Support Vectors and Gradient Dynamics for Implicit Bias in ReLU Networks

Sangmin Lee * 1  Byeongsu Sim * 1  Jong Chul Ye 1 2

Abstract

Understanding implicit bias of gradient descent has been an important goal in machine learning research. Unfortunately, even for a single-neuron ReLU network, it recently proved impossible to characterize the implicit regularization with the square loss by an explicit function of the norm of model parameters. In order to close the gap between the existing theory and the intriguing empirical behavior of ReLU networks, here we examine the gradient flow dynamics in the parameter space when training single-neuron ReLU networks. Specifically, we discover implicit bias in terms of support vectors in ReLU networks, which play a key role in why and how ReLU networks generalize well. Moreover, we analyze gradient flows with respect to the magnitude of the norm of initialization, and show the impact of the norm in gradient dynamics. Lastly, under some conditions, we prove that the norm of the learned weight strictly increases on the gradient flow.

1. Introduction

Recently, many researchers have investigated the intriguing generalization capability of ReLU networks even without explicit regularization (Alom et al., 2019; Goodfellow et al., 2016; Calin, 2020; Lee et al., 2019; Allen-Zhu et al., 2018). In particular, the number of trainable parameters in deep neural networks is often greater than the training data set, this situation being notorious for overfitting from the point of view of classical statistical learning theory. However, empirical results have shown that a deep neural network generalizes well in the test phase, resulting in high performance for the unseen data (Jiang et al., 2019).

This apparent contradiction has raised questions about the mathematical foundations of machine learning and their relevance to practitioners. A number of theoretical papers have been published to understand the intriguing generalization phenomenon in deep learning models (Neyshabur et al., 2015; Bartlett et al., 2017; Nagarajan & Kolter, 2019; Wei & Ma, 2019; Arora et al., 2018; Golowich et al., 2018; Neyshabur et al., 2018). In particular, the recent discovery of “double descent” (Belkin et al., 2019; 2020) extends the classical U-shaped bias-variance trade-off curve by showing that increasing the model capacity beyond the interpolation point leads to improved performance in the test phase. Particularly, the implicit bias by optimization algorithms offers simpler solutions that improve generalization in the over-parameterized regime (Gunasekar et al., 2018).

Many researchers have studied implicit bias and gradient flow. For linear networks, Gunasekar et al. (2017) showed that gradient flows with infinitesimally small norm converge to the minimum nuclear norm. Azulay et al. (2021) studied the initialization scale of gradient flows and obtained the closed-form implicit regularizer for several types of networks. Cornacchia et al. (2021) showed that noise labels guide the network to a sparse solution and reduce test error. The focus of these works is to find a regularization function of the model parameters, so that if we apply gradient descent on the average loss, then it converges in some sense to a global optimum that minimizes the regularization term.

Unfortunately, when we consider problems beyond simple linear classification and regression, the situation gets more complicated. For example, Vardi & Shamir (2021) showed that even for a simple single-neuron ReLU network, the implicit regularization cannot be expressed by any explicit function of the norm of model parameters.

In order to address the discrepancy between the theory and the empirical generalization power of ReLU networks, here we investigate the gradient flow dynamics when training single-neuron ReLU networks. While most of the theoretical analysis of ReLU networks focus on the input space partition (Hanin & Rolnick, 2019b;a; Park et al., 2021), we are particularly interested in the analysis in the parameter space since it has provided additional insight. Specifically, Xu et al. (2021) studied the partitioned parameter space by ReLU networks in terms of polytopes and suggested a traversing algorithm to visit all polytopes sequentially. Similarly, Lacotte & Pilanci (2020) consider the partitioned...
parameter space in two-layer ReLU networks. They provide an exact characterization of the set of all global optima of the non-convex loss landscape, and find explicit paths for non-increasing loss under $L^2$ regularization term.

Unlike the aforementioned works that mostly focus on the expressiveness and optimization landscape in terms of parameter space, the main aim of this paper is extending these ideas to understand the implicit bias and the dynamics of gradient flows of single-neuron ReLU networks. As such, the main findings and contributions of this work can be summarized as follows:

- We discover the implicit bias in terms of support vectors for single-neuron ReLU networks that play a key role in why and how ReLU networks can generalize well. We further prove that the global minimum of single-neuron ReLU networks has smaller losses than linear ones.
- We show that there is three realm of initialization: (i) small norm region where the gradient flow does not converge to bad local minima, (ii) large norm region where it converges to bad local minima, and (iii) critical region where the result is sensitive to initialization direction. This explains why a ReLU network is trained well, although there are many spurious minima.
- We provide simple proofs for norm-increasing property of single-neuron linear networks, and extend it to ReLU networks. More precisely, for a gradient flow of ReLU networks initialized with infinitesimally small norm, under some conditions the norm of the gradient flow is shown to strictly increase until it converges.

Most of the proofs can be found in Appendix. In addition, main theoretical findings of linear single-neuron networks, which are the basis of the main analysis for ReLU networks, are included in Appendix C.

2. Preliminaries

Notation. Throughout this article, boldface uppercase letters, boldface lowercase letters and normal lowercase letters denote matrices, vectors and scalars, respectively. $A^T$ and $v^T$ denote the transpose of a matrix $A$ and a vector $v$. We use $||\cdot||$ to Euclidean norm of a vector. $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ denote the largest and the smallest positive eigenvalues of a matrix $A$, respectively. ReLU activation function is denoted by $[x]_+ : \text{ReLU}(x) = \max\{0, x\}$. For two vectors $v_1, v_2 \in \mathbb{R}^d$, inequality $v_1 \geq v_2$ means $v_{1k} \geq v_{2k}$ for all $k = 1, 2, \ldots, d$. We denote the indicator function by

$$1_{\{c\}} : = \begin{cases} 1, & \text{if } c \text{ is True,} \\ 0, & \text{if } c \text{ is False.} \end{cases}$$

Training data set. The training data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is composed of input vectors $\mathbf{x}_i \in \mathbb{R}^d$ and output labels $y_i \in \mathbb{R}$, respectively. Throughout the paper, some of the following assumptions are often used:

- **A1** $\mathbf{x}_i \geq 0$, $\|\mathbf{x}_i\| \neq 0$, $\forall i$.
- **A2** $y_i > 0$, $\forall i$.
- **A3** $\text{rank}(\mathbf{x}_1 \cdots \cdot \mathbf{x}_n) = d$.

Specifically, using label-backpropagation described in Appendix B, we can consider each intermediate layer of deep ReLU networks as a single-neuron ReLU network. Since the input of each intermediate layer in deep ReLU networks is the output of the previous ReLU layer, we often use **A1**. Similarly, Lemma 3.1 shows **A2** is appropriate. Finally, reduction principle for ReLU networks (Corollary D.2) shows we can assume **A3** without loss of generality.

Single-neuron ReLU networks. The single-neuron ReLU networks training using the square loss is given by

$$\min_{w} L(w), \quad L(w) := \frac{1}{2} \sum_{i=1}^{n} (\mathbf{w}^T \mathbf{x}_i)_+ - y_i)^2$$

(1)

where $w \in \mathbb{R}^d$ is the model parameter that represents the neuronal weight. Here, a network with a bias term $b$ can be reduced to a network without bias by augmenting one dimension to input $x$ with a fixed scalar 1, i.e.,

$$\mathbf{w}^T \mathbf{x} + b = [\mathbf{w}^T \ 1]^{\top} \mathbf{x} = \tilde{w}^T \tilde{x}.$$

Thus, we only consider networks without bias. Subsequently, the gradient of $L$ is given by

$$\nabla L(w) = \sum_{i=1}^{n} 1_{\{\mathbf{w}^T \mathbf{x}_i > 0\}} (\mathbf{w}^T \mathbf{x}_i - y_i) \mathbf{x}_i$$

$$= \mathbf{H}(w) w - q(w)$$

(2)

where

$$\mathbf{H}(w) := \sum_{i=1}^{n} 1_{\{\mathbf{w}^T \mathbf{x}_i > 0\}} \mathbf{x}_i \mathbf{x}_i^T,$$

$$q(w) := \sum_{i=1}^{n} 1_{\{\mathbf{w}^T \mathbf{x}_i > 0\}} y_i \mathbf{x}_i.$$ (3)

Then, the goal of this paper is to investigate the implicit bias of the gradient flow given by

$$\dot{w} = -\nabla L(w) = -\mathbf{H}(w) w + q(w).$$

(4)

Parameter space partitioning. Let $\mathcal{W} \subset \mathbb{R}^d$ denote the parameter space, i.e. $w \in \mathcal{W}$ for all network weights $w$. Then, a partition $P \subset \mathcal{W}$ is referred to a subset of the parameter space that has the same activation pattern:

$$[1_{\{\mathbf{w}^T \mathbf{x}_1 > 0\}} \cdots 1_{\{\mathbf{w}^T \mathbf{x}_n > 0\}}] \in \{0, 1\}^n.$$
Figure 1. Description of terminology in single-neuron ReLU networks. In (a), in the parameter space $W$, a parameter $w$ is denoted by the yellow point. An input data $x$ is denoted by a thick green vector. The solid and dashed green lines are the activation boundary of $x$ and the solution hyperplane of $x$. The activated half space is filled by light green color. In (b), stationary points of a single-neuron ReLU network with $n = d = 2$ is denoted by red dots, which consist of a filled trapezoid, two dashed lines, and a point. Note that they are not connected, and each manifold has different dimension (Proposition 3.3). In (c), six data $\{x_i\}_{i=1}^6$ partition $W = \mathbb{R}^2$. Note that $x_1, x_2, x_5, x_6$ are activated in $P$, and $x_3, x_4$ are deactivated in $P$. The virtual minimum of $P$ is denoted by yellow star, which is not contained in $P$.

In the parameter space $W$, and for a given input data $x$, the subset $\{w \in W \mid w^T x > 0\}$ is called the activated half space with respect to $x$. Similarly, the subset $\{w \in W \mid w^T x < 0\}$ is called the deactivated half space, and the solution hyperplane of $x$ is the hyperplane defined by $\{w \in W \mid w^T x = 0\}$ (Figure 1(b)). We say data $x$ is activated in $P$ and denote by $x \sim P$ if $w^T x > 0$ for any $w \in P$. Similarly, a data $x$ is called deactivated in $P$ and denote by $x \not\sim P$ if $w^T x \leq 0$ for any $w \in P$.

The following lemma describes the relation between adjoined partitions (See Figure 9).

** Lemma 2.1.** Let $P_+$ and $P_-$ be two adjoined partitions, where the common boundary is determined by a data $x^*$. Let $P_+$ be a partition where $x^*$ is activated. Then,

$$\{x \in X \mid x \sim P_+\} = \{x \in X \mid x \sim P_-\} \cup \{x^*\}.$$ 

**Proof.** Since $P_+$ and $P_-$ are adjoined, their activation patterns are different in exactly one data, which is $x^*$. Note that $x^*$ is activated in $P_+$ and deactivated in $P_-$. Therefore, $\{x \in X \mid x \sim P_+\} = \{x \in X \mid x \sim P_-\} \cup \{x^*\}$.  

3. Loss Landscape and Support Vectors

By extending the analysis for the loss landscape of single-neuron linear networks in Appendix C, here we are interested in investigating the loss landscape of single-neuron ReLU networks. We first need the following lemmas.

** Lemma 3.1.** For single-neuron ReLU networks under $A1$, the parameter space has the following properties.

1. Every partition is convex and unbounded. In particular, if $w \in P$ for a partition $P$, then $\alpha w \in P$ for any scalar $\alpha > 0$ (i.e., $P$ is conic). Moreover, $H(w)$ and $q(w)$ in (3) are invariant under multiplication by a positive constant.

2. If there is a nonpositive label $y_i \leq 0$, a function $f_i(w) = \frac{1}{2}([w^T x_i]_+ - y_i)^2$ is convex with respect to $w$. In general, if there are $m$ positive labels among $n$ data, the number of partitions that contain a local minimum is at most $2 \sum_{i=0}^{d-1} \binom{m-1}{i} + 1$.

**Lemma 3.1** shows that negative labels do not affect to number of partitions which has a local minimum. This is the reason why we consider $A2$. Under this condition, the following lemma shows that every local minimum is strictly contained in some partition.

**Lemma 3.2 (Not on boundary lemma).** Consider a single-neuron ReLU network under $A2$. Then there is no minimizer on any activation boundary.

