OPT2020: 12th Annual Workshop on Optimization for Machine Learning

Convex Programs for Global Optimization of Convolutional Neural Networks in Polynomial-Time

Tolga Ergen ERGEN@STANFORD.EDU
Mert Pilanci PILANCI@STANFORD.EDU
Stanford University, Stanford, CA 94305

Abstract

We study training of Convolutional Neural Networks (CNNs) with ReLU activations and introduce exact convex optimization formulations with a polynomial complexity with respect to the number of data samples, the number of neurons and data dimension. Particularly, we develop a convex analytic framework utilizing semi-infinite duality to obtain equivalent convex optimization problems for two-layer CNNs, where convex problems are regularized by the sum of ℓ_2 norms of variables.

1. Introduction

Convolutional Neural Networks (CNNs) have shown a remarkable success across various machine learning problems [15]. However, our theoretical understanding of CNNs still remains restricted, where the main challenge arises from the highly non-convex and nonlinear structure of CNNs with nonlinear activations such as Rectified Linear Units (ReLU). To this end, we study the training problem for various CNN architectures with ReLU activations and introduce equivalent finite dimensional convex formulations that can be used to *globally* optimize these architectures. Our results characterize the role of network architecture in terms of *equivalent convex regularizers*. Remarkably, we prove that the proposed methods are *polynomial time* with respect to all problem parameters.

Convex neural network training was previously considered in [2, 4]. However, these studies are restricted to two-layer fully connected networks with infinite width, thus, the optimization problem involves infinite dimensional variables. Moreover, it has been shown that even adding a single neuron to a neural network leads to a non-convex optimization problem which cannot be solved efficiently [2]. Another line of research [5, 10, 14, 16, 20, 21, 24, 32] focuses on the effect of implicit and explicit regularization in neural network training and aims to explain why the resulting network generalizes well. Among these studies, [10, 20, 24] proved that the minimum ℓ_2 norm two-layer network that perfectly fits a one dimensional dataset outputs the linear spline interpolation. Moreover, [14] studied certain linear convolutional networks and showed an implicit non-convex quasi-norm regularization. However, as the number of layers increases, the regularization approaches to ℓ_0 quasi-norm, which is not computationally tractable. Recently, [21] showed that two-layer CNNs with linear activations can be equivalently optimized as nuclear and ℓ_1 norm regularized convex problems. Although all the norm characterizations provided by these studies are insightful for future research, existing results are quite restricted due to linear activations, simple settings or intractable optimization problems.

Shallow CNNs and their representational power: As opposed to their relatively simple and shallow architecture, CNNs with two layers are very powerful and efficient inference models. In [3], the authors show that greedy training of two layer CNNs can achieve comparable performance to deeper models such as VGG-11[26]. However, a full theoretical understanding and interpretable description of CNNs even with a single hidden layer is lacking in the literature.

Our contributions: Unlike previous studies, we introduce exact and finite dimensional convex programs to globally optimize various CNN architectures through our convex analytic framework utilizing semi-infinite duality. Our contributions can be summarized as follows. We develop convex programs that are *polynomial time* with respect to all input parameters: the number of samples, data dimension, and the number of neurons to globally train CNNs. To the best of our knowledge, this is the first work characterizing polynomial time trainability of non-convex CNN models. More importantly, we achieve this complexity with explicit and interpretable convex optimization problems. Consequently, training CNNs, especially in practice, can be further accelerated by leveraging extensive tools available from convex optimization theory.

Notation and preliminaries: We denote the matrices and vectors as uppercase and lowercase bold letters, respectively, for which a subscript indicates a certain element or column. We use $\mathbf{I}_k$ to denote the identity matrix of size k. We denote the set of integers from 1 to n as $[n]$. The Frobenius and nuclear norms are denoted as $\|\cdot\|_F$ and $\|\cdot\|_*$. We use $\mathcal{B}_p$ to denote the unit ℓ_p ball, i.e., $\mathcal{B}_p := \{\mathbf{u} \in \mathbb{C}^d : \|\mathbf{u}\|_p \leq 1\}$. We also use $1[x]$ as a function that outputs 1 if x is true, 0 otherwise.

In order to keep the presentation and notation simple, we will use a regression framework with scalar outputs and squared loss. In our regression framework, we denote the input data matrix and the corresponding label vector as $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{y} \in \mathbb{R}^n$, respectively. Moreover, we represent the patch matrices, i.e., subsets of columns, extracted from $\mathbf{X}$ as $\mathbf{X}_k \in \mathbb{R}^{n \times h}$, $k \in [K]$, where h denotes the filter size. With this notation, $\{\mathbf{X}_k \mathbf{u}\}_{k=1}^K$ describes a convolution operation between the filter $\mathbf{u} \in \mathbb{R}^h$ and the data matrix $\mathbf{X}$. Throughout the paper, we will use the ReLU activation function defined as $(x)_+ = \max\{0, x\}$. However, since CNN training problems with ReLUs are not convex in their conventional form, below we introduce an alternative formulation for this activation, which will be crucial for our derivations.

