Revealing the Structure of Deep Neural Networks via Convex Duality

Tolga Ergen [1] **Mert Pilanci** [1]

Abstract

We study regularized deep neural networks (DNNs) and introduce a convex analytic framework to characterize the structure of the hidden layers. We show that a set of optimal hidden layer weights for a norm regularized DNN training problem can be explicitly found as the extreme points of a convex set. For the special case of deep linear networks, we prove that each optimal weight matrix aligns with the previous layers via duality. More importantly, we apply the same characterization to deep ReLU networks with whitened data and prove the same weight alignment holds. As a corollary, we also prove that norm regularized deep ReLU networks yield spline interpolation for one-dimensional datasets which was previously known only for two-layer networks. Furthermore, we provide closed-form solutions for the optimal layer weights when data is rank-one or whitened. The same analysis also applies to architectures with batch normalization even for arbitrary data. Therefore, we obtain a complete explanation for a recent empirical observation termed Neural Collapse where class means collapse to the vertices of a simplex equiangular tight frame.

1. Introduction

Deep neural networks (DNNs) have become extremely popular due to their success in machine learning applications. Even though DNNs are highly over-parameterized and non-convex, simple first-order algorithms, e.g., Stochastic Gradient Descent (SGD), can be used to successfully train them. Moreover, recent work has shown that highly over-parameterized networks trained with SGD obtain simple solutions that generalize well (Savarese et al., 2019; Parhi & Nowak, 2019; Ergen & Pilanci, 2020a;b), where two-layer ReLU networks with the minimum Euclidean norm solution and zero training error are proven to fit a linear spline model in 1D regression. In addition, a recent series of work (Pilanci & Ergen, 2020; Ergen & Pilanci, 2021; Sahiner et al., 2021; Gupta et al., 2021) showed that regularized two-layer ReLU network training problems exhibit a convex loss landscape in a higher dimensional space, which was previously attributed to the benign impacts of overparameterization (Brutzkus et al., 2017; Li & Liang, 2018; Du et al., 2018b; Ergen & Pilanci, 2019). Therefore, regularizing the solution towards smaller norm weights might be the key to understand the generalization properties and loss landscape of DNNs. However, analyzing DNNs is still theoretically elusive even in the absence of nonlinear activations. To this end, we study norm regularized DNNs and develop a framework based on convex duality to characterize a set of optimal solutions to the training problem.

Deep linear networks have been the subject of extensive theoretical analysis due to their tractability. A line of research (Saxe et al., 2013; Arora et al., 2018a; Laurent & Brecht, 2018; Du & Hu, 2019; Shamir, 2018) focused on GD training dynamics, however, they lack the analysis of solution set and generalization properties of deep networks. Another line of research (Gunasekar et al., 2017; Arora et al., 2019; Bhojanapalli et al., 2016) studied the generalization properties via matrix factorization and showed that linear networks trained with GD converge to minimum nuclear norm solutions. Later on, (Arora et al., 2018b; Du et al., 2018a) showed that gradient flow enforces the layer weights to align. (Ji & Telgarsky, 2019) further proved that each layer weight matrix is asymptotically rank-one. These results provide insights to characterize the structure of the optimal layer weights, however, they require multiple strong assumptions, e.g., linearly separable training data and strictly decreasing loss function, which makes the results impractical. Furthermore, (Zhang et al., 2019) provided some characterizations for nonstandard networks, which are valid for hinge loss with an uncommon regularization. Unlike these studies, we introduce a complete characterization for regularized deep network training problems without requiring such assumptions.

[1] Department of Electrical Engineering, Stanford University, CA, USA. Correspondence to: Tolga Ergen <ergen@stanford.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

1.1. Our contributions

Our contributions can be summarized as follows

- We introduce a convex analytic framework that characterizes a set of optimal solutions to regularized training problems as the extreme points of a convex set.
- For deep linear networks, we prove that each optimal layer weight matrix aligns with the previous layers via convex duality.
- For deep ReLU networks, we obtain the same weight alignment result for whitened or rank-one data matrices. As a corollary, we achieve closed-form solutions for the optimal hidden layer weights when the data is whitened or rank-one (see Theorem 4.1 and 4.3).
- As another corollary, we prove that the optimal regularized ReLU networks are linear spline interpolators for one-dimensional, i.e., rank-one, data which generalizes the two-layer results for one-dimensional data in (Savarese et al., 2019; Parhi & Nowak, 2019; Ergen & Pilanci, 2020a;b) to arbitrary depth.
- We show that whitening/rank-one assumptions can be removed by placing batch normalization in between layers (see Theorem 4.4). Hence, our results explain a recent empirical observation, termed Neural Collapse (Papyan et al., 2020), where class means collapse to the vertices of a simplex equiangular tight frame (see Corollary 4.3).

1.2. Overview of our results

Notation: We denote matrices/vectors as uppercase/lowercase bold letters. We use $\mathbf{0}_k$ (or $\mathbf{1}_k$) and $\mathbf{I}_k$ to denote a vector of zeros (or ones) and the identity matrix of size $k \times k$, respectively. We denote the set of integers from 1 to n as $[n]$. To denote Frobenius, operator, and nuclear norms, we use $\|\cdot\|_F$, $\|\cdot\|_2$, and $\|\cdot\|_*$, respectively. We also use **tr** to denote the trace of a matrix. Furthermore, $\sigma_{max}(\cdot)$ and $\sigma_{min}(\cdot)$ represent the maximum and minimum singular values, respectively and the unit ℓ_2-ball $\mathcal{B}_2$ is defined as $\mathcal{B}_2 = \{\mathbf{u} \in \mathbb{R}^d \,|\, \|\mathbf{u}\|_2 \leq 1\}$. We also provide further explanations about our notation in Table 2 in Appendix.

We consider an L-layer network with layer weights $\mathbf{W}_{l,j} \in \mathbb{R}^{m_{l-1} \times m_l}$ and $\mathbf{w}_L \in \mathbb{R}^m$, $\forall l \in [L], \forall j \in [m]$, where $m_0 = d$ and $m_{L-1} = 1$, respectively. Then, given a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, the output is $f_{\theta,L}(\mathbf{X}) = \mathbf{A}_{L-1}\mathbf{w}_L,\ \mathbf{A}_{l,j} = g(\mathbf{A}_{l-1,j}\mathbf{W}_{l,j})\ \forall l \in [L-1]$, where $\mathbf{A}_{0,j} = \mathbf{X}$, $\mathbf{A}_{L-1} \in \mathbb{R}^{n \times m}$, and $g(\cdot)$ is the activation function. Given labels $\mathbf{y} \in \mathbb{R}^n$, the training problem is as follows

$$\min_{\{\theta_l\}_{l=1}^L} \mathcal{L}(f_{\theta,L}(\mathbf{X}), \mathbf{y}) + \beta \mathcal{R}(\theta)\,, \tag{1}$$

where $\mathcal{L}(\cdot,\cdot)$ is an arbitrary loss function, $\mathcal{R}(\theta)$ is regularization for the layer weights, $\beta > 0$ is a regularization parameter, $\theta_l = \{\{\mathbf{W}_{l,j}\}_{j=1}^m, m_l\}$, and $\theta = \{\theta_l\}_{l=1}^L$. In the paper, for the sake of presentation simplicity, we illustrate the conventional training setup with squared loss and ℓ_2^2-norm regularization. However, our analysis is valid for arbitrary convex loss functions as proven in Appendix A.1. Thus, we consider the following optimization problem

$$P^* = \min_{\{\theta_l\}_{l=1}^L} \mathcal{L}(f_{\theta,L}(\mathbf{X}), \mathbf{y}) + \frac{\beta}{2} \sum_{j=1}^m \sum_{l=1}^L \|\mathbf{W}_{l,j}\|_F^2. \tag{2}$$

Next, we show that the minimum ℓ_2^2-norm is equivalent to minimum ℓ_1-norm after a rescaling.