This implies that every local minimum lies inside of a partition. To investigate the specific condition for a partition to have a local minimum, we define the associated loss function on a partition $P$:

$$L_P(w) := \frac{1}{2} \sum_{x_i \sim P} ([w^T x_i]_+ - y_i)^2.$$ 

Then, its global minimizer

$$w_P^* = \arg \min_{w \in W} L_P(w)$$

is called the virtual minima of $P$. Note that $w_P^*$ may not be contained in $P$ (see Figure 1(c)). The following proposition states the precise condition when $P$ contains its virtual minima.

**Proposition 3.3.** Suppose $H(w)$ in (3) has rank $r_P$ on a partition $P$. If $r_P = d$, the unique virtual minimum of $P$ is
given by

$$ w^*_P = \left( \sum_{x_i \sim P} x_i x_i^T \right)^{-1} \sum_{x_i \sim P} y_i x_i $$

(5)

and it is contained in $P$ if and only if $w^*_P$ satisfies

$$ x_i^T w^*_P > 0 \quad \text{for all} \quad x_i \sim P, $$

$$ x_i^T w^*_P \leq 0 \quad \text{for all} \quad x_i \not\sim P. $$

(6)

If $r_P < d$, the virtual minima of $P$ exist in $P$ and they form a $(d - r_P)$-dimensional connected linear manifold in $P$.

Now we propose an interesting concept. For a partition $P$ such that $w^*_P \in P$, we call $x \sim P$ as the support vectors of $w^*_P$. Proposition 3.3 says that the data needed to compute $w^*_P$ is only its support vectors. It is worth noting that this terminology is closely related with the support vector machine (Cortes & Vapnik, 1995; Drucker et al., 1997; Vapnik et al., 1997) and support vectors in linear regression (Kavitha et al., 2016; Joki et al., 2020), in the sense that support vectors are the only required data to obtain the solution.

The following proposition provides a necessary condition for existence of local minima in two adjoined partitions in terms of the magnitude of labels.

**Proposition 3.4.** Consider a single-neuron ReLU network. Let $P_+$ and $P_-$ be two adjoined partitions, where the common boundary is determined by data $(x_0, y_0)$. Let $H := \sum_{x_i \sim P_-} x_i x_i^T$ and $q := \sum_{x_i \sim P_-} y_i x_i$. Suppose $H$ is full rank and $P_-$ contains a local minimum $w^*_-$.

Then there is no local minimum in $P_+$ if

$$ y_0 < -\frac{x_0^T H^{-1} q}{x_0^T H^{-1} x_0}. $$

(7)

This implies that labels with small magnitude do not contribute a lot to induce a new local minimum. Furthermore, the following result suggests an important advantage of ReLU networks.

**Theorem 3.5.** Under A2, the global minimum of a single-neuron ReLU network is smaller than the linear one.

This states that in order to have a smaller loss, some data could be deactivated during the training of ReLU networks. Specifically, a gradient flow gives up to fit some data and rather focuses on the best-fit of the remained data to learn larger common tendency of data. This may suggest why ReLU networks generalizes better than linear networks.

**Example 3.6** (Deactivation in ReLU networks.). Consider the three data ($n = 3$) in $\mathbb{R}^2$ ($d = 2$) described in Figure 2. In the parameter space $\mathcal{W} = \mathbb{R}^2$, there are three activation boundaries (blue, red and green solid lines). Corresponding three solution hyperplanes are drawn by dashed lines with same color. The gradient flow is denoted by the purple curve. Let $P_0$ be the partition where all three data are activated. The virtual minimum of $P_0$ is denoted by the orange star. After the gradient flow (the purple curve) deactivates the green data, the virtual minimum is changed to the yellow star, where the gradient flow finally converges.

One may wonder whether the large number of deactivation may be preferrable. However, the following theorem says that there is a trade-off, and if several local minima with different number of support vectors are feasible, the one with more support vectors is better.

**Theorem 3.7.** Consider a single-neuron ReLU network under A2. Let $w^*_1$ be a local minimum which has set of support vectors $S_1$, and $w^*_2$ be another local minimum which has set of support vectors $S_2$ such that $S_2 \subset S_1$. Then, $L(w^*_2) \geq L(w^*_1)$.

Later we will show in Theorem 4.3 that the gradient flow dynamics tends to avoid bad local minima and converges to the one with a large number of support vectors, which is another important implicit bias of the gradient flow.

### 4. Gradient Flow Dynamics

#### 4.1. Weight initialization

Figure 2 illustrates an example of a gradient flow initialized with small norm, goes through the partition with all the data being active, after which some of the data become deactivated to reach a local minimizer. It turns out that the initialization point $w_0$ plays the key roles in this dynamics.
as presented in the following lemma.

**Proposition 4.1.** For a single-neuron ReLU network, consider a parameter $w$ in a partition $P$ and define

$$
\alpha_j^* := \frac{x_j^T q}{x_j^T H w}
$$

for a data $x_j$, where $H$ and $q$ are given by (3). Then, $-\nabla L(\alpha w)^T x_j \leq 0$ if and only if $\alpha \geq \alpha_j^*$. In other words, a gradient flow with sufficiently large (resp., small) norm moves towards deactivating (resp., activating) all data.

![Figure 3](image)

Figure 3. Given a vector $\alpha w$ in a partition $P$ with $\alpha > 0$, $-\nabla L(\alpha w)$ is described by red arrows for various values of $\alpha$. By Proposition 4.1, red arrows tend to activate other data for $\alpha < \alpha_j^*$. Similarly, for $\alpha > \alpha_j^*$, red arrows tend to deactivate other data. Some gradient flows described by the purple curves converge to $w_j^*$, denoted by the yellow star. See Figure 4 to observe the shape of the loss level curves.

Another important issue in single-neuron ReLU networks is the assurance of activation. Example 3.6 also shows that an activated data is a crucial data. Interestingly, activation of data is also related with the initialization point $w_0$. In the following theorem, we suggest a condition of initialization point $w_0$ such that one specific data $x_j$ is kept activated on the gradient flow $w(t)$.

**Theorem 4.2 (No Deactivation).** Consider a gradient flow (4) in a single-neuron ReLU network under A1 and A2. Suppose there exists $w_{GM}^*$ such that $L(w_{GM}^*) = 0$. Then, $x_j$ is always activated on the gradient flow initialized at $w_0$ if $x_j^T w_0 > 0$ and

$$
\frac{y_j}{\|x_j\|} > \frac{\|w_0 - w_{GM}^*\|}{\|w_{GM}^*\|}.
$$

In particular, if (8) holds for all $j = 1, \cdots, n$, then the gradient flow initialized at $w_0$ coincides with that of a single-neuron linear network initialized at the same point $w_0$, which converges to the global minimum.

Although we assume the existence of $w_{GM}^*$ that activates all data in Theorem 4.2, this is easily satisfied in the overparameterized neural network. Then, there could be a deactivation if (8) does not hold for some $1 \leq j \leq n$. In that case, gradient flows of linear and ReLU networks do not coincide and the gradient flow of the ReLU network may not converge to the global minimum (see Example 5.2).

Nonetheless, the following theorem shows that a gradient flow initialized under some conditions does not converge to a local minimum which has no support vector of a crucial data, namely, a bad local minimum.

**Theorem 4.3 (Gradient flow does not converge to a bad local minimum).** Consider a gradient flow (4) under the same condition of Theorem 4.2. Let $w_{loc}^*$ be a local minimum and $S$ be the set of its support vectors. Now suppose the initialization point $w_0$ satisfies

$$
\max_{x_j \in S^c} \frac{y_j}{\|x_j\|} \geq \|w_0 - w_{GM}^*\|. \quad (9)
$$

Then, the gradient flow initialized at $w_0$ does not converge to $w_{loc}^*$.

For example, consider two local minima $w_1^*$ and $w_2^*$ with their sets of support vectors $S_1$ and $S_2$ such that $S_2 \subset S_1$. Then,

$$
\max_{x_j \in S_1^c} \frac{y_j}{\|x_j\|} \leq \max_{x_j \in S_2^c} \frac{y_j}{\|x_j\|}
$$

implies that (9) looks more feasible for $S_2$ than $S_1$. This suggests that a gradient flow may not converge to a ‘bad’ local minimum that has few support vectors without crucial data. With Theorem 3.7, we can say that a gradient flow does not converge to a local minimum with large loss value. There is another interpretation of this theorem. Since $L(w_{GM}^*) = 0$, i.e., $y_j = x_j^T w_{GM}^*$ for all $j$, (9) can be deformed to

$$
\max_{x_j \in S^c} \cos \theta_j > \frac{\|w_0 - w_{GM}^*\|}{\|w_{GM}^*\|}.
$$

(10)

where $\theta_j$ is the angle between $x_j$ and $w_{GM}^*$, i.e. $\cos \theta_j = \frac{x_j^T w_{GM}^*}{\|x_j\| \|w_{GM}^*\|}$. Then (10) says that a data $x_j$ which has large $\cos \theta_j$ value is not deactivated on the gradient flow, which is a candidate of crucial data.

Here we reveal the implicit bias of gradient flow dynamics with respect to the norm of initialization point. In terms of Proposition 4.1, the gradient flow with small norm tends to activate all data and the one with large norm does the opposite. Therefore, the data which aligns well on the $w_{GM}^*$ easily satisfy (10) and would be kept activated after it is activated with the help of small norm initialization. On the other hand, the gradient flow with large norm initialization does not allow the data to satisfy (10) and may converge to the minimum with few support vectors. In the middle of two realms, there is a critical region where $\|w_0\| \approx \|w_{GM}^*\|$. If the direction of $w_0$ is close with $w_{GM}^*$, then all data is
activated, and the gradient flow easily converge to the global minimum. In contrast, if the direction of \( w_0 \) is far from \( w^*_{GM} \), the right hand side of (10) is much larger. Thus, we can conjecture that the convergence is sensitive to the direction of \( w_0 \) in this critical region.

We end this subsection by emphasizing that the gradient flow initialized with infinitesimally small norm is a common setting of many researchers studying gradient flow dynamics (Gunasekar et al., 2017; Razin & Cohen, 2020; Arora et al., 2019; Woodworth et al., 2020; Li et al., 2020). Small norm initialization tries to activate all data (Proposition 4.1), which is recommended by Theorem 3.7. Additionally, the following proposition shows that every gradient flow initialized with sufficiently small norm converges to the same local minimum.

**Proposition 4.4.** Consider a gradient flow (4) of a single-neuron ReLU network under \( A_1, A_2, \) and \( A_3 \). Further suppose that the Hessian matrix \( \mathbf{H}(w(t)) \) has full rank on the gradient flow until it converges. Then there exists \( \delta > 0 \) such that for all \( w_0 \) with \( \|w_0\| < \delta \) and \( \nabla L(w_0) \neq 0 \), any gradient flow initialized at \( w_0 \) converges to the same point.

### 4.2. Norm increasing property

In Appendix C, we show that linear regression has implicit biases that the gradient flow initialized at zero converges to the minimum norm, with strictly increasing its norm until it converges. In this subsection, we extend the latter property to single-neuron ReLU networks.

**Theorem 4.5 (Norm increasing).** Consider a single-neuron ReLU network under \( A_1, A_2, \) and \( A_3 \). Consider a gradient flow \( w(t) \) moves from \( P \) to \( P \) across an activation boundary determined by \( x_0 \) at \( w_0 \) (either activating or deactivating). For \( \mathbf{H} := \sum_{x_i \sim P} x_i x_i^T \) and \( \hat{\mathbf{H}} := \sum_{x_i \sim \hat{P}} x_i x_i^T \), let \( \lambda_k, e_k \) be the eigenvalues and eigenvectors of \( \mathbf{H} \) and \( \hat{\mathbf{H}} \), respectively. Let \( w_0 = \sum_{k=1}^{d} c_k e_k = \sum_{k=1}^{d} \hat{c}_k \hat{e}_k \) has the coordinates by eigenvectors of \( \mathbf{H} \) and \( \hat{\mathbf{H}} \). Similarly, \( w_0^* = \sum_{k=1}^{d} c_k^* e_k \) and \( \hat{w}_0^* = \sum_{k=1}^{d} \hat{c}_k^* \hat{e}_k \) are the coordinates of the virtual minimum of \( P \) and \( \hat{P} \). Let \( \Delta e_k := \hat{e}_k - e_k \) be the difference of \( k \)-th eigenvectors of \( \mathbf{H} \) and \( \hat{\mathbf{H}} \). Similarly, \( \Delta w^* := w_0^* - \hat{w}_0^* \) is the difference of virtual minima of \( P \) and \( \hat{P} \). Now consider the following assumptions.