1.1. Hyperplane arrangements

Let $\mathcal{H}$ be the set of all hyperplane arrangement patterns of $\mathbf{X}$, defined as the following set

$$\mathcal{H} := \bigcup \big\{ \{\text{sign}(\mathbf{X}\mathbf{w})\} \ : \ \mathbf{w} \in \mathbb{R}^d \big\},$$

which has finitely many elements, i.e., $|\mathcal{H}| \leq N_H < \infty$, $N_H \in \mathbb{N}$. We now define a collection of sets that correspond to positive signs for each element in $\mathcal{H}$, by $\mathcal{S} := \big\{ \{\cup_{h_i=1}\{i\}\} \ : \ \mathbf{h} \in \mathcal{H} \big\}$. We extend ReLU to an elementwise function that masks the negative entries of a vector or matrix. Hence, given a set $S \in \mathcal{S}$, e define a diagonal mask matrix $\mathbf{D}(S) \in \mathbb{R}^{n \times n}$ defined as $\mathbf{D}(S)_{ii} := \mathbb{1}[i \in S]$. Then, we can equivalently represent $(\mathbf{X}\mathbf{w})_+$ as $\mathbf{D}(S)\mathbf{X}\mathbf{w}$ provided that $\mathbf{D}(S)\mathbf{X}\mathbf{w} \geq 0$ and $(\mathbf{I}_n - \mathbf{D}(S))\, \mathbf{X}\mathbf{w} \leq 0$. Note that these constraints can be compactly defined as $(2\mathbf{D}(S) - \mathbf{I}_n)\, \mathbf{X}\mathbf{w} \geq 0$. If we denote the cardinality of $\mathcal{S}$ as P, i.e., the number of regions in a partition of $\mathbb{R}^d$ by hyperplanes passing through the origin and are perpendicular to the rows of $\mathbf{X}$, then for $r \leq n$, where $r := \text{rank}(\mathbf{X}) \leq d$, we have [8, 19, 28, 30]

$$P \leq 2 \sum_{k=0}^{r-1} \binom{n-1}{k} \leq 2r \left(\frac{e(n-1)}{r} \right)^r .$$

1.2. Convolutional hyperplane arrangements

We now define a notion of hyperplane arrangements for CNNs, where we introduce the patch matrices $\{\mathbf{X}_k\}_{k=1}^K$ instead of directly operating on $\mathbf{X}$. We first construct a new data matrix as $\mathbf{M} = [\mathbf{X}_1; \, \mathbf{X}_2; \, \ldots \, \mathbf{X}_K] \in \mathbb{R}^{nK \times h}$. We then define *convolutional hyperplane arrangements* as the hyperplane arrangements for $\mathbf{M}$ and denote the cardinality of this set as P_{conv}. Then, we have

$$P_{conv} \leq 2 \sum_{k=0}^{r_c-1} \binom{nK-1}{k} \leq 2r_c \left(\frac{e(nK-1)}{r_c} \right)^{r_c}$$

where $r_c := \text{rank}(\mathbf{M}) \leq h$ and $K = \left\lfloor \frac{d-h}{\text{stride}} \right\rfloor + 1$. Note that when the filter size h is fixed, P_{conv} is polynomial in n and d.

Remark 1 *There exist P hyperplane arrangements of $\mathbf{X}$ where P is exponential in r. Thus, if $\mathbf{X}$ is full rank, $r = d$, then P can be exponentially large in the dimension d. As it will be shown, this makes the training problem for fully connected networks challenging. On the other hand, for CNNs, the number of relevant hyperplane arrangements P_{conv} is exponential in r_c. If $\mathbf{M}$ is full rank, then $r_c = h \ll d$ and accordingly $P_{conv} \ll P$. This shows that the parameter sharing structure in CNNs enables a significant reduction in the number of possible hyperplane arrangements. As shown in the sequel, our results imply that the complexity of training problem is significantly lower compared to fully connected networks.*

2. Two-layer Convolutional Neural Networks

In this section[1], we present exact convex formulation for two-layer CNN architectures with average pooling. Particularly, we consider an architecture with m filters, average pooling and standard weight decay regularization, which can be trained via the following problem

$$p_1^* = \min_{\{\mathbf{u}_j, w_j\}_{j=1}^m} \frac{1}{2} \left\| \sum_{j=1}^m \sum_{k=1}^K (\mathbf{X}_k \mathbf{u}_j)_+ w_j - \mathbf{y} \right\|_2^2 + \frac{\beta}{2} \sum_{j=1}^m \left(\|\mathbf{u}_j\|_2^2 + w_j^2 \right), \tag{1}$$

where $\mathbf{u}_j \in \mathbb{R}^h$ and $\mathbf{w} \in \mathbb{R}^m$ are the filter and output weights, respectively, and $\beta > 0$ is a regularization parameter. After a rescaling (see Appendix A.1), we obtain the following problem

$$p_1^* = \min_{\substack{\{\mathbf{u}_j, w_j\}_{i=1}^m \\ \mathbf{u}_j \in \mathcal{B}_2, \forall j}} \frac{1}{2} \left\| \sum_{j=1}^m \sum_{k=1}^K (\mathbf{X}_k \mathbf{u}_j)_+ w_j - \mathbf{y} \right\|_2^2 + \beta \|\mathbf{w}\|_1. \tag{2}$$

[1] All the proofs and details are presented in Appendix.

