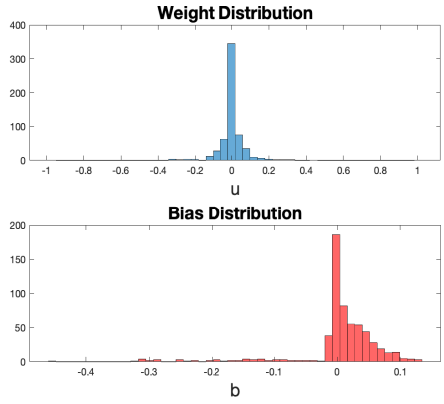
Supplementary File
Convex Geometry of Two-Layer ReLU Networks: Implicit Autoencoding and Interpretable Models

In this file, we present proofs of the main results, details on the algorithms and numerical results, and extra figures that are not included in the main paper due to the page limit. We refer to the equations in the main paper as [Main Paper,(#)] to prevent ambiguities.

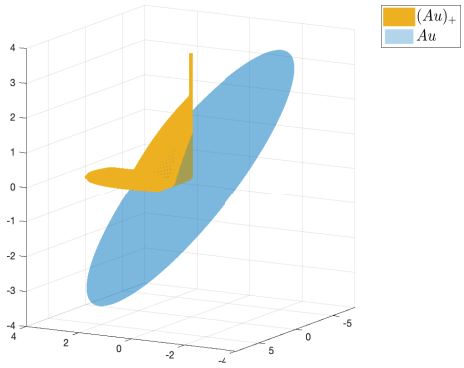
Contents

1 Additional figures

Here, we present additional figures (see Figure 1) that support our claims in the main paper. In 1D ReLU neural network experiments, we note that even though the weights and biases might take quite different values for each neuron, their activation points , i.e., $-b_i/u_i$ for each neuron i, correspond to the data samples. The distribution of the weights and biases are shown in Figure 1a in this document. In 1b, we also illustrate a rectified ellipsoid set in three dimensions to complement the two dimensional figures in the main paper.



(a) Weight and bias distributions for Figure 1 in the main paper.



(b) 3D Illustration of the rectified ellipsoid.

Figure 1: Additional figures.

2 Additional details on the numerical experiments

In this section, we provide further information about our experimental setup.

In the main paper, we evaluate the performance of the introduced approach on several real data sets. For comparison, we also include the performance of a two-layer NN trained with the backpropagation algorithm and the well-known linear least squares approach. For all the experiments, we use the regularization term (also known as weight decay) to let the algorithms generalize well on unseen data (Krogh and Hertz, 1992). In addition to this, we use the cutting plane based algorithm along with the neurons in [Main Paper,(9)] for our convex approach. In order to solve the convex optimization problems in our approach, we use CVX (Grant and Boyd, 2014). However, notice that when dealing with large data sets, e.g., CIFAR-10, plain CVX solvers might take a lot of time or return memory errors. In order to circumvent these issues, we use SPGL1 (van den Berg and Friedlander, 2007) and SuperSCS (Themelis and Patrinos, 2019) for large data sets. We also remark that all the data sets we use are publicly available and further information, e.g., training and test sizes, can be obtained through the provided references (LeCun; Krizhevsky et al., 2014; Torgo; new). Furthermore, we use the same number of hidden neurons for both our approach and the conventional backpropagation based approach to have a fair comparison.

For `Convex-RF`, we use the convolutional neural net architecture in (Coates and Ng, 2012). However, instead of using random filters as in (Zhang et al., 2016) or applying the k-means algorithm as in (Coates and Ng, 2012), we use filters that are extracted from the patches using the proposed convex approach. Particularly, we first randomly obtain patches from the dataset. We then apply [Main Paper, (9)] to obtain the filter weights. Using these filter weights, we train the architecture in (Coates and Ng, 2012) to obtain the provided results.

In order to gain further understanding of the connection between implicit regularization and initial standard deviation of the neuron weights, we perform an experiment that is presented in the main paper, i.e., Figure 1. In this experiment, using the backpropagation algorithm, we train two-layer

NNs with different initial standard deviations such that each network completely fits the training data. Then, we find the maximum absolute difference between the function fit by the NNs and the ground truth linear interpolation. After averaging our results over many random trials, we obtain Figure 1. The same settings are also used for the experiment with hinge loss.

3 Cutting plane algorithm with no bias term

In this section, we present the pseudocode for the algorithm provided in the main paper when there is no bias term.

In the cutting plane method, we first find a violating neuron using [Main Paper, (14)]. After adding these parameters to $\mathbf{U}$ as columns, we solve [Main Paper, (4)]. If we cannot find a new violating neuron then we terminate the algorithm. Otherwise, we find the dual parameter for the updated $\mathbf{U}$. We repeat this procedure till we find an optimal solution (see Algorithm 1).

Algorithm 1 Cutting Plane based Training Algorithm for Two-Layer NNs (without bias)

Initialize $\mathbf{v} = \mathbf{y}$
while there exists a violating neuron **do**
    Find $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{u}}_2$ via [Main Paper,(14)]
    $\mathbf{U} \leftarrow [\mathbf{U}\ \hat{\mathbf{u}}_1\ \hat{\mathbf{u}}_2]$
    Find $\mathbf{v}$ using the dual problem in [Main Paper,(11)]
    Check the existence of a violating neuron via [Main Paper,(14)]
Solve [Main Paper,(4)] using $\mathbf{U}$
Return $\theta = (\mathbf{U}, \mathbf{w})$

4 Cutting plane algorithm with a bias term

Here, we include the cutting plane algorithm which accommodates a bias term. This is slightly more involved than the case with no bias because of extra constraints. We have the corresponding dual problem as in Theorem 3.1

$$\max_{\mathbf{v}\in\mathbb{R}^n,\, \mathbf{1}^T\mathbf{v}=0} \mathbf{v}^T\mathbf{y} \text{ s.t. } \left|\mathbf{v}^T\left(\mathbf{A}\mathbf{u} + b\mathbf{1}\right)_+\right| \leq 1\,, \forall \mathbf{u} \in \mathcal{B}_2, \forall b \in \mathbb{R} \tag{1}$$

and an optimal $(\mathbf{U}^*,\, \mathbf{b}^*)$ satisfies

$$\|(\mathbf{A}\mathbf{U}^* + \mathbf{1}\mathbf{b}*^T)_+^T\mathbf{v}^*\|_\infty = 1\,,$$

where $\mathbf{v}^*$ is the optimal dual variable.

Among infinitely many possible unit norm weights, we need to find the weights that violate the inequality constraint in the dual form, which can be done by solving the following optimization problems

$$\mathbf{u}_1^* = \underset{\mathbf{u},b}{\operatorname{argmax}}\, \mathbf{v}^T(\mathbf{A}\mathbf{u} + b\mathbf{1})_+ \text{ s.t. } \|\mathbf{u}\|_2 \leq 1$$
$$\mathbf{u}_2^* = \underset{\mathbf{u},b}{\operatorname{argmin}}\, \mathbf{v}^T(\mathbf{A}\mathbf{u} + b\mathbf{1})_+ \text{ s.t. } \|\mathbf{u}\|_2 \leq 1.$$

However, the above problem is not convex since ReLU is a convex function. In this case, we can further relax the problem by applying the spike-free relaxation as follows

$$(\hat{\mathbf{u}}_1, \hat{b}_1) = \underset{\mathbf{u},b}{\operatorname{argmax}}\, \mathbf{v}^T\mathbf{A}\mathbf{u} + b\mathbf{v}^T\mathbf{1} \text{ s.t. } \mathbf{A}\mathbf{u} + b\mathbf{1} \succcurlyeq \mathbf{0}, \|\mathbf{u}\|_2 \leq 1$$
$$(\hat{\mathbf{u}}_2, \hat{b}_2) = \underset{\mathbf{u},b}{\operatorname{argmin}}\, \mathbf{v}^T\mathbf{A}\mathbf{u} + b\mathbf{v}^T\mathbf{1} \text{ s.t. } \mathbf{A}\mathbf{u} + b\mathbf{1} \succcurlyeq \mathbf{0}, \|\mathbf{u}\|_2 \leq 1,$$

where we relax the set $\{(\mathbf{A}\mathbf{u}+b\mathbf{1})_+|\mathbf{u}\in\mathbb{R}^d, \|\mathbf{u}\|_2\leq 1\}$ as $\{\mathbf{A}\mathbf{u}+b\mathbf{1}|\mathbf{u}\in\mathbb{R}^d, \|\mathbf{u}\|_2\leq 1\}\cap\mathbb{R}^n_+$. Now, we can find the weights and biases for the hidden layer using convex optimization. However, notice that depending on the sign of $\mathbf{1}^T\mathbf{v}$ one of the problems will be unbounded. Thus, if $\mathbf{1}^T\mathbf{v}\neq 0$, then we can always find a violating constraint, which will make the problem infeasible. However, note that here we do not use bias for the output layer. If we use add a bias term to the output layer then $\mathbf{1}^T\mathbf{v}=0$ will be enforced via the dual.