- **B1** \( \|\Delta e_k\| < \frac{\lambda^0_{max}(H) \cdot \|w_0\|}{\lambda^0_{min}(\hat{H}) \cdot \|w_0\|} \).
- **B2** \( y_0 < x_0^T w_0^* + \frac{1}{\|x_0\|} \min_{1 \leq k \leq d} \left( \frac{c_k^*}{|c_k|} - 1 \right) \frac{1}{\lambda_k} - \left( \frac{1}{\lambda^0_{max}(H)} \right) \cdot \|w_0 - w_0^*\| \cdot \|w_0\| \cdot \|w_0\| \).
- **B3** \( \hat{H} \) is full rank.
- **B4** \( x_0^T w_0^* < 0 \).

For the gradient flow initialized with infinitesimally small norm, suppose above assumptions hold on every boundary of partitions that gradient flow visits until it converges. Then, \( \|w(t)\| \) strictly increases until it converges.

By the **balancedness property** in deep ReLU networks shown by Du et al. (2018), the norm of each intermediate layers increase or decrease together. Therefore, if one intermediate layer and its backpropagated labels defined in Appendix B satisfy the conditions of Theorem 4.5, the norm of every layer increases together.

Also note that this norm increasing property does not contradict with the result of Vardi & Shamir (2021), since it does not need to converge to the minimum norm solution. In contrast to single-neuron linear networks, Example 5.2 shows that although the norm of a gradient flow strictly increases, it may not converge to the minimum norm solution.

### 4.3. Special case: \( d = 2 \)

We conclude this section with a special case \( d = 2 \). Although this case looks impractical, there are some useful geometric insights. In particular, we show the global convergence and norm-increasing property for the gradient flow initialized with infinitesimally small norm. We start with introducing an interesting lemma. In \( d = 2 \) case, for a gradient flow initialized with infinitesimally small norm, the following lemma shows once deactivated data never reactivated. We name it ‘No revisit lemma’, since it means the gradient flow never revisits a partition that already has been traversed.

**Lemma 4.6 (No revisit lemma for \( d = 2 \)).** Consider a single-neuron ReLU network with \( d = 2 \) under \( A_1, A_2, \) and \( A_3 \). Then a gradient flow initialized with infinitesimally small norm does not reactivate any deactivated data.

Using this lemma, we can obtain the global convergence and norm-increasing property, for \( d = 2 \) case. Note that this theorem does not require complicated assumptions other than \( A_1, A_2, \) and \( A_3 \), unlike Theorem 4.5.

**Theorem 4.7 (Global convergence for \( d = 2 \)).** Consider a single-neuron ReLU network with \( d = 2 \) under \( A_1, A_2, \) and \( A_3 \). Then a gradient flow initialized with infinitesimally small norm converges to the global minimum with strictly increasing its norm.

Unfortunately, Lemma 4.6 and Theorem 4.7 do not hold for \( d > 2 \) in general, as shown in Example 5.2 and Example 5.3. However, this proof technique is expected to be generalized in high dimension under some conditions, which is our future work.
5. Experiments

In this section, we provide some empirical examples for the results we proposed theoretically. More precisely, we show the necessity of assumptions of theorems with toy examples. Detail of these experiments is described in Appendix F.

We first observe the effect of the initialization norm. Recall that Theorem 4.5 and 4.7 consider gradient flows initialized with infinitesimally small norm. It is shown in the following example that the condition of small norm initialization is necessary.

Example 5.1 (Initialization with infinitesimally small norm is necessary). Consider a single-neuron ReLU network with \( d = 2 \) and \( n = 5 \). Detail data setting is described in Appendix F.2. In Figure 4, the level curves of the loss function (1) and gradient flows initialized with three different points are plotted by the black and the blue curves, respectively. The initialization points are denoted by black points (one point is out of scope). The two red dashed curves represent the gradient flows initialized with small norms. With regard to Proposition 4.1, Figure 4 illustrates that gradient flows of small norm tends to activate data and converge to local minima with many support vectors, while ones of large norm does the opposite. In addition, considering Proposition 4.4 and Theorem 4.7, we can observe the gradient flows initialized with infinitesimally small norms converge to the same point, global minimum (in the all-activated partition, thus maximum number of support vectors), while other gradient flows deactivate some data and converge to local minima. This implies the convergence of gradient flows depends on the norms of initial points.

The next example exhibits the case where the assumption of Theorem 4.2 does not hold, thus a data can be deactivated. Moreover, this would be a counter example of Theorem 4.7 if we omitted the assumption \( d = 2 \).

Example 5.2 (Deactivation of single-neuron linear and ReLU network). Consider a single-neuron ReLU network for \( n = d = 3 \), with given data

\[
X := [x_1 \ x_2 \ x_3] = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & 0 \\ 2 & 0 & 0 \end{bmatrix},
\]

\[
y := [y_1 \ y_2 \ y_3] = [0.05 \ 6 \ 0.5].
\]

For each \( x_i \), define \( h_i(w) := w^T x_i \). Then \( \{ w \in W \ | \ h_i(w) = 0 \} \) is the activation boundary of \( x_i \), and \( \{ w \in W \ | \ h_i(w) = y_i \} \) is the solution hyperplane of \( x_i \). Since \( X \) has full rank, there is a unique global minimum for both linear and ReLU networks, which is in the all-activated partition. However, we can observe that the gradient flow of the ReLU network deactivates \( x_1 \) during training. Note that gradient flows of the linear and ReLU networks coincide first, but bifurcate after deactivation of \( x_1 \) (See Figure 5(a)). Since \( x_1 \) is not reactivated again until the gradient flow converges, it is not a support vector of convergent local minimum. See Figure 5(a) and (c). Also see Figure 14 for more analysis of the examples.

In the next example, we show re-activation may occur for \( d > 2 \), which shows that Lemma 4.6 cannot be directly extended to higher dimension.

Example 5.3 (Reactivation of single-neuron linear and ReLU networks, for \( d > 2 \)). Consider a single-neuron ReLU network for \( n = 4 \) and \( d = 3 \) with given data

\[
X := [x_1 \ x_2 \ x_3 \ x_4] = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 1 & 2 & 0 \end{bmatrix},
\]

\[
y := [y_1 \ y_2 \ y_3 \ y_4] = [0.1 \ 0.2 \ 4 \ 0.1].
\]

For each \( x_i \), define \( h_i(w) := w^T x_i \) like in Example 5.2. Then we notice that the gradient flow of the ReLU network deactivates \( x_4 \) soon, and reactivates it later (see Figure 15(f)). Since all data are activated at the last, gradient flows of ReLU and linear networks converge to the same point, which is the unique global minimum. Note that the trace of two gradient flows are quite different although they initialize at and converge to the same points. See Figure 5(b) and (d). See Figure 15 for the detail result.

6. Conclusion and Future work

Understanding implicit bias of gradient descent has been an important goal in machine learning research. In this paper, we figured out implicit bias of gradient flow dynamics
Support Vectors and Gradient Dynamics for Implicit Bias in ReLU Networks

Figure 5. Gradient flows and its loss curves in Example 5.2 and Example 5.3. In (a), gradient flows of single-neuron linear and ReLU networks in Example 5.2 are plotted. Note that they coincide at the beginning, but distinguished after one data deactivated. In (b), gradient flows of single-neuron linear and ReLU networks in Example 5.3 are plotted. Although there is deactivation and gradient flows are distinct, they converge to the same point, which shows a reactivation. In (c) and (d), the loss curves of gradient flows in (a) and (b) are plotted. More detail of these experiments are plotted in Figure 14 and Figure 15 in Appendix.

in single-neuron ReLU networks with square loss. First, we showed the implicit bias of gradient flows in terms of support vectors of ReLU networks to answer why and how ReLU networks generalize well. Second, we revealed implicit bias of gradient flow dynamics with respect to the norm of initialization. We provide an initialization condition when a gradient flow keeps some data activated. Using this, we show that a gradient flow with some condition on initialization does not converge to bad local minima. Third, we extend the norm-increasing property of single-neuron linear networks to single-neuron ReLU networks under some conditions which is another implicit bias of gradient flows. Finally, for a special case $d = 2$, we prove the global convergence of the gradient flow and provide a novel proof technique.

Future work. First, this work should be extended for studying with another type of loss function, like cross-entropy loss used in classification tasks, since the role of ReLU activation in classification tasks could be a little different. Second, under some conditions, the global convergence for $d = 2$ case (Theorem 4.7) may be generalized to guarantee the global convergence for higher dimension.

Lastly, our result can be extended to understand the gradient flow dynamics in deep ReLU networks. Using the label-backpropagation we proposed in Appendix B, it is expected to apply this result for deep ReLU networks. We hope this work sheds a light to understand deep learning.
Support Vectors and Gradient Dynamics for Implicit Bias in ReLU Networks

References

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. *arXiv preprint arXiv:1811.04918*, 2018.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A., and Asuri, V. K. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3):292, 2019.

Anton, H. and Busby, R. C. *Contemporary linear algebra*. Wiley, 2003.

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pp. 254–263. PMLR, 2018.

Azulay, S., Moroshko, E., Nacson, M. S., Woodworth, B. E., Srebro, N., Globerson, A., and Soudry, D. On the implicit bias of initialization shape: Beyond infinitesimal mirror descent. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 468–477. PMLR, 18–24 Jul 2021.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. Spectrally-normalized margin bounds for neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6241–6250, 2017.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Belkin, M., Hsu, D., and Xu, J. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020.

Calin, O. *Deep learning architectures*. Springer, 2020.

Cornacchia, E., Hazla, J., Nachum, I., and Yehudayoﬀ, A. Regularization by misclassiﬁcation in relu neural networks. *arXiv preprint arXiv:2111.02154*, 2021.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V., et al. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997.

Du, S. S., Hu, W., and Lee, J. D. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Golowich, N., Rakhlin, A., and Shamir, O. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pp. 297–299. PMLR, 2018.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Gunasekar, S., Woodworth, B., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization, 2017.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018.

Hanin, B. and Rolnick, D. Complexity of linear regions in deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2596–2604. PMLR, 09–15 Jun 2019a.

Hanin, B. and Rolnick, D. Deep relu networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to ﬁnd them. In *International Conference on Learning Representations*, 2019.

Joki, K., Bagirov, A. M., Karmitsa, N., Mäkelä, M. M., and Taheri, S. Clusterwise support vector linear regression. *European Journal of Operational Research*, 287(1):19–35, 2020.

Kavitha, S., Varuna, S., and Ramya, R. A comparative analysis on linear regression and support vector regression. In *2016 online international conference on green engineering and technologies (IC-GET)*, pp. 1–5. IEEE, 2016.

Lacotte, J. and Pilanci, M. All local minima are global for two-layer relu neural networks: The hidden convex optimization landscape. *arXiv preprint arXiv:2006.05900*, 2020.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32:8572–8583, 2019.
A. Preliminaries

For a square matrix $A$, matrix exponential $e^A$ is defined by

$$e^A := \sum_{m=0}^{\infty} \frac{1}{m!} A^m.$$  

For a positive semidefinite matrix $A$, if $A$ is not invertible, we define its pseudo-inverse

$$A^+ := \sum_{k} \frac{1}{\lambda_k} e_k e_k^T,$$

where $(\lambda_k, e_k)$ are positive eigenvalues and corresponded eigenvectors of $A$. We use $\|\cdot\|_F$ to denote Frobenius norm of a matrix. $\odot$ denotes elementwise multiplication between two vectors. For a finite set $S$, $|S|$ means the number of elements of $S$.

B. Label-backpropagation of Deep ReLU Networks

In this section, we show how single-neuron linear and ReLU networks form building blocks of deep ReLU networks.

**Proposition B.1** (Label-backpropagation). Consider an $l$-layer ReLU network $f(\cdot ; W^{(1)}, \cdots, W^{(l)})$ with one data pair $(x, y)$ under square loss. Let $x^{(i)}$ and $\delta^{(i)}$ be input and output of $i$-th layer ($i = 2, 3, \cdots, l$), which are defined by

$$o^{(i)} := |W^{(i)}x^{(i)}|_+,$$
$$x^{(i)} := o^{(i)\ominus 1},$$

where $x^{(1)} := x$ and $o^{(1)} := |W^{(1)}x^{(1)}|_+$. Define $\delta^{(l)} := o^{(l)} - y^{(l)}$ and $y^{(l)} := y$. Now for $m = 1, 2, \cdots, l - 1$, recursively define $\delta^{(m)}$ and backpropagated-label $y^{(m)}$ by

$$\delta^{(m)} := \mathbb{I}_{x^{m+1} > 0} \odot W^{(m+1)^T} \delta^{(m+1)},$$
$$y^{(m)} := o^{(m)} - \delta^{(m)}.$$  

Then the gradient of $W^{(i)}$ of the $l$-layer ReLU network given by the data pair $(x, y)$ is same with the gradient of $W^{(i)}$ of a single-neuron ReLU or linear network $f^{(i)}$ given by the data pair $(x^{(i)}, y^{(i)})$ for $i = 1, 2, \cdots, l$.

