Inverse Problems and Imaging
Volume 15, No. 1, 2021, 41–62
doi:10.3934/ipi.2020077

GLOBAL CONVERGENCE AND GEOMETRIC CHARACTERIZATION OF SLOW TO FAST WEIGHT EVOLUTION IN NEURAL NETWORK TRAINING FOR CLASSIFYING LINEARLY NON-SEPARABLE DATA

Ziang Long*

Department of Mathematics
University of California, Irvine, CA 92697, USA

Penghang Yin

Department of Mathematics and Statistics
State University of New York at Albany, Albany, NY 12222, USA

Jack Xin

Department of Mathematics
University of California, Irvine, Irvine, CA 92697, USA

Abstract. In this paper, we study the dynamics of gradient descent in learning neural networks for classification problems. Unlike in existing works, we consider the linearly non-separable case where the training data of different classes lie in orthogonal subspaces. We show that when the network has sufficient (but not exceedingly large) number of neurons, (1) the corresponding minimization problem has a desirable landscape where all critical points are global minima with perfect classification; (2) gradient descent is guaranteed to converge to the global minima. Moreover, we discovered a geometric condition on the network weights so that when it is satisfied, the weight evolution transitions from a slow phase of weight direction spreading to a fast phase of weight convergence. The geometric condition says that the convex hull of the weights projected on the unit sphere contains the origin.

1. **Introduction.** Deep neural networks (DNN) have achieved remarkable performances in image and speech classification tasks among other AI applications in recent years; for examples, see [8, 9, 15, 17]. Although there have been numerous theoretical contributions to understand their success, the learning process in the actual network training remains largely empirical. One interesting phenomenon is that over-parametrized DNN's trained by stochastic gradient descent generalize [13, 14] instead of overfitting the training data contrary to conventional statistical learning. Though several convergence results are proved in the over-parameterized regime for deep networks [6, 11, 1], the network weights move only in a small neighborhood of the random initialization and so their dynamics are very localized. Partly, this may be attributed to the exceedingly large number of neurons in convergence theory, far surpassing what is used in practice where the weights evolve significantly

2020 *Mathematics Subject Classification.* 90C26, 68W40.
Key words and phrases. Neural networks, learning dynamics, geometric condition, slow-to-fast convergence, classification, gradient descent.
* Corresponding author: Ziang Long.

from random start through hundreds of epochs in training to reach best prediction accuracy.

Our work here addresses how the weights evolve towards a global minimum of loss function as the number of neurons increases from the feature dimension (the least necessary) to the over-parametrized regime. To facilitate analysis, our model network structure is motivated by [4] on classifying linearly separable data. We instead study a linearly non-separable multi-category classification problem with an emphasis on the dynamics of weights in terms of the two time scales of evolution and a geometric characterization of the transition time. Our training data of the two classes will lie in orthogonal sub-spaces, which extends the data configuration in [3] where the subspace of each class is one dimensional for an XOR detection problem. Orthogonality of input data from the two classes implies that the training process in each class can be analyzed independently of the other. In the one-dimensional case [3], each weight update does not increase the loss on any sample point. In the multi-dimensional case here, we find that during gradient descent weight update, it is not possible that the loss is non-increasing in the point-wise sense (on each input data). Instead, the population loss is decreasing (i.e. in the sense of expectation). The population loss here is based on the hinge loss function and the network activation function is ReLU. Under a mild non-degenerate data condition, we prove that all critical points of our non-convex and non-smooth population loss function are global minima. Similar landscapes (a local minimum is a global minimum) are known for deep networks with activation functions that are either strictly convex [12], or real analytic and strictly increasing [14].

1.1. **Prior works and our contributions.** In DNN training, one observes that the network learning consists of alternating phases: plateaus where the validation error remains fairly constant and periods of rapid improvement where a lot of progress is made over a few epochs. Prior to our work, [5] studied slow and fast weight dynamics in a solvable model while minimizing a binary cross entropy or hinge loss function on linearly separable data. In the regression context, [2] came across such two time-scale phenomenon in training a two-linear-layer convolutional network with prescribed ground truth and unit Gaussian input data. This particular data assumption makes it possible to readily derive the closed-form expressions of the population loss and gradient, and then analyze the energy landscape and convergence of the gradient descent algorithm.

In this work, we study network weight dynamics in training a one-hidden-layer ReLU network via hinge loss minimization on multi-category classification of linearly non-separable data lying in n orthogonal sub-spaces. Our main contributions are:

- We discovered a geometric condition (GC) to characterize the transition time T from the first (slow) phase of weight evolution to the second fast weight convergence. The condition says that the convex hull of the weights on the unit sphere contains the origin, see Fig. 1 for an illustration. Equivalent geometric conditions are also derived (Lemma 1). In the first (slow) phase, the weight directions spread out over the unit sphere to satisfy GC.
- We obtain upper bound on T in terms of data distribution function provided that the network weights are uniformly bounded during training which we observed numerically.

- We give probabilistic bounds on the validity of geometric condition for random initialization, which suggests that the larger the number of neurons, the more likely GC holds and the earlier the fast phase of evolution begins.
- We prove the global convergence of gradient descent training algorithm under the uniformly bounded weight assumption. In case of positive network bias, we prove a global Lipschitz gradient property of the loss function and subsequential convergence of weights to a global minimum. In case of zero network bias, we prove that the loss function has Lipschitz gradient away from the origin and is piece-wise C^1.
- We prove that all critical points of the population loss function are global minima under a non-degenerate data condition.
- We provide numerical examples to substantiate our theory, extend the data assumption, and illustrate the weight dynamics as the network size increases towards the over-parametrized regime. We visualize the feature and weight vectors in DNNs on MNIST data in connection with our model findings.

Organization. In section 2, we introduce the settings of the classification problem, including the assumptions on the data and network architecture. In section 3, we state the main results regarding the convergence guarantee of the gradient descent algorithm for training the neural net in the cases of with and without a bias term in the linear layer. In section 4, we present preliminaries about the landscape of the training loss function. The convergence analysis of main results will be sketched in section 5. In section 6, we substantiate our theoretical findings with numerical simulations. All the technical proofs are detailed in the appendix.

Notations. We denote by $\mathcal{S}^{d-1}$ the unit sphere in $\mathbb{R}^d$, and $\left|\mathcal{S}^{d-1}\right|$ the area of the unit sphere in the corresponding dimension. For any finite dimensional linear space $V \subseteq \mathbb{R}^d$, we define V^k to be the collection of matrices of form $[\boldsymbol{x}_1, \cdots, \boldsymbol{x}_k] \in \mathbb{R}^{d\times k}$, where $\boldsymbol{x}_j \in V$ is the j-th column vector. For any set $\mathcal{X}$, $\mathbb{1}_{\mathcal{X}}(x) = 1$ if $x \in \mathcal{X}$ else 0, is the indicator function of $\mathcal{X}$. For any vector $\boldsymbol{x} \in \mathbb{R}^d$, we denote $|\boldsymbol{x}|$ be the ℓ_2 norm of $\boldsymbol{x}$. For a matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_k] \in \mathbb{R}^{d\times k}$, $|\boldsymbol{W}| := \sum_{j=1}^k |\boldsymbol{w}_j|$ is the column-wise ℓ_2-norm sum.

2. **Problem setup.** In this section, we consider the multi-category classification problem in the d-dimensional space $\mathcal{X} = \mathbb{R}^d$. Let $\mathcal{Y} = [n] := \{1, 2, \cdots, n\}$ be the set of labels, and $\{\mathcal{D}_i\}_{i=1}^n$ be n probabilistic distributions over $\mathcal{X} \times \mathcal{Y}$. Throughout the theoretical analysis of this paper, we make the following assumptions on the data:

1. **(Separability)** There are n orthogonal subspaces $V_i \subseteq \mathcal{X}$ for $i \in [n]$ with $\bigoplus_{i=1}^n V_i = \mathcal{X}$, such that
$$\mathop{\mathbb{P}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}[\boldsymbol{x} \in V_i \text{ and } y = i] = 1.$$
2. **(Boundedness of data)** For $i \in [n]$, There exist positive constants m_i and M_i, such that
$$\mathop{\mathbb{P}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}[m_i \le |\boldsymbol{x}| \le M_i] = 1.$$
3. **(Boundedness of p.d.f.)** For $i \in [n]$, let p_i be the probability density function of distribution $\mathcal{D}_i$ restricted on V_i. For any $\boldsymbol{x} \in V_i$ with $m_i < |\boldsymbol{x}| < M_i$, it holds that
$$0 < p_{\min} \le p_i(\boldsymbol{x}) \le p_{\max} < \infty.$$

Later on, we denote $\mathcal{D}$ to be the evenly mixed distribution of $\mathcal{D}_i$'s. For notation simplicity, we let $m = \min_{i\in[n]} m_i$, $M = \max_{i\in[n]} M_i$ and $d_i = \dim V_i$.

We consider a two-layer neural network with k hidden neurons. Denote by $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_k] \in \mathbb{R}^{d\times k}$ the weight matrix in the hidden layer. For any input data $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^d$, we have

$$h_j = \langle \boldsymbol{w}_j, \boldsymbol{x}\rangle - b_j \text{ and } f_i = \sum_{j=1}^{n} v_{i,j}\sigma(h_j)$$

and the neural net outputs

$$f(\boldsymbol{W}; \boldsymbol{x}) = \boldsymbol{V}\sigma\left(\boldsymbol{W}^\top \boldsymbol{x}\right) = [f_1, \cdots, f_n]^\top, \tag{1}$$

where $\sigma := \max(\cdot, 0)$ is the ReLU function acting element-wise, and the bias $b_j \geq 0$ and $\boldsymbol{V} = (v_{i,j})$ are constants. Throughout this paper, we assume the following:

Assumption 1. *$\boldsymbol{V} = (v_{i,j}) \in \mathbb{R}^{k\times n}$ satisfies*

1. *For any $i \in [n]$, there exists some $j \in [n]$ such that $v_{i,j} > 0$.*
2. *If $v_{i,j} > 0$ then $v_{r,j} < 0$ for all $r \neq i$ and $r \in [n]$.*
3. *There exists some constant $v > 0$ such that $|v_{i,j}| = v$.*

One can show that as long as $k \geq n$, such $\boldsymbol{V}$ can be constructed easily.

