Reliably Learning the ReLU in Polynomial Time

Surbhi Goel
University of Texas at Austin
surbhi@cs.utexas.edu

Varun Kanade
University of Oxford
varunk@cs.ox.ac.uk

Adam Klivans
University of Texas at Austin
klivans@cs.utexas.edu

Justin Thaler
Georgetown University
jthaler@fas.harvard.edu

Abstract

We give the first dimension-efficient algorithms for learning Rectified Linear Units (ReLUs), which are functions of the form $\mathbf{x} \mapsto \mathsf{max}(0,\ \mathbf{w} \cdot \mathbf{x})$ with $\mathbf{w} \in \mathbb{S}^{n-1}$. Our algorithm works in the challenging Reliable Agnostic learning model of Kalai, Kanade, and Mansour [10] where the learner is given access to a distribution $\mathcal{D}$ on labeled examples but the labeling may be arbitrary. We construct a hypothesis that simultaneously minimizes the false-positive rate and the loss on inputs given positive labels by $\mathcal{D}$, for any convex, bounded, and Lipschitz loss function.

The algorithm runs in polynomial-time (in n) with respect to *any* distribution on $\mathbb{S}^{n-1}$ (the unit sphere in n dimensions) and for any error parameter $\epsilon = \Omega(1/\log n)$ (this yields a PTAS for a question raised by F. Bach on the complexity of maximizing ReLUs). These results are in contrast to known efficient algorithms for reliably learning linear threshold functions, where ϵ must be $\Omega(1)$ and strong assumptions are required on the marginal distribution. We can compose our results to obtain the first set of efficient algorithms for learning constant-depth networks of ReLUs.

Our techniques combine kernel methods and polynomial approximations with a "dual-loss" approach to convex programming. As a byproduct we obtain a number of applications including the first set of efficient algorithms for "convex piecewise-linear fitting" and the first efficient algorithms for noisy polynomial reconstruction of low-weight polynomials on the unit sphere.

1 Introduction

Let $\mathcal{X} = \mathbb{S}^{n-1}$, the set of all unit vectors in $\mathbb{R}^n$, and let $\mathcal{Y} = [0, 1]$. We define a ReLU (Rectified Linear Unit) to be a function $f : \mathcal{X} \to \mathcal{Y}$ equal to $\max(0,\ w \cdot x)$ where $w \in \mathbb{S}^{n-1}$ is a fixed element of $\mathbb{S}^{n-1}$ and $(w \cdot x)$ denotes the standard inner product. The ReLU is a key building block in the area of deep nets, where the goal is to construct a network or circuit of ReLUs that "fits" a training set with respect to various measures of loss. Recently, the ReLU has become the "activation function of choice" for practitioners in deep nets, as it leads to striking performance in various applications [14].

Surprisingly little is known about the computational complexity of learning even a single ReLU. In this work, we provide the first set of efficient algorithms for learning a ReLU. The algorithms succeed with respect to *any* distribution $\mathcal{D}$ on $\mathbb{S}^{n-1}$, tolerate arbitrary labelings (equivalently viewed as adversarial noise), and run in polynomial-time for any accuracy parameter $\epsilon = \Omega(1/\log n)$. This is in contrast to the problem of learning threshold functions, i.e., functions of the form $\mathsf{sign}(\mathbf{w} \cdot \mathbf{x})$, where only computational hardness results are known (unless stronger assumptions are made on the problem).

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Before formally defining our learning model and stating our results, we recall the following two fundamental problems: linear regression and learning a threshold function.

