Convergence of Electron-Proton Dynamics in Deep Learning

Anonymous Authors[1]

Abstract

We study the efficacy of learning neural networks with neural networks by the (stochastic) gradient descent method. While gradient descent enjoys empirical success in a variety of applications, there is a lack of theoretical guarantees that explains the practical utility of deep learning. We focus on two-layer neural networks with a linear activation on the output node. We show that under some mild assumptions and certain classes of activation functions, gradient descent does learn the parameters of the neural network and converges to the global minima. Using a node-wise gradient descent algorithm, we show that learning can be done in finite, sometimes $poly(d, 1/\epsilon)$, time and sample complexity.

1. Introduction

1.1. Background

Deep learning has been widely adopted to solve a variety of practical problems in artificial intelligence. Backpropagation, perhaps the most well-known algorithm in deep learning, applies stochastic gradient descent (SGD) to the squared loss to learn suitable parameters of a neural network that approximates some target function $f : \mathbb{R}^n \to \mathbb{R}$. However, are these discovered parameters provably good or even optimal?

The main difficulty in analysis is the non-convexity of the loss objectives that deep learning present. Recent work has shown that SGD will efficiently converge to a local minimizer and escape saddle points, under modest assumptions (Ge et al., 2015). Therefore, it suffices to analyze the local minima of the loss landscape and show that no spurious local minima exist. This direction has led to positive results in matrix sensing (Park et al., 2016), matrix completion (Ge et al., 2016), dictionary learning (Sun et al., 2015), phase retrieval (Sun et al., 2016), and orthogonal tensor decomposition (Ge et al., 2015).

A non-convex analysis of the deep learning has been largely elusive and more discouragingly, there has been hardness results for even simple networks. A neural network with one hidden unit and sigmoidal activation can admit exponentially many local minima (Auer et al., 1996). Backprogration has been proven to fail in a simple network due to the abundance of bad local minima (Brady et al., 1989). Training a 3-node neural network with one hidden layer is NP-complete (Blum & Rivest, 1988). But, these and many similar worst-case hardness results are based on worst case training data assumptions. However, by using a result in (Klivans & Sherstov, 2006) that learning a neural network with threshold activation functions is equivalent to learning intersection of half-spaces, several authors showed that under certain cryptographic assumptions, depth-two neural networks are not efficiently learnable with smooth activation functions (Livni et al., 2014) (Zhang et al., 2015b)(Zhang et al., 2015a).

Due to the difficulty of analysis, many have turned to improper learning and the study of non-gradient methods to train neural networks. Janzamin et. al use tensor decomposition methods to learn the shallow neural network weights, provided access to the score function of the training data distribution (Janzamin et al., 2015). Eigenvector and tensor methods are also used to train shallow neural networks with quadratic activation functions in (Livni et al., 2014). Combinatorial methods that exploit layerwise correlations in sparse networks have also been analyzed provably in (Arora et al., 2013). Kernel methods, ridge regression, and even boosting were explored for regularized neural networks with smooth activation functions in (Shalev-Shwartz et al., 2011)(Zhang et al., 2015b)(Zhang et al., 2015a). Non-smooth activation functions, such as the ReLU, can be approximated by polynomials and are also amenable to kernel methods(Goel et al., 2016). These methods require assumptions that lessen the relevance of these algorithms to practice and more importantly, come short of explaining the widespread success of simple SGD.

Therefore, we will only focus on gradient-based methods in the proper learning framework and hope to find a theoretical justification of the good convergence properties. If the activation functions are linear or if certain independence assumptions are made, Kawaguchi shows that the only local minima are the global minima (Kawaguchi, 2016). Under the spin-glass and other physical mod-

els, some have shown that the loss landscape admit well-behaving local minima that occur usually when the overall error is small (Choromanska et al., 2014),(Dauphin et al., 2014). When only training error is considered, some have shown that a global minima can be achieved if the neural network contains sufficiently many hidden nodes (Soudry & Carmon, 2016). Our research is largely inspired by (Andoni et al., 2014), in which the authors show that when the target functions are polynomials, gradient descent on neural networks with one hidden layer is shown to converge to low error, given a large number of hidden nodes. And when complex perturbations are allowed, there is no robust local minima.

1.2. Our Contribution

In this work, we ask the question: Can we use neural networks to learn neural networks with gradient descent methods? That is, if the function to be learnt is a neural network, and we try to learn it with a network of the same shape and randomly initialized edge weights, then will the gradient descent converge to the right function? Our experimental simulations show that for different widths and heights functions represented by neural networks with random edge weights can be learnt by stochastic gradient descent (see section 5).

Next, we provide a theoretical justification with simplifying assumptions: specifically we will focus on learning depth two networks with a linear activation on the output node. If the neural network takes inputs $x \in \mathbb{R}^d$ (say from some distribution $\mathcal{D}$) then the output $f(x) = \sum_{i=1}^{k} b_i \sigma(x, w_i)$ is a sum over $k = poly(d)$ hidden units scaled by output weights $b_i \in \mathbb{R}$ where $\sigma(x, w) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the activation function that takes in a hidden weight vector $w \in \mathbb{R}^d$ and the input vector $x \in \mathbb{R}^d$. given by the form $f(x) = \sum_{i=1}^{k} b_i \sigma(x, w_i)$. Under some assumptions, we will show that gradient descent can learn the underlying weight parameters $b, w_i's$ of the target network for certain activation functions. The algorithm will try to learn a guess $\widehat{f}(x_j) = \sum_{i=1}^{k} a_i \sigma(x_j, \theta_i)$ for f and then running gradient descent over the parameters a_i, θ_i will move them to b_i, w_i.

1.2.1. Electron-Proton Dynamics

Our main observation is that the gradient descent dynamics of learning such to layer networks is equivalent to the dynamics of a set of proton-electron charges under a certain electrical attraction force function. Assume for now that the coefficients b_i and a_i are 1. Thus we are only gradienting over θ_i to minimize the expected square loss of $f - \widehat{f}$.

Theorem 1.1. *(informal statement of Theorem 2.4) The gradient descent process for learning f is equivalent to the motion of k electrons in the presence of k fixed protons where the force between any pair of charges is given by a potential function that depends on σ.*

Table 1: Potentials and Convergence Results Summary

Name of Activation	Potential ($\Phi(\theta, w)$)	GD Convergence?
Sign	$1 - \frac{2}{\pi}\cos^{-1}(\theta^T w)$	Yes
Polynomial	$(\theta^T w)^m$	Yes, $poly(d, \frac{1}{\epsilon})$
Exponential	$e^{\theta^T w - 1}$	Yes, $e^{poly(d)}$
λ-harmonic	Complicated	Yes, $poly(d, \frac{1}{\epsilon})$
Gaussian	$e^{-\|\theta - w\|_2^2/2}$	Yes, $e^{O(d)}$
Bessel	$e^{-\|\theta - w\|_1}$	Yes for $d = 1$

The charges reside in R^d. The protons are fixed at locations $w_1, .., w_k$. The electrons are at positions $\theta_1, .., \theta_k$ and can move: the total force on each charge is the sum of the pairwise forces, in each infinitesimal time step dt an electron moves by a distance proportional to the net force acting on it. The force between a pair of charges is determined by the gradient of the potential function $\Phi(\theta, w) = E_{X \sim \mathcal{D}}[\sigma(X, \theta)\sigma(X, w)]$.

1.2.2. Convergence Analysis

Note that for the standard electric potential function given by $\Phi = 1/r$ where r is the distance between the charges, it is known that the electrons must converge with the protons, by Earnshaw's Theorem. However, there is no activation function σ corresponding to this Φ.

We derive some sufficient conditions that characterize potentials that arise from activation functions σ. The different σ that we study, their corresponding potentials, and their convergence results are summarized in Table 1. Specifically, for the sign activation, we show that if we consider the loss function with respect to a single variable θ_i, then the only local minimas are at the true weights w_j, with high probability. For the polynomial activation, we derive a similar result under the assumption that our weights w_j are orthonormal and show convergence of a SGD algorithm to learn these weights in $poly(d, 1/\epsilon)$ time. For the exponential and Gaussian activation, we show that under the assumption that the output weights $b_i = 1$, the only local minimas are within a small neighborhood of the true weights w_j and show convergence of a SGD algorithm to learn these weights in exponential time. Finally and most surprisingly, we are able to construct a complicated activation function, which we call λ-harmonic, for which convergence of the charges can be proven with no additional assumptions, and we can learn the function f in $poly(d, 1/\epsilon)$ iterations of a SGD algorithm.

Our main tool for analysis is to derive second-order information about our dynamics by using the Laplacian of the Hessian (or a submatrix of the Hessian) of our loss function. Together with some generalization error bounds and discrete optimization results, we can finally translate these convergence results into finite time convergence rates. We remark that our algorithms learn and return a neural network with the same architecture and number of hidden nodes as the target network. This is a big contrast to the improper learning setting of many proposed algorithms.

We acknowledge that there is still a large gap between our developed theory and practice. However, our work can offer theoretical explanations and guidelines for the design of better activation functions or gradient-based training algorithms. For example, better accuracy and training speed were reported when using the newly discovered exponential linear unit (ELU) activation function in (Clevert et al., 2015) (Shah et al., 2016). We hope for more theory-backed answers to these and many other questions in deep learning.

In section 2, we introduce our framework and assumptions, and derive and derive the equivalence between gradient descent and electron-proton dynamics under a suitable potential. In section 3, we give an overview of our main results and techniques for showing convergence of gradient descent to learn the hidden parameters for depth-2 neural networks with specified activation functions, including some common activation functions, such as the sign and the polynomial. These convergence results are proven under ideal assumptions to simply illustrate our ideas. In section 4, we address the steps necessary to prove finite convergence guarantees using a node-wise gradient descent algorithm, dealing with errors introduced from approximation and discretization. In section 5, experimental results confirm that depth-2 neural networks can be learned by gradient descent with common activation functions but seem to discredit that claim for higher depth networks.

2. Deep Learning, Potentials, and Electron-Proton Dynamics

2.1. Preliminaries

The target concept class $\mathcal{C}_{\sigma,k}$ of our learning procedure is the output of a two-layer neural network with linear output activation and has the form $f(x) = \sum_{i=1}^{k} b_i \sigma(x, w_i)$. Let $\boldsymbol{a} = (a_1, ..., a_k)$ and $\boldsymbol{\theta} = (\theta_1, ..., \theta_k)$; similarly for $\boldsymbol{b}, \boldsymbol{w}$. For some distribution $\mathcal{D}$, we can given n pairs of training data $(x_i, f(x_i))$, where x_i are drawn i.i.d. from $\mathcal{D}$. Our hypothesis class is the same as the concept class and our loss function is the squared loss.

$$\widehat{L}(\boldsymbol{a}, \boldsymbol{\theta}) = \frac{1}{n} \sum_{j=1}^{n} \left(\sum_{i=1}^{k} b_i \sigma(x_j, w_i) - \sum_{i=1}^{k} a_i \sigma(x_j, \theta_i) \right)^2 \tag{1}$$

We would rather work with the true loss directly: $L(\boldsymbol{a}, \boldsymbol{\theta}) = \mathbb{E}[\widehat{L}]$. To further simplify L, let us reparametrize a with $-a$ and expand.

$$L(\boldsymbol{a}, \boldsymbol{\theta}) = \mathop{\mathbb{E}}_{X \sim D} \left[\left(\sum_{i=1}^{k} a_i \sigma(X, \theta_i) + \sum_{i=1}^{k} b_i \sigma(X, w_i) \right)^2 \right]$$

$$= \sum_{i=1}^{k} \sum_{j=1}^{k} a_i a_j \Phi(\theta_i, \theta_j) + 2 a_i b_j \Phi(\theta_i, w_j) + b_i b_j \Phi(w_i, w_j) \tag{2}$$

Where $\Phi(\theta, w) = \mathbb{E}_{X \sim D}[\sigma(X, \theta)\sigma(X, w)]$ is the **potential function** *corresponding to the activation function* σ. Although our loss is jointly non-convex but is quadratic in $\boldsymbol{a}$, so we have convergence guarantees to $\boldsymbol{a^*(\theta)}$, the optimal set of output weights of a given $\boldsymbol{\theta}$.

