Learning Compact Neural Networks with Regularization

Samet Oymak [1]

Abstract

Proper regularization is critical for speeding up training, improving generalization performance, and learning compact models that are cost efficient. We propose and analyze regularized gradient descent algorithms for learning shallow neural networks. Our framework is general and covers weight-sharing (convolutional networks), sparsity (network pruning), and low-rank constraints among others. We first introduce covering dimension to quantify the complexity of the constraint set and provide insights on the generalization properties. Then, we show that proposed algorithms become well-behaved and local linear convergence occurs once the amount of data exceeds the covering dimension. Overall, our results demonstrate that near-optimal sample complexity is sufficient for efficient learning and illustrate how regularization can be beneficial to learn over-parameterized networks.

1. Introduction

Deep neural networks (DNN) find ubiquitous use in large scale machine learning systems. Applications include speech processing, computer vision, natural language processing, and reinforcement learning (Krizhevsky et al., 2012; Graves et al., 2013; Hinton et al., 2012; Silver et al., 2016). DNNs can be efficiently trained with first-order methods and provide state of the art performance for important machine learning benchmarks such as ImageNet and TIMIT (Russakovsky et al., 2015; Graves et al., 2013). They also lie at the core of complex systems such as recommendation and ranking models and self-driving cars (Covington et al., 2016; Wang et al., 2015; Bojarski et al., 2016).

The abundance of promising applications bring a need to understand the properties of deep learning models. Recent literature shows a growing interest towards theoretical properties of complex neural network models. Significant questions of interest include efficient training of such models and their generalization abilities. Typically, neural nets are trained with first order methods that are based on (stochastic) gradient descent. The variations include Adam, Adagrad, and variance reduction methods (Kingma & Ba, 2014; Duchi et al., 2011; Johnson & Zhang, 2013). The fact that SGD is highly parallellizable is often crucial to training large scale models. Consequently, there is a growing body of works that focus on the theoretical understanding of gradient descent algorithms (Zhong et al., 2017b; Tian, 2017; Panigrahy et al., 2018; Soltanolkotabi et al., 2017; Ge et al., 2017; Janzamin et al., 2015) and the generalization properties of DNNs (Zhang et al., 2016; Hardt et al., 2015; Bartlett et al., 2017; Kawaguchi et al., 2017; Neyshabur et al., 2017).

In this work, we propose and analyze regularized gradient descent algorithms to provably learn compact neural networks that have space-efficient representation. This is in contrast to existing theory literature where the focus is mostly fully-connected neural networks (FNN). Proper regularization is a critical tool for building models that are compact and that have better generalization properties. This is achieved by reducing degrees of freedom of the model. Sparsifying and quantizing neural networks lead to storage efficient compact models that will be building blocks intelligent mobile devices (Han et al., 2015a;b; Courbariaux et al., 2016; Denton et al., 2014; Jin et al., 2016; Dong et al., 2017; Aghasi et al., 2017). The pruning idea has been around for many years (Hassibi & Stork, 1993; Cun et al., 1990) however it gained recent attention due to the growing size of the state of the art DNN models. Convolutional neural nets (CNN) are also compact models that efficiently utilize their parameters by weight sharing (Krizhevsky et al., 2012).

We study neural network regularization and address both generalization and optimization problems with an emphasis on one hidden-layer networks. We introduce a machinery to measure the impact of regularization, namely the covering dimension of the constraint set. We show that covering dimension controls generalization properties as well as the optimization landscape. Hence, regularization can have substantial benefit over training unconstrained (e.g. fully-connected) models and can help with training over-parameterized networks.

[1] University of California, Riverside, CA, USA. Work done at The Voleon Group, Berkeley, CA, USA. Correspondence to: <oymak@ece.ucr.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

Specifically, we consider the networks that are parametrized as $y = \boldsymbol{o}^T\sigma(\boldsymbol{W}\boldsymbol{x})$ where $\boldsymbol{x} \in \mathbb{R}^p$ is the input data, $\boldsymbol{W} \in \mathbb{R}^{h\times p}$ is the weight matrix, $\boldsymbol{o} \in \mathbb{R}^h$ is the output layer and $h \le p$. We assume $\boldsymbol{W} \in \mathcal{C}$ for some constraint set $\mathcal{C}$. We provide insights on the generalization and optimization performance by studying the tradeoff between the constraint set and the amount of training data (n) as follows.

- **Generalization error:** We study the Rademacher complexity and show that good generalization is achieved when data size n is larger than the sum of the covering dimension of $\mathcal{C}$ and the number of hidden nodes h.

