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
2 Additional details on the numerical experiments
3 Cutting plane algorithm with no bias term
4 Cutting plane algorithm with a bias term
5 Infinite size neural networks
6 Proofs of the main results
7 Polar convex duality
8 Two-layer ReLU networks with general loss functions
9 Extension to vector output neural networks
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 $U$ 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 $U$. 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)**

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

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

$$\max_{v \in \mathbb{R}^n, \|v\|_2 = 0} v^T y \text{ s.t. } |v^T (Au + b1)_+| \leq 1, \forall u \in B_2, \forall b \in \mathbb{R} \quad (1)$$

and an optimal $(U^*, b^*)$ satisfies

$$\|(AU^* + 1b^*^T)^T v^*\|_\infty = 1,$$

where $v^*$ 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

$$u_1^* = \arg\max_{u, b} v^T (Au + b1)_+ \text{ s.t. } \|u\|_2 \leq 1$$

$$u_2^* = \arg\min_{u, b} v^T (Au + b1)_+ \text{ s.t. } \|u\|_2 \leq 1.$$ 

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

$$(\hat{u}_1, \hat{b}_1) = \arg\max_{u, b} v^T Au + bv^T 1 \text{ s.t. } Au + b1 \succ 0, \|u\|_2 \leq 1$$

$$(\hat{u}_2, \hat{b}_2) = \arg\min_{u, b} v^T Au + bv^T 1 \text{ s.t. } Au + b1 \succ 0, \|u\|_2 \leq 1,$$
where we relax the set \( \{(Au + b_1)|u \in \mathbb{R}^d, \|u\|_2 \leq 1\} \) as \( \{(Au + b_1)|u \in \mathbb{R}^d, \|u\|_2 \leq 1\} \cap \mathbb{R}^n_+ \). Now, we can find the weights and biases for the hidden layer using convex optimization. However, notice that depending on the sign of \( 1^T v \) one of the problems will be unbounded. Thus, if \( 1^T v \neq 0 \), 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 \( 1^T v = 0 \) 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 \( U \) as a column and to \( b \) 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 \( U \). We repeat this procedure until the optimality conditions are satisfied (see Algorithm 2 for the pseudocode). Since the constraint is bounded below and \( \hat{u}_j \)'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 \( v \) such that \( 1^T v = 0 \)
2: while there exists a violating neuron do
3: Find \( \hat{u}_1, \hat{u}_2, \hat{b}_1 \) and \( \hat{b}_2 \)
4: \( U \leftarrow [U \hat{u}_1 \hat{u}_2] \)
5: \( b \leftarrow [b^T \hat{b}_1 \hat{b}_2]^T \)
6: Find \( v \) using the dual problem
7: Check the existence of a violating neuron
8: Solve the problem using \( U \) and \( b \)
9: Return \( \theta = (U, b, w) \)

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 \( X \) with a set of continuous basis functions \( \phi_u : X \rightarrow \mathbb{R} \) parametrized by \( u \in B_2 \). We then consider real-valued Radon measures equipped with the uniform norm (Rudin, 1964). For a signed Radon measure \( \mu \), we define the infinite size neural network output for the input \( x \in X \) as

\[
f(x) = \int_{u \in B_2} \phi_u(x) d\mu(u).
\]

The total variation norm of the signed measure \( \mu \) is defined as the supremum of \( \int_{u \in B_2} q(u) d\mu(u) \) over all continuous functions \( q(u) \) that satisfy \( |q(u)| \leq 1 \). Now we consider the ReLU basis functions \( \phi_u(x) = (x^T u)_{+} \). For finitely many neurons, the network output is given by

\[
f(x) = \sum_{j=1}^{m} \phi_{u_j}(x) w_j,
\]

which corresponds to the signed measure \( \mu = \sum_{j=1}^{m} w_j \delta(u - u_j) \) where \( \delta \) is the Dirac delta function. And the total variation norm \( \|\mu\|_{TV} \) of \( \mu \) reduces to the \( \ell_1 \) norm \( \|w\|_1 \).