The prediction is given by the maximum coordinate index of the network output

$$\hat{y}(\boldsymbol{W}; \boldsymbol{x}) = \operatorname*{argmax}_{i\in[n]} f_i,$$

ideally $\hat{y}(x) = i$ if $x \in V_i$. The classification accuracy in percentage is the frequency that this occurs (when network output label $\hat{y}$ matches the true label) on a validation data set. Given the data sample $\{\boldsymbol{x}, y\}$, the associated hinge loss function reads

$$l(\boldsymbol{W}; \{\boldsymbol{x}, y\}) := \sum_{i\neq y} \max\{0, 1 - f_y + f_i\}. \tag{2}$$

For network training, we consider the gradient descent algorithm with step size $\eta > 0$

$$\boldsymbol{W}^t = \boldsymbol{W}^{t-1} - \eta\nabla l(\boldsymbol{W}^{t-1}) \tag{3}$$

to solve following population loss minimization problem

$$\min_{\boldsymbol{W}\in\mathbb{R}^{d\times k}} l(\boldsymbol{W}) = \mathop{\mathbb{E}}_{\{\boldsymbol{x},y\}\sim\mathcal{D}} \left[l(\boldsymbol{W}; \{\boldsymbol{x}, y\})\right], \tag{4}$$

where the sample loss function $l(\boldsymbol{W}; \{\boldsymbol{x}, y\})$ is given by (2). Let l_i be the population loss function of data type i. More precisely,

$$l_i(\boldsymbol{W}) = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i} \left[l(\boldsymbol{W}; \{\boldsymbol{x}, y\})\right] = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i} \left[\sum_{r\neq i} \sigma(1 - f_i + f_r)\right].$$

Thus, we can rewrite the loss function as

$$l(\boldsymbol{W}) = \frac{1}{n}\sum_{i=1}^{n} l_i(\boldsymbol{W}).$$

Note that the population loss function

$$l_i(\boldsymbol{W}) = \sum_{r\neq i} \int_{\{f_i < f_r + 1\}} (1 - f_i + f_r)\, p_i(\boldsymbol{x})\; d\boldsymbol{x}$$

has no closed-form solution even if p is a constant function on its support. We cannot use closed-form formula to analyze the learning process, which makes our work different from many other works.

3. **Main results.** Although (4) is a non-convex optimization problem, we show that under mild conditions, the gradient descent algorithm (3) converges to a global minimum with zero classification error. Specifically, we consider two different networks with a positive bias $b_j > 0$ (Theorem 3.1) and without a bias (Theorem 3.2), respectively. For both cases, we have the fact that any critical point of problem (4) is a global minimum (Proposition 1). The key difference between these two cases is that the population loss function has Lipschitz continuous gradient (Lemma 4.2) when $b_j > 0$, whereas this desirable property does not hold otherwise. For the latter case $b_j = 0$, we present a totally different analysis based on a geometric condition proved to emerge during the training process (Proposition 3). Under this geometric condition, the objective value converges zero (Proposition 4).

Theorem 3.1. *Assume* $0 < \sum_{j=1}^{k} b_j < 1$ *and assumption 1 holds in (1), and* $\{\boldsymbol{W}^t\}$ *generated by the algorithm (3) are bounded uniformly in* t*. If there exists* $(\boldsymbol{x}, i) \sim \mathcal{D}_i$ *and some indices* $j \in [k]$*, such that* $v_{i,j} > 0$ *and* $\left|\left\langle \boldsymbol{w}_j^0, \boldsymbol{x}\right\rangle\right| > b_j$*, then there exists some* $\eta_0\left(v, k, b, p_{max}, n, M\right) > 0$ *such that for the learning rate* $\eta < \eta_0$*,* $\lim_{t\to\infty} l_i\left(\boldsymbol{W}^t\right) = 0$ *and*

$$\lim_{t\to\infty} \mathop{\mathbb{P}}_{\{\boldsymbol{x},i\}\sim\mathcal{D}_i} \left[\hat{y}\left(\boldsymbol{W}^t; \boldsymbol{x}\right) \neq i\right] = 0, \quad i \in [n],\ \forall\, n \geq 2.$$

Proof of Theorem 3.1. By Lemma 4.1, we only need to prove the convergence of the simplified network (6). From Lemma 4.2, we know $l_i(\boldsymbol{W})$ has Lipschitz gradient. We can assume for any $\boldsymbol{W}_1, \boldsymbol{W}_2 \in V_i^k$, we have

$$\left|\nabla l_i(\boldsymbol{W}_1) - \nabla l_i(\boldsymbol{W}_2)\right| \leq L\left|\boldsymbol{W}_1 - \boldsymbol{W}_2\right|.$$

As long as we take $\eta < \frac{L}{2}$ in algorithm (3), we know

$$l_i(\boldsymbol{W}^{t+1}) \leq l_i(\boldsymbol{W}^t) - \left(\eta - \frac{\eta^2 L}{2}\right)\left|\nabla l_i(\boldsymbol{W}^t)\right|^2 \leq l_i(\boldsymbol{W}^t). \tag{5}$$

Hence, $l_i(\boldsymbol{W}^t)$ is monotonically decreasing. Therefore, for any convergent subsequence $\{\boldsymbol{W}^{t_k}\}$ with the limit $\boldsymbol{W}_0$, there exists $l_0 \geq 0$ such that

$$\lim_{t\to\infty} l\left(\boldsymbol{W}^t\right) = \lim_{k\to\infty} l\left(\boldsymbol{W}^{t_k}\right) = l_0.$$

Now, we can take subsequence and limit on both side of equation (5), we get

$$l_0 \leq l_0 - \left(\eta - \frac{\eta^2 L}{2}\right)\left|\nabla l_i(\boldsymbol{W}_0)\right|^2.$$

Now, we see that $\nabla l_i(\boldsymbol{W}_0) = \boldsymbol{0}$. □

Theorem 3.2. *Assume* $b_j = 0$ *and assumption 1 holds in (1), and* $\{\boldsymbol{W}^t\}$ *generated by algorithm (3) is bounded by* R *uniformly in* t*. If there exist some* $(\boldsymbol{x}, i) \sim \mathcal{D}_i$ *and some indices* $j \in [k]$*, such that* $v_{i,j} > 0$ *and* $\left\langle \boldsymbol{w}_j^0, \boldsymbol{x}\right\rangle \neq 0$*, then* $\lim_{t\to\infty} l_i\left(\boldsymbol{W}^t\right) = 0$ *and*

$$\lim_{t\to\infty} \mathop{\mathbb{P}}_{\{\boldsymbol{x},i\}\sim\mathcal{D}_i} \left[\hat{y}\left(\boldsymbol{W}^t; \boldsymbol{x}\right) \neq i\right] = 0, \quad i \in [n],\ \forall\, n \geq 2.$$

Proof of Theorem 3.2. By Proposition 3, we know the iterates $\{\boldsymbol{W}^t\}$ stay in the first phase is bounded by $|T_1|$. Combining Proposition 4 which shows the summation of squared loss values in phase two is bounded, the summation of all loss values in the learning process is bounded

$$\sum_{t\in T_2} l_i\left(\boldsymbol{W}^t\right)^2 < \infty.$$

The desired result follows. □

Remark 1. The assumptions on the initialization $|\langle \boldsymbol{w}_j^0, \boldsymbol{x}\rangle| > b_j$ in both theorems are natural. This assumption guarantees that the neuron $\boldsymbol{w}_j$ is activated by some input data. Without this assumption, the algorithm suffers zero gradient and fails to update.

Remark 2. Theorem 3.2 does not explicitly require the learning rate η to be small. However, a larger learning rate will implicitly result in a larger bound in Propositions 3 and 4 which we used to prove Theorem 3.2.

4. Preliminaries.

4.1. Decomposition.

Lemma 4.1. *1. For any $i \in [n]$, if $\boldsymbol{W}_i^* \in \mathbb{R}^{d\times k}$ solves the optimization problem*

$$\min_{\boldsymbol{W}\in V_i^k} \; l_i(\boldsymbol{W}),$$

then $\boldsymbol{W}^* = \sum_{i=1}^n \boldsymbol{W}_i^*$ solves the original problem

$$\min_{\boldsymbol{W}\in\mathbb{R}^{d\times k}} \; l(\boldsymbol{W}).$$

2. If $\boldsymbol{W}' = \boldsymbol{W} - \eta\nabla_{\boldsymbol{W}} l_i(\boldsymbol{W})$, then for any $r \neq i$, we have

$$l(\boldsymbol{W}'; \{\boldsymbol{x}, r\}) = l(\boldsymbol{W}; \{\boldsymbol{x}, r\})$$

for almost all $(\boldsymbol{x}, r) \sim \mathcal{D}_r$.

Proof of Lemma 4.1. Note that $\mathcal{X} = \mathbb{R}^d = \bigoplus_{i=1}^n V_i$, we decompose $\boldsymbol{w}_j = \sum_{i=1}^n \boldsymbol{w}_{j,i}$ where $\boldsymbol{w}_{j,i} \in V_i$.

Since V_i's are orthogonal spaces, we have

$$\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle = \langle \boldsymbol{w}_{j,i}, \boldsymbol{x}\rangle$$

if $\boldsymbol{x} \in V_i$. Now, assume $\boldsymbol{x} \in V_i$, let

$$\boldsymbol{W}_i = [\boldsymbol{w}_{1,i}, \cdots, \boldsymbol{w}_{k,i}],$$

we have $f(\boldsymbol{W}; \boldsymbol{x}) = f(\boldsymbol{W}_i, \boldsymbol{x})$ and hence we get our first claim.