- **Regularized first order methods:** We propose and analyze regularized gradient descent algorithms which incorporates the knowledge of $\mathcal{C}$ to iterations. We show that problem becomes well conditioned (around ground truth parameters) once the data size exceeds the covering dimension of the constraint set. This implies the local linear convergence of first order methods with *near-optimal sample complexity*. Recent results (as well as our experiments) indicate that it is not possible to do much better than this as random initialization can get stuck at spurious local minima (Zhong et al., 2017b; Safran & Shamir, 2017).

- **Application to CNNs:** We apply our results to CNNs and obtain improved global convergence guarantees when combined with the tensor initialization of (Zhong et al., 2017a). We also improve existing local convergence results on unconstrained problem (compared to (Zhong et al., 2017b)).

1.1. Related Works

Our results on the optimization landscape are closely related to the recent works on provably learning shallow neural nets (Zhong et al., 2017b; Tian, 2017; Soltanolkotabi, 2017; Panigrahy et al., 2018; Ge et al., 2017; Oymak & Soltanolkotabi, 2018; Zhong et al., 2017a; Safran & Shamir, 2017; Arora et al., 2014; Mei et al., 2016). (Janzamin et al., 2015) proposed tensor decomposition to learn shallow networks. (Tian, 2017) studies the gradient descent algorithm to train a model assuming population gradient. (Soltanolkotabi et al., 2017) focuses on training of shallow networks when they are over-parameterized and analyzes the global landscape for quadratic loss. More recently (Ge et al., 2017) shows global convergence of gradient descent by designing a new objective function instead of using ℓ_2-loss.

Our algorithmic results are closest to those of (Zhong et al., 2017b). Similar to us, authors focus on learning weights of a ground truth model where the input data is Gaussian. They propose a tensor based initialization followed by local gradient descent for learning one hidden-layer FNN. While we analyze a more general class of problems, when specialized to their setup, we improve their sample complexity and radius of convergence for local convergence. For instance, they need $\mathcal{O}\left(h^2p\right)$ samples to learn a FNN whereas we require $\mathcal{O}\left(hp\right)$ which is proportional to the *degrees of freedom* of the weight matrix.

Growing list of works (Brutzkus & Globerson, 2017; Oymak & Soltanolkotabi, 2018; Du et al., 2017b;a; Zhong et al., 2017a) investigate CNNs with a focus on non-overlapping structure. Unlike these, we formalize CNN as a low-dimensional subspace constraint and show sample optimal local convergence even with multiple kernels and overlapping structure. As discussed in Section 4, we also improve the global convergence bounds of (Zhong et al., 2017a).

Generalization properties of deep networks recently attracted significant attention (Zhang et al., 2016; Hardt et al., 2015; Bartlett et al., 2017; Konstantinos et al., 2017). Our results are closer to (Bartlett et al., 2017; Neyshabur et al., 2017; Konstantinos et al., 2017) which studies the problem in a learning theory framework. (Bartlett et al., 2017; Neyshabur et al., 2017) provide generalization bounds for deep FCNNs based on spectral norm of the individual layers. More recently, (Konstantinos et al., 2017) specializes such bounds to CNNs. Our result differs from these in two ways. First, our bound reflects the impact of regularization and secondly, we avoid the dependencies on input data length by taking advantage of the Gaussian data model.

2. Problem Statement

Here, we describe the general problem formulation. Our aim is learning neural networks that efficiently utilize their parameters by using gradient descent and proper regularization. For most of the discussion, the input/output $(y_i, \boldsymbol{x}_i)_{i=1}^n$ relation is given by

$$y_i = \boldsymbol{o}^T\sigma(\boldsymbol{W}^\star\boldsymbol{x}_i).$$

Here $\boldsymbol{o} \in \mathbb{R}^h$ is the vector that connects hidden to output layer and $\boldsymbol{W}^\star \in \mathbb{R}^{h\times p}$ is the weight matrix that connects input to hidden layer. Assuming $\boldsymbol{o}$ is known we are interested in learning $\boldsymbol{W}^\star$ which has hp degrees of freedom. The associated loss function for the regression problem is

$$\mathcal{L}(\boldsymbol{W}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \boldsymbol{o}^T\sigma(\boldsymbol{W}\boldsymbol{x}_i))^2.$$

Starting from an initial point $\boldsymbol{W}_0$, gradient descent algorithms learns $\boldsymbol{W}^\star$ using the following iterations

$$\boldsymbol{W}_{i+1} = \boldsymbol{W}_i - \mu\nabla\mathcal{L}(\boldsymbol{W}_i).$$

If we have a prior on $\boldsymbol{W}^\star$, such as sparse weights, this information can be incorporated by projecting $\boldsymbol{W}$ on the constraint set. Suppose $\boldsymbol{W}^\star$ lies in a constraint set $\mathcal{C}$. Denote the projection on $\mathcal{C}$ by $\mathcal{P}_{\mathcal{C}}(\cdot)$. Starting from an initial point $\boldsymbol{W}_0$, the **Projected Gradient Descent (PGD) algorithm** is characterized by the following iterations

$$\boldsymbol{W}_{i+1} = \mathcal{P}_{\mathcal{C}}(\boldsymbol{W}_i - \mu\nabla\mathcal{L}(\boldsymbol{W}_i)). \tag{1}$$

Our goal will be to understand the impact of $\mathcal{C}$ on generalization as well as the properties of the PGD algorithm.

