On the Inductive Bias of Neural Networks for Learning Read-once DNFs

Abstract

Learning functions over Boolean variables is a fundamental problem in machine learning. But not much is known about learning such functions using neural networks. Here we focus on learning read-once DNFs under the uniform distribution with a convex neural network and gradient methods. We first observe empirically that gradient methods converge to compact solutions with neurons that are aligned with the terms of the DNF. This is despite the fact that there are many zero training error networks that do not have this property. Thus, the learning process has a clear inductive bias towards such logical formulas. Following recent results which connect the inductive bias of gradient flow (GF) to KKT points of minimum norm problems, we study these KKT points in our setting. We prove that zero training error solutions that memorize training points are not KKT points and therefore GF cannot converge to them. On the other hand, we prove that globally optimal KKT points correspond exactly to networks that are aligned with the DNF terms. These results suggest a strong connection between the inductive bias of GF and solutions that align with the DNF. We conclude with extensive experiments which verify our findings.

1 INTRODUCTION

The training objective of overparameterized neural networks is non-convex and contains multiple global minima with different generalization properties. Therefore, just minimizing the training objective does not guarantee good generalization performance. Nonetheless, neural networks trained in practice with gradient-based methods show good test performance across numerous tasks [Krizhevsky et al., 2012, Silver et al., 2016], suggesting an *inductive bias* towards desirable solutions. Understanding this inductive bias and how it depends on the algorithm, architecture and data is one of the major open problems in machine learning [Zhang et al., 2017, Neyshabur et al., 2018].

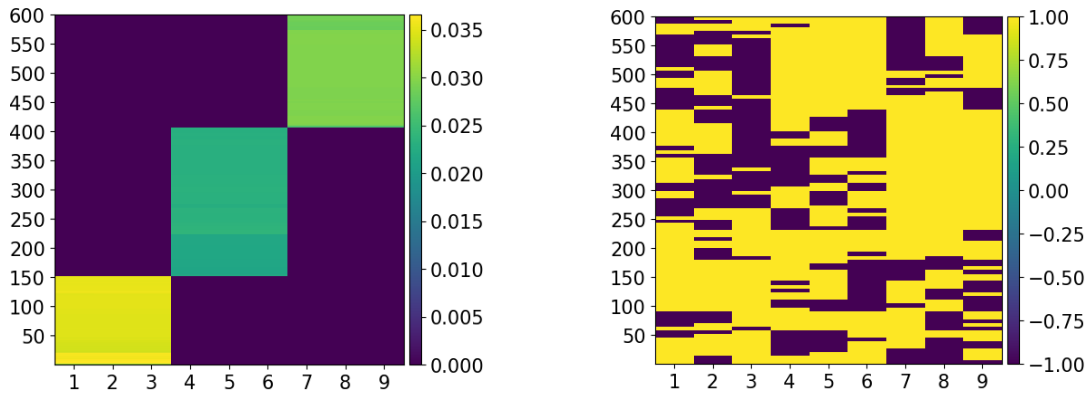
In recent years, there have been major efforts to tackle this challenge. One line of works considers the Neural Tangent Kernel (NTK) approximation of neural networks which reduces to a convex optimization problem [Jacot et al., 2018]. However, it has been shown that the NTK approximation is limited and does not accurately model neural networks as they are used in practice [Yehudai and Shamir, 2019, Daniely and Malach, 2020]. Other works tackle the non-convexity directly, usually in very simplified settings [e.g., diagonal linear networks Woodworth et al., 2019] or for special cases such as regression with 2-layer models and Gaussian distributions [Li et al., 2020] or infinitely wide two-layer networks [Chizat and Bach, 2020].

Here we focus on the important problem of learning Boolean functions with neural networks. While much is known about this problem from a computational and statistical perspective, little is understood on how they can be learned with neural networks, and in particular on the inductive bias of gradient descent in this case. In computational learning theory, the problem of learning DNFs has a long history. Learning DNFs is hard [Pitt and Valiant, 1988] and the best known algorithms for learning DNFs under the uniform distribution run in quasi-polynomial time [Verbeurgt, 1990]. On the other hand, for learning *read-once* DNFs under the uniform distribution there exist efficient learning algorithms [Mansour and Schain, 2001].[1] Therefore, it is interesting to understand whether neural networks can learn read-once DNFs under the uniform distribution and this motivates the study of the inductive bias in this case.

We focus specifically on a simple neural architecture with one-hidden layer and ReLU activations, and output weights

[1] In a read-once DNF each literal appears at most once. See Section 3 for a formal definition.

Submitted to the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022). **To be used for reviewing only**.



(a) (b)

Figure 1: The weight vectors for two convex networks that perfectly fit 250 training samples labeled with the read-once DNF: $(x_1 \wedge x_2 \wedge x_3) \vee (x_4 \wedge x_5 \wedge x_6) \vee (x_7 \wedge x_8 \wedge x_9)$. (a) A network trained with SGD using small Gaussian initialization. The weights can be seen to be well aligned with the DNF terms, and the test accuracy is 100%. (b) A network whose weight vectors are equal to the positive samples in the training set. This network "memorizes" the training data and fits it perfectly. However, the test error is 81.6%.

equal to one. We refer to it as a convex network, because the network output is a convex function of its inputs in this case. It is easy to see that this architecture is sufficiently expressive for learning DNFs. We show that for learning read-once DNFs there exist solutions that perfectly classify the training set with significantly different properties. Specifically, solutions which memorize the training points in their neurons, and other solutions whose neurons align exactly with the terms of the DNF, which we call DNF recovery solutions. Figure 1a-b shows an example of these solutions. Our first empirical finding is that SGD with small initialization converges to a DNF-recovery solution. This indicates a strong inductive bias of gradient methods towards simple logical forms in this case. We further observe that this bias allows the convex network to generalize better than algorithms designed specifically for learning read-once DNFs. Together these empirical observations establish neural nets as an attractive approach to learning read-once DNFs.

Given the above, we ask what can explain this inductive bias of gradient methods, and what theoretical guarantees can be obtained for its performance. We turn to recent line of works [Soudry et al., 2018, Ji and Telgarsky, 2019a, Lyu and Li, 2020, Ji and Telgarsky, 2020, Chizat and Bach, 2020] which study the inductive bias of gradient flow (GF) in several settings. Their results suggest that the inductive bias of GF is to KKT points or global solutions of minimum norm problems (or analogously maximum margin problems). Motivated by these results, we prove that any norm minimizing solution in our setting is a DNF Recovery solution, strongly suggesting why it is that GF converges to it. We further strengthen this result by proving that memorizing solutions (namely, solutions where there are neurons that are only activated by specific inputs, as in Figure 1b) are not KKT points of the min-norm problem. Therefore, GF will not converge to these.

We corroborate our findings with empirical results which show that our conclusions hold more broadly. Specifically, we perform experiments on DNFs of higher dimension and standard one-hidden layer neural networks. Taken together, our results demonstrate that gradient methods can recover simple descriptions of Boolean functions from data, which results in good generalization performance and may have important implications on the question of interpretability.

2 RELATED WORK

Recently, several works studied the inductive bias of neural networks and showed connections between gradient methods and margin maximization [Chizat and Bach, 2020, Lyu and Li, 2020, Nacson et al., 2019, Ji and Telgarsky, 2020]. These works motivate our theoretical analysis of minimum norm solutions. In our theoretical analysis, we apply the results of Lyu and Li [2020], Ji and Telgarsky [2020], which show that GF is biased towards KKT points of min-norm problems.

Other works study fully connected neural networks under certain assumptions on the data such as linearly separable data [Brutzkus et al., 2018, Sarussi et al., 2021, Frei et al., 2021] or Gaussian data [Safran and Shamir, 2018, Du et al., 2019]. Malach and Shalev-Shwartz [2020] show that certain structured Boolean circuits can be learned with a network architecture that is specialized for their data structure. Fully connected networks were also analyzed via the NTK approximation [Jacot et al., 2018, Du et al., 2019, 2018, Arora et al., 2019, Ji and Telgarsky, 2019b, Cao and Gu, 2019,

Jacot et al., 2018, Fiat et al., 2019, Allen-Zhu et al., 2019, Li and Liang, 2018, Daniely et al., 2016]. However, Yehudai and Shamir [2019], Daniely and Malach [2020] have highlighted limitations of the NTK framework, suggesting that it does not accurately model neural networks as they are used in practice.

Another line of works [Saad and Solla, 1996, Goldt et al., 2019, Tian, 2019] studies neural networks in student-teacher regression settings and shows a "specialization" effect, where a subset of student neurons aligns with teacher neurons. The main difference from our setting is that we consider classification on binary data, and they consider regression tasks on non-discrete data (e.g., Gaussian). We note that classification settings present unique theoretical challenges for studying inductive bias of gradient methods [Montanari et al., 2019].

In a recent result, Phuong and Lampert [2021] provide an end-to-end optimization analysis of a two-layer ReLU network on orthogonally separable data (which is a very simplified setup of linearly separable data). They consider the cross-entropy loss and their analysis implies that neurons specialize to certain directions. We focus on a significantly more challenging setting, where the data is labeled by a *nonlinear* read-once DNF.

An inductive bias towards specializing solutions has also been observed in Brutzkus and Globerson [2019] and proved for a simple setup with nonlinear data and a convolutional neural network. The notion of specialization is also related to the notion of "collapse" [Papyan et al., 2020]. We note that in our setting we do not observe the collapse phenomenon since the hidden-layer representations of the positive samples are not all in one small cluster (e.g., see Figure 1a).[2]

Rudin [2019] argues that methods for explaining large neural networks should be avoided because networks are too complex for humans to understand. However, we show, albeit in a restricted setting, that learned networks can be rather simple, and are easily mapped to the underlying DNF.

3 PROBLEM FORMULATION

DNFs and Read-Once DNFs: In what follows, we use $[n]$ to denote the set $\{1, 2, ..., n\}$. Let $\mathcal{X} = \{\pm 1\}^D$ and $\mathcal{Y} = \{\pm 1\}$. DNFs [e.g., see O'Donnell, 2014] are usually defined on inputs with entries in $\{0, 1\}$ to an output in $\{0, 1\}$. In this work, we consider DNFs on inputs with entries in $\{\pm 1\}$ and output in $\{\pm 1\}$.

We will use the following notation for DNFs. A DNF with K terms will be defined via K indicator vectors $\boldsymbol{t}_1^*, ..., \boldsymbol{t}_K^* \in \{0, 1\}^D$. We refer to each $\boldsymbol{t}_n^*$ as a *term* and define its set of active indices by $\mathbb{A}_n = \left\{j \in [D] \mid t_{nj}^* = 1\right\}$, where t_{nj}^* is the jth entry of $\boldsymbol{t}_n^*$. The corresponding DNF will be given by the function $f^* : \mathcal{X} \to \mathcal{Y}$ as follows: $f^*(\boldsymbol{x}) = 1$ if $\exists n \in [K]\ s.t.\ \boldsymbol{x} \cdot \boldsymbol{t}_n^* = |\mathbb{A}_n|$, and otherwise $f^*(\boldsymbol{x}) = -1$. Notice that f^* is monotone.

We say that a sample $\boldsymbol{x} \in \mathcal{X}$ satisfies the term $\boldsymbol{t}_n^*$ if $\boldsymbol{x} \cdot \boldsymbol{t}_n^* = |\mathbb{A}_n|$. We refer to $|\mathbb{A}_n|$ as the size of the term $\boldsymbol{t}_n^*$. To compare our notation with the standard one, for example, the DNF $(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$ with 4 inputs has terms $\boldsymbol{t}_1^* = (1, 1, 0, 0)$ and $\boldsymbol{t}_2^* = (0, 0, 1, 1)$. We will use the standard notation when convenient (e.g., as in Figure 1). In this work we will focus on ***read-once*** DNFs where for all $i \neq j \in [K]$, $\mathbb{A}_i \cap \mathbb{A}_j = \emptyset$ and the sizes of all the terms are greater than 1.

Learning Setup: Let $\mathcal{D}$ be a distribution on $\mathcal{X} \times \mathcal{Y}$. We assume that for $(\mathbf{x}, y) \sim \mathcal{D}$, $\mathbf{x} \sim \prod_{i=1}^n \text{Bernoulli}(0.5)$ and $y = f^*(\boldsymbol{x})$, where f^* is a monotone read-once DNF.[3]

We consider learning f^* given a training set $\mathbb{S} \subseteq \mathcal{X} \times \mathcal{Y}$, where for each $(\boldsymbol{x}, y) \in \mathbb{S}$, $\boldsymbol{x}$ is sampled IID from $\mathcal{D}$ and $y = f^*(\boldsymbol{x})$. Denote $\mathbb{S}_x = \{\boldsymbol{x} \mid (\boldsymbol{x}, y) \in \mathbb{S}\}$, the positive samples by $\mathbb{S}_+ = \{\boldsymbol{x} \mid (\boldsymbol{x}, 1) \in \mathbb{S}\}$, the negative samples by $\mathbb{S}_- = \{\boldsymbol{x} \mid (\boldsymbol{x}, -1) \in \mathbb{S}\}$ and the number of samples by $m = |\mathbb{S}|$. In some cases we will consider the population case where $\mathbb{S}_x = \mathcal{X}$.

Neural Architecture: We consider a **convex** one-hidden layer neural network (NN) with r hidden units and parameters $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b}, c) \in \mathbb{R}^{rD} \times \mathbb{R}^r \times \mathbb{R}$ which is defined by:

$$N(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{i \in [r]} \sigma(\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i) + c \tag{1}$$

where $\sigma(x) = max\{0, x\}$ is the ReLU function, $\boldsymbol{w}_i$ is the ith row of $\boldsymbol{W}$ and b_i is the ith entry of $\boldsymbol{b}$. We also use a scalar trainable bias $c \in \mathbb{R}$ in the second layer to allow for negative outputs.

The resulting network is positive homogeneous, and thus recent results on such networks can be applied [Lyu and Li, 2020, Ji and Telgarsky, 2020]. Note that the network is convex because it is a sum of convex ReLU functions Amos et al. [2017].

Loss Minimization: To learn f^* we consider minimizing the following loss:

$$L(\boldsymbol{\theta}) = \frac{1}{m} \sum_{(\boldsymbol{x}, y) \in \mathbb{S}} \ell\left(y N(\boldsymbol{x}; \boldsymbol{\theta})\right) \tag{2}$$

where $\ell(z) = \log\left(1 + e^{-z}\right)$ is the binary cross entropy loss. We note that $L(\boldsymbol{\theta})$ is generally non-convex (even though

[2] For the model in the figure, the representations of the positive points create 8 different clusters which correspond to each possible combination of the three terms.

[3] In the case of the uniform distribution, we can assume monotone DNFs WLOG. This follows since for the uniform distribution, by symmetry, any negated literal can be replaced with the original literal (without negation) and all our results still hold.

the network N is convex). For our theoretical analyses we consider Gradient Flow (GF). We denote the initialization of GF by $\boldsymbol{\theta}^{(0)} = \left(\boldsymbol{W}^{(0)}, \boldsymbol{b}^{(0)}, c^{(0)}\right)$ and the weights at iteration t by $\boldsymbol{\theta}^{(t)} = \left(\boldsymbol{W}^{(t)}, \boldsymbol{b}^{(t)}, c^{(t)}\right)$. If the iteration index is clear from context we omit it and use $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b}, c)$.

Recall that gradient flow is the infinitesimal step limit of gradient descent where $\boldsymbol{\theta}^{(t)}$ changes continuously in time and satisfies the differential inclusion $\frac{d\boldsymbol{\theta}^{(t)}}{dt} \in -\partial^\circ L(\boldsymbol{\theta}^{(t)})$ for a.e. t. Here $\partial^\circ L\left(\boldsymbol{\theta}^{(t)}\right)$ is the Clarke's sub-differential which is a generalization of the differential for non-differentiable functions:

$$\partial^\circ f(\boldsymbol{x}) = \text{conv}\Big\{ \lim_{k\to\infty} \nabla f(\boldsymbol{x}_k) \mid \boldsymbol{x}_k \to \boldsymbol{x} \text{ and } f \text{ is differentiable at } \boldsymbol{x}_k \Big\} \tag{3}$$

The differential inclusion allows to take any vector in $-\partial^\circ L(\boldsymbol{\theta}^{(t)})$ in each step of gradient flow. In our case, the differential inclusion has multiple possible values when the ReLU is 0 since ReLU has multiple sub-gradients at 0. For our theoretical results, we will assume that the subgradient of ReLU at 0 is determined in advance to a value $a \in [0, 1]$. This value of the subgradient is used for all neurons and in all iterations. Usually a is set to be either 0 or 1. This assumption corresponds to the common way gradient descent runs in practice. We provide a formal definition of this assumption in the supplementary.

Next, we define solutions which perfectly classify the training set. We will consider solutions with a margin constraint, since this will be convenient when we discuss minimum norm solutions.