We will optimize over the familiar $\mathcal{M} = \mathbb{R}^d$ but often over $\mathcal{M} = S^{d-1}$. Let $\Pi_{\mathcal{M}}$ be the projection operator on $\mathcal{M}$. On $\mathbb{R}^d$, the gradient and Hessian $\nabla_{\mathbb{R}^d} f, \nabla^2_{\mathbb{R}^d} f$ is defined as standard and the Laplacian is simply $\Delta_{\mathbb{R}^d} f = \mathrm{Tr}(\nabla^2_{\mathbb{R}^d} f)$. The simplest way to define these terms on S^{d-1} is $\nabla_{S^{d-1}} f(x) = \nabla_{\mathbb{R}^d} f(x/\|x\|)$, where $\|\cdot\|$ denotes the l_2 norm and $x \in S^{d-1}$. The Hessian and Laplacian are analogously defined and the subscripts are usually dropped where clear from context. If f is multivariate with variable x_i, then let f_{x_i} be a restriction of f onto the variable x_i with all other variables fixed. Let $\nabla_{x_i} f, \Delta_{x_i} f$ to be the gradient and Laplacian, respectively, of f_{x_i} with respect to x_i. Lastly, we say x is a critical point of f if ∇f does not exist or $\nabla f = 0$.

A gradient-based method is an algorithm that calls Algorithm 1 in all the optimization procedures to minimize the empirical loss. Given $\mathcal{D}$, the training data, and the activation function σ, we attempt to show that with high probability over some random choices of b_i, w_i, there exists a gradient-based algorithm that optimizes $\widehat{L}$ on $\mathcal{M}$ and finds $\widehat{f} \in \mathcal{C}_{\sigma,k}$ with high probability such that $L(\boldsymbol{a}, \boldsymbol{\theta}) < \epsilon$, in time $poly(d, 1/\epsilon)$ or some other meaningful bound.

2.2. Activation-Potential Correspondence

We restrict our attention to potential (and activation) functions with some natural symmetry, so they are either *translationally or rotationally invariant*. Specifically, we may write $\Phi = h(\theta - w)$ for some function $h(x)$ in the first case, with $\theta, w \in \mathbb{R}^d$. In the second case,

$\Phi = h(\theta^T w)$ and we enforce $\theta, w \in S^{d-1}$. Such potentials are called **natural**. Our results focus on rotationally invariant potentials, as they correspond to classical neural networks.

Remark: Natural potentials satisfy $\Phi(\theta, \theta)$ *is a positive constant and we will also normalize* $\Phi(\theta, \theta) = 1$ *for the rest of the paper.*

We make a distributional assumption that our input distribution $\mathcal{D} = \mathcal{N}(0, \mathbf{I_{d \times d}})$ is fixed as the standard Gaussian in $\mathbb{R}^d$. This assumption is not critical and a simpler distribution might lead to better statistical bounds. However, if we allow arbitrary distributions, then hardness results in PAC-learning halfspaces would apply (Klivans & Sherstov, 2006).

We call a potential function **realizable** if it corresponds to some activation σ. We briefly state some results about characterizations of realizable potentials for translationally and rotationally invariant potentials. Full proofs and calculations of activation-potential correspondences, such as those claimed in Table 1, can be found in the supplementary material.

Theorem 2.1. *Let* $\mathcal{M} = \mathbb{R}^d$ *and* $\Phi(\theta, w) = f(\theta - w)$. *Then,* Φ *is realizable if* $\mathfrak{F}(f)(\omega) \geq 0$ *and* $\mathfrak{F}^{-1}(\sqrt{\mathfrak{F}(f)})$ *is bounded almost everywhere, where* $\mathfrak{F}$ *is the standard fourier transform in* $\mathbb{R}^d$

Theorem 2.2. *Let* $\mathcal{M} = S^{d-1}$ *and* $\Phi(\theta, w) = f(\theta^T w)$. *Then,* Φ *is realizable if* f *has non-negative Taylor coefficients,* $c_i \geq 0$ *, and*

$$h(x) = \sum_{i=1}^{\infty} \sqrt{c_i} h_i(x)$$

converges almost everywhere, where $h_i(x)$ *is the i-th Hermite polynomial.*

2.3. Electron-Proton Dynamics

By interpreting the pairwise potentials as electrostatic attraction potentials, we notice that our dynamics is similar to electron-proton type dynamics under some potential Φ, where w_i are fixed point charges in $\mathbb{R}^d$ and θ_i are moving point charges in $\mathbb{R}^d$ that are trying to find w_i. We note that standard electron-proton dynamics interprets the force between particles as an acceleration vector; in our case, it is more accurately interpreted as a velocity vector.

Algorithm 1 $x = GradientDescent(L, x_0, T, \alpha)$

Input: $L : \mathcal{M} \to \mathbb{R}$; $x_0 \in \mathcal{M}$; $T \in \mathbb{N}$; $\alpha \in \mathbb{R}$
Initialize $x = x_0$
for $i = 1$ **to** T **do**
  $x = x - \alpha \nabla L(x)$
  $x = \Pi_{\mathcal{M}} x$
end for

Definition 2.3. *Given a potential* $\Phi : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ *and particle locations* $\theta_1, ..., \theta_k \in \mathcal{M}$ *along with their respective charges* $a_1, ..., a_k \in \mathbb{R}$. *We define* **Electron-Proton Dynamics** *under* Φ *to be the solution* $(\theta_1(t), ..., \theta_k(t))$ *to the following system of differential equations: For each pair* (θ_i, θ_j)*, there is a force from* θ_j *exerted on* θ_i *that is given by* $\mathbf{F}_i(\theta_j) = a_i a_j \nabla_{\theta_i} \Phi(\theta_i, \theta_j)$ *and*

$$-\frac{d\theta_i}{dt} = \sum_{j \neq i} \mathbf{F}_i(\theta_j)$$

for $i \in [k]$*, with* $\theta_i(0) = \theta_i$. *If some subset* $S \subseteq [k]$ *of particles are fixed, then for* $i \in S$*, we have instead* $\theta_i(t) = \theta_i$ *for* $t \geq 0$.

Theorem 2.4. *Let* Φ *be a natural potential. Running continuous gradient descent on* (2) *with respect to* θ*, initialized at* $(\theta_1, ..., \theta_k)$ *produces the same dynamics as Electron-Proton Dynamics under* Φ *with fixed particles at* $w_1, ..., w_k$ *with respective charges* $b_1, .., b_k$ *and moving particles at* $\theta_1, ..., \theta_k$ *with respective charges* $a_1, ..., a_k$.

Proof. The initial values are the same. Notice that continuous gradient descent on $\frac{1}{2} L(\boldsymbol{a}, \boldsymbol{\theta})$ with respect to θ produces dynamics given by $\frac{d\theta_i(t)}{dt} = -\nabla_{\theta_i} L(\boldsymbol{a}, \boldsymbol{\theta})$. Therefore,

$$\frac{d\theta_i(t)}{dt} = -\sum_{j \neq i} a_i a_j \nabla_{\theta_i} \Phi(\theta_i, \theta_j) - \sum_{j=1}^{k} a_i b_j \nabla_{\theta_i} \Phi(\theta_i, w_j)$$

And gradient descent does not move w_i. By definition, the dynamics corresponds to Electron-Proton Dynamics as claimed. □

3. Main Results

When running gradient descent on non-convex loss functions, we often can and do get stuck at a local minima. In this section, we use second-order information to deduce the non-existence of such local minima for potentials with unique properties. These properties often lead to non-smooth and un-realizable potentials. However, in later section, we can show that gradient-type algorithms can globally optimize approximations of these potentials. (summarize more later)

3.1. Earnshaw's Theorem

Earnshaw's theorem in electrodynamics shows that there is no stable local minima for electron-proton dynamics. This hinges on the property that the electric potential

$\Phi(\theta, w) = \|\theta - w\|^{2-d}$ is harmonic (with $d = 3$ in natural setting). First, we notice that our usual loss in (2) has constant terms that can be dropped to further simplify.

$$\overline{L}(a, \theta) = 2\sum_{i=1}^{k}\sum_{i<j} a_i a_j \Phi(\theta_i, \theta_j) + 2\sum_{i=1}^{k}\sum_{j=1}^{k} a_i b_j \Phi(\theta_i, w_j) \tag{3}$$

Definition 3.1. *$\Phi(\theta, w)$ is a* **harmonic** *potential on Ω if $\Delta_\theta \Phi(\theta, w) = 0$ for all $\theta \in \Omega$, except possibly at $\theta = w$.*

Definition 3.2. *Let $\Omega \subseteq \mathbb{R}^d$ and consider a function $f : \Omega \to \mathbb{R}$. A critical point $x^* \in \Omega$ is a* **local minimum** *if it is differentiable and $\lambda_{min}(\nabla^2 f(x^*)) \geq 0$ and it is a* **strict local minimum** *if also $\lambda_{min}(\nabla^2 f(x^*)) > 0$.*

Theorem 3.3. *(Arnold et al., 1985)[Earnshaw] Let $\mathcal{M} = \mathbb{R}^d$ and let Φ be harmonic and L be as in (3). Then, L admits no strict local minima.*

Proof. If $(\boldsymbol{a}, \boldsymbol{\theta})$ is a strict local minima, we must have

$$\nabla_{\theta_1} L = 0, \text{ and } \operatorname{Tr}(\nabla^2_{\theta_i} L) > 0$$

But since Φ is harmonic,

$$\operatorname{Tr}(\nabla^2_{\theta_i} L(q_1, ..., q_n)) = \Delta_{\theta_i} L$$

$$= 2\sum_{j\neq i} a_i a_j \Delta_{\theta_i} \Phi(\theta_i, \theta_j) + 2\sum_{j=1}^{k} a_i b_j \Delta_{\theta_i} \Phi(\theta_i, w_j) = 0$$

□

Intuitively, the trace of the Hessian being 0 implies every differentiable critical point has a direction of negative curvative (unless the Hessian is the zero matrix altogether, in which case some complex analysis still finds a direction of negative change (Arnold et al., 1985)). Therefore, at convergence, we should expect $\theta_i = w_{\pi(i)}$ for some permutation π, as long as θ_i are all distinct.

Such a result is desired but we run into two main problems. First, the singularity of the natural harmonic potentials at 0 disqualifies them from being a realizable potential. Furthermore, harmonic potentials are not robust to approximation and statistical error as the convergence guarantees hinge on $\Delta_\theta \Phi$ being exactly 0.

Second, the lack of strict local minima *does not* imply convergence to the global minima under gradient descent. Notice that harmonic potentials can admit local minima, as the Hessian matrix can be the zero matrix, and so gradient descent could converge to these local minima. However, if our loss function admits no local minima (other than the global minima), then we can guarantee that gradient descent with small stepsize converges to the global minima.

3.2. Alternative Notions of Harmonic Potentials

3.2.1. Strictly Subharmonic Potentials

The first alternative to harmonic potentials is the natural consideration of strictly subharmonic potentials, which have a positive Laplacian value almost everywhere. Subharmonic potentials are also difficult to realize; however their convergence properties are more robust than harmonic potentials. In section 4, we will develop convergence properties of approximations of strictly subharmonic potentials.

Definition 3.4. *$\Phi(\theta, w)$ is a* **strictly subharmonic** *potential on Ω if $\Delta_\theta \Phi(\theta, w) > 0$ for all $\theta \in \Omega$, except possibly at $\theta = w$.*

An example of such a potential is $\Phi(\theta, w) = \|\theta - w\|^{2-d-\epsilon}$ for any $\epsilon > 0$. In this case, the sign of the output weights a_i, b_i matter in determining the sign of the Laplacian of our loss function. Therefore, we need to make suitable assumptions in this framework.

Assumption 1. *All output weights $b_i = 1$ and therefore the output weights $a_i = -b_i = -1$ are fixed throughout the learning algorithm.*

Under Assumption 1, we are working with an even simpler loss function:

$$L(\theta) = 2\sum_{i=1}^{k}\sum_{i<j} \Phi(\theta_i, \theta_j) - 2\sum_{i=1}^{k}\sum_{j=1}^{k} \Phi(\theta_i, w_j) \tag{4}$$

Theorem 3.5. *Let Φ be a natural strictly subharmonic potential on $\mathcal{M}$. Let Assumption 1 hold and let L be as in (4). Then, L admits no local minima.*

Proof. First, let Φ be translationally invariant and $\mathcal{M} = \mathbb{R}^d$. Let $\boldsymbol{\theta}$ be a critical point. Assume, for sake of contradiction, that for all i, j, $\theta_i \neq w_j$.