On the other hand, we have

$$\nabla_{\boldsymbol{w}_j} l(\boldsymbol{W}; \{\boldsymbol{x}, y\}) = -\sum_{i\neq y} (v_{y,j} - v_{i,j})\, \mathbb{1}_{\Omega_{y,i}}(\boldsymbol{x}) \mathbb{1}_{\Omega_{\boldsymbol{w}_j}}(\boldsymbol{x})\boldsymbol{x},$$

where

$$\Omega_{y,i} = \{\boldsymbol{x} : f_y < f_i + 1\} \text{ and } \Omega_{\boldsymbol{w}_j} = \{\boldsymbol{x} : \langle \boldsymbol{w}_j, \boldsymbol{x}\rangle > b_j\}.$$

Now, we see that $\nabla_{\boldsymbol{w}_j} l(\boldsymbol{W}; \{\boldsymbol{x}, y\}) \in V_i$ for almost all $(\boldsymbol{x}, i) \sim \mathcal{D}_i$ so that $\boldsymbol{W}_r' = \boldsymbol{W}_r$ for all $r \neq i$ and hence our desired result follows. □

The optimization problem can be decomposed to n independent problems of the same form i.e. the optimization of l_i's. Therefore, it suffices to consider only one subproblem. Let $\boldsymbol{W}_i = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_k] \in V_i^k$, where $\boldsymbol{w}_j \in V_i = \mathbb{R}^{d_i}$, the network output for the input data $\boldsymbol{x} \in V_i = \mathbb{R}^{d_i}$ is given by

$$\tilde{f}^i(\boldsymbol{W}_i; \boldsymbol{x}) = [f_1, \cdots, f_n]^\top = \boldsymbol{V}\sigma\left(\boldsymbol{W}_i^\top \boldsymbol{x}\right). \tag{6}$$

Networks (1) and (6) are different, since the parameters in (1) are $\boldsymbol{W} \in \mathbb{R}^{d\times k}$, whereas in (6), we have $\boldsymbol{W}_i \in V_i^k \cong \mathbb{R}^{d_i \times k}$. The corresponding input data are also in different intrinsic dimensions. From now on, we just focus on the loss function associated with data of Class i:

$$l_i(\boldsymbol{W}) = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\left[\sum_{r\neq i} \max\{0, 1 - f_i + f_r\}\right].$$

4.2. **Landscape.** The following Proposition 1 shows that while the loss function is non-convex, any critical point is in fact a global minimum, except for some degenerate cases.

Proposition 1. *Consider the neural network in (6). Assume $d > 1$, if $\boldsymbol{W}$ is a critical point of $l_i(\boldsymbol{W})$ and there exists some $\boldsymbol{x} \in V_i$ such that $f(\boldsymbol{W}; \boldsymbol{x}) \neq \boldsymbol{0}$ then we have $l_i(\boldsymbol{W}) = 0$.*

Proof of Proposition 1. For any $j \in [k]$, we have

$$\nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}) = -\sum_{r\neq i}^{n} (v_{i,j} - v_{r,j}) \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\left[\mathbb{1}_{\Omega_{i,r}}(\boldsymbol{x})\mathbb{1}_{\Omega_{\boldsymbol{w}_j}}(\boldsymbol{x})\boldsymbol{x}\right] = \boldsymbol{0}.$$

Recall the definition of $\Omega_{\boldsymbol{w}_j}$, we know that each summand is a zero vector. By assumption 1, we know that either $v_{i,j} = v_{r,j}$ or

$$\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\left[\mathbb{1}_{\Omega_{i,r}}(\boldsymbol{x})\mathbb{1}_{\Omega_{\boldsymbol{w}_j}}(\boldsymbol{x})\boldsymbol{x}\right] = \boldsymbol{0},$$

where the latter implies $\Omega_{i,r} \cap \Omega_{\boldsymbol{w}_j} = \emptyset$. Observe that

$$f_i - f_r = \sum_{j=1}^{k} (v_{i,j} - v_{r,j})\,\sigma(h_j),$$

we see that if there exists some $(\boldsymbol{x}, y) \sim \mathcal{D}_i$ such that $f_i - f_r < 1$ then there must exist some $\boldsymbol{x} \in \Omega_{i,r}$ but this implies $\boldsymbol{x} \notin \Omega_{\boldsymbol{w}_j}$ for all $j \in [k]$ such that $v_{i,j} \neq v_{r,j}$ which gives $f_i = f_r = 0$. This contradicts with our assumption, so we get $f_i - f_r \geq 1$ for all $\boldsymbol{x} \sim \mathcal{D}_i$ and this implies $l_i(\boldsymbol{W}) = 0$. □

The above result holds only when the global minimum of training loss function exists. The following proposition shows that the loss function has plenty of global minima.

Proposition 2. *Consider the network in (6). If the convex hull spanned by vertices $\{\boldsymbol{w}_j : v_{i,j} > 0\}$ contains a ball centered at the origin with radius $\max\limits_{j\in[k]} \frac{1+b_j}{m_i}$, and $\{\boldsymbol{w}_j : v_{i,j} < 0\}$ lies in a ball with radius $\min\limits_{j\in[k]} \frac{b_j}{M_i}$, then $l_i(\boldsymbol{W}) = 0$.*

The above proposition shows that if number of neurons is greater than the dimension of input data, then global minimum exists. Next, we study the smoothness of the loss function. The following proposition shows that as long as weights are bounded away from 0, then the loss function has Lipschitz gradient.

Lemma 4.2. *Consider the network in (1) with positive bias* $0 < \sum_{j=1}^{k} b_j < 1$. *The loss function* $l(W)$ *is Lipschitz differentiable, i.e, there exists some constant* $L > 0$ *depending on* k, $\boldsymbol{b}, p_{max}, , M$, $\boldsymbol{V}$, *such that*

$$|\nabla l(\boldsymbol{W}_1) - \nabla l(\boldsymbol{W}_2)| \leq L\,|\boldsymbol{W}_1 - \boldsymbol{W}_2|\,.$$

Proof of Lemma 4.2. Note that $l(\boldsymbol{W}) = \frac{1}{n}\sum_{i=1}^{n} l_i(\boldsymbol{W})$, it suffice to show that each l_i has Lipschitz gradient. Note that

$$l_i(\boldsymbol{W}) = \sum_{r\neq i} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i} \left[\sigma\left(1 - f_i + f_r\right)\right],$$

it suffice to show each summand has Lipschitz gradient. Now, we write the gradient of the summands as

$$\begin{aligned}
&\nabla_{\boldsymbol{w}_j} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i} \left[\sigma\left(1 - f_i + f_r\right)\right] \\
=& \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i} \left[\nabla_{\boldsymbol{w}_j}\sigma\left(1 - f_i + f_r\right)\right] \\
=& \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i} \left[\mathbb{1}_{\Omega_{i,r}}(\boldsymbol{x})\mathbb{1}_{\Omega_{\boldsymbol{w}_j}}(\boldsymbol{x})\left(v_{r,j} - v_{i,j}\right)\boldsymbol{x}\right].
\end{aligned}$$

On one hand, if $v_{r,j} = v_{i,j}$, then the formula above is obviously zero and Lipschitz gradient follows. On the other hand, we can without loss of generality assume $|v_{i,j} - v_{r,j}| = 1$. For notation simplicity, we fix i and r and denote

$$\varphi(\boldsymbol{W}) := \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i} \left[\mathbb{1}_{\Omega_{i,r}}(\boldsymbol{x})\mathbb{1}_{\Omega_{\boldsymbol{w}_j}}(\boldsymbol{x})\boldsymbol{x}\right] \text{ and } \phi(\boldsymbol{W};\boldsymbol{x}) = f_i(\boldsymbol{W};\boldsymbol{x}) - f_r(\boldsymbol{W};\boldsymbol{x}).$$

Now, our desired result becomes $\varphi(\boldsymbol{W})$ is Lipschitz in $\boldsymbol{W}$. Denote $\Omega_1 = \Omega_{i,r}(\boldsymbol{W}_1)$ and $\Omega_2 = \Omega_{i,r}(\boldsymbol{W}_2)$ where $\boldsymbol{W}_i = \left[\boldsymbol{w}_1^i, \cdots, \boldsymbol{w}_k^i\right]$, we only need to show there exists some constant L such that

$$|\varphi(\boldsymbol{W}_1) - \varphi(\boldsymbol{W}_2)| \leq L\,|\boldsymbol{W}_1 - \boldsymbol{W}_2|\,.$$

Note that

$$\begin{aligned}
&|\varphi(\boldsymbol{W}_1) - \varphi(\boldsymbol{W}_2)| \\
=& \left|\mathbb{E}\left[\mathbb{1}_{\Omega_1}(\boldsymbol{x})\mathbb{1}_{\Omega_{\boldsymbol{w}_j^1}}(\boldsymbol{x})\boldsymbol{x}\right] - \mathbb{E}\left[\mathbb{1}_{\Omega_2}(\boldsymbol{x})\mathbb{1}_{\Omega_{\boldsymbol{w}_j^2}}(\boldsymbol{x})\boldsymbol{x}\right]\right| \\
\leq& \underbrace{\mathbb{E}\left[\mathbb{1}_{\Omega_1\Delta\Omega_2}(\boldsymbol{x})\,|\boldsymbol{x}|\right]}_{①} + \underbrace{\mathbb{E}\left[\mathbb{1}_{\Omega_{\boldsymbol{w}_j^1}\Delta\Omega_{\boldsymbol{w}_j^2}}\,|\boldsymbol{x}|\right]}_{②},
\end{aligned}$$

we can deal with ① and ② respectively.

W.l.o.g, we assume $\epsilon = |\boldsymbol{W}_1 - \boldsymbol{W}_2| \leq \frac{1}{2}$. On one hand, if $\boldsymbol{x} \in \Omega_1\Delta\Omega_2$, then we claim

$$1 - \epsilon\,|\boldsymbol{x}| \leq \phi(\boldsymbol{W}_1;\boldsymbol{x}) \leq 1 + \epsilon\,|\boldsymbol{x}|$$

because

$$|\phi(\boldsymbol{W}_1;\boldsymbol{x})-\phi(\boldsymbol{W}_2;\boldsymbol{x})| = \left|\sum_{j=1}^{k}(v_{i,j}-v_{r,j})\left[\sigma(h_j^1)-\sigma(h_j^2)\right]\right|$$
$$\leq \left|\sum_{j=1}^{k}\left\langle \boldsymbol{w}_j^1-\boldsymbol{w}_j^2,\boldsymbol{x}\right\rangle\right| \leq \left|\boldsymbol{W}_1^\top\boldsymbol{x}-\boldsymbol{W}_2^\top\boldsymbol{x}\right| \leq \epsilon\,|\boldsymbol{x}|\,.$$

Furthermore, we claim for such $\boldsymbol{x}$'s the gradient of ϕ is bounded away from zero. More precisely, with $\boldsymbol{x}=r\omega$ where $r=|\boldsymbol{x}|$, we have

$$\frac{d}{dr}\phi(\boldsymbol{W};\boldsymbol{x}) = \left[\sum_{j=1}^{k}(v_{i,j}-v_{r,j})\mathbb{1}_{\Omega_{\boldsymbol{w}_j}}(\boldsymbol{x})\boldsymbol{w}_j\right]\omega$$
$$\geq \left(\phi(\boldsymbol{W};\boldsymbol{x})-\sum_{j=1}^{k}b_j\right)/r \geq \frac{1}{M}\left(\frac{1}{2}-\sum_{j=1}^{k}b_j\right) =: C>0.$$