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

\[
\min \|\mu\|_{TV}
\]

s.t. \( f(x_i) = y_i, \forall i \in [n] \).
For finitely many neurons, i.e., when the measure \( \mu \) is a mixture of Dirac delta basis functions, the equivalent problem is

\[
\min \| w \|_1 \\
\text{s.t. } f(x_i) = y_i, \forall i \in [n].
\]

which is identical to [Main Paper, (3)]. 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 \( \tilde{u}_j = \alpha_j u_j, \tilde{b}_j = \alpha_j b_j \) and \( \tilde{w}_j = w_j / \alpha_j \), for any \( \alpha_j > 0 \). Then, [Main Paper, (1)] becomes

\[
\bar{f}_\theta(A) = \sum_{j=1}^{m} \tilde{w}_j (A\tilde{u}_j + \tilde{b}_j \mathbf{1})_+ = \sum_{j=1}^{m} \frac{w_j}{\alpha_j} (\alpha_j A u_j + \alpha_j b_j \mathbf{1})_+ = \sum_{j=1}^{m} w_j (A u_j + b_j \mathbf{1})_+,
\]

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

\[
\frac{1}{2} \sum_{j=1}^{m} (w_j^2 + \|u_j\|^2) \geq \sum_{j=1}^{m} (|w_j| \|u_j\|_2),
\]

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

Proof of Lemma 2.2. Consider the following problem

\[
\min_{\tilde{\theta} \in \tilde{\Theta}} \| w \|_1 \text{ s.t. } \tilde{f}_\tilde{\theta}(A) = y, \| u_j \|_2 \leq 1, \forall j,
\]

where the unit norm equality constraint is relaxed. Let us assume that for a certain index \( j \), we obtain \( \|u_j\|_2 < 1 \) 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 \( u_j \), and hence removing the constraint for \( u_j \) will not change the optimal solution. However, when we remove the constraint, \( \|u_j\|_2 \to \infty \) 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 \( \{ (A u)_+ | u \in B_2 \} \subseteq AB_2 \). Since we also have \( \{ (A u)_+ | u \in B_2 \} \subseteq \mathbb{R}_+^n \), it holds that \( \{ (A u)_+ | u \in B_2 \} \subseteq AB_2 \cap \mathbb{R}_+^n \). The projection of \( AB_2 \cap \mathbb{R}_+^n \) onto the positive orthant is a subset of \( Q_A \), and consequently we have \( Q_A = AB_2 \cap \mathbb{R}_+^n \).

The second conditions follow from the min-max representation

\[
\max_{u \in B_2} \min_{z : Az = (A u)_+} \|z\|_2 \leq 1 \iff \text{[Main Paper, (6)]},
\]

by noting that \( (I_n - A A^\dagger)(A u)_+ = 0 \) if and only if there exists \( z \) such that \( Az = (A u)_+ \), which in that case provided by \( A^\dagger (A u)_+ \). The third condition follows from the fact that the minimum norm solution to \( Az = (A u)_+ \) is given by \( A^\dagger (A u)_+ \) under the full row rank assumption on \( A \), which in turn implies \( I_n - AA^\dagger = 0 \).
Proof of Lemma 2.4 We have
\[
\max_{u: \|u\|_2 \leq 1} \|A^T(AA^T)^{-1}(Au)\|_2 \leq \sigma_{\text{max}}(A^T(AA^T)^{-1}) \max_{u: \|u\|_2 \leq 1} \|Au\|_2
\]
\[
= \sigma_{\text{min}}^{-1}(A) \max_{u: \|u\|_2 \leq 1} \|Au\|_2
\]
\[
\leq \sigma_{\text{min}}^{-1}(A) \max_{u: \|u\|_2 \leq 1} \|Au\|_2
\]
\[
\leq \sigma_{\text{min}}^{-1}(A) \sigma_{\text{max}}(A)
\]
\[
\leq 1.
\]
where the last inequality follows from the fact that A is whitened.

