Learning ReLUs via Gradient Descent

Authored by:

Mahdi Soltanolkotabi

Abstract

In this paper we study the problem of learning Rectified Linear Units (ReLUs) which are functions of the form $vct{x}mapsto max(0,langle vct{w},vct{x}rangle)$ with $vct{w}inRd$ denoting the weight vector. We study this problem in the high-dimensional regime where the number of observations are fewer than the dimension of the weight vector. We assume that the weight vector belongs to some closed set (convex or nonconvex) which captures known side-information about its structure. We focus on the realizable model where the inputs are chosen i.i.d.from a Gaussian distribution and the labels are generated according to a planted weight vector. We show that projected gradient descent, when initialized at $vct{0}$, converges at a linear rate to the planted model with a number of samples that is optimal up to numerical constants. Our results on the dynamics of convergence of these very shallow neural nets may provide some insights towards understanding the dynamics of deeper architectures.

1 Paper Body

Nonlinear data-fitting problems are fundamental to many supervised learning tasks in signal processing and machine learning. Given training data consisting of n pairs of input features xi ? Rd and desired outputs yi ? R we wish to infer a function that best explains the training data. In this paper we focus on fitting Rectified Linear Units (ReLUs) to the data which are functions ?w ? Rd ? R of the form ?w (x) = max (0, ?w, x?) . A natural approach to fitting ReLUs to data is via minimizing the least-squares misfit aggregated over the data. This optimization problem takes the form min

w?Rd

L(w) ?=

1 n 2 ? (max (0, ?w, xi ?) ? yi) n i=1

subject to R(w) ? R,

(1.1)

with R ? Rd ? R denoting a regularization function that encodes prior information on the weight vector. Fitting nonlinear models such as ReLUs have a rich history in statistics and learning theory [12] with interesting new developments emerging [6] (we shall discuss all these results in greater detail in Section 5). Most recently, nonlinear data fitting problems in the form of neural networks (a.k.a. deep learning) have emerged as powerful tools for automatically extracting interpretable and actionable information from raw forms of data, leading to striking breakthroughs in a multitude of applications [13, 15, 4]. In these and many other empirical domains it is common to use local search heuristics such as gradient or stochastic gradient descent for nonlinear data fitting. These local search heuristics are surprisingly effective on real or randomly generated data. However, despite their empirical success the reasons for their effectiveness remains mysterious. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Focusing on fitting ReLUs, a-priori it is completely unclear why local search heuristics such as gradient descent should converge for problems of the form (1.1), as not only the regularization function maybe nonconvex but also the loss function! Efficient fitting of ReLUs in this highdimensional setting poses new challenges: When are the iterates able to escape local optima and saddle points and converge to global optima? How many samples do we need? How does the number of samples depend on the a-priori prior knowledge available about the weights? What regularizer is best suited to utilizing a particular form of prior knowledge? How many passes (or iterations) of the algorithm is required to get to an accurate solution? At the heart of answering these questions is the ability to predict convergence behavior/rate of (non)convex constrained optimization algorithms. In this paper we build up on a new framework developed in the context of phase retrieval [21] for analyzing nonconvex optimization problems to address such challenges.

2

Precise measures for statistical resources

We wish to characterize the rates of convergence for the projected gradient updates (3.2) as a function of the number of samples, the available prior knowledge and the choice of the regularizer. To make these connections precise and quantitative we need a few definitions. Naturally the required number of samples for reliable data fitting depends on how well the regularization function R can capture the properties of the weight vector w. For example, if we know that the weight vector is approximately sparse, naturally using an ‘1 norm for the regularizer is superior to using an ‘2 regularizer. To quantify this capability we first need a couple of standard definitions which we adapt from [17, 18, 21]. Definition 2.1 (Descent set and cone) The set of descent of a function R at a point w? is defined as DR (w?) = {h ? R(w? + h) ? R(w?)}. The cone of descent is defined as a closed cone CR (w?) that contains the descent set, i.e. DR (w?) ? CR (w?). The tangent cone is the conic hull of the descent set. That is, the smallest closed cone CR (w?) obeying DR (w?) ? CR (w?). We note that the capability of the regularizer R in capturing the properties of the unknown weight vector w? depends on the size of the descent cone CR (w?). The smaller this cone is the more suited the function R is at capturing the properties of w? . To quantify the size of this set we shall use the notion of mean width. Definition 2.2 (Gaussian width) The Gaussian width of a set C ? Rd is defined as: ?(C) ?= Eg [sup ?g, z?], z?C