Definition 3.1. *We say that a solution $\boldsymbol{\theta}$ is perfect if for all $(\boldsymbol{x}, y) \in \mathbb{S}$, $yN(\boldsymbol{x}; \boldsymbol{\theta}) \geq 1$.*

Norm Minimization: Multiple recent works have highlighted interesting connections between gradient methods and norm minimization or margin maximization [Lyu and Li, 2020, Neyshabur et al., 2018, Nacson et al., 2019, Ji and Telgarsky, 2020, Chizat and Bach, 2020]. The norm minimization problem is to minimize the norm of the model weights subject to the correct classification with a margin. Namely, the problem is:

$$\begin{array}{ll} \min & \sum_{i\in[r]} ||(\boldsymbol{w}_i, b_i)||_2^2 + c^2 \\ \text{s.t.} & yN(\boldsymbol{x}; \boldsymbol{\theta}) \geq 1 \ , \ \forall (\boldsymbol{x}, y) \in \mathbb{S} \end{array} \tag{4}$$

It was shown [Lyu and Li, 2020, Ji and Telgarsky, 2020] that gradient flow converges to a KKT point of the above problem under certain conditions.

3.1 EXPRESSIVE POWER

Here we show that the network in Eq. (1) has the expressive power to implement *any* Boolean function over $\mathcal{X}$. Therefore, in terms of expressive power, the network is suitable for learning Boolean functions and has the same expressive power for implementing Boolean functions as a standard one-hidden layer NN.

Theorem 3.1. *Let $f : \mathcal{X} \to \mathcal{Y}$. Then, there exists $\boldsymbol{\theta}$ and a network N in Eq. (1) with $r \leq 2^D$ neurons such that for all $\boldsymbol{x} \in \mathcal{X}$, $\text{sign}\left(N(\boldsymbol{x}; \boldsymbol{\theta})\right) = f\left(\boldsymbol{x}\right)$.*

Proof. Let $\mathcal{X}_+ = \{\boldsymbol{x} \mid f(x) = 1\}$. Define $r = |\mathcal{X}_+|$. Then, $\mathcal{X}_+ = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r\}$. Define $c = -1$ and for each $i \in [r]$ define $\boldsymbol{w}_i = \boldsymbol{x}_i$ and $b_i = -D + 2$. Then $\forall \boldsymbol{x}_i \in \mathcal{X}_+$ it holds that $\sigma(\boldsymbol{w}_i \cdot \boldsymbol{x}_i + b_i) = 2$ and $\forall \boldsymbol{x} \neq \boldsymbol{x}_i$ it holds that $\sigma(\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i) = 0$. Therefore $\forall \boldsymbol{x} \in \mathcal{X}_+$ we have $N(\boldsymbol{x}; \boldsymbol{\theta}) = 1$ and for $\boldsymbol{x} \in \mathcal{X} \backslash \mathcal{X}_+$ it holds that $N(\boldsymbol{x}; \boldsymbol{\theta}) = -1$, from which the claim follows. □

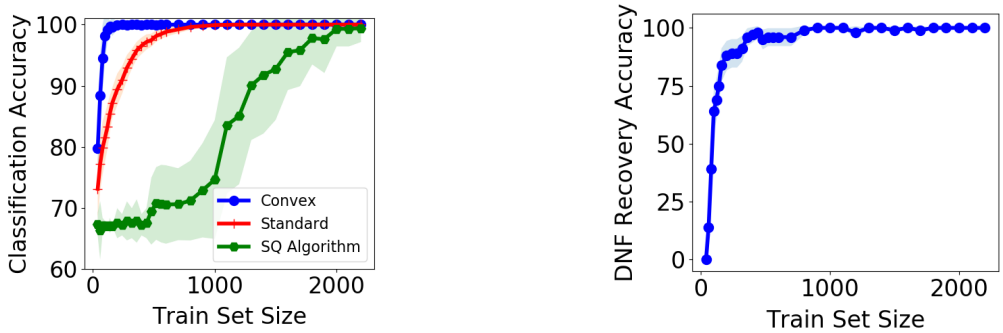
3.2 EMPIRICAL PERFORMANCE

Thus far we described the setting where the ground truth function is a read-once DNF that is learned by a convex neural net. We have seen in Theorem 3.1 that the convex network is sufficiently expressive. However, this does not imply that the network can learn read-once DNFs in practice. To examine this, we performed experiments for learning read-once DNFs under the uniform distribution with the convex network. We compared its test performance to a standard two-layer neural network, and an algorithm based on Statistical Queries (SQ) for learning read-once DNFs that has polynomial sample complexity guarantees [Mansour and Schain, 2001]. We note that the convex network was implemented with a relatively small initialization. In Section 7 and in the Supplementary we conduct experiments with a convex network with large initialization which is analogous to training in the NTK regime [Chizat et al., 2019].

Figure 2a shows the evaluation results. It can be seen that the convex network outperforms the other algorithms across all training set sizes. Therefore, together with Theorem 3.1, we conclude that the convex network we consider is a good test-bed for analyzing the inductive bias of neural networks in the setting of read-once DNFs and uniform distribution.

4 EMPIRICAL OBSERVATIONS ON THE INDUCTIVE BIAS

Overparameterized networks (i.e., large r), can fit the training data with multiple different solutions. Some have good generalization performance while others overfit. When training is performed with gradient methods, only certain solutions will be found and not others. In other words, gradient methods have an *inductive bias* towards certain solutions. Understanding this inductive bias is key for understanding the generalization performance of gradient methods in practice.



(a) (b)

Figure 2: Learning the read-once DNF: $(x_1 \wedge x_2 \wedge x_3) \vee (x_4 \wedge x_5 \wedge x_6) \vee (x_7 \wedge x_8 \wedge x_9)$ (a) Test accuracy for the following models: convex neural network, standard two layer neural network and an algorithm based on statistical queries (SQ). (b) Accuracy of the DNF recovery procedure for finding exactly the true DNF from the network weights.

Figure 1a shows an example of the weight vectors learned by an overparameterized convex network (Eq. (1)) optimized with SGD. It can be seen that although the network has many weight vectors, they form tight clusters around the terms of the true underlying DNF. This network has perfect accuracy on both the training set and the test set. We further devise a simple procedure to recover the DNF terms from the neurons. The procedure removes low norm neurons and rounds the weights, the exact details are provided in the supplementary. Figure 2b shows that the DNF recovery procedure can accurately find the DNF terms. In the supplementary we provide further details on this experiment and show many more examples of the inductive bias of SGD towards solutions which align with terms.

Another solution that minimizes the training error is shown in Figure 1b. In this solution, the network memorizes in its neurons the positive training points.[4] However, this network does not generalize well, and has 81% test accuracy. Thus, we see that SGD converges to the true DNF and not to a "memorization" solution, despite the latter also minimizing the training loss.

These observations raise the following intriguing questions: (1) Why do gradient methods have an inductive bias towards solutions that align with the terms of the DNF? (2) Why do gradient methods not converge to solutions that overfit and memorize training points in their neurons?

In the next sections we provide theoretical results which address these questions.

5 GRADIENT FLOW DOES NOT MEMORIZE

In this section, we prove that gradient flow (GF) does not converge to solutions that memorize training points. First we formally define a memorizing neuron and a memorization solution.

Definition 5.1. *A neuron $i \in [r]$ is a* memorizing neuron*, if there exists a sample $\hat{\boldsymbol{x}} \in \mathbb{S}_x$ such that:*

$$\boldsymbol{w}_i \cdot \hat{\boldsymbol{x}} + b_i > 0 \text{ and } \forall \boldsymbol{x} \in \mathcal{X} \backslash \{\hat{\boldsymbol{x}}\} \;\; \boldsymbol{w}_i \cdot \boldsymbol{x} + b_i \leq 0 \quad (5)$$

In this case, we say that neuron i memorizes *$\hat{\boldsymbol{x}}$.*

Thus, a neuron i memorizes $\hat{\boldsymbol{x}}$ if it is the only point in $\mathcal{X}$ that has a positive dot product with the neuron. Since the nonlinear activation of the network is ReLU, this implies that only the point $\hat{\boldsymbol{x}}$ activates neuron i.

Definition 5.2. *$\boldsymbol{\theta}$ is a memorization solution if $\boldsymbol{\theta}$ is perfect (Definition 3.1) and there exists $i \in [r]$ and a sample $\hat{\boldsymbol{x}} \in \mathbb{S}_x$ such that neuron i memorizes $\hat{\boldsymbol{x}}$.*

Note that the solution in Figure 1b is a memorization solution where all positive points in $\mathbb{S}_+$ are memorized. Definition 5.2 defines a broader set of solutions in which at least one point is memorized.

We now state two assumptions that are required for our main result. We apply recent results of Lyu and Li [2020], Ji and Telgarsky [2020], which assume that GF is in the late phase of training. Therefore, we will need the following assumption.

Assumption 5.1. *There exists t_0 s.t. $L\left(\boldsymbol{\theta}^{(t_0)}\right) < \frac{\ln 2}{n}$.*

The next theorem shows that GF cannot converge to memorization solutions.

Theorem 5.1. *Assume that Assumption 5.1 holds and $D > 2$, $K \geq 2$. Let $\boldsymbol{\theta}$ be a memorization solution. Then GF does not converge to $\boldsymbol{\theta}$.*

The proof ideas is as follows. Lyu and Li [2020] and Ji and Telgarsky [2020] show that under Assumption 5.1, GF

[4] See a formal definition of memorization in the next section.

converges to a KKT point of Eq. (4). KKT points must satisfy the KKT conditions: stationarity and complementary slackness (see supplementary for details). We use this fact together with the structure of the subgradient updates to show that memorization solutions cannot satisfy the KKT conditions. To show this, we first characterize the memorizing neuron using the following lemma:

Lemma 5.1. *For $D > 2$. Let $\boldsymbol{\theta}$ be a solution with a neuron $i \in [r]$ that memorizes a sample $\hat{\boldsymbol{x}} \in \mathbb{S}_x$, then $\boldsymbol{\theta}$ satisfies the following properties:*

1. $\hat{x}_j = \text{sign}(w_{ij})$ *for all* $1 \leq j \leq D$.
2. *For* $\boldsymbol{x} \in \mathcal{X}$ *if* $\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i = 0$ *then* $\boldsymbol{x} \cdot \hat{\boldsymbol{x}} = D - 2$.
3. $b_i < 0$

Then, by complementary slackness and the non-negativity of the slack variables we show that the stationarity conditions of the weights and biases cannot hold. Thus, memorization solutions are not KKT points and GF cannot converge to them.

6 DNF RECOVERY AS NORM MINIMIZATION

In the previous section we have shown that GF does not converge to memorization solutions. This result followed since GF converges to KKT points of the minimum norm problem and memorization solutions are not KKT. However, we would like to understand what are the KKT points that GF does converge to.

To address this question, we focus on characterizing KKT points which are also *global minimizers* of the minimum norm problem. The reason is that there is growing theoretical evidence that GF is biased towards solutions that minimize norms (or analogously, solutions that maximize margins). This has been shown for logistic regression [Soudry et al., 2018] and linear networks [Ji and Telgarsky, 2019a]. In a setting which is closest to ours, Chizat and Bach [2020] show that GF trained on two-layer nonlinear networks converges to maximum margin solutions. Their result holds for infinite 2-homogeneous networks with squared ReLU activations. Therefore, it cannot be applied in our setting. Nonetheless, all of the aforementioned results provide a strong motivation to study minimum norm solutions in our setting, to better understand the inductive bias of GF.

Analyzing the global solutions of the optimization problem in Eq. (4) is a major challenge since the problem is non-convex. To make headway, in this section we analyze these solutions under two technical assumptions (see Assumption 6.1 and Assumption 6.2). We define DNF recovery solutions (see Definition 6.2) as solutions which are aligned with the terms of the DNF (similar to Figure 1a). We then prove our main result: that a network that globally optimizes Eq. (4) is a DNF recovery solution. This means that if GF globally optimizes Eq. (4) then it must converge to a DNF recovery solution. Furthermore, this result is in line with our experiments in Section 7 which show that GF converges to DNF recovery solutions.

We next formally define a DNF recovery solutions. We first define alignment of a neuron with a term.

Definition 6.1. *A neuron $i \in [r]$ is an aligning neuron with respect to a DNF f^*, if there exists $n \in [K]$ and $\lambda_i > 0$ such that $\boldsymbol{w}_i = \lambda_i \boldsymbol{t}_n^*$, and $b_i = \lambda_i(2 - \|\boldsymbol{t}_n^*\|_1)$. We refer to the neuron i as aligning with the term n, and to λ_i as the alignment coefficient of i.*

Next we define a DNF recovery solution.

Definition 6.2. *$\boldsymbol{\theta}$ is a DNF-recovery solution if $\forall n \in [K]$ there exists a set of neurons $\mathbb{I}$ such that every $i \in \mathbb{I}$ aligns with term n, $\sum_{i \in \mathbb{I}} \lambda_i = 1$, $\forall i_1, i_2 \in \mathbb{I}$ $\lambda_{i_1} = \lambda_{i_2}$ and all other neurons are zero.*

Thus, DNF recovery solutions are networks where all terms in f^* have corresponding neurons aligned with them. Furthermore, all other neurons are zero. In other words, a recovery solution encodes the DNF explicitly in the weights of the network (and thus the DNF can be easily recovered from the weights). The conditions on the λ_i are required for the global optimality results to hold. We note that any DNF recovery solution perfectly classifies the data.

Next, we state our technical assumptions:

Assumption 6.1. *The output layer bias is fixed to $c = -1$, and $(\boldsymbol{W}, \boldsymbol{b})$ are learned.*

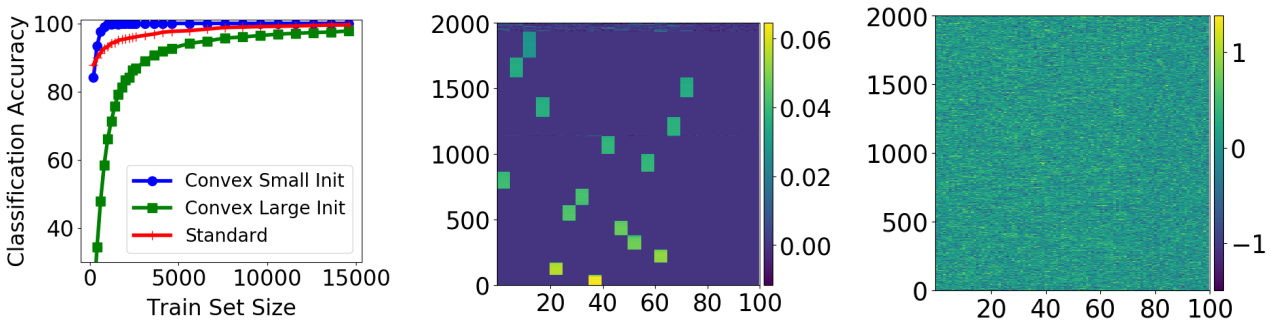
Assumption 6.2. *$\mathbb{S}_{\boldsymbol{x}} = \mathcal{X}$, i.e., we are in the population setting.*

Assumption 6.1 does not limit the expressive power of the network (see Theorem 3.1) and in the supplementary we show qualitatively that fixing $c = -1$ does not change the inductive bias of GF.[5] Many works have studied the population setting (Assumption 6.2) as a proxy to understand the performance in the empirical case [Daniely and Malach, 2020, Brutzkus and Globerson, 2017]. The population setting is a good test-bed to understand the inductive bias of GF since the loss has multiple zero training error solutions in this case (e.g., memorization and DNF recovery solutions).

We now state the main result of this section.

Theorem 6.1. *Consider the minimum norm optimization problem Eq. (4) under Assumption 6.1 and Assumption 6.2. Then, any globally optimal solution $\boldsymbol{\theta}^* = (\boldsymbol{W}^*, \boldsymbol{b}^*)$ of Eq. (4) is a DNF-recovery solution.*

[5] Surprisingly, without this assumption the theoretical analysis becomes substantially more difficult. We leave the extension to any c for future work.



(a) (b) (c)

Figure 3: (a) **Accuracy Comparison for Different Models**: Test performance of a convex network with small Gaussian initialization, a convex network with large Gaussian initialization, and a standard two-layer networks with Xavier initialization. Here $D = 25$ and the target DNF has 4 terms of sizes 4, 5, 5 and 6. Each dot corresponds to the mean of 10 experiments with different initializations. (b) **SGD Learns a DNF recovery solution for $D = 100$:** The last 25 dimensions are noisy inputs whose corresponding literals do not appear in the DNF. (c) **SGD With large initialization doesn't learn a DNF recovery solution:** The target function is the same as in Figure 3b. The only different is the initialization size.

We next provide a high level sketch of the proof. The full proof is given in the supplementary.

First we notice two key properties of any perfect solution $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b})$ (Definition 3.1):[6]

(1) $\forall \boldsymbol{x} \in \mathbb{S}_+$ there exists $\mathbb{I} \subseteq [r]$ such that $\sum_{i \in \mathbb{I}} \boldsymbol{w}_i \cdot \boldsymbol{x} + b_i \geq 2$.

(2) $\forall \boldsymbol{x} \in \mathbb{S}_-, \forall i \in [r] \;\; \boldsymbol{w}_i \cdot \boldsymbol{x} + b_i \leq 0$.