The main technical detail is to remove interaction terms between pairwise θ_i by considering a correlated movement, where each θ_i are moved along the same direction v. In this case, notice that our objective, as a function of v, is simply

$$H(v) = L(\theta_1 + v, \theta_2 + v, ..., \theta_k + v)$$

$$= 2\sum_{i=1}^{k}\sum_{i<j} \Phi(\theta_i + v, \theta_j + v) - 2\sum_{i=1}^{k}\sum_{j=1}^{k} \Phi(\theta_i + v, w_j)$$

Note that the first term is constant as a function of v, by translation invariance. Therefore,

$$\nabla^2 H = -2\sum_{i=1}^{k}\sum_{j=1}^{k} \nabla^2 \Phi(\theta_i, w_j)$$

By the subharmonic condition, $tr(\nabla^2 H) = -2\sum_{i=1}^{k}\sum_{j=1}^{k}\Delta_{\theta_i}\Phi(\theta_i, w_j) < 0$. Therefore, we conclude that θ is not a local minima of H and L.

WLOG, let's say $\theta_1 = w_1$. If $\Phi(\theta, w)$ is not differentiable when $\theta = w$, then we are done. Otherwise, simply realize that we can write L with θ_1, w_1 terms cancelled out. We reduce to the case where we have $k-1$ variable charges and $k-1$ fixed charges, so by induction, we conclude that $\theta_i = w_{\pi(i)}$ for some permutation π. The above technique generalizes to Φ being rotationally invariant case by working in spherical coordinates and correlated translations are simply rotations, with full details in the supplementary material. □

3.2.2. λ-HARMONIC POTENTIALS

When we drop Assumption 1 and allow the output weights to be variable and possibly negative, then our techniques above will fail. So, how do we generalize harmonic potentials to give us convergence properties in this more general case? In order to relate our loss function with its Laplacian, we consider potentials that are non-negative eigenfunctions of the Laplacian operator. Since the zero eigenvalue case simply gives rise to harmonic potentials, we restrict our attention to positive eigenfunctions.

Definition 3.6. *A potential Φ is* **λ-harmonic** *on Ω if there exists $\lambda > 0$ such that for every $\theta \in \Omega$, $\Delta_\theta \Phi(\theta, w) = \lambda \Phi(\theta, w)$, except possibly at $\theta \neq w$.*

We note that there are realizable versions of these potentials; for example $\Phi(a, b) = e^{-\|a-b\|_1}$ in $\mathbb{R}^1$. In Section 4, we show that approximations of these potentials are robust and retain good convergence behavior.

Theorem 3.7. *Let Φ be λ-harmonic and L be as in 2. Then, L admits no local minima $(\boldsymbol{a}, \boldsymbol{\theta})$, except possibly when $\boldsymbol{a} = 0$.*

Proof. Let $(\boldsymbol{a}, \boldsymbol{\theta})$ be a critical point of L. Now, a_i^* being optimal implies

$$0 = \frac{\partial L}{\partial a_i} = 2a_i + 2\sum_{j\neq i} a_j \Phi(\theta_i, \theta_j) + 2\sum_{j=1}^{k} b_j \Phi(\theta_i, w_j)$$

Computing the Laplacian with respect to θ_i gives

$$\begin{aligned}\Delta_{\theta_i} L &= 2\sum_{j\neq i} a_i a_j \Delta_{\theta_i}\Phi(\theta_i, \theta_j) + 2\sum_{j=1}^{k} a_i b_j \Delta_{\theta_i}\Phi(\theta_i, w_j) \\ &= \lambda a_i \left(2\sum_{j\neq i} a_j \Phi(\theta_i, \theta_j) + 2\sum_{j=1}^{k} b_j \Phi(\theta_i, w_j)\right) \\ &= -2\lambda a_i^2\end{aligned}$$

If $a_i^2 \neq 0$, then we conclude that the Laplacian is strictly negative. Thus all local minima of L satisfy $\mathbf{a} = 0$. □

4. Runtime Bounds with Stochastic Gradients

The gradient descent convergence results in the previous sections lay the foundation for reasoning about the convergence of stochastic gradient descent (SGD) in standard backpropagation. To derive finite runtime bounds, we need to address three technical details: 1) realizable potentials are only approximately strict subharmonic or λ-harmonic, 2) the variance of the stochastic gradient and 3) SGD should escape local minima in a bounded (hopefully polynomial) amount of time. The second and third detail are analyzed with standard techniques in optimization and statistics: the former requires generalization bounds for neural networks and the latter requires a lower bound on the negative curvature, which leads to the follow definition.

Definition 4.1. *Let $\Omega \subseteq \mathbb{R}^d$. A point x^* is a* **ϵ-strict** *point of $f : \Omega \to \mathbb{R}$ if f is twice differentiable at x^* and $\lambda_{min}(\nabla^2 f(x^*)) < -\epsilon$*

Note that if a critical point is not a local minimum, then it is 0-strict. By reusing techniques in (Ge et al., 2015), we show that SGD can avoid all points that are ϵ-strict in $poly(1/\epsilon)$ iterations and will converge to a point in $\mathcal{M}_\epsilon$, where

$$\mathcal{M}_\epsilon = \left\{ x \in \mathcal{M} \middle| \|\nabla L(x)\| \leq \epsilon \text{ and } \lambda_{min}(\nabla^2 L(x)) \geq -\epsilon \right\}$$

Intuitively, this means that SGD will converge to points with small gradient and small negative curvature. The full proofs of all these claims are found in supplementary material.

Because of the errors, we cannot simply analyze the convergence of SGD on all θ_i simultaneously since as before, the pairwise interaction terms between the θ_i present complications. To derive a tighter control on $(\boldsymbol{a}, \boldsymbol{\theta})$, we run a greedy node-wise SGD algorithm to learn the hidden weights, i.e. we run a full SGD algorithm with respect to (a_i, θ_i) sequentially. The main idea is that after running SGD with respect to θ_1, θ_1 should be close to some w_j for some j. Then, we can carefully induct and show that θ_2 must be some other w_k for $k \neq j$ and so on. The pseudocode of the exact node-wise SGD algorithm as well as the full theorems of the convergence results, as mentioned in Table 1, is given in the supplementary material

In addition, we remark that node-wise SGD allows us to learn the value of k, the number of hidden nodes in the target function. This may suggest that node-wise SGD

Table 2: Test Error of Learning Neural Networks of Various Depth and Width

	WIDTH 5	WIDTH 10	WIDTH 20	WIDTH 40
DEPTH 3	0.0033	0.0264	0.1503	0.2362
DEPTH 5	0.0036	0.0579	0.2400	0.4397
DEPTH 9	0.0085	0.1662	0.4171	0.6071
DEPTH 17	0.0845	0.3862	0.4934	0.5777

serves as a good initialization algorithm for training neural networks. A similar claim was also made in (Wu & Magdon-Ismail, 2016).

Will add algorithms and theorem statements here once fully verified.

5. Experiments

For our experiments, our training data is given by $(x_i, f(x_i))$, where x_i are randomly chosen from a standard Gaussian in $\mathbb{R}^d$ and f is a randomly generated neural network with weights chosen from a standard Gaussian. We run gradient descent (Algorithm 1) on the empirical loss, with stepsize around $\alpha = 10^{-5}$, for $T = 10^6$ iterations. Our experiments show that for depth-2 neural networks, even with non-linear outputs, the training error diminishes quickly to a small constant. This seems to hold even when the width, the number of hidden nodes, is substantially increased (even up to 125 nodes), but depth is held constant; although as the number of nodes increases, the rate of decrease is slower. This substantiates our claim that depth-2 neural networks are learnable.

However, it seems that for depth greater than 2, gradient descent does not learn the neural network when width is high. When the width is a small constant, the increase in depth also impedes the learnability of the neural network and the training error does not get close enough to 0. It seems that for neural networks with greater depth, positive convergence results in practice are elusive. We note that we have been using training error as a measure of success, but it's possible that the true underlying parameters are not learned. If our loss function were strongly convex, small training error would imply a small norm in the parameter space. For the sign activation function, we can show a related result.

Theorem 5.1. *Let $\mathcal{M} = S^{d-1}$ and σ be the sign activation function and $b_2, ..., b_k = 0$. If the loss (2) at $(\boldsymbol{a}, \boldsymbol{\theta})$ is less than $O(1)$, then there must exist θ_i such that $w_1^T \theta_i > \Omega(1/\sqrt{k})$.*

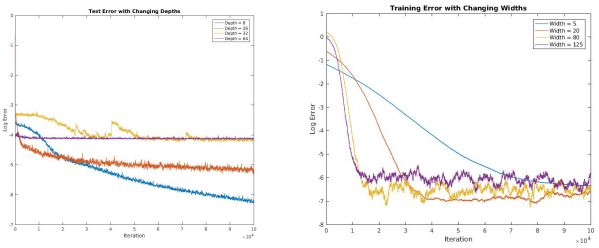

Figure 1: Left:Test Error of Networks of Varying Depth. Right: Test Error of Networks of Varying Width.

6. Conclusion

In this work, we view deep learning of neural networks in the context of electron-proton dynamics and analyzed the convergence of the underlying weight parameters of the neural network using arguments inspired from physics and non-convex optimization. To do so, we first established mathematical relationship between activation functions and their corresponding potentials. Next, we interpreted gradient descent as electrodynamics under a certain potential. Finally, we discovered classes of activation functions that give rise to positive convergence results, some of which relate to very commonplace activations, such as the sign and polynomial. For these classes of depth-2 neural networks, our results imply that they are provably learnable by deep learning. Our experiments seem to imply that higher depth neural networks are not learnable.

However, we believe that convergence results for depth-2 neural networks can be extended to even more activation functions, such as the sigmoid or the ReLU. Also, we believe these convergence results can be proven with minimal assumptions. We hope that our work is a step in the theoretical understanding of the performance of neural networks seen in practice.

References

Andoni, Alexandr, Panigrahy, Rina, Valiant, Gregory, and Zhang, Li. Learning polynomials with neural networks. 2014.

Arnold, Vladimir I, Kozlov, Valery V, and Neishtadt, Anatoly I. Mathematical aspects of classical and celestial mechanics. *Encyclopaedia Math. Sci*, 3:1–291, 1985.

Arora, Sanjeev, Bhaskara, Aditya, Ge, Rong, and Ma, Tengyu. Provable bounds for learning some deep representations. *CoRR*, abs/1310.6343, 2013. URL http://arxiv.org/abs/1310.6343.

Auer, Peter, Herbster, Mark, and Warmuth,

Manfred K. Exponentially many local minima for single neurons. pp. 316–322, 1996. URL http://papers.nips.cc/paper/1028-exponentially-many-local-minima-for-single-neurons.pdf.

Bartlett, Peter L and Mendelson, Shahar. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Blum, Avrim and Rivest, Ronald L. Training a 3-node neural network is np-complete. pp. 9–18, 1988.

Brady, Martin L, Raghavan, Raghu, and Slawny, Joseph. Back propagation fails to separate where perceptrons succeed. *IEEE Transactions on Circuits and Systems*, 36(5):665–674, 1989.

Choromanska, Anna, Henaff, Mikael, Mathieu, Michaël, Arous, Gérard Ben, and LeCun, Yann. The loss surface of multilayer networks. *CoRR*, abs/1412.0233, 2014. URL http://arxiv.org/abs/1412.0233.

Clevert, Djork-Arné, Unterthiner, Thomas, and Hochreiter, Sepp. Fast and accurate deep network learning by exponential linear units (elus). *CoRR*, abs/1511.07289, 2015. URL http://arxiv.org/abs/1511.07289.

Daniely, Amit, Frostig, Roy, and Singer, Yoram. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. *CoRR*, abs/1602.05897, 2016. URL http://arxiv.org/abs/1602.05897.

Dauphin, Yann, Pascanu, Razvan, Gülçehre, Çaglar, Cho, Kyunghyun, Ganguli, Surya, and Bengio, Yoshua. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *CoRR*, abs/1406.2572, 2014. URL http://arxiv.org/abs/1406.2572.

Ge, Rong, Huang, Furong, Jin, Chi, and Yuan, Yang. Escaping from saddle points - online stochastic gradient for tensor decomposition. *CoRR*, abs/1503.02101, 2015. URL http://arxiv.org/abs/1503.02101.