where the expectation is taken over g ? N (0, Ip). Throughout we use B d /Sd?1 to denote the the unit ball/sphere of Rd . We now have all the definitions in place to quantify the capability of the function R in capturing the properties of the unknown parameter w? . This naturally leads us to the definition of the minimum required number of samples. Definition 2.3 (minimal number of samples) Let CR (w?) be a cone of descent of R at w? . We define the minimal sample function as M(R, w?) = ? 2 (CR (w?) ? B d). We shall often use the short hand n0 = M(R, w?) with the dependence on R, w? implied. We note that n0 is exactly the minimum number of samples required for structured signal recovery from linear measurements when using convex regularizers [3, 1]. Specifically, the optimization problem n

? ? (yr ? ?xi , w ?)

2

subject to R(w) ? R(w?),

i=1

2

(2.1)

succeeds at recovering an unknown weight vector w? with high probability from n observations of the form yi = ?ai , w? ? if and only if n ? n0 .1 While this result is only known to be true for convex regularization functions we believe that n0 also characterizes the minimal number of samples even for nonconvex regularizers in (2.1). See [17] for some results in the nonconvex case as well as the role this quantity plays in the computational complexity of projected gradient schemes for linear inverse problems. Given that with nonlinear samples we have less information (we loose some information compared to linear observations) we can not hope to recover the weight vector from n ? n0 when using (1.1). Therefore, we can use n0 as a lower-bound on the minimum number of observations required for projected gradient descent iterations (3.2) to succeed at finding the right model.

3

Theoretical results for learning ReLUs

A simple heuristic for optimizing (1.1) is to use gradient descent. One challenging aspect of the above loss function is that it is not differentiable and it is not clear how to run projected gradient descent. However, this does not pose a fundamental challenge as the loss function is differentiable except for isolated points and we can use the notion of generalized gradients to define the gradient at a non-differentiable point as one of the limit points of the gradient in a local neighborhood of the non-differentiable point. For the loss in (1.1) the generalized gradient takes the form ?L(w) ?=

1 n ? (ReLU (?w, xi ?) ? yi) (1 + sgn(?w, xi ?)) xi . n i=1

(3.1)

Therefore, projected gradient descent takes the form w? +1 = PK (w? ? ?? ?L(w?)) ,

(3.2)

where ?? is the step size and K = {w ? Rd ? R(w) ? R} is the constraint set with PK denoting the Euclidean projection onto this set. Theorem 3.1 Let w? ? Rd be an arbitrary weight vector and R ? Rd ? R be a proper function (convex or nonconvex). Suppose the feature vectors xi ? Rd are i.i.d. Gaussian random vectors distributed as N (0, I) with the corresponding labels given by yi = max (0, ?xi , w? ?) . To estimate w? , we start from the initial point w0 = 0 and apply the Projected Gradient Descent (PGD) updates of the form w? +1 = PK (w? ? ?? ?L(w?)) ,

(3.3)

with K ?= {w ? Rd ? R(w) ? R(w?)} and ?L defined via (3.1). Also set the learning parameter sequence to ?0 = 2 and ?? = 1 for all ? = 1, 2, . . . and let n0 = M(R, w?), per Definition 2.3, be our lower bound on the number of observations. Also assume n ¿ cn0 ,

(3.4)

holds for a fixed numerical constant c. Then there is an event of probability at least 1 ? 9e??n such that on this event the updates (3.3) obey 1 ? ?w? ? w? ¿2 ? ( ) ?w? ¿2 . 2