Lemma 1.1. *The following problems are equivalent :*

$$\begin{aligned} &\min_{\{\theta_l\}_{l=1}^L} \mathcal{L}(f_{\theta,L}(\mathbf{X}), \mathbf{y}) + \frac{\beta}{2} \sum_{j=1}^m \sum_{l=1}^L \|\mathbf{W}_{l,j}\|_F^2 \\ &= \min_{\{\theta_l\}_{l=1}^L, \{t_j\}_{j=1}^m} \mathcal{L}(f_{\theta,L}(\mathbf{X}), \mathbf{y}) + \beta \|\mathbf{w}_L\|_1 \\ &\qquad\qquad + \frac{\beta}{2}(L-2) \sum_{j-1}^m t_j^2. \\ &\textit{s.t. } \mathbf{w}_{L-1,j} \in \mathcal{B}_2, \|\mathbf{W}_{l,j}\|_F \leq t_j,\ \forall l \in [L-2] \end{aligned}$$

Using Lemma 1.1[1], we first take the dual with respect to the output layer weights $\mathbf{w}_L$ and then change the order of min-max to achieve the following dual as a lower bound [2]

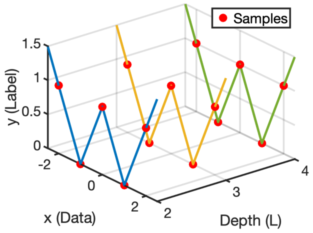
$$\begin{aligned} P^* \geq D^* = &\min_{\{t_j\}_{j=1}^m} \max_{\boldsymbol{\lambda}} -\mathcal{L}^*(\boldsymbol{\lambda}) + \frac{\beta}{2}(L-2) \sum_{j-1}^m t_j^2 \\ &\text{s.t.} \max_{\substack{\mathbf{w}_{L-1,j} \in \mathcal{B}_2 \\ \|\mathbf{W}_{l,j}\|_F \leq t_j}} \|\mathbf{A}_{L-1,j}^T \boldsymbol{\lambda}\|_\infty \leq \beta\,. \end{aligned} \tag{3}$$

To the best of our knowledge, the dual DNN characterization (3) is novel. Using this result, we first characterize a set of weights that minimize the objective via the optimality conditions and active constraints in (3). We then prove the optimality of these weights by proving strong duality, i.e., $P^* = D^*$, for DNNs. We then show that, for deep linear networks, optimal weight matrices align with the previous layers.

More importantly, the same analysis and conclusions also apply to deep ReLU networks when the input is whitened and/or rank-one. Here, we even obtain closed-form solutions for the optimal layer weights. As a corollary, we show that deep ReLU networks fit a linear spline interpolation when the input is one-dimensional. We also provide an experiment in Figure 1 to verify this claim. Note that

[1] The proof is presented in Appendix A.3.

[2] For the definitions and details see Appendix A.1 and A.2.



	Width (m)	Assumption	Depth (L)	# of outputs (K)
(Savarese et al., 2019)	∞	1D data ($d = 1$)	2	✗ ($K = 1$)
(Parhi & Nowak, 2019)	∞	1D data ($d = 1$)	2	✗ ($K = 1$)
(Ergen & Pilanci, 2020a;b)	finite	rank-one/whitened	2	✓ ($K \geq 1$)
Our results	finite	rank-one/whitened or BatchNorm	$L \geq 2$	✓ ($K \geq 1$)

Figure 1 & Table 1: One dimensional interpolation using L-layer ReLU networks with 20 neurons in each hidden layer. As predicted by Corollary 4.2, the optimal solution is given by piecewise linear splines for any $L \geq 2$. Additionally, we provide a comparison with previous studies about this characterization.

this result was previously known only for two-layer networks (Savarese et al., 2019; Parhi & Nowak, 2019; Ergen & Pilanci, 2020a;b) and here we extend it to arbitrary depth L (see Table 1 for details). We also show that the whitened/rank-one assumption can be removed by introducing batch normalization in between layers, which reflects the training setup in practice.

2. Warmup: Two-layer linear networks

As a warmup, we first consider the simple case of two-layer linear networks with the output $f_{\theta,2}(\mathbf{X}) = \mathbf{X}\mathbf{W}_1\mathbf{w}_2$ and the parameters as $\theta \in \Theta = \{(\mathbf{W}_1, \mathbf{w}_2, m) \,|\, \mathbf{W}_1 \in \mathbb{R}^{d\times m}, \mathbf{w}_2 \in \mathbb{R}^m, m \in \mathbb{Z}_+\}$. Motivated by recent results (Neyshabur et al., 2014; Savarese et al., 2019; Parhi & Nowak, 2019; Ergen & Pilanci, 2020a;b), we first focus on a minimum norm[3] variant of (1) with squared loss, which can be written as

$$\min_{\theta\in\Theta} \|\mathbf{W}_1\|_F^2 + \|\mathbf{w}_2\|_2^2 \text{ s.t. } f_{\theta,2}(\mathbf{X}) = \mathbf{y}. \tag{4}$$

Using Lemma A.1[4], we equivalently have

$$P^* = \min_{\theta\in\Theta} \|\mathbf{w}_2\|_1 \text{ s.t. } f_{\theta,2}(\mathbf{X}) = \mathbf{y}, \mathbf{w}_{1,j} \in \mathcal{B}_2, \forall j, \tag{5}$$

which has the following dual form.

Theorem 2.1. *The dual of the problem in* (5) *is given by*

$$P^* \geq D^* = \max_{\boldsymbol{\lambda}\in\mathbb{R}^n} \boldsymbol{\lambda}^T\mathbf{y} \text{ s.t. } \max_{\mathbf{w}_1\in\mathcal{B}_2} \left|\boldsymbol{\lambda}^T\mathbf{X}\mathbf{w}_1\right| \leq 1\,. \tag{6}$$

For (5), $\exists m^* \leq n+1$ *such that strong duality holds, i.e.,* $P^* = D^*$, $\forall m \geq m^*$ *and* $\mathbf{W}_1^*$ *satisfies* $\|(\mathbf{X}\mathbf{W}_1^*)^T\boldsymbol{\lambda}^*\|_\infty = 1\,,$ *where* $\boldsymbol{\lambda}^*$ *is the dual optimal parameter.*

Using Theorem 2.1, we now characterize the optimal neurons as the extreme points of a convex set.

Corollary 2.1. *By Theorem 2.1, the optimal neurons are extreme points which solve* $\operatorname{argmax}_{\mathbf{w}_1\in\mathcal{B}_2} |{\boldsymbol{\lambda}^*}^T\mathbf{X}\mathbf{w}_1|$.

Definition 1. We call the maximizers of the constraint in Corollary 2.1 *extreme points* throughout the paper.

From Theorem 2.1, we have the following dual problem

$$\max_{\boldsymbol{\lambda}} \boldsymbol{\lambda}^T\mathbf{y} \text{ s.t. } \max_{\mathbf{w}_1\in\mathcal{B}_2} |\boldsymbol{\lambda}^T\mathbf{X}\mathbf{w}_1| \leq 1. \tag{7}$$