Based on our analysis, we propose the following convex optimization approach to train the two-layer NN. We first find a violating neuron. After adding these parameters to $\mathbf{U}$ as a column and to $\mathbf{b}$ as a row, we try to solve the original problem. If we cannot find a new violating neuron then we terminate the algorithm. Otherwise, we find the dual parameter for the updated $\mathbf{U}$. We repeat this procedure until the optimality conditions are satisfied (see Algorithm 2 for the pseudocode). Since the constraint is bounded below and $\hat{\mathbf{u}}_j$'s are bounded, Algorithm 2 is guaranteed to converge in finitely many iterations Theorem 11.2 of (Goberna and López-Cerdá, 1998).

Algorithm 2 Cutting Plane based Training Algorithm for Two-Layer NNs (with bias)

1: Initialize $\mathbf{v}$ such that $\mathbf{1}^T\mathbf{v}=0$
2: **while** there exists a violating neuron **do**
3: Find $\hat{\mathbf{u}}_1$, $\hat{\mathbf{u}}_2$, $\hat{b}_1$ and $\hat{b}_2$
4: $\mathbf{U}\leftarrow[\mathbf{U}\ \hat{\mathbf{u}}_1\ \hat{\mathbf{u}}_2]$
5: $\mathbf{b}\leftarrow[\mathbf{b}^T\ \hat{b}_1\ \hat{b}_2]^T$
6: Find $\mathbf{v}$ using the dual problem
7: Check the existence of a violating neuron
8: Solve the problem using $\mathbf{U}$ and $\mathbf{b}$
9: Return $\theta=(\mathbf{U},\mathbf{b},\mathbf{w})$

5 Infinite size neural networks

Here we briefly review infinite size, i.e., infinite width, two-layer NNs (Bach, 2017). We refer the reader to (Bengio et al., 2006; Wei et al., 2018) for further background and connections to our work. Consider an arbitrary measurable input space $\mathcal{X}$ with a set of continuous basis functions $\phi_{\mathbf{u}}:\mathcal{X}\to\mathbb{R}$ parametrized by $\mathbf{u}\in\mathcal{B}_2$. We then consider real-valued Radon measures equipped with the uniform norm (Rudin, 1964). For a signed Radon measure $\boldsymbol{\mu}$, we define the infinite size neural network output for the input $\mathbf{x}\in\mathcal{X}$ as

$$f(\mathbf{x})=\int_{\mathbf{u}\in\mathcal{B}_2}\phi_{\mathbf{u}}(\mathbf{x})d\boldsymbol{\mu}(\mathbf{u})\,.$$

The total variation norm of the signed measure $\boldsymbol{\mu}$ is defined as the supremum of $\int_{\mathbf{u}\in\mathcal{B}_2}q(\mathbf{u})d\boldsymbol{\mu}(\mathbf{u})$ over all continuous functions $q(\mathbf{u})$ that satisfy $|q(\mathbf{u})|\leq 1$. Now we consider the ReLU basis functions $\phi_{\mathbf{u}}(\mathbf{x})=\left(\mathbf{x}^T\mathbf{u}\right)_+$. For finitely many neurons, the network output is given by

$$f(\mathbf{x})=\sum_{j=1}^m\phi_{\mathbf{u}_j}(\mathbf{x})w_j\,,$$

which corresponds to the signed measure $\boldsymbol{\mu}=\sum_{j=1}^m w_j\delta(\mathbf{u}-\mathbf{u}_j)$ where δ is the Dirac delta function. And the total variation norm $\|\boldsymbol{\mu}\|_{TV}$ of $\boldsymbol{\mu}$ reduces to the ℓ_1 norm $\|\mathbf{w}\|_1$.

The infinite dimensional version of the problem [Main Paper, (4)] corresponds to

$$\begin{aligned}&\min\|\boldsymbol{\mu}\|_{TV}\\&\text{s.t.}\quad f(\mathbf{x}_i)=y_i\,,\forall i\in[n]\,.\end{aligned}$$

For finitely many neurons, i.e., when the measure $\boldsymbol{\mu}$ is a mixture of Dirac delta basis functions, the equivalent problem is

$$\begin{aligned} &\min \|\mathbf{w}\|_1 \\ &\text{s.t.} \quad f(\mathbf{x}_i) = y_i \,, \forall i \in [n] \,. \end{aligned}$$

which is identical to [Main Paper, (4)]. Similar results also hold with regularized objective functions, different loss functions and vector outputs.

6 Proofs of the main results

In this section, we present the proofs of the theorems and lemmas provided in the main paper.

Proof of Lemma 2.1. For any $\theta \in \Theta$, we can rescale the parameters as $\bar{\mathbf{u}}_j = \alpha_j \mathbf{u}_j$, $\bar{b}_j = \alpha_j b_j$ and $\bar{w}_j = w_j/\alpha_j$, for any $\alpha_j > 0$. Then, [Main Paper, (1)] becomes

$$f_{\bar{\theta}}(\mathbf{A}) = \sum_{j=1}^{m} \bar{w}_j (\mathbf{A}\bar{\mathbf{u}}_j + \bar{b}_j \mathbf{1})_+ = \sum_{j=1}^{m} \frac{w_j}{\alpha_j} (\alpha_j \mathbf{A}\mathbf{u}_j + \alpha_j b_j \mathbf{1})_+ = \sum_{j=1}^{m} w_j (\mathbf{A}\mathbf{u}_j + b_j \mathbf{1})_+,$$

which proves $f_\theta(\mathbf{A}) = f_{\bar{\theta}}(\mathbf{A})$. In addition to this, we have the following basic inequality

$$\frac{1}{2} \sum_{j=1}^{m} (w_j^2 + \|\mathbf{u}_j\|_2^2) \geq \sum_{j=1}^{m} (|w_j| \; \|\mathbf{u}_j\|_2),$$

where the equality is achieved with the scaling choice $\alpha_j = \left(\frac{|w_j|}{\|\mathbf{u}_j\|_2}\right)^{\frac{1}{2}}$. Since the scaling operation does not change the right-hand side of the inequality, we can set $\|\mathbf{u}_j\|_2 = 1, \forall j$. Therefore, the right-hand side becomes $\|\mathbf{w}\|_1$. □

Proof of Lemma 2.2. Consider the following problem

$$\min_{\theta \in \Theta} \|\mathbf{w}\|_1 \text{ s.t. } f_\theta(\mathbf{A}) = \mathbf{y}, \|\mathbf{u}_j\|_2 \leq 1, \forall j,$$

where the unit norm equality constraint is relaxed. Let us assume that for a certain index j, we obtain $\|\mathbf{u}_j\|_2 < 1$ with $w_j \neq 0$ as the optimal solution of the above problem. This shows that the unit norm inequality constraint is not active for $\mathbf{u}_j$, and hence removing the constraint for $\mathbf{u}_j$ will not change the optimal solution. However, when we remove the constraint, $\|\mathbf{u}_j\|_2 \to \infty$ reduces the objective value since it yields $w_j = 0$. Hence, we have a contradiction, which proves that all the constraints that correspond to a nonzero w_j must be active for an optimal solution. □

Proof of Lemma 2.3. The first condition immediately implies that $\{\big(\mathbf{A}\mathbf{u}\big)_+ | \mathbf{u} \in \mathcal{B}_2\} \subseteq \mathbf{A}\mathcal{B}_2$. Since we also have $\{\big(\mathbf{A}\mathbf{u}\big)_+ | \mathbf{u} \in \mathcal{B}_2\} \subseteq \mathbb{R}^n_+$, it holds that $\{\big(\mathbf{A}\mathbf{u}\big)_+ | \mathbf{u} \in \mathcal{B}_2\} \subseteq \mathbf{A}\mathcal{B}_2 \cap \mathbb{R}^n_+$. The projection of $\mathbf{A}\mathcal{B}_2 \cap \mathbb{R}^n_+$ onto the positive orthant is a subset of $\mathcal{Q}_\mathbf{A}$, and consequently we have $\mathcal{Q}_\mathbf{A} = \mathbf{A}\mathcal{B}_2 \cap \mathbb{R}^n_+$. The second conditions follow from the min-max representation

$$\max_{\mathbf{u} \in \mathcal{B}_2} \quad \min_{\mathbf{z}:\, \mathbf{A}\mathbf{z} = (\mathbf{A}\mathbf{u})_+} \|\mathbf{z}\|_2 \leq 1 \iff [MainPaper, (6)] \,,$$

by noting that $(\mathbf{I}_n - \mathbf{A}\mathbf{A}^\dagger)(\mathbf{A}\mathbf{u})_+ = \mathbf{0}$ if and only if there exists $\mathbf{z}$ such that $\mathbf{A}\mathbf{z} = (\mathbf{A}\mathbf{u})_+$, which in that case provided by $\mathbf{A}^\dagger(\mathbf{A}\mathbf{u})_+$. The third condition follows from the fact that the minimum norm solution to $\mathbf{A}\mathbf{z} = (\mathbf{A}\mathbf{u})_+$ is given by $\mathbf{A}^\dagger(\mathbf{A}\mathbf{u})_+$ under the full row rank assumption on $\mathbf{A}$, which in turn implies $\mathbf{I}_n - \mathbf{A}\mathbf{A}^\dagger = \mathbf{0}$. □