Proof of Lemma 2.5 Let us consider a data matrix A such that A = ca^T, where c ∈ R^n and a ∈ R^d. Then, (Au)^+ = c(a^T u)^+. If (a^T u)^+ = 0, then we can select z = 0 to satisfy the spike-free condition (ca^T u)^+ = Az. If (a^T u)^+ ≠ 0, then (Au)^+ = ca^T u = Au, where the spike-free condition can be trivially satisfied with the choice of z = u.

Proof of Lemma 2.6 The extreme point along the direction of v can be found as follows
\[
\max_{u,b} \sum_{i=1}^n v_i(a_i u + b) \text{ s.t. } |u| = 1, \quad (2)
\]
Since each neuron separates the samples into two sets, for some samples, ReLU will be active, i.e., \( S = \{i | a_i u + b \geq 0\} \), and for the others, it will be inactive, i.e., \( S^c = \{j | a_j u + b < 0\} = [n]/S \). Thus, we modify \( 2 \) as
\[
\max_{u,b} \sum_{i \in S} v_i(a_i u + b) \text{ s.t. } a_i u + b \geq 0, \forall i \in S, a_j u + b \leq 0, \forall j \in S^c, |u| = 1. \quad (3)
\]
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 \( 3 \), \(-a_i \leq b \leq -a_j, \forall i \in S, \forall j \in S^c \). Thus, the range for the possible bias values is \( \max_{i \in S}(-a_i), \min_{j \in S^c}(-a_j) \). Therefore, depending on the direction v, the optimal bias can be selected as follows
\[
b_v = \begin{cases} 
\max_{i \in S}(-a_i), & \text{if } \sum_{i \in S} v_i \leq 0 \\
\min_{j \in S^c}(-a_j), & \text{otherwise}
\end{cases} \quad (4)
\]
Similar arguments also hold for u = −1 and the min version of \( 2 \). Note that when \( \sum_{i \in S} v_i = 0 \), the value of the bias does not change the objective in \( 3 \). Thus, all the bias values in the range \( \max_{i \in S}(-a_i), \min_{j \in S^c}(-a_j) \) 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 e_i, we need to solve the following optimization problem
\[
\max_{u,b} (a_i^T u + b) \text{ s.t. } a_j^T u + b \leq 0, \forall i \neq j, \|u\|_2 = 1. \quad (5)
\]
Then the Lagrangian of \( 5 \) is
\[
L(\lambda, u, b) = a_i^T u + b - \sum_{j=1}^n \lambda_j(a_j^T u + b), \quad (6)
\]
where we do not include the unit norm constraint for \( u \). For (6), \( \lambda \) must satisfy \( \lambda \succ 0 \) and \( 1^T \lambda = 1 \). With these specifications, the problem can be written as

\[
\min_{\lambda} \max_{u} u^T \left( a_i - \sum_{j=1, j\neq i}^n \lambda_j a_j \right) \quad \text{s.t.} \quad \lambda \succ 0, \, 1^T \lambda = 1, \|u\|_2 = 1. \tag{7}
\]

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

\[
\min_{\lambda} \left\| a_i - \sum_{j=1, j\neq i}^n \lambda_j a_j \right\|_2 \quad \text{s.t.} \quad \lambda \succ 0, \, 1^T \lambda = 1. \tag{8}
\]

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

\[
u_i = \frac{a_i - \sum_{j=1, j\neq i}^n \lambda_j a_j}{\left\| a_i - \sum_{j=1, j\neq i}^n \lambda_j a_j \right\|_2} \quad \text{and} \quad b_i = \min(-a_j^T u_i),
\]

where the bias computation follows from the constraint in (7).