Let $\mathbf{X} = \mathbf{U}_x\boldsymbol{\Sigma}_x\mathbf{V}_x^T$ be the singular value decomposition (SVD) of $\mathbf{X}$[5]. If we assume that there exists $\mathbf{w}^*$ such that $\mathbf{X}\mathbf{w}^* = \mathbf{y}$ due to Proposition 2.1, then (7) is equivalent to

$$\max_{\tilde{\boldsymbol{\lambda}}} \tilde{\boldsymbol{\lambda}}^T\boldsymbol{\Sigma}_x\tilde{\mathbf{w}}^* \text{ s.t. } \|\boldsymbol{\Sigma}_x^T\tilde{\boldsymbol{\lambda}}\|_2 \leq 1, \tag{8}$$

where $\tilde{\boldsymbol{\lambda}} = \mathbf{U}_x^T\boldsymbol{\lambda}$, $\tilde{\mathbf{w}}^* = \mathbf{V}_x^T\mathbf{w}^*$, and we changed the constraint since the extreme point is achieved when $\mathbf{w}_1 = \mathbf{X}^T\boldsymbol{\lambda}/\|\mathbf{X}^T\boldsymbol{\lambda}\|_2$. Given $\operatorname{rank}(\mathbf{X}) = r$, we have

$$\tilde{\boldsymbol{\lambda}}^T\boldsymbol{\Sigma}_x\tilde{\mathbf{w}}^* = \tilde{\boldsymbol{\lambda}}^T\boldsymbol{\Sigma}_x \underbrace{\begin{bmatrix}\mathbf{I}_r & \mathbf{0}_{r\times d-r}\\ \mathbf{0}_{d-r\times r} & \mathbf{0}_{d-r\times d-r}\end{bmatrix}\tilde{\mathbf{w}}^*}_{\tilde{\mathbf{w}}_r^*}$$

$$\leq \|\boldsymbol{\Sigma}_x^T\tilde{\boldsymbol{\lambda}}\|_2\|\tilde{\mathbf{w}}_r^*\|_2 \leq \|\tilde{\mathbf{w}}_r^*\|_2, \tag{9}$$

which shows that the maximum objective value is achieved when $\boldsymbol{\Sigma}_x^T\tilde{\boldsymbol{\lambda}} = c_1\tilde{\mathbf{w}}_r^*$. Thus, we have

$$\mathbf{w}_1^* = \frac{\mathbf{V}_x\boldsymbol{\Sigma}_x^T\tilde{\boldsymbol{\lambda}}}{\|\mathbf{V}_x\boldsymbol{\Sigma}_x^T\tilde{\boldsymbol{\lambda}}\|_2} = \frac{\mathbf{V}_x\tilde{\mathbf{w}}_r^*}{\|\tilde{\mathbf{w}}_r^*\|_2} = \frac{\mathcal{P}_{\mathbf{X}^T}(\mathbf{w}^*)}{\|\mathcal{P}_{\mathbf{X}^T}(\mathbf{w}^*)\|_2},$$

where $\mathcal{P}_{\mathbf{X}^T}(\cdot)$ projects its input onto the range of $\mathbf{X}^T$. In the sequel, we first show that one can consider a planted model without loss of generality and then prove strong duality for (5).

Proposition 2.1. *[(Du & Hu, 2019)] Given* $\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2$, *we have*

$$\operatorname*{argmin}_{\mathbf{W}_1,\mathbf{w}_2} \|\mathbf{X}\mathbf{W}_1\mathbf{w}_2 - \mathbf{X}\mathbf{w}^*\|_2^2 = \operatorname*{argmin}_{\mathbf{W}_1,\mathbf{w}_2} \|\mathbf{X}\mathbf{W}_1\mathbf{w}_2 - \mathbf{y}\|_2^2.$$

Theorem 2.2. *Let* $\{\mathbf{X}, \mathbf{y}\}$ *be feasible for* (5), *then strong duality holds for finite width networks.*

[3] This corresponds to weak regularization, i.e., $\beta \to 0$ in (1) (see e.g. (Wei et al., 2018).).

[4] All the equivalence lemmas are presented in Appendix A.3.

[5] In this paper, we use full SVD unless otherwise stated.

2.1. Regularized training problem

In this section, we define the regularized version of (5) as

$$\min_{\theta\in\Theta} \frac{1}{2}\|f_{\theta,2}(\mathbf{X})-\mathbf{y}\|_2^2+\beta\|\mathbf{w}_2\|_1 \text{ s.t. } \mathbf{w}_{1,j}\in\mathcal{B}_2, \quad (10)$$

which has the following dual form

$$\max_{\boldsymbol{\lambda}} -\frac{1}{2}\|\boldsymbol{\lambda}-\mathbf{y}\|_2^2+\frac{1}{2}\|\mathbf{y}\|_2^2 \text{ s.t. } \max_{\mathbf{w}_1\in\mathcal{B}_2}|\boldsymbol{\lambda}^T\mathbf{X}\mathbf{w}_1|\leq\beta.$$

Then, an optimal neuron needs to satisfy the condition

$$\mathbf{w}_1^*=\frac{\mathbf{X}^T\mathcal{P}_{\mathbf{X},\beta}(\mathbf{y})}{\|\mathbf{X}^T\mathcal{P}_{\mathbf{X},\beta}(\mathbf{y})\|_2}$$

where $\mathcal{P}_{\mathbf{X},\beta}(\cdot)$ projects to $\{\mathbf{u}\in\mathbb{R}^n \mid \|\mathbf{X}^T\mathbf{u}\|_2\leq\beta\}$. We now prove strong duality.

Theorem 2.3. *Strong duality holds for* (10) *with finite width.*

2.2. Training problem with vector outputs

Here, our model is $f_{\theta,2}(\mathbf{X})=\mathbf{X}\mathbf{W}_1\mathbf{W}_2$ to estimate $\mathbf{Y}\in\mathbb{R}^{n\times K}$, which can be optimized as follows

$$\min_{\theta\in\Theta}\|\mathbf{W}_1\|_F^2+\|\mathbf{W}_2\|_F^2 \text{ s.t. } f_{\theta,2}(\mathbf{X})=\mathbf{Y}. \quad (11)$$

Using Lemma A.2, we reformulate (11) as

$$\min_{\theta\in\Theta}\sum_{j=1}^m\|\mathbf{w}_{2,j}\|_2 \text{ s.t. } f_{\theta,2}(\mathbf{X})=\mathbf{Y},\mathbf{w}_{1,j}\in\mathcal{B}_2, \quad (12)$$

which has the following dual with respect to $\mathbf{W}_2$

$$\max_{\boldsymbol{\Lambda}}\mathbf{tr}(\boldsymbol{\Lambda}^T\mathbf{Y}) \text{ s.t. } \|\boldsymbol{\Lambda}^T\mathbf{X}\mathbf{w}_1\|_2\leq 1,\ \forall\mathbf{w}_1\in\mathcal{B}_2. \quad (13)$$

Since we can assume $\mathbf{Y}=\mathbf{X}\mathbf{W}^*$ due to Proposition 2.1,

$$\begin{aligned}\mathbf{tr}(\boldsymbol{\Lambda}^T\mathbf{Y})&=\mathbf{tr}(\boldsymbol{\Lambda}^T\mathbf{X}\mathbf{W}^*)=\mathbf{tr}(\boldsymbol{\Lambda}\mathbf{U}_x\boldsymbol{\Sigma}_x\tilde{\mathbf{W}}_r^*)\\&\leq\sigma_{max}(\boldsymbol{\Lambda}^T\mathbf{U}_x\boldsymbol{\Sigma}_x)\left\|\tilde{\mathbf{W}}_r^*\right\|_*\leq\|\tilde{\mathbf{W}}_r^*\|_* \quad (14)\end{aligned}$$

where $\sigma_{max}(\boldsymbol{\Lambda}^T\mathbf{X})\leq 1$ due to (13) and $\tilde{\mathbf{W}}_r^*=\begin{bmatrix}\mathbf{I}_r & \mathbf{0}_{r\times d-r}\\ \mathbf{0}_{d-r\times r} & \mathbf{0}_{d-r\times d-r}\end{bmatrix}\mathbf{V}_x^T\mathbf{W}^*$. Given the SVD of $\tilde{\mathbf{W}}_r^*$, i.e., $\mathbf{U}_w\boldsymbol{\Sigma}_w\mathbf{V}_w^T$, choosing

$$\boldsymbol{\Lambda}^T\mathbf{U}_x\boldsymbol{\Sigma}_x=\mathbf{V}_w\begin{bmatrix}\mathbf{I}_{r_w} & \mathbf{0}_{r_w\times d-r_w}\\ \mathbf{0}_{K-r_w\times r_w} & \mathbf{0}_{K-r_w\times d-r_w}\end{bmatrix}\mathbf{U}_w^T$$

achieves the upper-bound above, where $r_w=\operatorname{rank}(\tilde{\mathbf{W}}_r^*)$. Thus, optimal neurons are a subset of the first r_w right singular vectors of $\boldsymbol{\Lambda}^T\mathbf{X}$. We next prove strong duality.