Ge, Rong, Lee, Jason D., and Ma, Tengyu. Matrix completion has no spurious local minimum. *CoRR*, abs/1605.07272, 2016. URL http://arxiv.org/abs/1605.07272.

Goel, Surbhi, Kanade, Varun, Klivans, Adam, and Thaler, Justin. Reliably learning the relu in polynomial time. *CoRR*, abs/1611.10258, 2016. URL http://arxiv.org/abs/1611.10258.

Janzamin, Majid, Sedghi, Hanie, and Anandkumar, Anima. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *CoRR*, abs/1506.08473, 2015. URL http://arxiv.org/abs/1506.08473.

Kakade, Sham M, Sridharan, Karthik, and Tewari, Ambuj. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pp. 793–800, 2009.

Kawaguchi, Kenji. Deep learning without poor local minima. *CoRR*, abs/1605.07110, 2016. URL http://arxiv.org/abs/1605.07110.

Klivans, Adam R and Sherstov, Alexander A. Cryptographic hardness for learning intersections of halfspaces. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 553–562. IEEE, 2006.

Lee, Jason D, Simchowitz, Max, Jordan, Michael I, and Recht, Benjamin. Gradient descent only converges to minimizers. 2016.

Livni, Roi, Shalev-Shwartz, Shai, and Shamir, Ohad. On the computational efficiency of training neural networks. *CoRR*, abs/1410.1141, 2014. URL http://arxiv.org/abs/1410.1141.

Nesterov, Yurii. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Panageas, Ioannis and Piliouras, Georgios. Gradient descent converges to minimizers: The case of non-isolated critical points. *CoRR*, abs/1605.00405, 2016. URL http://arxiv.org/abs/1605.00405.

Park, Dohyung, Kyrillidis, Anastasios, Caramanis, Constantine, and Sanghavi, Sujay. Non-square matrix sensing without spurious local minima via the burer-monteiro approach. *CoRR*, abs/1609.03240, 2016. URL http://arxiv.org/abs/1609.03240.

Shah, Anish, Kadam, Eashan, Shah, Hena, and Shinde, Sameer. Deep residual networks with exponential linear unit. *CoRR*, abs/1604.04112, 2016. URL http://arxiv.org/abs/1604.04112.

Shalev-Shwartz, Shai, Shamir, Ohad, and Sridharan, Karthik. Learning kernel-based halfspaces with the 0-1 loss. *SIAM Journal on Computing*, 40(6):1623–1646, 2011.

Soudry, Daniel and Carmon, Yair. No bad local minima: Data independent training error guarantees for multilayer neural networks. *CoRR*, abs/1605.08361, 2016. URL http://arxiv.org/abs/1605.08361.

Sun, Ju, Qu, Qing, and Wright, John. Complete dictionary recovery over the sphere. *CoRR*, abs/1504.06785, 2015. URL http://arxiv.org/abs/1504.06785.

Sun, Ju, Qu, Qing, and Wright, John. A geometric analysis of phase retrieval. *CoRR*, abs/1602.06664, 2016. URL http://arxiv.org/abs/1602.06664.

Wu, Ke and Magdon-Ismail, Malik. Node-by-node greedy deep learning for interpretable features. *CoRR*, abs/1602.06183, 2016. URL http://arxiv.org/abs/1602.06183.

Zhang, Yuchen, Lee, Jason D., and Jordan, Michael I. l1-regularized neural networks are improperly learnable in polynomial time. *CoRR*, abs/1510.03528, 2015a. URL http://arxiv.org/abs/1510.03528.

Zhang, Yuchen, Lee, Jason D., Wainwright, Martin J., and Jordan, Michael I. Learning halfspaces and neural networks with random initialization. *CoRR*, abs/1511.07948, 2015b. URL http://arxiv.org/abs/1511.07948.

A. Realizable Potentials

A.1. Activation-Potential Calculations

First define the *dual* of a function $f : \mathbb{R} \to \mathbb{R}$ is defined to be

$$\widehat{f}(\rho) = E_{X,Y\sim N(\rho)}[f(X)f(Y)],$$

where $N(\rho)$ is the bivariate normal distribution with X, Y unit variance and ρ covariance. This is as in (Daniely et al., 2016).

Lemma A.1. *Let $\mathcal{M} = S^{d-1}$ and σ be our activation function, then $\widehat{\sigma}$ is the corresponding potential function.*

Proof. If u, v have norm 1 and if X is a standard Gaussian in $\mathbb{R}^d$, then note that $X_1 = u^T X$ and $X_2 = v^T X$ are both standard Gaussian variables in $\mathbb{R}^1$ and the covariance is $E[X_1 X_2] = u^T v$.

Therefore, the dual function of the activation gives us the potential function.

$$\begin{aligned} E_X[\sigma(u^T X)\sigma(v^T X)] &= E_{X,Y\sim N(u^T v)}[\sigma(X)\sigma(Y)] \\ &= \widehat{\sigma}(u^T v). \end{aligned}$$

□

By Lemma A.1, the calculations of the activation-potential for the sign, ReLU, Hermite, exponential functions are given in (Daniely et al., 2016). For the Gaussian and Bessel activation functions, we can calculate directly. In both case, we notice that we may write the integral as a product of integrals in each dimension. Therefore, it suffices to check the following 1-dimensional identities.

$$\begin{aligned} &\int_{-\infty}^{\infty} \sqrt{2}e^{x^2/4}e^{-(x-\theta)^2}\sqrt{2}e^{x^2/4}e^{-(x-w)^2}\frac{1}{\sqrt{2\pi}}e^{-x^2/2}\,dx \\ &\quad = \sqrt{\frac{2}{\pi}}\int_{-\infty}^{\infty} e^{-(x-\theta)^2}e^{-(x-w)^2}\,dx = e^{-(\theta-w)^2/2} \end{aligned}$$

$$\begin{aligned} &\int_{-\infty}^{\infty} (\frac{2}{\pi})^{3/2}e^{x^2/2}K_0(|x-\theta|)K_0(|x-w|)\frac{1}{\sqrt{2\pi}}e^{-x^2/2}\,dx \\ &\quad = \int_{-\infty}^{\infty} \frac{2}{\pi^2}K_0(|x-\theta|)K_0(|x-w|)\,dx = e^{-|\theta-w|} \end{aligned}$$

The last equality follows by Fourier uniqueness and taking the Fourier transform of both sides, which are both equality $\sqrt{2/\pi}(\omega^2+1)^{-1}$.

A.2. Characterization Theorems

Theorem 2.1. *Let $\mathcal{M} = \mathbb{R}^d$ and $\Phi(\theta, w) = f(\theta - w)$. Then, Φ is realizable if $\mathfrak{F}(f)(\omega) \geq 0$ and $\mathfrak{F}^{-1}(\sqrt{\mathfrak{F}(f)})$ is bounded almost everywhere, where $\mathfrak{F}$ is the standard fourier transform in $\mathbb{R}^d$*

Proof. Let $h(x) = \mathfrak{F}^{-1}(\sqrt{\mathfrak{F}(f)})(x)$ and this is well-defined since the fourier transform was non-negative everywhere. Now, let $\sigma(x, w) = (2\pi)^{1/4} e^{x^2/4} h(x - w)$ and by assumption, it is bounded almost everywhere. Realizability follows by checking:

$$\begin{aligned} E_{X\sim N}[\sigma(X, w)\sigma(X, \theta)] &= \int_{\mathbb{R}^n} h(x - w)h(x - \theta)\, dx \\ &= \int_{\mathbb{R}^n} h(x)h(x - (\theta - w))\, dx \\ &= \mathfrak{F}^{-1}(\mathfrak{F}(h * h)(\theta - w)) \\ &= \mathfrak{F}^{-1}(\mathfrak{F}(h)^2(\theta - w)) \\ &= \mathfrak{F}^{-1}(\mathfrak{F}(f)(\theta - w)) \\ &= f(\theta - w) \end{aligned}$$

□

Theorem 2.2. *Let $\mathcal{M} = S^{d-1}$ and $\Phi(\theta, w) = f(\theta^T w)$. Then, Φ is realizable if f has non-negative Taylor coefficients, $c_i \geq 0$, and*

$$h(x) = \sum_{i=1}^{\infty} \sqrt{c_i} h_i(x)$$

converges almost everywhere, where $h_i(x)$ is the i-th Hermite polynomial.

Proof. By A.1 and due to the orthogonality of hermite polynomials, if $f = \sum_i a_i h_i$, where $h_i(x)$ is the i-th Hermite polynomial, then

$$\widehat{f}(\rho) = \sum_i a_i^2 \rho^i$$

Therefore, any function with non-negative taylor coefficients is a valid potential function, with the corresponding activation function determined by the sum of hermite polynomials, and the sum is bounded almost everywhere by assumption. □

B. Extra Stuff in section 3

Clean this up later

Theorem B.1. *Let $\mathcal{M} = S^{d-1}$ and Φ be strictly subharmonic. Let Assumption 1 hold and L be as in (4). Then, all critical points of L are not local minima, except at the global minima.*

Proof. Note that we can change to spherical coordinates (without the radius parameter) and let $\widetilde{\theta_1}, ..., \widetilde{\theta_k}$ be the standard spherical representation of $\theta_1, ..., \theta_k$.

We will consider a correlated translation in the spherical coordinate space, which are simply rotations on the sphere. Let v be a vector in $\mathbb{R}^{d-1}$ and our objective is simply

$$H(v) = L(\widetilde{\theta_1} + v, ..., \widetilde{\theta_k} + v)$$

Then, we apply the same proof since $\Phi(\widetilde{\theta_i} + v, \widetilde{\theta_j} + v,)$ is constant as a function of v by rotationally invariance. □

Theorem B.2. *(can be cut out actually...) Let Φ be strictly subharmonic on $\mathcal{M}$ and $\sup_{x\in\mathcal{M}} \|\nabla^2\Phi\| \leq B$ and let Assumption 1 hold. Let L be as in (4) and $x_0 \in \mathcal{M}$ randomly chosen.*

Running Algorithm 1 on L with stepsize $\alpha = \frac{1}{4k^2B}$ converges to the global minima almost surely as $T \to \infty$.

Proof. First note that Φ is twice differentiable and by our bounds on $\|\nabla^2\Phi\|$, we can naively bound $\sup_{x\in\Omega} \|\nabla^2 L\| \leq 4k^2B$. By Theorem C.6, we converge to a critical point $\theta = (\theta_1, ..., \theta_k)$ that is a local minima. Finally, Theorem 3.5 tells us we must have converged to the global minima. □

B.1. Realistic Potentials

The analysis of convergence through Laplacian information do not directly extend to analyzing potentials that correspond to more widely-used activation functions in the deep learning community, such as sign, ReLU, or polynomial. However, for these more realistic potentials, we can still make some progress under further assumptions.

Analyzing standard gradient descent on (2) is complicated by pairwise interaction terms between θ_i, θ_j; our techniques in the previous section do not generalize well. Therefore, in this section, we only consider the convergence guarantee of gradient descent on the first node, θ_1, to some w_j, while the other nodes are inactive (i.e. $a_2, ..., a_k = 0$). In essence, we are working with the following simplified loss function.

$$L(a_1, \theta_1) = a_1^2\Phi(\theta_1, \theta_1) + 2\sum_{j=1}^{k} a_1 b_j \Phi(\theta_1, w_j) \quad (5)$$

While a convergence result of (5) is weaker, it motivates the introduction of a node-wise gradient descent algorithm, where instead of applying gradient descent on all θ_i simultaneously, we run gradient descent on $\theta_1, ..., \theta_k$ sequentially. We will introduce and apply node-wise gradient descent to derive finite time bounds in Section 4.