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

\[
\max_{u,b} \alpha^T (A u + b 1), \quad \text{s.t.} \quad \|u\|_2 = 1 \tag{9}
\]

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

\[
\max_{u,b} \sum_{i \in S} \alpha_i (a_i^T u + b) \quad \text{s.t.} \quad \langle a_i^T u + b \rangle \geq 0, \forall i \in S, \langle a_j^T u + b \rangle \leq 0, \forall j \in S^c, \|u\|_2 = 1,
\]

which has the following dual form

\[
\min_{\lambda, \nu} \max_{u,b} u^T \left( \sum_{i \in S} (\alpha_i + \lambda_i) a_i - \sum_{j \in S^c} \nu_j a_j \right) \quad \text{s.t.} \quad \lambda, \nu \succ 0, \sum_{i \in S} (\alpha_i + \lambda_i) = \sum_{j \in S^c} \nu_j, \|u\|_2 = 1.
\]

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

\[
u_i = \frac{\sum_{i \in S} (\alpha_i + \lambda_i) a_i - \sum_{j \in S^c} \nu_j a_j}{\left\| \sum_{i \in S} (\alpha_i + \lambda_i) a_i - \sum_{j \in S^c} \nu_j a_j \right\|_2} \quad \text{and} \quad b_i = \begin{cases} \max_{i \in S} (-a_i^T u), & \text{if } \sum_{i \in S} \alpha_i \leq 0 \\ \min_{j \in S^c} (-a_j^T u), & \text{otherwise} \end{cases}
\]

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

\[
P^* = \min_{\theta \in \Theta \setminus \{ w \}} \max_v v^T y + I(\| (AU)_+^T v \|_\infty \leq 1), \quad \text{s.t.} \quad \|u\|_2 \leq 1, \forall j,
\]

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

\[
\min \|\mu\|_{TV}
\text{s.t.} \int_{u \in B_2} (Au)_{+} d\mu(u) = y,
\]

where TV is the total variation norm of the Radon measure \(\mu\). 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

\[
K = \text{cone} \left\{ \left( \frac{s(Au)}{1} \right), u \in B_2, s \in \{-1, +1\}; \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}.
\]

Note that \(K\) is the union of finitely many convex closed sets, since the function \((Au)_{+}\) can be expressed as the union of finitely many convex closed sets. Therefore the set \(K\) 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 \(y = (AU^*)_{+}w^*\) for some vector \(w^*\). Since the primal and dual values are equal, we have \(v^*T y = v^*T (AU^*)_{+}w^* = \|w^*\|_1\), which shows that the primal-dual pair \((\{w^*, U^*\}, v^*)\) is optimal. Thus, the optimal neuron weights \(U^*\) satisfy \(\|\frac{1}{\lambda} \lambda (AU^*)_{+}v^*\|_\infty = 1\).

**Proof of Proposition 3.1** Here, we particularly examine the problem in [Main Paper, (11)] when we have a one dimensional dataset, i.e., \(\{a_i, y_i\}_{i=1}^n\). Then, [Main Paper, (11)] can be modified as

\[
\min_{\theta \in \Theta} \|w\|_1 \text{ s.t.} \ (au^T + 1b^T)_{+}w = y, |u_j| \leq 1, \forall j.
\]

Then, using Lemma 2.6 we can construct the following matrix

\[
A_e = (au^T + 1b^T)_{+},
\]

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

\[
\min_{w} \|w\|_1 \text{ s.t.} \ A_e w = y.
\]

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

\[
A_{e,s} w^* = y
A_{e,s}^T v^* + \text{sign}(w^*) = 0
\|A_{e,s}^T v^*\|_\infty \leq 1.
\]

Proof of Proposition 3.1
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., $a = [-2 -1 0 1 2]^T$ and $y = [1 -1 1 1 -1]^T$, and yields the following

$$A_e = (au^T + 1b^T) = \begin{bmatrix}
0 & 0 & 0 & 1 & 2 & 3 & 4 \\
1 & 0 & 0 & 0 & 1 & 2 & 3 \\
2 & 1 & 0 & 0 & 0 & 1 & 2 \\
3 & 2 & 1 & 0 & 0 & 0 & 1 \\
4 & 3 & 2 & 1 & 0 & 0 & 0 \\
\end{bmatrix},$$