2.1. Compact Models and Associated Regularizers

In order to learn parameter-efficient compact networks, practical approaches include weight-sharing, weight pruning, and quantization as explained below.

- **Convolutional model (weight-sharing):** Suppose we have a CNN with k kernels of width b. Each kernel is shifted and multiplied with length b patches of the input data i.e. same kernel weights are used many times across the input. In Section 4, we formulate this as an FNN subject to a subspace constraint where the constraint $\mathcal{C}$ is a kb dimensional subspace.
- **Sparsity:** Weight matrix $\boldsymbol{W}^\star$ has at most s nonzero weights out of hp entries.
- **Quantization:** Weights are restricted to be discrete values. In the extreme case, entries of $\boldsymbol{W}^\star$ are ±1.
- **Low-rank approximation:** Weight matrix $\boldsymbol{W}^\star$ obeys $\text{rank}(\boldsymbol{W}^\star) \le r$ for some $r \le h$.

We also consider convex regularizers which can yield smoother optimization landscape (e.g. subspace, ℓ_1). Convexified version of sparsity constraint is the ℓ_1 regularization. Parametrized by $\tau > 0$, the constraint set is given by

$$\mathcal{C} = \{\boldsymbol{W} \in \mathbb{R}^{h\times p} \bigm| \|\boldsymbol{W}\|_1 \le \tau\}.$$

Similarly, the convexified version of low-rank projection is the nuclear norm regularization, which corresponds to the ℓ_1 norm of singular values (Recht et al., 2010).

Finally, we remark that our results can be specialized to the **unconstrained problem** where the constraint set is $\mathcal{C} = \mathbb{R}^{h\times p}$ and PGD reduces to gradient descent.

Notation: Throughout the paper, h denotes the number of hidden nodes, p denotes the input dimension, and n denotes the number of data points unless otherwise stated. $\boldsymbol{s}_{\min}(\cdot), \boldsymbol{s}_{\max}(\cdot)$ returns the minimum/maximum singular values of a matrix. $\kappa(\boldsymbol{V})$ returns the condition number of the matrix $\boldsymbol{s}_{\max}(\boldsymbol{V})/\boldsymbol{s}_{\min}(\boldsymbol{V})$. Similarly, for a vector $\boldsymbol{v}$, $\kappa(\boldsymbol{v}) = \max_i |\boldsymbol{v}_i|/\min_i |\boldsymbol{v}_i|$. Frobenius norm and spectral norm are denoted by $\|\cdot\|_F$, $\|\cdot\|$ respectively. $c, C > 0$ denote absolute constants. $\mathcal{N}(0, \boldsymbol{I}_d)$ will denote a vector in $\mathbb{R}^d$ with i.i.d. standard normal entries. $\textbf{var}[\cdot]$ returns the variance of a random variable.

3. Main Results

We first introduce covering numbers to quantify the impact of regularization.

3.1. Covering Dimension

If constraint set $\mathcal{C}$ is a d-dimensional subspace (e.g. $\mathcal{C} = \mathbb{R}^{h\times p}$), weight matrices $\boldsymbol{W} \in \mathcal{C}$ has d degrees of freedom. This model applies to convolutional and unconstrained problems. For subspaces, the dimension d is sufficient to capture the problem complexity and our main results apply when the data size n obeys $n \ge \mathcal{O}(d)$. For other constraint types such as sparsity and matrix rank, we consider the constraint set given by

$$\mathcal{C} = \{\boldsymbol{W} \in \mathbb{R}^{h\times p} \bigm| \mathcal{R}(\boldsymbol{W}) \le \tau\}$$

where $\mathcal{R}$ is the regularizer function such as ℓ_1 norm. To capture the impact of regularizer, we define *feasible ball* which is the set of feasible directions given by

$$\mathcal{T} = \mathcal{B}^{h\times p} \bigcap \textbf{cl}\left(\left\{\alpha \boldsymbol{U} \in \mathbb{R}^{h\times p} \bigm| \boldsymbol{W}^\star + \boldsymbol{U} \in \mathcal{C},\ \alpha \ge 0\right\}\right) \quad (2)$$

where $\textbf{cl}(\cdot)$ is the set closure and $\mathcal{B}^{h\times p}$ is the unit Frobenius norm ball. For instance, when $\mathcal{R}$ is the ℓ_0 norm, $\mathcal{T}$ is a subset of $\tau + \|\boldsymbol{W}^\star\|_0$ sparse weight matrices.