Then, taking dual with respect to $\mathbf{w}$ and changing the order of min-max yields the weak dual

$$p_1^* \geq d_1^* = \max_{\mathbf{v}} -\frac{1}{2}\|\mathbf{v}-\mathbf{y}\|_2^2 + \frac{1}{2}\|\mathbf{y}\|_2^2 \text{ s.t. } \max_{\mathbf{u}\in\mathcal{B}_2} \left|\sum_{k=1}^{K} \mathbf{v}^T \left(\mathbf{X}_k\mathbf{u}\right)_+\right| \leq \beta. \tag{3}$$

which is a semi-infinite optimization problem with n variables. The dual of (3) can be obtained as a finite dimensional convex program using semi-infinite optimization theory [12]. The same dual also corresponds to the bidual of (1). Suprisingly, strong duality holds as soon as the number of neurons exceed a critical value. In the sequel, we use this result to derive an exact convex formulation for (1).

Theorem 1 *Let m be a number such that $m \geq m^*$ for some $m^* \in \mathbb{N}, m^* \leq n+1$, then strong duality holds for (3), i.e., $p_1^* = d_1^*$, and the equivalent convex program for (1) is*

$$\min_{\substack{\{\mathbf{w}_i,\mathbf{w}_i'\}_{i=1}^{P_{conv}}\\ \mathbf{w}_i,\mathbf{w}_i'\in\mathbb{R}^h,\forall i}} \frac{1}{2}\left\|\sum_{i=1}^{P_{conv}}\sum_{k=1}^{K}\mathbf{D}(S_i^k)\mathbf{X}_k\left(\mathbf{w}_i'-\mathbf{w}_i\right)-\mathbf{y}\right\|_2^2 + \beta\sum_{i=1}^{P_{conv}}\left(\|\mathbf{w}_i\|_2+\|\mathbf{w}_i'\|_2\right)$$

$$\text{s.t. } (2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\mathbf{w}_i \geq 0,\ (2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\mathbf{w}_i' \geq 0, \forall i,k. \tag{4}$$

Moreover, an optimal solution to (1) with m^* filters can be constructed from (4) as follows

$$(\mathbf{u}_{j_{1i}}^*, w_{j_{1i}}^*) = \left(\frac{\mathbf{w}_i'^*}{\sqrt{\|\mathbf{w}_i'^*\|_2}}, \sqrt{\|\mathbf{w}_i'^*\|_2}\right) \quad \text{if} \quad \|\mathbf{w}_i'^*\|_2 > 0$$

$$(\mathbf{u}_{j_{2i}}^*, w_{j_{2i}}^*) = \left(\frac{\mathbf{w}_i^*}{\sqrt{\|\mathbf{w}_i^*\|_2}}, -\sqrt{\|\mathbf{w}_i^*\|_2}\right) \quad \text{if} \quad \|\mathbf{w}_i^*\|_2 > 0\,,$$

where $\{\mathbf{w}_i'^*, \mathbf{w}_i^*\}_{i=1}^{P_{conv}}$ are the optimal solutions to (4). Here, we have $m^* := \sum_{i=1}^{P_{conv}} 1[\|\mathbf{w}_i^*\|_2 \neq 0] + \sum_{i=1}^{P_{conv}} 1[\|\mathbf{w}_i'^*\|_2 \neq 0]$.

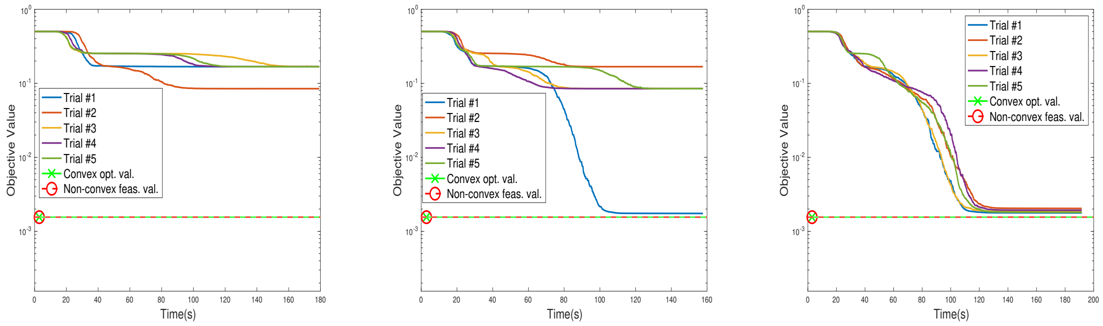
Therefore, we obtain a finite dimensional convex formulation with $2hP_{conv}$ variables and $2nP_{conv}K$ constraints for the non-convex problem in (1). Since P_{conv} is polynomial in n and d given a fixed $r_c \leq h$, (4) can be solved by a standard convex optimization solver in polynomial time.