Now, we have

$$\textcircled{1} \leq \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\left[\mathbb{1}_{1-\epsilon|\boldsymbol{x}|\leq\phi(\boldsymbol{W}_1,x)\leq 1+\epsilon|\boldsymbol{x}|}\right](\boldsymbol{x})\,|\boldsymbol{x}|$$
$$\leq \int_{1-\epsilon|\boldsymbol{x}|\leq\phi(\boldsymbol{W}_1,x)\leq 1+\epsilon|\boldsymbol{x}|}|\boldsymbol{x}|\,p_{max}\,d\,\boldsymbol{x} \leq 2\left(\left|\mathcal{S}^{d_i-1}\right|\frac{M^{d_i}p_{max}}{C}\right)\epsilon.$$

As for $\textcircled{2}$, w.l.o.g. we can assume $\left|\boldsymbol{w}_j^1\right| \geq \left|\boldsymbol{w}_j^2\right|$. Note that when $\left|\boldsymbol{w}_j^1\right| \leq \frac{b_j}{M}$ then $h_j \leq 0$ so that $\nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}) = \boldsymbol{0}$ and this $\textcircled{2} = 0$. Hence, we only need to take care of the case when $\left|\boldsymbol{w}_j^1\right| \geq \frac{b_j}{M}$. Note that $\left|\boldsymbol{w}_j^1-\boldsymbol{w}_j^2\right| \leq \epsilon$ we know

$$\sin\theta \leq \frac{\epsilon M}{b_j},$$

where θ denotes the acute angle between $\boldsymbol{w}_j^1$ and $\boldsymbol{w}_j^2$. We have the following estimate

$$\textcircled{2} \leq p_{max}\frac{\epsilon M}{b_j}\left|\mathcal{S}^{d_i-1}\right|.$$

Combine with $\textcircled{1}$, we get our desired result. □

Note that Lipschitz differentiability in Lemma 4.2 does not hold for the case $b_j = 0$, as the gradient might be volatile near the origin.

5. **Convergence analysis for non-bias case.** With the Lipschitz differentiability shown in Lemma 4.2 in the case $b_j > 0$, it is not hard to prove the convergence result in Theorem 3.1. In this section, we focus on the non-bias case ($b_j = 0$) where the Lipschitz differentiability fails and sketch the convergence analysis.

Lemma 5.1. *Consider the network (6), $|\boldsymbol{w}_j^t|$ is non-decreasing in t if $v_{i,j} > 0$. For any $r > 0$, choosing learning rate $\eta < \min\left\{\frac{r}{C_p M_i^2}, \frac{r}{2vnM_i}\right\}$, then $|\boldsymbol{w}_j^t|$ is non-increasing if $v_{i,j} < 0$ and $\left|\boldsymbol{w}_j^t\right| > r$, where C_p is a constant satisfying*

$$C_p = \max_{\boldsymbol{v}\in V_i, a\in\mathbb{R}}\int_{\{\langle\boldsymbol{v},\boldsymbol{x}\rangle=a\}} p_i(\boldsymbol{x})\,d\,\boldsymbol{x} = O\left(p_{max}M^{d_i}\right).$$

Proof of Lemma 5.1. We first define

$$C_p = \max_{\boldsymbol{v}\in V_1, a\in\mathbb{R}} \int_{\langle \boldsymbol{v},\boldsymbol{x}\rangle = a} p_1(\boldsymbol{x})\, dS \le M^{d-1} p_{\max}.$$

Recall the definition of $\Omega_{\boldsymbol{w}_j}$, for all $\boldsymbol{x}\in\Omega_{\boldsymbol{w}_j}$, we have $\langle \tilde{\boldsymbol{w}}_j, \boldsymbol{x}\rangle > 0$. For $v_{i,j} > 0$,

$$\left\langle \tilde{\boldsymbol{w}}_j, -\nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W})\right\rangle = 2v \sum_{r\neq i} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_i}\left[\mathbb{1}_{\Omega_{i,r}}(\boldsymbol{x})\mathbb{1}_{\Omega_{\boldsymbol{w}_j}}(\boldsymbol{x})\langle \tilde{\boldsymbol{w}}_j, \boldsymbol{x}\rangle\right] \ge 0.$$

Note we have from (3) that

$$\boldsymbol{w}_j^{t+1} = \boldsymbol{w}_j^t - \eta \nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}^t).$$

So,

$$\left|\boldsymbol{w}_j^{t+1}\right| = \left\langle \boldsymbol{w}_j^{t+1}, \tilde{\boldsymbol{w}}_j^{t+1}\right\rangle \ge \left\langle \boldsymbol{w}_j^{t+1}, \tilde{\boldsymbol{w}}_j^{t}\right\rangle \ge \left\langle \boldsymbol{w}_j^{t}, \tilde{\boldsymbol{w}}_j^{t}\right\rangle = \left|\boldsymbol{w}_j^{t}\right|.$$

Let $\Omega_{i,r}^j = \Omega_{i,r}\cap\Omega_{\boldsymbol{w}_j}$. For $v_{i,j} < 0$, we know

$$\left|\nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W})\right| = 2v\left|\sum_{r\neq i}\mathbb{E}\left[\mathbb{1}_{\Omega_{i,r}^j}(\boldsymbol{x})\boldsymbol{x}\right]\right| \le 2vM\sum_{r\neq i}\mathbb{P}\left[\Omega_{i,r}^j\right]^2,$$

where we omit the distribution $\boldsymbol{x}\sim\mathcal{D}_i$.

On the other hand, by definition of C_p, we know

$$\left|\left\langle \nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}), \tilde{\boldsymbol{w}}_j\right\rangle\right| = \left|2v\sum_{r\neq i}\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_i}\left[\mathbb{1}_{\Omega_{i,r}^j}(\boldsymbol{x})\langle \tilde{\boldsymbol{w}}_j, \boldsymbol{x}\rangle\right]\right| \ge \frac{v}{C_p}\sum_{r\neq i}\mathbb{P}\left[\Omega_{i,r}^j\right]^2.$$

When $0 < \eta < \frac{r}{2vnM}$, we have

$$\left\langle \boldsymbol{w}_j - \eta\nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}), \tilde{\boldsymbol{w}}_j\right\rangle = \left\langle \boldsymbol{w}_j, \tilde{\boldsymbol{w}}_j\right\rangle - \eta\left\langle \nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}), \tilde{\boldsymbol{w}}_j\right\rangle > r - 2v\eta M\sum_{r\neq i}\mathbb{P}\left[\Omega_{i,r}^j\right] > 0.$$

Now, we decompose $\nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}^t)$ into two parts,

$$\nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}^t) = \underbrace{\left\langle \tilde{\boldsymbol{w}}_j^t, \nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}^t)\right\rangle \tilde{\boldsymbol{w}}_j^t}_{\boldsymbol{n}} + \underbrace{\left(\nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}^t) - \left\langle \tilde{\boldsymbol{w}}_j^t, \nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}^t)\right\rangle \tilde{\boldsymbol{w}}_j^t\right)}_{\boldsymbol{\nu}}.$$

So that when $\eta < \frac{r}{M^2 C_p}$, we have

$$\begin{aligned}
&\left|\boldsymbol{w}_j^{t+1}\right|^2 \\
=&\left|\boldsymbol{w}_j^t - \eta\nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}^t)\right|^2 \\
=&\left|\boldsymbol{w}_j^t - \eta\boldsymbol{n}\right|^2 + \left|\eta\boldsymbol{\nu}\right|^2 \\
=&\left|\boldsymbol{w}_j^t\right|^2 - \eta\left(2\left|\boldsymbol{w}_j^t\right|\left|\boldsymbol{n}\right| - \eta\left(\left|\boldsymbol{n}\right|^2 + \left|\boldsymbol{\nu}\right|^2\right)\right) \\
\le&\left|\boldsymbol{w}_j^t\right|^2 - 2v\eta\left(\frac{\left|\boldsymbol{w}_j^t\right|}{C_p} - \eta M^2\right)\sum_{r\neq i}\mathbb{P}\left[\Omega_{i,r}^j\right]^2 \\
\le&\left|\boldsymbol{w}_j^t\right|^2,
\end{aligned}$$

as long as $|\boldsymbol{w}_j| \ge r$. Now, we get the desired result. □

The above theorem provides the dynamic information of the weights. When we input data with distribution $\mathcal{D}_i$, the weights $\{\boldsymbol{w}_j : v_{i,j} > 0\}$ become more useful for classification as the norm of those $\boldsymbol{w}_j$'s grows larger on every iteration. On the other hand, $\{\boldsymbol{w}_j : v_{i,j} < 0\}$ serve only as noise when $v_{i,j} < 0$. On one hand, when $\boldsymbol{w} \in \{\boldsymbol{w}_j : v_{i,j} < 0\}$ has large magnitude, the learning process guarantees the decreasing of $|\boldsymbol{w}|$. On the other hand, when $\boldsymbol{w} \in \{\boldsymbol{w}_j : v_{i,j} < 0\}$ has a small magnitude, it shall be trapped into a small region near the origin and contribute little to classification.

The learning process consists of two phases. In the beginning, since the weights are randomly distributed, there may well be some data that does not activate any neurons. The weights then automatically spread out. We call this process the first (slow learning) phase, during which the learning process is rather slow. The next Lemma lists four equivalent statements of a geometric condition. When the geometrical condition holds, we say that the learning process enters the second (fast learning) phase.