Problem 1.1 (Ordinary Least Squares)**.** *Let $\mathcal{D}$ be a distribution on $\mathbb{S}^{n-1} \times [0,1]$. Given i.i.d. examples drawn from $\mathcal{D}$, find $w \in \mathbb{S}^{n-1}$ that minimizes $\mathbb{E}_{(x,y)\sim\mathcal{D}}[(w \cdot x - y)^2]$.*

Problem 1.2 (Agnostically Learning a Threshold Function)**.** *Let $\mathcal{D}$ be a distribution on $\mathbb{S}^{n-1} \times \{0,1\}$. Given i.i.d. examples drawn from $\mathcal{D}$, find $w \in \mathbb{S}^{n-1}$ that minimizes $\Pr_{(x,y)\sim\mathcal{D}}[\mathsf{sign}(w \cdot x) \neq y]$.*

The term *agnostic* refers to the fact that the labeling on $\{-1, 1\}$ may be *arbitrary*. In this work, we relax the notion of success to *improper learning*, where the learner may output any polynomial-time computable hypothesis achieving a loss that is within ϵ of the optimal solution from the concept class.

Taken together, these two problems are at the core of many important techniques from modern Machine Learning and Statistics. It is well-known how to solve the Ordinary Least Squares problem and other variants of linear regression efficiently. We know of multiple polynomial-time solutions, all extensively in practice [17]. In contrast, the threshold learning problem defined above is thought to be computationally intractable due to the many existing hardness results in the literature [3, 7, 11, 12]!

The ReLU is a hybrid function that lies "in-between" a linear function and a threshold function in the following sense: restricted to inputs $\mathbf{x}$ such that $\mathbf{w} \cdot \mathbf{x} > 0$, the ReLU is linear, and for inputs $\mathbf{x}$ such that $\mathbf{w} \cdot \mathbf{x} \leq 0$, the ReLU thresholds the value $\mathbf{w} \cdot \mathbf{x}$ and simply outputs zero. In this sense, we could view the ReLU as a "one-sided" threshold function. Since learning a ReLU has aspects of both linear regression and threshold learning, it is not straightforward to identify a notion of loss that captures both of these aspects.

We introduce a natural model for learning ReLUs inspired by the Reliable Agnostic learning model that was introduced by Kalai et al. [10] in the context of Boolean functions. The goal will be to minimize both the false positive rate and a loss function (for example, square-loss) on points the distribution labels non-zero. In this work, we give efficient algorithms for learning a ReLU over the unit sphere with respect to any loss function that satisfies mild properties (convexity, monotonicity, boundedness, and Lipschitz-ness). The Reliable Agnostic model is motivated by the Neyman-Pearson criteria, and is intended to capture settings in which false positive errors are more costly than false negative errors (e.g., spam detection) or vice versa. We observe that the asymmetric manner in which the Reliable Agnostic model [10] treats different types of errors naturally corresponds to the one-sided nature of a ReLU. In particular, there may be settings in which mistakenly predicting a positive value instead of zero carries a high cost.

More formally, for a function h and distribution $\mathcal{D}$ over $\mathbb{R}^n \times [0,1]$ define the following losses

$$\mathcal{L}_{=0}(h) = \Pr_{(\mathbf{x},y)\sim\mathcal{D}}[h(\mathbf{x}) \neq 0 \wedge\ y = 0]$$
$$\mathcal{L}_{>0}(h) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(h(\mathbf{x}), y) \cdot \mathbb{I}(y > 0)].$$

Here, ℓ is a desired loss function, and $\mathbb{I}(y > 0)$ equals 0 if $y \leq 0$ and 1 otherwise. These two quantities are respectively the false-positive rate and the expected loss (under ℓ) on examples for which the true label y is positive.[1]

Let $\mathcal{C}$ be a class of functions mapping $\mathbb{S}^{n-1}$ to $[0,1]$ (e.g., $\mathcal{C}$ may be the class of all ReLUs). Let $\mathcal{C}^+ = \{c \in \mathcal{C} \mid \mathcal{L}_{=0}(c) = 0\}$. We say $\mathcal{C}$ is *reliably learnable* if there exists a learning algorithm $\mathcal{A}$ that (with high probability) outputs a hypothesis that 1) has at most ϵ false positive rate and 2) on points with positive labels, has expected loss that is within ϵ of the best c from $\mathcal{C}^+$. That is, the hypothesis must be both *reliable* and competitive with the optimal classifier from the class $\mathcal{C}^+$ (*agnostic*).