Theorem B.3. *Let $\mathcal{M} = S^{d-1}$. Let $w_1, ..., w_k$ be orthonormal vectors in $\mathbb{R}^d$ and Φ is of the form $\Phi(\theta, w) = (\theta^T w)^l$ for some fixed integer $l \geq 3$. Let L be as in (5). Then, all critical points of L are not local minima, except at the global minima.*

Proof. WLOG, we can consider $w_1, ..., w_d$ to be the basis vectors $e_1, ..., e_d$. Note that this is a manifold optimization problem, so our optimality conditions are given by introducing a Lagrange multiplier λ, as in (Ge et al., 2015).

$$\frac{\partial L}{\partial a} = 2\sum_{i=1}^{d} ab_i(\theta_i)^l + 2a = 0$$

$$(\nabla_\theta L)_i = 2ab_i l(\theta_i)^{l-1} - 2\lambda\theta_i = 0$$

where λ is chosen that minimizes

$$\lambda = \arg\min_\lambda \sum_i (ab_i l(\theta_i)^{l-1} - \lambda\theta_i)^2 = \sum ab_i l(\theta_i)^l$$

Therefore, either $\theta_i = 0$ or $b_i(\theta_i)^{l-2} = \lambda/(al)$. Note also that the manifold Hessian is a diagonal matrix with diagonal entry:

$$(\nabla^2 L)_{ii} = 2ab_i l(l-1)(\theta_i)^{l-2} - 2\lambda$$

Assume that there exists $\theta_i, \theta_j \neq 0$, then we claim that θ is not a local minima. First, our optimality conditions imply $b_i(\theta_i)^{l-2} = b_j(\theta_j)^{l-2} = \lambda/(al)$. So,

$$(\nabla^2 L)_{ii} = (\nabla^2 L)_{jj} = 2ab_i l(l-1)(\theta_i)^{l-2} - 2\lambda$$

$$= 2(l-2)\lambda = -2(l-2)la^2$$

Now, there must exist a vector $v \in S^{d-1}$ such that $v_k = 0$ for $k \neq i, j$ and $v^T\theta = 0$, so v is in the tangent space at θ. Finally, $v^T(\nabla^2 L)v = -2(l-2)la^2 < 0$, implying our claim when $a \neq 0$. Note that $a = 0$ occurs with probability 0 since our objective function is non-increasing throughout the gradient descent algorithm and is almost surely initialized to be negative with a optimized upon initialization. □

Next, we consider the sign activation function. Under restrictions on the size of the input dimension or the number of hidden units, we can prove convergence results under the sign activation function.

Theorem B.4. *Let $\mathcal{M} = S^1$ and L be as in (5) and σ is the sign activation function. Then, almost surely over random choices of $b_1, ..., b_k$, all local minima of L are at $\pm w_j$.*

Proof. In S^1, notice that the pairwise potential function is $\Phi(\theta, w) = 1 - 2\cos^{-1}(\theta^T w)/\pi = 1 - 2\alpha/\pi$, where α is the angle between θ, w. So, let us parameterize in polar coordinates, calling our true parameters as $\widetilde{w_1}, ..., \widetilde{w_k} \in [0, 2\pi]$ and rewriting our loss as a function of $\widetilde{\theta} \in [0, 2\pi]$.

Since Φ is a linear function of the angle between θ, w_j, each w_j exerts a constant gradient on $\widetilde{\theta}$ towards $\widetilde{w_j}$, with discontinuities at $\widetilde{w_j}, \pi + \widetilde{w_j}$. Almost surely over $b_1, .., b_k$, the gradient is non-zero almost everywhere, except at the discontinuities, which are at $\widetilde{w_j}, \pi + \widetilde{w_j}$ for some j. □

Corollary B.5. *Let $\mathcal{M} = S^{d-1}$ and let there be k hidden nodes with $k \leq 2$. Let L be as in (5) and σ is the sign activation function. Then, almost surely over random choices of $b_1, ..., b_k$, all local minima of L are at $\pm w_j$.*

Proof. Let w_1, w_2 be the two hidden weights. As shown in Theorem B.4, there is a constant force gradient of w_1, w_2 on θ. The set of points where the sum of the two forces can be 0 lies on the great circle passing though w_1, w_2. Therefore, it reduces to a problem in S^1, whose convergence is shown in Theorem B.4. □

We conjecture that for the sign potential function, convergence results can be derived when running gradient descent on all θ_i simultaneously.

Conjecture B.6. *Let $\mathcal{M} = S^{d-1}$ and L be as in (2) and σ is the sign activation function. Then, almost surely over random choices of $b_1, ..., b_k$, all critical points $(\boldsymbol{a}, \boldsymbol{\theta})$ of L with $a \neq 0$ are not local minima, except at the global minima.*

C. Generalization Error and Iteration Bounds

The design and analysis of gradient descent has so far assumed that we can calculate expectations perfectly. In reality, these expectations are instead replaced with empirical means. The approximate calculation of our potential and all its derivatives with samples are justified by the generalization error bounds implied by Rademacher complexities.

Unfortunately, we cannot directly use the Gaussian distribution as it is unbounded. Therefore, we assume that our drawing distribution is the truncated Gaussian distribution in $\mathbb{R}^d$ such that our samples are always bounded in l_2 norm by B. We will show that applying this truncation will not affect the expectation very much.

Definition C.1. *Y follows a truncated Gaussian distribution at B in $\mathbb{R}^d$ if $Y = X|(\|X\| \leq B)$, where X is a standard Gaussian in $\mathbb{R}^d$ and $X\Big|S$ indicates the random variable X conditioned on event S.*

Lemma C.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function such that $|f(x)| \leq c\|x\|^p$. Let X be a standard Gaussian in $\mathbb{R}^d$ and let Y be a truncated Gaussian at B. Then, there exists $B = poly(d, p, \log(1/\epsilon))$ such that*

$$|\mathbb{E}[f(X)] - \mathbb{E}[f(Y)]| \leq \epsilon$$

Proof. By standard concentration bounds and analysis,

$$\begin{aligned}
&|E[f(X)] - E[f(X)|\|X\| \le B]| \\
&\quad \le (1 - \frac{1}{P(\|X\| \le B)})E[|f(X)|\mathbf{1}_{\|X\| \le B}]| \\
&\qquad + |E[f(X)\mathbf{1}_{\|X\|>B}]| \\
&\quad \le E[|f(X)|\mathbf{1}_{\|X\|>B}] + 2cB^p e^{-B^2/8d}
\end{aligned}$$

[S: Check the first term, prob is < 1, making the first term negative. Also, in the terms in the proof, you don't want conditioning, but rather an and with an indicator of the event.]

Taking $B = poly(d, p, \log(1/\epsilon))$ will make the second term $< \epsilon/2$, then the first term is also bounded by:

$$\begin{aligned}
E[|f(X)|\mathbf{1}_{\|X\|>B}] &\le (2\pi)^d \int_B^\infty cr^p e^{-r^2/2} r^{d-1}\, dr \\
&\le C(2\pi)^d B^{p+d} e^{-B^2/2} < \epsilon/2
\end{aligned}$$

□

Next, we can appeal to the following well-cited theorems and standard techniques. For a better understanding of the notation and theorems used, we refer the refer to (Bartlett & Mendelson, 2002).

Theorem C.3 ((Bartlett & Mendelson, 2002))**.** *Consider a function class $\mathcal{F}$ of functions $f : \mathcal{X} \to [0, 1]$. And let $x_1, ..., x_n \in \mathcal{X}$ be i.i.d. samples selected according to some distribution $\mathcal{D}$. Let the Rademacher complexity of $\mathcal{F}$ to be*

$$R_n(\mathcal{F}) = E_{x_i,\sigma_i}\left[\sup_{f\in\mathcal{F}}\left|\frac{2}{n}\sum_{i=1}^n f(x_i)\sigma_i\right|\right]$$

where σ_i are Rademacher variables. Then, for any n and $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$|E_X[f(X)] - \frac{1}{n}\sum_{i=1}^n f(x_i)| \le R_n(\mathcal{F}) + \sqrt{\frac{8\ln(2/\delta)}{n}}$$

for all $f \in \mathcal{F}$ simultaneously.

For all of our polynomial time bounds, whether we are calculating the potential or its derivatives, we are interested in bounding the Rademacher complexity of the class of functions of the form $\sigma_1(x^T\theta)\sigma_2(x^Tw)$, where σ_1, σ_2 are scalar functions.

Theorem C.4. *Let $\mathcal{X} = \{x \in \mathbb{R}^d, \|x\|_2 \le B\}$ and let $\sigma_1, \sigma_2 : \mathbb{R} \to [-C, C]$ be L-Lipschitz functions. Consider the following class of functions*

$$\mathcal{F}_{\sigma_1,\sigma_2} = \{x \to \sigma_1(x^T\theta)\sigma_2(x^Tw)|x \in \mathcal{X}, \theta, w \in S^{d-1}\}.$$

Then,

$$R_n(\mathcal{F}_{\sigma_1,\sigma_2}) \le 48CLB\sqrt{\frac{2}{n}}$$

Proof. First, let's define a simpler function class: $\mathcal{G} = \{x \to x^T\theta | x \in \mathcal{X}, \theta \in S^{d-1}\}$. By simple Rademacher bounds on the class of linear functions (Kakade et al., 2009),

$$R_n(\mathcal{G}) \le B\sqrt{\frac{2}{n}}$$

Since σ_1, σ_2 are L-lipschitz, we apply standard structural results in (Bartlett & Mendelson, 2002)

$$R_n(\sigma_i \circ \mathcal{G}) \le 2LB\sqrt{\frac{2}{n}}$$

Let $s(x) = x^2$, then s is $2C$-Lipschitz when $|x| \le C$, so

$$R_n(s\circ\sigma_i\circ\mathcal{G}) \le 8CLB\sqrt{\frac{2}{n}}, R_n(s\circ(\sigma_1\circ\mathcal{G}+\sigma_2\circ\mathcal{G})) \le 32CLB\sqrt{\frac{2}{n}}$$

Since $2\sigma_1(x^T\theta)\sigma_2(x^Tw) = (\sigma_1(x^T\theta) + \sigma_2(x^Tw))^2 - \sigma_1(x^T\theta)^2 - \sigma_2(x^Tw)^2$, we conclude that

$$R_n(\mathcal{F}_{\sigma_1,\sigma_2}) \le 48CLB\sqrt{\frac{2}{n}}$$

□

Theorem C.5. *Let $\mathcal{M} = S^{d-1}$ and L be as in 2 with potential function Φ corresponding to activation σ. Assume that $|\sigma(x)|, |\sigma'(x)|, |\sigma''(x)|, |\sigma'''(x)|$ are all $O(|x|^p)$ for some p.*

Then, in $poly(d^p, 1/\epsilon, \log(1/\zeta))$ samples, we can compute $\hat{L}$ such that with probability $1 - \zeta$, we have simultaneously $\|\widehat{L}(x) - L(x)\|, \|\nabla\widehat{L}(x) = L(x)\|, \|\nabla^2\widehat{L}(x) - \nabla^2 L(x)\| \le \epsilon$ for all $x \in \Omega$.

Proof. We first bound the generalization error of each Φ. Notice that our approximation to Φ is done by first drawing n i.i.d. samples $x_i \sim \mathcal{D}_B$, where $\mathcal{D}_B$ is the Gaussian in $\mathbb{R}^d$ truncated by the ball of radius $B = poly(d, p, \log(1/\epsilon))$. Then, we calculate the empirical average.

$$\widehat{\Phi}(\theta, w) = \frac{1}{n}\sum_{i=1}^n \sigma(x_i^T\theta)\sigma(x_i^Tw)$$

Since $|x_i^T\theta| \le B$, we conclude that σ, σ' is bounded by $poly(B^p)$. Therefore, by our error bound theorems and by simple union bounds over the $poly(k) = poly(d)$ sum of Φ, we conclude that with probability $1 - \zeta$, if we choose $n = poly(d^p, 1/\epsilon, \log(1/\zeta))$, we have

$$|E_{X\sim\mathcal{D}_B}[\sigma(X^Tw)\sigma(X^T\theta)] - \widehat{\Phi}(\theta^Tw)| \le \epsilon/2,$$

for all $\theta, w \in \mathcal{M}$. And combining with Lemma C.2, we conclude that $|\widehat{\Phi}(\theta, w) - \Phi(\theta, w)| \le \epsilon$. We proceed with the same proof for the first and second derivatives and use a union bound to derive our claim. □

C.1. Infinite Iteration Bounds

Theorem C.6. *(Lee et al., 2016; Panageas & Piliouras, 2016) Let $f : \Omega \to \mathbb{R}$ be a twice differentiable function such that $\sup_{x\in\Omega} \|\nabla^2 f\| \leq L$. Let $\mathcal{S} \subseteq \Omega$ be the set of critical points of f that are not local minima. Also, if $g(x) = x - \frac{1}{2L}\nabla f(x)$, then $g(\Omega) \subseteq \Omega$.*

Then, running Algorithm 1 with gradient input ∇f and stepsize $\alpha = 1/(2L)$, as the iteration $T \to \infty$, will converge to a point x_∞ outside of S almost surely over randomly chosen initial points x_0.