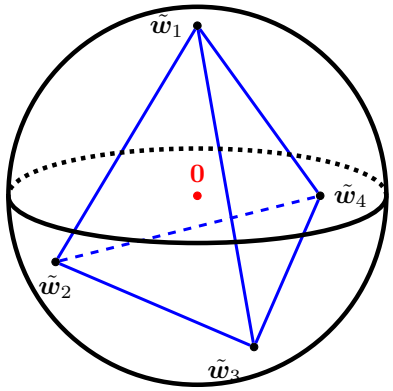

FIGURE 1. Geometric Condition in Lemma 5.2 ($d = 3$)

Lemma 5.2. *Let $\tilde{\boldsymbol{w}}_j = \frac{\boldsymbol{w}_j}{|\boldsymbol{w}_j|} \in \mathcal{S}^{d-1}$ be k points on the unit sphere, where $1 \le j \le k$. Let Λ_W be the convex hull of $\{\tilde{\boldsymbol{w}}_j\}$ and $\{H_i\}_{i=1}^{l}$ be the facets of convex hull. The following statements are equivalent:*

1. *For any unit vector $\boldsymbol{n} \in \mathcal{S}^{d-1}$, there exist some j_1 and j_2 such that $\langle \boldsymbol{n}, \tilde{\boldsymbol{w}}_{j_1} \rangle > 0$ and $\langle \boldsymbol{n}, \tilde{\boldsymbol{w}}_{j_2} \rangle < 0$.*
2. *There exists no closed hemisphere that contains all $\tilde{\boldsymbol{w}}_j$.*
3. *$\mathbf{0}$ lies in the interior of Λ_W.*
4. *For each $0 \le i \le l$, we have $\mathbf{0}$ and all $\tilde{\boldsymbol{w}}_j$ lie on the same side of H_i.*

Proof of Lemma 5.2. $(1 \Leftrightarrow 2)$ is trivial.

$(1 \Rightarrow 3)$ Proof by contradiction. Assume $\mathbf{0} \notin \Lambda_W^\circ$. Since Λ_W° is a open convex set, and $\{\mathbf{0}\}$ is a convex set, we know from geometric form of Hahn-Banach Theorem, that there exist a closed hyper-plane that separates $\{\mathbf{0}\}$ and Λ_W°. Hence, there exist a unit vector $\boldsymbol{n} \in \mathcal{S}^{d-1}$, such that $\langle \boldsymbol{n}, \tilde{\boldsymbol{w}}_j \rangle \ge \langle \boldsymbol{n}, \mathbf{0} \rangle = 0$ for all j, but this contradict with our assumptions in 1.

$(3 \Rightarrow 4)$ Assume that $H_i = [\langle \boldsymbol{v}_i, \boldsymbol{w} \rangle = \alpha_i]$, where $\alpha > 0$. Note that the convex hull Λ_W is a polytope with faces H_i. We know, Λ_W° all lies on one side of H_i. Since

$\mathbf{0} \in \Lambda_W^\circ$, we know for all $\boldsymbol{w} \in \Lambda_W^\circ$ we have $\langle \boldsymbol{v}_i, \boldsymbol{w} \rangle < \alpha_i$, and hence $\langle \boldsymbol{v}_i, \tilde{\boldsymbol{w}}_j \rangle \leq \alpha_i$ for all j.

$(4 \Rightarrow 3)$ is trivial.

$(3 \Rightarrow 1)$ $\mathbf{0} \in \Lambda_W^\circ$ implies there exist positive numbers λ_j, such that

$$\sum_{j=1}^{k} \lambda_j \tilde{\boldsymbol{w}}_j = \mathbf{0}.$$

Hence for any unit vector $\boldsymbol{n}$, we have

$$\sum_{j=1}^{k} \lambda_j \langle \boldsymbol{n}, \tilde{\boldsymbol{w}}_j \rangle = 0.$$

Since $\tilde{\boldsymbol{w}}_j$ are in general position, so $\langle \boldsymbol{n}, \tilde{\boldsymbol{w}}_j \rangle$ cannot all be 0. Hence, there must be both positive and negative terms. Now, we get the desired result. □

Remark 3. If $\boldsymbol{w}_j$ is initialized such that $\tilde{\boldsymbol{w}}_j$ is uniformly distributed on $\mathcal{S}^{d-1}$, then the probability that the geometric condition (GC) in Lemma 5.2 holds is

$$\mathrm{P_{gc}} := \mathrm{Prob}(\text{GC holds}) = 2^{1-k} \sum_{j=d}^{k-1} \binom{k-1}{j}.$$

In particular, at any fixed feature dimension d,

$$\lim_{k \to \infty} \mathrm{P_{gc}} = 1.$$

Proof of Remark 3. For any $J \in \{\pm 1\}^k$, we let $S_J(\{\tilde{\boldsymbol{w}}_j\}) = \{J_j \tilde{\boldsymbol{w}}_j : 1 \leq j \leq k\}$. From the choice of $\tilde{\boldsymbol{w}}_j$, we know all S_J have the same distribution, so that

$$\mathrm{Prob}_{\{\tilde{\boldsymbol{w}}_j\}}(\text{GC holds}) = \mathrm{Prob}_{S_J}(\text{GC holds}).$$

Hence, we can simplify the probability as follows

$$\mathrm{Prob}(\text{GC holds}) = \frac{1}{2^n} \mathbb{E}\left[\sum_J \mathbb{1}_{gc}\left(S_J\left(\{\tilde{\boldsymbol{w}}_j\}\right)\right) \right].$$

We claim that $\sum_J \mathbb{1}_{gc}\left(S_J\left(\{\tilde{\boldsymbol{w}}_j\}\right)\right)$ is a constant independent of choice of $\tilde{\boldsymbol{w}}_j$ as long as they are in general position.

Let $\tilde{\boldsymbol{w}}_j \in \mathcal{S}^{d-1}$ where $1 \leq j \leq k$. Each $\tilde{\boldsymbol{w}}_j$ corresponds to a subspace H_j with codimension 1 in $\mathbb{R}^d$, that is

$$H_j = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \langle \tilde{\boldsymbol{w}}_j, \boldsymbol{x} \rangle = 0 \right\}.$$

Note that any connected region of $(\cup_J H_J)^c$ corresponds to a choice of J such that the geometric condition fails. More precisely, for any given connected region D of $(\cup_J H_J)^c$, we know $\langle \tilde{\boldsymbol{w}}_j, \boldsymbol{x} \rangle$ is one sign for all $\boldsymbol{x} \in D$, so with

$$J_j = \mathrm{sign}\left(\langle \tilde{\boldsymbol{w}}_j, \boldsymbol{x} \rangle\right),$$

we know $\boldsymbol{x}$ correspond to $S_J(\{\tilde{\boldsymbol{w}}_j\})$ where the geometric condition fails. Hence, the number of connected regions of $(\cup_J H_J)^c$ is same as $2^n - \sum_J \mathbb{1}_{gc}\left(S_J\left(\{\tilde{\boldsymbol{w}}_j\}\right)\right)$. By [7], we have

$$\sum_J \mathbb{1}_{gc}\left(S_J\left(\{\tilde{\boldsymbol{w}}_j\}\right)\right) = 2^n - 2\sum_{j=0}^{d-1} \binom{k-1}{j} = 2\sum_{j=d}^{k} \binom{k-1}{j}.$$

The desired result follows. □

Per Remark 3, the more neurons the network has, higher possibility the geometric condition in Lemma 5.2 holds upon initialization. As a consequence, the learning process skips the first phase and goes straight to the fast learning phase. This explains why gradient descent for learning a over-parameterized network converges rapidly to a nearby critical point from random initialization.

The following proposition gives an upper bound on the maximum number of iterations for the learning process to enter the second phase.

Proposition 3. *Let $b_j = 0$ in (6), and assume that $|\boldsymbol{W}^t| \leq R$ for all t. Let T_1 be the set of t such that $\left\{\boldsymbol{w}_j^t : v_{i,j} > 0\right\}$ does not satisfy the geometric condition in Lemma 5.2, then $|T_1| \leq \frac{C_p R}{v\eta p_R^2}$, where p_R is a positive constant. Also, the following estimate holds for p_R:*

$$p_R = \Omega\left(\frac{p_{min}}{\sqrt{d_i}\left(M_i R\right)^{d_i}}\right)$$

where C_p is the constant in Lemma 5.1. More precisely,

$$|T_1| = O\left(\frac{C_p d_i R^{2d_i+1} M_i^{2d_i}}{v\eta p_{min}^2}\right) = O\left(\frac{C_p d R^{2d+1} M^{2d}}{v\eta p_{min}^2}\right).$$

Proof of Proposition 3. W.l.o.g, assume $i = 1$ and $v_{1,j} > 0$ if and only if $j \in [k_1]$. Let $t \in T_1$, by Lemma 5.2, we know, for any fixed $t \in T_1$, there exists a $\alpha \in [0, \frac{\pi}{2}]$ and a unit vector $\boldsymbol{v} \in \mathbb{R}^{d_1}$, such that $\langle \boldsymbol{v}, \tilde{\boldsymbol{w}}_j^t \rangle \geq \sin\alpha$ for all $j \in [k_1]$ and $\langle \boldsymbol{v}, \tilde{\boldsymbol{w}}_1^t \rangle = \sin\alpha$. W.l.o.g, we assume $\boldsymbol{v} = (1, 0, \cdots, 0)$ and $\langle \boldsymbol{v}, \tilde{\boldsymbol{w}}_1^t \rangle = \sin\alpha$. For any non-zero $\boldsymbol{x} \in V_1$, we write $\tilde{\boldsymbol{x}} = \frac{\boldsymbol{x}}{|\boldsymbol{x}|} \in \mathcal{S}^{d_1-1}$.