**Proof.** The $l$-layer ReLU network $f(\cdot ; W^{(1)}, \cdots, W^{(l)})$ is modeled by

$$f(x; W^{(1)}, \cdots, W^{(l)}) = W^{(l)}[\cdots W^{(2)}[W^{(1)}x]_+ \cdots]_+.$$

For a given data pair $(x, y)$, square loss function is defined by

$$L(W^{(1)}, \cdots, W^{(l)}) = \frac{1}{2} \left\| f(x; W^{(1)}, \cdots, W^{(l)}) - y \right\|_F^2.$$
Then the gradient of $\mathbf{W}^{(m)}$ is computed by

\[
\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \mathbf{\delta}^{(l)} x^{(l)T}, \\
\frac{\partial L}{\partial \mathbf{W}^{(m)}} = \mathbf{\delta}^{(m)} x^{(m)T} \quad \text{for} \quad m = 1, 2, \cdots, l - 1.
\]

(11)

See Calin (2020) for a detail. Now see the last layer $\mathbf{W}^{(l)}$. Consider a single-neuron linear network $f^{(l)}(\cdot ; \mathbf{W}^{(l)})$ with a given data pair $(x^{(l)}, y^{(l)})$. The square loss provides gradient of $\mathbf{W}^{(l)}$ by

\[
\frac{\partial}{\partial \mathbf{W}^{(l)}} \frac{1}{2} \left\| f^{(l)}(x^{(l)} ; \mathbf{W}^{(l)}) - y^{(l)} \right\|_F^2 = \frac{\partial}{\partial \mathbf{W}^{(l)}} \frac{1}{2} \left\| \mathbf{W}^{(l)} x^{(l)} - y^{(l)} \right\|_F^2 = (\mathbf{W}^{(l)} x^{(l)} - y^{(l)}) x^{(l)T} = \mathbf{\delta}^{(l)} x^{(l)T},
\]

which is equal to (11). Therefore, the single-neuron linear network $f^{(l)}(\cdot ; \mathbf{W}^{(l)})$ with a given data pair $(x^{(l)}, y^{(l)})$ provides the same gradient (11) for $\mathbf{W}^{(l)}$.

Similarly, we can apply this argument for intermediate layers. For $m = 1, \cdots, l - 1$, consider a single-neuron ReLU network $f^{(m)}(\cdot ; \mathbf{W}^{(m)})$ with a data pair $(x^{(m)}, y^{(m)})$.

Then the gradient of $\mathbf{W}^{(m)}$ is computed by

\[
\frac{\partial}{\partial \mathbf{W}^{(m)}} \frac{1}{2} \left\| f^{(m)}(x^{(m)} ; \mathbf{W}^{(m)}) - y^{(m)} \right\|_F^2 = \frac{\partial}{\partial \mathbf{W}^{(m)}} \frac{1}{2} \left\| \mathbf{W}^{(m)} x^{(m)} + y^{(m)} \right\|_F^2 = (\mathbf{W}^{(m)} x^{(m)} + y^{(m)}) x^{(m)T} = \mathbf{\delta}^{(m)} x^{(m)T},
\]

which is same with (12). Therefore, the single-neuron ReLU network $f^{(m)}(\cdot ; \mathbf{W}^{(m)})$ with a data pair $(x^{(m)}, y^{(m)})$ provides same gradient (12) for $\mathbf{W}^{(m)}$.

To sum up, gradient of the $l$-layer ReLU network $f$ can be equivalently obtained from $(l - 1)$ single-neuron ReLU networks $f^{(m)}$ with $m = 1, 2, \cdots, l - 1$ and one single-neuron linear network $f^{(l)}$.

This proposition means that training $l$-layer ReLU networks can be understood as training $(l - 1)$ single-neuron ReLU networks and one linear network. See Figure 6.

---

Figure 6. Label-backpropagation. For a given data pair $(x, y)$, the gradient of an $l$-layer ReLU network is equivalent the gradient of $(l - 1)$ single-neuron ReLU networks and one linear network, where data pair is given by Proposition B.1.
Lemma B.2 (Du et al. (2018)). For a l-layer ReLU network, on a gradient flow, the difference of Frobenius norm of weight matrices of adjoined layers is invariant. i.e.,

$$\frac{d}{dt} \left( \left\| W^{(m)} \right\|_F^2 - \left\| W^{(m+1)} \right\|_F^2 \right) = 0.$$ 

By this property of deep ReLU network, if a intermediate single-layer ReLU network has norm-increasing property, then others also have it. This is one reason why study a single-layer ReLU network is crucial.

C. Single-Neuron Linear Networks

Single-neuron linear networks are trained as follows:

$$\arg\min_{w \in \mathcal{W}} L(w) := \frac{1}{2} \sum_{i=1}^{n} (w^T x_i - y_i)^2, \quad (13)$$

Then, the gradient flow initialized at $w_0$ is solution of the following differential equation

$$\frac{dw(t)}{dt} = -\nabla L = -Hw + q, \quad w(0) = w_0. \quad (14)$$

where $X = [x_1, \cdots, x_n]$ and

$$H := \sum_{i=1}^{n} x_i x_i^T = XX^T, \quad q := \sum_{i=1}^{n} y_i x_i = Xy.$$ 

Since $H = \sum_{i=1}^{n} x_i x_i^T$ is positive semi-definite, $L$ is convex with respect to $w$. Therefore, the set of minima of (13) is equal to the set of stationary points of $L$, which is $\{w \in \mathcal{W} \mid Hw = q\}$.

**Proposition C.1** (The manifold of stationary points). Consider the loss function $L(w)$ in (13) and suppose the data matrix $X$ has rank $r$. Then,

(i) any gradient flow converges to the global minimum.

(ii) If $r \geq d$, the stationary point is unique, which is the global minimum given by $w^* = H^{-1}q$.

(iii) Otherwise (i.e., $r < d$), the set of stationary points is a $(d - r)$-dimensional connected linear manifold containing $w^* = H^{1/2}q$.

**Proof.** This is a basic property of system of linear regression. See (Anton & Busby, 2003).

**Definition C.2** (Hyperrectangle of $H$). Consider (13) and suppose the Hessian matrix $H$ has full rank. For the unique global minimum $w^* = H^{-1}q$, let $e_1, e_2, \cdots, e_d$ be the eigenvectors of $H$ with direction such that $e_k^T w^* \geq 0$. Then the hyperrectangle of $H$ is defined by the region

$$\left\{ w = \sum_{k=1}^{d} c_k e_k \in \mathcal{W} \mid 0 \leq c_k \leq c_k^* \right\} \subset \mathcal{W}.$$ 

**Definition C.3.** Consider (13) and a gradient flow $w(t)$. Define a function $g(w) := -w^T w$. Then the norm-increasing subset is defined by the set

$$\{ w \in \mathcal{W} \mid g(w) < 0 \} \subset \mathcal{W}.$$ 

**Figure 7.** Examples of the hyperrectangle of $H$ and the norm-increasing subset. (a) In $\mathcal{W} = \mathbb{R}^3$, $e_1, e_2$ and $e_3$ are eigenvectors of $H$. The global minimum $w^*$ is denoted by a yellow star, and the green curve is the gradient flow initialized at $0$. Note that the gradient flow converges to $w^*$ inside the hyperrectangle. (b) In $\mathcal{W} = \mathbb{R}^2$, the norm-increasing subset is described by the blue solid ellipse, where the hyperrectangle of $H$ is described by the orange rectangle. Note that the axes of the rectangle and the norm-increasing ellipse are parallel to eigenvectors of $H$, and have same center $\frac{1}{2}w^*$.

The following proposition explains why the hyperrectangle of $H$ and norm-increasing subset (see Figure 7) are key ingredients for understanding single-neuron linear networks.

**Proposition C.4** (Properties of the hyperrectangle of $H$ and the norm-increasing subset). Suppose we have a unique minimum $w^*$ for (13). Then the following statements hold.

1. The norm of a gradient flow increases if and only if it is inside the norm-increasing subset.

2. The norm-increasing subset forms an $d$-dimensional ellipsoid, where axes of the ellipsoid are parallel to the eigenvectors of $H$.

3. If a gradient flow initializes in the hyperrectangle of $H$, it never escapes the hyperrectangle.

4. The ellipsoid contains the hyperrectangle. In particular, their center points coincide to $\frac{1}{2}w^*$ and every vertex of the hyperrectangle lies on the boundary of the norm-increasing subset.
Proof. 1. Consider the time derivative of $\|w(t)\|^2$.

$$\frac{d}{dt} \|w\|^2 = 2w^T \dot{w}$$

$$= -2g(w)$$

Therefore, the norm of the gradient flow $w$ increases if and only if $g(w) < 0$, which means $w$ is inside the norm-increasing subset.

2. Since $\nabla L = Hw - q = H(w - w^*)$ by Proposition C.1,

$$g(w) = -w^T \dot{w}$$

$$= w^T H(w - w^*)$$

$$= (w - \frac{1}{2} w^*)^T H (w - \frac{1}{2} w^*) - \frac{1}{4} w^T H w^*.$$  

Therefore, $g(w) < 0$ is an ellipsoid centered at $\frac{1}{2} w^*$, where its axes are parallel to eigenvectors of $H$ by Anton & Busby (2003).

3. Since $Hw^* = q$, (14) becomes

$$w = -H(w - w^*), \quad w(0) = w_0.$$  \tag{15}

For $k$-th eigenvectors $e_k$ of $H$, define $c_0^k := w_0^T e_k$ and $c_k^2 := w^* e_k$. From (Zill, 2020), the closed form solution of (15) is given by

$$w(t) = e^{-Ht}(w_0 - w^*) + w^*$$  \tag{16}

$$= \sum_{k=1}^d e^{-\lambda_k t} e_k^T (w_0 - w^*) + w^*$$  \tag{17}

$$= \sum_{k=1}^d (e^{-\lambda_k t} c_k^0 - c_k^*) e_k^T (w_0 - w^*) + c_k^* e_k$$

$$= \sum_{k=1}^d \left( c_k^* - e^{-\lambda_k t} (c_k^* - c_0^k) \right) e_k.$$  

Since $w_0$ is inside the hyperrectangle, we get $0 \leq c_0^k \leq c_k^*$. Then $c_k^* - e^{-\lambda_k t} c_0^k \leq c_k^*$ and $c_k^* - e^{-\lambda_k t} (c_k^* - c_0^k) = (1 - e^{-\lambda_k t}) c_k^* + e^{-\lambda_k t} c_0^k \geq 0$ imply $0 \leq w(t)^T e_k = c_k^* - e^{-\lambda_k t} (c_k^* - c_0^k) \leq c_k^*$ for all $t \geq 0$. Therefore, the gradient flow never escapes the hyperrectangle.

4. Since the hyperrectangle is convex, it is enough to show that the vertices of the hyperrectangle lie on the boundary of the ellipsoid. The hyperrectangle has $2^d$ vertices. Let $v_j$ be a vertex. Then it can be represented as

$$v_j = \sum_{k=1}^d j_k c_k^* e_k, \quad j_k = 0 \text{ or } 1$$

where $c_k^* = w^* e_k$. Subsequently,

$$g(v_j) = v_j^T H (v_j - w^*)$$

$$= \left( \sum_{k=1}^d j_k c_k^* e_k \right)^T \left( \sum_{k=1}^d \lambda_k e_k e_k^T \right) \left( \sum_{k=1}^d (1 - j_k) c_k^* e_k \right)$$

$$= \sum_{k=1}^d j_k (1 - j_k) \lambda_k c_k^2$$

$$= 0.$$  

Therefore, all vertices of the hyperrectangle lie on the boundary of the norm-increasing ellipsoid, thus the hyperrectangle is contained in the ellipsoid.

We now rewrite the well-known properties of least-square linear regression in Gunasekar et al. (2017).

**Theorem C.5** (Norm increasing and implicit bias of single-neuron linear networks). Consider (14) and suppose $w_0 = 0$. Then,

1. the converged point is the global minimum with minimum Euclidean norm.