Proof of Lemma 2.4. We have

$$\begin{aligned}\max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2\leq 1} \|\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\big(\mathbf{A}\mathbf{u}\big)_+\|_2 &\leq \sigma_{\max}(\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}) \max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2\leq 1} \|\big(\mathbf{A}\mathbf{u}\big)_+\|_2 \\ &= \sigma_{\min}^{-1}(\mathbf{A}) \max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2\leq 1} \|\big(\mathbf{A}\mathbf{u}\big)_+\|_2 \\ &\leq \sigma_{\min}^{-1}(\mathbf{A}) \max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2\leq 1} \|\mathbf{A}\mathbf{u}\|_2 \\ &\leq \sigma_{\min}^{-1}(\mathbf{A})\sigma_{\max}(\mathbf{A}) \\ &\leq 1\,.\end{aligned}$$

where the last inequality follows from the fact that $\mathbf{A}$ is whitened. □

Proof of Lemma 2.5. Let us consider a data matrix $\mathbf{A}$ such that $\mathbf{A} = \mathbf{c}\mathbf{a}^T$, where $\mathbf{c} \in \mathbb{R}^n_+$ and $\mathbf{a} \in \mathbb{R}^d$. Then, $\big(\mathbf{A}\mathbf{u}\big)_+ = \mathbf{c}\big(\mathbf{a}^T\mathbf{u}\big)_+$. If $\big(\mathbf{a}^T\mathbf{u}\big)_+ = 0$, then we can select $\mathbf{z} = \mathbf{0}$ to satisfy the spike-free condition $\big(\mathbf{c}\mathbf{a}^T\mathbf{u}\big)_+ = \mathbf{A}\mathbf{z}$. If $\big(\mathbf{a}^T\mathbf{u}\big)_+ \neq 0$, then $\big(\mathbf{A}\mathbf{u}\big)_+ = \mathbf{c}\mathbf{a}^T\mathbf{u} = \mathbf{A}\mathbf{u}$, where the spike-free condition can be trivially satisfied with the choice of $\mathbf{z} = \mathbf{u}$. □

Proof of Lemma 2.6. The extreme point along the direction of $\mathbf{v}$ can be found as follows

$$\max_{u,b} \sum_{i=1}^{n} v_i(a_i u + b)_+ \text{ s.t. } |u| = 1, \tag{2}$$

Since each neuron separates the samples into two sets, for some samples, ReLU will be active, i.e., $\mathcal{S} = \{i|a_i u + b \geq 0\}$, and for the others, it will be inactive, i.e., $\mathcal{S}^c = \{j|a_j u + b < 0\} = [n]/\mathcal{S}$. Thus, we modify (2) as

$$\max_{u,b} \sum_{i\in\mathcal{S}} v_i(a_i u + b) \text{ s.t. } a_i u + b \geq 0, \forall i \in \mathcal{S}, a_j u + b \leq 0, \forall j \in \mathcal{S}^c, |u| = 1. \tag{3}$$

In (3), u can only take two values, i.e., ± 1. Thus, we can separately solve the optimization problem for each case and then take the maximum one as the optimal. Let us assume that $u = 1$. Then, (3) reduces to finding the optimal bias. We note that due to the constraints in (3), $-a_i \leq b \leq -a_j, \forall i \in \mathcal{S}, \forall j \in \mathcal{S}^c$. Thus, the range for the possible bias values is $[\max_{i\in\mathcal{S}}(-a_i),\ \min_{j\in\mathcal{S}^c}(-a_j)]$. Therefore, depending on the direction $\mathbf{v}$, the optimal bias can be selected as follows

$$b_v = \begin{cases} \max_{i\in\mathcal{S}}(-a_i), & \text{if } \sum_{i\in\mathcal{S}} v_i \leq 0 \\ \min_{j\in\mathcal{S}^c}(-a_j), & \text{otherwise} \end{cases}. \tag{4}$$

Similar arguments also hold for $u = -1$ and the min version of (2). Note that when $\sum_{i\in\mathcal{S}} v_i = 0$, the value of the bias does not change the objective in (3). Thus, all the bias values in the range $[\max_{i\in\mathcal{S}}(-a_i),\ \min_{j\in\mathcal{S}^c}(-a_j)]$ become optimal. In such cases, there might exist multiple optimal solutions for the training problem. □

Proof of Lemma 2.7. For the extreme point in the span of $\mathbf{e}_i$, we need to solve the following optimization problem

$$\max_{\mathbf{u},b} (\mathbf{a}_i^T\mathbf{u} + b) \text{ s.t. } \mathbf{a}_j^T\mathbf{u} + b \leq 0, \forall i \neq j, \|\mathbf{u}\|_2 = 1. \tag{5}$$

Then the Lagrangian of (5) is

$$L(\boldsymbol{\lambda}, \mathbf{u}, b) = \mathbf{a}_i^T\mathbf{u} + b - \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_j(\mathbf{a}_j^T\mathbf{u} + b), \tag{6}$$

where we do not include the unit norm constraint for $\mathbf{u}$. For (6), $\boldsymbol{\lambda}$ must satisfy $\boldsymbol{\lambda} \succcurlyeq \mathbf{0}$ and $\mathbf{1}^T\boldsymbol{\lambda} = 1$. With these specifications, the problem can be written as

$$\min_{\boldsymbol{\lambda}} \max_{\mathbf{u}} \mathbf{u}^T \left(\mathbf{a}_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_j \mathbf{a}_j \right) \text{ s.t. } \boldsymbol{\lambda} \succcurlyeq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1, \|\mathbf{u}\|_2 = 1. \tag{7}$$

Since the $\mathbf{u}$ vector that maximizes (7) is the normalized version of the term inside the parenthesis above, the problem reduces to

$$\min_{\boldsymbol{\lambda}} \left\| \mathbf{a}_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_j \mathbf{a}_j \right\|_2 \text{ s.t. } \boldsymbol{\lambda} \succcurlyeq \mathbf{0}, \mathbf{1}^T\boldsymbol{\lambda} = 1. \tag{8}$$

After solving the convex problem (8) for each i, we can find the corresponding neurons as follows

$$\mathbf{u}_i = \frac{\mathbf{a}_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_j \mathbf{a}_j}{\left\| \mathbf{a}_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_j \mathbf{a}_j \right\|_2} \text{ and } b_i = \min_{j\neq i}(-\mathbf{a}_j^T \mathbf{u}_i),$$

where the bias computation follows from the constraint in (5). □

Proof of Lemma 2.8. For any $\boldsymbol{\alpha} \in \mathbb{R}^n$, the extreme point along the direction of $\boldsymbol{\alpha}$ can be found by solving the following optimization problem

$$\max_{\mathbf{u},b} \boldsymbol{\alpha}^T (\mathbf{A}\mathbf{u} + b\mathbf{1})_+ \text{ s.t. } \|\mathbf{u}\|_2 = 1 \tag{9}$$

where the optimal $(\mathbf{u}, b)$ groups samples into two sets so that some of them activates ReLU with the indices $\mathcal{S} = \{i | \mathbf{a}_i^T\mathbf{u} + b \geq 0\}$ and the others deactivate it with the indices $\mathcal{S}^c = \{j | \mathbf{a}_j^T\mathbf{u} + b < 0\} = [n]/\mathcal{S}$. Using this, we equivalently write (9) as

$$\max_{\mathbf{u},b} \sum_{i\in\mathcal{S}} \alpha_i(\mathbf{a}_i^T\mathbf{u} + b) \text{ s.t. } (\mathbf{a}_i^T\mathbf{u} + b) \geq 0, \forall i \in \mathcal{S}, (\mathbf{a}_j^T\mathbf{u} + b) \leq 0, \forall j \in \mathcal{S}^c, \|\mathbf{u}\|_2 = 1,$$

which has the following dual form

$$\min_{\boldsymbol{\lambda},\boldsymbol{\nu}} \max_{\mathbf{u},b} \mathbf{u}^T \left(\sum_{i\in\mathcal{S}} (\alpha_i + \lambda_i)\mathbf{a}_i - \sum_{j\in\mathcal{S}^c} \nu_j \mathbf{a}_j \right) \text{ s.t. } \boldsymbol{\lambda}, \boldsymbol{\nu} \succcurlyeq \mathbf{0}, \sum_{i\in\mathcal{S}} (\alpha_i + \lambda_i) = \sum_{j\in\mathcal{S}^c} \nu_j, \|\mathbf{u}\|_2 = 1.$$