Remark 2 *Our analysis shows that for fixed rank r_c, or fixed filter size h, the complexity is polynomial in all problem parameters: n (number of samples), m (number of filters, i.e., neurons), and d (dimension). The filter size h is typically a small constant, e.g., $h = 9$ for 3×3 filters. We also note that for fixed n and rank$(\mathbf{X}) = d$, the complexity of fully connected networks is exponential in d, which cannot be improved unless $P = NP$ even for $m = 2$ [6, 21]. However, this result shows that CNNs can be trained to global optimality with polynomial complexity as a convex program.*

Interpreting non-convex CNNs as convex variable selection models: Interestingly, we have the sum of the squared ℓ_2 norms of the weights in the non-convex problem (1) as the regularizer, however, the equivalent convex program in (4) is regularized by the sum of the ℓ_2 norms of the weights. This particular regularizer is known as group ℓ_1 norm, and is well-studied in the context of sparse recovery and variable selection [17, 31]. Hence, our convex program reveals an implicit variable selection mechanism in the original non-convex problem in (4). More specifically, the original features in $\mathbf{X}$ are mapped to higher dimensions via convolutional hyperplane arrangements as $\{\mathbf{D}(S_i^k)\mathbf{X}_k\}_{i=1}^{P_{conv}}$ and followed by a convex variable selection strategy using the group ℓ_1 norm.

3. Numerical experiments

In this section[2], we present numerical experiments to verify our results in the previous sections. We first perform an experiment with a synthetic dataset, where $\mathbf{X} \in \mathbb{R}^{6\times 15}$ is generated using a multivariate normal distribution with zero mean and identity covariance, and $\mathbf{y} = [1\ -1\,1\,1\,1\ -1]^T$. In this case, we use the two-layer CNN model in (1) and the corresponding convex program in (4). In Figure 1, we plot the regularized objective value with respect to the computation time. Here, we plot 5 different independent realizations for SGD. We also plot both the non-convex objective in (1) and the convex objective in (4) for our convex program, where we use the prescribed rescaling in Theorem 1 to achieve the optimal objective value. We perform the experiment using $m = 3, 8, 15$ filters of size $h = 10$ and stride 5, where we observe that as the number of filters increases, the ratio of the trials converging to the optimal objective value increases as well.



$m = 3$ $m = 8$ $m = 15$

Figure 1: Training cost of a two-layer CNN (with average pooling and $m = 3, 8, 15$) trained with SGD (5 initialization trials) on a synthetic dataset ($n = 6$, $d = 15$, $h = 10$, stride $= 5$), where the green line with a marker represents the objective value obtained by the proposed convex program in (4) and the red line with a marker represents the non-convex objective value in (1) of a feasible network with the weights found by the convex program. Here, we use markers to denote the total computation time of the convex optimization solver.

4. Concluding remarks

We studied non-convex CNN training problems and introduced exact finite dimensional convex programs. Particularly, we provide equivalent convex characterizations for ReLU CNN architectures in a higher dimensional space. Unlike the previous studies, we prove that these equivalent characterizations have polynomial complexity in all input parameters and can be globally optimized via convex optimization solvers. In the light of our results, efficient optimization algorithms can be developed to exactly (or approximately) optimize deep CNN architectures for large scale experiments in practice, which is left for future research.

[2] We use CVX [13] and CVXPY [1, 9] with the SDPT3 solver [29] to solve convex optimization problems.

References

[1] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

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

[3] Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to imagenet. In *International Conference on Machine Learning*, pages 583–593, 2019.

[4] 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.

[5] Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. *CoRR*, abs/1904.09080, 2019. URL http://arxiv.org/abs/1904.09080.

[6] Digvijay Boob, Santanu S Dey, and Guanghui Lan. Complexity of training relu neural network. *arXiv preprint arXiv:1809.10787*, 2018.

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

[8] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 1965.

[9] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[10] Tolga Ergen and Mert Pilanci. Convex geometry and duality of over-parameterized neural networks. *arXiv preprint arXiv:2002.11219*, 2020.

[11] Tolga Ergen and Mert Pilanci. Convex geometry of two-layer relu networks: Implicit autoencoding and interpretable models. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4024–4033, Online, 26–28 Aug 2020. PMLR. URL http://proceedings.mlr.press/v108/ergen20a.html.

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

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

[14] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors,

Advances in Neural Information Processing Systems 31, pages 9461–9471. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8156-implicit-bias-of-gradient-descent-on-linear-convolutional-networks.pdf.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[16] Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient descent quantizes relu network features. *arXiv preprint arXiv:1803.08367*, 2018.

[17] L. Meier, S. van de Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society, Series B*, 70:53–71, 2008.

[18] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[19] Piyush C Ojha. Enumeration of linear threshold functions from the lattice of hyperplane intersections. *IEEE Transactions on Neural Networks*, 11(4):839–850, 2000.

[20] Rahul Parhi and Robert D. Nowak. Minimum "norm" neural networks are splines, 2019.

[21] Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7695–7705, Virtual, 13–18 Jul 2020. PMLR. URL http://proceedings.mlr.press/v119/pilanci20a.html.

[22] Saharon Rosset, Grzegorz Swirszcz, Nathan Srebro, and Ji Zhu. L1 regularization in infinite dimensional feature spaces. In *International Conference on Computational Learning Theory*, pages 544–558. Springer, 2007.

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

[24] Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? *CoRR*, abs/1902.05040, 2019. URL http://arxiv.org/abs/1902.05040.