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

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

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

$$\hat{A}_e = (au^T + 1\hat{b}^T) = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\
1 & 0 & 0 & 0 & 1 & 2 & 3 \\
2 & 1 & 0 & 0 & 0 & 1 & 2 \\
3 & 2 & 1 & 0 & 0 & 0 & 1 \\
4 & 3 & 2 & 1 & 0 & 0 & 0 \\
\end{bmatrix},$$

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

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

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

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

$$\min_w \beta \|w\|_1 + \frac{1}{2n} \|A_e \cdot w - y\|_2^2, \quad (13)$$

where the optimal solution $w^*$ satisfies

$$\frac{1}{n} A^T_{e,s} (A_e \cdot w^* - y) + \beta \text{sign}(w^*_s) = 0 \quad (14)$$

$$\|A^T_{e,s'} (A_e \cdot w^* - y)\|_\infty \leq \beta n,$$
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 $s'$ denotes the remaining entries (or columns). Now, let us consider a specific dataset, i.e., $a = [-2 - 1 0 1 2]^T$ and $y = [1 - 1 1 1 - 1]^T$. We then construct the following matrix

$$A_e = (au^T + 1b^T) = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 2 & 2.5 & 3 & 4 \\
1 & 0 & 0 & 0 & 1 & 1.5 & 2 & 3 \\
2 & 1 & 0 & 0 & 0 & 0.5 & 1 & 2 \\
3 & 2 & 1 & 0.5 & 0 & 0 & 0 & 0 & 1 \\
4 & 2 & 1.5 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},$$

where $u^T = [1 1 1 1 -1 -1 1 -1 -1]$ and $b^T = [2 1 0 -0.5 -1 -1 0 5 1 2]$. For this dataset with $\beta = 0.1$, the optimal value of (13) can be achieved by the following solutions

$$w_1 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 2 & 2.5 & 3 & 4 \\
0 & 1 & 0 & 0 & 1 & 1.5 & 2 & 3 \\
0 & 2 & 1 & 0 & 0 & 0.5 & 1 & 2 \\
0 & 3 & 2 & 1 & 0.5 & 0 & 0 & 0 & 0 & 1 \\
0 & 4 & 2 & 1.5 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},$$

$$w_2 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 2 & 2.5 & 3 & 4 \\
0 & 0 & 1 & 0 & 1 & 1.5 & 2 & 3 \\
0 & 0 & 2 & 1 & 0 & 0.5 & 1 & 2 \\
0 & 0 & 3 & 2 & 1 & 0.5 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 4 & 2 & 1.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},$$

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

\textbf{Proof of Lemma 3.1.} Since $y$ has both positive and negative entries, we need at least two $u$'s with positive and negative output weights to represent $y$ using the output range of ReLU. Therefore the optimal value of the $\ell_0$ problem is at least 2. Note that $AA^\dagger = I_n$ since $A$ is full row rank. Then let us define the output weights

$$w_1 = \|A^\dagger(y)_+\|_2$$

$$w_2 = -\|A^\dagger(-y)_+\|_2.$$

Then note that

$$w_1(Au_1)_+ + w_2(Au_2)_+ = (AA^\dagger(y)_+) - (AA^\dagger(-y)_+) = (y)_+ - (-y)_+ = y$$

where the second equality follows from $AA^\dagger = I_n$. 

\hfill $\Box$
Figure 2: Solutions provided by $w_1$ and $w_2$ 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 $U$ be a weight matrix for [Main Paper, (4)]. Then, $U \in \mathbb{R}^{d \times m}$ is an optimal solution for the regularized training problem if

$$\exists \alpha \in \mathbb{R}^n, w \in \mathbb{R}^m \text{ s.t. } (AU)_+ w = y, \quad (AU)_+ \alpha = \text{sign}(w)$$

and

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

(15)

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 $A = U$ where $U^T U = UU^T = I_n$ since $A$ is assumed full row rank. Consider the dual optimization problem

$$\max_{|v^T (Au)_+| \leq 1, \ \forall u \in B_2} v^T y$$

(17)