Thus, we obtain the following neuron and bias choice for the extreme point

$$\mathbf{u}_\alpha = \frac{\sum_{i\in\mathcal{S}} (\alpha_i + \lambda_i)\mathbf{a}_i - \sum_{j\in\mathcal{S}^c} \nu_j \mathbf{a}_j}{\| \sum_{i\in\mathcal{S}} (\alpha_i + \lambda_i)\mathbf{a}_i - \sum_{j\in\mathcal{S}^c} \nu_j \mathbf{a}_j \|_2} \text{ and } b_\alpha = \begin{cases} \max_{i\in\mathcal{S}}(-\mathbf{a}_i^T\mathbf{u}), & \text{if } \sum_{i\in\mathcal{S}} \alpha_i \leq 0 \\ \min_{j\in\mathcal{S}^c}(-\mathbf{a}_j^T\mathbf{u}), & \text{otherwise} \end{cases}.$$

□

Proof of Theorem 3.1 and Corollary 3.1. Using an indicator function, we can reformulate the problem as follows

$$P^* = \min_{\theta\in\Theta\setminus\{\mathbf{w}\}} \max_{\mathbf{v}} \mathbf{v}^T\mathbf{y} + \mathcal{I}(\|(\mathbf{A}\mathbf{U})_+^T\mathbf{v}\|_\infty \leq 1), \text{ s.t. } \|\mathbf{u}_j\|_2 \leq 1, \forall j,$$

where $\mathcal{I}(x \leq a) = 0$, if $x \leq a$, $\mathcal{I}(x \leq a) = -\infty$, otherwise. Since the set $\|(\mathbf{A}\mathbf{U})_+^T\mathbf{v}\|_\infty \leq 1$ is closed, the function $\Phi(\mathbf{v}, \mathbf{U}) := \mathbf{v}^T\mathbf{y} + \mathcal{I}(\|(\mathbf{A}\mathbf{U})_+^T\mathbf{v}\|_\infty \leq 1)$ is the sum of a linear function and an upper-semicontinuous indicator function and therefore upper-semicontinuous. The constraint over $\mathbf{U}$ is convex and compact. We use P^* to denote the value of the above min-max program. Exchanging the order of

min and max we obtain the dual problem given in [Main Paper,(11)], which establishes a lower bound D^* for the above problem:

$$P^* \geq D^* = \max_{\mathbf{v}} \min_{\theta \in \Theta \setminus \{\mathbf{w}\}} \mathbf{v}^T \mathbf{y} + \mathcal{I}(\|(\mathbf{A}\mathbf{U})_+^T \mathbf{v}\|_\infty \leq 1), \text{ s.t. } \|\mathbf{u}_j\|_2 \leq 1, \forall j,$$

We now show that strong duality holds for infinite size NNs. The dual of the semi-infinite program in [Main Paper, (11)] is given by (see Section 2.2 of (Goberna and López-Cerdá, 1998) and also (Bach, 2017))

$$\begin{aligned} &\min \|\boldsymbol{\mu}\|_{TV} \\ &\text{s.t.} \int_{\mathbf{u} \in \mathcal{B}_2} \big(\mathbf{A}\mathbf{u}\big)_+ d\boldsymbol{\mu}(\mathbf{u}) = \mathbf{y}\,, \end{aligned}$$

where TV is the total variation norm of the Radon measure $\boldsymbol{\mu}$. This expression coincides with the infinite-size neural network as given in Section 5, and therefore strong duality holds. Next we invoke the semi-infinite optimality conditions for the dual problem in [Main Paper,(11)], in particular we apply Theorem 7.2 of (Goberna and López-Cerdá, 1998). We first define the set

$$\mathbf{K} = \mathbf{cone}\left\{ \begin{pmatrix} s\big(\mathbf{A}\mathbf{u}\big)_+ \\ 1 \end{pmatrix}, \mathbf{u} \in \mathcal{B}_2, s \in \{-1, +1\}; \begin{pmatrix} \mathbf{0} \\ -1 \end{pmatrix} \right\}.$$

Note that $\mathbf{K}$ is the union of finitely many convex closed sets, since the function $\big(\mathbf{A}\mathbf{u}\big)_+$ can be expressed as the union of finitely many convex closed sets. Therefore the set $\mathbf{K}$ is closed. By Theorem 5.3 (Goberna and López-Cerdá, 1998), this implies that the set of constraints in [Main Paper,(11)] forms a Farkas-Minkowski system. By Theorem 8.4 of (Goberna and López-Cerdá, 1998), primal and dual values are equal, given that the system is consistent. Moreover, the system is discretizable, i.e., there exists a sequence of problems with finitely many constraints whose optimal values approach to the optimal value of [Main Paper,(11)]. The optimality conditions in Theorem 7.2 (Goberna and López-Cerdá, 1998) implies that $\mathbf{y} = \big(\mathbf{A}\mathbf{U}^*\big)_+ \mathbf{w}^*$ for some vector $\mathbf{w}^*$. Since the primal and dual values are equal, we have $\mathbf{v}^{*T}\mathbf{y} = \mathbf{v}^{*T}\big(\mathbf{A}\mathbf{U}^*\big)_+ \mathbf{w}^* = \|\mathbf{w}^*\|_1$, which shows that the primal-dual pair $(\{\mathbf{w}^*, \mathbf{U}^*\}, \mathbf{v}^*)$ is optimal. Thus, the optimal neuron weights $\mathbf{U}^*$ satisfy $\|(\mathbf{A}\mathbf{U}^*)_+^T \mathbf{v}^*\|_\infty = 1$. □

Proof of Proposition 3.1. Here, we particularly examine the problem in [Main Paper, (4)] when we have a one dimensional dataset, i.e., $\{a_i, y_i\}_{i=1}^n$. Then, [Main Paper, (4)] can be modified as

$$\min_{\theta \in \Theta} \|\mathbf{w}\|_1 \text{ s.t. } (\mathbf{a}\mathbf{u}^T + \mathbf{1}\mathbf{b}^T)_+ \mathbf{w} = \mathbf{y}, |u_j| \leq 1, \forall j. \tag{10}$$

Then, using Lemma 2.6, we can construct the following matrix

$$\mathbf{A}_e = (\mathbf{a}\mathbf{u}^{*^T} + \mathbf{1}\mathbf{b}^{*^T})_+,$$

where $\mathbf{u}^*$ and $\mathbf{b}^*$ consist of all possible extreme points. Using this definition and Corollary 3.2, we can rewrite (10) as

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 \text{ s.t. } \mathbf{A}_e \mathbf{w} = \mathbf{y}. \tag{11}$$

In the following, we first derive optimality conditions for (11) and then provide an analytic counter example to disprove uniqueness. Then, we also follow the same steps for the regularized version of (11).

Equality constraint: The optimality conditions for (11) are

$$\begin{aligned} &\mathbf{A}_e \mathbf{w}^* = \mathbf{y} \\ &\mathbf{A}_{e,s}^T \mathbf{v}^* + \text{sign}(\mathbf{w}_s^*) = 0 \\ &\|\mathbf{A}_{e,s^c}^T \mathbf{v}^*\|_\infty \leq 1, \end{aligned} \tag{12}$$

where the subscript s denotes the entries of a vector (or columns for matrices) that correspond to a nonzero weight, i.e. $w_i \neq 0$, and the subscript s^c denotes the remaining entries (or columns). We aim to find an optimal primal-dual pair that satisfies (12).