Corollary C.7. *Assume all the assumptions of Theorem C.6 and let f admit a global minima in $\overline{\Omega}$. Assume all critical points of f in Ω are not local minima, except at the global minima. Then, running Algorithm 1 with gradient input ∇f and stepsize $\alpha = 1/(2L)$ will converge to the global minima almost surely as the iteration count $T \to \infty$.*

C.2. Finite Iteration Bounds

To derive a finite iteration bound, we will apply stochastic gradient descent to our objective function and use standard martingale techniques for analysis. We will need to slightly alter the main result in (Ge et al., 2015) because we lack strong convexity assumptions. Also, we will accordingly alter the stochastic gradient descent algorithm to terminate upon finding a critical point that is γ-strict, for small γ.

Algorithm 2 $x = SGD(\widehat{L}, x_0, T, \alpha, \epsilon)$

Input: $\widehat{L} : \mathcal{M} \to \mathbb{R}$; $x_0 \in \mathcal{M}$; $T \in \mathbb{N}$; $\alpha \in \mathbb{R}$; $\epsilon \in \mathbb{R}$

Initialize $x = x_0$
for $i = 1$ **to** T **do**
  $x = x - \alpha(\nabla\widehat{L}(x) + n)$ [S: What is n?]
  $x = \Pi_{\mathcal{M}} x$
  if $\|\nabla\widehat{L}(x)\| \leq 2\epsilon/3$ **and** $\lambda_{min}(\nabla^2\widehat{L}(x)) \geq -2\epsilon/3$ **then**
    return x
  end if
end for
return FAIL

Theorem C.8. *Let $L : \Omega \to \mathbb{R}$ be a twice differentiable function such that $|L(x)| \leq B, \|\nabla^2 L(x)\| \leq C, \|\nabla^2 L(x) - \nabla^2 L(y)\| \leq \rho\|x - y\|$ for all $x, y \in \Omega$. Also, assume that we access to stochastic function $\widehat{L}$ such that $\|\widehat{L}(x) - L(x)\|, \|\nabla\widehat{L}(x) = L(x)\|, \|\nabla^2\widehat{L}(x) - \nabla^2 L(x)\| \leq \epsilon/3$ for all $x \in \Omega$.*

Then, we can choose stepsize $\eta = 1/poly(d, B, C, \rho, 1/\epsilon, \log(1/\zeta))$, such that with probability at least $1 - \zeta$, running Algorithm 2 on $\widehat{L}$ with stepsize η returns a point that is in $\mathcal{M}_\epsilon$ after at most $poly(d, B, C, \rho, 1/\epsilon, \log(1/\zeta))$ iterations.

Proof. If Algorithm 2 succeeds, then by triangle inequality and our error bounds on the stochastic function $\widehat{L}$ and its derivatives, we know that $x \in \mathcal{M}_\epsilon$. So, it suffices to argue that our algorithm succeeds with high probability.

By (Ge et al., 2015) Lemma 7 and 9, we see that if $x_i \notin \mathcal{M}_{\epsilon/3}$ for all the iterations, then in expectation, our objective function decreases by $poly(\eta)$ and by using Azuma's inequality, this occurs with probability $1 - \zeta$. Since our objective function is bounded by $poly(d, B)$, we conclude that stochastic gradient descent must have encountered a point x_i such that $x_i \in \mathcal{M}_{\epsilon/3}$ after at most $poly(d, B, C, \rho, 1/\epsilon, 1/\gamma, \log(1/\zeta))$. Then, by our bounds on $\widehat{L}$ and L, we conclude that both $\|\nabla\widehat{L}(x_i)\| \leq 2\epsilon/3$ and $\lambda_{min}(\nabla^2\widehat{L}(x_i)) \geq -2\epsilon/3$. So, our algorithm succeeds with probability $1 - \zeta$. □

D. Convergence of Almost Strictly Subharmonic Potentials

For concreteness, we will focus on a specific potential function with this property: the Gaussian kernel $\Phi(\theta, w) = \exp(-\|\theta - w\|^2/2)$. In $\mathbb{R}^d$, the Laplacian is $\Delta\Phi = (\|\theta - w\|^2 - d)\exp(-\|\theta - w\|^2/2)$, which becomes positive when $\|\theta - w\|^2 \geq d$. Thus, Φ is strictly subharmonic outside a ball of radius $\sqrt{d}$. This informally implies that θ_1 converges to a $\sqrt{d}$-ball around some w_j.

When a is fixed under Assumption 1, the node-wise SGD algorithm is given by Algorithm 3. For the Gaussian, we are only able to derive exponential time bounds on the runtime due to the fast decay of the potentials. Note that Gaussian potential restricted to S^{d-1} gives rise to the exponential activation function, so we can show convergence similarly.

Algorithm 3 Node-wise SGD Algorithm

Input: $\theta = (\theta_1, ..., \theta_k), \theta_i \in \mathcal{M}$; $T \in \mathbb{N}$; $\widehat{L}$; $\alpha \in \mathbb{R}$; $\delta \in \mathbb{R}$

Initialize $a = 0$
for $i = 1$ **to** k **do**
  $a_i = -1$
  $\theta_i = SGD(\widehat{L}_{\theta_i}, \theta_i, T, \alpha, \delta)$
end for
return $\theta = (\theta_1, ..., \theta_k)$

Theorem D.1. *Let $\mathcal{M} = \mathbb{R}^d$ and $\Phi(\theta, w) = e^{-\|a-b\|^2/2}$ and Assumption 1 holds. If $2d < \|\theta_i - w_j\|^2 < O(d)$ for all i, j, then $\theta = (\theta_1, ..., \theta_k)$ is a $e^{-O(d)}$-strict point of our simplified loss (4).*

Let $w_1, ..., w_k$ are randomly initialized according to $\mathcal{N}(0, c * \mathbf{I}_{d\times d})$, for some constant $c > 2$ such that $kc^2e^{-c} \leq e^{-2d}$. Then, with high probability, running Algorithm 3 initialized with $\theta = 0$, and error $\delta = e^{-O(d)}$, and stepsize $\alpha = e^{-O(d)}$ returns a point θ that is within $e^{-O(d)}$ of the global minima in $T = e^{O(d)}$ number of iterations.

Proof. Consider again a correlated movement, where each θ_i are moved along the same direction v. As before, this drops the pairwise θ_i terms, so since $O(d) > \|\theta_i - w_j\|^2 > 2d$, we see that $\Delta_{\theta_i}\Phi = (\|\theta_i - w_j\|^2 - d)\Phi(\theta_i, w_j) > e^{-O(d)}$.

$$\text{Tr}(\nabla^2 L) = -2\sum_{i=1}^{k}\sum_{j=1}^{k}\Delta_{\theta_i}\Phi(\theta_i, w_j) < -e^{-O(d)}$$

Therefore, $\nabla^2 L$ must admit a strictly negative eigenvalue that is less than $e^{-c_3 d}$, which implies our claim (we drop the $poly(d, k)$ terms).

Now, we proceed with the convergence proof. Consider the gradient descent output of the first node θ_1. The loss function is given by:

$$L(\theta_1) = -2\sum_{j=1}^{k}\Phi(\theta_1, w_j)$$

By standard Chernoff bounds, we have $\|w_i\|^2 < 3cd$ for all i and $\|w_i - w_j\|^2 \in (1.5cd, 2.5cd)$ for all i, j with high probability as $k = poly(d)$. We just need to consider $\Omega = \{x \in \mathbb{R}^d | \|x\|^2 < 3cd\}$ since the convex hull of w_i is contained in Ω and so the gradient operator g satisfy $g(\Omega) \subseteq \Omega$ (the gradient outside the convex hull points towards the convex hull).

To use Thm C.8, we check the regularity conditions: note that we can choose L, B, ρ to be $poly(d)$ since the first, second and third partials of Φ are all bounded by $poly(d)$. Now, by choosing $\epsilon = e^{-O(d)}$ for some c, we know that with high probability, running SGD in $T = e^{O(d)}$ iterations will output θ_1 such that 1) $\|\nabla L(\theta_1)\| < e^{-O(d)}$ and 2) $\lambda_{min}(\nabla^2 L(\theta_1)) > -e^{-O(d)}$.

WLOG, w_1 be a closest point to θ_1. Since $\lambda_{min}(\nabla^2 L(\theta_1)) > -e^{-O(d)}$, by the first part of our claim, $\|w_1 - \theta_1\|^2 < 1.1d$ and from triangle inequality, $\|w_i - \theta_1\|^2 > c * d$ for all $i \neq 1$.

Next, we calculate the gradient value at θ_1,

$$\begin{aligned}\|\nabla L(\theta_1)\| &= \|2\sum_{j=1}^{k}\nabla_{\theta_1}\Phi(\theta_1, w_j)\| \\ &\geq \|\nabla_{\theta_1}\Phi(\theta_1, w_1)\| - \|\sum_{j>1}\nabla_{\theta_1}\Phi(\theta_1, w_j)\| \\ &\geq \|\theta_1 - w_1\|e^{-1.1d} - kc^2e^{-cd} \geq e^{-O(d)}\end{aligned}$$

Therefore, $\epsilon = e^{-O(d)}$ can be chosen small enough to ensure that $\|\theta_1 - w_1\| = e^{-O(d)}$.

Finally, we proceed by induction. Since θ_1, w_1 are paired to high accuracy, we note that we can treat it as if w_1, θ_1 are removed from our loss equation. Therefore, we conclude that for all θ_i, it will match with some $w_{\pi(i)}$ for some permutation π, with error $e^{-O(d)}$. □

E. Convergence of Almost λ-Harmonic Potentials

Definition E.1. *$\Phi(\theta, w)$ is almost (λ, m)-Harmonic on S^{d-1} if for all $\theta, w \in \Omega$, $|\Delta_\theta\Phi(\theta, w) - \lambda\Phi(\theta, w)| \leq |\theta^T w|^m$*

[R: Not sure if there is a nice general definition to be honest...to be discussed]

When the output weights are variable, notice that our convergence results often rely on the optimality of the output weights. For this reason, we will optimize the output weights at every gradient descent step, by carefully choosing the stepsize. As our loss function (2) is quadratic in a, we know that gradient descent will find a^* efficiently.

Theorem E.2. *(Nesterov, 2013) Let $x_0 \in \Omega = \{x \in \mathbb{R}^d | \|x\| \leq poly(d)\}$ and let $L(x) = x^T Ax + b^T x$ be a quadratic loss, where A is a positive semi-definite matrix with maximum eigenvalue β. Then, running Algorithm 1 on L with stepsize $\alpha = 1/\beta$ converges to x_T such that*

$$L(x_T) - \min_{x\in\Omega} L(x) \leq \epsilon$$

in $T = poly(d, \beta, 1/\epsilon)$ iterations.

The pseudocode is given in Algorithm 4.

First, we need some control on the size of the squares of variable charges, a_i^2. Node-wise gradient descent allows us to maintain that control.