These properties are together necessary and sufficient for solutions to be perfect. Next, we show the following upper bound on the bias of every neuron in a perfect solution, which depends on the non-bias weights of the neuron:

Lemma 6.1. *Under Assumption 6.1 and Assumption 6.2, if $\boldsymbol{\theta}$ is a perfect solution then every $i \in [r]$ satisfies:*

$$b_i \leq -||\boldsymbol{w}_i||_1 + 2 \sum_{n \in [K]} \max \left\{ \min_{j \in \mathbb{A}_n} \{w_{ij}\}, 0 \right\} \qquad (6)$$

Then, we characterize every neuron in the min-norm solution. We show that the bias of every neuron is exactly this bound. From this we deduce that: (1) All the weights of the neuron are non-negative. (2) Every neuron can have positive weights only in positions that correspond to one term. (3) Every neuron aligns with some term or is equal to zero. To show each property we assume by contradiction that it holds and construct a perfect solution with lower norm. This leads to a contradiction since the solution has minimal norm. Finally, given the above properties we show that (1) neurons that are aligned with the same term have the same alignment coefficients. (2) The sum of the alignment coefficients over the a set of all neurons that are aligned with the same term is 1. The theorem follows from the aforementioned properties.

[6]Note that the set of perfect solutions is exactly the set of feasible solutions of Eq. (4).

7 EMPIRICAL RESULTS

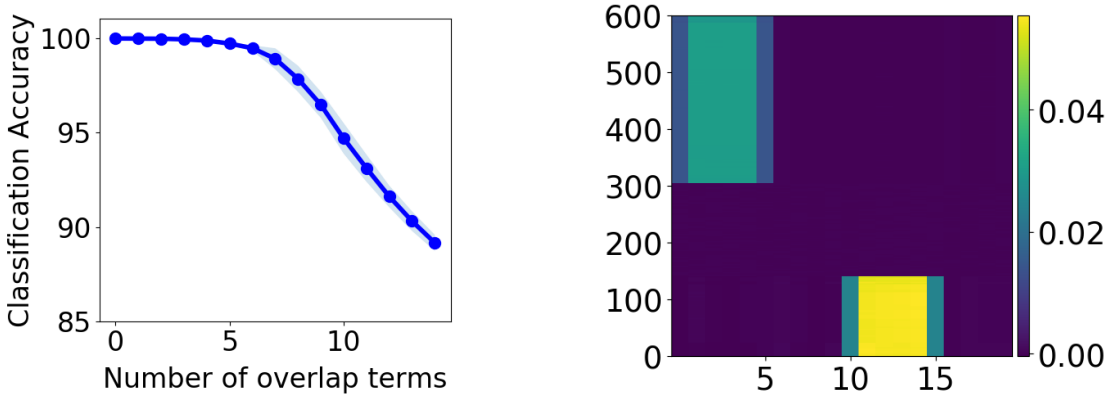
In this section we perform numerous experiments that support our analysis and show that our conclusions hold in different settings. For each experiment we show a sample of the empirical results due to space constraints. Further details and results are provided in the supplementary.

Comparing convex and standard networks: Our analysis focused on a convex network. Here we compare it to a standard two-layer network with trainable output layer. Figure 3a reports results, showing that the convex network outperforms the standard one for a DNF with $D = 25$. This shows that the convex network is a good model for studying inductive bias in our setting.

Comparing large and small initializations: In Figure 3a we compare a convex network with small initialization and a convex network with large initialization which is analogous to training in the NTK regime [Chizat et al., 2019]. The small initialization convex network performs better. We also show in Figure 3c that the small initialization network converges to a solution that aligns with the terms of the DNF while the large initialization network does not.

DNFs with large input dimension: Here we show an experiment for learning a DNF with 15 terms of size 5 and $D = 100$. We learned the DNF using a convex network and SGD with small Gaussian initialization and 15,000 training samples drawn from the uniform distribution. In Figure 3b, we see that SGD converges to a solution that aligns with the terms of the DNF and has 100 % test accuracy.

Non read-once DNFs: Our work is restricted to read-once DNFs. We preform a series of experiments for learning DNFs with overlapping terms. Figure 4a shows that when we increase the number of overlapping terms, the general-



(a) (b)

Figure 4: (a) **Evaluating the Effect of Overlap:** Here we experimented with $D = 100$ and all DNFs had 15 terms of size 5. We considered non read-once DNFs where the number of terms that share the variable x_1 varies. The x axis shows the number of overlapping terms. The training size was $8,500$ for all DNFs. Each dot corresponds to the mean of 10 experiments with different initializations. (b) **SGD Does not Learn a DNF recovery solution:** We trained a convex network to learn the following 4-term DNF: $(x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5) \vee (x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_6) \vee (x_{10} \wedge x_{11} \wedge x_{12} \wedge x_{13} \wedge x_{14}) \vee (x_{11} \wedge x_{12} \wedge x_{13} \wedge x_{14} \wedge x_{15})$ where $D = 20$. The training set size has 15,000 samples and the test classification accuracy is 100%.

ization error gets worse. Figure 4b shows an example of the neurons when learning a DNF with 4 overlapping terms. Here, the neurons do not align with the terms, and therefore the inductive bias is different from DNF recovery soluitions.

Experiments on Tabular Datasets: The fact that SGD recovers simple Boolean formulas is very attractive in the context of interpretability. In Section 4 we showed that we can reconstruct DNFs under certain idealized assumptions (e.g., uniform distribution, read-once). However, our reconstruction method might produce meaningless reconstructions on datasets which are not uniform nor labeled with a read-once DNF. We tested our reconstruction method on three tabular UCI datasets kr-vs-kp, diabetes and Splice [Dua and Graff, 2017]. We note that these datasets do not contain personally identifiable information or offensive content.

Learning with our convex network resulted in test accuracies of 100%, 96% and 97% on these datasets, respectively. Our reconstruction method obtained a small DNF (6 terms of size less than 4) on kr-vs-kp with test accuracy 91%. For diabetes, the reconstruction method returned a large DNF (more than 10 terms) with test accuracy 88%. On Splice we got a 2-term DNF of sizes 2 and 3 with 95% test accuracy. The latter is a very compact DNF with very small loss in accuracy, illustrating the potential of recovery on interpretability.

8 CONCLUSIONS

Understanding the inductive bias of neural networks for learning DNFs is an important and difficult theoretical challenge. In this work we focused on a setup of learning read-once DNFs under the uniform distribution with a convex network. We empirically observed that GF converges to solutions that align with the DNF terms. We then proved that GF cannot converge to solutions that memorize the training points, despite the fact that they minimize the training accuracy. We additionally proved that under certain assumptions, solutions that minimize the norm are solutions that align with the terms of the DNF. Together with recent results which show that GF is biased towards minimization of norms, this corroborates our empirical findings.

Our work has several limitations which are mainly due to the fact that analyzing nonlinear networks with nonlinear data is a major challenge: (1) We only consider a setup of uniform distribution and read-once DNF. We also restrict analysis to convex networks, but our empirical results actually suggest that convex networks may be preferable to standard ones in our setting. (2) We do not show an end-to-end convergence of GF to DNF recovery solutions. (3) We do not address the sample complexity of GF in our setting (however, it is clear that an inductive bias towards alignment will improve sample complexity, since alignment reduces the effective number of parameters the network uses).

Our work opens up many interesting directions for future work. For example, it would be interesting to understand if DNF recovery is possible for other distributions and DNFs that are not read-once. Another interesting direction is to understand the sample complexity of neural networks for learning DNFs and how it relates to DNF-recovery. Finally, it will be interesting to understand how learning dynamics in neural nets are related to other algorithms for learning DNFs.

REFERENCES

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

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155, 2017.

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

Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. In *International Conference on Machine Learning*, pages 605–614, 2017.

Alon Brutzkus and Amir Globerson. Why do larger models generalize better? a theoretical perspective via the xor problem. In *International Conference on Machine Learning*, pages 822–830, 2019.

Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. SGD learns over-parameterized networks that provably generalize on linearly separable data. *International Conference on Learning Representations*, 2018.

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 10836–10846, 2019.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. *arXiv preprint arXiv:2002.04486*, 2020.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2933–2943, 2019.

Amit Daniely and Eran Malach. Learning parities with neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, pages 2253–2261, 2016.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685, 2019.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *International Conference on Learning Representations*, 2018.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Joydeep Dutta, Kalyanmoy Deb, Rupesh Tulshyan, and Ramnik Arora. Approximate kkt points and a proximity measure for termination. *Journal of Global Optimization*, 56(4):1463–1499, 2013.

Jonathan Fiat, Eran Malach, and Shai Shalev-Shwartz. Decoupling gating from linearity. *arXiv preprint arXiv:1906.05032*, 2019.

Spencer Frei, Yuan Cao, and Quanquan Gu. Provable generalization of sgd-trained neural networks of any width in the presence of adversarial label noise. *arXiv preprint arXiv:2101.01152*, 2021.

Sebastian Goldt, Madhu Advani, Andrew M Saxe, Florent Krzakala, and Lenka Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. In *Advances in Neural Information Processing Systems*, pages 6981–6991, 2019.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *ICLR*, 2019a.

Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. In *International Conference on Learning Representations*, 2019b.

Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. *arXiv preprint arXiv:1808.01204*, 2018.

Yuanzhi Li, Tengyu Ma, and Hongyang R Zhang. Learning over-parametrized two-layer neural networks beyond ntk. In *Conference on Learning Theory*, pages 2613–2682, 2020.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *ICLR*, 2020.

Eran Malach and Shai Shalev-Shwartz. The implications of local correlation on learning some deep functions. *Advances in Neural Information Processing Systems*, 33, 2020.

Yishay Mansour and Mariono Schain. Learning with maximum-entropy distributions. In *Machine Learning*, pages 123–145, 2001.

Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv preprint arXiv:1911.01544*, 2019.

Mor Shpigel Nacson, Suriya Gunasekar, Jason Lee, Nathan Srebro, and Daniel Soudry. Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models. In *International Conference on Machine Learning*, pages 4683–4692, 2019.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

Ryan O'Donnell. *Analysis of Boolean functions*. Cambridge University Press, 2014.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Mary Phuong and Christoph H Lampert. The inductive bias of relu networks on orthogonally separable data. *ICLR*, 2021.

Leonard Pitt and Leslie G Valiant. Computational limitations on learning from examples. *Journal of the ACM (JACM)*, 35(4):965–984, 1988.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

David Saad and Sara A Solla. Dynamics of on-line gradient descent learning for multilayer neural networks. In *Advances in neural information processing systems*, pages 302–308, 1996.

Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer relu neural networks. In *International Conference on Machine Learning*, pages 4433–4441. PMLR, 2018.

Roei Sarussi, Alon Brutzkus, and Amir Globerson. Towards understanding learning in neural networks with linear teachers. *arXiv preprint arXiv:2101.02533*, 2021.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Yuandong Tian. Student specialization in deep relu networks with finite width and input dimension. *arXiv*, pages arXiv–1909, 2019.

Karsten Verbeurgt. Learning dnf under the uniform distribution in quasi-polynomial time. In *Proceedings of the third annual workshop on Computational learning theory*, pages 314–326, 1990.

Blake Woodworth, Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Kernel and deep regimes in overparametrized models. *arXiv preprint arXiv:1906.05827*, 2019.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, 2017.

Here we provide complete proofs for the results in the main paper, as well as additional empirical results.

A GRADIENT FLOW - ASSUMPTIONS

When gradient flow is implemented on non-differentiable functions (e.g., ReLU) the implementation can choose from among a set of possible sub-differentials. Here we define which of these our analysis will use. This choice corresponds to the common way gradient methods are implemented in practice for the ReLU function.

Recall that the gradient flow step is $\frac{d\boldsymbol{\theta}^{(t)}}{dt} \in -\partial^\circ L(\boldsymbol{\theta}^{(t)})$ for a.e. t, where:

$$\partial^\circ f(\boldsymbol{x}) = \text{conv}\left\{\lim_{k\to\infty} \nabla f(\boldsymbol{x}_k) \mid \boldsymbol{x}_k \to \boldsymbol{x} \text{ and } f \text{ is differentiable at } \boldsymbol{x}_k\right\} \tag{7}$$

is the Clarke's subdifferential.

As discussed in the main text, we will assume that the gradient flow step selects a specific vector in the subdifferential. This is done by setting the subgradient of ReLU at 0 in advance to a constant value $a \in [0,1]$. Namely, this value of the subgradient is used for all neurons and in all iterations. Usually a is set to be either 0 or 1.

Formally, for each $i \in r$ we denote $\frac{d\boldsymbol{w}_i^{(t)}}{dt} = \frac{1}{m}\sum_{\boldsymbol{x}\in\mathbb{S}} \frac{d\boldsymbol{w}_i^{(t)}(\boldsymbol{x})}{dt}$. Here, $\frac{d\boldsymbol{w}_i^{(t)}(\boldsymbol{x})}{dt}$ is the gradient update of $\boldsymbol{w}_i^{(t)}$ restricted to the summand that depends on $\boldsymbol{x}$ in L (Eq. (2) of the main paper). Similarly, we denote $\frac{db_i^{(t)}}{dt} = \frac{1}{m}\sum_{\boldsymbol{x}\in\mathbb{S}} \frac{db_i^{(t)}(\boldsymbol{x})}{dt}$. For our result, we need the following technical assumption.

Assumption A.1. *There exists an $a \in [0,1]$ such that for every step $t > 0$, every neuron $i \in [r]$, and every sample $\boldsymbol{x} \in \mathbb{S}$ if $\boldsymbol{w}_i^{(t)} \cdot \boldsymbol{x} + b_i^{(t)} = 0$ then $\frac{d\boldsymbol{w}_i^{(t)}(\boldsymbol{x})}{dt} = ay\ell'\left(yN(\boldsymbol{x};\boldsymbol{\theta}^{(t)})\right)\boldsymbol{x}$ and $\frac{db_i^{(t)}}{dt} = ay\ell'\left(yN(\boldsymbol{x};\boldsymbol{\theta}^{(t)})\right)$.*

B PROOF OF THEOREM 5.1

We first need the following notation and recall the KKT conditions in our context. Let $\partial^\circ\sigma(\boldsymbol{w}_i \cdot \boldsymbol{x}_l + b) \subseteq \mathbb{R}^{D+1}$ be the subdifferential of neuron i given input $\boldsymbol{x}_l$. It holds that:

$$\partial^\circ\sigma(\boldsymbol{w}_i \cdot \boldsymbol{x}_l + b) = \begin{cases} \{(\boldsymbol{x}_l, 1)\} & \text{if } \boldsymbol{w}_i \cdot \boldsymbol{x}_l + b_i > 0 \\ \{(\boldsymbol{0}, 0)\} & \text{if } \boldsymbol{w}_i \cdot \boldsymbol{x}_l + b_i < 0 \\ [0,1]^{D+1} & \text{if } \boldsymbol{w}_i \cdot \boldsymbol{x}_l + b_i = 0 \end{cases}$$

KKT conditions: A feasible point $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b}, c)$ of the min norm problem (Eq. (4) of the main paper) is a KKT point if there exist $\lambda_1, \ldots \lambda_m \geq 0$ such that:

1. **Stationarity**:
$$\forall i \in [r],\ \ \boldsymbol{w}_i = \sum_{l\in[m]} \lambda_l y_l \boldsymbol{h}_{il} \text{ and } b_i = \sum_{l\in[m]} \lambda_l y_l g_{il} \tag{8}$$
and
$$c = \sum_{j\in[m]} \lambda_j y_j \tag{9}$$
where $(\boldsymbol{h}_{il}, g_{il}) \in \partial^\circ\sigma(\boldsymbol{w}_i \cdot \boldsymbol{x}_l + b)$.
2. **Complementary slackness**: if $y_i N(\boldsymbol{x}_i, \theta) > 1$, then $\lambda_i = 0$.

We now proceed to prove the theorem. In the first part, we show that by Assumption A.1, we can restrict the possible values of $\boldsymbol{h}_{il}, g_{il}$ for a KKT point that GF converges to. In the second part, we prove properties of neurons that memorize samples. In the third part, we use the previous parts to show that memorizing solutions cannot be KKT points.

<u>Part 1:</u> In this part we prove the following lemma.

Lemma B.1. *Assume that Assumption A.1 holds and GF converges to a KKT point with parameters $\boldsymbol{h}_{il}$ and g_{il} for $1 \leq i \leq r$ and $1 \leq l \leq n$, as defined in Eq. (8). Then, $(\boldsymbol{h}_{il}, g_{il}) \in \{(\boldsymbol{x}_l, 1), (a\boldsymbol{x}_l, a), (0,0)\}$.*

Proof. For each $1 \le l \le n$, $1 \le i \le r$ and $t > 0$ let $\left(\boldsymbol{h}_{il}^{(t)}, g_{il}^{(t)}\right) \in \partial^\circ \sigma\left(\boldsymbol{w}_i^{(t)} \cdot \boldsymbol{x}_l + b_i^{(t)}\right)$ be the corresponding values of the GF step at time t in the subdifferential $\partial^\circ \sigma\left(\boldsymbol{w}_i^{(t)} \cdot \boldsymbol{x}_l + b_i^{(t)}\right)$. By inspecting the proofs of Lyu and Li [2020] and Dutta et al. [2013], we see that $(\boldsymbol{h}_{il}, g_{il})$ is equal to the limit of a convergent subsequence of $\left\{\left(h_{il}^{(t_j)}, g_{il}^{(t_j)}\right)\right\}_{j=0}^{\infty}$.