2 Main Results

All of our results hold for loss functions ℓ that satisfy convexity, monotonicity, boundedness, and Lipschitz-ness. For brevity, we avoid making these requirements explicit in the theorem statements, and we omit the dependence of the runtime on the failure probability δ of the algorithm or on the boundedness and Lipschitz parameters of the loss function.

[1] We restrict $\mathcal{Y} = [0,1]$ as it is a natural setting for the case of ReLUs. However, our results can easily be extended to larger ranges.

Theorem 2.1. *Let $\mathcal{C} = \{\mathbf{x} \mapsto \mathsf{max}(0, \mathbf{w} \cdot \mathbf{x})\colon \|\mathbf{w}\|_2 \leq 1\}$ be the class of ReLUs with weight vectors $\mathbf{w}$ satisfying $\|\mathbf{w}\|_2 \leq 1$. There exists a learning algorithm $\mathcal{A}$ that reliably learns $\mathcal{C}$ in time $2^{O(1/\epsilon)} \cdot n^{O(1)}$.*

Remark 2.2. *We can obtain the same complexity bounds for learning ReLUs in the standard agnostic model with respect to the same class of loss functions. This yields a PTAS (polynomial-time approximation scheme) for an optimization problem regarding ReLUs posed by Bach [2].*

For the problem of learning threshold functions, all known polynomial-time algorithms require strong assumptions on the marginal distribution (e.g., Gaussian [11] or large-margin [18]). In contrast, for ReLUs, we succeed with respect to *any* distribution on $\mathbb{S}^{n-1}$. We leave open the problem of improving the dependence of Theorem 2.1 on ϵ. We note that for the problem of learning threshold functions—even assuming the marginal distribution is Gaussian—the run-time complexity must be at least $n^{\Omega(\log 1/\epsilon)}$ under the widely believed assumption that learning sparse parities is hard [13]. Further, the best *known* algorithms for agnostically learning threshold functions with respect to Gaussians run in time $n^{O(1/\epsilon^2)}$ [4, 11]. Contrast this to our result for learning ReLUs, where we give polynomial-time algorithms even for ϵ as small as $1/\log n$.

We can compose our results to obtain efficient algorithms for small-depth networks of ReLUs. For brevity, here we state results only for linear combinations of ReLUs (which are often called *depth-two* networks of ReLUs, see, e.g., [5]).

Theorem 2.3. *Let $\mathcal{C}$ be a depth-2 network of ReLUs with k hidden units. Then $\mathcal{C}$ is reliably learnable in time $2^{O(\sqrt{k}/\epsilon)} \cdot n^{O(1)}$.*

The above results are perhaps surprising in light of the hardness result due to Livni et al. [15] who showed that for $\mathcal{X} = \{0,1\}^n$, learning the difference of even two ReLUs is as hard as learning a threshold function.

We also obtain results for *noisy polynomial reconstruction* on the sphere (equivalently, agnostically learning a polynomial) with respect to a large class of loss functions:

Theorem 2.4. *Let $\mathcal{C}$ be the class of polynomials $p\colon \mathbb{S}^{n-1} \to [-1,1]$ in n variables such that that the total degree of p is at most d, and the sum of squares of coefficients of p (in the standard monomial basis) is at most B. Then $\mathcal{C}$ is agnostically learnable under any (unknown) distribution over $\mathbb{S}^{n-1} \times [-1,1]$ in time $\mathsf{poly}(n, d, B, 1/\epsilon)$.*

Andoni et al. [1] were the first to give efficient algorithms for noisy polynomial reconstruction over non-Boolean domains. In particular, they gave algorithms that succeed on the unit cube but require an underlying product distribution and do not work in the agnostic setting (they also run in time exponential in the degree d).