Covering number is a standard way to measure the complexity of a set (Shalev-Shwartz & Ben-David, 2014). We will quantify the *impact of regularization* by using "covering dimension" which is defined as follows.

Definition 3.1 (Covering dimension)**.** *Let $T \subset \mathcal{B}^{h\times p}$ and $C > 0$ be an absolute constant. Covering dimension of T is denoted by cover(T) and is defined as follows. Suppose there exists a set S satisfying*

- *$T \subset \overline{conv}(S)$ where $\overline{conv}(S)$ is the minimal closed convex set containing S.*
- *Radius of S obeys $\sup_{\boldsymbol{v}\in S} \|\boldsymbol{v}\|_{\ell_2} \le C$.*
- *For all $\varepsilon > 0$, ℓ_2 ε-covering number of S obeys $N_\varepsilon(S) \le (1 + \frac{B}{\varepsilon})^s$ for some $s \ge 0, B > 1$ and all $\varepsilon > 0$.*

Then, cover$(T) \le s \log B$. Hence cover(T) is the infimum of all such upper bounds.

As illustrated in Table 1, covering dimension captures the degrees of freedom for practical regularizers. This includes sparsity, low-rank, and weight-sharing constraints discussed previously. Note that Table 1 is obtained by setting $\tau = \mathcal{R}(\boldsymbol{W}^\star)$. In practice, a good choice for τ can be found by using cross-validation. It is also known that the performance of PGD is robust to choice of τ (see Thm 2.6 of (Oymak et al., 2017)). For unstructured constraint sets without a clean covering number, one can use stronger tools from geometric functional analysis. In the extended manuscript (Oymak, 2018), we discuss how more general complexity estimates can be achieved by using *Gaussian width* of $\mathcal{T}$ (Chandrasekaran et al., 2012) and establish a connection to covering dimension.

Our results will apply in the regime $n \gtrsim \text{cover}(\mathcal{T})$ where n is the number of data points. This will allow sample size to be proportional to the *degrees of freedom* of the constraint space implying data-efficient learning. Now that we can quantify the impact of regularization, we proceed to state our results.

Constraint	Weight matrix model	cover$(\mathcal{T})$
None	$\boldsymbol{W}^\star \in \mathbb{R}^{h\times p}$	hp
Convolutional	k kernels of b width	kb
Sparsity $\|\cdot\|_0$	s nonzero weights	$s\log(6hp/s)$
ℓ_1 norm $\|\cdot\|_1$	s nonzero weights	$s\log(6hp/s)$
Subspace	$\boldsymbol{W}^\star \in S$, $\dim(S) = k$	k
Matrix rank	rank$(\boldsymbol{W}^\star) \le r$	rh

Table 1: The list of low-dimensional models and corresponding covering dimensions (up to a constant factor) for the constraint sets $\mathcal{C} = \{\boldsymbol{W} \mid \mathcal{R}(\boldsymbol{W}) \le \mathcal{R}(\boldsymbol{W}^\star)\}$. If constraint is set membership such as subspace, $\mathcal{R}(\boldsymbol{W}) = 0$ inside the set and ∞ outside.

3.2. Generalization Properties

To provide insights on generalization, we derive the Rademacher complexity of regularized neural networks with 1-hidden layer. To be consistent with the rest of the paper, we focus on Gaussian data distribution. Rademacher complexity is a useful tool that measures the richness of a function class and that allows us to give generalization bounds. Given sample size n, let $\boldsymbol{r} \in \mathbb{R}^n$ be an i.i.d. Rademacher vector. Let $\{\boldsymbol{x}_i\}_{i=1}^n$ are input data points that are i.i.d. with $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_p)$. Finally, let $\mathcal{F}$ be the class of neural nets we analyze. Then, Rademacher complexity of $\mathcal{F}$ with respect to Gaussian data with n samples is given by

$$\text{Rad}(\mathcal{F}) = \frac{1}{n}\mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^n}[\mathbb{E}_{\boldsymbol{r}}[\sup_{f\in\mathcal{F}}\sum_{i=1}^n \boldsymbol{r}_i f(\boldsymbol{x}_i)]]$$

The following lemma provides the result on Rademacher complexity of networks with low-covering numbers.