[25] Alexander Shapiro. Semi-infinite programming, duality, discretization and optimality conditions. *Optimization*, 58(2):133–161, 2009.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] Maurice Sion. On general minimax theorems. *Pacific J. Math.*, 8(1):171–176, 1958. URL https://projecteuclid.org:443/euclid.pjm/1103040253.

[28] Richard P Stanley et al. An introduction to hyperplane arrangements. *Geometric combinatorics*, 13:389–496, 2004.

[29] RH Tütüncü, KC Toh, and MJ Todd. Sdpt3—a matlab software package for semidefinite-quadratic-linear programming, version 3.0. *Web page http://www. math. nus. edu. sg/mattohkc/sdpt3. html*, 2001.

[30] RO Winder. Partitions of n-space by hyperplanes. *SIAM Journal on Applied Mathematics*, 14 (4):811–818, 1966.

[31] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

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

Appendix A.

In this section, we present additional materials and proofs of the main results that are not included in the main paper due to the page limit.

A.1. Equivalence of the ℓ_1 penalized objectives

In this section, we prove the equivalence between the original problems with ℓ_2 regularization and their ℓ_1 penalized versions. We also note that similar equivalence results were also presented in [10, 11, 18, 24]. We start with the equivalence between (1) and (2).

Lemma A.1 *The following two problems are equivalent:*

$$\min_{\{\mathbf{u}_j, w_j\}_{j=1}^m} \frac{1}{2}\left\| \sum_{j=1}^m \sum_{k=1}^K (\mathbf{X}_k \mathbf{u}_j)_+ w_j - \mathbf{y} \right\|_2^2 + \frac{\beta}{2} \sum_{j=1}^m \left(\|\mathbf{u}_j\|_2^2 + w_j^2 \right)$$

$$= \min_{\substack{\{\mathbf{u}_j, w_j\}_{j=1}^m \\ \mathbf{u}_j \in \mathcal{B}_2, \forall j}} \frac{1}{2}\left\| \sum_{j=1}^m \sum_{k=1}^K (\mathbf{X}_k \mathbf{u}_j)_+ w_j - \mathbf{y} \right\|_2^2 + \beta \sum_{j=1}^m \|\mathbf{w}\|_1.$$

Proof of Lemma A.1 We rescale the parameters as $\bar{\mathbf{u}}_j = \gamma_j \mathbf{u}_j$ and $\bar{w}_j = w_j/\gamma_j$, for any $\gamma_j > 0$. Then, the output becomes

$$\sum_{j=1}^m \sum_{k=1}^K (\mathbf{X}_k \bar{\mathbf{u}}_j)_+ \bar{w}_j = \sum_{j=1}^m \sum_{k=1}^K (\mathbf{X}_k \mathbf{u}_j \gamma_j)_+ \frac{w_j}{\gamma_j} = \sum_{j=1}^m \sum_{k=1}^K (\mathbf{X}\mathbf{u}_j)_+ w_j,$$

which proves that the scaling does not change the network output. In addition to this, we have the following basic inequality

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

where the equality is achieved with the scaling choice $\gamma_j = \left(\frac{|w_j|}{\|\mathbf{u}_j\|_2}\right)^{\frac{1}{2}}$ is used. 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 $\|\mathbf{w}\|_1$.

Now, let us consider a modified version of the problem, where the unit norm equality constraint is relaxed as $\|\mathbf{u}_j\|_2 \leq 1$. Let us also assume that for a certain index j, we obtain $\|\mathbf{u}_j\|_2 < 1$ with $w_j \neq 0$ as an optimal solution. 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 $w_j = 0$. Therefore, we have a contradiction, which proves that all the constraints that correspond to a nonzero w_j must be active for an optimal solution. This also shows that replacing $\|\mathbf{u}_j\|_2 = 1$ with $\|\mathbf{u}_j\|_2 \leq 1$ does not change the solution to the problem. ■

A.2. Proof of the main result (Theorem 1)

We start with the following claim.

Proposition A.1 *Given $m \geq m^*$, strong duality holds for (3), i.e., $p_1^* = d_1^*$.*[3]

We now focus on the single-sided dual constraint

$$\max_{\mathbf{u} \in \mathcal{B}_2} \sum_{k=1}^{K} \mathbf{v}^T \left(\mathbf{X}_k \mathbf{u}\right)_+ \leq \beta, \tag{5}$$

which can be written as

$$\max_{\substack{S^k \subseteq [n] \\ S^k \in \mathcal{S}}} \max_{\mathbf{u} \in \mathcal{B}_2} \sum_{k=1}^{K} \mathbf{v}^T \mathbf{D}(S^k) \mathbf{X}_k \mathbf{u} \text{ s.t. } (2\mathbf{D}(S^k) - \mathbf{I}_n)\mathbf{X}_k \mathbf{u} \geq 0, \, \forall k. \tag{6}$$