By assumption A.1, we know that for each j, $\left(\boldsymbol{h}_{il}^{(t_j)}, g_{il}^{(t_j)}\right) \in \{(\boldsymbol{x}_l, 1), (a\boldsymbol{x}_l, a), (\boldsymbol{0}, 0)\}$. Therefore, the limit also satisfies $(\boldsymbol{h}_{il}, g_{il}) \in \{(\boldsymbol{x}_l, 1), (a\boldsymbol{x}_l, a), (\boldsymbol{0}, 0)\}$.

□

Part 2: We will first need the following definition.

Definition B.1. *Given a sample $\hat{\boldsymbol{x}} \in \mathcal{X}$ and index $j \in [D]$, the sample with Hamming distance one from $\hat{\boldsymbol{x}}$ at index j is defined as $\mathcal{H}(\hat{\boldsymbol{x}}, j) \in \mathcal{X}$ and satisfies the following:*

$$\mathcal{H}(\hat{\boldsymbol{x}}, j)_j = -x_j \text{ and } \forall j' \in [D] \backslash \{j\} \;\; \mathcal{H}(\hat{\boldsymbol{x}}, j)_{j'} = x_{j'} \tag{10}$$

The set of all samples with Hamming distance one from $\hat{\boldsymbol{x}}$ is defined as: $\Psi(\hat{\boldsymbol{x}}) = \{\boldsymbol{x}' \in \mathcal{X} \mid \exists j \in [D] \;\; \mathcal{H}(\hat{\boldsymbol{x}}, j) = \boldsymbol{x}'\}$. Note, $|\Psi(\boldsymbol{x})| = D$

Using this definition we rephrase Lemma 5.1 and prove those properties of memorizing neurons (Definition 5.1 of the main paper):

Lemma B.2. *Let $D > 2$. If a neuron $i \in [r]$ memorizes a sample $\hat{\boldsymbol{x}} \in \mathbb{S}_x$, then it satisfies the following properties:*

1. $\hat{x}_j = \operatorname{sign}(w_{ij})$ *for all* $1 \le j \le D$.
2. *For $\boldsymbol{x} \in \mathcal{X}$ if $\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i = 0$ then $\boldsymbol{x} \in \Psi(\hat{\boldsymbol{x}})$.*
3. $b_i < 0$

Proof. **Property 1:** Assume by contradiction that there exists $j \in [D]$ such that $\operatorname{sign}(w_{ij}) \neq \hat{x}_j$. Then $\boldsymbol{w}_i \cdot \mathcal{H}(\hat{\boldsymbol{x}}, j) + b_i \ge \boldsymbol{w}_i \cdot \hat{\boldsymbol{x}} + b_i > 0$, in contradiction to the memorization assumption in Eq. (5) of the main paper.

Property 2: Assume by contradiction that there exists $\boldsymbol{x} \in \mathcal{X} \backslash (\Psi(\hat{\boldsymbol{x}}) \cup \{\hat{\boldsymbol{x}}\})$ such that $\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i = 0$. We define $J = \{j \in [D] \mid x_j = -\hat{x}_j\}$ and $j' \in J$ for some index in J. By Property 1, the sample $\widetilde{\boldsymbol{x}} = \mathcal{H}(\hat{\boldsymbol{x}}, j')$ satisfies the following:

$$\begin{aligned}
\boldsymbol{w}_i \cdot \boldsymbol{x} - \boldsymbol{w}_i \cdot \widetilde{\boldsymbol{x}} &= \sum_{j \in [D] \backslash J} w_{ij} x_j + \sum_{j \in J} w_{ij} x_j - \sum_{j \in [D] \backslash \{j'\}} w_{ij} \widetilde{x}_j - w_{ij'} \widetilde{x}_j \\
&= \sum_{j \in [D] \backslash J} w_{ij} \hat{x}_j - \sum_{j \in J} w_{ij} \hat{x}_j - \sum_{j \in [D] \backslash \{j'\}} w_{ij} \hat{x}_j + w_{ij'} \hat{x}_j \\
&= \sum_{j \in [D] \backslash J} |w_{ij}| - \sum_{j \in J} |w_{ij}| - \sum_{j \in [D] \backslash \{j'\}} |w_{ij}| + |w_{ij'}| = -2 \sum_{j \in J \backslash \{j'\}} |w_{ij}|
\end{aligned} \tag{11}$$

Since $\boldsymbol{x} \notin \Psi(\hat{\boldsymbol{x}})$, it holds that $J \backslash \{j'\} \neq \emptyset$. Furthermore, by Property 1, $\forall j \in J \backslash \{j'\} \;\; w_{ij} \neq 0$. Thus, $\sum_{j \in J \backslash \{j'\}} |w_{ij}| > 0$ which implies that $\boldsymbol{w}_i \cdot \boldsymbol{x} < \boldsymbol{w}_i \cdot \widetilde{\boldsymbol{x}}$. We know that $\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i = 0$, therefore $\boldsymbol{w}_i \cdot \widetilde{\boldsymbol{x}} + b_i > 0$. This contradicts the memorization assumption in Eq. (5) of the main paper.

Property 3: Assume by contradiction that $b_i \ge 0$. We define $j' = \arg\min_{j \in [D]} \{|w_{ij}|\}$. For the sample $\widetilde{\boldsymbol{x}} = \mathcal{H}(\hat{\boldsymbol{x}}, j')$, the following holds by Property 1:

$$\boldsymbol{w}_i \cdot \widetilde{\boldsymbol{x}} = \sum_{j \in [D] \backslash \{j'\}} w_{ij} \hat{x}_j - w_{ij'} \hat{x}_{j'} = \sum_{j \in [D] \backslash \{j'\}} |w_{ij}| - |w_{ij'}| > 0 \tag{12}$$

Since $D > 2$, we have $\sum_{j \in [D] \backslash \{j'\}} |w_{ij}| - |w_{ij'}| > 0$. Thus, $\boldsymbol{w}_i \cdot \widetilde{\boldsymbol{x}} + b_i > 0$ which contradicts the memorization assumption in Eq. (5) of the main paper. □

Part 3: Now, we can proceed to prove Theorem 5.1.

Consider a network with parameters $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b}, c)$, neuron $i \in [r]$ and a sample $(\hat{\boldsymbol{x}}, \hat{y}) \in \mathbb{S}$ such that Eq. (5) of the main paper holds (i.e., the neuron i memorizes the sample $\hat{x}$). We assume by contradiction that there exists an initialization $\boldsymbol{\theta}^{(0)}$ such that if we run gradient flow from $\boldsymbol{\theta}^{(0)}$ using μ then the weights $\boldsymbol{\theta}^{(t)}$ will converge to $\boldsymbol{\theta}$. According to the results of Lyu and Li [2020], Ji and Telgarsky [2020], we know that there exists $\alpha > 0$ such that $\tilde{\boldsymbol{\theta}} = (\bar{\boldsymbol{W}}, \bar{\boldsymbol{b}}, \bar{c}) = \alpha\bar{\boldsymbol{\theta}}$ is a KKT point of Eq. (4) of the main paper and Eq. (8) holds for $\tilde{\boldsymbol{\theta}}$. Note that for $\tilde{\boldsymbol{\theta}}$, neuron i memorizes the sample $(\hat{\boldsymbol{x}}, \hat{y})$ as well.

Given a sample $\boldsymbol{x}_l \in \mathbb{S}_x$ we can see that the following holds:

1. If $\boldsymbol{x}_l = \hat{\boldsymbol{x}}$ then $(\boldsymbol{h}_{il}, g_{il}) = \{(\hat{\boldsymbol{x}}, 1)\}$.
2. If $\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}})$ then $(\boldsymbol{h}_{il}, g_{il}) = \{(a\boldsymbol{x}, a)\}$ where $a \in [0, 1]$ by Assumption A.1.
3. If $\boldsymbol{x}_l \notin \Psi(\hat{\boldsymbol{x}})$ then $(\boldsymbol{h}_{il}, g_{il}) = \{(\boldsymbol{0}, 0)\}$ by Property 2 of Lemma B.2.

We will show that for every $\lambda_1, \ldots, \lambda_{|\mathbb{S}|} \geq 0$, Eq. (8) does not hold. We can assume without loss of generality that all the samples in $\boldsymbol{x}_l \in \Psi(\boldsymbol{x}) \cap \mathbb{S}_x$ are support vectors and $(\boldsymbol{h}_{il}, g_{il}) = (\boldsymbol{x}_l, 1)$. This is because if one of the samples is not a support vector then we can take $\lambda_l = 0$. Furthermore, if $(\boldsymbol{h}_{il}, g_{il}) = (\boldsymbol{0}, 0)$ then we can take $\lambda_l = 0$ and if $(\boldsymbol{h}_i, g_i) = (a\boldsymbol{x}_l, a)$ for $a > 0$ we can set $\frac{\lambda_l}{a}$ instead of λ_l. Under this assumption we can write Eq. (8) for $\tilde{\boldsymbol{\theta}}$ using only $\lambda_1, \ldots, \lambda_D$ and $\hat{\lambda}$ that correspond to the samples of $\Psi(\hat{\boldsymbol{x}})$ and $\hat{\boldsymbol{x}}$, respectively:

$$\boldsymbol{w}_i = \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}}) \cap \mathbb{S}_x} \lambda_l y_l \boldsymbol{x}_l + \hat{\lambda}\hat{y}\hat{\boldsymbol{x}} \quad \text{and} \quad b_i = \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}}) \cap \mathbb{S}_x} \lambda_l y_l + \hat{\lambda}\hat{y} \tag{13}$$

Assume by contradiction that there exists $\boldsymbol{x}_{l'} = \mathcal{H}(\hat{\boldsymbol{x}}, j')$ such that $\boldsymbol{x}_{l'} = \Psi(\hat{\boldsymbol{x}}) \backslash \mathbb{S}_x$. The following holds by Eq. (13) and Definition B.1:

$$\begin{aligned} &(1)\ \ w_{ij'} = \hat{\lambda}\hat{y}\hat{x}_{j'} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}}) \cap \mathbb{S}_x} \lambda_l y_l \hat{x}_{j'} \\ &(2)\ \ b_i = \hat{\lambda}\hat{y} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}}) \cap \mathbb{S}_x} \lambda_l y_l \end{aligned} \tag{14}$$

Using the first property in Lemma B.2:

$$\begin{aligned} &(1)\ \ w_{ij'} = \hat{\lambda}\hat{y}\,\mathrm{sign}(w_{ij'}) + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}}) \cap \mathbb{S}_x} \lambda_l y_l\,\mathrm{sign}(w_{ij'}) \\ &(2)\ \ b_i = \hat{\lambda}\hat{y} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}}) \cap \mathbb{S}_x} \lambda_l y_l \end{aligned} \tag{15}$$

Therefore,

$$\begin{aligned} &(1)\ \ |w_{ij'}| = \hat{\lambda}\hat{y} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}}) \cap \mathbb{S}_x} \lambda_l y_l \\ &(2)\ \ b_i = \hat{\lambda}\hat{y} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}}) \cap \mathbb{S}_x} \lambda_l y_l \end{aligned} \tag{16}$$

This means that $0 \leq |w_{ij'}| = b_i$, which is in contradiction to the third property of Lemma B.2. Therefore, we can assume from now on that $\Psi(\hat{\boldsymbol{x}}) \subseteq \mathbb{S}_x$, and we can write the KKT conditions as follows:

$$\boldsymbol{w}_i = \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}})} \lambda_l y_l \boldsymbol{x}_l + \hat{\lambda}\hat{y}\hat{\boldsymbol{x}} \quad \text{and} \quad b_i = \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}})} \lambda_l y_l + \hat{\lambda}\hat{y} \tag{17}$$

Given $\boldsymbol{x}_{l'} = \mathcal{H}(\hat{\boldsymbol{x}}, j')$, the following holds by Eq. (17) and Definition B.1:

$$\begin{aligned} &(1)\ \ w_{ij'} = \hat{\lambda}\hat{y}\hat{x}_{j'} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}}) \backslash \{\boldsymbol{x}_{l'}\}} \lambda_l y_l \hat{x}_{j'} - \lambda_{l_1} y_{l_1} \hat{x}_{j'} \\ &(2)\ \ b_i = \hat{\lambda}\hat{y} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}})} \lambda_l y_l \end{aligned} \tag{18}$$

Using the first property in Lemma B.2:

$$(1)\ \ w_{ij'} = \hat{\lambda}\hat{y}\,\text{sign}(w_{ij'}) + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}})\setminus\{\boldsymbol{x}_{l'}\}} \lambda_l y_l\,\text{sign}(w_{ij'}) - \lambda_{l_1} y_{l'}\,\text{sign}(w_{ij'})$$
$$(2)\ \ b_i = \hat{\lambda}\hat{y} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}})} \lambda_l y_l \tag{19}$$

Therefore,

$$(1)\ \ |w_{ij'}| = \hat{\lambda}\hat{y} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}})\setminus\{\boldsymbol{x}_{l'}\}} \lambda_l y_l - \lambda_{l'} y_{l'}$$
$$(2)\ \ b_i = \hat{\lambda}\hat{y} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}})} \lambda_l y_l \tag{20}$$

The result of subtracting $(2) - (1)$ is:

$$b_i - |w_{ij'}| = 2\lambda_{l'} y_{l'} \tag{21}$$

Next we show that it must hold that $y_{l'} = -1$. To see this, note that the third property of Lemma B.2 implies $b_i < 0$ and therefore $b_i - |w_{ij'}| < 0$. Assuming in contradiction that $y_{l'} = 1$, the RHS of Eq. (21) satisfies $0 \leq 2\lambda_{l'} y_{l'}$. But we just saw that the LHS satisfies $b_i - |w_{ij'}| < 0$. We therefore have a contradiction and conclude that $y_{l'} = -1$. We can conclude that $\Psi(\hat{\boldsymbol{x}})$ contains only negative samples.

Next we argue that $\hat{y} = -1$. To see this, assume in contradiction that $\hat{y} = 1$. Then there exists a $n \in [K]$ such that $\hat{\boldsymbol{x}}$ satisfies the term t_n^*. Due to the fact that $K \geq 2$, we know that there exists $j \in [D]\setminus\mathbb{A}_n$. Then, $\mathcal{H}(\hat{\boldsymbol{x}}, j) \in \Psi(\hat{\boldsymbol{x}})$ is a positive sample in contradiction to the fact that $\Psi(\hat{\boldsymbol{x}})$ contains only negative samples, and therefore $\hat{y} = -1$

By Eq. (21) we know that for every $j \in [D]$ a sample $\boldsymbol{x}_l = \mathcal{H}(\hat{\boldsymbol{x}}, j)$ satisfies the following: $\lambda_l = \frac{1}{2}(|w_{ij}| - b_i)$. If we assign this in Eq. (20), we get:

$$|w_{ij'}| = \hat{\lambda}\hat{y} + \sum_{\boldsymbol{x}_l \in \Psi(\hat{\boldsymbol{x}})\setminus\{\boldsymbol{x}_{l'}\}} \frac{1}{2}(b_i - |w_{ij}|) - \frac{1}{2}(b_i - |w_{ij'}|) \tag{22}$$

Therefore,

$$0 = \hat{\lambda}\hat{y} + \frac{1}{2}(D-2)b_i - \frac{1}{2}||\boldsymbol{w}_i||_1 \tag{23}$$

But this results in a contradiction, because we know that $\hat{\lambda}\hat{y} \leq 0$, $\frac{1}{2}(D-2)b_i < 0$ and $-\frac{1}{2}||\boldsymbol{w}_i||_1 < 0$.

Thus, we conclude that gradient flow cannot converge to memorizing solutions.

C PROOF OF THEOREM 6.1

We prove the theorem in several parts. We first prove a property of a perfect solution (Section C.1). In Section C.2 we prove several results regarding the bias threshold. In Section C.3 we prove auxiliary lemmas and in Section C.4 we prove the alignment of the neurons of the optimal solutions to the terms of the DNF. We conclude the proof in Section C.5.

C.1 A SIMPLE PROPERTY OF PERFECT SOLUTIONS

Recall the definition $\mathbb{S}_+ = \{\boldsymbol{x} \mid (\boldsymbol{x}, 1) \in \mathbb{S}\}$. We also define $\mathbb{S}_- = \{\boldsymbol{x} \mid (\boldsymbol{x}, -1) \in \mathbb{S}\}$. We first need the following definitions. We say that a solution $(\boldsymbol{W}, \boldsymbol{b})$ satisfies the MIN$_+$ property if for any positive point $\boldsymbol{x} \in \mathbb{S}_+$ there exists $\mathbb{I} \subseteq [r]$ such that $\sum_{i\in\mathbb{I}} \boldsymbol{w}_i \cdot \boldsymbol{x} + b_i \geq 2$. We say that a solution satisfies the MIN$_-$ property if for any negative point $\boldsymbol{x} \in \mathbb{S}_-$ and for all $i \in [r]$, $\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i \leq 0$.