Lemma 3.2. *Suppose the activation function σ is L-Lipschitz. Consider the class of one hidden-layer networks $\mathcal{F}$ where $f \in \mathcal{F}$ is parametrized by its input matrix $\boldsymbol{W}$ and output vector $\boldsymbol{o}$ and satisfies*

- *input/output relation is $f_{\boldsymbol{o},\boldsymbol{W}}(\boldsymbol{x}) = \boldsymbol{o}^T\sigma(\boldsymbol{W}\boldsymbol{x})$,*
- *$\|\boldsymbol{W}\| \le R_{\boldsymbol{W}}$ and $\boldsymbol{W} \in \mathcal{C}$ where ε-covering number of $\mathcal{C}$ obeys $N_\varepsilon(\mathcal{C}) \le (1 + B/\varepsilon)^s$ for some $B > 0,\ s \ge 0$,*
- *$\|\boldsymbol{o}\|_{\ell_2} \le R_{\boldsymbol{o}}$.*

For Gaussian input data $\{\boldsymbol{x}_i\}_{i=1}^n \sim \mathcal{N}(0, \boldsymbol{I}_p)^n$, Rademacher complexity of class $\mathcal{F}$ is bounded by

$$Rad(\mathcal{F}) \le LR_{\boldsymbol{o}}R_{\boldsymbol{W}}\mathcal{O}\left(\frac{(h+s)\log(n+p) + s\log(1 + \frac{B}{R_{\boldsymbol{W}}})}{n}\right)^{1/2}$$

This result obeys typical Rademacher complexity bounds however the ambient dimension hp is replaced by the total degrees of freedom which is given in terms of $h + s\log B$. Furthermore, unlike (Bartlett et al., 2017), we do not have dependence on the length of the input data which is $\mathbb{E}[\|\boldsymbol{x}\|_{\ell_2}] \approx \sqrt{p}$. This is because we take advantage of the Gaussianity of input data which allows us to escape from the worst-case analysis that suffer from $\mathbb{E}[\|\boldsymbol{x}\|_{\ell_2}]$. Combined with standard learning theory results (Shalev-Shwartz & Ben-David, 2014), this bound shows that *empirical risk minimization* achieves small generalization error as soon as $n \sim \mathcal{O}(h + s\log B)$ samples. Observe that $\mathcal{O}(s)$ components of Rad$(\mathcal{F})$ relate to the covering dimension of $\mathcal{C}$ and become dominant as soon as $s \ge h$.

We remark that typically $B \sim \mathcal{O}(R_{\boldsymbol{W}})$. For instance, if $\mathcal{C}$ is a B scaled unit ℓ_2 ball, in order to ensure it contains $R_{\boldsymbol{W}}$ scaled spectral ball $\{\boldsymbol{W} \mid \|\boldsymbol{W}\| \le R_{\boldsymbol{W}}\}$, we need to pick $B = \sqrt{h}R_{\boldsymbol{W}}$.

Our main results are dedicated to the properties of the PGD algorithm where the aim is to learn compact neural nets efficiently. We show that Rademacher complexity bounds are highly consistent with the sample complexity requirements of PGD which is governed by the local optimization landscape such as positive-definiteness of the Hessian matrix.

3.3. Local Convergence of Regularized Training

A crucial ingredient of the convergence analysis of PGD is the positive-definiteness of Hessian along restricted directions dictated by $\mathcal{T}$ (Negahban & Wainwright, 2012). Denoting Hessian at the ground truth $\boldsymbol{W}^\star$ by $\boldsymbol{H}_{\boldsymbol{W}^\star}$, we investigate its restricted eigenvalue

$$H(\boldsymbol{W}^\star, \mathcal{T}) = \inf_{\boldsymbol{v}\in\mathcal{T}} \boldsymbol{v}^T\boldsymbol{H}_{\boldsymbol{W}^\star}\boldsymbol{v}$$

in the regime $h \le p$. Positivity of $H(\boldsymbol{W}^\star, \mathcal{T})$ will ensure that the problem is well conditioned around $\boldsymbol{W}^\star$ and is locally convergent. However, radius of convergence is not guaranteed to be large. Below, we present a summary of our results to provide basic insights about the actual technical contribution while avoiding the exact technical details.

• **Sample size:** Whether the constraint set $\mathcal{C}$ is *convex or nonconvex*, we have $H(\boldsymbol{W}^\star, \mathcal{T}) > 0$ as soon as

$$n \ge \mathcal{O}(\text{cover}(\mathcal{T})).$$

This implies *sample optimal* local convergence for subspace, sparsity and rank constraints among others.