2. The norm of gradient flow $\|w\|$ monotonically increases until it converges.

Proof. 1. If $X$ is full rank, since there is only one stationary point, the statement trivially holds. Now suppose $X$ is not full rank, and recall Reduction principle (Theorem D.1). Let $\overline{w}^*$ be the convergent point of the gradient flow. Since the gradient flow lies on $0 + \overline{w}$, $\overline{w}^* \in \overline{w}$. Let $w^*$ be another minimum of $L$ in $W$. By the last statement of Theorem D.1, $w^* = \overline{w}^* + v$ for some $v \in \ker(\varphi)$. Then,

$$\|w^*\|^2 = \|\overline{w}^* + v\|^2$$

$$\leq \|\overline{w}^*\|^2 + \|v\|^2$$

$$\geq \|\overline{w}^*\|^2.$$  

Therefore, the convergent point $\overline{w}^*$ is the minimum Euclidean norm stationary point.

2. Since the gradient flow is contained in $\overline{w}$, consider the hyperrectangle of $H$ in $\overline{w}$. Since $0 \in \overline{w}$, the gradient flow initializes in the hyperrectangle. By Proposition C.4, the gradient flow never escapes the hyperrectangle, which is contained in the norm-increasing subset. Therefore, the norm of the gradient flow increases until it converges $\overline{w}^*$.
D. Reduction Principle

In this section, we explain why we can assume $X$ to have full rank without loss of generality.

![Figure 8. Reduction principle. There are two gradient flows denoted by dashed green curves. Note that the trajectory of the gradient flows coincide in $\overline{W}$ through the projection $\varphi$.](image)

**Theorem D.1** (Reduction principle). *In the second case of Proposition C.1, there exists a projection $\varphi : \mathcal{W} \to \overline{W}$ and orthogonal decomposition $\mathcal{W} = \overline{W} \oplus \ker(\varphi)$ such that*

1. for any $w \in \overline{W}$ and $v \in \ker(\varphi)$, $L(w) = L(w + v)$.
2. The gradient flow (14) lies on $w_0 + \overline{W}$.
3. $L$ has a unique minimum $\overline{w}^*$ in $\overline{W}$.
4. For every minimum $w^*$ of $L$, $\varphi(w^*) = \overline{w}^*$.

*In other words, the dynamics of the gradient flow $w(t)$ in $\mathcal{W} = \mathbb{R}^d$ can be considered in $\overline{W} \cong \mathbb{R}^r$.*

**Proof.** Since $r < d$, the Hessian matrix $H = XX^T$ has $r$ positive eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_d = 0.$$  

Now consider spectral decomposition of $H = \sum_{k=1}^{r} \lambda_k e_k e_k^T$.

Define a projection operator $\varphi : \mathcal{W} \to \overline{W}$ by

$$\varphi : \mathcal{W} \to \overline{W}$$

$$w \mapsto \sum_{k=1}^{r} (w^T e_k) e_k^T.$$  

Then it is easily checked that $\varphi$ is onto. Therefore, we get $\overline{W} := \varphi(\mathcal{W}) \cong \mathbb{R}^r$ and a direct sum decomposition $\mathcal{W} = \overline{W} \oplus \ker(\varphi)$. Now we prove the statements of the theorem.

1. For $v \in \ker(\varphi)$,

$$L(w + v) = \frac{1}{2} \sum_{i=1}^{n} (w^T x_i^2 - y_i^2)^2$$

$$= \frac{1}{2} \sum_{i=1}^{n} (w^T x_i + v^T x_i - y_i)^2$$

$$= \frac{1}{2} \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

$$= L(w).$$

2. Let $w_\infty$ be a stationary point of $L$. Then, $H w_\infty = q$ implies $\dot{w} = -\nabla L(w) = -H w + q = -H (w - w_\infty) \in \text{range}(H)$, Thus $\varphi(\dot{w}) = \dot{w}$. Then,

$$\varphi(w - w_0) = \varphi(w(t) - w(0))$$

$$= \varphi(\int_{0}^{t} w(s)ds)$$

$$= \int_{0}^{t} \varphi(w(s))ds$$

$$= \int_{0}^{t} \dot{w}(s)ds$$

$$= w(t) - w(0)$$

$$= w - w_0.$$  

Therefore, $(w - w_0)$ is in $\overline{W}$, which means the gradient flow lies on $w_0 + \overline{W}$.

3. From the construction, $\overline{W} = \text{range}(H)$. Therefore, Proposition C.1 gives the unique stationary point $\overline{w}^*$ in $\overline{W}$.

4. Let $w^*$ be a minimum of $L$. From orthogonal decomposition $\mathcal{W} = \overline{W} \oplus \ker(\varphi)$, $w^* = \varphi(w^*) + (w^* - \varphi(w^*))$. By the first statement, $L(w^*) = L(\varphi(w^*) + (w^* - \varphi(w^*))) = L(\varphi(w^*)^*$. Therefore, $\varphi(w^*)$ is a minimum in $\overline{W}$. Since $\overline{W}$ has a unique minimum $\overline{w}^*$, we conclude $\varphi(w^*) = \overline{w}^*$.

In particular, by (ii), we can say that the geometry of the gradient flow can be reduced to $\overline{W}$. See Figure 8.

**Corollary D.2** (Reduction principle for ReLU networks). *Consider a single-neuron ReLU network with $\{x_i\}_{i=1}^{n}$. Suppose $X$ has rank $r < d$. Then the dynamics of gradient flows in $\mathcal{W} = \mathbb{R}^d$ can be reduced into $\overline{W} = \mathbb{R}^r$, by the projection $\varphi$ in Theorem D.1. In other words, for overparameterized setting, we can reduce it to critically determined case with $d = r$.  

Figure 9. Proof of ‘Not on boundary lemma’ (Lemma 3.2). A local minimum $w^*$ is assumed to lie on an activation boundary of $x^*$. Note that $x^*$ is activated in $P_+$, and deactivated in $P_-$. 

Proof. The proof is exactly same with the proof of Theorem D.1. For the last statement, overparameterized setting implies rank$(X) \leq n < d$.  

Thanks to this corollary, we can assume $X$ has full rank without loss of generality (A3).

E. Proofs of Lemmas, Propositions and Theorems

E.1. Loss Landscape and Support Vectors

Proposition E.1. Consider $n$ training data $\{x_i\}_{i=1}^n$ in $\mathbb{R}^d$ under A1. Then the number of partitions in the parameter space is at most 

$$2 \sum_{k=0}^{d-1} \binom{n-1}{k}. \quad (18)$$

Proof. Zaslavsky’s Theorem tells general $n$ hyperplanes in $\mathbb{R}^d$ generate at most $\sum_{k=0}^{d-1} \binom{n}{k}$ partitions. From here, we induce the case for central hyperplanes. Consider a hyperplane $H$ in $\mathcal{W}$, and define two parallel hyperplanes $H_+$ and $H_-$ such that $H$ is between them. Every partition is on one side of $H$ or another, so it either intersects with $H_+$ or $H_-$. Since either of $H_+$ or $H_-$ is $(d-1)$ dimensional, it is divided by other $n-1$ hyperplanes into at most $\sum_{k=0}^{d-1} \binom{n-1}{k}$ partitions. Therefore, the total number of partitions in $\mathcal{W}$ is at most $2 \sum_{k=0}^{d-1} \binom{n-1}{k}. \quad \Box$

Proof of Lemma 3.1. 1. Let $P$ be a linearly partitioned region with an activation pattern $\mathbb{I}_{\{w^T x > 0\}}$. Then for $w_1, w_2 \in P$ and $0 \leq \lambda \leq 1$, every interpolation point $\lambda w_1 + (1-\lambda) w_2$ keeps the value $\mathbb{I}_{\{w^T x > 0\}}$ and thus it is contained in $P$. Similarly, for any positive scalar $\alpha > 0$, $\alpha w$ keeps the value $\mathbb{I}_{\{w^T x > 0\}}$, thus it is in $P$. Therefore, each linearly partitioned region is convex and unbounded. Finally, from (3), we can check that $H(\alpha w) = H(w)$ and $q(\alpha w) = q(w)$ for any $\alpha > 0$.

2. Note that ReLU is convex and $w^T x$ is affine with respect to $w$, thus $[w^T x]_+$ is convex again. For $y \leq 0$, a function $(|t| + y)^2$ is non-decreasing convex with respect to $t \in \mathbb{R}$. Finally, by using the fact that the composition of convex and convex nondecreasing function is again convex, we conclude that $f_i(w) = \frac{1}{2}([w^T x_i]_+ - y_i)^2$ is convex if $y_i \leq 0$.

Now suppose we have $m$ positive labels among $n$ training data, i.e., $y_i > 0$ for $1 \leq i \leq m$ and $y_i \leq 0$ for $m+1 \leq i \leq n$. Consider $\sum_{i=1}^n f_i(w)$ with partitions generated by $\{x_i\}_{i=1}^m$. Since it is convex on each partition, the number of partitions contain a local minimum is at most 

$$2 \sum_{i=0}^{d-1} \binom{m-1}{i}$$

by Proposition E.1. For the last $n-m$ data, by the above statement 1., we know $\sum_{i=m+1}^n f_i(w)$ is globally convex. Considering the sum of a convex function and a piecewise convex function, the number of partitions contain a local minimum increases at most 1. Therefore, the number of partitions that contain a local minimum is at most 

$$2 \sum_{i=0}^{d-1} \binom{m-1}{i} + 1.$$ 

Proof of Lemma 3.2. We will prove this by contradiction. Suppose a local minimum $w^*$ is on an activation boundary, and let $x^*$ be the corresponding data of the activation boundary with the label $y^* > 0$. Consider two perturbed vectors $w_\pm = w^* \pm \varepsilon x^*$ with sufficiently small $\varepsilon > 0$. Let $P_+$ and $P_-$ be adjoined linearly partitioned regions, where each partition contains $w_+$ and $w_-$, respectively (see Figure 9). Then, we have 

$$\nabla L(w_+) = \sum_{x_i \sim P_+} (w^T_+ x_i - y_i) x_i,$$

$$\nabla L(w_-) = \sum_{x_i \sim P_-} (w^T_- x_i - y_i) x_i.$$ 

Furthermore, by Lemma 2.1, we have 

$$\nabla L(w_+) = \sum_{x_i \sim P_+} (w^T_+ x_i - y_i) x_i + (w^*_+ x^* - y^*) x^*.$$ 

Since $L(w)$ is convex on each linearly partitioned region,
following properties hold within each partition:

\[ 0 \geq L(w^*) - L(w_+) \geq \nabla L(w_+)^T (w^* - w_+) , \]
\[ 0 \geq L(w^*) - L(w_-) \geq \nabla L(w_-)^T (w^* - w_-) \]

which imply \( \nabla L(w_+)^T x^* \geq 0 \) and \( \nabla L(w_-)^T x^* \leq 0 \), respectively. Therefore, we get \((\nabla L(w_+) - \nabla L(w_-))^T x^* \geq 0 \). Now taking \( \varepsilon \to 0^+ \), we have \( w_+, w_- \to w^* \) and

\[
\lim_{\varepsilon \to 0^+} (\nabla L(w_+) - \nabla L(w_-)) = (w^*^T x^* - y^*) x^* = -y^* x^* \\
\]

since \( w^*^T x^* = 0 \). Therefore, we get

\[ 0 \leq [(\nabla L(w_+) - \nabla L(w_-))^T x^*]_{\varepsilon \to 0^+} - y^* \| x^* \|^2 < 0, \]

which is a contradiction. This completes the proof. \( \square \)

**Proof of Proposition 3.3.** Since \( L \) is convex on each partition, we can apply Proposition C.1. Thus we need to check whether \( w^*_P \) is in \( P \), which is the (6). More precisely, by Proposition C.1 again, the set of local minimum forms a \((d - r_P)\)-dimensional connected linear manifold. \( \square \)

![Figure 10](image_url)

**Figure 10.** A necessary condition of two local minima in adjacent partitions \( P_- \) and \( P_+ \). If each partition contains its local minimum, (7) holds.