Theorem E.3. *Let $\Phi(\theta, w)$ be $poly(d)$-Lipschitz in θ, $|\Phi| \leq poly(d)$ on $\mathcal{M}$, and $\sum_{j\neq i}|a_j| + \sum_j |b_j| \leq poly(d)$. Also, assume that if θ_i is drawn uniformly on*

Algorithm 4 Node-wise Gradient Descent Algorithm with Output Weights Optimization

Input: $(\boldsymbol{a}, \boldsymbol{\theta}) = (a_1, ..., a_k, \theta_1, ..., \theta_k), a_i \in \mathbb{R}, \theta_i \in \mathcal{M}; T \in \mathbb{N}; \widehat{L}; \alpha \in \mathbb{R}; \delta \in \mathbb{R}; \gamma \in R;$

Initialize $a = 0$
for $i = 1$ **to** k **do**
  repeat
    Sample θ_i uniformly from $\mathcal{M}$
  until $(\frac{\partial \widehat{L}(\boldsymbol{a},\boldsymbol{\theta})}{\partial a_i})^2 \geq \gamma$
  for $j = 1$ **to** T **do**
    $a_i = a_i^* = GradientDescent(\widehat{L}_{a_i}, a_i, 1, \frac{\partial^2 \hat{L}}{\partial a_i^2})$
    $\theta_i = SGD(\widehat{L}_{\theta_i}, \theta_i, 1, \alpha, \delta)$
  end for
  $a = GradientDescent(\widehat{L}_{a_1,..,a_i}, (a_1, .., a_i), T, \alpha)$
end for
return $a = (a_1, ..., a_k), \theta = (\theta_1, ..., \theta_k)$

$\mathcal{M}$,

$$\mathop{\mathbb{E}}_{\theta_i}\left[\left(\sum_{j\neq i} a_j \Phi(\theta_i, \theta_j) + \sum_{j=1}^{k} b_j \Phi(\theta_i, w_j)\right)^2\right] \geq \epsilon$$

Then, with high probability, running Algorithm 4 on $\widehat{L}$ with stepsize at most $1/poly(d)$ and $\gamma = \epsilon$ will enforce that $a_i^2 = \Omega(\epsilon)$ when running SGD on θ_i. The sample complexity is $poly(d, 1/\epsilon)$

Proof. We want to show that $a_i^2 = \Omega(\epsilon)$. First, we analyze the random drawing of θ_i. Let

$$C(\theta_i) = \sum_{j\neq i} a_j \Phi(\theta_i, \theta_j) + \sum_{j=1}^{k} b_j \Phi(\theta_i, w_j)$$

By assumption, we know that $\mathbb{E}[C(\theta_i)^2] \geq \epsilon$. Since a_i, b_i, Φ is $poly(d)$-bounded, then $C(\theta_i)$ is $poly(d)$-bounded. From Hoeffding bounds, we see that after $poly(d, 1/\epsilon)$ samples, we know that $\hat{\mathbb{E}}C(\theta_i)^2 \geq \epsilon/2$, where $\hat{\mathbb{E}}$ denotes the empirical average. Therefore, we must have found some θ_i such that $C(\theta_i)^2 \geq \epsilon/2$. [S: I dont think Hoeffdings bounds apply here. If Φ is always bounded, you can just apply a Markov argument here].

Next, notice that $\hat{L}$ is a quadratic in a_i, which is a scalar. Therefore, the gradient step on a_i is chosen with the exact stepsize that will optimize $a_i = a_i^*(\theta_i)$. By optimality of a^*,

$$0 = 2a_i \widehat{\Phi}(\theta_i, \theta_i) + 2\sum_{j\neq i} a_j \widehat{\Phi}(\theta_i, \theta_j) + 2\sum_{j=1}^{k} b_j \widehat{\Phi}(\theta_i, w_j) \tag{6}$$

Therefore,

$$\begin{aligned} a_i^* &= -\frac{1}{\widehat{\Phi}(\theta_i, \theta_i)}\left(\sum_{j\neq i} a_j \widehat{\Phi}(\theta_i, \theta_j) + \sum_{j=1}^{k} b_j \widehat{\Phi}(\theta_i, w_j)\right) \\ &= -\frac{1}{\widehat{\Phi}(\theta_i, \theta_i)} \widehat{C}(\theta_i) \end{aligned}$$

[S: Is $\widehat{\Phi}$ computable efficiently? What about $\widehat{C}$?]

By our generalization bounds [S: Give a ref to the theorem], in $poly(d, 1/\epsilon)$ samples, we can find θ_i such that $\widehat{C}(\theta_i)^2 \geq \epsilon/4$ (notice $\Phi(\theta_i, \theta_i) = 1$ [S: but the above expression has $\widehat{\Phi}$]). As a_i is initially 0 when sampling θ_i, we conclude that we can find θ_i such that

$$\left(\frac{\partial \hat{L}}{\partial a_i}\right)^2 = 4\widehat{C}(\theta_i)^2 \geq \epsilon.$$

Therefore, after $poly(d, 1/\epsilon)$, our sampling algorithm will return θ_i such that $a_i^2 = a_i^*(\theta_i)^2 \geq \epsilon/4$.

Next, we examine the SGD steps. Note

$$\begin{aligned} \nabla_{\theta_i} \widehat{C}(\theta_i) &= \sum_{j\neq i} a_j \nabla_{\theta_i} \widehat{\Phi}(\theta_i, \theta_j) + \sum_{j=1}^{k} b_j \nabla_{\theta_i} \widehat{\Phi}(\theta_i, w_j) \\ &= \frac{1}{2a_i^*} \nabla_{\theta_i} \widehat{L} \end{aligned}$$

Now, we calculate the following gradient:

$$\nabla_{\theta_i}(a_i^*)^2 = 2a_i^* \frac{-1}{\widehat{\Phi}(\theta_i, \theta_i)} \nabla_{\theta_i} \widehat{C}(\theta_i) = \frac{1}{\widehat{\Phi}(\theta_i, \theta_i)}(-\nabla_{\theta_i} \widehat{L}) \tag{7}$$

Therefore, since θ_i moves in the direction of the gradient of $(a_i^*)^2$ in expectation and $(a_i^*)^2$ is $poly(d)$-Lipschitz in θ_i, so we conclude by a standard analysis of gradient descent with stepsize $1/poly(d)$ that $(a_i^*)^2$ is a supermartingale with a bounded difference of $poly(d)$. Finally, we can conclude with Azuma's, that in $poly(1/\epsilon, d, \log(1/\zeta))$ iterations, $(a_i^*)^2 = \Omega(\epsilon)$ with probability $1 - \zeta$. [S: The last part was too fast. Can you please elaborate what do you mean by moves in the same direction of the gradient of ..? Also, what analysis of SGD are you using, please cite a theorem and plug it in here? I dont know how do supermartingale bounds applies to SGD.] □

Theorem E.4. *Let $\mathcal{M} = S^{d-1}$ and let $b_1, ..., b_k$ be reals bounded by $poly(d)$. For any m, we can construct a realizable $(1, m)$-Harmonic potential, Φ_m.* [S: almost harmonic?]

Furthermore, let L as in (2) with potential Φ_m. Then, for all $1 > \epsilon > 0$, we can choose $m = O(\log(d)/\epsilon)$

such that if $(\boldsymbol{a}, \boldsymbol{\theta}) \in \mathcal{M}_{\epsilon^2/(4d)}$ *and* $\|a\| \le poly(d)$*, then each* (a_i, θ_i) *must satisfy at least one of the following conditions:*

i) There exists j *such that* $|w_j^T \theta_i| > 1 - \epsilon$

or *ii) There exists* $j \neq i$ *such that* $|\theta_j^T \theta_i| > 1 - \epsilon$

or *iii)* $|a_i| < \epsilon$.

Proof. We start with the construction. Let u, v be on the unit sphere, and WLOG let $v = (0, ..., 1)$. Then, we can reparametrize any function of the dot product $f(u^T v) = f(\cos(\theta))$, where $\theta \in [0, \pi]$ is the angle between u and the positive z-axis. Using the Laplacian formula on S^{d-1} gives:

$$\begin{aligned}\Delta f &= (\sin\theta)^{2-d} \frac{\partial}{\partial \theta}\left[(\sin\theta)^{d-2} \frac{\partial f(\cos\theta)}{\partial \theta}\right] \\ &= f''(\cos\theta)(\sin\theta)^2 - (d-1)\cos(\theta) f'(\cos\theta)\end{aligned}$$

We want f to be approximately 1-harmonic, so we want to approximately solve:

$$f''(x)(1-x^2) - (d-1)xf'(x) = f(x)$$

Let us construct our approximate function f as follows. Let c_n are the Taylor coefficients of $f(x)$, then we want to solve the equation: Therefore, we get the following recurrence: $c_n(n(n-1) + (d-1) * n + 1) = c_{n+2}(n+2)(n+1)$. Let $c_0 = 1, c_1 = 0$, then we run this recurrence holds $n = 0, ..., N-2$ for some N even and set $c_{N+i} = 0$ for $i > 0$, then

$$\begin{aligned}\Delta f &= f''(\cos\theta)(\sin\theta)^2 - (d-1)\cos(\theta) f'(\cos\theta) \\ &= f(\cos(x)) + k_N x^N\end{aligned}$$

Where $k_N = N^2 + (d-2)N + 1$. Notice that $f(x)$ has non-negative Taylor coefficients and is a bounded polynomial, so f is a realizable potential function by Theorem 2.2. We can also choose N to be odd by setting $c_0 = 0, c_1 = 1$. Therefore, we can construct a realizable $(1, m)$-Harmonic potential for any m.

We show our result under this potential function $\Phi = f$ and then we can normalize later. Assume that $(\boldsymbol{a}, \boldsymbol{\theta}) \in \mathcal{M}_{\epsilon^2/(4d)}$ and we have a (a_i, θ_i) that does not satisfy (i), (ii), (iii). Then,

$$|\frac{\partial L}{\partial a_i}| = |2\sum_{j=1}^{k} b_j \Phi(\theta_i^T w_j) + 2\sum_{j\neq i} a_j \Phi(\theta_i^T \theta_j) + 2a_i\Phi(1)| \le \epsilon$$

And,

$$\Delta_{\theta_i} L = 2\sum_{j=1}^{k} a_i b_j(-\Phi(\theta_i^T w_j) + k_N(\theta_i^T w_j)^N) + 2\sum_{j\neq i} a_i a_j(-\Phi(\theta_i^T\theta_j) + k_N(\theta_i^T\theta_j)^N)$$

Therefore,

$$|\Delta_{\theta_i} L + 2a_i^2\Phi(1)| \le \epsilon|a_i|$$

$$+2|a_i|k_N|\sum_{j=1}^{k} b_j(\theta_i^T w_j)^N + \sum_{j\neq i} a_j(\theta_i^T\theta_j)^N|$$

Since (i), (ii) does not hold and $\|a\| \le poly(d)$, we can choose $N = O(\frac{1}{\epsilon}\log(d))$ large enough such that

$$2k_N|\sum_{j=1}^{k} b_j(\theta_i^T w_j)^N + \sum_{j\neq i} a_j(\theta_i^T\theta_j)^N| \le 2poly(d,k,N)e^{-\epsilon N}$$

$$\le \epsilon/2$$

Therefore, by (iii) and $\Phi(1) > 1$,

$$\Delta_{\theta_i} L \le -2a_i^2\Phi(1) + \epsilon|a_i| + (\epsilon/2)|a_i| \le |a_i|(-2\epsilon\Phi(1) + 3\epsilon/2)$$

$$\le -\epsilon^2/4$$

This implies that $(\boldsymbol{a}, \boldsymbol{\theta})$ is $\epsilon^2/(4d)$-strict, contradicting that it is in $\mathcal{M}_{\epsilon^2/(4d)}$. □

Theorem E.5. *Let* $0 < \epsilon < 1/2$*. We can construct a realizable potential* Φ *such that with high probability, running Algorithm 4 on (2) with error* $\delta = poly(\epsilon, 1/d)$*,* $\gamma = \epsilon$ *and stepsize* $\alpha = 1/poly(d, 1/\epsilon)$ *converges in* $T = poly(d, 1/\epsilon)$ *iterations to* $(\boldsymbol{a}, \boldsymbol{\theta})$ *such that either* θ *is within* ϵ*-neighborhood of the global minima or there exists* i *such that if* θ_i *is picked uniformly in* $\mathcal{M}$

$$\mathbb{E}\left[\left(\sum_{j<i} a_j\Phi(\theta_i, \theta_j) + \sum_{j=1}^{k} b_j\Phi(\theta_i, w_j)\right)^2\right] < \epsilon$$

The sample complexity is $d^{poly(d,1/\epsilon)}$.

Proof. Let Φ_m be the $(1, m)$-Harmonic potential in Theorem E.4 with $m = poly(d, 1/\epsilon)$ and m odd. We first consider the algorithm on node θ_1 and claim that it will merge with some w_j and then we will proceed with induction.

If

$$\mathbb{E}\left[\left(\sum_{j<1} a_j\Phi(\theta_i, \theta_j) + \sum_{j=1}^{k} b_j\Phi(\theta_i, w_j)\right)^2\right] < \epsilon,$$

then we are done. Otherwise, with high probability, Theorem E.3 allows us to deduce that throughout the SGD algorithm applied on θ_1, $a_1^2 = \Omega(\epsilon)$.