• **Radius of convergence:** Basin of attraction for the PGD iterations (1) are $\mathcal{O}(h^{-1})$ neighborhood of $\boldsymbol{W}^\star$ i.e. we require

$$\|\boldsymbol{W}_0 - \boldsymbol{W}^\star\|_F \le \mathcal{O}(h^{-1}\|\boldsymbol{W}^\star\|_F).$$

As there are more hidden nodes, we require a tighter initialization. However, the result is independent of p.

- **Rate of convergence:** Within radius of convergence, weight matrix distance $\|\boldsymbol{W}_i - \boldsymbol{W}^\star\|_F^2$ reduces by a factor of

$$\rho = 1 - \mathcal{O}\left(\frac{1}{\max\{1, n^{-1} p \log p\} h \log p}\right),$$

at each iteration, which implies *linear convergence*. As long as the problem is not extremely overparametrized (i.e. $n \geq p \log p$), ignoring log terms, rate of convergence is $1 - \mathcal{O}(1/h)$. This implies accurate learning in $\mathcal{O}\left(h \log \varepsilon^{-1}\right)$ steps given target precision ε.

We are now in a place to state the main results. We place the following assumptions on the activation function for our results. It is a combination of smoothness and nonlinearity conditions.

Assumption 1 (Activation function). *$\sigma(\cdot)$ obeys following properties:*

- *$\sigma(\cdot)$ is differentiable, $\sigma'(\cdot)$ is an L-Lipschitz function and $|\sigma'(0)| \leq L_0$ for some $L, L_0 > 0$.*
- *Given $g \sim \mathcal{N}(0,1)$ and $\theta > 0$, define $\zeta(\theta)$ as*

$$\zeta(\theta) = \min\{\mathbf{var}[\sigma'(\theta g)] - \mathbb{E}[\sigma'(\theta g) g]^2, \\ \mathbf{var}[\sigma'(\theta g) g] - \mathbb{E}[\sigma'(\theta g) g^2]^2\}$$

where expectations are with respect to g. $\zeta(\theta) > 0$.

Example functions that satisfy the assumptions are

- Sigmoid and hyperbolic tangent,
- Error function $\sigma(x) = \int_0^x \exp(-t^2) dt$,
- Squared ReLU $\sigma(x) = \max\{0, x\}^2$,
- Softplus $\sigma(x) = \log(1 + \exp(x))$ (for sufficiently large θ).

While ReLU does not satisfy the criteria, a smooth ReLU approximation such as softplus works. In general, definition of $\zeta(\cdot)$ reveals that our assumptions are satisfied if σ i) is nonlinear, ii) is increasing, iii) has bounded second derivative, and iv) has symmetric first derivative (see Theorem 5.3 of (Zhong et al., 2017b)).

The $\zeta(\theta)$ quantity is a measure of the nonlinearity of the activation function. It will be used to control the minimum eigenvalue of Hessian. A very similar quantity is used by (Zhong et al., 2017b) where they have an extra term which is not needed by us. This implies, our $\zeta(\theta)$ is positive under milder conditions.

Definition 3.3 (Critical quantities). *Θ will be used to lower bound $H(\boldsymbol{W}^\star, \mathcal{T})$ and Ω will control the learning rate. They are defined as follows*

$$\Theta = \frac{L^2 \boldsymbol{s}_{\max}^2 \kappa^2(\boldsymbol{o}) \kappa^{h+2}(\boldsymbol{W}^\star)}{\zeta(\boldsymbol{s}_{\min})}, \quad \Omega = h(\log p + \frac{L_0^2}{L^2 \boldsymbol{s}_{\max}^2})$$

Θ will be a measure of the conditioning of the problem. It is essentially unitless and obeys $\Theta \geq 1$ since $L^2 \boldsymbol{s}_{\max}^2 \geq \zeta(\boldsymbol{s}_{\min})$. Ω will be inversely related to the radius of convergence and learning rate. If $L_0 = 0$ (e.g. quadratic activation), Ω simplifies to $h \log p$

3.3.1. Restricted Eigenvalue of Hessian

Our first result is a sample complexity bound for the restricted positive definiteness of the Hessian matrix at $\boldsymbol{W}^\star$. It implies that problem is locally well-conditioned with minimal data ($n \sim$ cover($\mathcal{T}$)).