We also establish a novel connection between learning networks of ReLUs and a broad class of piecewise-linear regression problems studied in machine learning and optimization. The following problem was defined by Boyd and Magnani [16] as a generalization of the well-known MARS (multivariate adaptive regression splines) framework due to Friedman [9]:

Problem 2.5 (Convex Piecewise-Linear Regression: Max k-Affine)**.** *Let $\mathcal{C}$ be the class of functions of the form $f(x) = \mathsf{max}(\mathbf{w}_1 \cdot \mathbf{x}, \ldots, \mathbf{w}_k \cdot \mathbf{x})$ with $\mathbf{w}_1, \ldots, \mathbf{w}_k \in \mathbb{S}^{n-1}$ mapping $\mathbb{S}^{n-1}$ to $\mathbb{R}$. Let $\mathcal{D}$ be an (unknown) distribution on $\mathbb{S}^{n-1} \times [-1,1]$. Given i.i.d. examples drawn from $\mathcal{D}$, output h such that $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[(h(\mathbf{x}) - y)^2] \leq \min_{c\in\mathcal{C}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[(c(\mathbf{x}) - y)^2] + \epsilon$.*

Applying our learnability results for networks of ReLUs, we obtain the first polynomial-time algorithms for solving the above *max-k-affine* regression problem and the *sum of max-2-affine* regression problem when $k = O(1)$. Boyd and Magnani specifically highlight the case of $k = O(1)$ and provide a variety of heuristics; we obtain the first provably efficient results.

Theorem 2.6. *There is an algorithm $\mathcal{A}$ for solving the convex piecewise-linear fitting problem (cf. Definition 2.5) in time $2^{O\left((k/\epsilon)^{\log k}\right)} \cdot n^{O(1)}$.*

We can also use our results for learning networks of ReLUs to learn the so-called "leaky ReLUs" and "parameterized" ReLUs (PReLUs). We obtain these results by composing various "ReLU gadgets," i.e., constant-depth networks of ReLUs with a small number of bounded-weight hidden units.

We also establish the first hardness results for learning a ReLU. These results highlight the difference between learning Boolean and real-valued functions and justify our focus on input distributions over

$\mathbb{S}^{n-1}$, rather than the Boolean hypercube. Notice that for the domain $\mathcal{X} = \{0,1\}^n$, the conjunction of literals $x_1, \ldots, x_k$ can be computed exactly as $\mathsf{max}(0,\ x_1 + \cdots + x_k - (k-1))$. Due to known connections between agnostically learning conjunctions and learning sparse parities with noise [8] we obtain the following result.

Theorem 2.7. *Let $\mathcal{C}$ be the class of ReLUs over the domain $\mathcal{X} = \{0,1\}^n$. Then any algorithm for reliably learning $\mathcal{C}$ in time $g(\epsilon) \cdot \mathsf{poly}(n)$ for any function g will give a polynomial time algorithm for learning $\omega(1)$-sparse parities with noise (for any $\epsilon = O(1)$).*

Efficiently learning sparse parities (of any superconstant length) with noise is considered one the most challenging problems in theoretical computer science.

3 Techniques and Related Work

Let $\mathcal{C}$ be the class of all ReLUs, and let $S = \{(x^1, y^1), \ldots, (x^m, y^m)\}$ be a training set of examples drawn i.i.d. from some arbitrary distribution on $\mathcal{D}$ on $\mathbb{S}^{n-1} \times [0,1]$. To obtain our main result for learning a single ReLU, our starting point is the following optimization problem.

Optimization Problem 1

$$\begin{aligned} \underset{\mathbf{w} \in \mathbb{S}^{n-1}}{\text{minimize}} \quad & \sum_{i: y_i > 0} \ell(y_i, \mathsf{max}(0, \mathbf{w} \cdot \mathbf{x}_i)) \\ \text{subject to} \quad & \mathsf{max}(0, \mathbf{w} \cdot \mathbf{x}_i) = 0 \quad \text{for all } i \text{ such that } y_i = 0 \\ & \|\mathbf{w}\|_2 \leq 1 \end{aligned}$$

Unfortunately Optimization Problem 1 is not convex in $\mathbf{w}$, and hence it may not be possible to find an optimal solution in polynomial time. Instead, we will give an efficient approximate solution that will suffice for reliable learning.