Now, let us consider a specific dataset, i.e., $\mathbf{a} = [-2\ -1\ 0\ 1\ 2]^T$ and $\mathbf{y} = [1\ -1\ 1\ 1\ -1]^T$, and yields the following

$$\mathbf{A}_e = (\mathbf{a}\mathbf{u}^{*^T} + \mathbf{1}\mathbf{b}^{*^T}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 0 & 0 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 0 & 0 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 0 & 0 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where $\mathbf{u}^{*^T} = [1\ 1\ 1\ 1\ -1\ -1\ -1\ -1]$ and $\mathbf{b}^{*^T} = [2\ 1\ 0\ -1\ -1\ 0\ 1\ 2]$. Solving (11) for this dataset gives

$$\mathbf{v}^* = \begin{bmatrix} 1 \\ -3 \\ 2 \\ 1 \\ -1 \end{bmatrix} \text{ and } \mathbf{w}^* = \begin{bmatrix} 0 \\ 6419/5000 \\ -3919/2500 \\ -8581/5000 \\ 13581/5000 \\ -1081/2500 \\ -1419/5000 \\ 0 \end{bmatrix} \implies \|\mathbf{w}^*\|_1 = 8.$$

We can also achieve the same objective value by using the following matrix

$$\hat{\mathbf{A}}_e = (\mathbf{a}\hat{\mathbf{u}}^T + \mathbf{1}\hat{\mathbf{b}}^T) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 2.5 & 4 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1.5 & 3 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0.5 & 2 \\ 3 & 2 & 1 & 0.5 & 0 & 0 & 0 & 1 \\ 4 & 3 & 2 & 1.5 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where $\hat{\mathbf{u}}^T = [1\ 1\ 1\ 1\ -1\ -1\ -1\ -1]$ and $\hat{\mathbf{b}}^T = [2\ 1\ 0\ -0.5\ -1\ 0\ 0.5\ 2]$. Solving (11) for this dataset yields

$$\hat{\mathbf{v}} = \begin{bmatrix} 1 \\ -11/4 \\ 5/4 \\ 7/4 \\ -5/4 \end{bmatrix} \text{ and } \hat{\mathbf{w}} = \begin{bmatrix} 0 \\ 4/3 \\ 0 \\ -10/3 \\ 8/3 \\ 0 \\ -2/3 \\ 0 \end{bmatrix} \implies \|\hat{\mathbf{w}}\|_1 = 8.$$

We also note that both solutions satisfy the optimality conditions in (12).

Regularized case: The regularized version of (11) is as follows

$$\min_{\mathbf{w}} \beta\|\mathbf{w}\|_1 + \frac{1}{2n}\|\mathbf{A}_e\mathbf{w} - \mathbf{y}\|_2^2, \tag{13}$$

where the optimal solution $\mathbf{w}^*$ satisfies

$$\begin{aligned} &\frac{1}{n}\mathbf{A}_{e,s}^T(\mathbf{A}_e\mathbf{w}^* - \mathbf{y}) + \beta\text{sign}(\mathbf{w}_s^*) = 0 \\ &\|\mathbf{A}_{e,s^c}^T(\mathbf{A}_e\mathbf{w}^* - \mathbf{y})\|_\infty \leq \beta n, \end{aligned} \tag{14}$$

where the subscript s denotes the entries of a vector (or columns for matrices) that correspond to a nonzero weight, i.e. $w_i \neq 0$, and the subscript s^c denotes the remaining entries (or columns). Now, let us consider a specific dataset, i.e., $\mathbf{a} = [-2 \ -1 \ 0 \ 1 \ 2]^T$ and $\mathbf{y} = [1 \ -1 \ 1 \ 1 \ -1]^T$. We then construct the following matrix

$$\mathbf{A}_e = (\mathbf{a}\mathbf{u}^{*^T} + \mathbf{1}\mathbf{b}^{*^T}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2.5 & 3 & 4 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1.5 & 2 & 3 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 2 \\ 3 & 2 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & 3 & 2 & 1.5 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where $\mathbf{u}^{*^T} = [1 \ 1 \ 1 \ 1 \ 1 \ -1 \ -1 \ -1 \ -1 \ -1]$ and $\mathbf{b}^{*^T} = [2 \ 1 \ 0 \ -0.5 \ -1 \ -1 \ 0 \ 0.5 \ 1 \ 2]$. For this dataset with $\beta = 0.1$, the optimal value of (13) can be achieved by the following solutions

$$\mathbf{w}_1 = \begin{bmatrix} 0 \\ 3197/2400 \\ -2497/1500 \\ 0 \\ -19997/12000 \\ 31961/12000 \\ -997/3000 \\ 0 \\ -3997/12000 \\ 0 \end{bmatrix} \implies \beta\|\mathbf{w}_1\|_1 + \frac{1}{2n}\|\mathbf{A}_e\mathbf{w}_1 - \mathbf{y}\|_2^2 = \frac{1999}{2500000}$$

$$\mathbf{w}_2 = \begin{bmatrix} 0 \\ 191823/140000 \\ -990613/840000 \\ -471683/420000 \\ -128017/120000 \\ 367547/140000 \\ -127357/840000 \\ -87827/420000 \\ -31993/120000 \\ 0 \end{bmatrix} \implies \beta\|\mathbf{w}_2\|_1 + \frac{1}{2n}\|\mathbf{A}_e\mathbf{w}_2 - \mathbf{y}\|_2^2 = \frac{1999}{2500000},$$

where each solution satisfies the optimality conditions in (14). We also provide a visualization for the output functions of each solution in Figure 2.

□

Proof of Lemma 3.1. Since $\mathbf{y}$ has both positive and negative entries, we need at least two $\mathbf{u}$'s with positive and negative output weights to represent $\mathbf{y}$ using the output range of ReLU. Therefore the optimal value of the ℓ_0 problem is at least 2. Note that $\mathbf{A}\mathbf{A}^\dagger = \mathbf{I}_n$ since $\mathbf{A}$ is full row rank. Then let us define the output weights

$$\begin{aligned} w_1 &= \|\mathbf{A}^\dagger(\mathbf{y})_+\|_2 \\ w_2 &= -\|\mathbf{A}^\dagger(-\mathbf{y})_+\|_2 . \end{aligned}$$

Then note that

$$\begin{aligned} w_1(\mathbf{A}\mathbf{u}_1)_+ + w_2(\mathbf{A}\mathbf{u}_2)_+ &= (\mathbf{A}\mathbf{A}^\dagger(\mathbf{y})_+)_+ - (\mathbf{A}\mathbf{A}^\dagger(-\mathbf{y})_+)_+ \\ &= ((\mathbf{y})_+)_+ - ((-\mathbf{y})_+)_+ \\ &= (\mathbf{y})_+ - (-\mathbf{y})_+ \\ &= \mathbf{y} \end{aligned}$$

where the second equality follows from $\mathbf{A}\mathbf{A}^\dagger = \mathbf{I}_n$. □

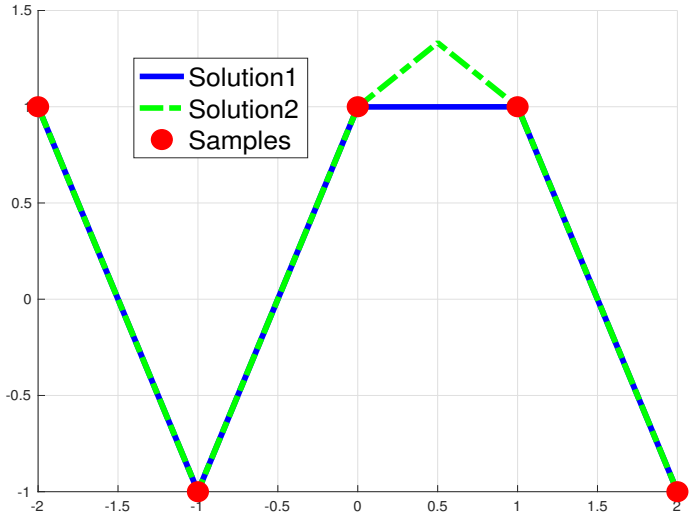


Figure 2: Solutions provided by $\mathbf{w}_1$ and $\mathbf{w}_2$ for the problem in (13).

Proof of Lemma 3.2. We first provide the optimality conditions for the convex program in the following proposition:

Proposition 1. *Let* $\mathbf{U}$ *be a weight matrix for [Main Paper, (4)]. Then,* $\mathbf{U} \in \mathbb{R}^{d\times m}$ *is an optimal solution for the regularized training problem if*

$$\exists \boldsymbol{\alpha} \in \mathbb{R}^n, \mathbf{w} \in \mathbb{R}^m \ s.t. \ \left(\mathbf{AU}\right)_+ \mathbf{w} = \mathbf{y}, \quad \left(\mathbf{AU}\right)_+^T \boldsymbol{\alpha} = sign(\mathbf{w}) \tag{15}$$

and

$$\max_{\mathbf{u}\,:\,\|\mathbf{u}\|_2 \leq 1} |\boldsymbol{\alpha}^T (\mathbf{Au})_+| \leq 1\,. \tag{16}$$

These conditions follow from linear semi-infinite optimality conditions given in Theorem 7.1 and 7.6 (Goberna and López-Cerdá, 1998) for Farkas-Minkowski systems. Then the proof Lemma 3.2 directly follows from the solution of minimum cardinality problem given in Lemma 3.1.