Theorem 3.4. *Suppose $\mathcal{C}$ is a closed set that includes $\boldsymbol{W}^\star$, $h \leq p$, and let $\{\boldsymbol{x}_i\}_{i=1}^n$ be i.i.d. $\mathcal{N}(0, \boldsymbol{I}_p)$ data points. Set $\bar{v} = C\Theta \log^2(C\Theta)$ and suppose*

$$n \geq \mathcal{O}\left((\sqrt{cover(\mathcal{T})} + t)^2 \bar{v}^4\right).$$

With probability $1 - \exp(-n/\bar{v}^2) - 2\exp(-\mathcal{O}\left(\min\{t\sqrt{n}, t^2\}\right))$, we have that[1]

$$H(\boldsymbol{W}^\star, \mathcal{T}) \geq \frac{\zeta(\boldsymbol{s}_{\min}) \boldsymbol{o}_{\min}^2}{\kappa^{h+2}(\boldsymbol{W}^\star) \bar{v}^3}.$$

Proof sketch. Given a data point $\boldsymbol{x} \in \mathbb{R}^p$, we define $d(\boldsymbol{x}) = \boldsymbol{o} \odot \sigma'(\boldsymbol{W}^\star \boldsymbol{x}) \in \mathbb{R}^h$ where σ' is the entrywise derivative and $\odot$ is entrywise product. Then, define $\rho(\boldsymbol{x}) = d(\boldsymbol{x}) \otimes \boldsymbol{x} \in \mathbb{R}^{hp}$ where $\otimes$ is the Kronecker product. At ground truth, we have

$$H_{\boldsymbol{W}^\star} = n^{-1} \sum_{i=1}^n \rho(\boldsymbol{x}_i) \rho(\boldsymbol{x}_i)^T. \tag{3}$$

After showing $\boldsymbol{\Sigma} = \mathbb{E}[\rho(\boldsymbol{x}) \rho(\boldsymbol{x})^T]$ is positive definite, we need to ensure that

$$H(\boldsymbol{W}^\star, \mathcal{T}) \geq \mathcal{O}(\boldsymbol{s}_{\min}(\boldsymbol{\Sigma})), \tag{4}$$

with finite sample size n. This boils down to a high-dimensional statistics problem. We first show that $\rho(\boldsymbol{x})$ has subexponential tail for $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_p)$ i.e. for all unit vectors $\boldsymbol{v}$, $\mathbb{P}(|\boldsymbol{v}^T(\rho(\boldsymbol{x}) - \mathbb{E}[\rho(\boldsymbol{x})])| \geq t) \leq 2\exp(-C_{\sigma, \boldsymbol{o}, \boldsymbol{W}^\star} t)$. Next, we prove a novel restricted eigenvalue result for random matrices with subexponential rows as in (3). This is done by combining Mendelson's small-ball argument with tools from generic chaining (Mendelson, 2014; Talagrand, 2006). Careful treatment is necessary to address the facts that $\rho(\boldsymbol{x}_i)$ is *not zero-mean* and *its tail depends on* $\sigma, \boldsymbol{o}, \boldsymbol{W}^\star$. Our final result ensures (4) with $n \geq \mathcal{O}(\text{cover}(\mathcal{T}))$ samples where $\mathcal{O}()$ has the dependencies on the aforementioned variables. □

[1] If $\boldsymbol{W}^\star$ has orthogonal rows, $\kappa^{h+2}(\boldsymbol{W}^\star)$ term can be removed from Θ by utilizing a more involved analysis that uses an alternative definition of ζ. The reader is referred to the supplementary material.

3.3.2. Linear Convergence of PGD

Our next result utilizes Theorem 3.4 to characterize PGD around $\mathcal{O}(1/h)$ neighborhood of the ground truth.

Theorem 3.5. *Suppose $\mathcal{C}$ is a convex and closed set that includes $\boldsymbol{W}^\star$ and let $\{\boldsymbol{x}_i\}_{i=1}^n$ be i.i.d. $\mathcal{N}(0,\boldsymbol{I}_p)$ data points. Set $\bar{v} = C\Theta\log^2(C\Theta)$ and suppose*

$$n \ge \mathcal{O}\left((\sqrt{cover(\mathcal{T})}+t)^2\bar{v}^4\right), \tag{5}$$

Set $q = \max\{1, 8n^{-1}p\log p\}$. Define learning rate μ and rate of convergence ρ as

$$\mu = \frac{1}{6q\boldsymbol{o}_{\max}^2L^2\Omega},\ \rho = 1 - \frac{1}{12q\bar{v}^4\Omega} \tag{6}$$

Given $\boldsymbol{W}$ (independent of data points), consider the PGD iteration

$$\hat{\boldsymbol{W}} = \mathcal{P}_{\mathcal{C}}(\boldsymbol{W} - \mu\nabla\mathcal{L}(\boldsymbol{W}))$$

Suppose $\boldsymbol{W}$ satisfies $\|\boldsymbol{W}-\boldsymbol{W}^\star\|_F \le \mathcal{O}\left(\frac{\|\boldsymbol{W}^\star\|_F}{q\sqrt{h\Omega\log p}\bar{v}^4}\right)$. Then, $\hat{\boldsymbol{W}}$ obeys

$$\|\hat{\boldsymbol{W}}-\boldsymbol{W}^\star\|_F^2 \le \rho\|\boldsymbol{W}-\boldsymbol{W}^\star\|_F^2,$$

with probability $1 - P$ where $P = \exp(-n/\bar{v}^2) + 2\exp(-\mathcal{O}\left(\min\{t\sqrt{n}, t^2\}\right)) + 8(n\exp(-p/2) + np^{-10} + \exp(-qn/4p))$.