Changing the variable to $u' = U u$ in the dual problem we next show that

$$\max_{|v^T (u')_+| \leq 1, \ \forall u' \in B_2} v^T y = \max_{\|v\|_2 \leq 1, \ \|(-v)_+\|_2 \leq 1} v^T y.$$  

(18)

where the equality follows from the upper bound

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

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

$$-v^T (u')_+ \leq (-v)_+^T (u')_+ \leq \|(-v)_+\|_2 (u')_+ \|_2 \leq \|(-v)_+\|_2.$$

11
which is achieved when \( \mathbf{u}' = \frac{(-\mathbf{y})_+}{\|(-\mathbf{y})_+\|_2} \), 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

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

since we have

\[
\mathbf{v}^T \mathbf{y} = \frac{\mathbf{y}^T (\mathbf{y})_+}{\|\mathbf{y}\|_2 + \|(-\mathbf{y})_+\|_2} - \frac{\mathbf{y}^T (-\mathbf{y})_+}{\|\mathbf{y}\|_2 + \|(-\mathbf{y})_+\|_2} = \frac{(\mathbf{y})_+^T (\mathbf{y})_+}{\|\mathbf{y}\|_2 + \|(-\mathbf{y})_+\|_2} + \frac{(-\mathbf{y})_+^T (-\mathbf{y})_+}{\|\mathbf{y}\|_2 + \|(-\mathbf{y})_+\|_2}
\]

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

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

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

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

or equivalently

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

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

**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-Cerdà, 1998]. \( \square \)

**Proof of Theorem 3.2** Given a vector \( \mathbf{u} \) we partition \( \mathbf{A} \) according to the subset \( S = \{i|a_{i}\mathbf{u} \geq 0\} \), where \( \mathbf{A}_S \mathbf{u} \geq 0 \) and \( -\mathbf{A}_S \mathbf{u} \geq 0 \) 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 \( (\mathbf{A} \mathbf{u})_+ \) as follows

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

Then we use the block matrix pseudo-inversion formula [Baksalary and Baksalary, 2007]

\[
\mathbf{A}^\dagger = \begin{bmatrix} (\mathbf{A}_S \mathbf{P}^\perp_S)^\dagger & (\mathbf{A}_S \mathbf{P}^\perp_S \mathbf{A}_S^T)^\dagger \\
\end{bmatrix},
\]

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

\[
\mathbf{P}_S = \mathbf{I} - \mathbf{A}_S^T (\mathbf{A}_S \mathbf{A}_S^T)^{-1} \mathbf{A}_S
\]

\[
\mathbf{P}_{S^c} = \mathbf{I} - \mathbf{A}_{S^c}^T (\mathbf{A}_{S^c} \mathbf{A}_{S^c}^T)^{-1} \mathbf{A}_{S^c}.
\]
Note that the matrices \( A_S A_S^T \in \mathbb{R}^{\lvert S \rvert \times \lvert S \rvert} \), \( A_S A_S^T \in \mathbb{R}^{\lvert S' \rvert \times \lvert S' \rvert} \) are full column rank with probability one since the matrix \( A \in \mathbb{R}^{n \times d} \) is i.i.d. Gaussian where \( n < d \). Hence the inverses \( (A_S A_S^T)^{-1} \) and \( (A_{S'} A_{S'}^T)^{-1} \) exist with probability one. Plugging in the above representation in the spike-free condition we get

\[
A^\dagger (A u)_+ = (A_S P_{S'}^{-1})^\dagger A_S u.
\]

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

\[
P \left[ \max_{u \notin B_2} \| A^\dagger (A u)_+ \|_2 > 1 \right] = P \left[ \exists u \in B_2 \mid \| A^\dagger (A u)_+ \|_2 > 1 \right]
\]

\[
\leq P \left[ \exists u \in B_2, S \subseteq [n] \mid \| (A_S P_{S'}^{-1})^\dagger A_S u \|_2 > 1 \right].
\]