Our starting point will be to prove the existence of low-degree, low-weight polynomial approximators for every $c \in \mathcal{C}$. The polynomial method has a well established history in computational learning theory (e.g., Kalai et al. [11] for agnostically learning halfspaces under distributional assumptions), and we can apply classical techniques from approximation theory and recent work due to Sherstov [19] to construct low-weight, low-degree approximators for any ReLU.

We can then relax Optimization Problem 1 to the space of low-weight polynomials and follow the approach of Shalev-Shwartz et al. [18] who used tools from Reproducing Kernel Hilbert Spaces (RKHS) to learn low-weight polynomials efficiently (Shalev-Shwartz et al. focused on a relaxation of the 0/1 loss for halfspaces).

The main challenge is to obtain reliability; i.e., to simultaneously minimize the false-positive rate and the loss dictated by the objective function. To do this we take a "dual-loss" approach and carefully construct two loss functions that will both be minimized with high probability. Proving that these losses generalize for a large class of objective functions is subtle and requires "clipping" in order to apply the appropriate Rademacher bound.

Due to space constraints, we omit a technical discussion of our other results and refer the reader to the full version [2] of the paper.

4 Summary and Open Questions

We have given the first set of efficient algorithms for ReLUs in a natural learning model. ReLUs are both effective in practice and, unlike linear threshold functions (halfspaces), admit non-trivial learning algorithms for *all* distributions with respect to adversarial noise. We "sidestepped" the hardness results in Boolean function learning by focusing on problems that are not entirely scale-invariant with respect to the choice of domain (e.g., reliably learning ReLUs). The obvious open question is to improve the dependence of our main result on $1/\epsilon$. We have no reason to believe that $2^{O(1/\epsilon)}$ is the best possible.

[2] Full version available at `https://arxiv.org/pdf/1611.10258.pdf`

Acknowledgments

Adam Klivans acknowledges support from an NSF Grant CCF-1018829.

References

[1] Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning sparse polynomial functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 500–510, 2014.

[2] Francis Bach. Breaking the curse of dimensionality with convex neural networks. 2014.

[3] Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *STOC*, pages 105–117. ACM, 2016.

[4] Ilias Diakonikolas, Daniel M. Kane, and Jelani Nelson. Bounded independence fools degree-2 threshold functions. In *FOCS*, pages 11–20. IEEE Computer Society, 2010.

[5] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 907–940. JMLR.org, 2016.

[6] Bassey Etim. Approve or Reject: Can You Moderate Five New York Times Comments? *The New York Times*, 2016. Originally published September 20, 2016. Retrieved October 4, 2016.

[7] V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM J. Comput*, 39(2):606–645, 2009.

[8] Vitaly Feldman and Pravesh Kothari. Agnostic learning of disjunctions on symmetric distributions. *Journal of Machine Learning Research*, 16:3455–3467, 2015.

[9] Jerome H. Friedman. Multivariate adaptive regression splines. *Ann. Statist*, 1991.

[10] Adam Tauman Kalai, Varun Kanade, and Yishay Mansour. Reliable agnostic learning. *Journal of Computer and System Sciences*, 78(5):1481–1495, 2012.

[11] Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

[12] A. R. Klivans and A. A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *J. Comput. Syst. Sci*, 75(1):2–12, 2009.

[13] Adam Klivans and Pravesh Kothari. Embedding hard learning problems into gaussian space. In *RANDOM*, 2014.

[14] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7533):436–444, May 2015.

[15] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. pages 855–863, 2014.

[16] Alessandro Magnani and Stephen P. Boyd. Convex piecewise-linear fitting. *Optimization and Engineering*, 10(1):1–17, 2009.

[17] Phillippe Rigollet. *High-Dimensional Statistics*. MIT, 1st edition, 2015.

[18] Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Learning kernel-based halfspaces with the 0-1 loss. *SIAM J. Comput.*, 40(6):1623–1646, 2011.

[19] Alexander A. Sherstov. Making polynomials robust to noise. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 747–758, New York, NY, USA, 2012. ACM.

[20] Yuchen Zhang, Jason Lee, and Michael Jordan. ℓ_1 networks are improperly learnable in polynomial-time. In *ICML*, 2016.