(3.5)

Here ? is a fixed numerical constant. The first interesting and perhaps surprising aspect of this result is its generality: it applies not only to convex regularization functions but also nonconvex ones! As we mentioned earlier the optimization problem in (1.1) is not known to be tractable even for convex regularizers. Despite the nonconvexity of both the objective and regularizer, the theorem above shows that with a near minimal number 1

We would like to note that n0 only approximately characterizes the minimum number of samples required. A ? ?(t+1) ? more precise characterization is ??1 (? 2 (CR (w?) ? Bd)) ? ? 2 (CR (w?) ? Bd) where ?(t) = 2 ?(2t) ? t. 2 However, since our results have unspecified constants we avoid this more accurate characterization.

3

1 ReLU samples Linear samples

Estimation error

0.8 0.6 0.4 0.2 0

0

5

10

15

20

Figure 1: Estimation error (?w? ? w? ¿2) obtained via running PGD iterates as a function of the number of iterations ? . The plots are for two different observations models: 1) ReLU observations of the form y =ReLU(Xw?) and 2) linear observations of the form y = Xw? . The bold colors depict average behavior over 100 trials. None bold color depict the estimation error of some sample trials. of data samples, projected gradient descent provably learns the original weight vector w? without getting trapped in any local optima. Another interesting aspect of the above result is that the convergence rate is

linear. Therefore, to achieve a relative error of the total number of iterations is on the order of O(log(1/)). Thus the overall computational complexity is on the order of O (nd log(1/)) (in general the cost is the total number of iterations multiplied by the cost of applying the feature matrix X and its transpose). As a result, the computational complexity is also now optimal in terms of dependence on the matrix dimensions. Indeed, for a dense matrix even verifying that a good solution has been achieved requires one matrix-vector multiplication which takes O(nd) time.

4

Numerical experiments

In this section we carry out a simple numerical experiment to corroborate our theoretical results. For this purpose we generate a unit norm sparse vector w? ? Rd of dimension d = 1000 containing s = d/50 non-zero entries. We also generate a random feature matrix X ? Rn?d with n = ?8s log(d/s)? and containing i.i.d. N (0, 1) entries. We now take two sets of observations of size n from ? ? : ? ReLU observations: the response vector is equal to y =ReLU(Xw?). ? Linear observations: the response is y = Xw? . We apply the projected gradient iterations to both observation models starting from w0 = 0. For the ReLU observations we use the step size discussed in Theorem 3.1. For the linear model we apply projected gradient descent updates of the form w? +1 = PK (w? ?

1 T X (Xw? ? y)) . n

In both cases we use the regularizer R(w) = ?w¿0 so that the projection only keeps the top s entries of the vector (a.k.a. iterative hard thresholding). In Figure 1 the resulting estimation errors (?w? ? w? ¿2) is depicted as a function of the number of iterations ? . The bold colors depict average behavior over 100 trials. The estimation error of some sample trials are also depicted in none bold 4

colors. This plot clearly show that PGD iterates applied to ReLU observations converge quickly to the ground truth. This figure also clearly demonstrates that the behavior of the PGD iterates applied to both models are similar, further corroborating the results of Theorem 3.1. We note that the sample complexity used in this simulation is 8s log(n/s) which is a constant factor away from n0 ? s log(n/s) confirming our assertion that the required sample complexity is a constant factor away from n0 (as predicted by Theorem 3.1).