Finally, observe that \( P_{S'} \in \mathbb{R}^{d \times d} \) is a uniformly random projection matrix of subspace of dimension \( |S| \leq n \). Therefore as \( d \to \infty \), we have \( P_{S'} 
abla \to I_d \), and consequently

\[
\lim_{d \to \infty} \| (A_S P_{S'}^{-1})^\dagger A_S u \|_2 = \| A_S^\dagger A_S u \|_2,
\]

with probability one, and we have

\[
\lim_{d \to \infty} P \left[ \exists u \in B_2, S \subseteq [n] \mid \| (A_S P_{S'}^{-1})^\dagger A_S u \|_2 > 1 \right] = 0.
\]

**Proof of Theorem 3.3** Since each sample \( a_i \) is a vertex of \( C_a \), we can find a separating hyperplane defined by the parameters \((u_j, b_j)\) so that \( a_i^T u_j + b_j > 0 \) and \( a_i^T u_j + b_j \leq 0, \forall i \neq j \). Then, choosing \( \{(u_j, b_j)\}_{j=1}^n \) yields that \( (A U + 1 b^T)_+ \) is a diagonal matrix. Using these hidden neurons, we write the constraint of \([\text{Main Paper, (4)}]\) in a more compact form as

\[
(A U + 1 b^T)_+ w = y,
\]

which is a least squares problem with a full rank square data matrix. Therefore, selecting \( w = ((A U + 1 b^T)_+)^\dagger y \) along with \( U \) and \( b \) 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 \( d_i \)

\[
d_i \triangleq \min_{z \in \mathbb{R}^n : \sum_{j \neq i} z_j = 1, z_{j_i} = 1, z_{j_i} = -1} \| a_i - \sum_{j \neq i} a_j z_j \|_2 = \min_{z \in \mathbb{R}^n : \sum_{j \neq i} z_j = 1} \| \sum_{j \neq i} z_j a_j \|_2 = \min_{z \in \mathbb{R}^n : \sum_{j \neq i} z_j = 1} \max_{v \in \mathbb{R}^d : \| v \|_2 \leq 1} v^T \sum_{j \neq i} z_j a_j.
\]

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

\[
E d_i \geq E \min_{z \in \mathbb{R}^n : \sum_{j \neq i} z_j = 1, z_{j_i} = 1} \max_{v \in \mathbb{R}^d : \| v \|_2 \leq 1} h^T v \| z \|_2 + z^T g
\]

\[
= E \min_{z \in \mathbb{R}^n : \sum_{j \neq i} z_j = 1, z_{j_i} = 1} \| h \|_2 \| z \|_2 + z^T g
\]

\[
\geq \sqrt{d} \| z \|_2 - E \max_{j \in [n], j \neq i} g_j + g_i
\]

\[
\geq \frac{\sqrt{d}}{\sqrt{n}} - \sqrt{2 \log(n - 1)}, \quad (19)
\]
where \( h \in \mathbb{R}^d \) and \( g \in \mathbb{R}^n \) are random vectors with i.i.d. standard Gaussian components, and the second inequality follows from a well-known result on finite Gaussian suprema \( \text{Ledoux and Talagrand, 2013} \). 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 \( A \). This can be seen via the following argument

\[
\min_{z \in \mathbb{R}^n} \{ \sum_{j \neq i} z_j = 1, z_j \geq 0, z_i = -1 \} \| A^T z \|_2 \\
\leq \| (A - \hat{A}) \|_F \max_{z \in \mathbb{R}^n} \{ \sum_{j \neq i} z_j = 1, z_j \geq 0, z_i = -1 \| z \|_1 \\
\leq 2 \| (A - \hat{A}) \|_F 
\]

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

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

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 \( \ell_1 \) solutions to linear systems. Recognizing the constraint \( v \in Q_A \) can be stated as

\[
v \in Q^w_A, v \in -Q^w_A,
\]

which is equivalent to

\[
v \in Q^w_A \cap -Q^w_A.
\]