Lemma C.1. *Assume that $\mathbb{S}_x = \mathcal{X}$. Then $(\boldsymbol{W}, \boldsymbol{b})$ is a perfect solution if and only if $(\boldsymbol{W}, \boldsymbol{b})$ satisfies MIN$_+$ and MIN$_-$.*

Proof. If $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b})$ is a perfect solution, then for all $(\boldsymbol{x}, y) \in \mathbb{S}$, $yN(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{b}) \geq 1$. Therefore, if $y = 1$, $\sum_{i\in[r]} \sigma(\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i) \geq 2$. Thus, there exists $\mathbb{I} \subseteq [r]$ such that $\sum_{i\in\mathbb{I}} \boldsymbol{w}_i \cdot \boldsymbol{x} + b_i \geq 2$ and the MIN$_+$ holds. If $y = -1$, then $\sum_{i\in[r]} \sigma(\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i) \leq 0$ and therefore for all $i \in [r]$, $\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i \leq 0$. The other direction follows similarly. □

We note that one direct consequence of Lemma C.1 is that given $\boldsymbol{\theta}$ if a negative $\boldsymbol{x} \in \mathbb{S}_-$ is activated by a neuron $i \in [r]$, i.e., $\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i > 0$, then the MIN$_-$ property doesn't hold, and $\boldsymbol{\theta}$ is not a perfect solution.

C.2 PROOF OF LEMMA 6.1

In this section we show that when $\mathbb{S}_x = \mathcal{X}$, the bias of any neuron in a perfect solution is upper bounded by a certain value which we call the *bias threshold*. To simplify the formulation of this section we define the following:

Definition C.1. *We define the set of indices which are not active in any term as the noisy indices and denote them by* $\mathbb{A}_{K+1} = [D] \setminus \cup_{n\in[K]} \mathbb{A}_n$.

Definition C.2. *For each term $n \in [K]$ of f^* and $i \in [r]$ define* $V_n(\boldsymbol{w}_i) = \max\left\{\min_{j\in\mathbb{A}_n}\{w_{ij}\}, 0\right\}$.

Definition C.3. *The bias threshold for a weight $\boldsymbol{w}$ is* $BT(\boldsymbol{w}) = -||\boldsymbol{w}||_1 + 2\sum_{n\in[K]} V_n(\boldsymbol{w})$.

Using those definitions we rephrase Lemma 6.1 and prove it:

Lemma C.2. *Assume that $\mathbb{S}_x = \mathcal{X}$. $(\boldsymbol{W}, \boldsymbol{b})$ satisfies that $\forall i \in [r]\ \ b_i \leq BT(\boldsymbol{w}_i)$ and satisfies the MIN$_+$ property if and only if the network is a perfect solution.*

Proof. We will show that given a neuron $(\boldsymbol{w}, b)$, there is negative sample $\boldsymbol{x} \in \mathbb{S}_-$ for which $\boldsymbol{w} \cdot \boldsymbol{x} + b > 0$ if and only if $b > BT(\boldsymbol{w})$. By showing that we can conclude that $\forall i \in [r]\ \ b_i \leq BT(\boldsymbol{w}_i)$ if and only if the MIN$_-$ property holds. By Combining this with Lemma C.1, we can prove our claim.

Given neuron $(\boldsymbol{w}, b)$, we define the minimum index of a term $n \in [K]$ as $J_n = \arg\min_{j\in\mathbb{A}_n}\{w_j\}$. Consider a sample $\hat{\boldsymbol{x}} \in \mathbb{S}_x$ that is defined by:

$$\hat{x}_j = \begin{cases} -\operatorname{sign}(w_j) & \exists n \in [K]:\ \ V_n(\boldsymbol{w}) > 0\ \ \wedge\ \ j = J_n \\ \operatorname{sign}(w_j) & otherwise \end{cases} \tag{24}$$

For every term $n \in [K]$, if $V_n(\boldsymbol{w}) > 0$ then $\hat{x}_{J_n} = -\operatorname{sign}(\boldsymbol{w}_{J_n}) = -1$. Otherwise, $V_n(\boldsymbol{w}) = 0$ and there exists $j \in \mathbb{A}_n$ such that $w_j < 0$, i.e., $\boldsymbol{x}_j = \operatorname{sign}(w_j) = -1$. In any case, $\boldsymbol{x} \cdot \boldsymbol{t}_n^* < |\mathbb{A}_n|$. Therefore, the label of this sample is negative and denote it by $\hat{y} = -1$.

We show that $\boldsymbol{w} \cdot \hat{\boldsymbol{x}} = -BT(\boldsymbol{w})$ by,

$$\begin{aligned}
\boldsymbol{w} \cdot \hat{\boldsymbol{x}} &= \sum_{j\in[D]} w_j \cdot \hat{x}_j = \sum_{n\in[K+1]}\left[\sum_{j\in\mathbb{A}_n} w_j \cdot \hat{x}_j\right] \\
&= \sum_{n\in[K] \text{ and } V_n(\boldsymbol{w})>0}\left[\sum_{j\in\mathbb{A}_n\setminus\{J_n\}} w_j \cdot \operatorname{sign}(w_j) - w_{J_n} \cdot \operatorname{sign}(w_{J_n})\right] \\
&+ \sum_{n\in[K] \text{ and } V_n(\boldsymbol{w})=0}\left[\sum_{j\in\mathbb{A}_n} w_j \cdot \operatorname{sign}(w_j)\right] + \sum_{j\in\mathbb{A}_{K+1}} w_j \cdot \operatorname{sign}(w_j) \\
&= \sum_{n\in[K] \text{ and } V_n(\boldsymbol{w})>0}\left[\sum_{j\in\mathbb{A}_n\setminus\{J_n\}} |w_j| - |w_{J_n}|\right] + \sum_{n\in[K] \text{ and } V_n(\boldsymbol{w})=0}\left[\sum_{j\in AI_n} |w_j|\right] + \sum_{j\in\mathbb{A}_{K+1}} |w_j| \\
&= \sum_{n\in[K] \text{ and } V_n(\boldsymbol{w})>0}\left[\sum_{j\in\mathbb{A}_n} |w_j| - 2V_n(\boldsymbol{w})\right] + \sum_{n\in[K] \text{ and } V_n(\boldsymbol{w})=0}\left[\sum_{j\in AI_n} |w_j| - 2V_n(\boldsymbol{w})\right] + \sum_{j\in\mathbb{A}_{K+1}} |w_j| \\
&= \sum_{n\in[K]}\left[\sum_{j\in AI_n} |w_j| - 2V_n(\boldsymbol{w})\right] + \sum_{j\in\mathbb{A}_{K+1}} |w_j| = \sum_{j\in[D]} |w_j| - 2\sum_{n\in[K]} V_n(\boldsymbol{w}) \\
&= ||\boldsymbol{w}||_1 - 2\sum_{n\in[K]} V_n(\boldsymbol{w}) = -BT(\boldsymbol{w})
\end{aligned} \tag{25}$$

For the first direction, if $b > BT(\boldsymbol{w})$ then $\boldsymbol{w} \cdot \hat{\boldsymbol{x}} + b = BT(\boldsymbol{w}) + b > 0$, as desired.

In the second direction, assume that there is a negative sample $\boldsymbol{x} \in \mathbb{S}_-$ such that $\boldsymbol{w} \cdot \boldsymbol{x} + b > 0$. We will show that $\boldsymbol{x} \cdot \boldsymbol{w} \leq \hat{\boldsymbol{x}} \cdot \boldsymbol{w}$. Every term $n \in [K]$ satisfies for all $j \in \mathbb{A}_n \backslash \{J_n\}$:

$$x_j w_j \leq |w_j| = \text{sign}(w_j) w_j = \hat{x}_j w_j \tag{26}$$

If $V_n(\boldsymbol{w}) = 0$, then the index $j = J_n$ also satisfies Eq. (26), by the definition of $\hat{\boldsymbol{x}}$. Otherwise $V_n(\boldsymbol{w}) > 0$ and we know that there exists $j' \in \mathbb{A}_n$ such that $x_{j'} = -1$ (since $\boldsymbol{x}$ is negative), and $w_{J_n}, w_{j'} > 0$. If $J_n = j'$, then $x_{j'} w_{j'} = \hat{x}_{J_n} w_{J_n}$. Otherwise, the following holds:

$$x_{j'} w_{j'} + x_{J_n} w_{J_n} \leq -w_{j'} + w_{J_n} \leq 0 \leq w_{j'} - w_{J_n} \leq \hat{x}_{j'} w_{j'} + \hat{x}_{J_n} w_{J_n} \tag{27}$$

Note that every index in $j \in A_{K+1}$ satisfies Eq. (26) as well. Therefore, $\boldsymbol{x} \cdot \boldsymbol{w} \leq \hat{\boldsymbol{x}} \cdot \boldsymbol{w}$. We can conclude that:

$$0 < \boldsymbol{x} \cdot \boldsymbol{w} + b \leq \hat{\boldsymbol{x}} \cdot \boldsymbol{w} + b = -BT(\boldsymbol{w}) + b \tag{28}$$

which implies that $b > BT(\boldsymbol{w})$ as desired. □

C.3 AUXILIARY LEMMAS

We first define a special positive sample for every term. The special sample is a sample where all indices corresponding to the term will have positive values, and all other indexes will have negative values. The special samples will be used as the hardest positive samples to satisfy the MIN$_+$ property.

Definition C.4. *For a term $n \in [K]$, we define the special sample $\boldsymbol{x}^{(n)} \in \mathbb{S}_x$ of this term as follows:*

$$\forall j \in \mathbb{A}_n \;\; x_j^{(n)} = 1 \text{ and } \forall j \in [D] \backslash \mathbb{A}_n \;\; x_j^{(n)} = -1 \tag{29}$$

We denote the set of all the special samples by $\mathbb{O} = \left\{ \boldsymbol{x} \in \mathbb{S}_+ \mid \exists n \in [K] \;\; \boldsymbol{x} = \boldsymbol{x}^{(n)} \right\}$

Lemma C.3. *Given $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b})$, assume the following conditions are satisfied:*

1. $\forall i \in [r], \;\; \forall j \in [D] \quad \boldsymbol{w}_{ij} \geq 0$.
2. *For every $\boldsymbol{x} \in \mathbb{O}$ there exists $\mathbb{I} \subseteq [r]$ such that $\sum_{i \in \mathbb{I}} \boldsymbol{w}_i \cdot \boldsymbol{x} + b_i \geq 2$.*

Then $\boldsymbol{\theta}$ satisfies the MIN$_+$ property.

Proof. Let $\boldsymbol{x} \in \mathbb{S}_+$. Then $\exists n \in [K]$ such that $\forall j \in \mathbb{A}_n \;\; \boldsymbol{x}_j = 1$. By the second assumption, $\exists \mathbb{I} \subseteq [r]$ such that $\sum_{i \in \mathbb{I}} \boldsymbol{w}_i \cdot \boldsymbol{x}^{(n)} + b_i \geq 2$.

For every $i \in [r]$ the following holds:

$$\boldsymbol{w}_i \cdot \boldsymbol{x} = \sum_{j \in [D]} w_{ij} x_j = \sum_{j \in \mathbb{A}_n} w_{ij} + \sum_{j \in [D] \backslash \mathbb{A}_n} x_j w_{ij} \tag{30}$$

From the first condition of the claim we can deduce that

$$\sum_{j \in \mathbb{A}_n} w_{ij} + \sum_{j \in [D] \backslash \mathbb{A}_n} x_j w_{ij} \geq \sum_{j \in \mathbb{A}_n} w_{ij} - \sum_{j \in [D] \backslash \mathbb{A}_n} w_{ij} = \sum_{j \in [D]} w_{ij} x_j^{(n)} = \boldsymbol{w}_i \cdot \boldsymbol{x}^{(n)} \tag{31}$$

Then:

$$\sum_{i \in \mathbb{I}} \sigma(\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i) \geq \sum_{i \in \mathbb{I}} \sigma(\boldsymbol{w}_i \cdot \boldsymbol{x}^{(n)} + b_i) \geq 2 \tag{32}$$

and $\boldsymbol{\theta}$ satisfies the MIN$_+$ property for $\boldsymbol{x}$ as required. □

The following definition will be very useful in our analysis.

Definition C.5. *Given a min-norm solution $\boldsymbol{\theta}^* = (\boldsymbol{W}^*, \boldsymbol{b}^*)$, we say that the a solution $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{W}}, \hat{\boldsymbol{b}})$ is an i-modified solution if the following holds:*

$$\forall i' \in [r] \backslash \{i\} \;\; \hat{\boldsymbol{w}}_{i'} = \boldsymbol{w}^*_{i'} \text{ and } \hat{b}_{i'} = b^*_{i'} \tag{33}$$

Thus, given a min-norm solution, to define an i-modified solution, we only need to define the neuron $(\boldsymbol{w}_i, b_i)$.

Lemma C.4. *Given a min-norm solution $\boldsymbol{\theta}^* = (\boldsymbol{W}^*, \boldsymbol{b}^*)$, every $i \in [r]$ satisfies:*

1. $b^*_i = BT(\boldsymbol{w}^*_i)$.
2. $\forall j \in [D] \;\; \boldsymbol{w}^*_{ij} \geq 0$.
3. $\exists n \in [K]$ *such that* $\forall j \in \mathbb{A}_n \;\; \boldsymbol{w}^*_{ij} \geq 0$ *and* $\forall j \in [D] \backslash \mathbb{A}_n \;\; \boldsymbol{w}^*_{ij} = 0$.

Proof. **Property 1:** Assume by contradiction that $b^*_i \neq BT(\boldsymbol{w}^*_i)$. By Lemma C.2, b_i has to be smaller than $BT(\boldsymbol{w}^*_i)$, because otherwise $\boldsymbol{\theta}^*$ is not a perfect solution. Now consider the i-modified solution, $\hat{\boldsymbol{\theta}}$, which is defined by:

$$\hat{\boldsymbol{w}}_i = \boldsymbol{w}^*_i, \quad \hat{b}_i = BT(\hat{\boldsymbol{w}}) \tag{34}$$

By the assumption $\hat{b}_i > b^*_i$. Then, every $\boldsymbol{x} \in \mathbb{S}_x$ satisfies the following:

$$\boldsymbol{x} \cdot \boldsymbol{w}^*_i + b_i < \boldsymbol{x} \cdot \hat{\boldsymbol{w}}_i + \hat{b}_i \tag{35}$$

Since $\boldsymbol{\theta}^*$ satisfies the MIN$_+$ property, the above implies that $\hat{\boldsymbol{\theta}}$ satisfies it as well.

From Definition C.5 every $\boldsymbol{x} \in \mathbb{S}_-$ satisfies:

$$\forall i' \in [r] \backslash \{i\} \quad 0 > \boldsymbol{x} \cdot \boldsymbol{w}^*_{i'} + b_{i'} = \boldsymbol{x} \cdot \hat{\boldsymbol{w}}_{i'} + \hat{b}_{i'} \tag{36}$$

In addition, we repeat the proof of the second direction of Lemma C.2, and we get that $0 > \boldsymbol{x} \cdot \hat{\boldsymbol{w}}_i + \hat{b}_i$. Therefore, $\hat{\boldsymbol{\theta}}$ satisfies the MIN$_-$ property. By Lemma C.1, $\hat{\boldsymbol{\theta}}$ is a perfect solution. From Definition C.3 the bias threshold is nonpositive and therefore $b^*_i < \hat{b}_i \leq 0$ implies that $|\hat{b}_i| < |b^*_i|$. We know that $\hat{\boldsymbol{w}}_i = \boldsymbol{w}^*_i$ and therefore $||(\hat{\boldsymbol{w}}_i, \hat{b}_i)||_2^2 < ||(\boldsymbol{w}^*_i, b^*_i)||_2^2$ which contradicts the optimally of $\boldsymbol{\theta}^*$.

Property 2: Assume by contradiction that $\exists j' \in [D]$ such that $\boldsymbol{w}^*_{ij'} < 0$. Consider the following i-modified solution $\hat{\boldsymbol{\theta}}$:

$$\forall j \in [D] \backslash \{j'\} \;\; \hat{\boldsymbol{w}}_{ij} = \boldsymbol{w}^*_{ij} \;\; \wedge \;\; \hat{\boldsymbol{w}}_{ij'} = 0 \;\; \wedge \;\; \hat{b}_i = BT(\hat{\boldsymbol{w}}_i) \tag{37}$$

We want to show that:

$$\sum_{n \in [K]} V_n(\boldsymbol{w}^*_i) = \sum_{n \in [K]} V_n(\hat{\boldsymbol{w}}_i) \tag{38}$$

If $\exists n' \in [K]$ such that $j' \in \mathbb{A}_{n'}$, then it follows that $V_{n'}(\boldsymbol{w}^*_i) = V_{n'}(\hat{\boldsymbol{w}}_i) = 0$ and Eq. (38) is satisfied. Otherwise, $j' \in \mathbb{A}_{K+1}$, by Definition C.1, the indices of $\mathbb{A}_{K+1}$ don't affect the value of the sums in Eq. (38). Therefore, this equation is satisfied in this case as well.