Since the inner maximization is convex and there exists a strictly feasible solution for fixed $\mathbf{D}(S^k)$ matrices, (6) can also be written as

$$\begin{aligned}
&\max_{\substack{S^k \subseteq [n] \\ S^k \in \mathcal{S}}} \min_{\boldsymbol{\alpha}_k \geq 0} \max_{\mathbf{u} \in \mathcal{B}_2} \sum_{k=1}^{K} \left(\mathbf{v}^T \mathbf{D}(S^k) \mathbf{X}_k + \boldsymbol{\alpha}_k^T (2\mathbf{D}(S^k) - \mathbf{I}_n)\mathbf{X}_k \right) \mathbf{u} \\
&= \max_{\substack{S^k \subseteq [n] \\ S^k \in \mathcal{S}}} \min_{\boldsymbol{\alpha}_k \geq 0} \left\| \sum_{k=1}^{K} \mathbf{v}^T \mathbf{D}(S^k) \mathbf{X}_k + \boldsymbol{\alpha}_k^T (2\mathbf{D}(S^k) - \mathbf{I}_n)\mathbf{X}_k \right\|_2 .
\end{aligned}$$

We now enumerate all hyperplane arrangements and index them in an arbitrary order, i.e., denoted as $\left(S_i^1, \ldots, S_i^K\right)$, where $i \in [P_{conv}]$, $P_{conv} = |\mathcal{S}_K|$, $\mathcal{S}_K := \{(S_i^1, \ldots, S_i^K) : S_i^k \in \mathcal{S}, \forall k, i\}$. Then,

$$\begin{aligned}
(5) \iff \; & \forall i \in [P_{conv}], \min_{\boldsymbol{\alpha}_k \geq 0} \left\| \sum_{k=1}^{K} \mathbf{v}^T \mathbf{D}(S_i^k) \mathbf{X}_k + \boldsymbol{\alpha}_k^T (2\mathbf{D}(S_i^k) - \mathbf{I}_n)\mathbf{X}_k \right\|_2 \leq \beta \\
\iff \; & \forall i \in [P_{conv}], \exists \boldsymbol{\alpha}_{ik} \geq 0 \text{ s.t. } \left\| \sum_{k=1}^{K} \mathbf{v}^T \mathbf{D}(S_i^k) \mathbf{X}_k + \boldsymbol{\alpha}_{ik}^T (2\mathbf{D}(S_i^k) - \mathbf{I}_n)\mathbf{X}_k \right\|_2 \leq \beta.
\end{aligned}$$

We now use the same approach for the two-sided constraint in (3) to represent (3) as a finite dimensional convex problem as follows

$$\begin{aligned}
\max_{\substack{\mathbf{v} \\ \boldsymbol{\alpha}_{ik}, \boldsymbol{\alpha}'_{ik} \geq 0}} -\frac{1}{2}\|\mathbf{v} - \mathbf{y}\|_2^2 + \frac{1}{2}\|\mathbf{y}\|_2^2 \text{ s.t. } & \left\| \sum_{k=1}^{K} \mathbf{v}^T \mathbf{D}(S_i^k) \mathbf{X}_k + \boldsymbol{\alpha}_{ik}^T (2\mathbf{D}(S_i^k) - \mathbf{I}_n)\mathbf{X}_k \right\|_2 \leq \beta \\
& \left\| \sum_{k=1}^{K} -\mathbf{v}^T \mathbf{D}(S_i^k) \mathbf{X}_k + \boldsymbol{\alpha}_{ik}^{\prime T} (2\mathbf{D}(S_i^k) - \mathbf{I}_n)\mathbf{X}_k \right\|_2 \leq \beta, \, \forall i.
\end{aligned} \tag{7}$$

[3] The proof is presented in Appendix A.3, where the definition of m^* is given.

We note that the above problem is convex and strictly feasible for $\mathbf{v} = \boldsymbol{\alpha}_{ik} = \boldsymbol{\alpha}'_{ik} = \mathbf{0}$. Therefore, Slater's conditions and consequently strong duality holds [7], and (7) can be written as

$$\min_{\lambda_i,\lambda_i'\geq 0} \max_{\substack{\mathbf{v}\\ \boldsymbol{\alpha}_{ik},\boldsymbol{\alpha}'_{ik}\geq 0}} -\frac{1}{2}\|\mathbf{v}-\mathbf{y}\|_2^2+\frac{1}{2}\|\mathbf{y}\|_2^2+\sum_{i=1}^{P_{conv}}\lambda_i\left(\beta-\left\|\sum_{k=1}^{K}\mathbf{v}^T\mathbf{D}(S_i^k)\mathbf{X}_k+\boldsymbol{\alpha}_{ik}^T(2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\right\|_2\right)$$
$$+\sum_{i=1}^{P_{conv}}\lambda_i'\left(\beta-\left\|\sum_{k=1}^{K}-\mathbf{v}^T\mathbf{D}(S_i^k)\mathbf{X}_k+\boldsymbol{\alpha}_{ik}'^T(2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\right\|_2\right). \tag{8}$$