5

Discussions and prior art

There is a large body of work on learning nonlinear models. A particular class of such problems that have been studied are the so called idealized Single Index Models (SIMs) [9, 10]. In these problems the inputs are labeled examples {(xi , yi)}ni=1 ? Rd ? R which are guaranteed to satisfy yi = f (?w, xi ?) for some w ? Rd and nondecreasing (Lipchitz continuous) f ? R ? R. The goal in this problem is to find a (nearly) accurate such f and w. An interesting polynomial-time algorithm called the Isotron exists for this problem [12, 11]. In principle, this approach can also be used to fit ReLUs. However, these results differ from ours in term of both assumptions and results. On the one had, the assumptions are slightly more restrictive as they require bounded features xi , outputs yi and weights. On the other hand, these result hold for much more general distributions and more general models than the realizable model studied in this paper. These results also do not apply in the high dimensional regime where the number of observations is significantly smaller than the number of parameters (see [5] for some results in this direction). In the realizable case, the Isotron result require O(1) iterations to achieve error in objective value. In comparison, our results guarantee convergence to a solution with relative error (?w? ? w? ¿2 / ?w? ¿2 ?) after log (1/) iterations. Focusing on the specific case of ReLU functions, an interesting recent result [6] shows that reliable learning of ReLUs is possible under very general but bounded distributional assumptions. To achieve an accuracy of the algorithm runs in poly(1/) time. In comparison, as mentioned earlier our result rquires log(1/) iterations for reliable parameter estimation. We note however we study the problem in different settings and a direct comparison is not possible between the two results. We would like to note that there is an interesting growing literature on learning shallow neural networks with a single hidden layer with i.i.d. inputs, and under a realizable model (i.e. the labels are generated from a network with planted weights) [23, 2, 25]. For isotropic Gaussian inputs, [23] shows that with two hidden unites (k = 2) there are no critical points for configurations where both weight vectors fall into (or outside) the cone of ground truth weights. With the same assumptions, [2] proves that for a single-hidden ReLU network with a single non-overlapping convolutional filter, all local minimizers of the population loss are global; they also give counter-examples in the overlapping case and prove the problem is NP-hard when inputs are not Gaussian. [25] studies general single-hidden layer networks and shows that a version of gradient descent which uses a fresh batch of samples in each iteration converges to the planted model. This holds using an initialization obtained via a tensor decomposition method. Our approach and convergence results differ from this literature in a variety of different ways. First, we focus on zero hidden layers with a regularization term. Some of this literature focuses on one-hidden layers without (or with specific) regularization. Second, unlike some of these results such as [2, 14], we study the optimization properties of the empirical function, not its expected value. Third, we initialize at zero in lieu of sophisticated initialization schemes. Finally, our framework does not require a fresh batch of samples per new gradient iteration as in [25]. We also note that several publications study the effect of over-parametrization on the training of neural networks without any regularization [19, 8, 16, 22]. Therefore, the global optima are not unique and hence the solutions may not generalize. In comparison we study the problem with an arbitrary regularization which allows for a unique global optima.

6 6.1

Proofs Preliminaries

In this section we gather some useful results on concentration of stochastic processes which will be crucial in our proofs. These results are mostly adapted from [21]. We begin with a lemma which is a direct consequence of Gordon?s escape from the mesh lemma [7].

5

Lemma 6.1 Assume C ? Rd is a cone and Sd?1 is the unit sphere of Rd . Also assume that n ? max (20

? 2 (C ? Sd?1) 1 , ? 1) , ?2 2?

for a fixed numerical constant c. Then for all h ? C ?

1 n 2 2 2 ?(?xi , h?) ? ?h¿2 ? ? ? ?h¿2 , n i=1 ?2

holds with probability at least 1 ? 2e? 360 n . We also need a generalization of the above lemma stated below. Lemma 6.2 ([21]) Assume C ? Rd is a cone (not necessarily convex) and Sd?1 is the unit sphere of Rd . Also assume that n ? max (80

? 2 (C ? Sd?1) 2 , ? 1) , ?2 ?

for a fixed numerical constant c. Then for all u, h ? C ?

1 n ? ??xi , u??xi , h? ? u h? ? ? ?u¿2 ?h¿2 , n i=1 ?2

holds with probability at least 1 ? 6e? 1440 n . We next state a generalization of Gordon?s escape through the mesh lemma also from [21]. Lemma 6.3 ([21]) Let s ? Rd be fixed vector with nonzero entries and construct the diagonal matrix S = diag(s). Also, let X ? Rn?d have i.i.d. N (0, 1) entries. Furthermore, assume T ? Rd and define bd (s) = E[?Sg¿2], where g ? Rd is distributed as N (0, In). Also, define ?(T) ?= max ?v¿2 . v?T

Then for all u ? T ??SAu¿2 ? bd (s) ?u¿2 ? ? ?s¿? ?(T) + ?, holds with probability at least 1 ? 6e

?