We know that $b^*_i = BT(\boldsymbol{w}^*_i)$ according to Property 1 above, thus every $\boldsymbol{x} \in \mathbb{S}_x$ satisfies the following:

$$\begin{aligned} \boldsymbol{x} \cdot \boldsymbol{w}^*_i + b^*_i &= \sum_{j \in [D]} x_j w^*_{ij} + BT(\boldsymbol{w}^*_i) \\ &= \sum_{j \in [D] \backslash \{j'\}} x_j w^*_{ij} + x_{j'} w^*_{ij'} - |w^*_{ij'}| - \sum_{j \in [D] \backslash \{j'\}} |w^*_{ij}| + \sum_{n \in [K]} 2V_n(\boldsymbol{w}^*_i) \end{aligned} \tag{39}$$

We can see that $x_{j'} w^*_{ij'} - |w^*_{ij'}| \leq 0$. Then,

$$\begin{aligned} &\sum_{j \in [D] \backslash \{j'\}} x_j w^*_{ij} + x_{j'} w^*_{ij'} - |w^*_{ij'}| - \sum_{j \in [D] \backslash \{j'\}} |w^*_{ij}| + \sum_{n \in [K]} 2V_n(\boldsymbol{w}^*_i) \\ &\leq \sum_{j \in [D] \backslash \{j'\}} x_j w^*_{ij} - \sum_{j \in [D] \backslash \{j'\}} |w^*_{ij}| + \sum_{n \in [K]} 2V_n(\boldsymbol{w}^*_i) \\ &= \sum_{j \in [D]} x_j \hat{w}_{ij} - ||\hat{\boldsymbol{w}}_i||_1 + \sum_{n \in [K]} 2V_n(\hat{\boldsymbol{w}}_i) = \boldsymbol{x} \cdot \hat{\boldsymbol{w}}_i + BT(\hat{\boldsymbol{w}}_i) = \boldsymbol{x} \cdot \hat{\boldsymbol{w}}_i + \hat{b}_i \end{aligned} \tag{40}$$

Using the fact that $\boldsymbol{x} \cdot \boldsymbol{w}_i^* + b_i^* \leq \boldsymbol{x} \cdot \hat{\boldsymbol{w}}_i + \hat{b}_i$ with the fact that $\boldsymbol{\theta}^*$ satisfies the MIN$_+$ property, we can conclude that $\hat{\boldsymbol{\theta}}$ satisfies this property too.

According to Property 1 above and Definition C.5 we have:

$$\forall i' \in [r] \backslash \{i\} \;\; BT(\hat{\boldsymbol{w}}_{i'}) = BT(\boldsymbol{w}_{i'}^*) = b_{i'}^* = \hat{b}_{i'} \tag{41}$$

In addition, we know that $\hat{b}_i = BT(\hat{\boldsymbol{w}}_i)$ by Eq. (37). According to Lemma C.2, we know that $\hat{\boldsymbol{\theta}}$ is a perfect solution.

From Eq. (37), we know that $|w_{ij'}^*| > |\hat{w}_{ij'}| \rightarrow ||\boldsymbol{w}_i^*||_1 > ||\hat{\boldsymbol{w}}_i||_1$. Combining this with Eq. (38) and the fact that the bias threshold is nonpositive we can conclude that

$$-||\boldsymbol{w}_i^*||_1 + 2\sum_{n \in [K]} V_n(\boldsymbol{w}_i^*) < -||\hat{\boldsymbol{w}}_i||_1 + 2\sum_{n \in [K]} V_n(\hat{\boldsymbol{w}}_i) \rightarrow |BT(\boldsymbol{w}_i^*)| > |BT(\hat{\boldsymbol{w}}_i)| \rightarrow |b_i^*| > \left|\hat{b}_i\right| \tag{42}$$

Therefore, $||(\hat{\boldsymbol{w}}_i, \hat{b}_i)||_2^2 < ||(\boldsymbol{w}_i^*, b_i^*)||_2^2$ in contradiction to the optimality of θ^*.

Property 3: Assume by contradiction that there exists $i \in [r]$ such that:

$$\exists n_1 \neq n_2 \in [K+1] \text{ such that } \exists j \in \mathbb{A}_{n_1} \;\; w_{ij}^* > 0 \text{ and } \exists j \in \mathbb{A}_{n_2} \;\; w_{ij}^* > 0 \tag{43}$$

Without loss of generality we assume:

$$\sum_{j \in \mathbb{A}_{n_1}} w_{ij}^* \geq \sum_{j \in \mathbb{A}_{n_2}} w_{ij}^* \tag{44}$$

Let's look on the following i-modified $\hat{\boldsymbol{\theta}}$ which is defined by:

$$\forall j \in \mathbb{A}_{n_2} \;\; \hat{w}_{ij} = 0 \text{ and } \forall j \in [D] \backslash \mathbb{A}_{n_2} \;\; \hat{w}_{ij} = w_{ij}^* \text{ and } \hat{b}_i = BT(\hat{\boldsymbol{w}}_i) \tag{45}$$

First, we will show that $\hat{b}_i \geq b_i^*$. If $n_2 \neq K+1$, from Definition C.3 and the assumption that $|\mathbb{A}_{n_2}| > 1$, the following holds:

$$\hat{b}_i = BT(\hat{\boldsymbol{w}}_i) = BT(\boldsymbol{w}_i^*) + \sum_{j \in \mathbb{A}_{n_2}} |w_{ij}| - 2V_{n_2}(\boldsymbol{w}_i^*) = b_i^* + \sum_{j \in \mathbb{A}_{n_2}} \left|w_{ij}^*\right| - 2V_{n_2}(\boldsymbol{w}_i) \geq b_i^* \tag{46}$$

Otherwise $n_2 = K+1$ and from Definition C.3 the following holds:

$$\hat{b}_i = BT(\hat{\boldsymbol{w}}_i) = BT(\boldsymbol{w}_i^*) + \sum_{j \in \mathbb{A}_{n_2}} |w_{ij}| = b_i^* + \sum_{j \in \mathbb{A}_{n_2}} \left|w_{ij}^*\right| \geq b_i^* \tag{47}$$

In both cases $\hat{b}_i \geq b_i^*$ as required.

Given $\widetilde{n} \in [K]$, we know that $\boldsymbol{\theta}^*$ is a perfect solution and thus it satisfies the MIN$_+$ property. Then, for $\boldsymbol{x}^{(\widetilde{n})}$ there exists $\mathbb{I} \subseteq [r]$ such that:

$$\sum_{i' \in \mathbb{I}} \boldsymbol{w}_{i'}^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_{i'}^* \geq 2 \tag{48}$$

We will show that there exists $\mathbb{I}' \subseteq [r]$ such that:

$$\sum_{i' \in \mathbb{I}'} \hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x}^{(\widetilde{n})} + \hat{b}_{i'} \geq 2 \tag{49}$$

Recall, by Definition C.5, for any $i' \in [r] \backslash \{i\}$ we know that $\hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x}^{(\widetilde{n})} + \hat{b}_{i'} = \boldsymbol{w}_{i'}^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_{i'}^*$.

If $\widetilde{n} \neq n_2$, due to Property 1 and Property 2 above and the fact that $\hat{b}_i \geq b_i^*$ the following holds:

$$\begin{aligned} \boldsymbol{w}_i^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_i^* &= \sum_{j \in [D] \backslash \mathbb{A}_{n_2}} w_{ij}^* x_j^{(\widetilde{n})} - \sum_{j \in \mathbb{A}_{n_2}} w_i^* + b_i^* < \sum_{j \in [D] \backslash \mathbb{A}_{n_2}} w_{ij}^* x_j^{(\widetilde{n})} + b_i^* \leq \sum_{j \in [D]} \hat{w}_{ij} x_j^{(\widetilde{n})} + \hat{b}_i \\ &= \hat{\boldsymbol{w}}_i \cdot \boldsymbol{x}^{(\widetilde{n})} + \hat{b}_i \end{aligned} \tag{50}$$

Therefore,

$$\sum_{i' \in \mathbb{I}} \hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x}^{(\widetilde{n})} + \hat{b}_{i'} \geq \sum_{i' \in \mathbb{I}} \boldsymbol{w}_{i'}^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_{i'}^* \geq 2 \tag{51}$$

Otherwise, $\widetilde{n} = n_2$. By the fact that $b_i^* = BT(\boldsymbol{w}_i^*) \leq 0$, Property 2 above and Eq. (44) the following holds:

$$\boldsymbol{w}_i^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_i^* = \sum_{j \in \mathbb{A}_{n_2}} w_{ij}^* - \sum_{j \in [D] \backslash \mathbb{A}_{n_2}} w_{ij}^* + b_i^* \leq \sum_{j \in \mathbb{A}_{n_2}} w_{ij}^* - \sum_{j \in \mathbb{A}_{n_1}} w_{ij}^* + b_i^* \leq \sum_{j \in \mathbb{A}_{n_2}} w_{ij}^* - \sum_{j \in \mathbb{A}_{n_1}} w_{ij}^* \leq 0 \quad (52)$$

Therefore, using Definition C.5, $\hat{\boldsymbol{\theta}}$ satisfies the following:

$$\sum_{i' \in \mathbb{I} \backslash \{i\}} \hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x}^{(\widetilde{n})} + \hat{b}_{i'} \geq \sum_{i' \in \mathbb{I} \backslash \{i\}} \boldsymbol{w}_{i'}^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_{i'}^* \geq \sum_{i' \in \mathbb{I}} \boldsymbol{w}_{i'}^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_{i'}^* \geq 2 \quad (53)$$

We can conclude that $\hat{\boldsymbol{\theta}}$ satisfies Eq. (49). Combining this with Property 2 above, we can see that $\hat{\boldsymbol{\theta}}$ meets the condition of Lemma C.3 and then it satisfies the MIN$_+$ property.

According to Property 1 above and Definition C.5:

$$\forall i' \in [r] \backslash \{i\} \;\; BT(\hat{\boldsymbol{w}}_{i'}) = BT(\boldsymbol{w}_{i'}^*) = b_{i'}^* = \hat{b}_{i'} \quad (54)$$

In addition, we know that $\hat{b}_i = BT(\hat{\boldsymbol{w}}_i)$ by Eq. (45). According to Lemma C.2, the solution $\hat{\boldsymbol{\theta}}$ is a perfect solution.

As we saw $0 \geq \hat{b}_i \geq b_i^* \rightarrow \left|\hat{b}_i\right| \leq |b_i^*|$, $\forall j \in \mathbb{A}_{n_2} \;\; w_{ij}^* \geq 0 = \hat{w}_{ij}$ and $\exists j \in \mathbb{A}_{n_2} \;\; w_{ij}^* > 0 = \hat{w}_{ij}$ and therefore $||(\boldsymbol{w}_i^*, b_i^*)||_2^2 > ||(\hat{\boldsymbol{w}}_i, \hat{b}_i)||_2^2$ which contradicts the optimality of $\boldsymbol{\theta}^*$. □

C.4 ALIGNMENT LEMMAS

The following three lemmas show the alignment properties of the min-norm solution.

Lemma C.5. *Given a min-norm solution $\boldsymbol{\theta}^* = (\boldsymbol{W}^*, \boldsymbol{b}^*)$, every $i \in [r]$ either aligns with some term $n \in [K]$ or it holds that $\boldsymbol{w}_i^* = \boldsymbol{0}, b_i^* = 0$*

Proof. Given $i \in [r]$, if $\boldsymbol{w}_i^* = \boldsymbol{0}$ the claim is true by Property 1 of Lemma C.4. Otherwise, by Property 3 of Lemma C.4, $\exists n \in [K]$ such that:

$$\forall j \in \mathbb{A}_n \;\; \boldsymbol{w}_{ij}^* \geq 0 \text{ and } \exists j \in \mathbb{A}_n \;\; \boldsymbol{w}_{ij}^* > 0 \text{ and } \forall j \in [D] \backslash \mathbb{A}_n \;\; \boldsymbol{w}_{ij}^* = 0 \quad (55)$$

We will prove the claim for n by assuming by contradiction that:

$$\exists j_1, j_2 \in \mathbb{A}_n \text{ such that } w_{ij_1}^* \neq w_{ij_2}^* \quad (56)$$

Without loss of generality, we assume $w_{ij_1}^* > w_{ij_2}^*$ and $w_{ij_2}^* = \min_{j \in \mathbb{A}_n} \{w_{ij}^*\}$.

Define the following i-modified solution $\hat{\boldsymbol{\theta}}$:

$$\forall j \in \mathbb{A}_n \;\; \hat{w}_{ij} = w_{ij_2}^* \text{ and } \forall j \in [D] \backslash \mathbb{A}_n \;\; \hat{w}_{ij} = w_{ij}^* \text{ and } \hat{b}_i = BT(\hat{\boldsymbol{w}}_i) \quad (57)$$

Note that $V_n(\boldsymbol{w}_i^*) = V_n(\hat{\boldsymbol{w}}_i) = \boldsymbol{w}_{ij_2}^*$.

Given $\boldsymbol{x} \in \mathbb{S}_+$, we know that $\boldsymbol{\theta}^*$ satisfies the MIN$_+$ property. Then, $\exists I \subseteq [r]$ such that:

$$\sum_{i' \in \mathbb{I}} \boldsymbol{w}_{i'}^* \cdot \boldsymbol{x} + b_{i'}^* \geq 2 \quad (58)$$

Recall, by Definition C.5, for any $i' \in [r] \backslash \{i\}$ we know that $\hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x} + \hat{b}_{i'} = \boldsymbol{w}_{i'}^* \cdot \boldsymbol{x} + b_{i'}^*$

If $\boldsymbol{x} \cdot \boldsymbol{t}_n^* = ||\boldsymbol{t}_n^*||_1$, due to Property 3 of Lemma C.4 the following holds:

$$\begin{aligned} \boldsymbol{w}_i^* \cdot \boldsymbol{x} + b_i^* &= \sum_{j \in \mathbb{A}_n} w_{ij}^* + BT(\boldsymbol{w}_i^*) = \sum_{j \in \mathbb{A}_n} w_{ij}^* - \sum_{j \in \mathbb{A}_n} \left|w_{ij}^*\right| + 2V_n(\boldsymbol{w}_i^*) = 2\boldsymbol{w}_{ij_2}^* \\ &= \sum_{j \in \mathbb{A}_n} \hat{w}_{ij} - \sum_{j \in \mathbb{A}_n} \hat{w}_{ij} + 2w_{ij_2}^* = \sum_{j \in \mathbb{A}_n} \hat{w}_{ij} - ||\hat{\boldsymbol{w}}||_1 + 2V_n(\hat{\boldsymbol{w}}_i) = \hat{\boldsymbol{w}}_i \cdot \boldsymbol{x} + \hat{b}_i \end{aligned} \quad (59)$$

Then we can conclude:

$$\sum_{i' \in \mathbb{I}} \hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x} + \hat{b}_{i'} = \sum_{i' \in \mathbb{I}} \boldsymbol{w}^*_{i'} \cdot \boldsymbol{x} + b^*_{i'} \geq 2 \tag{60}$$

Otherwise, $\boldsymbol{x} \cdot \boldsymbol{t}^*_n < ||\boldsymbol{t}^*_n||_1$ and by Definition C.3:

$$\boldsymbol{w}^*_i \cdot \boldsymbol{x} + b_i = \sum_{j \in \mathbb{A}_n} w^*_{ij} x_j + BT(\boldsymbol{w}^*_i) \leq \sum_{j \in \mathbb{A}_n \setminus \{j_2\}} w^*_{ij} - w^*_{ij_2} - \sum_{j \in \mathbb{A}_n \setminus \{j_2\}} w^*_{ij} - w^*_{ij_2} + 2\boldsymbol{w}^*_{ij_2} = 0 \tag{61}$$

Therefore,

$$\sum_{i' \in \mathbb{I} \setminus \{i\}} \hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x} + \hat{b}_{i'} = \sum_{i' \in \mathbb{I} \setminus \{i\}} \boldsymbol{w}^*_{i'} \cdot \boldsymbol{x} + b^*_{i'} \geq \sum_{i' \in \mathbb{I}} \boldsymbol{w}^*_{i'} \cdot \boldsymbol{x} + b^*_{i'} \geq 2 \tag{62}$$

We can conclude that $\hat{\boldsymbol{\theta}}$ satisfies the MIN$_+$ property.

According to Property 1 of Lemma C.4 and Definition C.5:

$$\forall i' \in [r] \setminus \{i\} \;\; BT(\hat{\boldsymbol{w}}_{i'}) = BT(\boldsymbol{w}^*_{i'}) = b^*_{i'} = \hat{b}_{i'} \tag{63}$$

In addition, we know that $\hat{b}_i = BT(\hat{\boldsymbol{w}}_i)$ by Eq. (57). According to Lemma C.2, the solution $\hat{\boldsymbol{\theta}}$ is a perfect solution.

Finally, we can see that, $||\boldsymbol{w}^*_i||_1 > ||\hat{\boldsymbol{w}}_i||_1$ implies that $|BT(\boldsymbol{w}^*_i)| > |BT(\hat{\boldsymbol{w}}_i)|$ and $\forall j \in [D] : \; w^*_{ij} \geq \hat{w}_{ij}$. Thus, we have $||(\boldsymbol{w}^*_i, b^*_i)||_2^2 > ||(\hat{\boldsymbol{w}}_i, \hat{b}_i)||_2^2$. This is in contradiction to the optimally of $\boldsymbol{\theta}^*$, as desired.