**Proof of Proposition 3.4.** See Figure 10. By Lemma 2.1, \( \{ x \sim P_+ \} = \{ x \sim P_- \} \cup \{ x_0 \} \). Using reduction principle for ReLU (Corollary D.2), we can assume \( H \) is invertible. To use contradiction, we assume there is a local minimum \( w^*_+ \) in \( P_+ \) and derive the reverse inequality. By Proposition 3.3, we have

\[ w^*_+ = H^{-1} q, \]
\[ w^*_+ = (H + x_0 x_0^T)^{-1} (q + y_0 x_0). \]

Using Sherman-Morrison formula, we get

\[ w^*_+ = (H + x_0 x_0^T)^{-1} (q + y_0 x_0) \]

\[ = \left( H^{-1} - \frac{H^{-1} x_0 x_0^T H^{-1}}{1 + x_0^T H^{-1} x_0} \right) (q + y_0 x_0) \]
\[ = w^*_+ - \frac{H^{-1} x_0 x_0^T w^*_+}{1 + x_0^T H^{-1} x_0} + y_0 \frac{H^{-1} x_0}{1 + x_0^T H^{-1} x_0} H^{-1} x_0 \]
\[ = w^*_+ - \frac{H^{-1} x_0 x_0^T w^*_+}{1 + x_0^T H^{-1} x_0} + y_0 \left( 1 - \frac{x_0^T H^{-1} x_0}{1 + x_0^T H^{-1} x_0} \right) H^{-1} x_0 \]
\[ = w^*_+ - \frac{H^{-1} x_0 x_0^T w^*_+}{1 + x_0^T H^{-1} x_0} + \frac{y_0 H^{-1} x_0}{1 + x_0^T H^{-1} x_0}. \]

Since \( w^*_+ \in P_+ \) and \( w^*_+ \in P_- \), we know \( w^*_+^T x_0 \leq 0 \) and \( w^*_+^T x_0 > 0 \). Therefore,

\[ x_0^T w^*_+ = x_0^T \left( w^*_+ - \frac{H^{-1} x_0 x_0^T w^*_+}{1 + x_0^T H^{-1} x_0} + \frac{y_0 H^{-1} x_0}{1 + x_0^T H^{-1} x_0} \right) \]
\[ = x_0^T w^*_+ + y_0 x_0^T H^{-1} x_0 \]
\[ > 0. \]

Since \( H^{-1} \) is also positive semi-definite, \( 1 + x_0^T H^{-1} x_0 > 0 \) and we get

\[ y_0 > - \frac{x_0^T w^*_+}{x_0^T H^{-1} x_0} = - \frac{x_0^T H^{-1} q}{x_0^T H^{-1} x_0} > 0. \]

\( \square \)

**Proof of Theorem 3.5.** From (13) and (1), let \( L_l \) and \( L_r \) be the loss functions of single-neuron linear and ReLU networks, which are defined by

\[ L_l (w) := \frac{1}{2} \sum_{i=1}^{n} (w^T x_i - y_i)^2 \]
\[ L_r (w) := \frac{1}{2} \sum_{i=1}^{n} ([w^T x_i]_+ - y_i)^2. \]

Let \( w^*_l \) and \( w^*_r \) be the global minimum of \( L_l (w) \) and \( L_r (w) \), respectively. Let \( S \) be the set of support vectors of \( w^*_+ \). Then,

\[ L_l (w^*_+) = \frac{1}{2} \sum_{i \in S} (w^*_+^T x_i - y_i)^2 \]
\[ = \frac{1}{2} \sum_{i \in S} (w^*_+^T x_i - y_i)^2 + \frac{1}{2} \sum_{i \notin S} (w^*_+^T x_i - y_i)^2 \]
\[ \geq \frac{1}{2} \sum_{i \in S} (w^*_+^T x_i - y_i)^2 + \frac{1}{2} \sum_{i \notin S} y_i^2 \]
\[ = L_r (w^*_+) \]
\[ \geq L_r (w^*_+) \]

\( \square \)
where the first inequality comes from \( y_i > 0 \) and \( w_i^* x_i \leq 0 \) for \( x_i \not\in S \), and the last inequality comes from the fact that \( w_i^* \) is the global minimum of \( L_r \). Therefore, the global minimum of single-neuron ReLU networks achieve equal or smaller loss than global minimum of single-neuron linear networks.

Proof of Theorem 3.7. It is almost same with the proof of Theorem 3.5.

\[
L(w^*_2) = \min_{w} \sum_{x_i \in S_2} (w^T x_i - y_i)^2 + \sum_{x_i \in S_1^c} (0 - y_i)^2
\]
\[
= \min_{w} \sum_{x_i \in S_2} (w^T x_i - y_i)^2 + \sum_{x_i \in S_1 - S_2} (0 - y_i)^2
\]
\[
= \min_{w} \sum_{x_i \in S_1} (w^T x_i - y_i)^2
\]
\[
\geq \min_{w} \sum_{x_i \in S_1} (w^T x_i - y_i)^2
\]
\[
= L(w^*_1).
\]

\]

E.2. Weight initialization

Proof of Proposition 4.1. Recall the positive homogeneity described in Proposition 3.1: \( \alpha w \in P \) for \( \alpha > 0 \). From (3), we get

\[
-\nabla L(\alpha w)^T x_j = x_j^T (-H\alpha w + q)
\]
\[
= -\alpha x_j^T Hw + x_j^T q.
\]

Therefore, \( -\nabla L(\alpha w)^T x_j \geq 0 \) if and only if

\[
\alpha \leq \frac{x_j^T q}{x_j^T H w} = \alpha_j^*.
\]

This means that if for \( 0 < \alpha < \min \alpha_j^* \), \( -\nabla L(\alpha w)^T x_j \) \( < 0 \) for all \( j \). Similarly, if \( \alpha > \max \alpha_j^* \), \( -\nabla L(\alpha w)^T x_j \) \( > 0 \) for all \( j \). Namely, for a gradient flow \( w(t) \) with large enough \( \|w\| \), it moves to a direction that deactivates all data. Similarly, for \( \|w_0\| \) is small enough, it moves to a direction that activates all data. See Figure 3.

Proof of Theorem 4.2. Consider a situation that the gradient flow \( w(t) \) initialized at \( w_0 \) is in the \( l \)-th partition (\( w_0 \) is in the 0-th partition). Let \( 0 < t_1 < t_2 < \cdots < t_l \) be the time when \( w(t) \) crosses an activation boundary. Then \( w(t) \) is contained in one partition for \( t_i \leq t_i+1 \). Define \( H_i := H_i(\cdot t_i, t_{i+1}) \). Since \( w_{GM} \) satisfies \( x_j^T w_{GM} = y_j \), the gradient flow \( w(t) \) is represented by

\[
\begin{align*}
w(t_1) - w_{GM} &= e^{-H_{01}}(w_0 - w_{GM}) \\
w(t_2) - w_{GM} &= e^{-H_{12}}(w(t_1) - w_{GM}) \\
& \vdots \\
w(t_l) - w_{GM} &= e^{-H_{l-1}}(w(t_{l-1}) - w_{GM}).
\end{align*}
\]

Therefore,

\[
w(t_i) - w_{GM} = e^{-H_{i-1}(t_{i-1} - t_{i-1})} e^{-H_{i-2}(t_{i-2} - t_{i-1})} \cdots e^{-H_{01}}(w_0 - w_{GM}).
\]

To investigate \( h_j(t) := x_j^T w(t) \), consider

\[
\begin{align*}
\|x_j^T (w(t_i) - w_{GM})\| &= \|x_j e^{-H_{i-1}(t_{i-1} - t_{i-1})} e^{-H_{i-2}(t_{i-2} - t_{i-1})} \cdots e^{-H_{01}}(w_0 - w_{GM})\|
\leq \|x_j e^{-H_{i-1}(t_{i-1} - t_{i-1})} e^{-H_{i-2}(t_{i-2} - t_{i-1})} \cdots e^{-H_{01}}(w_0 - w_{GM})\|
\end{align*}
\]

The first term of the last line can be approximated by a simple exponential term.

\[
\begin{align*}
\|x_j e^{-H_{i-1}(t_{i-1} - t_{i-1})} e^{-H_{i-2}(t_{i-2} - t_{i-1})} \cdots e^{-H_{01}}(w_0 - w_{GM})\|
\leq \|x_j e^{-H_{i-1}(t_{i-1} - t_{i-1})} e^{-H_{i-2}(t_{i-2} - t_{i-1})} \cdots e^{-H_{01}}(w_0 - w_{GM})\|
\end{align*}
\]

We modify the conventional inequality \( e^{-A^T x} \leq -\lambda_{\min}(A^T) \|x\| \) to the inequality that \( e^{-A^T x} \leq -\lambda_{\min}(A^T) \|x\| \) when \( x \) is in \( \text{col}(A) \), which is induced by the reduction principle(Theorem D.1). Indeed, the inequality proposed above holds since \( x_j^T e^{-H_{i-1}(t_{i-1} - t_{i-1})} e^{-H_{i-2}(t_{i-2} - t_{i-1})} \cdots e^{-H_{01}(t_{i-1} - t_{i-1})} \) is contained in \( \text{col}(H_{i-1}) \). Then,

\[
\begin{align*}
\|x_j^T (w(t_i) - w_{GM})\| &\leq \|x_j\| \cdot \|w_0 - w_{GM}\| e^{-m_{lt_1}}
\leq \|x_j\| e^{-m_{lt_1}}
\end{align*}
\]

Therefore, we get

\[
\begin{align*}
h_j(t_i) &= y_j - x_j^T (w(t_i) - w_{GM}) > 0.
\end{align*}
\]

This shows that \( x_j \) is activated for \( 0 \leq t \leq t_i \). Since \( l \) is arbitrary, we show that \( h_j(t) > 0 \) for \( t \geq 0 \).
Now we prove the last statement of the theorem. If (8) holds for all \( j = 1, \ldots, n \), there is no deactivation on the gradient flow. Therefore, it coincides with the gradient flow of a single-neuron linear network with the same initial point \( w_0 \), which converges to the global minimum by Proposition C.1.

**Proof of Theorem 4.3.** By Theorem 4.2, it is enough to show that there is some \( x_j \in S^c \) satisfying (8). If (9) holds, there exists some \( x_j \in S^c \) which is always activated on the gradient flow \( w(t) \). Therefore, the gradient flow cannot converge to \( w_{\text{loc}}^* \).

**Proof of Proposition 4.4.** From Proposition 4.1, we know that a gradient flow with small \( \|w(t)\| \) tends to activate all data. More precisely, we first show that a gradient flow initialized with infinitesimally small norm moves to the all-activated partition soon. Recall (4) and taking \( \|w_0\| \searrow 0 \) with keeping direction of \( w_0 \), we get

\[
\lim_{\|w_0\| \to 0^+} \dot{w}(0) = \lim_{\|w_0\| \to 0^+} -\nabla L(w_0) = \lim_{\|w_0\| \to 0^+} -H(w_0)w_0 + q(w_0) = q(w_0) = \sum_{i=1}^n \mathbb{1}_{\{w_0^T x_i > 0\}} y_i x_i \\
\neq 0.
\]

Since \( x_i \geq 0 \), if \( \|w_0\| \to 0^+ \), then \( x_j^Tw = x_j^T \sum_{i=1}^n \mathbb{1}_{\{w_0^T x_i > 0\}} y_i x_i > 0 \) for all \( x_j > 0 \), which means the gradient flow moves to the direction that activating every data at \( t = 0 \). Thus it activates all data immediately, at the beginning.

Thus WLOG we can suppose \( w_0 \) is in the all-activated partition \( P_0 \). Recall (16) and taking \( \|w_0\| \to 0 \), we get

\[
w(t) = (I - e^{-Ht})w^*
\]

in \( P_0 \). Let \( w_1 \) be the point that the gradient flow reaches a boundary of \( P_0 \), if it does. Then for given \( \varepsilon > 0 \), there exists \( \delta_1 > 0 \) such that \( \|w_0\| < \delta_1 \) implies \( \|w(t) - w_1\| < \varepsilon \) by continuity. We can apply the same argument for each linearly partitioned region consecutively. Therefore, gradient flows initialized with sufficiently small \( \|w_0\| \) converge to the same point.

**E.3. Norm increasing property**

**Proof of Theorem 4.5.** Before we start, recall Proposition C.4. To prove norm-increasing property, it is enough to show that the gradient flow is contained in the associated hyperrectangle of each partition. More precisely, since a gradient flow never escapes associated hyperrectangle if it was initialized in the hyperrectangle, we only need to consider the boundary of partitions. See Figure 11.

Now we prove the gradient flow is always contained in the hyperrectangle of each partition by induction. First, the gradient flow initializes with infinitesimally small norm is contained in the hyperrectangle. By induction hypothesis, suppose \( w_0 \) is contained in the hyperrectangle of \( P \) and assumptions B1, B2 hold. We want to show that \( w_0 \) is in the hyperrectangle of \( P \). In other words, \( 0 < \dd c_k < \dd c_k^* \) for all \( k = 1, 2, \ldots, d \). The proof consists of two parts. Note that \( H = H \pm x_0 \dd x_0^T \) whether \( x_0 \) is activated or deactivated.