Note that $|\boldsymbol{W}|$ is bounded by R, we know there exist $\beta \in (0, \frac{\pi}{2} - \alpha)$ such that $\sum_{j=1}^k \left|\boldsymbol{w}_j^t\right| \leq R = \frac{1}{2vM_1 \sin\beta}$. Now, w.l.o.g, we can assume $\tilde{\boldsymbol{w}}_1^t = (\cos\alpha, \sin\alpha, 0, \cdots, 0)$. Take $\boldsymbol{n} = (-\cos\alpha, \sin\alpha, 0, \cdots, 0)$, we know $\langle \boldsymbol{n}, \tilde{\boldsymbol{w}}_1^t \rangle = 0$. For all $\boldsymbol{x} \in V_1$ such that $\langle \boldsymbol{n}, \tilde{\boldsymbol{x}} \rangle > \cos\beta$, we have

$$\cos\beta < \langle \boldsymbol{n}, \tilde{\boldsymbol{x}} \rangle = -\tilde{x}_1 \cos\alpha + \tilde{x}_2 \sin\alpha \leq -\tilde{x}_1 \cos\alpha + \sqrt{1 - \tilde{x}_1^2} \sin\alpha.$$

So, we have

$$\tilde{x}_1 < -\cos(\alpha + \beta).$$

Now, for all $\boldsymbol{x} \in V_1$ such that $\tilde{x}_1 < -\cos(\alpha + \beta)$, we have

$$f_1(\boldsymbol{W}^t; \boldsymbol{x}) - f_r(\boldsymbol{W}^t; \boldsymbol{x}) \leq 2vM_1 \sum_{j=1}^k \sigma\left(\left\langle \boldsymbol{w}_j^t, \tilde{\boldsymbol{x}} \right\rangle\right) \leq 2vM_1 \sum_{j=1}^k \left|\boldsymbol{w}_j^t\right| \sin\beta < 1.$$

So, with

$$D^1 := \left\{\boldsymbol{x} \in V_1 : \langle \boldsymbol{n}, \tilde{\boldsymbol{x}} \rangle > \cos\beta \text{ and } \langle \tilde{\boldsymbol{w}}_1, \tilde{\boldsymbol{x}} \rangle > 0\right\},$$

we have for any $r > 1$

$$D^1 \subset \Omega_{1,r}.$$

See Fig 2 for a intuition of D^1. Note that $\boldsymbol{n}$ and $\tilde{\boldsymbol{w}}_1$ perpendicular to each other, and the probability distribution p_1 is bounded from below, so there exists a constant p_R, such that

$$\mathbb{P}\left[\Omega_{1,r} \cap \Omega_{\boldsymbol{w}_j^t}\right] \geq \mathbb{P}\left[D^1\right] = p_R > 0,$$

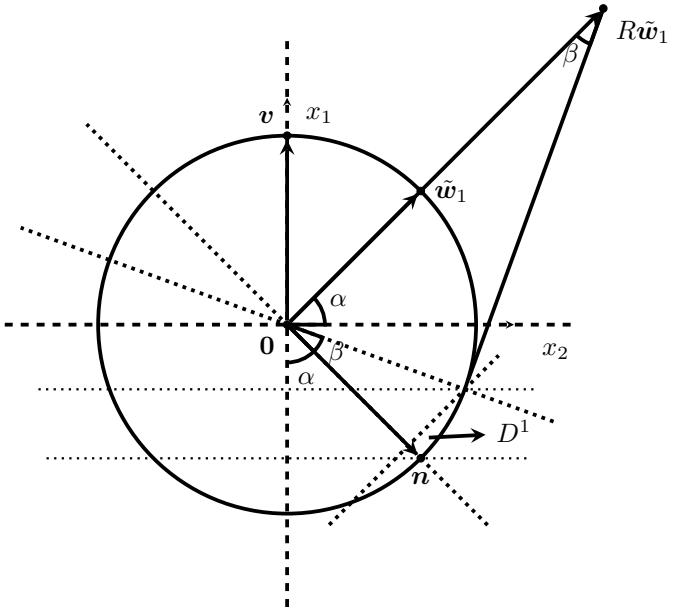

FIGURE 2. 2-dim section of $\mathbb{R}^d$ spanned by $\tilde{\boldsymbol{w}}_1$ and $\boldsymbol{n}$

and we have the following estimate for p_R

$$\begin{aligned} p_R &= \int_{D^1} \left\langle \tilde{\boldsymbol{w}}_1^t, \nabla_{\boldsymbol{w}_1} l(\boldsymbol{W}) \right\rangle p_1(\boldsymbol{x})\ d\boldsymbol{x} \\ &\geq \frac{p_{\min} \left|\mathcal{S}^{d_1-2}\right|}{\left|\mathcal{S}^{d_1-1}\right|} \int_{\cos\beta}^{1} \left(1-y^2\right)^{\frac{d_1-3}{2}} dy \\ &= \frac{p_{\min} \left|\mathcal{S}^{d_1-2}\right|}{\left|\mathcal{S}^{d_1-1}\right|} \int_{0}^{\beta} (\sin\theta)^{d_1-2}\, d\theta \\ &= \Omega\left(\frac{p_{\min} (\sin\beta)^{d_1-1}}{(d_1-1)} \frac{\left|\mathcal{S}^{d_1-2}\right|}{\left|\mathcal{S}^{d_1-1}\right|} \right) \\ &= \Omega\left(\frac{p_{\min}}{\sqrt{d_1}\,(M_1 R)^{d_1}} \right). \end{aligned}$$

Now, we know the gradient of $\boldsymbol{w}_1$ on $\tilde{\boldsymbol{w}}_1$ direction is bounded below, with same arguments in proof of Lemma 5.1, we have

$$\left|\left\langle \tilde{\boldsymbol{w}}_1^t, \nabla_{\boldsymbol{w}_1} l_1\left(\boldsymbol{W}^t\right)\right\rangle\right| \geq \frac{v\, p_R^2}{C_p}.$$

Note that

$$\sum_{j=1}^{k_1} \left|\boldsymbol{w}_j^{t+1}\right| - \left|\boldsymbol{w}_j^t\right| \geq \left|\boldsymbol{w}_1^{t+1}\right| - \left|\boldsymbol{w}_1^t\right| \geq \eta \left|\left\langle \tilde{\boldsymbol{w}}_1^t, \nabla_{\boldsymbol{w}_1} l\left(\boldsymbol{W}^t\right)\right\rangle\right| \geq \frac{v\,\eta\, p_R^2}{C_p}$$

and since $\left|\boldsymbol{w}_j^t\right|$ is non-decreasing for $j \in [k_1]$, we have

$$\sum_{t\in T_1}\sum_{j=1}^{k_1}\left|\boldsymbol{w}_j^{t+1}\right| - \left|\boldsymbol{w}_j^t\right| \leq R.$$

Combining the above two equations, it follows $|T_1| \leq \dfrac{C_p R}{v\,\eta\,p_R^2}$. □

At the beginning of the second phase, $\tilde{w}_j$ are already evenly distributed. That is, the convex hull of $\tilde{\boldsymbol{w}}_j$ contains the origin, and any input data must at least activate some of the neurons. As long as an input data contributes to the loss, it also contributes to the gradient. So learning process becomes faster during the second phase. The following proposition shows the loss decays faster than before.

Proposition 4. *Let $b_j = 0$ in (6). Let T_2 be the set of t such that $\{\boldsymbol{w}_j^t : v_{i,j} > 0\}$ satisfies the geometric condition in Lemma 5.2, and that $|\boldsymbol{W}^t|$ is upper-bounded by R at all t. Then:*

$$\sum_{t\in T_2} l_i\left(\boldsymbol{W}^t\right)^2 \leq 4\eta^{-1}vn^2C_pR^2M_i^2R.$$

Proof of Proposition 4. First, we estimate $l_i(\boldsymbol{W}^t)$ as follows

$$\begin{aligned} l_i(\boldsymbol{W}^t) &= \sum_{r\neq i}\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\left[\sigma\left(1-f_i+f_r\right)\right] \\ &\leq \sum_{r\neq i}\left(2vRM_i\right)\mathop{\mathbb{P}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\left[f_i < 1+f_r\right] \\ &= \left(2vRM_i\right)\sum_{r\neq i}\mathop{\mathbb{P}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\left[\Omega_{i,r}\right]. \end{aligned}$$

Now, we have

$$\sum_{r\neq i}\mathop{\mathbb{P}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\left[\Omega_{i,r}\right] \geq \frac{l_i(\boldsymbol{W}^t)}{2vRM_i}.$$

Second, we estimate the gradient. Let $\Omega_{i,r}^j = \Omega_{i,r}\cap\Omega_{\boldsymbol{w}_j}$ and assume $v_{i,j} > 0$, we have

$$\nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}^t) = \sum_{r\neq i} 2v \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_i}\left[\mathbb{1}_{\Omega_{i,r}^j}(\boldsymbol{x})\boldsymbol{x}\right].$$

By the same arguments in proof of Lemma 5.1, we know

$$\left\langle \nabla_{\boldsymbol{w}_j} l_i(\boldsymbol{W}^t), \tilde{\boldsymbol{w}}_j^t\right\rangle \geq 2v\frac{\mathbb{P}\left[\Omega_{i,r}^j\right]^2}{2C_p} = \frac{v\,\mathbb{P}\left[\Omega_{i,r}^j\right]^2}{C_p}.$$

Next, we utilize the geometric condition which implies

$$\sum_{v_{i,j}>0}\mathbb{P}\left[\Omega_{i,r}^j\right] \geq \mathbb{P}\left[\Omega_{i,r}\right]$$

and get

$$\sum_{v_{i,j}>0} \left|\boldsymbol{w}_j^{t+1}\right| - \left|\boldsymbol{w}_j^{t}\right| \geq \sum_{\substack{v_{i,j}>0 \\ r\neq i}} \frac{v\eta\,\mathbb{P}\left[\Omega_{i,r}^j\right]^2}{C_p} = \frac{v\eta}{C_p} \sum_{\substack{v_{i,j}>0 \\ r\neq i}} \mathbb{P}\left[\Omega_{i,r}^j\right]^2$$

$$\geq \frac{v\eta}{C_p}\frac{1}{k_i}\sum_{r\neq i}\mathbb{P}\left[\Omega_{i,r}\right]^2 \geq \frac{v\eta}{C_p}\frac{1}{k_i(n-1)}\left(\sum_{r\neq i}\mathbb{P}\left[\Omega_{i,r}\right]\right)^2$$

$$\geq \frac{v\eta}{n^2 C_p}\frac{l_i(\boldsymbol{W}^t)^2}{4v^2R^2M_i^2} = \frac{\eta}{4vn^2C_pR^2M_i^2} l_i(\boldsymbol{W}^t)^2.$$

where k_i is the number of j's such that $V_{i,j} > 0$.

Finaly, we combine the inequalities above and the assumption $|\boldsymbol{W}^t| < R$ and get

$$\sum_{t\in T_2} l_i(\boldsymbol{W}^t)^2 \leq \frac{4vn^2C_pR^2M_i^2}{\eta} \sum_{\substack{v_{i,j}>0 \\ t\in T_2}} \left|\boldsymbol{w}_j^{t+1}\right| - \left|\boldsymbol{w}_j^{t}\right| \leq 4\eta^{-1}vn^2C_pR^2M_i^2R.$$

□

From the above proposition, we see that in order to get $l_i\left(\boldsymbol{W}^t\right) < \epsilon$ for some $t \in T_2$, we only need

$$|T_2| \geq \frac{4vn^2C_pR^2M_i^2R}{\eta\epsilon}.$$

Compared with Proposition 3, we see that $|T_2|$ does not rely on the dimension d_i, whereas the upper bound of $|T_1|$ is exponential in d_i.