Now we prove the second claim. For whitened data matrices, denoting the Singular Value Decomposition of the input data as $\mathbf{A} = \mathbf{U}$ where $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_n$ since $\mathbf{A}$ is assumed full row rank. Consider the dual optimization problem

$$\max_{|\mathbf{v}^T(\mathbf{Au})_+| \leq 1,\ \forall \mathbf{u} \in \mathcal{B}_2} \mathbf{v}^T\mathbf{y} \tag{17}$$

Changing the variable to $\mathbf{u}' = \mathbf{Uu}$ in the dual problem we next show that

$$\max_{|\mathbf{v}^T(\mathbf{u}')_+| \leq 1,\ \forall \mathbf{u}' \in \mathcal{B}_2} \mathbf{v}^T\mathbf{y} = \max_{\|(\mathbf{v})_+\|_2 \leq 1,\ \|(-\mathbf{v})_+\|_2 \leq 1} \mathbf{v}^T\mathbf{y}\,. \tag{18}$$

where the equality follows from the upper bound

$$\mathbf{v}^T(\mathbf{u}')_+ \leq (\mathbf{v})_+^T(\mathbf{u}')_+ \leq \|(\mathbf{v})_+\|_2 \|(\mathbf{u}')_+\|_2 \leq \|(\mathbf{v})_+\|_2\,,$$

which is achieved when $\mathbf{u}' = \frac{(\mathbf{v})_+}{\|(\mathbf{v})_+\|_2}$. Similarly we have

$$-\mathbf{v}^T(\mathbf{u}')_+ \leq (-\mathbf{v})_+^T(\mathbf{u}')_+ \leq \|(-\mathbf{v})_+\|_2 \|(\mathbf{u}')_+\|_2 \leq \|(-\mathbf{v})_+\|_2\,,$$

which is achieved when $\mathbf{u}' = \frac{(-\mathbf{v})_+}{\|(-\mathbf{v})_+\|_2}$, which verifies the right-hand-side of (18). Now note that

$$\mathbf{v}^T\mathbf{y} \leq (\mathbf{v})_+^T(\mathbf{y})_+ + (-\mathbf{v})_+^T(-\mathbf{y})_+ \,.$$

Therefore the right-hand-side of (18) is upper-bounded by $\|(\mathbf{y})_+\|_2 + \|(-\mathbf{y})_+\|_2$. This upper-bound is achieved by the choice

$$\mathbf{v} = \frac{(\mathbf{y})_+}{\|(\mathbf{y})_+\|_2} - \frac{(-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2} \,,$$

since we have

$$\begin{aligned}\mathbf{v}^T\mathbf{y} &= \frac{\mathbf{y}^T(\mathbf{y})_+}{\|(\mathbf{y})_+\|_2} - \frac{\mathbf{y}^T(-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2} = \frac{(\mathbf{y})_+^T(\mathbf{y})_+}{\|(\mathbf{y})_+\|_2} + \frac{(-\mathbf{y})_+^T(-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2} \\ &= \|(\mathbf{y})_+\|_2 + \|(-\mathbf{y})_+\|_2 \,.\end{aligned}$$

Therefore the preceding choice of $\mathbf{v}$ is optimal. Consequently, the corresponding optimal neuron weights satisfy

$$\mathbf{u}_1' = \frac{(\mathbf{y})_+}{\|(\mathbf{y})_+\|_2} \quad \text{and} \quad \mathbf{u}_2' = \frac{(-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2} \,.$$

Changing the variable back via $\mathbf{u} = \mathbf{U}^T\mathbf{u}' = \mathbf{A}^\dagger\mathbf{u}'$ we conclude that the optimal neurons are given by

$$\mathbf{u}_1 = \frac{\mathbf{A}^\dagger(\mathbf{y})_+}{\|(\mathbf{y})_+\|_2} \quad \text{and} \quad \mathbf{u}_2 = \frac{\mathbf{A}^\dagger(-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2} \,,$$

or equivalently

$$\mathbf{u}_1 = \frac{\mathbf{A}^\dagger(\mathbf{y})_+}{\|\mathbf{A}^\dagger(\mathbf{y})_+\|_2} \quad \text{and} \quad \mathbf{u}_2 = \frac{\mathbf{A}^\dagger(-\mathbf{y})_+}{\|\mathbf{A}^\dagger(-\mathbf{y})_+\|_2} \,,$$

since $\mathbf{A}^\dagger$ is orthonormal and yields the claimed expression. Finally, note that the corresponding output weights are $\|\mathbf{A}^\dagger(\mathbf{y})_+\|_2$ and $\|\mathbf{A}^\dagger(-\mathbf{y})_+\|_2$, respectively. □

Proof of Proposition 3.2. Since the constraint in [Main Paper,(14)] is bounded below and the hidden layer weights are constrained to the unit Euclidean ball, the convergence of the cutting plane method directly follows from Theorem 11.2 of (Goberna and López-Cerdá, 1998). □

Proof of Theorem 3.2. Given a vector $\mathbf{u}$ we partition $\mathbf{A}$ according to the subset $S = \{i | \mathbf{a}_i^T\mathbf{u} \geq 0\}$, where $\mathbf{A}_S\mathbf{u} \succcurlyeq 0$ and $-\mathbf{A}_{S^c}\mathbf{u} \succcurlyeq 0$ into

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_S \\ \mathbf{A}_{S^c} \end{bmatrix} .$$

Here $\mathbf{A}_S$ is the sub-matrix of $\mathbf{A}$ consisting of the rows indexed by S, and S^c is the complement of the set S. Consequently, we partition the vector $\big(\mathbf{A}\mathbf{u}\big)_+$ as follows

$$\big(\mathbf{A}\mathbf{u}\big)_+ = \begin{bmatrix} \mathbf{A}_S\mathbf{u} \\ \mathbf{0} \end{bmatrix} .$$

Then we use the block matrix pseudo-inversion formula (Baksalary and Baksalary, 2007)

$$\mathbf{A}^\dagger = \begin{bmatrix} \big(\mathbf{A}_S\mathbf{P}_{S^c}^\perp\big)^\dagger & \big(\mathbf{A}_{S^c}\mathbf{P}_S^\perp\big)^\dagger \end{bmatrix} ,$$

where $\mathbf{P}_S$ and $\mathbf{P}_{S^c}$ are projection matrices defined as follows

$$\begin{aligned}\mathbf{P}_S &= \mathbf{I}_d - \mathbf{A}_S^T\big(\mathbf{A}_S\mathbf{A}_S^T\big)^{-1}\mathbf{A}_S \\ \mathbf{P}_{S^c} &= \mathbf{I}_d - \mathbf{A}_{S^c}^T\big(\mathbf{A}_{S^c}\mathbf{A}_{S^c}^T\big)^{-1}\mathbf{A}_{S^c} \,.\end{aligned}$$

Note that the matrices $\mathbf{A}_S\mathbf{A}_S^T \in \mathbb{R}^{|S|\times|S|}$, $\mathbf{A}_{S^c}\mathbf{A}_{S^c}^T \in \mathbb{R}^{|S^c|\times|S^c|}$ are full column rank with probability one since the matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ is i.i.d. Gaussian where $n < d$. Hence the inverses $\left(\mathbf{A}_S\mathbf{A}_S^T\right)^{-1}$ and $\left(\mathbf{A}_{S^c}\mathbf{A}_{S^c}^T\right)^{-1}$ exist with probability one. Plugging in the above representation in the spike-free condition we get

$$\mathbf{A}^\dagger\left(\mathbf{A}\mathbf{u}\right)_+ = \left(\mathbf{A}_S\mathbf{P}_{S^c}^\perp\right)^\dagger \mathbf{A}_S\mathbf{u}\,.$$

Then we can express the probability of the matrix being spike-free as

$$\begin{aligned}\mathbb{P}\left[\max_{\mathbf{u}\in\mathcal{B}_2} \|\mathbf{A}^\dagger\left(\mathbf{A}\mathbf{u}\right)_+\|_2 > 1\right] &= \mathbb{P}\left[\exists \mathbf{u}\in\mathcal{B}_2 \mid \|\mathbf{A}^\dagger\left(\mathbf{A}\mathbf{u}\right)_+\|_2 > 1\right] \\ &\le \mathbb{P}\left[\exists \mathbf{u}\in\mathcal{B}_2, S\subseteq[n] \mid \|\left(\mathbf{A}_S\mathbf{P}_{S^c}^\perp\right)^\dagger \mathbf{A}_S\mathbf{u}\|_2 > 1\right].\end{aligned}$$

Finally, observe that $\mathbf{P}_{S^c} \in \mathbb{R}^{d\times d}$ is a uniformly random projection matrix of subspace of dimension $|S| \le n$. Therefore as $d\to\infty$, we have $\mathbf{P}_{S^c}^\perp \to \mathbf{I}_d$, and consequently

$$\lim_{d\to\infty} \|\left(\mathbf{A}_S\mathbf{P}_{S^c}^\perp\right)^\dagger \mathbf{A}_S\mathbf{u}\|_2 = \|\mathbf{A}_S^\dagger\mathbf{A}_S\mathbf{u}\|_2\,,$$

with probability one, and we have

$$\lim_{d\to\infty} \mathbb{P}\left[\exists \mathbf{u}\in\mathcal{B}_2, S\subseteq[n] \mid \|\left(\mathbf{A}_S\mathbf{P}_{S^c}^\perp\right)^\dagger \mathbf{A}_S\mathbf{u}\|_2 > 1\right] = 0\,.$$