?2 8?s?2 ? 2 (T) ‘?

.

The previous lemma leads to the following Corollary. Corollary 6.4 Let s ? Rd be fixed vector with nonzero entries and assume T ? B d . Furthermore, assume 2

2

?s¿2 ? max (20 ?s¿? Then for all u ? T ,

? 2 (T) 3 , ? 1) . ?2 2?

RRR n 2 R 2 RRR ?i=1 si (?xi , u?) ? ?u?2 RRRRR ? ?, ‘2 RR 2 RRRR RRR ?s¿2 R ?2

2

holds with probability at least 1 ? 6e? 1440 ?s¿2 . 6.2

Convergence proof (Proof of Theorem 3.1)

In this section we shall prove Theorem 3.1. Throughout, we use the shorthand C to denote the descent cone of R at w? , i.e. C = CR (w?). We begin by analyzing the first iteration. Using w0 = 0 we have w1 ?= PK (w0 ? ?0 ?L(w0)) = PK (

2 n 2 n ? ? yi xi) = PK (? ReLU(?xi , w ?)xi) . n i=1 n i=1 6

We use the argument of [21][Page 25, inequality (7.34)] which shows that ?w1 ? w? ¿2 ? 2 ? sup uT (u?C?Bd

Using ReLU(z) =

z+?z? 2

2 n ? ? ? ReLU(?xi , w ?)xi ? w) . n i=1

(6.1)

we have

2 n 1 n T ? ? ? T 1 ? ? ReLU(?xi , w ?)?xi , u? ? ?u, w ? = u (X X ? I) w + ? ??xi , w ?? ?xi , u?. (6.2) n i=1 n n i=1 We proceed by bounding the first term in the above equality. To this aim we decompose u in the direction parallel/perpendicular to that of w? and arrive at T ? 1 (uT w?) 1 w? (w?) ? ? T 1 T ? uT (X T X ? I) w? = (w) (u, Xw? ?, X X ? I) w + ?X I ? 2 2 ? ? n n n ? ? ?w ¿2 ?w ¿2 2

T ? ?g¿2 ? ? ?w? ¿ w? (w?) ? u, ? 1 + ? 2 aT I ? 2 n ? n ? ? ?w? ¿2 ? RRR ?g?2 RRR ?w? ? T ? w? (w?) ? ‘2 ‘ R ? R ? ?w ¿2 RRR u, ? 1RRRRR + ? 2 sup aT I ? 2 n u?C?Bd ? RRR n RRR ?w? ¿2 ?

?(uT w?)

(6.3)

with g ? Rn and a ? Rd are independent random Gaussian random vectors distributed as N (0, Id) and N (0, In). By concentration of Chi-squared random variables 2

??g¿2 /n ? 1? ? ?, holds with probability at least 1 ? 2e?n

?2 8

(6.4)

. Also,

T ? w? (w?) ? 1 1 ? aT I ? u ? ? (? (C ? B d) + ?) , 2 ? n n ? ? ?w ¿2

?2

holds with probability at least 1 ? e? 2 . Plugging (6.4) with ? = ? 2 (C ? B d), then (6.3), as long as n ? 36 ?2 sup u?C?Bd

? 6

and (6.5) with ? =

(6.5) ?? n 6

1 ? uT (X T X ? I) w? ? ?w? ¿2 , n 2

into

(6.6)