Now we want to apply Theorem C.8, so we check the regularity conditions. Since $\mathcal{M} = S^{d-1}$, then we can choose B, L, ρ to be $poly(d)$ since Φ and the second and third partials of Φ are all bounded by $poly(d)$. Furthermore, by our construction, our activation function $\sigma(x)$ and its derivatives are $O(|x|^{poly(d,1/\epsilon)})$. By Theorem C.5, with high probability, we can construct a stochastic oracle up to $poly(\epsilon, 1/d)$ error with sample complexity $d^{poly(d,1/\epsilon)}$.

Therefore, by Theorem C.8 we conclude that we converge to $\theta_1 \in \mathcal{M}_{poly(\epsilon,1/d)}$. By Theorem E.4, since $|a_i| = \Omega(\sqrt{\epsilon})$, this implies that it is in an $poly(\epsilon, 1/d)$-neighborhood of some w_i in $poly(d, 1/\epsilon)$ time. Note that θ_1 will close to with $\pm w_j$ for some j but since Φ_m is odd, WLOG, it is close to w_j.

Furthermore, note that $|a_1| \leq poly(d)$ by using the explicit formula. And lastly, by Theorem E.2, since the maximum eigenvalue of our matrix A is bounded by poly(d), our gradient descent steps on the quadratic loss L_{a_1} will converge to the optimum in $\Omega = \{a \in \mathbb{R}^n | \|a\| \leq poly(d)\}$ with $O(\epsilon)$ error in T iterations.

Now, we proceed with induction and repeat the same argument on θ_2. We can simply treat θ_1 as w_{k+1} and so applying the same argument tells us that θ_2 is close to some w_j for some j. The issue is that θ_2 could be in a $poly(\epsilon, 1/d)$-neighborhood of $w_{k+1} = \theta_1$ or w_i. We claim that this will not occur. First, since w_i, w_{k+1} are in a $poly(\epsilon, 1/d)$-neighborhood of each other, we will assume WLOG that θ_2 is close to θ_1.

Now, by the optimality of a_1, we know that $L(a_1, \theta_1) \leq \min_{a \in \Omega} L(a, \theta_1) + O(\epsilon)$. We claim that if θ_2 is close to θ_1, then $L(a_1 + a_2, \theta_1) \leq L(a_1, \theta_1) - \Omega(\epsilon)$. This, combined with the fact that $a_1 + a_2$ is bounded by poly(d), would lead to a contradiction.

First, notice that since a_2 is always optimal, we have

$$\begin{aligned} &L(a_1, \theta_1) - L(a_1, \theta_1, a_2, \theta_2) \\ &\quad = a_2^2 + 2a_2\Big(\sum_{j<2} \Phi(\theta_2, \theta_j) + \sum_{j=1}^{k} b_j \Phi(\theta_2, w_j)\Big) \\ &\quad = -a_2^2 = \Omega(\epsilon) \end{aligned}$$

Therefore, it suffices to show that $|L(a_1 + a_2, \theta_1) - L(a_1, \theta_1, a_2, \theta_2)| \leq O(\epsilon)$. Since $L(a_1 + a_2, \theta) = L(a_1, \theta_1, a_2, \theta_1)$, this follows immediately from the $poly(d)$-Lipschitz of Φ and the fact that θ_2 is in a $poly(d, 1/\epsilon)$-neighborhood of θ_1. We conclude that θ_2 cannot be in a $poly(\epsilon, 1/d)$-neighborhood of θ_1 but converges to a point close to $w_j, j \neq i, k+1$. Therefore, the no two θ_i are matched to one w_j.

By applying this logic to all θ_i through induction, we deduce that θ is within ϵ of the global minima. □

F. Convergence of Polynomial Potentials

For realistic potentials, we can deduce finite time convergence results for polynomial activations, excluding the simple cases of linear and quadratic activations. For the sign activation, no such convergence results exist because of its non-smoothness. However, we believe that Lipschitz approximations to the sign activation should enjoy some related convergence guarantees.

Theorem F.1. *Let $\mathcal{M} = S^{d-1}$. Let $w_1, ..., w_d$ be orthonormal vectors in $\mathbb{R}^d$ and Φ is of the form $\Phi(\theta, w) = (\theta^T w)^l$ for some fixed integer $l \geq 3$. Furthermore, $1 \leq |b_i| \leq poly(d)$.*

Then, with high probability, running Algorithm 4 on (2) converges to an ϵ-neighborhood of the global minima in $poly(d, 1/\epsilon)$ time. The sample complexity is $poly(d, 1/\epsilon)$.

Proof. Without loss of generality let $w_1, ..., w_k$ be the standard basis vectors $e_1, .., e_k$. and these basis vectors span the whole optimization space. Consider the algorithm on just the first node: (a_1, θ_1). For simplicity, we will drop the subscripts in the proof.

First, consider the initialization of θ, notice that upon initialization and optimization, if θ is drawn from a standard Gaussian, then by independence and $\mathbb{E}[\theta_i] = 0$,

$$\mathbb{E}[C(\theta)^2] = \mathbb{E}\left[\left(\sum_{i=1}^{k} b_i (\theta_i)^l\right)^2\right] = \sum_{i=1}^{k} b_i^2 \, \mathbb{E}[\theta_i^{2l}]$$

There must exists j such that $|b_j| \geq 1$ and since drawing θ_i uniformly on S^{d-1} is just a $poly(d)$ rescaling of a Gaussian, we conclude that $E[a^2] \geq 1/poly(d)$. By Theorem E.3, we conclude that $E[a^2] \geq 1/poly(d)$ throughout the gradient descent algorithm. Next, to use Theorem C.8, we check the regularity conditions. By assumption, we can choose B, L, ρ to be $poly(d)$ since Φ and the second and third partials of Φ are all bounded by $poly(d)$ in Ω. Furthermore, our activation function σ and its derivatives are bounded in magnitude by $|x|^l$, where l is fixed. By Theorem C.5, with high probability, we can construct a stochastic oracle up to ϵ error with sample complexity $poly(d, 1/\epsilon)$.

Therefore, we conclude that we converge to $\theta \in \mathcal{M}_{poly(\epsilon,1/d)}$ in $poly(d, 1/\epsilon)$ iterations. Note that this is a constrained optimization problem, so by introducing a

Lagrange multiplier λ, the optimality conditions are:

$$|\frac{\partial L}{\partial a}| = |2\sum_{i=1}^{k} b_i(\theta_i)^l + 2a| \le poly(\epsilon, 1/d), \text{ and}$$

$$|(\nabla_\theta L)_i| = |\theta_i||2ab_i l(\theta_i)^{l-2} - 2\lambda| \le poly(\epsilon, 1/d),$$

where λ is chosen to minimize

$$\lambda = \arg\min_\lambda \sum_i (ab_i l(\theta_i)^{l-1} - \lambda\theta_i)^2 = \sum_i ab_i l(\theta_i)^l = -la^2 + poly(\epsilon, 1/d)$$

Therefore, either $|\theta_i| \le poly(\epsilon, 1/d)$ or $|2ab_i l(\theta_i)^{l-2} - 2\lambda| \le poly(\epsilon, 1/d)$. Next, the projected Hessian is a diagonal matrix with diagonal entry:

$$(\nabla^2 L)_{ii} = 2ab_i l(l-1)(\theta_i)^{l-2} - 2\lambda$$

Assume that there exists θ_i, θ_j such that $|\theta_i|, |\theta_j| \ge poly(\epsilon, 1/d)$, then we conclude that the other inequality must hold for coordinates i, j. So,

$$\begin{aligned}(\nabla^2 L)_{ii} &= 2ab_i l(l-1)(\theta_i)^{l-2} - 2\lambda \\ &\le 2(l-2)\lambda + (l-1)\epsilon \\ &\le -2(l-2)la^2 + poly(\epsilon, 1/d)\end{aligned}$$

Since $l - 2 \ge 1$ and $a^2 \ge 1/poly(d)$, we conclude that there exists a vector with $v_k = 0$ for all $k \ne i, j$ such that $v^T\theta = 0$ (in the tangent space) and $v^T(\nabla^2 L)v = -2(l-2)la^2 + poly(\epsilon, 1/d) < -poly(\epsilon, 1/d)$. This contradicts $\theta \in \mathcal{M}_{poly(\epsilon,1/d)}$, so θ is in some ϵ neighborhood of some w_j. Furthermore, note that $|a_1| \le poly(d)$ and there exists b_j such that $|b_j| \ge 1$ as some w_i has not yet been paired with a θ.

So we can proceed with induction and repeat the same argument on the second node θ_2 and so on. Since θ_1 is matched to some w_i, note that we can treat θ_1 as w_i up to some $poly(\epsilon, 1/d)$ error term. Furthermore, θ_2 will not converge to w_i because by the optimality of a_1, we note that $a_2^*(\theta_2) = 0$ if $\theta_2 = \theta_1$. (COPY SAME ARGUMENT AS λ-harmonic case) Since a_2^* is $poly(d)$-Lipschitz and $a_2^2 = a_2^*(\theta_2)^2$ is bounded below by $\Omega(\epsilon)$, we conclude that θ_2 cannot be close to θ_1 but converges to a point close to w_j, $j \ne i$. By applying this logic to all θ_i, we deduce that θ is within ϵ of the global minima. □

G. Proof of Sign Uniqueness

Proof. WLOG let $w_1 = e_1$. Notice that our loss can be bounded below by Jensen's:

$$\mathop{\mathbb{E}}_X \left[\left(\sum_{i=1}^k a_i\sigma(\theta_i^T X) - \sigma(X_1)\right)^2\right] \ge \mathop{\mathbb{E}}_{X_1}\left[\left(\mathop{\mathbb{E}}_{X_2\dots X_d}\left[\sum_{i=1}^k a_i\sigma(\theta_i^T X)\right] - \sigma(X_1)\right)^2\right],$$

where X is a standard Gaussian in $\mathbb{R}^d$.

$$\begin{aligned}E_{X_2,..,X_d}\left[\sum_{i=1}^k a_i\sigma(\theta_i^T X)\right] &= \sum_{i=1}^k a_i E_{X_2,\dots X_d}\left[\sigma(\theta_{i1}X_1 + \sum_{j>1}\theta_{ij}X_j)\right] \\ &= \sum_{i=1}^k E_Y\left[\sigma(\theta_{i1}X_1 + \sqrt{1-\theta_{i1}^2}Y)\right] \\ &= \sum_{i=1}^k a_i E_Y\left[\sigma(\tfrac{\theta_{i1}}{\sqrt{1-\theta_{i1}^2}}X_1 + Y)\right],\end{aligned}$$

where Y is an independent standard Gaussian and for any small δ, if $p(y)$ is the standard Gaussian density,

$$E_Y[\sigma(\delta + Y)] = \int_{-\delta}^{\delta} p(y)\,dy = 2p(0)\delta + O(\delta^2)$$

If $w_1^T\theta_i = \theta_{i1} < \epsilon$ for all i, then notice that with high probability on X_1 (say condition on $|X_1| \le 1$),

$$\mathop{\mathbb{E}}_Y\left[\sigma(\tfrac{\theta_{i1}}{\sqrt{1-\theta_{i1}^2}}X_1 + Y)\right] = 2p(0)\tfrac{\theta_{i1}}{\sqrt{1-\theta_{i1}^2}}X_1 + O(\epsilon^2 X_1^2)$$

Therefore, since $\epsilon < O(1/\sqrt{k})$,

$$\begin{aligned}\mathop{\mathbb{E}}_{X_2,..,X_d}\left[\sum_{i=1}^k a_i\sigma(\theta_i^T X)\right] &= X_1\sum_{i=1}^k 2p(0)a_i\tfrac{\theta_{i1}}{\sqrt{1-\theta_{i1}^2}} + O(k\epsilon^2 X_1^2) \\ &= cX_1 + O(1)\end{aligned}$$

Finally, our error bound is now

$$\begin{aligned}&\mathop{\mathbb{E}}_{X_1}\left[\left(\mathop{\mathbb{E}}_{X_2\dots X_d}\left[\sum_{i=1}^k a_i\sigma(\theta_i^T X)\right] - \sigma(X_1)\right)^2\right] \\ &\ge \mathop{\mathbb{E}}_{|X_1|\le 1}[(cX_1 + O(1) - \sigma(X_1))^2]\end{aligned}$$

And the final expression is always larger than some constant, regardless of c. □