□

Proof of Theorem 3.3. Since each sample $\mathbf{a}_j$ is a vertex of $\mathcal{C}_a$, we can find a separating hyperplane defined by the parameters $(\mathbf{u}_j, b_j)$ so that $\mathbf{a}_j^T\mathbf{u}_j + b_j > 0$ and $\mathbf{a}_i^T\mathbf{u}_j + b_j \le 0, \forall i \ne j$. Then, choosing $\{(\mathbf{u}_j, b_j)\}_{j=1}^n$ yields that $(\mathbf{A}\mathbf{U} + \mathbf{1}\mathbf{b}^T)_+$ is a diagonal matrix. Using these hidden neurons, we write the constraint of [Main Paper,(4)] in a more compact form as

$$(\mathbf{A}\mathbf{U} + \mathbf{1}\mathbf{b}^T)_+\mathbf{w} = \mathbf{y},$$

which is a least squares problem with a full rank square data matrix. Therefore, selecting $\mathbf{w} = ((\mathbf{A}\mathbf{U} + \mathbf{1}\mathbf{b}^T)_+)^\dagger\mathbf{y}$ along with $\mathbf{U}$ and $\mathbf{b}$ achieves a feasible solution for the original problem, i.e., 0 training error. □

Proof of Theorem 3.4. Let us define the distance of the i^{th} sample vector to the convex hull of the remaining sample vectors as d_i

$$\begin{aligned}d_i \triangleq \min_{\substack{\mathbf{z}\in\mathbb{R}^n:\\ \sum_{j\ne i} z_j = 1\\ \mathbf{z}\succcurlyeq 0}} \|\mathbf{a}_i - \sum_{j\ne i}\mathbf{a}_j z_j\|_2 &= \min_{\substack{\mathbf{z}\in\mathbb{R}^n:\\ \sum_{j\ne i} z_j = 1\\ \mathbf{z}\succcurlyeq 0, z_i = -1}} \|\mathbf{A}^T\mathbf{z}\|_2 \\ &= \min_{\substack{\mathbf{z}\in\mathbb{R}^n:\\ \sum_{j\ne i} z_j = 1\\ \mathbf{z}\succcurlyeq 0, z_i = -1}} \max_{\mathbf{v}:\|\mathbf{v}\|_2\le 1} \mathbf{v}^T\mathbf{A}^T\mathbf{z}\end{aligned}$$

Using Gordon's escape from a mesh theorem (Gordon, 1988; Ledoux and Talagrand, 2013), we obtain the following lower-bound on the expectation of d_i

$$\begin{aligned}\mathbb{E}\, d_i &\ge \mathbb{E} \min_{\substack{\mathbf{z}\in\mathbb{R}^n:\\ \sum_{j\ne i} z_j = 1\\ z_j\ge 0, z_i = -1}} \max_{\mathbf{v}:\|\mathbf{v}\|_2\le 1} \mathbf{h}^T\mathbf{v}\|\mathbf{z}\|_2 + \mathbf{z}^T\mathbf{g} \\ &= \mathbb{E} \min_{\substack{\mathbf{z}\in\mathbb{R}^n:\\ \sum_{j\ne i} z_j = 1\\ z_j\ge 0, z_i = -1}} ||\mathbf{h}||_2\|\mathbf{z}\|_2 + \mathbf{z}^T\mathbf{g} \\ &\ge \sqrt{d}\|\mathbf{z}\|_2 - \mathbb{E}\max_{j\in[n], j\ne i} g_j + g_i \\ &\ge \frac{\sqrt{d}}{\sqrt{n}} - \sqrt{2\log(n-1)}\,, \end{aligned} \tag{19}$$

where $\mathbf{h} \in \mathbb{R}^d$ and $\mathbf{g} \in \mathbb{R}^n$ are random vectors with i.i.d. standard Gaussian components, and the second inequality follows from a well-known result on finite Gaussian suprema (Ledoux and Talagrand, 2013). Therefore, the expected distance of the i^{th} sample to the convex hull is guaranteed to be positive whenever $d > 2n\log(n-1)$. Note that the lower bound (19) is vacuous for $d < 2n\log(n-1)$ since the random variable d_i can only take non-negative values.

The distance d_i is a Lipschitz function of the random Gaussian matrix $\mathbf{A}$. This can be seen via the following argument

$$\begin{aligned}
\min_{\substack{\mathbf{z}\in\mathbb{R}^n:\\ \sum_{j\neq i} z_j = 1\\ z_j\geq 0, z_i = -1}} \|\mathbf{A}^T\mathbf{z}\|_2 - \min_{\substack{\mathbf{z}\in\mathbb{R}^n:\\ \sum_{j\neq i} z_j = 1\\ z_j\geq 0, z_i = -1}} \|\tilde{\mathbf{A}}^T\mathbf{z}\|_2 &\leq \min_{\substack{\mathbf{z}\in\mathbb{R}^n:\\ \sum_{j\neq i} z_j = 1\\ z_j\geq 0, z_i = -1}} \|\big(\mathbf{A} - \tilde{\mathbf{A}}^T\big)\mathbf{z}\|_2 \\
&\leq \|(\mathbf{A} - \tilde{\mathbf{A}})\|_2 \max_{\substack{\mathbf{z}\in\mathbb{R}^n:\\ \sum_{j\neq i} z_j = 1\\ z_j\geq 0, z_i = -1}} \|\mathbf{z}\|_2 \\
&\leq \|(\mathbf{A} - \tilde{\mathbf{A}})\|_F \max_{\substack{\mathbf{z}\in\mathbb{R}^n:\\ \sum_{j\neq i} z_j = 1\\ z_j\geq 0, z_i = -1}} \|\mathbf{z}\|_1 \\
&\leq 2\|(\mathbf{A} - \tilde{\mathbf{A}})\|_F
\end{aligned}$$

Applying the Lipschitz concentration for Gaussian measure (Ledoux and Talagrand, 2013) yields that

$$\mathbb{P}\big[d_i > \sqrt{d} - \sqrt{2n\log(n-1)} - t\big] \geq 1 - 2e^{-t^2/2}\,.$$

Therefore, we have $d_i > 0$ for $d > 2n\log(n-1)$ with probability exceeding $1 - 2e^{-t^2/2}$. Taking a union bound over every index $i \in \{0, ..., n\}$, we can upper-bound the failure probability by $2ne^{-t^2/2}$. Choosing $t^2 = 4\log(n-1)$ will yield a failure probability $O(1/n)$ and conclude the proof. □

Proof of Theorem 4.1. The proof follows from a similar argument as in the proof of Theorem 3.1, and is omitted. □

Proof of Theorem 5.1. The proof follows from a similar argument as in the proof of Theorem 3.1, and is omitted. □

7 Polar convex duality

In this section we derive the polar duality and present a connection to minimum ℓ_1 solutions to linear systems. Recognizing the constraint $\mathbf{v} \in \mathcal{Q}_{\mathbf{A}}$ can be stated as

$$\mathbf{v} \in \mathcal{Q}_{\mathbf{A}}^{\circ},\, \mathbf{v} \in -\mathcal{Q}_{\mathbf{A}}^{\circ}\,,$$

which is equivalent to

$$\mathbf{v} \in \mathcal{Q}_{\mathbf{A}}^{\circ} \cap -\mathcal{Q}_{\mathbf{A}}^{\circ}\,.$$

Note that the support function of a set can be expressed as the gauge function of its polar set (see e.g. (Rockafellar, 1970)). The polar set of $\mathcal{Q}_{\mathbf{A}}^{\circ} \cap -\mathcal{Q}_{\mathbf{A}}^{\circ}$ is given by

$$\big(\mathcal{Q}_{\mathbf{A}}^{\circ} \cap -\mathcal{Q}_{\mathbf{A}}^{\circ}\big)^{\circ} = \operatorname{conv} \mathcal{Q}_{\mathbf{A}} \cup -\mathcal{Q}_{\mathbf{A}}\,.$$

Using this fact, we express the dual problem [Main Paper, (11)] as

$$\begin{aligned}
D^* = \inf_{t\in\mathbb{R}}\ & t \\
\text{s.t. }\ & \mathbf{y} \in t\operatorname{conv}\left\{\mathcal{Q}_{\mathbf{A}} \cup -\mathcal{Q}_{\mathbf{A}}\right\},
\end{aligned} \tag{20}$$

where conv represents the convex hull of a set.