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

\[
(Q^w_A \cap -Q^w_A)^0 = \text{conv } Q_A \cup -Q_A.
\]

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

\[
D^* = \inf_{t \in \mathbb{R}} t \text{ s.t. } y \in t \text{ conv } \{ Q_A \cup -Q_A \} \,
\]

where \( \text{conv} \) represents the convex hull of a set.
Let us restate dual of the two-layer ReLU neural network training problem given by
\[
\max_v v^T y \quad \text{s.t.} \quad v \in Q_A^o, \quad -v \in Q_A^o
\]
where \(Q_A^o\) is the polar dual of \(Q_A\) defined as \(Q_A^o = \{v | v^T u \leq 1 \text{ for all } u \in Q_A\}\).

**Remark 1.** The dual problem given in (21) is analogous to the convex duality in minimum \(\ell_1\) norm solutions to linear systems. In particular, for the latter it holds that
\[
\min_{w : Aw = y} \|w\|_1 = \max_{v \in \text{conv}\{\hat{a}_1, \ldots, \hat{a}_d\}} v^T y,
\]
where \(\hat{a}_1, \ldots, \hat{a}_d\) are the columns of \(A\). The above optimization problem can also be put in the gauge optimization form as follows.
\[
\min_{w : Aw = y} \|w\|_1 = \inf_{t \in \mathbb{R}} t \quad \text{s.t.} \quad y \in t \text{conv}\{\pm \hat{a}_1, \ldots, \hat{a}_d\},
\]
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
\[
\min_{\theta \in \Theta} \ell((AU)_+ w, y) + \beta \|w\|_1 \quad \text{s.t.} \quad \|u_j\|_2 \leq 1, \forall j,
\]
where \(\ell(\cdot, y)\) is a convex loss function.

**Theorem 1.** The dual of (22) is given by
\[
\max_v -\ell^*(v) \quad \text{s.t.} \quad v \in \beta Q_A^o, \quad -v \in \beta Q_A^o,
\]
where \(\ell^*\) is the Fenchel conjugate function defined as
\[
\ell^*(v) = \max_z z^T v - \ell(z, y).
\]

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 \(Y \in \mathbb{R}^{n \times o}\) and \(f(A) = (AU)_+ W\), where \(W \in \mathbb{R}^{m \times o}\). Then, we formulate the following dual problem
\[
\max_{V} \text{tr}(V^T Y) \quad \text{s.t.} \quad \|V^T (Au)\|_\infty \leq 1, \forall u \in B_2
\]
and an optimal \(U\) satisfies
\[
\|(AU^*)^T V^*\|_\infty = 1,
\]
where \(V^*\) is the optimal dual variable and \(\text{tr}\) represents the trace of a matrix. Note that we can also consider block \(\ell_1-\ell_2\) 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 $k \in [o]$

\[
\hat{u}_{k,1} = \arg\max_{u} v_k^T A u \text{ s.t. } Au \succ 0, \|u\|_2 \leq 1
\]

\[
\hat{u}_{k,2} = \arg\min_{u} v_k^T A u \text{ s.t. } Au \succ 0, \|u\|_2 \leq 1,
\]

where $v_k$ is the $k^{th}$ column of $V$. 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

\[
\min_{\theta \in \Theta} \|W\|_0 \text{ s.t. } f_\theta(A) = Y, \|u_j\|_2 = 1, \forall j.
\]

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

**Lemma 1.** Suppose that $n \leq d$, $A$ is full row rank, and $Y \in \mathbb{R}^{n \times o}$, 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

\[
u_k = \frac{A^\dagger(y_k)_+}{\|A^\dagger(y_k)_+\|_2} \text{ and } w_k = \|A^\dagger(y_k)_+\|_2 e_k
\]

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

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

**Lemma 2.** We have $\ell_1-\ell_0$ equivalence if the following condition holds

\[
\min_{v : v^T (Au)_+ = 1, \forall k} \max_{u : u \in B^2_2} v^T (Au)_+ \leq 1.
\]

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

### References


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


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