6. **Experiments.** In this section, we report the results of our experiments on both synthetic and MNIST data. The experiments on synthetic data aim to show that *convergence to global minimum continues to hold if data subspaces form acute angles, going beyond the theoretical orthogonality assumption* under which convergence is observed to be the fastest. Our theoretical results and the geometric conditions are supported by simulations. Experiments on MNIST dataset exhibit subspace structures in data flow and slow-to-fast training dynamics on LeNet-5. These phenomena from our model are worth further study in deep networks.

6.1. **Synthetic data.** Let $\{\boldsymbol{e}_j\}_{j\in[4]}$ be orthonormal basis of $\mathbb{R}^4$, θ be an acute angle and $\boldsymbol{v}_1 = \boldsymbol{e}_1$, $\boldsymbol{v}_2 = \sin\theta\,\boldsymbol{e}_2 + \cos\theta\,\boldsymbol{e}_3$, $\boldsymbol{v}_3 = \boldsymbol{e}_3$, $\boldsymbol{v}_4 = \boldsymbol{e}_4$. Now, we have two linearly independent subspaces of $\mathbb{R}^4$ namely $V_1 = \mathrm{Span}\left(\{\boldsymbol{v}_1, \boldsymbol{v}_2\}\right)$ and $V_2 = \mathrm{Span}\left(\{\boldsymbol{v}_3, \boldsymbol{v}_4\}\right)$. We can easily calculate that the angle between V_1 and V_2 is θ. Next, we define

$$\hat{\mathcal{X}}_1 = \left\{r\left(\cos\varphi\,\boldsymbol{v}_1 + \sin\varphi\,\boldsymbol{v}_2\right) : r \in S_r, \varphi \in S_\varphi\right\},$$

$$\hat{\mathcal{X}}_2 = \left\{r\left(\cos\varphi\,\boldsymbol{v}_3 + \sin\varphi\,\boldsymbol{v}_4\right) : r \in S_r, \varphi \in S_\varphi\right\},$$

where

$$S_r = \left\{\frac{20}{j} : j \in [20] - [9]\right\},\ S_\varphi = \left\{\frac{j\pi}{40} : j \in [80]\right\}.$$

Let $\mathcal{X}_1$ corresponds to label $y = 1$ and $\mathcal{X}_2$ corresponds to label $y = -1$. Since we are considering a binary classification problem, the neural network structure can be

simplified as (7):

$$\tilde{f}(\boldsymbol{W};\boldsymbol{x}) = \sum_{j=1}^{k} \sigma(h_j) - \sum_{j=k+1}^{2k} \sigma(h_j), \tag{7}$$

and the prediction is given by $\hat{y}(\boldsymbol{x}) = \operatorname{sign} f(\tilde{\boldsymbol{W}};\boldsymbol{x})$. Now, the population loss becomes

$$l_i(\boldsymbol{W}) := \frac{1}{|\hat{\mathcal{X}}_i|} \sum_{\boldsymbol{x}\in\hat{\mathcal{X}}_i} \max\left\{0, 1 + (-1)^i \tilde{f}(\boldsymbol{W};\boldsymbol{x})\right\}.$$

In our first simulation, we set $k = 4$ in (7) and run gradient descent (3) on $l_1 + l_2$ with learning rate $\eta = 0.1$. Fig. 3 shows the iterations it takes to converge to global minima given θ and a Gaussian noise added to $\mathcal{X}_i$'s. From this simulation, we see that the orthogonal data assumptions are only technically needed and our convergence result holds in more general settings.

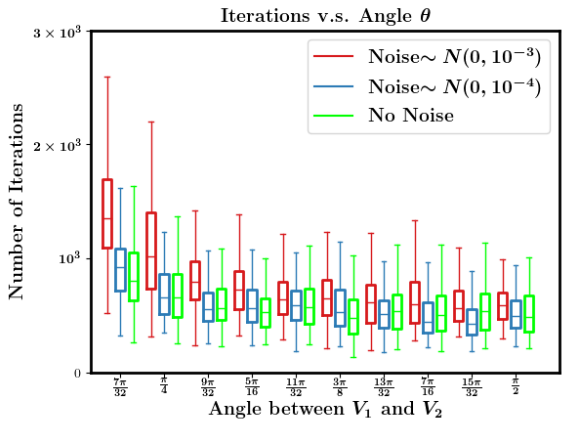

FIGURE 3. Number of iterations to convergence v.s. θ, the anlge between subspaces V_1 and V_2.

In our second and remaining simulations of this subsection, we take $\theta = \frac{\pi}{2}$ so that V_1 and V_2 are orthogonal. Lemma 4.1 suggests the learning process of l_1 and l_2 are independent, so we only simulate the training process of l_1 and assume $\boldsymbol{w}_j \in V_1 \cong \mathbb{R}^2$. We take entries of $\boldsymbol{W}^0$ to be i.i.d. standard normal i.e. $\boldsymbol{w}_j^0 \sim N(\boldsymbol{0}, I_2)$. We train the network (7) with gradient descent (3) in all our simulations, where learning rate $\eta = 0.1$. The left plot of Fig. 4 shows how many iterations algorithm (3) takes in searching for a global minima from the random initialization mentioned above. For each box, the red mark indicates the median, and the bottom and top of the box indicate the 25th and 75th percentiles, respectively. As we can see from the graph, as number of hidden neurons ($2k$) becomes larger, the algorithm (3) tends to need less iterations in searching for a global minima.

In the third simulation, we compare the convergence speed with and without the geometric condition being satisfied. We introduce two initialization method: random initialization and half space initialization i.e. with $\hat{w}_{j,i} \sim N(0, 1)$, random initialization takes $w_{j,i}^0 = \hat{w}_{j,i}$ whereas half space initialization takes $w_{j,1}^0 = |\hat{w}_{j,1}|$ and $w_{j,2}^0 = \hat{w}_{j,2}$. We run the algorithm for 100 times with different numbers of

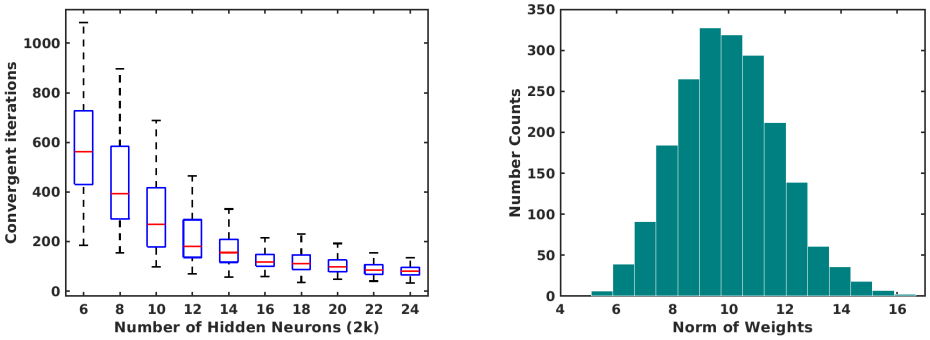

FIGURE 4. Left: convergent iterations vs. number of neurons ($d = 2$). Right: histogram of norm of weights: $\max_t |\boldsymbol{W}^t|$ ($d = 2$ and $k = 4$).

hidden neurons using initialization methods, and report the means and standard variances of the number of iterations in Table 1. We see from Remark 3 how the P_{gc} increases when the number of hidden neurons grows. However, the half space initialization never satisfies the geometric condition, as all the weights lie in the same half space. A widely believed explanation on why a neural network can fit all training labels is that the neural network is over-parameterized. Our work explained one of the reasons why over-parameterization helps convergence: it helps the weights to spread more 'evenly' and quickly after initialization. Table 1 shows that when we randomly initialize, the iterations for convergence in gradient descent (3) come down a lot as the number of hidden neurons increases; much less so in half space initialization.

TABLE 1. Iterations taken (mean $\pm$ std) to convergence with random and half space initializations.

# of Neurons ($2k$)	Random Init.	Half Space Init.
6	578.90±205.43	672.41±226.53
8	423.96±190.91	582.16±200.81
10	313.29±178.67	550.19±180.59
12	242.72±178.94	517.26±172.46
14	183.53±108.60	500.42±215.87
16	141.00±80.66	487.42±220.48
18	126.52±62.07	478.25±202.71
20	102.09±32.32	412.46±195.92
22	90.65±28.01	454.08±203.00
24	82.93±26.76	416.82±216.58

Our fourth simulation take specifically $2k = 8$. With 2000 runs we did a histogram of the maximum norm of $\boldsymbol{W}$ during the training process shown in the right plot of Fig. 4. In fact, our third simulation suggests our boundedness assumption on $\boldsymbol{W}$ in Theorem 3.1 and Theorem 3.2 are reasonable.

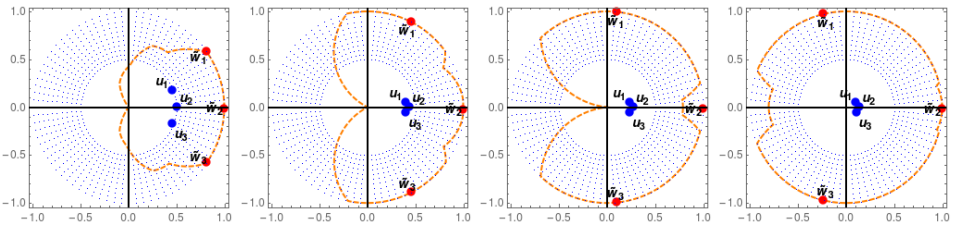

FIGURE 5. Dynamics of weights: $\tilde{\boldsymbol{w}}_j$ and $\boldsymbol{u}_j$

In our last simulation, we take $k = 3$ so that there are in total 6 hidden neurons. For notation simplicity, we denote $\boldsymbol{u}_j = \boldsymbol{w}_{j+3}$. For $j \in [3]$, we plot $\tilde{\boldsymbol{w}}_j$'s and $\boldsymbol{u}_j$'s in Fig. 5, where we plot $\tilde{\boldsymbol{w}}_j$'s instead of $\boldsymbol{w}_j$'s since some of $|\boldsymbol{w}_j|$'s are greater than one. Before algorithm (3) starts, the parameters in neural network (6) are initialized to be

$$\boldsymbol{w}_j^0 = \boldsymbol{u}_j^0 = \frac{3}{4}\left(\cos\frac{(2-j)\,\pi}{6}, \sin\frac{(2-j)\,\pi}{6}\right)$$

for $j \in \{1, 2, 3\}$. In Fig. 5, the tiny blue points are input data under Kelvin transformation: $\boldsymbol{x} \to \boldsymbol{x}^* = \frac{\boldsymbol{x}}{|\boldsymbol{x}|^2}$. Take $\boldsymbol{x} = \frac{1}{r}(\cos\theta, \sin\theta)$ so that under Kelvin transformation $\boldsymbol{x}^* = r(\cos\theta, \sin\theta)$. For convenience, we let $\tilde{\boldsymbol{x}} = r\boldsymbol{x} = (\cos\theta, \sin\theta)$. The orange dashed curve has expression in polar coordinates:

$$\rho(\theta) = \min\left\{1, \sigma\left(\tilde{f}\left(\boldsymbol{W}; \tilde{\boldsymbol{x}}\right)\right)\right\}.$$