Theorem 2.4. *Let* $\{\mathbf{X},\mathbf{Y}\}$ *be feasible for* (12)*, then strong duality holds for finite width networks.*

2.2.1. Regularized case

Here, we define the regularized version of (12) as follows

$$\min_{\theta\in\Theta}\frac{1}{2}\|f_{\theta,2}(\mathbf{X})-\mathbf{Y}\|_F^2+\beta\sum_{j=1}^m\|\mathbf{w}_{2,j}\|_2 \text{ s.t. } \mathbf{w}_{1,j}\in\mathcal{B}_2,$$

which has the following dual with respect to $\mathbf{W}_2$

$$\max_{\boldsymbol{\Lambda}}-\frac{1}{2}\|\boldsymbol{\Lambda}-\mathbf{Y}\|_F^2+\frac{1}{2}\|\mathbf{Y}\|_F^2 \text{ s.t. } \sigma_{max}(\boldsymbol{\Lambda}^T\mathbf{X})\leq\beta.$$

Then, the optimal neurons are a subset of the maximal right singular vectors of $\mathcal{P}_{\mathbf{X},\beta}(\mathbf{Y})^T\mathbf{X}$, where $\mathcal{P}_{\mathbf{X},\beta}(\cdot)$ projects its input to the set $\{\mathbf{U}\in\mathbb{R}^{n\times K}\mid\sigma_{max}(\mathbf{U}^T\mathbf{X})\leq\beta\}$.

Remark 2.1. *Note that the optimal neurons are the right singular vectors of* $\mathcal{P}_{\mathbf{X},\beta}(\mathbf{Y})^T\mathbf{X}$ *that achieve* $\|\mathcal{P}_{\mathbf{X},\beta}(\mathbf{Y})^T\mathbf{X}\mathbf{w}_1^*\|_2=\beta$*, where* $\|\mathbf{w}_1^*\|_2=1$*. This implies that* $\|\mathbf{Y}^T\mathbf{X}\mathbf{w}_1^*\|_2\geq\beta$*, therefore, the number of optimal neurons and rank*$(\mathbf{W}_1^*)$ *are determined by* β*.*

Remark 2.2. *The right singular vectors of* $\mathcal{P}_{\mathbf{X},\beta}(\mathbf{Y})^T\mathbf{X}$ *are not the only solutions. Consider* $\mathbf{u}_1$ *and* $\mathbf{u}_2$ *as the optimal right singular vectors. Then,* $\mathbf{u}=\alpha_1\mathbf{u}_1+\alpha_2\mathbf{u}_2$ *with* $\alpha_1^2+\alpha_2^2=1$ *also achieves the upper-bound, thus, optimal.*

3. Deep linear networks[6]

We now consider an L-layer linear network with the output function $f_{\theta,L}(\mathbf{X})=\sum_{j=1}^m\mathbf{X}\mathbf{W}_{1,j}\ldots w_{L,j}$, and the training problem

$$\min_{\{\theta_l\}_{l=1}^L}\frac{1}{2}\sum_{j=1}^m\sum_{l=1}^L\|\mathbf{W}_{l,j}\|_F^2 \text{ s.t. } f_{\theta,L}(\mathbf{X})=\mathbf{y}. \quad (15)$$

Proposition 3.1. *First* $L-2$ *layer weight matrices in* (15) *have the same operator and Frobenius norms, i.e.,* $t_j=\|\mathbf{W}_{l,j}\|_F=\|\mathbf{W}_{l,j}\|_2,\forall l\in[L-2],\ \forall j\in[m]$.

This result shows that the layer weights obey an alignment condition. After using the scaling in Lemma A.3 and the same convex duality arguments, a set of optimal solutions to the training problem can be described as follows.

Theorem 3.1. *Optimal layer weights for* (15) *are*

$$\mathbf{W}_{l,j}^*=\begin{cases}t_j^*\frac{\mathbf{V}_x\tilde{\mathbf{w}}_r^*}{\|\tilde{\mathbf{w}}_r^*\|_2}\boldsymbol{\rho}_{1,j}^T & \text{if } l=1\\ t_j^*\boldsymbol{\rho}_{l-1,j}\boldsymbol{\rho}_{l,j}^T & \text{if } 1<l\leq L-2\\ \boldsymbol{\rho}_{L-2,j} & \text{if } l=L-1\end{cases},$$

where $\boldsymbol{\rho}_{l,j}\in\mathbb{R}^{m_l}$ *such that* $\|\boldsymbol{\rho}_{l,j}\|_2=1$, $\forall l\in[L-2]$, $\forall j\in[m]$ *and* $\tilde{\mathbf{w}}_r^*$ *is defined in* (9).

Next, we prove strong duality holds.

[6] Since the derivations are similar, we present the details in Appendix A.7.

Theorem 3.2. *Let $\{\mathbf{X}, \mathbf{y}\}$ be feasible for* (15)*, then strong duality holds for finite width networks.*

Corollary 3.1. *Theorem 3.1 implies that deep linear networks can obtain a scaled version of $\mathbf{y}$ using only the first layer, i.e., $\mathbf{X}\mathbf{W}_1\boldsymbol{\rho}_1 = c\mathbf{y}$, where $c > 0$. Therefore, the remaining layers do not contribute to the expressive power of the network.*

3.1. Regularized training problem

We now present the regularized training problem as follows

$$\min_{\{\theta_l\}_{l=1}^L} \frac{1}{2}\|f_{\theta,L}(\mathbf{X}) - \mathbf{y}\|_2^2 + \frac{\beta}{2}\sum_{j=1}^m\sum_{l=1}^L \|\mathbf{W}_{l,j}\|_F^2. \quad (16)$$

Next result provides a set of optimal solutions to (16).

Theorem 3.3. *Optimal layer weights for* (16) *are*

$$\mathbf{W}_{l,j}^* = \begin{cases} t_j^* \frac{\mathbf{X}^T\mathcal{P}_{\mathbf{X},\beta}(\mathbf{y})}{\|\mathbf{X}^T\mathcal{P}_{\mathbf{X},\beta}(\mathbf{y})\|_2}\boldsymbol{\rho}_{1,j}^T & \text{if } l = 1 \\ t_j^*\boldsymbol{\rho}_{l-1,j}\boldsymbol{\rho}_{l,j}^T & \text{if } 1 < l \leq L-2 \\ \boldsymbol{\rho}_{L-2,j} & \text{if } l = L-1 \end{cases},$$

where $\mathcal{P}_{\mathbf{X},\beta}(\cdot)$ projects to $\left\{\mathbf{u} \in \mathbb{R}^n \mid \|\mathbf{X}^T\mathbf{u}\|_2 \leq \beta {t_j^*}^{2-L}\right\}$.

Corollary 3.2. *Theorem 3.2 also shows that strong duality holds for the training problem in* (16).

3.2. Training problem with vector outputs

Here, we consider vector output deep networks with the output function $f_{\theta,L}(\mathbf{X}) = \sum_{j=1}^m \mathbf{X}\mathbf{W}_{1,j}\ldots\mathbf{w}_{L,j}^T$. In this case, we have the following training problem

$$\min_{\{\theta_l\}_{l=1}^L} \sum_{j=1}^m\sum_{l=1}^L \|\mathbf{W}_{l,j}\|_F^2 \text{ s.t. } f_{\theta,L}(\mathbf{X}) = \mathbf{Y}. \quad (17)$$

Using the scaling in Lemma A.4 and the same convex duality arguments, optimal layer weights for (17) are as follows.

Theorem 3.4. *Optimal layer weight for* (17) *are*

$$\mathbf{W}_{l,j}^* = \begin{cases} t_j^*\tilde{\mathbf{v}}_{w,j}\boldsymbol{\rho}_{1,j}^T & \text{if } l = 1 \\ t_j^*\boldsymbol{\rho}_{l-1,j}\boldsymbol{\rho}_{l,j}^T & \text{if } 1 < l \leq L-2 \\ \boldsymbol{\rho}_{L-2,j} & \text{if } l = L-1 \end{cases},$$

where $j \in [K]$, $\tilde{\mathbf{v}}_{w,j}$ is the j^{th} maximal right singular vector of ${\boldsymbol{\Lambda}^*}^T\mathbf{X}$ and $\{\boldsymbol{\rho}_{l,j}\}_{l=1}^{L-2}$ are arbitrary unit norm vectors such that $\boldsymbol{\rho}_{l,j}^T\boldsymbol{\rho}_{l,k} = 0,\ \forall j \neq k$.

