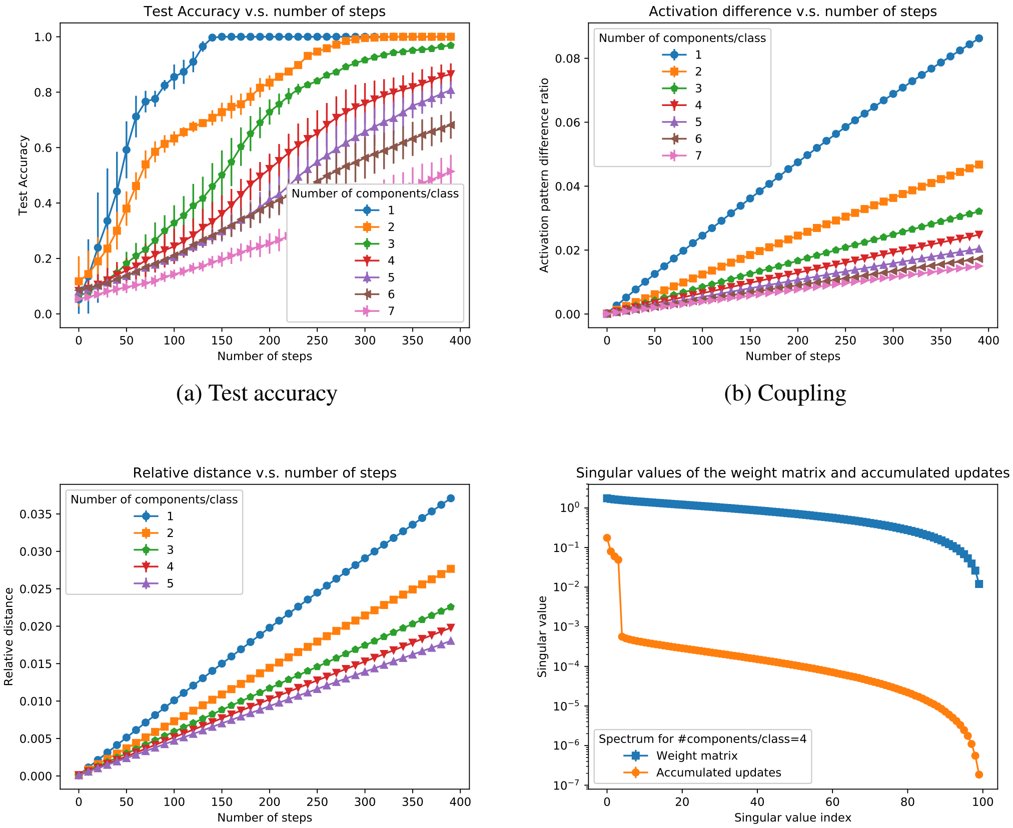
(d) Rank of accumulated updates ($y$-axis in log-scale)

Figure 6: Results for synthetic data with different variances.

### D.3 Synthetic Data with Larger Number of Components in Each Class

Here we test the effect of the number of components in each class on the learning process. First recall that the synthetic data are of 1000 dimension and consist of $k = 10$ classes, each having $\ell$ components. Each component is of equal probability $1/(kl)$, and is a Gaussian with covariance $1/\sqrt{d}I$ and its mean is i.i.d. sampled from a Gaussian distribution $\mathcal{N}(0, 5/\sqrt{d})$. 1000 training data points and 1000 test data points are sampled. Here we vary $\ell$ from 1 to 7 and plot the test accuracy, the coupling, the distance across different time steps, and the spectrum of the final solution.

Figure 7 shows that the test accuracy decreases with increasing number of components $\ell$ in each class, and it takes longer time to get a good solution. On the other hand, a larger $\ell$ leads to more significant coupling and smaller relative distances at the same time step. This is probably because the learning makes less progress due to the more complicated structure of the data.



(a) Test accuracy

(b) Coupling

(c) Distance from the initialization

(d) Rank of accumulated updates for 4 components in each class ($y$-axis in log-scale)

Figure 7: Results for synthetic data with larger number of components in each class.