Next, we introduce new variables $\mathbf{z}_i, \mathbf{z}'_i \in \mathbb{R}^h$ to represent (8) as

$$\min_{\lambda_i,\lambda_i'\geq 0} \max_{\substack{\mathbf{v}\\ \boldsymbol{\alpha}_{ik},\boldsymbol{\alpha}'_{ik}\geq 0}} \min_{\substack{\mathbf{z}_i\in\mathcal{B}_2\\ \mathbf{z}'_i\in\mathcal{B}_2}} -\frac{1}{2}\|\mathbf{v}-\mathbf{y}\|_2^2+\frac{1}{2}\|\mathbf{y}\|_2^2+\sum_{i=1}^{P_{conv}}\lambda_i\left(\beta+\left(\sum_{k=1}^{K}\mathbf{v}^T\mathbf{D}(S_i^k)\mathbf{X}_k+\boldsymbol{\alpha}_{ik}^T(2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\right)\mathbf{z}_i\right)$$
$$+\sum_{i=1}^{P_{conv}}\lambda_i'\left(\beta+\left(\sum_{k=1}^{K}-\mathbf{v}^T\mathbf{D}(S_i^k)\mathbf{X}_k+\boldsymbol{\alpha}_{ik}'^T(2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\right)\mathbf{z}'_i\right), \tag{9}$$

which is concave in $\mathbf{v}, \boldsymbol{\alpha}_{ik}, \boldsymbol{\alpha}'_{ik}$ and convex in $\mathbf{z}_i$ and $\mathbf{z}'_i$. Moreover the set $\mathcal{B}_2$ is convex and compact. By recalling Sion's minimax theorem [27] for the inner max-min, we express the strong dual of (9) as

$$\min_{\lambda_i,\lambda_i'\geq 0} \min_{\substack{\mathbf{z}_i\in\mathcal{B}_2\\ \mathbf{z}'_i\in\mathcal{B}_2}} \max_{\substack{\mathbf{v}\\ \boldsymbol{\alpha}_{ik},\boldsymbol{\alpha}'_{ik}\geq 0}} -\frac{1}{2}\|\mathbf{v}-\mathbf{y}\|_2^2+\frac{1}{2}\|\mathbf{y}\|_2^2+\sum_{i=1}^{P_{conv}}\lambda_i\left(\beta+\left(\sum_{k=1}^{K}\mathbf{v}^T\mathbf{D}(S_i^k)\mathbf{X}_k+\boldsymbol{\alpha}_{ik}^T(2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\right)\mathbf{z}_i\right)$$
$$+\sum_{i=1}^{P_{conv}}\lambda_i'\left(\beta+\left(\sum_{k=1}^{K}-\mathbf{v}^T\mathbf{D}(S_i^k)\mathbf{X}_k+\boldsymbol{\alpha}_{ik}'^T(2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\right)\mathbf{z}'_i\right). \tag{10}$$

We now compute the maximum with respect to $\mathbf{v}, \boldsymbol{\alpha}_{ik}, \boldsymbol{\alpha}'_{ik}$ analytically to obtain the following problem

$$\begin{aligned}&\min_{\lambda_i,\lambda_i'\geq 0} \min_{\substack{\mathbf{z}_i\in\mathcal{B}_2\\ \mathbf{z}'_i\in\mathcal{B}_2}} \frac{1}{2}\left\|\sum_{i=1}^{P_{conv}}\sum_{k=1}^{K}\mathbf{D}(S_i^k)\mathbf{X}_k\left(\lambda_i'\mathbf{z}'_i-\lambda_i\mathbf{z}_i\right)-\mathbf{y}\right\|_2^2+\beta\sum_{i=1}^{P_{conv}}\sum_{k=1}^{K}\left(\lambda_i+\lambda_i'\right)\\ &\text{s.t. } (2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\mathbf{z}_i\geq 0,\ (2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\mathbf{z}'_i\geq 0, \forall i,k.\end{aligned} \tag{11}$$

Then, we apply a change of variables and define $\mathbf{w}_i = \lambda_i\mathbf{z}_i$ and $\mathbf{w}'_i = \lambda'_i\mathbf{z}'_i$. Thus, we obtain

$$\begin{aligned}&\min_{\mathbf{w}_i,\mathbf{w}'_i\in\mathbb{R}^h} \frac{1}{2}\left\|\sum_{i=1}^{P_{conv}}\sum_{k=1}^{K}\mathbf{D}(S_i^k)\mathbf{X}_k\left(\mathbf{w}'_i-\mathbf{w}_i\right)-\mathbf{y}\right\|_2^2+\beta\sum_{i=1}^{P_{conv}}\left(\|\mathbf{w}_i\|_2+\|\mathbf{w}'_i\|_2\right)\\ &\text{s.t. } (2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\mathbf{w}_i\geq 0,\ (2\mathbf{D}(S_i^k)-\mathbf{I}_n)\mathbf{X}_k\mathbf{w}'_i\geq 0, \forall i,k,\end{aligned} \tag{12}$$

where we eliminate the variables λ_i, λ'_i, since $\lambda_i = \|\mathbf{w}_i\|_2$ and $\lambda'_i = \|\mathbf{w}'_i\|_2$ are feasible and optimal. We now note that there will be m^* pairs $\{\mathbf{w}_i'^*, \mathbf{w}_i^*\}, \forall i \in [P_{conv}]$. Then, using the prescribed