The next theorem formally proves that strong duality holds for the primal problem in (17).

Theorem 3.5. *Let $\{\mathbf{X}, \mathbf{Y}\}$ be feasible for* (17)*, then strong duality holds for finite width networks.*

3.2.1. Regularized Case

We now examine the following regularized problem

$$\min_{\{\theta_l\}_{l=1}^L} \frac{1}{2}\|f_{\theta,L}(\mathbf{X}) - \mathbf{y}\|_2^2 + \frac{\beta}{2}\sum_{j=1}^m\sum_{l=1}^L \|\mathbf{W}_{l,j}\|_F^2. \quad (18)$$

Next result provides a set of optimal solutions to (18).

Theorem 3.6. *Optimal layer weights for* (18) *are*

$$\mathbf{W}_{l,j}^* = \begin{cases} t_j^*\tilde{\mathbf{v}}_{x,j}\boldsymbol{\rho}_{1,j}^T & \text{if } l = 1 \\ t_j^*\boldsymbol{\rho}_{l-1,j}\boldsymbol{\rho}_{l,j}^T & \text{if } 1 < l \leq L-2 \\ \boldsymbol{\rho}_{L-2,j} & \text{if } l = L-1 \end{cases},$$

where $j \in [K]$, $\tilde{\mathbf{v}}_{x,j}$ is a maximal right singular vector of $\mathcal{P}_{\mathbf{X},\beta}(\mathbf{Y})^T\mathbf{X}$ and $\mathcal{P}_{\mathbf{X},\beta}(\cdot)$ projects to $\{\mathbf{U} \in \mathbb{R}^{n\times k} \mid \sigma_{max}(\mathbf{U}^T\mathbf{X}) \leq \beta {t_j^*}^{2-L}\}$. Additionally, $\boldsymbol{\rho}_{l,j}$'s is an orthonormal set. Therefore, the rank of each hidden layer is determined by β as in Remark 2.1.

4. Deep ReLU networks

Here, we consider an L-layer ReLU network with the output function $f_{\theta,L}(\mathbf{X}) = \mathbf{A}_{L-1}\mathbf{w}_L$, where $\mathbf{A}_{l,j} = (\mathbf{A}_{l-1,j}\mathbf{W}_{l,j})_+,$, $\mathbf{A}_{0,j} = \mathbf{X}$, $\forall l, j$, and $(x)_+ = \max\{0, x\}$. Below, we first state the minimum norm training problem and then present our results

$$\min_{\{\theta_l\}_{l=1}^L} \sum_{j=1}^m\sum_{l=1}^L \|\mathbf{W}_{l,j}\|_F^2 \text{ s.t. } f_{\theta,L}(\mathbf{X}) = \mathbf{y}. \quad (19)$$

Theorem 4.1. *Let $\mathbf{X}$ be a rank-one matrix such that $\mathbf{X} = \mathbf{c}\mathbf{a}_0^T$, where $\mathbf{c} \in \mathbb{R}_+^n$ and $\mathbf{a}_0 \in \mathbb{R}^d$, then strong duality holds and the optimal weights are*

$$\mathbf{W}_{l,j} = \frac{\boldsymbol{\phi}_{l-1,j}}{\|\boldsymbol{\phi}_{l-1,j}\|_2}\boldsymbol{\phi}_{l,j}^T,\ \forall l \in [L-2],\ \mathbf{w}_{L-1,j} = \frac{\boldsymbol{\phi}_{L-2,j}}{\|\boldsymbol{\phi}_{L-2,j}\|_2},$$

where $\boldsymbol{\phi}_{0,j} = \mathbf{a}_0$ and $\{\boldsymbol{\phi}_{l,j}\}_{l=1}^{L-2}$ is a set of vectors such that $\boldsymbol{\phi}_{l,j} \in \mathbb{R}_+^{m_l}$ and $\|\boldsymbol{\phi}_{l,j}\|_2 = t_j^*$, $\forall l \in [L-2], \forall j \in [m]$.

In the sequel, we first examine a two-layer network training problem with bias and then extend this to multi-layer.

Theorem 4.2. *Let $\mathbf{X}$ be a matrix such that $\mathbf{X} = \mathbf{c}\mathbf{a}_0^T$, where $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{a}_0 \in \mathbb{R}^d$. Then, when $L = 2$, a set of optimal solutions to* (19) *is $\{(\mathbf{w}_i, b_i)\}_{i=1}^m$, where $\mathbf{w}_i = s_i\frac{\mathbf{a}_0}{\|\mathbf{a}_0\|_2}$, $b_i = -s_i c_i\|\mathbf{a}_0\|_2$ with $s_i = \pm 1, \forall i \in [m]$.*

Corollary 4.1. *As a result of Theorem 4.2, when we have one dimensional data, i.e., $\mathbf{x} \in \mathbb{R}^n$, an optimal solution to* (19) *can be formulated as $\{(w_i, b_i)\}_{i=1}^m$, where $w_i = s_i$, $b_i = -s_i x_i$ with $s_i = \pm 1, \forall i \in [m]$. Therefore, the optimal network output has kinks only at the input data points, i.e., the output function is in the following form: $f_{\theta,2}(\hat{x}) = \sum_i (\hat{x} - x_i)_+$. Hence, the network output becomes a linear spline interpolation.*

We now extend the results in Theorem 4.2 and Corollary 4.1 to multi-layer ReLU networks.

Proposition 4.1. *Theorem 4.1 still holds when we add a bias term to the last hidden layer, i.e.,* $\sum_j \left(\mathbf{A}_{L-2,j}\mathbf{w}_{L-1,j} + \mathbf{1}_n b_j\right)_+ w_{L,j} = \mathbf{y}$.

Corollary 4.2. *As a result of Theorem 4.2 and Proposition 4.1, for one dimensional data, i.e.,* $\mathbf{x} \in \mathbb{R}^n$*, the optimal network output has kinks only at the input data points, i.e., the output function is in the following form:* $f_{\theta,L}(\hat{x}) = \sum_i \left(\hat{x} - x_i\right)_+$*. Therefore, the optimal network output is a linear spline interpolation.*

In Corollary 4.1 and 4.2, the optimal output function for multi-layer ReLU networks are linear spline interpolators for rank-one data, which generalizes the two-layer results for one-dimensional data in (Savarese et al., 2019; Parhi & Nowak, 2019; Ergen & Pilanci, 2020a;b) to arbitrary depth.

4.1. Regularized problem with vector outputs

We now extend the analysis to regularized training problems with K outputs, i.e., $\mathbf{Y} \in \mathbb{R}^{n\times K}$.

The result in Theorem 4.1 also holds for vector output multi-layer ReLU networks as shown below.

Proposition 4.2. *Theorem 4.1 extends to deep ReLU networks with vector outputs, therefore, the optimal layer weights can be formulated as in Theorem 4.1.*

Now, we extend our characterization to arbitrary rank whitened data matrices and fully characterize the optimal layer weights of a deep ReLU network with K outputs. We also note that one can even obtain closed-form solutions for all the layers weights as proven in the next result.