**Claim 1.** \( \dd c_k > 0 \):

For \( k \)-th eigenvector \( w_k \),

\[
\dd \lambda_k \dd c_k - \lambda_k c_k = w_0^T (\dd \lambda_k \dd e_k - \lambda_k e_k) = w_0^T (H \dd e_k - H e_k) = w_0^T ((H \pm x_0 \dd x_0^T)(e_k + \Delta e_k) - H e_k) = w_0^T (H \Delta e_k \pm x_0 \dd \hat{x}_k) (w_0^T x_0) = w_0^T H \Delta e_k
\]

since \( w_0^T x_0 = 0 \) \( (w_0 \) is on the activation boundary). However, from B1,

\[
|w_0^T H \Delta e_k| \leq \|w_0\| \cdot \|H\|_2 \cdot \|\Delta e_k\| < \|w_0\| \cdot \|H\|_2 \cdot \lambda_{\min}^2(H) \cdot \lambda_{\max}^{-1}(H) \cdot c_k
\]
\[ \lambda_{\min}(H) c_k \]
\[ \leq \lambda_k c_k. \]

Therefore, \( \lambda_k \tilde{c}_k = \lambda_k c_k + w_0^T H \Delta e_k > 0. \)

**Claim 2.** \( \tilde{c}_k < \tilde{c}_k^* : \)
\[
\tilde{c}_k - \tilde{c}_k = (w_p - w_0)^T e_k \\
= (w_p + \Delta w^* - w_0)^T (e_k + \Delta e_k) \\
= (w_p - w_0)^T e_k + (w_p + \Delta w^* - w_0)^T \Delta e_k \\
+ \Delta w^T e_k \\
= (c_k^* - c_k) + (w^* - w_0)^T \Delta e_k + \Delta w^T e_k.
\]

For the last two terms of the right hand side,
\[
\left| (\Delta w^* - w_0)^T \Delta e_k + \Delta w^T e_k \right| \\
< \left| \Delta w^T e_k \right| + \left| (\Delta w^* - w_0)^T \Delta e_k \right| \\
= \left| (\Delta w^* - w_0)^T e_k \right| + \left| (\Delta w^* - w_0)^T \Delta e_k \right| \\
= \left| \left( H^T x_0 x_0^T - H^{-1} q \right)^T e_k \right| \\
+ \left| (\Delta w^* - w_0)^T \Delta e_k \right| \\
= \left| \left( H^{-1} q \right)^T e_k \right| \\
= \left| \left( H^T x_0 x_0^T - H^{-1} q \right)^T e_k \right| \\
= \left| (\Delta w^* - w_0)^T \Delta e_k \right| \\
< (\Delta w^* - w_0)^T \Delta e_k \\
< \lambda_k c_k.
\]

where the second to last inequality holds by triangle inequality, \( B1 \) and \( B4. \) The last inequality holds by \( B2 : \)
\[
0 \leq \left| (\Delta w^* - w_0)^T \Delta e_k \right| \\
< \frac{1}{\lambda_k} c_k^* - c_k.
\]

Finally, Claim 1 and 2 with induction hypothesis prove \( w_0 \) is in the hyperrectangle of \( P \). Therefore, the gradient flow always contained in hyperrectangles and thus \( \|w\| \) increases until it converges.

### E.4. Special case : \( d = 2 \)

In this section we consider a special case, which is \( d = 2 \). In \( \mathbb{R}^2 \), there exists a special structure in the partitioned parameter space. More precisely, there is an ‘order’ for activation and deactivation, as the following lemma states.

**Lemma E.2 (Ordering of partitions in \( \mathbb{R}^2 \)).** Consider the partitioned parameter space \( \mathcal{W} = \mathbb{R}^2 \) under \( A1 \). Then we can impose an relative order between partitions. In particular, for any two partitions \( P_1 \) and \( P_2 \) in the 2nd (or 4th) quadrant, either \( \{ x_i \mid x_i \sim P_1 \} \subset \{ x_i \mid x_i \sim P_2 \} \) or \( \{ x_i \mid x_i \sim P_1 \} \supset \{ x_i \mid x_i \sim P_2 \} \) holds.

**Proof.** Since \( d = 2 \), each partition has exactly two activation boundaries. Now recall the positive homogeneity of partitions from Proposition 3.1. Define an equivalent relation in the parameter space, defined by \( w_1 \sim w_2 \) if and only if \( w_1 = \alpha w_2 \) for some \( \alpha > 0 \). Now omit the origin point from \( \mathcal{W} = \mathbb{R}^2 \), and take quotient by this equivalent relation. Then we get
\[
(\mathbb{R}^2 - \{0\})/\sim \cong S^1.
\]

Therefore, we can impose an (cyclic) ‘order’ for partitions. For example, Figure 12 shows 8 partitions ordered by clockwise direction.

Now consider any two partitions in 2nd quadrant. since we can impose an order from 3rd quadrant \( \rightarrow \) 2nd quadrant \( \rightarrow \) 1st quadrant by clockwise direction, one partition is obtained by activating (or deactivating) some data from another partition. Similar argument holds for any two partitions in 4th quadrant, which completes the proof. 

Still we need some lemmas to prove the global convergence. Now we refer the norm-increasing subset. Proposition C.4 changes to the following proposition.

**Lemma E.3 (Norm-increasing subset).** Consider a gradient flow (4) under \( A1 \) and \( A2 \). Define \( q(w) := -\frac{d}{2} \|w\|^2 = w^T (H(w)w - q(w)) \). Then, for a gradient flow \( w(t) \),
1. \( g(w) < 0 \) if and only if \( \|w(t)\| \) increases.
2. \( g(w) \) is continuous.
3. Any local minimum \( w^* \) is on the boundary of the norm-increasing subset, i.e. \( g(w^*) = 0 \).

**Proof.**
1. Since \( g(w) = -\frac{d}{dt} \frac{1}{2} \|w\|^2 \), \( \|w\| \) increases if and only if \( g(w) < 0 \).
2. Now we show that the function \( g(w) \) is continuous. Indeed, it is continuous on each linearly partitioned region from the definition. Now consider a boundary of partitions and let \( x_0 \) be the data which determines the boundary. Let \( P \) and \( \tilde{P} \) be the two partitions such that \( \{x \in X \mid x \sim P\} = \{x \in X \mid x \sim \tilde{P}\} \cup \{x_0\} \). See Figure 11. Now distinguish \( g(w) \) in each subset by

\[
\begin{align*}
g_P(w) &= w^T \left( \sum_{x_i \sim P} x_i x_i^T \right) w - \sum_{x_i \sim P} y_i x_i^T w, \\
g_{\tilde{P}}(w) &= w^T \left( \sum_{x_i \sim P} x_i x_i^T \right) w - \sum_{x_i \sim P} y_i x_i^T w \\
&= w^T \left( \sum_{x_i \sim P} x_i x_i^T \right) w - w^T x_0 x_0^T w \\
&\quad - \sum_{x_i \sim \tilde{P}} y_i x_i^T w + y_0 x_0^T w. \\
&= g_P(w) - w^T x_0 x_0^T w + y_0 x_0^T w.
\end{align*}
\]

Then, for \( w \) on the activation boundary of \( x_0 \), since \( x_0^T w = 0 \), we get \( g_P(w) = g_{\tilde{P}}(w) \). Therefore, \( g(w) \) is continuous in \( \mathcal{W} \).
3. Let \( w^* \) be a local minimum. Then \( w^Tw = 0 \) at \( w^* \), since it is a stationary point. Thus \( g(w^*) = 0 \).

Now we define **all-activated partition** as the partition which activates all data. The following lemma guarantees the global convergence, if all-activated region contains a local minimum.

**Lemma E.4 (All-activated partition).** Consider a single-neuron ReLU network under \( \textbf{A1}, \textbf{A2}, \textbf{A3} \). Suppose there is a stationary point in all-activated region. Then a gradient flow initialized with infinitesimally small norm converges to the global minimum, without any deactivation.

**Proof.** For any data \( x_i \), define \( h_i(t) := w(t)^T x_i \). For a gradient flow initialized with infinitesimally small norm, it is enough to show that \( h_i(t) \geq 0 \) for all \( t \geq 0 \). Let \( w^* \) be a stationary point with \( w^*x_i > 0 \). For the Hessian matrix

\[
H := \sum_{i=1}^n x_i x_i^T, \text{ let } \{(\lambda_k, e_k)\}_{k=1}^2 \text{ be its eigenvalues and eigenvectors with } \lambda_1 > \lambda_2 > 0. \text{ Considering the hyperrectangle of } H \text{ (C.2), we know } c_k^* := w^*e_k > 0. \text{ Then from (17), we get}
\]

\[
\begin{align*}
w(t) &= (1 - e^{-\lambda_1 t})e_1 e_1^T w^* + (1 - e^{-\lambda_2 t})e_2 e_2^T w^* \\
&= (1 - e^{-\lambda_1 t})c_1^* e_1 + (1 - e^{-\lambda_2 t})c_2^* e_2.
\end{align*}
\]

Now, fix a data \( x_i \) and define \( c_k := x_i^T e_k \). Then,

\[
h_i(t) = ((1 - e^{-\lambda_1 t})c_1^* e_1 + (1 - e^{-\lambda_2 t})c_2^* e_2)^T x_i
\]

\[
= (1 - e^{-\lambda_1 t})c_1^* + (1 - e^{-\lambda_2 t})c_2^*. \quad (19)
\]

From the eigenvector decomposition, from \( \textbf{A1} \), we know \( c_1 > 0 \). However, we are not sure for the sign of \( c_2 \). Thus we consider both two cases, separately. Note that \( c_1^2 + c_2^2 = x_i^T \sum_{k=1}^2 e_k e_k^T w^* = x_i^T w^* > 0 \).

If \( c_2 \geq 0 \), then all terms in (19) are nonnegative, thus \( h_i(t) \geq 0 \) for all \( t \geq 0 \). Otherwise, if \( c_2 < 0 \), deforms (19) to

\[
h_i(t) = c_1 c_1^* + c_2 c_2^* - e^{-\lambda_1 t}c_1 c_1^* - e^{-\lambda_2 t}c_2 c_2^*
\]

\[
= c_1 c_1^* + c_2 c_2^* - e^{-\lambda_1 t}c_1 c_1^* - e^{-\lambda_2 t}c_2 c_2^*
\]

\[
\quad + e^{-\lambda_1 t}c_2 c_2^* - e^{-\lambda_2 t}c_2 c_2^*
\]

\[
= (c_1 c_1^* + c_2 c_2^*)(1 - e^{-\lambda_1 t}) + c_2 c_2^*(e^{-\lambda_1 t} - e^{-\lambda_2 t})
\]

\[
= x_i^T w^*(1 - e^{-\lambda_1 t}) + c_2 c_2^*(e^{-\lambda_1 t} - e^{-\lambda_2 t}).
\]

Then all terms are nonnegative, since \( c_2(e^{-\lambda_1 t} - e^{-\lambda_2 t}) > 0 \). Therefore, \( c_2 \geq 0 \) or not, we get \( h_i(t) \geq 0 \) for all \( t \geq 0 \). Since \( x_i \) could be any data, we proved that no data is deactivated on the gradient flow initialized with infinitesimally small norm. Finally, by Theorem 3.7, we conclude that the convergent point is the global minimum. \( \square \)

Now we show norm-increasing property for \( d = 2 \) case, without further conditions in the theorem.

**Lemma E.5 (Norm increasing for \( d = 2 \)).** Consider a single-neuron ReLU network under \( \textbf{A1}, \textbf{A2}, \textbf{A3} \). Recall the function \( g(w) \) defined in Proposition E.3. Then for a gradient flow initialized with infinitesimally small norm, \( g(w) < 0 \) until \( w(t) \) converges. Let \( P^* \) be the partition that \( w(t) \) converges in. Then, \( P^* \) is the first partition that the gradient flow ever meets, such that it contains a stationary point.

**Proof.** By Proposition E.3, it is enough to show that \( g(w(t)) < 0 \) for all \( t \geq 0 \). Since \( g(w) \) is continuous on boundary of partitions, we only need to consider interior part of each partition. Now we prove 1. \( g(w) < 0 \) and 2. convergence in the first partition that contains a stationary point, by induction.

First, \( g(w) < 0 \) at the initialization, since the gradient flow initialized with infinitesimally small norm. For the
second statement, by the proof of Proposition 4.4, we know that a gradient flow initialized with infinitesimally small norm activates all data at the beginning. If the all-activated partition contains a stationary point, then the gradient flow converges to this minimum without any deactivation by Lemma E.4.