$\{\mathbf{u}_j^*, w_j^*\}_{j=1}^{m^*}$, we evaluate the non-convex objective in (1) as follows

$$p_1^* \leq \frac{1}{2}\left\|\sum_{j=1}^{m^*}\sum_{k=1}^{K}(\mathbf{X}_k\mathbf{u}_j^*)_+ w_j^* - \mathbf{y}\right\|_2^2 + \frac{\beta}{2}\sum_{i=1,\mathbf{w}_i'^*\neq 0}^{P_{conv}}\left(\left\|\frac{\mathbf{w}_i'^*}{\sqrt{\|\mathbf{w}_i'^*\|_2}}\right\|_2^2 + \left\|\sqrt{\|\mathbf{w}_i'^*\|_2}\right\|_2^2\right)$$
$$+ \frac{\beta}{2}\sum_{i=1,\mathbf{w}_i^*\neq 0}^{P_{conv}}\left(\left\|\frac{\mathbf{w}_i^*}{\sqrt{\|\mathbf{w}_i^*\|_2}}\right\|_2^2 + \left\|\sqrt{\|\mathbf{w}_i^*\|_2}\right\|_2^2\right)$$

which has the same objective value with (12). Since strong duality holds for the convex program, we have $p_1^* = d_1^*$, which is equal to the value of (12) achieved by the prescribed parameters above.

A.3. Proof of Proposition A.1

We first review the basic properties of infinite size neural networks and introduce technical details to derive the dual of (3). We refer the reader to [2, 22] for further details. Let us first consider a measurable input space $\mathcal{X}$ with a set of continuous basis functions (i.e., neurons or filters in our context) $\psi_{\mathbf{u}} : \mathcal{X} \rightarrow \mathcal{R}$, which are parameterized by $\mathbf{u} \in \mathcal{B}_2$. Next, we use real-valued Radon measures with the uniform norms [23]. Let us consider a signed Radon measure denoted as μ. Now, we can use μ to formulate an infinite size neural network as $f(x) = \int_{\mathbf{u}\in\mathcal{B}_2}\psi_{\mathbf{u}}(x)d\mu(\mathbf{u})$, where $x \in \mathcal{X}$ is the input. The norm for μ is usually defined as its total variation norm, which is the supremum of $\int_{\mathbf{u}\in\mathcal{B}_2} g(\mathbf{u})d\mu(\mathbf{u})$ over all continuous functions $g(\mathbf{u})$ that satisfy $|g(\mathbf{u})| \leq 1$. Now, we consider the case where the basis functions are ReLUs, i.e., $\psi_{\mathbf{u}} = \left(\mathbf{x}^T\mathbf{u}\right)_+$. Then, the output of a network with finitely many neurons, say m neurons, can be written as

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

which can be obtained by selecting μ as a weighted sum of Dirac delta functions, i.e., $\mu = \sum_{j=1}^m w_j\delta(\mathbf{u}-\mathbf{u}_j)$. In this case, the total variation norm, denoted as $\|\mu\|_{TV}$, corresponds to the ℓ_1 norm $\|\mathbf{w}\|_1$.

Now, we ready to derive the dual of (3), which can be stated as follows (see Section 8.6 of [12] and Section 2 of [25] for further details)

$$d_1^* \leq p_{1,\infty} = \min_{\mu}\frac{1}{2}\left\|\int_{\mathbf{u}\in\mathcal{B}_2}\sum_{k=1}^{K}(\mathbf{X}_k\mathbf{u})_+ d\mu(\mathbf{u}) - \mathbf{y}\right\|_2^2 + \beta\|\mu\|_{TV}. \tag{13}$$

Although (13) involves an infinite dimensional integral form, by Caratheodory's theorem, we know that the integral can be represented as a finite summation, to be more precise, a summation of at most $n+1$ Dirac delta functions [22]. If we denote the number of Dirac delta functions as m^*, where $m^* \leq n+1$, then we have

$$p_{1,\infty} = \min_{\substack{\{\mathbf{u}_j,w_j\}_{j=1}^{m^*}\\ \mathbf{u}_j\in\mathcal{B}_2,\forall j}}\frac{1}{2}\left\|\sum_{j=1}^{m^*}\sum_{k=1}^{K}(\mathbf{X}_k\mathbf{u}_j)_+ w_j - \mathbf{y}\right\|_2^2 + \beta\|\mathbf{w}\|_1$$
$$= p_1^*$$

provided that $m \geq m^*$. We now need to show that strong duality holds, i.e., $p_1^* = d_1^*$.

We first note that the semi-infinite problem (3) is convex. Then, we prove that the optimal value is finite. Since $\beta > 0$, we know that $\mathbf{v} = \mathbf{0}$ is strictly feasible, and achieves 0 objective value. Moreover, since $-\|\mathbf{y} - \mathbf{v}\|_2^2 \leq 0$, the optimal objective value p_1^* is finite. Therefore, by Theorem 2.2 of [25], strong duality holds, i.e., $p_{1,\infty}^* = d_1^*$ provided that the solution set of (3) is nonempty and bounded. We also note that the solution set of (3) is the Euclidean projection of $\mathbf{y}$ onto a convex, closed and bounded set since $(\mathbf{X}_k \mathbf{u})_+$ can be expressed as the union of finitely many convex closed and bounded sets.