Theorem 4.3. *Let* $\{\mathbf{X}, \mathbf{Y}\}$ *be a dataset such that* $\mathbf{X}\mathbf{X}^T = \mathbf{I}_n$ *and* $\mathbf{Y}$ *is one-hot encoded, then a set of optimal solutions for the following regularized training problem*

$$\min_{\theta\in\Theta} \frac{1}{2}\|f_{\theta,L}(\mathbf{X}) - \mathbf{Y}\|_F^2 + \frac{\beta}{2}\sum_{j=1}^{m}\sum_{l=1}^{L}\|\mathbf{W}_{l,j}\|_F^2 \tag{20}$$

can be formulated as follows

$$\mathbf{W}_{l,j} = \begin{cases} \frac{\boldsymbol{\phi}_{l-1,j}}{\|\boldsymbol{\phi}_{l-1,j}\|_2}\boldsymbol{\phi}_{l,j}^T, & \textit{if } l \in [L-1] \\ \left(\|\boldsymbol{\phi}_{0,j}\|_2 - \beta\right)_+ \boldsymbol{\phi}_{l-1,j}\mathbf{e}_r^T & \textit{if } l = L \end{cases},$$

where $\boldsymbol{\phi}_{0,j} = \mathbf{X}^T\mathbf{y}_j$*,* $\{\boldsymbol{\phi}_{l,j}\}_{l=1}^{L-2}$ *are vectors such that* $\boldsymbol{\phi}_{l,j} \in \mathbb{R}_+^{m_l}$*,* $\|\boldsymbol{\phi}_{l,j}\|_2 = t_j^*$*, and* $\boldsymbol{\phi}_{l,i}^T\boldsymbol{\phi}_{l,j} = 0,\ \forall i \neq j$*, Moreover,* $\boldsymbol{\phi}_{L-1,j} = \mathbf{e}_j$ *is the* j^{th} *ordinary basis vector.*

Remark 4.1. *We note that the whitening assumption* $\mathbf{X}\mathbf{X}^T = \mathbf{I}_n$ *necessitates that* $n \leq d$*, which might appear to be restrictive. However, this case is common in few-shot classification problems with limited labels (Chen et al., 2018). Moreover, it is challenging to obtain reliable labels in problems involving high dimensional data such as in medical imaging (Hyun et al., 2020) and genetics (Singh & Yamada, 2020), where* $n \leq d$ *is typical. More importantly, SGD employed in deep learning frameworks, e.g., PyTorch and Tensorflow, operate in mini-batches rather than the full dataset. Therefore, even when* $n > d$*, each gradient descent update can only be evaluated on small batches, where the batch size* n_b *satisfies* $n_b \ll d$*. Hence, the* $n \leq d$ *case implicitly occurs during the training phase.*

Remark 4.2. *We also note that the conditions in Theorem 4.3 are common in practical frameworks. As an example, for image classification, it has been shown that whitening significantly improves the classification accuracy of the state-of-the-art architectures, e.g., ResNets, on benchmark datasets such as ImageNet (Huang et al., 2018). Furthermore, the label matrix is one hot encoded in image classification. Therefore, in such cases, there is no need to train a deep ReLU network in an end-to-end manner. Instead one can directly use the closed-form formulas in Theorem 4.3.*

4.2. Regularized problem with Batch Normalization

We now consider a more practical setting with an arbitrary L-layer network and batch normalization (Ioffe & Szegedy, 2015). We first define batch normalization as follows. For the activation matrix $\mathbf{A}_{l-1} \in \mathbb{R}^{n\times m_{l-1}}$, batch normalization applies to each column j independently as follows

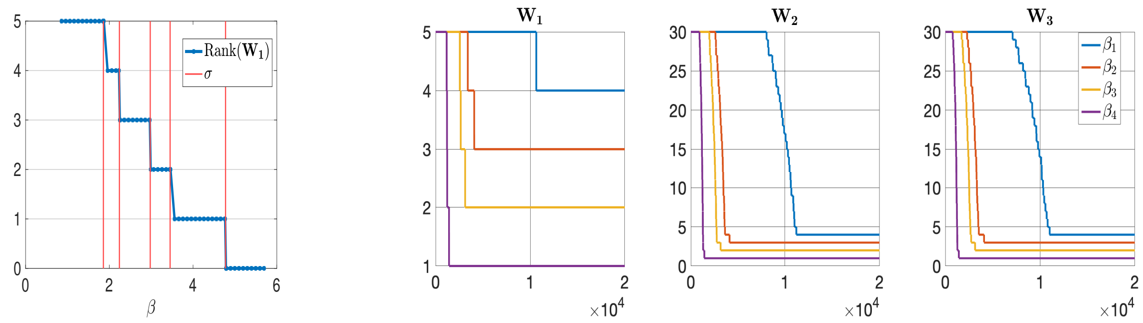
$$\mathrm{BN}_{\gamma,\alpha}\left(\mathbf{A}_{l-1,j}\mathbf{w}_{l,j}\right) = \frac{(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_{n\times n})\mathbf{A}_{l-1,j}\mathbf{w}_{l,j}}{\|(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_{n\times n})\mathbf{A}_{l-1,j}\mathbf{w}_{l,j}\|_2}\gamma_j^{(l)} + \frac{\mathbf{1}_n}{\sqrt{n}}\alpha_j^{(l)},$$

where $\gamma_j^{(l)}$ and $\alpha_j^{(l)}$ scales and shifts the normalized value, respectively. The following theorem presents a complete characterization for the last two layers' weights.

Theorem 4.4. *Suppose* $\mathbf{Y}$ *is one hot encoded and the network is overparameterized such that the range of* $\mathbf{A}_{L-2,j}$ *is* $\mathbb{R}^n$*, then an optimal solution to the following problem*[7]

$$\min_{\theta\in\Theta} \frac{1}{2}\left\|\sum_{j=1}^{m}\left(\mathrm{BN}_{\gamma,\alpha}\left(\mathbf{A}_{L-2,j}\mathbf{w}_{L-1,j}\right)\right)_+ {\mathbf{w}_{L,j}}^T - \mathbf{Y}\right\|_F^2 + \frac{\beta}{2}\sum_{j=1}^{m}\left({\gamma_j^{(L-1)}}^2 + {\alpha_j^{(L-1)}}^2 + \|\mathbf{w}_{L,j}\|_2^2\right),$$

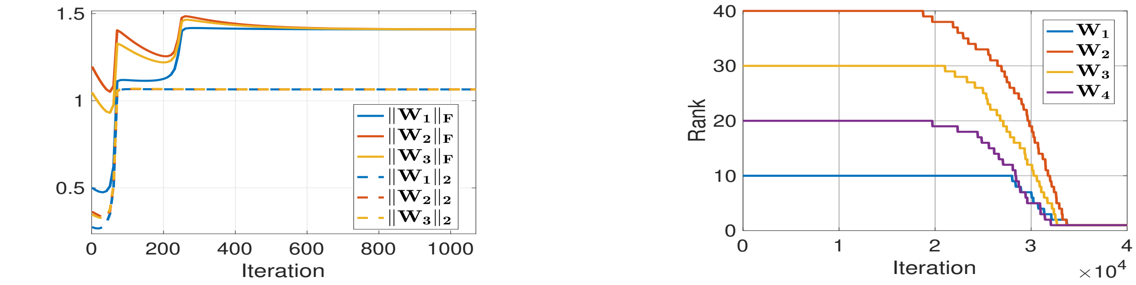
[7] Notice here we only regularize the last layer's parameters, however, regularizing all the parameters does not change the analysis and conclusion as proven in Appendix A.4.



(a)

(b)

Figure 2: Verification of Remark 2.1. (a) Rank of the hidden layer weight matrix as a function of β and (b) rank of the hidden layer weights for different regularization parameters, i.e., $\beta_1 < \beta_2 < \beta_3 < \beta_4$.