Note that we are taking Hinge loss $l(\boldsymbol{W}; \{\boldsymbol{x}, 1\}) = 0$ if and only if $\tilde{f}(\boldsymbol{W}; \boldsymbol{x}) \geq 1$, i.e.

$$\tilde{f}\left(\boldsymbol{W}; \tilde{\boldsymbol{x}}\right) = r\tilde{f}\left(\boldsymbol{W}; \boldsymbol{x}\right) \geq r = |\boldsymbol{x}^*|.$$

Here, in our data set $\hat{\mathcal{X}}$, all data point have norm less than one under Kelvin transformation, so $l(\boldsymbol{W}; \{\boldsymbol{x}, 1\}) = 0$ if and only if $\rho(\theta) \geq |\boldsymbol{x}^*|$. This means, the blue points when surrounded by the orange dashed curve provide zero loss. In particular, when $\rho(\theta) = 1$, the population loss is 0.

6.2. **MNIST experiments.** The two-phase dynamics we proved in our model does appear in deep network training on real (non-synthetic) data sets. In experiments on MNIST, we used a simplified version of LeNet-5 [16] with 2 convolutional layers and two fully-connected (fc) layers; see [10] for the full version where F6 and output layers correspond to our fc layers. The simplified Lenet-5 is trained via stochastic gradient descent (SGD) at constant learning rate 0.01, batch size 1000 and without momentum or regularization. We show in Fig. 6 (left) the loss value vs. iterations during training. At the early stage, the loss decays slowly, then the fast phase sets in after 400 iterations. Fig. 6 (left) clearly supports our theory on the slow and fast dynamics of gradient descent.

Network visualization helps understand its geometric properties. In Fig. 6 (right), we plot 2D projections of feature vectors at input to fc layer extracted by a neural network [18] consisting of 6 convolutional layers followed by an fc layer. Projected features from different classes cluster around linearly independent subspaces. The plot suggests that in trained deep networks, the linearly independent subspace assumption approximately holds for the input to the fully connected layer before classification output. Similar subspace structure of high level feature vectors

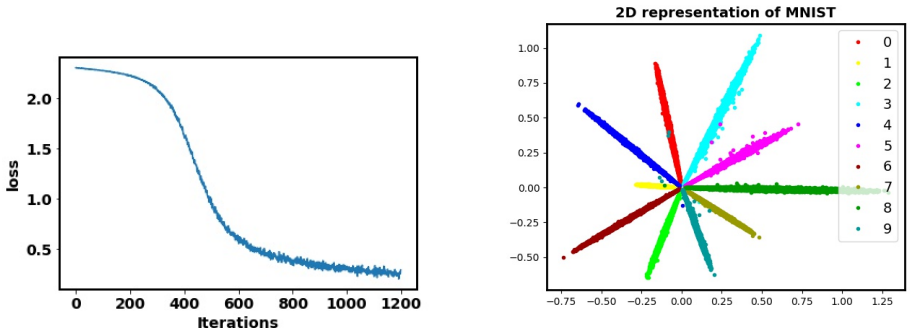

FIGURE 6. **Left**: Slow-to-Fast transition during LeNet [16] training on MNIST dataset. **Right**: 2D projections of MNIST features from a trained convolutional neural network [18]. The 10 classes are color coded, the feature points cluster near linearly independent subspaces.

on CIFAR-10 and enlargement of subspace angles to improve classification accuracy have been studied in [19].

In Fig. 7, we show four projections onto unit sphere $\mathcal{S}^2$ (inside randomly selected 3D subspaces) of the weight vectors of the first and second fc layers of LeNet [16]. Visual inspection on these and others (not shown) suggests that our geometric condition holds with high probability.

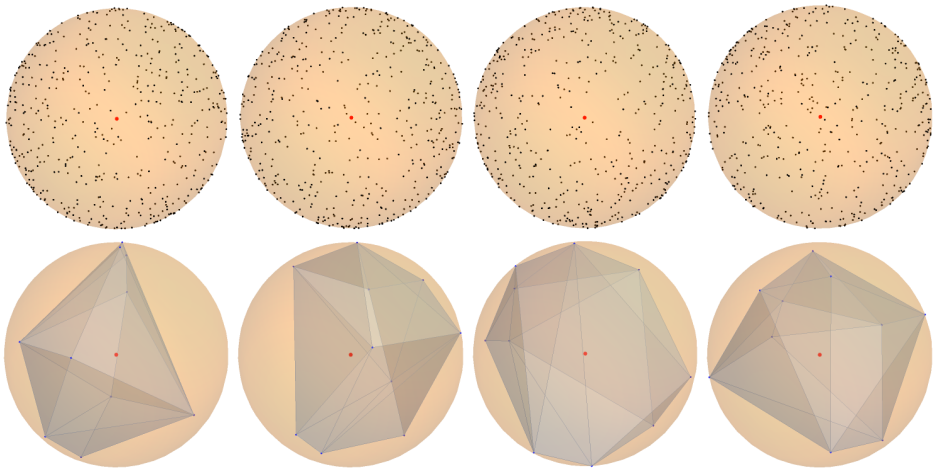

FIGURE 7. **Top row**: Projections onto $\mathcal{S}^2$ (inside randomly selected 3D subspaces) of weight vectors in the first fully connected layer of a trained LeNet. **Bottom row**: Projections onto $\mathcal{S}^2$ (inside randomly selected 3D subspaces) of weight vectors and their convex hull in the second fully connected layer of a trained LeNet.

7. **Conclusions and future work.** The slow and fast dynamics of neural network weights under gradient descent is critical for understanding the learning process. We performed the first theoretical study on training neural networks to classify linearly inseparable data sets away from the over-parametrized regime. We discovered a two time-scale phenomenon of network weights during gradient descent training: a slow phase where the weights spread out to satisfy a geometric condition, and a subsequent fast phase where the weights converge to a global minimum. Both the two scale dynamics and geometric conditions are supported by LeNet-5 training on MNIST data.

A future direction is to provide a concrete relation between the number of weights and the rate of convergence, and quantify the effect of over-parameterization on the rate of convergence. Another direction is to devise efficient method to verify the geometric condition computationally for weight vectors in training deep neural networks on general linearly non-separable data sets.

Acknowledgments. This work was partially supported by NSF grants IIS-1632935, DMS-1854434, DMS-1924548, and DMS-1924935.

REFERENCES

[1] Z. Allen-Zhu, Y. Li and Z. Song, A convergence theory for deep learning via over-parameterization, preprint, arXiv:1811.03962.
[2] A. Brutzkus and A. Globerson, Globally optimal gradient descent for a ConvNet with Gaussian inputs, preprint, arXiv:1702.07966.
[3] A. Brutzkus and A. Globerson, Over-parameterization improves generalization in the XOR detection problem, preprint.
[4] A. Brutzkus, A. Globerson, E. Malach and S. Shalev-Shwartz, SGD learns over-parameterized networks that provably generalize on linearly separable data, 6th International Conference on Learning Representations, Vancouver, BC, Canada, 2018, preprint, arXiv:1710.10174.
[5] R. T. des Combes, M. Pezeshki, S. Shabanian, A. Courville and Y. Bengio, Convergence properties of deep neural networks on separable data, 2019. Available from: https://openreview.net/forum?id=HJfQrs0qt7.
[6] S. S. Du, X. Zhai, B. Poczós and A. Singh, Gradient descent provably optimizes over-parameterized neural networks, preprint, arXiv:1810.02054.
[7] C. Ho and S. Zimmerman, On the number of regions in an m-dimensional space cut by n hyperplanes, *Austral. Math. Soc. Gaz.*, **33** (2006), 260–264.
[8] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780.
[9] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems*, 2012, 1097–1105.
[10] *LeNet-5 – A Classic CNN Architecture*. Available from: https://engmrk.com/lenet-5-a-classic-cnn-architecture/.
[11] Y. Li and Y. Liang, Learning overparameterized neural networks via stochastic gradient descent on structured data, preprint, arXiv:1808.01204.
[12] S. Liang, R. Sun, Y. Li and R. Srikant, Understanding the loss surface of neural networks for binary classification, preprint, arXiv:1803.00909.
[13] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun and N. Srebro, Towards understanding the role of over-parametrization in generalization of neural networks, preprint, arXiv:1805.12076.
[14] Q. Nguyen, M. C. Mukkamala and M. Hein, On the loss landscape of a class of deep neural networks with no bad local valleys, preprint, arXiv:1809.10749.
[15] S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Machine Intell.*, **39** (2017), 1137–1149.
[16] A. Rosebrock, LeNet - Convolutional neural network in Python, 2016. Available from: https://www.pyimagesearch.com/2016/08/01/lenet-convolutional-neural-network-in-python/.

[17] D. Silver, A. Huang, C. J. Maddison, A. Guez and L. Sifre, et al., Mastering the game of Go with deep neural networks and tree search, *Nature*, **529** (2016), 484–489.
[18] H. Wang, Y. Wang, Z. Zhou, X. Ji and D. Gong, et al., CosFace: Large margin cosine loss for deep face recognition, preprint, arXiv:1801.09414.
[19] P. Yin, J. Xin and Y. Qi, Linear feature transform and enhancement of classification on deep neural network, *J. Sci. Comput.*, **76** (2018), 1396–1406.

Received November 2019; 1st revision July 2020; final revision October 2020.

E-mail address: zlong6@uci.edu
E-mail address: pyin@albany.edu
E-mail address: jxin@math.uci.edu