?2

holds with probability at least 1 ? 3e?n 288 . We now focus on bounding the second term in (6.2). To this aim we decompose u in the direction parallel/perpendicular to that of w? and arrive at RRR RRR 1 n 1 n ??xi , w? ?? ?xi , w? ? 1 n RRR , ? + ? ? ??xi , w? ?? ?xi , u?? = RRRRR(uT w?) ? ??x , w ?? ?x , u ? ? i i ? RRR 2 n i=1 n i=1 n i=1 ?w? ¿2 RRR RR R RRR n ??xi , w? ?? ?xi , w? ? RRRR 1 n 1 ? ?w? ¿2 RRRRR ? RRR + ? ? ??xi , w? ?? ?xi , u? ?? . (6.7) 2 RRR n i=1 RRR n i=1 ?w? ¿2 with u? = (I ?

w? (w?)T ?w? ?2‘ 2

) u. Now note that ?

??xi ,w? ???xi ,w? ? ?w? ?2‘

is sub-exponential and

2

??xi , w? ?? ?xi , w? ? 2

?w? ¿2

?

? c,

?1

with fixed numerical constant. Thus by Bernstein?s type inequality ([24][Proposition 5.16]) RRR n R ??xi , w? ?? ?xi , w? ? RRRR RRR 1 RRR ? t, RRR n ? 2 RRR ?w? ¿2 RR i=1 7

(6.8)

holds with probability at least 1 ? 2e??n min(t

2

,t)

with ? a fixed numerical constant.. Also note that ? ?1 n 1 n ? ? ? ??xi , w? ??2 ?1 ?g, u? ?. ? ??xi , w ?? ?xi , u? ? ? ? n i=1 n i=1 n

Furthermore,

1 n

2

2

n ?i=1 ??xi , w? ?? ? (1 + ?) ?w? ¿2 , holds with probability at least 1 ? 2e?n

sup u?C?Sd?1

holds with probability at least 1 ? e? ?

??g, u? ?? ? (2? (C ? S

d?1

?2 2

. Combining the last two inequalities we conclude that

holds with probability at least 1 ? 2e?n ? ? ? = 6? n into (6.7) 2

?2 8

? e?

?2 2

2

n

holds with probability at least 1 ? 3e??n? ? 2e? 8 as long as n ? 288 and (6.10) into (6.1) we conclude that for ? = 7/400 u?C?Bd

(6.9)

. Plugging (6.8) and (6.9) with t = 6? , ? = 1, and

1 n ? ? ? ? ??xi , w ?? ?xi , u?? ? ?w ¿2 , n i=1 2

?w1 ? w? ¿2 ? 2 ? sup uT (

and

) + ?),

? (2? (C ? Sd?1) + ?) 1 n ? ? ?w? ¿2 , ? ??xi , w ?? ?xi , u? ?? ? 1 + ? n i=1 n

?

?2 8

(6.10) ? 2 (C?Sd?1) . ?2

Thus pluggin (6.6)

7 2 n ? ? ? ?w? ¿2 , ? ReLU(?xi , w ?)xi ? w) ? 2? ?w ¿2 ? n i=1 200

holds with probability at least 1 ? 8e??n as long as n ? c? 2 (C ? Sd?1) for a fixed numerical constant c. To introduce our general convergence analysis we begin by defining 7 . 200 To prove Theorem 3.1 we use [21][Page 25, inequality

(7.34)] which shows that if we apply the projected gradient descent update w? +1 = PK (w? ? ?L(w?)), the error h? = w? ? w? obeys E() = {w ? Rd ? R(w) ? R(w?), ?w ? w? ¿2 ? ?w? ¿2 } with =

?h? +1 ¿2 = ?w? +1 ? w? ¿2 ? 2 ? sup u? (h? ? ?L(w?)) .

(6.11)

u?C?Bn

To complete the convergence analysis it is then sufficient to prove 1 1 ?h? ¿2 = ?w? ? w? ¿2 . (6.12) 4 4 We will instead prove that the following stronger result holds for all u ? C ? B n and w ? E() sup u? (h? ? ?L(w?)) ?

u?C?Bn

1 ?w ? w? ¿2 . (6.13) 4 The equation (6.13) above implies (6.12) which when combined with (6.11) proves the convergence result of the Theorem (specifically equation (3.5)). The rest of this section is dedicated to proving i ,w?? (6.13). To this aim note that ReLU(?xi , w?) = ?xi ,w?+??x . Thus (see the extended version of this 2 paper [20] for more detailed derivation of the identity below) u? (w ? w? ? ?L(w)) ?