(a)

(b)

Figure 3: Verification of Proposition 3.1 and 4.1. (a) Evolution of the operator and Frobenius norms for the layer weights of a linear network and (b) Rank of the layer weights of a ReLU network with $K = 1$.

can be formulated in closed-form as follows

$$\left(\mathbf{w}_{L-1,j}^*, \mathbf{w}_{L,j}^*\right) = \left(\mathbf{A}_{L-2,j}^\dagger \mathbf{y}_j, (\|\mathbf{y}_j\|_2 - \beta)_+ \mathbf{e}_j\right)$$

$$\begin{bmatrix}\gamma_j^{(L-1)*}\\ \alpha_j^{(L-1)*}\end{bmatrix} = \frac{1}{\|\mathbf{y}_j\|_2}\begin{bmatrix}\|\mathbf{y}_j - \frac{1}{n}\mathbf{1}_{n\times n}\mathbf{y}_j\|_2\\ \frac{1}{\sqrt{n}}\mathbf{1}_n^T\mathbf{y}_j\end{bmatrix}$$

$\forall j \in [K]$*, where* $\mathbf{e}_j$ *is the* j^{th} *ordinary basis vector.*

Remark 4.3. *We note that the results in Theorem 4.3 and 4.4 indicate that whitened data and arbitrary data trained with batch normalization effectively yield the same results in the last layer, i.e., both achieve a scaled version of the labels. The difference is that in Theorem 4.3, the labels are obtained after the first layer and carried out to the last layer by aligned layer weights. However, in Theorem 4.4, since batch normalization normalizes layers individually, the scaled labels are obtained after the last hidden layer.*

One-hot encoding is one of the common strategies to convert categorical variables into a binary representation that can be processed by DNNs. Although (Papyan et al., 2020) empirically verified the emergence of certain patterns, termed Neural Collapse, for one-hot encoded labels trained with batch normalization, the theory behind these findings are still unknown. Therefore, we first define a new notion of simplex Equiangular Tight Frame (ETF) and then explain the Neural Collapse phenomenon where class means collapse to the vertices of a simplex ETF. We also note that all of our derivations hold for arbitrary convex loss functions, therefore, are also valid for the commonly adopted cross entropy loss as proven in Appendix A.1.

Definition 2. A standard simplex ETF is a set of points in $\mathbb{R}^K$ selected from the columns of the following matrix

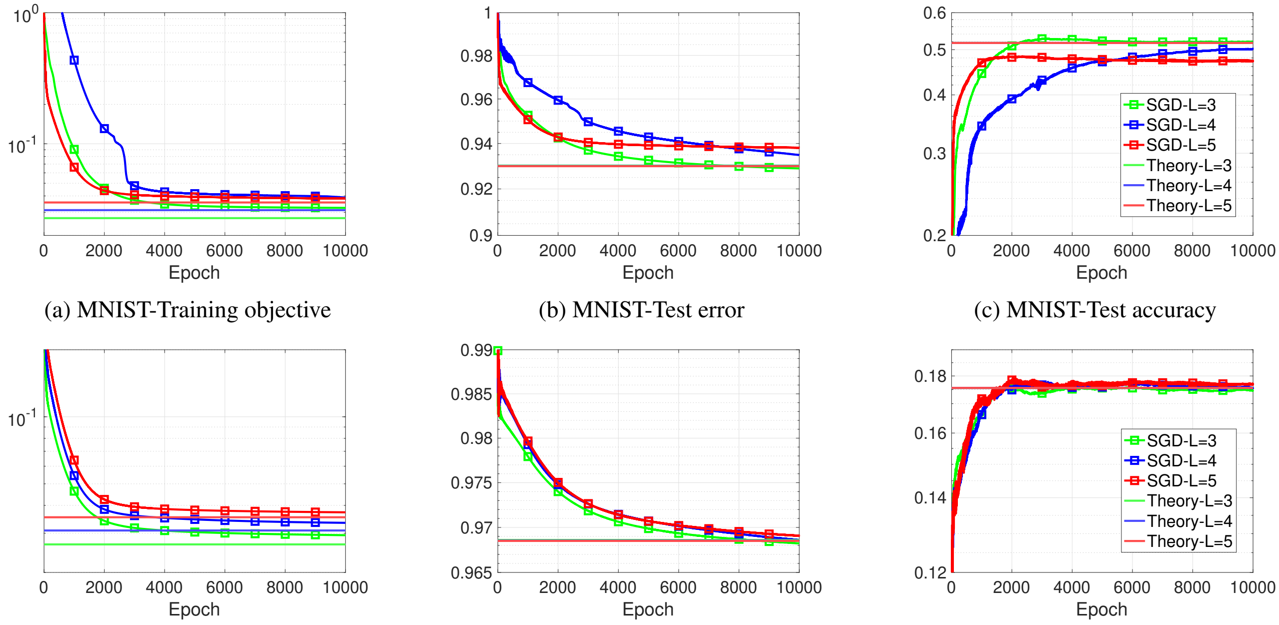
$$\mathbf{S} = \sqrt{\frac{K}{K-1}}\left(\mathbf{I}_K - \frac{1}{K}\mathbf{1}_{K\times K}\right).$$

However, (Papyan et al., 2020) also allows rescaling and rotations of $\mathbf{S}$, i.e., they define a general simplex ETF as $\mathbf{S}_g = \alpha\mathbf{U}\mathbf{S} \in \mathbb{R}^{p\times K}$, where $\mathbf{U}^T\mathbf{U} = \mathbf{I}_K$ and $\alpha \in \mathbb{R}_+$.

Corollary 4.3. *Computing the last hidden layer activations after BN, i.e.,* $\mathbf{A}_{L-1} \in \mathbb{R}^{n\times K}$*, using the optimal layer weight in Theorem 4.4 and then subtracting their global mean as in (Papyan et al., 2020) yields*

$$\left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_{n\times n}\right)\mathbf{A}_{L-1} = \sqrt{\frac{K}{n}}\left(\mathbf{I}_K \otimes \mathbf{1}_{\frac{n}{K}} - \frac{1}{K}\mathbf{1}_{n\times K}\right),$$

where we assume that samples are ordered, i.e., the first n/K *samples belong to class 1, next* n/K *samples belong*



(a) MNIST-Training objective

(b) MNIST-Test error

(c) MNIST-Test accuracy

(d) CIFAR-10-Training objective

(e) CIFAR-10-Test error

(f) CIFAR-10-Test accuracy

Figure 4: Training and test performance on whitened and sampled datasets, where $(n, d) = (60, 90)$, $K = 10$, $L = 3, 4, 5$ with 50 neurons per layer and we use squared loss with one hot encoding. For Theory, we use the layer weights in Theorem 4.3, which achieves the optimal performance as guaranteed by Theorem 4.3.

to class 2 and so on. Therefore, all the activations for a certain class k are the same and their mean is given by $(\sqrt{K/n})(\mathbf{e}_k - \mathbf{1}_K/K)$, which is the k^{th} column of a general simplex ETF with $\alpha = \sqrt{(K-1)/n}$ and $\mathbf{U} = \mathbf{I}_K$.

5. Numerical experiments

Here, we present numerical results to verify our theoretical analysis. We first use synthetic datasets generated from a random data matrix with zero mean and identity covariance and the corresponding output vector is obtained via a randomly initialized teacher network[8]. We first consider a two-layer linear network with $\mathbf{W}_1 \in \mathbb{R}^{20\times 50}$ and $\mathbf{W}_2 \in \mathbb{R}^{50\times 5}$. To prove our claim in Remark 2.1, we train the network using GD with different β. In Figure 2a, we plot the rank of $\mathbf{W}_1$ as a function of β, as well as the location of the singular values of $\mathbf{Y}^T\mathbf{X}$ using vertical red lines. This shows that the rank of the layer changes when β is equal to one of the singular values, which verifies Remark 2.1. We also consider a four-layer linear network with $\mathbf{W}_{1,j} \in \mathbb{R}^{5\times 50}$, $\mathbf{W}_{2,j} \in \mathbb{R}^{50\times 30}$, $\mathbf{W}_{3,j} \in \mathbb{R}^{30\times 40}$, and $\mathbf{W}_{4,j} \in \mathbb{R}^{40\times 5}$. We then select different regularization parameters as $\beta_1 < \beta_2 < \beta_3 < \beta_4$. As illustrated in Figure 2b, β determines the rank of each weight matrix and the rank is same for all the layers, which matches with our results. Moreover, to verify Proposition 3.1, we choose β such that the weights are rank-two. In Figure 3a, we numerically show that all the hidden layer weight matrices have the same operator and Frobenius norms. We also conduct an experiment for a five-layer ReLU network with $\mathbf{W}_{1,j} \in \mathbb{R}^{10\times 50}$, $\mathbf{W}_{2,j} \in \mathbb{R}^{50\times 40}$, $\mathbf{W}_{3,j} \in \mathbb{R}^{40\times 30}$, $\mathbf{W}_{4,j} \in \mathbb{R}^{30\times 20}$, and $\mathbf{w}_{5,j} \in \mathbb{R}^{20\times 1}$. Here, we use data such that $\mathbf{X} = \mathbf{c}\mathbf{a}_0^T$, where $\mathbf{c} \in \mathbb{R}^n_+$ and $\mathbf{a}_0 \in \mathbb{R}^d$. In Figure 3b, we plot the rank of each weight matrix, which converges to one as claimed Proposition 4.1.