Now we suppose $g(w) < 0$ in a partition $\tilde{P}$, and considering the next partition $P$. Let $x_0$ be the data which determines the common boundary of $\tilde{P}$ and $P$. See Figure 13. Let $w_0$ be the meeting point of gradient flow and the activation boundary of $x_0$. By Proposition E.3, we know that $g(w_0) < 0$. Let $e_1$ and $e_2$ be the eigenvectors of $H(w)$, such that $\lambda_1 > \lambda_2$. Note that a line through $w_0^*$ with direction $e_2$ is the asymptotic line (Zill, 2020). Now we consider two cases.

1. $w = w_0^*$ is in $P$.
   See Figure 13 (a). Since $w_0^*$ is in $P$ and $g(w_0^*) = 0$ by Proposition E.3, gradient flow converges to $w_0^*$ keeping $g(w) < 0$. Therefore, $g(w)$ is negative until it converges and the gradient flow converges in the first partition that contains its local minimum inside.

2. $w = w_0^*$ is not in $P$.
   By the induction hypothesis, the gradient flow has never visited a partition that contains its minimum itself. Therefore, $w_0^*$ is under the partition $P$. See Figure 13 (b). By the similar argument above, the gradient flow escape $P$ keeping $g(w) < 0$ since $\lambda_2 < \lambda_1$. Therefore, in $P$, $g(w)$ is negative until the gradient flow escapes $P$.

By the induction hypothesis, this completes the proof. □

**Proof of Lemma 4.6.** Consider a deactivation of the gradient flow. Suppose the gradient flow $w(t)$ enters to a partition $P$ by deactivates $x_0$ at $w_0$. Let $u := \frac{w_0}{\|w_0\|}$, $H := \sum_{x_i \in P} x_i x_i^T$, and $q := \sum y_i x_i$. Now for scalar $\alpha > 0$, define

\[
\zeta(\alpha) := -x_0^T \nabla L(\alpha u) = x_0^T (-\alpha H u + q).
\]

$\zeta(\alpha)$ tells what values of $\alpha$ deactivates or activates $x_0$. Indeed, $x_0$ is deactivated if and only if

\[
\zeta(\alpha) = -\alpha x_0^T H u + x_0^T q < 0,
\]

which is equivalent to

\[
\alpha > \frac{x_0^T q}{x_0^T H u}.
\]

Defining $\alpha^* := \frac{x_0^T q}{x_0^T H u}$, we conclude that $x_0$ is deactivated if and only if $\alpha > \alpha^*$. See Figure 13 (a).

However, by Lemma E.5, we know that norm of the gradient flow $\|w\|$ strictly increases. Therefore, the norm has been greater than the point where $x_0$ was deactivated. It means, there is no re-activation of $x_0$. In other words, a gradient flow initialized with infinitesimally small norm never revisit a partition that already been traversed. □

**Proof of Theorem 4.7.** By No revisit lemma (Lemma 4.6), there is no re-activation. In other words, a gradient flow initialized with infinitesimally small norm only deactivates data until it converges. Moreover, by Lemma E.5, it converges to a minimum in the first partition such that it contains. Therefore, it has maximum number of support vectors, which is the global minimum by Theorem 3.7. □
However, we also prove that this global convergence does not hold in high dimension \((d > 2)\) in general. The key point is that No revisit lemma (Lemma 4.6) does not hold for \(d > 2\). The following lemma and theorem show that for single-neuron linear networks with \(d > 2\), a gradient flow may reactivate at most \(d - 1\) times.

**Lemma E.6.** For some constants \(a_i, b_i, \text{ and } c\), the equation \(\sum_{i=1}^{n} a_i e^{b_i t} = c\) has at most \(n\) zeros.

**Proof:** We prove a weak version of this Lemma first, which is \(c = 0\) case.

**Claim:** Equation \(\sum_{i=1}^{n} a_i e^{b_i t} = 0\) has at most \(n - 1\) solutions.

**Proof of Claim:** We use mathematical induction on \(n\). For \(n = 1\), equation \(a_1 e^{b_1 t} = 0\) has no solution and for \(n = 2\), it has at most one solution \(t = \frac{1}{b_1 - b_2} \log \frac{-a_2}{a_1}\) if \(-\frac{a_2}{a_1} > 0\). Now suppose it holds for \(n\) and consider \(n + 1\) step. WLOG assume \(a_{n+1} \neq 0\) and deform the equation \(\sum_{i=1}^{n+1} a_i e^{b_i t} = 0\) to

\[
\sum_{i=1}^{n+1} a_i e^{b_i t} = a_{n+1} e^{b_{n+1} t} + \sum_{i=1}^{n} \frac{a_i}{a_{n+1}} e^{(b_i - b_{n+1}) t} = 0.
\]

(20)

Now let \(g(t) = 1 + \sum_{i=1}^{n} \frac{a_i}{a_{n+1}} e^{(b_i - b_{n+1}) t}\). Then,

\[
g'(t) = \sum_{i=1}^{n} \frac{a_i(b_i - b_{n+1})}{a_{n+1}} e^{(b_i - b_{n+1}) t} = 0
\]

has at most \(n - 1\) solutions, by the induction. Therefore, (20) has at most \(n\) solutions, by Rolle’s Theorem. This completes the proof of the claim.

Now we return to prove the lemma. For given \(\sum_{i=1}^{n} a_i e^{b_i t} = c\), it has a solution if and only if \(\sum_{i=1}^{n} a_i e^{b_i t} = c\) has at most \(n + 1\) zeros by the claim.

Finally, we provide one example that \(\sum_{i=1}^{n} a_i e^{b_i t} = c\) has exactly \(n\) zeros. Take \(c = 1\), then there exists \(a_i\’s\) such that \(t = 0, 1, 2, \cdots, n - 1\) are zeros for some distinct \(b_i\’s\) because

\[
\begin{pmatrix}
1 & 1 & \cdots & 1 \\
e^{b_1} & e^{b_2} & \cdots & e^{b_n} \\
\vdots & \vdots & \ddots & \vdots \\
e^{(n-1)b_1} & e^{(n-1)b_2} & \cdots & e^{(n-1)b_n}
\end{pmatrix}
\begin{pmatrix}
1 \\
a_2 \\
\vdots \\
a_n
\end{pmatrix}
= \begin{pmatrix}
1 \\
0 \\
\vdots \\
0
\end{pmatrix}
\]

above matrix is a Vandermonde matrix, which is known to be invertible (Anton & Busby, 2003).

**Theorem E.7** (Reactivation of linear networks for \(d > 2\)). Consider single-neuron linear networks under A1, A2, and A3. There exists dataset \(\{(x_i, y_i)\}_{i=1}^{n}\) such that for a hyperplane \(V\), a gradient flow crosses \(V\) at most \(d\) times.

**Proof.** Recall (17). By translation and rotation, WLOG we can assume \(w^*\) to be origin and \(H = \text{diag}(e^{\lambda_1}, \cdots, e^{\lambda_d})\). Therefore, the gradient flow is given by

\[
w(t) = \text{diag}(e^{\lambda_1 t}, \cdots, e^{\lambda_d t}) w_0 = \begin{bmatrix}
e^{\lambda_1 t} w_{01} \\
e^{\lambda_2 t} w_{02} \\
\vdots \\
e^{\lambda_d t} w_{0d}
\end{bmatrix}
\]

Now suppose that we have a hyperplane \(V\) determined by \(\{w \mid w^T v = c\}\). Then the number of intersection points of the gradient flow and this hyperplane is given by the number of zeros of equation

\[
w(t)^T v - c = \sum_{k=1}^{d} e^{\lambda_k t} w_{0k} v_i - c = 0
\]

(21)

Finally, by Lemma E.6, equation (21) has at most \(d\) zeros. Therefore, a gradient flow of single-neuron linear networks can across the hyperplane \(V\) at most \(d\) times.

We provide Example 5.3 in Section 5 which shows an reactivation in a single-neuron ReLU network, for \(d = 3\) case.

**F. Detail Settings for the Experiments.**

We use PyTorch library with GPU Gigabyte GeForce GTX 1080 Ti to implement all experiments. The gradient flow is implemented by a gradient descent with small learning rate.

**F.1. Detail Settings for Example 5.1**

This toy example is a single-neuron ReLU network with \(n = 5\) and \(d = 2\) setting. The training dataset used in this example is

\[
\begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5
\end{pmatrix}
= \begin{bmatrix}
0.8858 & 0.4338 & 0.6739 & 0.0221 & 0.2322 \\
0.0244 & 0.8852 & 0.0399 & 0.4778 & 0.8717
\end{bmatrix},
\]

\[
\begin{pmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4 \\
y_5
\end{pmatrix}
= \begin{bmatrix}
0.6111 & 0.9397 & 1.8694 & 2.7104 & 1.3089
\end{bmatrix}.
\]

Three blue gradient flows in Figure 4 are initialized at

\[
w_0 = \begin{bmatrix}
0.0001 \\
0.0001 \\
0 \\
8
\end{bmatrix}, \text{ and } \begin{bmatrix}
0 \\
45
\end{bmatrix}.
\]

Two red gradient flows are initialized at
\[ w_0 = \begin{bmatrix} -0.1 \\ 0.1 \end{bmatrix}, \text{ and } \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix}. \]

Learning rate is set to 0.05, and total number of iterations is set to 200K.

### F.2. Detail Settings for Example 5.2

This toy example considers single-neuron linear and ReLU networks with \( n = d = 3 \). The training dataset used in this example is

\[
[x_1 \ x_2 \ x_3] = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & 0 \\ 2 & 0 & 0 \end{bmatrix},
\]

\[
[y_1 \ y_2 \ y_3] = [0.05 \ 6 \ 0.5].
\]

As it is mentioned in the main paper, we define \( h_i(w) := w^T x_i \) for each data \( x_i \). We used the initial point sampled from

\[ w_0 = 0.0001 \times U([0,1]^3), \]

where \( U([0,1]) \) means uniform distribution in \([0,1]\). Both linear and ReLU networks share the initial point, and learning rate and the total number of iterations are set to 0.005 and 10,000, respectively.

The value of \( h_i(w) \) functions are plotted by graphs in Figure 14. Note that \( x_1 \) is deactivated on the gradient flow of ReLU network \( (h_1(w) < 0) \) at the beginning, and activated after. Since it converges in the all-activated partition, gradient flows of linear and ReLU networks converge to the same point, which is the global minimum. See Figure 14(b).

### F.3. Detail Settings for Example 5.3

This toy example considers single-neuron linear and ReLU networks with \( n = 4 \) and \( d = 3 \). The training dataset used in this example is

\[
[x_1 \ x_2 \ x_3 \ x_4] = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 1 & 2 & 0 \end{bmatrix},
\]

\[
[y_1 \ y_2 \ y_3 \ y_4] = [0.1 \ 0.2 \ 4 \ 0.1].
\]

As it is mentioned in the main paper, we define \( h_i(w) := w^T x_i \) for each data \( x_i \). We used the initial point sampled from

\[ w_0 = 0.0001 \times U([0,1]^3), \]

and both linear and ReLU networks share the initial point. For both networks, learning rate is set to 0.005 and the total number of iterations is set to 20K.

The value of \( h_i(w) \) functions are plotted as graphs in Figure 15. Note that \( x_4 \) is deactivated on the gradient flow of ReLU network \( (h_4(w) < 0) \) at the beginning, and activated after. Since it converges in the all-activated partition, gradient flows of linear and ReLU networks converge to the same point, which is the global minimum. See Figure 15(b).
Figure 14. Deactivation in ReLU network during training single-neuron linear and ReLU networks on the same dataset. In (a), gradient flows of linear and ReLU networks are denoted by blue and orange curves. In (b), training loss curves of two gradient flows are plotted. In (c), (d) and (e), the graph of $h_i(w) = \mathbf{w}_i^T \mathbf{x}_i$ values for linear and ReLU networks are plotted. The solution line $h_i(w) = y_i$ is drawn by red dashed line. Observe that $x_1$ is deactivated at Epochs $\approx 500$, and the ReLU network never reactivate it ($h_1(w_r^*) < 0$) while the linear network reactivates it again ($h_1(w_l^*) > 0$).
Figure 15. Reactivation in ReLU networks. In (a), gradient flows of linear and ReLU networks are denoted by blue and orange curves. In (b), training loss curves of two gradient flows are plotted. In (c), (d), (e), and (f), the graph of $h_i(w) = w^T x_i$ values for linear and ReLU networks are plotted. The solution line $h_i(w) = y_i$ is drawn by red dashed line. Observe that $x_4$ is deactivated immediately after initialization, but the ReLU network reactivates it at Epochs $\approx 8500$. 