??L(w), u? =

1 n 1 n ? ? ? ??xi , w ? w ??xi , u? + ? sgn(?xi , w ?)?xi , w ? w ??xi , u? n i=1 n i=1 +

1 n ? ? ? (sgn(?xi , w?) ? sgn(?xi , w ?)) ?xi , w ? w ??xi , u? n i=1

+

1 n ? ? ? ? (1 ? sgn(?xi , w ?)) (sgn(?xi , w ?) ? sgn(?xi , w?)) ??xi , w ?? ?xi , u? 2n i=1 8

Now defining h = w ? w? we conclude that ?u, w ? w? ? ?L(w)? = ?u, h ? ?L(w)? is equal to 1 1 n ?u, h ? ?L(w)? =uT (I ? XX T) h ? ? sgn(?xi , w? ?)?xi , h??xi , u?, n n i=1 +

?h, w? ? 1 n ? ? ? (1 ? sgn(?xi , w?)sgn(?xi , w ?)) sgn(?xi , w ?)?xi , h??xi , u?, 2 ? n ?w ¿ i=1 2

sgn(?xi , w?) n ? ? + ? (1 ? sgn(?xi , w ?)) (1 ? sgn(?xi , w?)sgn(?xi , w ?)) 2n i=1 ??xi , w? ?? ?xi , u?. 2

Now define h? = h ? (hT w?)/(?w? ¿2)w? . Using this we can rewrite the previous expression in the form (see the proof in the extended version of this paper [20] for more detailed derivation) 1 1 n ?u, w ? w? ? ?L(w)? =uT (I ? XX T) h ? ? sgn(?xi , w? ?)?xi , h??xi , u?, n n i=1 +

1 n ? ? ? (1 ? sgn(?xi , w?)sgn(?xi , w ?)) sgn(?xi , w ?)?xi , h? ??xi , u?, n i=1

+

1 n sgn(?xi , w?) ?h, w? ? (1 ? sgn(?xi , w? ?)) +] ?[2 n i=1 2 ?w? ¿2

(1 ? sgn(?xi , w?)sgn(?xi , w? ?)) ??xi , w? ?? ?xi , u? (6.14) We now proceed by stating bounds on each of the four terms in (6.14). The detailed derivation of these bounds appear in the the extended version of this paper [20]. Lemma 6.5 Assume the setup of Theorem 3.1. Then as long as n ? cn0 , we have 1 u? (I ? X ? X) h ? ? ?h¿2 , n 1 n ? ? sgn(?xi , w? ?)?xi , h??xi , u? ? ? ?h¿2 , n i=1 ? ? 1 n ? ? ? (1 ? sgn(?xi , w?)sgn(?xi , w ?)) sgn(?xi , w ?)?xi , h? ??xi , u? ?2 1 + ? ? + n i=1 ?

(6.15) (6.16) ?

21 ? ?h¿2 , 20 ? (6.17)

1 n sgn(?xi , w?) ?h, w? ? (1 ? sgn(?xi , w? ?)) +] ?[2 n i=1 2 ?w? ¿2

? ? 21 ? 4 1+? ? (1 ? sgn(?xi , w?)sgn(?xi , w ?)) ??xi , w ?? ?xi , u? ? ?+ ?h¿2 , 2 (1 ?) ? 20 ? (6.18) ?