We also verify our theory on two real benchmark datasets, i.e., MNIST (LeCun) and CIFAR-10 (Krizhevsky et al., 2014). We first randomly undersample and whitened these datasets. We then convert the labels into one hot encoded form. Then, we consider a ten class classification/regression task using three multi-layer ReLU network architectures with $L = 3, 4, 5$. For each architecture, we use SGD with momentum for training and compare the training/test performance with the corresponding network constructed via the closed-form solutions (without any sort of training) in Theorem 4.3, i.e., denoted as "Theory". In Figure 4, Theory achieves the optimal training objective, which also yields smaller error and higher test accuracy. Thus, we numerically verify the claims in Theorem 4.3.

6. Concluding remarks

We studied regularized DNN training problems and developed an analytic framework to characterize the optimal so-

[8] Additional numerical results can be found in Appendix A.5.

lutions. We showed that optimal weights can be explicitly formulated as the extreme points of a convex set via the dual problem. We then proved that strong duality holds for both deep linear and ReLU networks and provided a set of optimal solutions. We also extended our derivations to the vector outputs and many other loss functions. More importantly, our analysis shows that when the input data is whitened or rank-one, instead of training an L-layer deep ReLU network in an end-to-end manner, one can directly use the closed-form solutions provided in Theorem 4.1 and 4.3. Furthermore, we showed that whitening/rank-one assumptions can be removed via batch normalization (see Theorem 4.4). After our work, this was also realized by (Ergen et al., 2021), where the authors proved that batch normalization effectively whitens the input data matrix. As a corollary, we uncovered theoretical reasons behind a recent empirical observation termed Neural Collapse (Papyan et al., 2020). As another corollary, we proved that the kinks of ReLU occur exactly at the input data so that the optimal network outputs linear spline interpolations for one-dimensional datasets, which was previously known only for two-layer networks (Savarese et al., 2019; Parhi & Nowak, 2019; Ergen & Pilanci, 2020a;b).

As the limitation of this work, we note that for networks with more than two-layers (i.e., $L > 2$), we use a non-standard architecture, where each layer consists of m weight matrices. Thus, we are able to achieve strong duality which is essential for our analysis. We leave the strong duality analysis of standard deep networks as an open research problem for future work.

Acknowledgements

This work was partially supported by the National Science Foundation under grants IIS-1838179 and ECCS-2037304, Facebook Research, Adobe Research and Stanford SystemX Alliance.

References

Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks. *CoRR*, abs/1810.02281, 2018a. URL http://arxiv.org/abs/1810.02281.

Arora, S., Cohen, N., and Hazan, E. On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*, 2018b.

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization. *arXiv preprint arXiv:1905.13655*, 2019.

Bach, F. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

Bhojanapalli, S., Neyshabur, B., and Srebro, N. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, pp. 3873–3881, 2016.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Brutzkus, A., Globerson, A., Malach, E., and Shalev-Shwartz, S. SGD learns over-parameterized networks that provably generalize on linearly separable data. *CoRR*, abs/1710.10174, 2017. URL http://arxiv.org/abs/1710.10174.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2018.

Du, S. and Hu, W. Width provably matters in optimization for deep linear neural networks. In *International Conference on Machine Learning*, pp. 1655–1664, 2019.

Du, S. S., Hu, W., and Lee, J. D. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, pp. 384–395, 2018a.

Du, S. S., Zhai, X., Póczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *CoRR*, abs/1810.02054, 2018b. URL http://arxiv.org/abs/1810.02054.

Ergen, T. and Pilanci, M. Convex duality and cutting plane methods for over-parameterized neural networks. In *OPT-ML workshop*, 2019.

Ergen, T. and Pilanci, M. Convex optimization for shallow neural networks. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 79–83, 2019.

Ergen, T. and Pilanci, M. Convex geometry of two-layer relu networks: Implicit autoencoding and interpretable models. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4024–4033, Online, 26–28 Aug 2020a. PMLR. URL http://proceedings.mlr.press/v108/ergen20a.html.

Ergen, T. and Pilanci, M. Convex geometry and duality of over-parameterized neural networks. *arXiv preprint arXiv:2002.11219*, 2020b.

Ergen, T. and Pilanci, M. Convex programs for global optimization of convolutional neural networks in polynomial-time. In *OPT-ML workshop*, 2020c.

Ergen, T. and Pilanci, M. Implicit convex regularizers of cnn architectures: Convex optimization of two- and three-layer networks in polynomial time. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=0N8jUH4JMv6`.

Ergen, T., Sahiner, A., Ozturkler, B., Pauly, J. M., Mardani, M., and Pilanci, M. Demystifying batch normalization in relu networks: Equivalent convex optimization models and implicit regularization. *CoRR*, abs/2103.01499, 2021. URL `https://arxiv.org/abs/2103.01499`.

Goberna, M. A. and López-Cerdá, M. *Linear semi-infinite optimization*. 01 1998. doi: 10.1007/978-1-4899-8044-1_3.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 6151–6159, 2017.

Gupta, V., Bartan, B., Ergen, T., and Pilanci, M. Convex neural autoregressive models: Towards tractable, expressive, and theoretically-backed models for sequential forecasting and generation. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3890–3894, 2021. doi: 10.1109/ICASSP39728.2021.9413662.

Huang, L., Yang, D., Lang, B., and Deng, J. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 791–800, 2018.

Hyun, C. M., Kim, K. C., Cho, H. C., Choi, J. K., and Seo, J. K. Framelet pooling aided deep learning network: the method to process high dimensional medical data. *Machine Learning: Science and Technology*, 1(1):015009, 2020.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL `http://proceedings.mlr.press/v37/ioffe15.html`.

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HJflg30qKX`.

Krizhevsky, A., Nair, V., and Hinton, G. The CIFAR-10 dataset. `http://www.cs.toronto.edu/kriz/cifar.html`, 2014.

Laurent, T. and Brecht, J. Deep linear networks with arbitrary loss: All local minima are global. In *International Conference on Machine Learning*, pp. 2902–2907, 2018.

LeCun, Y. The MNIST database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. *CoRR*, abs/1808.01204, 2018. URL `http://arxiv.org/abs/1808.01204`.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

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

Parhi, R. and Nowak, R. D. Minimum "norm" neural networks are splines. *arXiv preprint arXiv:1910.02333*, 2019.

Pilanci, M. and Ergen, T. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7695–7705. PMLR, 13–18 Jul 2020. URL `http://proceedings.mlr.press/v119/pilanci20a.html`.

Rosset, S., Swirszcz, G., Srebro, N., and Zhu, J. L1 regularization in infinite dimensional feature spaces. In *International Conference on Computational Learning Theory*, pp. 544–558. Springer, 2007.

Sahiner, A., Ergen, T., Pauly, J. M., and Pilanci, M. Vector-output relu neural network problems are copositive programs: Convex analysis of two layer networks and polynomial-time algorithms. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=fGF8qAqpXXG`.

Savarese, P., Evron, I., Soudry, D., and Srebro, N. How do infinite width bounded norm networks look in function space? *CoRR*, abs/1902.05040, 2019. URL `http://arxiv.org/abs/1902.05040`.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Shamir, O. Exponential convergence time of gradient descent for one-dimensional deep linear neural networks. *arXiv preprint arXiv:1809.08587*, 2018.

Singh, D. and Yamada, M. Fsnet: Feature selection network on high-dimensional biological data. *arXiv preprint arXiv:2001.08322*, 2020.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. On the margin theory of feedforward neural networks. *arXiv preprint arXiv:1810.05369*, 2018.

Zhang, H., Shao, J., and Salakhutdinov, R. Deep neural networks with multi-branch architectures are intrinsically less non-convex. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1099–1109, 2019.