Let us restate dual of the two-layer ReLU neural network training problem given by

$$\max_{\mathbf{v}} \mathbf{v}^T \mathbf{y} \text{ s.t. } \mathbf{v} \in \mathcal{Q}_{\mathbf{A}}^{\circ} \ , \ -\mathbf{v} \in \mathcal{Q}_{\mathbf{A}}^{\circ} \tag{21}$$

where $\mathcal{Q}_{\mathbf{A}}^{\circ}$ is the polar dual of $\mathcal{Q}_{\mathbf{A}}$ defined as $\mathcal{Q}_{\mathbf{A}}^{\circ} = \{\mathbf{v} | \mathbf{v}^T \mathbf{u} \leq 1 \, \forall \mathbf{u} \in \mathcal{Q}_{\mathbf{A}}\}$.

Remark 1. *The dual problem given in* (21) *is analogous to the convex duality in minimum ℓ_1 norm solutions to linear systems. In particular, for the latter it holds that*

$$\min_{\mathbf{w} \, : \, \mathbf{A}\mathbf{w} = \mathbf{y}} \|\mathbf{w}\|_1 = \max_{\mathbf{v} \in \text{conv}\{\hat{\mathbf{a}}_1, ..., \hat{\mathbf{a}}_d\}^{\circ}, \ -\mathbf{v} \in \text{conv}\{\hat{\mathbf{a}}_1, ..., \hat{\mathbf{a}}_d\}^{\circ}} \mathbf{v}^T \mathbf{y} \, ,$$

where $\hat{\mathbf{a}}_1, ..., \hat{\mathbf{a}}_d$ are the columns of $\mathbf{A}$. The above optimization problem can also be put in the gauge optimization form as follows.

$$\min_{\mathbf{w} \, : \, \mathbf{A}\mathbf{w} = \mathbf{y}} \|\mathbf{w}\|_1 = \inf_{t \in \mathbb{R}} \ t \ \textit{s.t.} \ \mathbf{y} \in t \, \text{conv}\{\pm \hat{\mathbf{a}}_1, ... \hat{\mathbf{a}}_d\},$$

which parallels the gauge optimization form in (20).

8 Two-layer ReLU networks with general loss functions

Now we consider the scalar output two-layer ReLU networks with an arbitrary loss function

$$\min_{\theta \in \Theta} \ell((\mathbf{A}\mathbf{U})_+ \mathbf{w}, \mathbf{y}) + \beta \|\mathbf{w}\|_1 \text{ s.t. } \|\mathbf{u}_j\|_2 \leq 1, \forall j, \tag{22}$$

where $\ell(\cdot, \mathbf{y})$ is a convex loss function.

Theorem 1. *The dual of* (22) *is given by*

$$\max_{\mathbf{v}} -\ell^*(\mathbf{v}) \ \textit{s.t.} \ \mathbf{v} \in \beta \mathcal{Q}_{\mathbf{A}}^{\circ}, -\mathbf{v} \in \beta \mathcal{Q}_{\mathbf{A}}^{\circ} \, ,$$

where ℓ^* is the Fenchel conjugate function defined as

$$\ell^*(\mathbf{v}) = \max_{\mathbf{z}} \mathbf{z}^T \mathbf{v} - \ell(\mathbf{z}, \mathbf{y}) \, .$$

The proof follows from classical Fenchel duality (Boyd and Vandenberghe, 2004), and a similar argument as in the proof of Theorem 3.1. We omit the details. The general form of the dual can be easily extended to vector output networks.

9 Extension to vector output neural networks

In this section, we describe the implementation of the cutting plane algorithm when we have vector outputs, particularly, o outputs. In this case, we have $\mathbf{Y} \in \mathbb{R}^{n \times o}$ and $f(\mathbf{A}) = \big(\mathbf{A}\mathbf{U}\big)_+ \mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{m \times o}$. Then, we formulate the following dual problem

$$\max_{\mathbf{V}} \mathbf{tr}(\mathbf{V}^T \mathbf{Y}) \text{ s.t. } \|\mathbf{V}^T \big(\mathbf{A}\mathbf{u}\big)_+\|_{\infty} \leq 1 \, , \forall \mathbf{u} \in \mathcal{B}_2$$

and an optimal $\mathbf{U}$ satisfies

$$\|(\mathbf{A}\mathbf{U}^*)_+^T \mathbf{V}^*\|_{\infty} = 1 \, ,$$

where $\mathbf{V}^*$ is the optimal dual variable and $\mathbf{tr}$ represents the trace of a matrix. Note that we can also consider block ℓ_1-ℓ_2 norms and their duals in formulating the vector output objective. We use this particular form as it admits a simpler solution with the cutting-plane method.

We again relax the problem using the spike-free relaxation and then we solve the following problem for each $k \in [o]$

$$\hat{\mathbf{u}}_{k,1} = \underset{\mathbf{u}}{\operatorname{argmax}}\, \mathbf{v}_k^T \mathbf{A}\mathbf{u} \text{ s.t. } \mathbf{A}\mathbf{u} \succcurlyeq \mathbf{0}, \|\mathbf{u}\|_2 \leq 1$$
$$\hat{\mathbf{u}}_{k,2} = \underset{\mathbf{u}}{\operatorname{argmin}}\, \mathbf{v}_k^T \mathbf{A}\mathbf{u} \text{ s.t. } \mathbf{A}\mathbf{u} \succcurlyeq \mathbf{0}, \|\mathbf{u}\|_2 \leq 1,$$

where $\mathbf{v}_k$ is the k^{th} column of $\mathbf{V}$. After solving these optimization problems, we select the two neurons that achieve the maximum and minimum objective value among o neurons for each problem. Thus, we can find the weights for the hidden layers using convex optimization.

Consider the minimal cardinality problem

$$\min_{\theta \in \Theta} \|\mathbf{W}\|_0 \text{ s.t. } f_\theta(\mathbf{A}) = \mathbf{Y}, \|\mathbf{u}_j\|_2 = 1, \forall j.$$

The following result provides a characterization of the optimal solutions to the above problem

Lemma 1. *Suppose that $n \leq d$, $\mathbf{A}$ is full row rank, and $\mathbf{Y} \in \mathbb{R}_+^{n \times o}$, e.g., one hot encoded outputs for multiclass classification and we have at least one sample in each class. Then an optimal solution to [Main Paper, (12)] is given by*

$$\mathbf{u}_k = \frac{\mathbf{A}^\dagger \left(\mathbf{y}_k\right)_+}{\|\mathbf{A}^\dagger \left(\mathbf{y}_k\right)_+\|_2} \text{ and } \mathbf{w}_k = \|\mathbf{A}^\dagger \left(\mathbf{y}_k\right)_+\|_2 \mathbf{e}_k$$

for each $k \in [o]$, where $\mathbf{w}_k$ and $\mathbf{y}_k$ are the k^{th} row and column of $\mathbf{W}$ and $\mathbf{Y}$, respectively.

Proof. The proof is a straightforward generalization of the scalar output case in Lemma 3.1. □

Lemma 2. *We have ℓ_1-ℓ_0 equivalence if the following condition holds*

$$\min_{\mathbf{v}: \mathbf{v}^T \left(\mathbf{A}\mathbf{u}_k\right)_+ = 1, \forall k} \; \max_{\mathbf{u}: \mathbf{u} \in \mathcal{B}_2} \mathbf{v}^T \left(\mathbf{A}\mathbf{u}\right)_+ \leq 1 \,.$$

Proof. The proof is a straightforward generalization of the scalar output case in Lemma 3.2. □

References

20 newsgroups. `http://qwone.com/~jason/20Newsgroups/`.

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

Jerzy K Baksalary and Oskar Maria Baksalary. Particular formulae for the moore–penrose inverse of a columnwise partitioned matrix. *Linear algebra and its applications*, 421(1):16–23, 2007.

Yoshua Bengio, Nicolas L Roux, Pascal Vincent, Olivier Delalleau, and Patrice Marcotte. Convex neural networks. In *Advances in neural information processing systems*, pages 123–130, 2006.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Adam Coates and Andrew Y Ng. Learning feature representations with k-means. In *Neural networks: Tricks of the trade*, pages 561–580. Springer, 2012.

Miguel Angel Goberna and Marco López-Cerdá. *Linear semi-infinite optimization*. 01 1998. doi: 10.1007/978-1-4899-8044-1_3.

Yehoram Gordon. On milman's inequality and random subspaces which escape through a mesh in rn. In *Geometric Aspects of Functional Analysis*, pages 84–106. Springer, 1988.

Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, March 2014.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The CIFAR-10 dataset. http://www.cs.toronto.edu/kriz/cifar.html, 2014.

Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992.

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

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 1964.

Andreas Themelis and Panagiotis Patrinos. Supermann: a superlinearly convergent algorithm for finding fixed points of nonexpansive operators. *IEEE Transactions on Automatic Control*, 2019.

L. Torgo. Regression data sets. http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html.

E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, June 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. On the margin theory of feedforward neural networks. *arXiv preprint arXiv:1810.05369*, 2018.

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