We can now define $\lambda_i = V_n(\boldsymbol{w}^*_i)$ and we know that the neuron i satisfies:

$$\forall j \in \mathbb{A}_n \;\; w^*_{ij} = \lambda_i \text{ and } \forall j \in [D] \setminus \mathbb{A}_n \;\; w^*_{ij} = 0 \tag{64}$$

Therefore, $\boldsymbol{w}^*_i = \lambda_i \boldsymbol{t}^*_n, b^*_i = \lambda_i(2 - ||t^*_n||_1)$ and we can say that neuron i aligns the term n. □

Lemma C.6. *Given a min-norm solution $\boldsymbol{\theta}^* = (\boldsymbol{W}^*, \boldsymbol{b}^*)$, every 2 neurons $i_1, i_2 \in [r]$ that align with term $n \in [K]$ satisfy $\lambda_{i_1} = \lambda_{i_2}$.*[7]

Proof. Given $i_1, i_2 \in [r]$ that align with term $n \in [K]$ we have $\boldsymbol{w}^*_{i_1} = \lambda_{i_1} t^*_n$ and $\boldsymbol{w}^*_{i_2} = \lambda_{i_2} t^*_n$. Assume by contradiction that $\lambda_{i_2} \neq \lambda_{i_1}$. Define the following solution $\hat{\boldsymbol{\theta}}$:

$$\begin{aligned} \forall i \in [r] \setminus \{i_1, i_2\} \;\; \hat{\boldsymbol{w}}_i &= \boldsymbol{w}^*, \; \hat{b}_i = b^*_i \\ \hat{\boldsymbol{w}}_{i_1} &= \frac{\lambda_{i_1} + \lambda_{i_2}}{2} \boldsymbol{t}^*_n, \;\; \hat{b}_{i_1} = BT(\hat{\boldsymbol{w}}_{i_1}) \\ \hat{\boldsymbol{w}}_{i_2} &= \frac{\lambda_{i_1} + \lambda_{i_2}}{2} \boldsymbol{t}^*_n, \;\; \hat{b}_{i_2} = BT(\hat{\boldsymbol{w}}_{i_2}) \end{aligned} \tag{65}$$

Note, $\forall i' \in [r] \setminus \{i_1, i_2\}$ we know that $\hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x} + \hat{b}_{i'} = \boldsymbol{w}^*_{i'} \cdot \boldsymbol{x} + b^*_{i'}$

Given $\boldsymbol{x} \in \mathbb{S}_+$, we know that $\boldsymbol{\theta}^*$ satisfies the MIN$_+$ property. Then, $\exists \mathbb{I} \subseteq [r]$ such that:

$$\sum_{i' \in \mathbb{I}} \boldsymbol{w}^*_{i'} \cdot \boldsymbol{x} + b^*_{i'} \geq 2 \tag{66}$$

If $\boldsymbol{x} \cdot \boldsymbol{t}^*_n = ||\boldsymbol{t}^*_n||_1$ we can calculate the following:

$$\boldsymbol{w}^*_{i_1} \cdot \boldsymbol{x} + b^*_{i_1} + \boldsymbol{w}^*_{i_2} \cdot \boldsymbol{x} + b^*_{i_2} = 2\lambda_{i_1} + 2\lambda_{i_2} = (\lambda_{i_2} + \lambda_{i_2}) + (\lambda_{i_2} + \lambda_{i_2}) = \hat{\boldsymbol{w}}_{i_1} \cdot \boldsymbol{x} + \hat{b}^*_{i_1} + \hat{\boldsymbol{w}}_{i_2} \cdot \boldsymbol{x} + \hat{b}^*_{i_2} \tag{67}$$

Therefore,

$$\sum_{i' \in \mathbb{I} \cup \{i_1, i_2\}} \hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x} + \hat{b}^*_{i'} = \sum_{i' \in \mathbb{I} \cup \{i_1, i_2\}} \boldsymbol{w}^*_{i'} \cdot \boldsymbol{x} + b^*_{i'} \geq \sum_{i' \in \mathbb{I}} \boldsymbol{w}^*_{i'} \cdot \boldsymbol{x} + b^*_{i'} \geq 2 \tag{68}$$

[7] λ_i were defined in the proof of the previous lemma.

Otherwise, $\boldsymbol{x} \cdot \boldsymbol{t}_n^* < ||\boldsymbol{t}_n^*||_1$ thus $\boldsymbol{w}_{i_1}^* \cdot \boldsymbol{x} + b_{i_1}^* \leq 0$, $\boldsymbol{w}_{i_2}^* \cdot \boldsymbol{x} + b_{i_2}^* \leq 0$. Then,

$$\sum_{i' \in \mathbb{I} \backslash \{i_1, i_2\}} \hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x} + \hat{b}_{i'}^* = \sum_{i' \in \mathbb{I} \backslash \{i_1, i_2\}} \boldsymbol{w}_{i'}^* \cdot \boldsymbol{x} + b_{i'}^* \geq \sum_{i' \in \mathbb{I}} \boldsymbol{w}_{i'}^* \cdot \boldsymbol{x} + b_{i'}^* \geq 2 \tag{69}$$

Combining this with Property 2 of Lemma C.4 we can see that $\hat{\boldsymbol{\theta}}$ meets the condition of Lemma C.3 and then it satisfies the MIN_+ property.

According to Property 1 of Lemma C.4:

$$\forall i' \in [r] \backslash \{i_1, i_2\} \;\; BT(\hat{\boldsymbol{w}}_{i'}) = BT(\boldsymbol{w}_{i'}^*) = b_{i'}^* = \hat{b}_{i'} \tag{70}$$

In addition, we know that $\hat{b}_{i_1} = BT(\hat{\boldsymbol{w}}_{i_1})$ and $\hat{b}_{i_2} = BT(\hat{\boldsymbol{w}}_{i_2})$ by Eq. (65). According to Lemma C.2, the solution $\hat{\boldsymbol{\theta}}$ is a perfect solution.

We will prove that $\sum_{i \in [r]} ||(\boldsymbol{w}_i^*, b_i^*)||_2^2 > \sum_{i \in [r]} ||(\hat{\boldsymbol{w}}_i, \hat{b}_i)||_2^2$. This will contradict the optimality of $(\boldsymbol{w}_i^*, b_i^*)$. Note that $||(\boldsymbol{w}_{i_1}^*, b_{i_1}^*)||_2^2 = \lambda_{i_1}^2 |\mathbb{A}_n| + \lambda_{i_1}^2 (|\mathbb{A}_n| - 2)^2$. Then:

$$\begin{aligned}
\sum_{i \in [r]} ||(\boldsymbol{w}_i^*, b_i^*)||_2^2 - \sum_{i \in [r]} ||(\hat{\boldsymbol{w}}_i, \hat{b}_i)||_2^2 &= ||(\boldsymbol{w}_{i_1}^*, b_{i_1}^*)||_2^2 + ||(\boldsymbol{w}_{i_2}^*, b_{i_2}^*)||_2^2 - ||(\hat{\boldsymbol{w}}_{i_1}, \hat{b}_{i_1})||_2^2 - ||(\hat{\boldsymbol{w}}_{i_2}, \hat{b}_{i_2})||_2^2 \\
&= \left(\lambda_{i_1}^2 + \lambda_{i_2}^2 - 2 \left(\frac{\lambda_{i_1} + \lambda_{i_2}}{2} \right)^2 \right) \left(|\mathbb{A}_n| + (|\mathbb{A}_n| - 2)^2 \right) \\
&= \left(\frac{1}{2} \lambda_{i_1}^2 - \lambda_{i_1} \lambda_{i_2} + \frac{1}{2} \lambda_{i_2}^2 \right) \left(|\mathbb{A}_n| + (|\mathbb{A}_n| - 2)^2 \right) \\
&= \left(\frac{1}{\sqrt{2}} \lambda_{i_1} - \frac{1}{\sqrt{2}} \lambda_{i_2} \right)^2 \left(|\mathbb{A}_n| + (|\mathbb{A}_n| - 2)^2 \right)
\end{aligned}$$

For $\lambda_{i_1} \neq \lambda_{i_2}$, we get $\sum_{i \in [r]} ||(\boldsymbol{w}_i^*, b_i^*)||_2^2 - \sum_{i \in [r]} ||(\hat{\boldsymbol{w}}_i, \hat{b}_i)||_2^2 > 0$, as needed. □

Lemma C.7. *Given a min-norm solution $\boldsymbol{\theta}^* = (\boldsymbol{W}^*, \boldsymbol{b}^*)$, if $\mathbb{I} \subseteq [r]$ is the set of all neurons that align with term $n \in [K]$, then $\sum_{i \in \mathbb{S}} \lambda_i = 1$.*

Proof. Assume by contradiction that $\sum_{i \in \mathbb{I}} \lambda_i \neq 1$. If $\sum_{i \in \mathbb{I}} \lambda_i < 1$, then $\forall i' \in [r] \backslash \mathbb{I}$, by Lemma C.5 we know that the neuron i' aligns with another term or is equal to 0, then $\boldsymbol{x}^{(n)} \cdot \boldsymbol{w}_i^* \leq 0$. By Property 1 of Lemma C.4, $b_{i'}^* = BT(\boldsymbol{w}_{i'}^*) \leq 0$. Therefore, $\boldsymbol{x}^{(n)} \cdot \boldsymbol{w}_{i'}^* + b_{i'}^* \leq 0$. Then, for every $\mathbb{I}' \subseteq [r]$ the following holds:

$$\sum_{i \in \mathbb{I}'} \boldsymbol{w}_i^* \cdot \boldsymbol{x}^{(n)} + b_i^* \leq \sum_{i \in \mathbb{I}' \bigcap \mathbb{I}} \boldsymbol{w}_i^* \cdot \boldsymbol{x}^{(n)} + b_i^* = \sum_{i \in \mathbb{I}' \bigcap \mathbb{I}} |\mathbb{A}_n| \lambda_i + (2 - |\mathbb{A}_n|) \lambda_i = 2 \sum_{i \in \mathbb{I}' \bigcap \mathbb{I}} \lambda_i < 2 \tag{71}$$

and thus $\boldsymbol{\theta}^*$ doesn't satisfy the MIN_+ property. By Lemma C.1, this contradicts the fact that $\boldsymbol{\theta}^*$ is a perfect solution.

If $\sum_{i \in \mathbb{I}} \lambda_i > 1$, we choose an arbitrarily $\hat{i} \in \mathbb{I}$. Define $\hat{\boldsymbol{\theta}}$ as follows:

$$\begin{aligned}
\forall i \in [r] \backslash \mathbb{I} \;\; \hat{\boldsymbol{w}}_i &= \boldsymbol{w}_i^*, \;\; \hat{b}_i = b_i^* \\
\forall i \in \mathbb{I} \backslash \{\hat{i}\} \;\; \hat{\boldsymbol{w}}_i &= 0, \;\; \hat{b}_i = 0 \\
\hat{\boldsymbol{w}}_{\hat{i}} &= \boldsymbol{t}_n^*, \;\; \hat{b}_{\hat{i}} = BT(\hat{\boldsymbol{w}}_{\hat{i}})
\end{aligned} \tag{72}$$

Given $\boldsymbol{x}^{(\widetilde{n})} \in \mathbb{O}$, we know that $\boldsymbol{\theta}^*$ satisfies the MIN_+. Thus, $\exists \mathbb{I}' \subseteq [r]$ such that:

$$\sum_{i \in \mathbb{I}'} \boldsymbol{w}_{i'}^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_i^* \geq 2 \tag{73}$$

We will show that there exists $\hat{\mathbb{I}} \subseteq [r]$ such that:

$$\sum_{i' \in \hat{\mathbb{I}}} \hat{\boldsymbol{w}}_{i'} \cdot \boldsymbol{x}^{(\widetilde{n})} + \hat{b}_{i'} \geq 2 \tag{74}$$

If $\widetilde{n} = n$, then:

$$\hat{\boldsymbol{w}}_{\hat{i}} \cdot \boldsymbol{x}^{(\widetilde{n})} + \hat{b}_{\hat{i}} = \boldsymbol{t}_n^* \cdot \boldsymbol{x}^{(\widetilde{n})} + BT(\boldsymbol{t}_n^*) = ||\boldsymbol{t}_n^*||_1 - ||\boldsymbol{t}_n^*||_1 + 2 = 2 \tag{75}$$

Therefore, by choosing $\hat{\mathbb{I}} = \{\hat{i}\}$, we can show that $\hat{\boldsymbol{\theta}}$ satisfies Eq. (74).

Otherwise, $\widetilde{n} \neq n$ and then every $i \in \mathbb{I}$ satisfies:

$$\boldsymbol{w}_i^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_i^* = -\lambda_i * |\mathbb{A}_n| - \lambda_i(|\mathbb{A}_n| - 2) < 0 \tag{76}$$

From Eq. (72) we can conclude:

$$\sum_{i \in \mathbb{I}' \setminus \mathbb{I}} \hat{\boldsymbol{w}}_i \cdot \boldsymbol{x}^{(\widetilde{n})} + \hat{b}_i = \sum_{i \in \mathbb{I}' \setminus \mathbb{I}} \boldsymbol{w}_i^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_i^* \geq \sum_{i \in \mathbb{I}'} \boldsymbol{w}_i^* \cdot \boldsymbol{x}^{(\widetilde{n})} + b_i^* \geq 2 \tag{77}$$

Therefore, $\hat{\boldsymbol{\theta}}$ satisfies Eq. (74) for $\boldsymbol{x}^{(\widetilde{n})}$. In addition, by Property 2 of Lemma C.4 and Eq. (72), we know that all the weights of the neurons in $\hat{\boldsymbol{\theta}}$ are nonnegative. Then, $\hat{\boldsymbol{\theta}}$ meets the condition of Lemma C.3 and it satisfies the MIN_+ property.

According to Property 1 of Lemma C.4 and Eq. (72):

$$\forall i \in [r] \setminus \mathbb{I} \;\; BT(\hat{\boldsymbol{w}}_i) = BT(\boldsymbol{w}_i^*) = b_i^* = \hat{b}_i \tag{78}$$

In addition, we know that $\forall i \in \mathbb{I} \setminus \{\hat{i}\} \quad \hat{b}_i = 0 = BT(\hat{\boldsymbol{w}}_i)$ and $\hat{b}_{\hat{i}} = BT(\hat{\boldsymbol{w}}_{\hat{i}})$. Therefore, according to Lemma C.2, the solution $\boldsymbol{\theta}$ is a perfect solution.

We will prove that $\sum_{i \in [r]} ||(\boldsymbol{w}_i^*, b_i^*)||_2^2 > \sum_{i \in [r]} ||(\hat{\boldsymbol{w}}_i, \hat{b}_i)||_2^2$. This will contradict the optimality of $\boldsymbol{\theta}^*$. Indeed:

$$\begin{aligned} \sum_{i \in [r]} ||(\boldsymbol{w}_i^*, b_i^*)||_2^2 - \sum_{i \in [r]} ||(\hat{\boldsymbol{w}}_i, \hat{b}_i)||_2^2 = \sum_{i \in I} ||(\boldsymbol{w}_i^*, b_i^*)||_2^2 - ||(\hat{\boldsymbol{w}}_{\hat{i}}, \hat{b}_{\hat{i}})||_2^2 = \\ \sum_{i \in I} \lambda_i^2 |\mathbb{A}_n| + \lambda_i^2 (|\mathbb{A}_n| - 2)^2 - |\mathbb{A}_n| - (|\mathbb{A}_n| - 2)^2 = \\ \left(\sum_{i \in I} \lambda_i^2 - 1 \right) \left(|\mathbb{A}_n| + (|\mathbb{A}_n| - 2)^2 \right) \end{aligned} \tag{79}$$

Since $\sum_{i \in \mathbb{I}} \lambda_i > 1$, we have $\sum_{i \in [r]} ||(\boldsymbol{w}_i^*, b_i^*)||_2^2 - \sum_{i \in [r]} ||(\hat{\boldsymbol{w}}_i, \hat{b}_i)||_2^2 > 0$, which completes the proof.

□

C.5 FINISHING THE PROOF OF THEOREM 6.1

Proof. Given a min-norm solution $\boldsymbol{\theta}^* = (\boldsymbol{W}^*, \boldsymbol{b}^*)$, by Lemma C.5, each neuron $i \in [r]$ aligns with some term $n_i \in [K]$ or it is equal to 0. Assume by contradiction that there exists a term $n \in [K]$ that is not aligned, namely $\forall i \in [r] \;\; n_i \neq n$. Consider the special positive sample $\boldsymbol{x}^{(n)} \in \mathbb{S}_+$. From the definition of $\boldsymbol{x}^{(n)}$, every $j \in [D] \setminus \mathbb{A}_n$ satisfies $\boldsymbol{x}_j^{(n)} = -1$. Then,

$$\forall i \in [r] \quad \boldsymbol{x}^{(n)} \cdot \boldsymbol{w}_i^* = \lambda_n \boldsymbol{x}^{(n)} \cdot \boldsymbol{t}_{n_i}^* = -\lambda_n ||\boldsymbol{t}_{n_i}^*||_1 < 0 \tag{80}$$

By the first property of Lemma C.5, $\forall i \in [r], \; b_i^* = BT(\boldsymbol{w}_i^*) \leq 0$. Therefore, $\forall i \in [r] : \; \boldsymbol{x}^{(n)} \cdot \boldsymbol{w}_i^* + b_i^* < 0$, in contradiction to the fact that $\boldsymbol{\theta}^*$ is a perfect solution.