?

holds for all u ? C ? Sd?1 and w ? E() with probability at least 1 ? 9e??n . Combining (6.15), (6.16), (6.17), and (6.18) we conclude that ? ? ? ? 2 21 ?? ? ?u, w ? w ? ?L(w)? ? 2 ? + 1 + ? (1 +) ?+ ?w ? w? ¿2 , 2 (1 ?) ? 20 ?? ? 2

holds for all u ? C ? Sd?1 and w ? E() with probability at least 1 ? 16e??? n ? (n + 10)e??n . Using this inequality with ? = 10?4 and = 7/200 we conclude that ?u, w ? w? ? ?L(w)? ? 14 ?w ? w? ¿2 , holds for all u ? C ? Sd?1 and w ? E() with high probability.

Acknowledgements This work was done in part while the author was visiting the Simon?s Institute for the Theory of Computing. M.S. would like to thank Adam Klivans and Matus Telgarsky for discussions related to [6] and the Isotron algorithm. 9

2 References

[1] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp. Living on the edge: Phase transitions in convex programs with random data. Information and Inference, 2014. [2] A. Brutzkus and A. Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. International Conference on Machine Learning (ICML), 2017. [3] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. Foundations of Computational Mathematics, 12(6):805?849, 2012. [4] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pages 160?167. ACM, 2008. [5] R. Ganti, N. Rao, R. M. Willett, and R. Nowak. Learning single index models in high dimensions. arXiv preprint arXiv:1506.08910, 2015. [6] S. Goel, V. Kanade, A. Klivans, and J. Thaler. Reliably learning the ReLU in polynomial time. arXiv preprint arXiv:1611.10258, 2016. [7] Y. Gordon. On Milman?s inequality and random subspaces which escape through a mesh in Rn . Springer, 1988. [8] B. D. Haeffele and R. Vidal. Global optimality in tensor factorization, deep learning, and beyond. arXiv preprint arXiv:1506.07540, 2015. [9] J. L. Horowitz and W. Hardle. Direct semiparametric estimation of single-index models with discrete covariates. Journal of the American Statistical Association, 91(436):1632?1640, 1996. [10] H. Ichimura. Semiparametric least squares (SLS) and weighted SLS estimation of single-index models. Journal of Econometrics, 58(1-2):71?120, 1993. [11] S. M. Kakade, V. Kanade, O. Shamir, and A. Kalai. Efficient learning of generalized linear and single index models with isotonic regression. In Advances in Neural Information Processing Systems, pages 927?935, 2011. [12] A. T. Kalai and R. Sastry. The isotron algorithm: High-dimensional isotonic

regression. In COLT, 2009. [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097?1105, 2012. [14] Y. Li and Y. Yuan. Convergence analysis of two-layer neural networks with ReLU activation. arXiv preprint arXiv:1705.09886, 2017. [15] A. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. IEEE Transactions on Audio, Speech, and Language Processing, 20(1):14?22, 2012. [16] Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. arXiv preprint arXiv:1704.08045, 2017. [17] S. Oymak, B. Recht, and M. Soltanolkotabi. Sharp time?data tradeoffs for linear inverse problems. arXiv preprint arXiv:1507.04793, 2015. [18] S. Oymak and M. Soltanolkotabi. Fast and reliable parameter estimation from nonlinear observations. arXiv preprint arXiv:1610.07108, 2016. [19] T. Poston, C-N. Lee, Y. Choie, and Y. Kwon. Local minima and back propagation. In Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on, volume 2, pages 173?176. IEEE, 1991. [20] M. Soltanolkotabi. Learning ReLUs via gradient descent. arXiv preprint arXiv:1705.04591, 2017. 10

[21] M. Soltanolkotabi. Structured signal recovery from quadratic measurements: Breaking sample complexity barriers via nonconvex optimization. arXiv preprint arXiv:1702.06175, 2017. [22] M. Soltanolkotabi, A. Javanmard, and J. D. Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. 07 2017. [23] Y. Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. International Conference on Machine Learning (ICML), 2017. [24] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. arXiv preprint arXiv:1011.3027, 2010. [25] K. Zhong, Z. Song, P. Jain, P. L. Bartlett, and I. S. Dhillon. Recovery guarantees for one-hiddenlayer neural networks. arXiv preprint arXiv:1706.03175, 2017.

11