In conclusion, every term $n \in [K]$ aligns with a set of neurons $\mathbb{I} \subseteq [r]$. By Lemma C.7, we know that $\sum_{i \in \mathbb{I}} \lambda_i = 1$. By Lemma C.6, we know that every two neurons i_1, i_2 that align with the same term satisfy $\lambda_{i_1} = \lambda_{i_2}$. Therefore, $\boldsymbol{\theta}^*$ is a DNF recovery solution. □

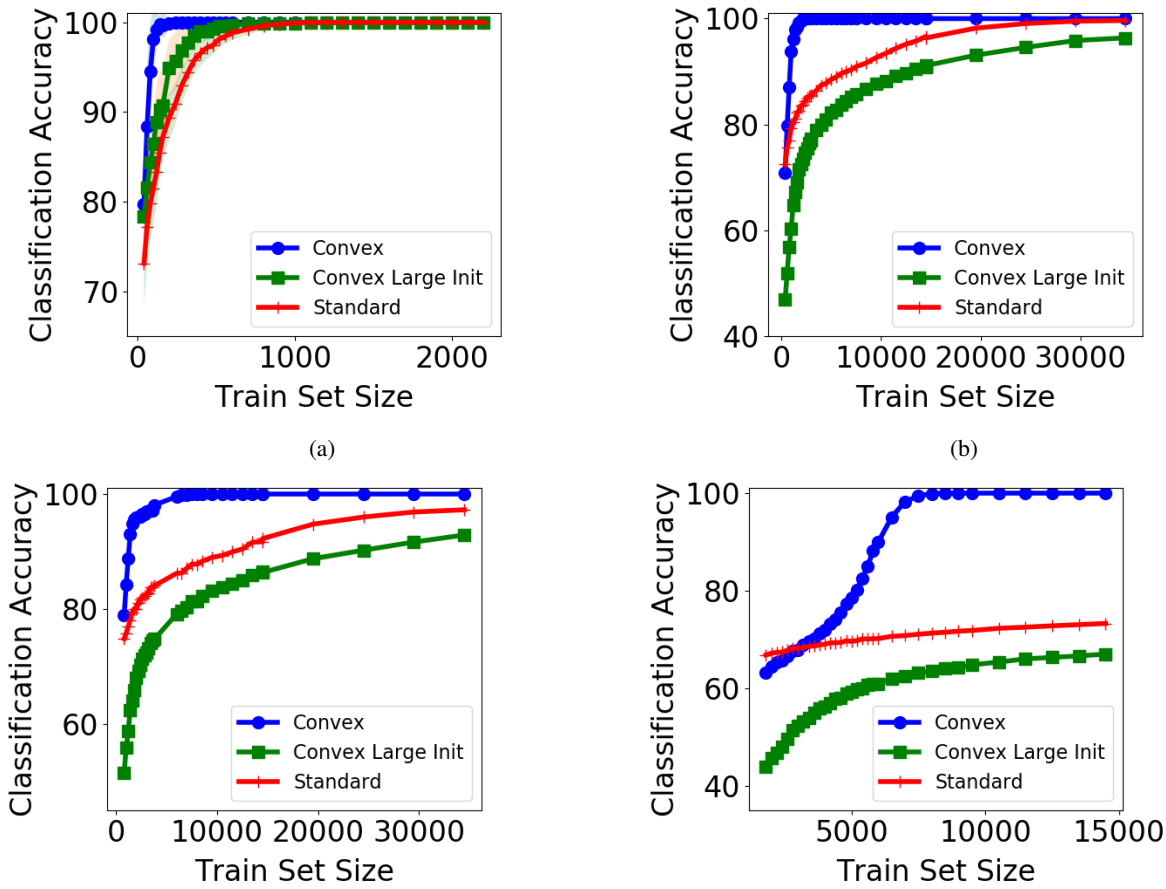


Figure 5: Test accuracy for the convex network with small initialization, convex network with large initialization and standard networks. Figure (a) shows the performance when learning f_1^*, Figure (b) for f_3^*, Figure (c) for f_4^* and Figure (d) for f_3^* (Results for f_2^* were presented in the main paper).

D EXPERIMENT DETAILS AND ADDITIONAL RESULTS

Selected read-once DNFs: In this section we provide additional experiments for the following types of read-once DNFs.

- f_1^* - 3-term read-once DNF where the length of every term is 3 for $D = 9$
- f_2^* - 4-term read-once DNF where the length of the terms are 4,5,5,6 for $D = 25$
- f_3^* - 8-term read-once DNF where the length of the terms are 3,3,4,4,4,4,5,5 for $D = 40$
- f_4^* - 10-term read-once DNF where the length of the terms are 3,3,4,4,4,4,6,6,6,6 for $D = 50$
- f_5^* - 15-term read-once DNF where of every term is 5 for $D = 100$

General details: In all the experiments, "small initialization" refers to initializing weights from $\boldsymbol{W}^{(0)} \sim \mathcal{N}(0, 10^{-6})$ and $\boldsymbol{b}^{(0)} = [0]^D$. The learning rate for SGD is $\eta = 10^{-3}$, the number of hidden units is $r = 2000$ and the batch size is 32. We create the train set by sampling uniformly from $[\pm 1]^D$. All the experiments can run on any single GPU. Training a single network can take up to two hours.

Weight Matrix Visualization: When presenting weight matrices, we first cluster the neurons using the Hierarchical clustering algorithm.[8] We then plot the weight values in an image, where neurons clustered together appear in consecutive

[8] We used scipy.cluster.hierarchy.linkage. with centroid as a method.

rows. Note this of course does not change the model itself, but makes it easy to see if there are well clustered neurons (as in the DNF recovery case.

Sample Complexity Experiments: We evaluate test accuracy as a function of the training sample size for different models. Results are shown in Figure 5. Specifically, we compare the convex network with small initialization (see details above), the convex network with large initialization (we take $\boldsymbol{W}^{(0)} \sim \mathcal{N}(0,1)$), and a "standard" network with one hidden layer, same width as the convex network and Xavier initialization (we checked different initialization schemes, including small Gaussian initialization, and verified that this not affect the results). We run every experiment 10 times and present the mean performance and the std of this mean (the std is small and smaller than the line width).

Implementation of the Statistical Queries (SQ) Methods: In Figure 2a of the main paper, we present results of the statistical query algorithm. We implemented the algorithm described in Mansour and Schain [2001]. We view the ϵ therein as a hyperparameter. Therefore, we use 10% of the train set as validation for finding ϵ. We present the performance of the SQ algorithm only for $D = 9$, because for larger dimension the algorithm failed in creating a DNF for the range of train set sizes tested.

Learned models for different training sizes: In the main paper, we show empirically that learning convex networks with small init and GD leads to a DNF recovery solution, and we also show formally that in the population risk DNF recovery is norm minimizing. Here we show explicit model weights for different training sizes, demonstrating that approximate DNF recovery solutions are obtained for fairly small sample sizes. Figures 6,7,8,9,10 (panels a-c) show these results. In panel d of these figures we show the learned model for when learning with large Gaussian initialization and with the same train set size. It can be seen that larger initialization does not results in the recovery-DNF solution (note we are also visualizing these solutions using clustering as explained above, and there is clearly no cluster structure in the solution).

DNF reconstruction: In Figure 2b of the main paper we present accuracy results for DNF reconstruction. To obtain these, we take the learned model and apply a simple rounding procedure to obtain a DNF. Specifically, we calculate the ℓ_∞ norm of each weight vector and round all weight values above a threshold to this value, and set the others to zero. This threshold is a hyper-parameter that we validate against the test set. We then set all non-zero values to 1. This results in a $\{0,1\}$ matrix that can be viewed as a DNF. We compare this DNF to the true underlying read-once DNF, and if the learned and ground-truth DNF are different in even one variable, we count this as an error. The figure reports the average accuracy of this procedure across 100 training sessions. Full implementation is provided in the attached code.

The effect of learning c: Figure 11 shows that fixing the learnable parameter c to -1 does not effect the structure of the solution. In this experiment, we took 2 networks with the same width: One with learnable c initialized to 0, and the second with fixed $c = -1$. We initialize the other weights with the same values, and train them with the same train set for the same number of steps. Finally we plot the solution that the network learns. We can say they both recover the underlying DNF.

Tabular datasets: We consider the three UCI datasets: kr-vs-kp, Splice, and diabetes. For these, we convert the input into binary by changing categorical variables to one-hot. We also consider binary classification such that in kr-vs-kp the class 'won' is positive considered and 'notwon' is negative, in Splice the classes 'EI' and 'IE' are considered positive and 'N' negative, and diabetes is binary by design. We train on 90% of the data and test on 10%.

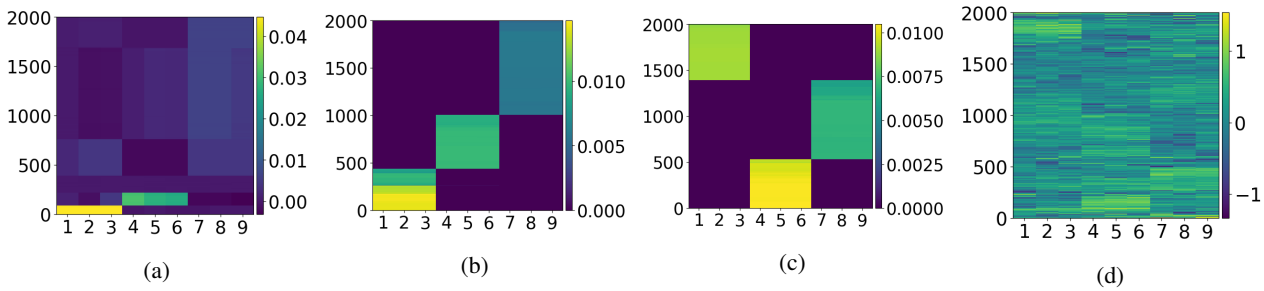


Figure 6: (a-c) Effect of training size on the learned model (see **Learned models for different training sizes** in the text) for the ground-truth model f_1^*. Panels a-c correspond to training sizes 800,1500 and 7500. (d) Result for training on 7500 with large initialization.

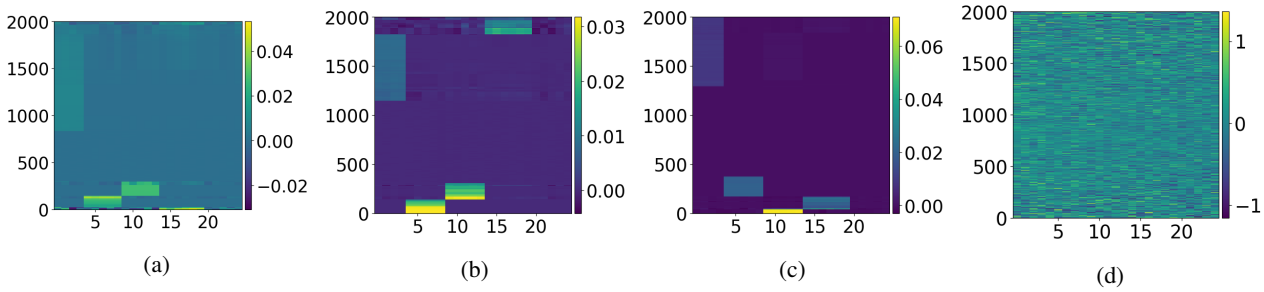


Figure 7: Same as Figure 6 but with f_2^* as ground truth.

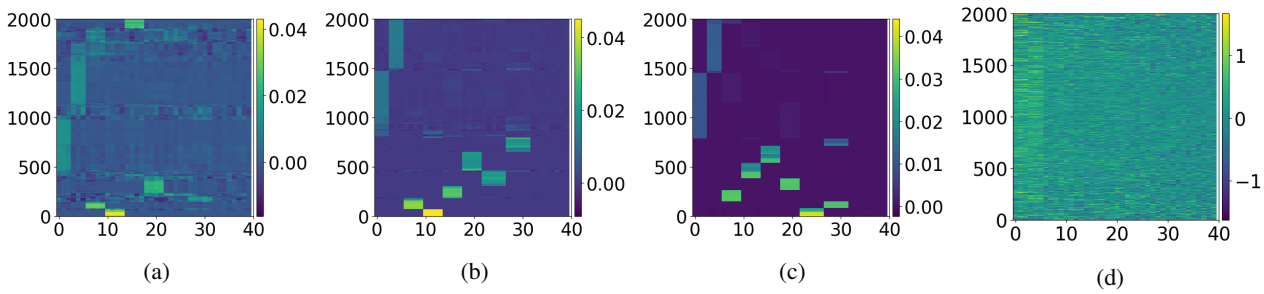


Figure 8: Same as Figure 6 but with f_3^* as ground truth.

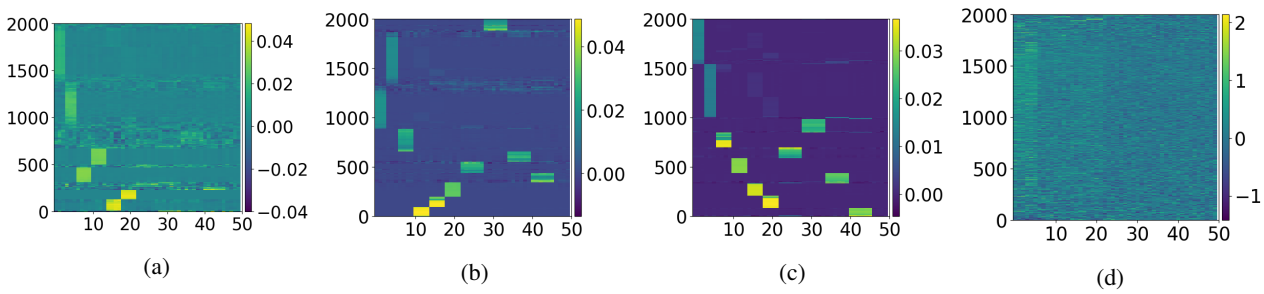


Figure 9: Same as Figure 6 but with f_4^* as ground truth.

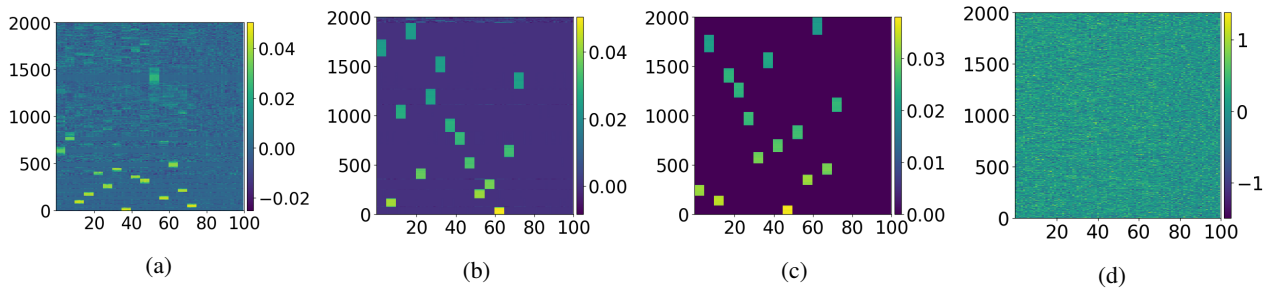


Figure 10: Same as Figure 6 but with f_5^* as ground truth.

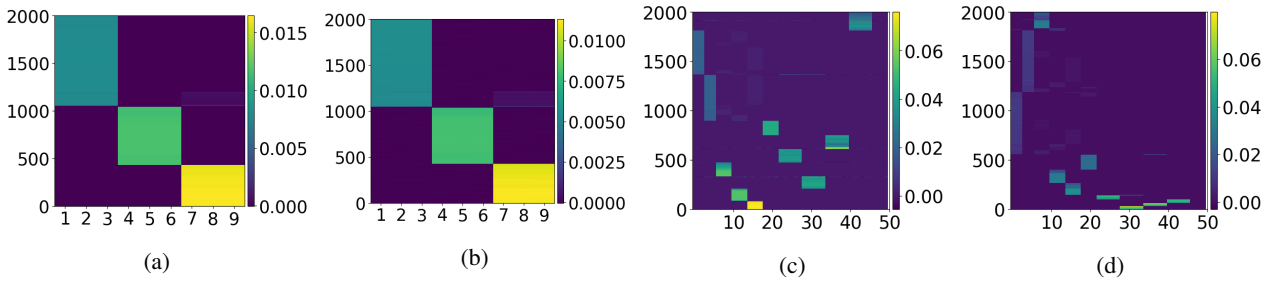


Figure 11: (a) Learning f_1^* using convex network with learnable c and 250 train samples. (b) Learning f_1^* using convex network with fix c to -1 and 250 train samples (a) Learning f_4^* using convex network with learnable c and 30,000 train samples. (b) Learning f_4^* using convex network with fix c to -1 and 30,000 train samples