3.3.3. Convergence to the Ground Truth

Theorem 3.5 shows the improvement of a single iteration. Unfortunately, it requires the existence of fresh data points at every iteration. Once the initialization radius becomes tighter ($\mathcal{O}\left(p^{-1/2}h^{-1}\right)$ rather than $\mathcal{O}\left(h^{-1}\right)$), we can show a uniform convergence result that allows $\boldsymbol{W}$ to depend on data points. Combining both, the following corollary shows that repeated applications of projected gradient converges in $\mathcal{O}\left(h\log\varepsilon^{-1}\right)$ steps to ε neighborhood of $\boldsymbol{W}^\star$ using $\mathcal{O}\left(\text{cover}(\mathcal{T})h\log^2 p\right)$ samples. This is in contrast to related works (Zhong et al., 2017b;a) which always require fresh data points.

Theorem 3.6. *Consider the setup of Theorem 3.5. Let $K = \mathcal{O}\left(q\bar{v}^4\Omega\log p\right)$. Given $\bar{n} = Kn$ independent data points (where n obeys (5)), split dataset into K equal batches. Starting from a point $\|\boldsymbol{W}_0-\boldsymbol{W}^\star\|_F \le \mathcal{O}\left(\frac{\|\boldsymbol{W}^\star\|_F}{q\sqrt{h\Omega\log p}\bar{v}^4}\right)$, apply the PGD iterations*

$$\boldsymbol{W}_{i+1} = \boldsymbol{W}_i - \mu\nabla\mathcal{L}_{\min\{i,K\}}(\boldsymbol{W}_i),$$

where $\mathcal{L}_i$ is the loss function associated with ith batch. With probability $1-KP$, all $\boldsymbol{W}_i$ for $i \ge 1$ obey

$$\|\boldsymbol{W}_i-\boldsymbol{W}^\star\|_F^2 \le \rho^i\|\boldsymbol{W}_0-\boldsymbol{W}^\star\|_F^2.$$

4. Application to Convolutional Neural Nets

We now illustrate how CNNs can be treated under our framework. To describe shallow CNN, suppose we have k kernels $\{\boldsymbol{k}_i\}_{i=1}^k$ each with width b. Set $\mathbf{K} = [\boldsymbol{k}_1\ \dots\ \boldsymbol{k}_k]^T \in \mathbb{R}^{k\times b}$. Denote stride size by s and set $r = \lfloor p/s\rfloor$.

To describe our argument, we introduce some notation specific to the convolutional model. Let $\boldsymbol{v}^\ell \in \mathbb{R}^b$ denote ith subvector of $\boldsymbol{v} \in \mathbb{R}^p$ corresponding to entries from $\ell s+1$ to $\ell s+b$ for $0 \le \ell \le r-1$. Also given $\boldsymbol{b} \in \mathbb{R}^b$, let $\boldsymbol{v} = \text{map}_\ell(\boldsymbol{b}) \in \mathbb{R}^p$ be the vector obtained by mapping $\boldsymbol{b}$ to the ith subvector i.e. $\boldsymbol{v}^j = \boldsymbol{b}$ if $j = \ell$ and 0 otherwise.

For each data point $\boldsymbol{x}_j$, we consider its r subvectors $\{\boldsymbol{x}_j^l\}_{l=1}^r$ and filter each subvector with each of the kernels. Then, the input/output relation has the following form (assuming output layer weights $\boldsymbol{o}_{i,l}$)

$$y_{CNN}(\mathbf{K},\boldsymbol{x}_j) = \sum_{i=1}^k\sum_{l=1}^r \boldsymbol{o}_{i,l}\sigma(\boldsymbol{k}_i^T\boldsymbol{x}_j^l)$$

Given labels $\{y_i\}_{i=1}^n$, the gradient of ℓ_2^2-loss with respect to $\boldsymbol{k}_i$ and jth label, takes the form

$$\nabla\mathcal{L}_{\text{CNN,j}}(\boldsymbol{k}_i) = \sum_{l=1}^r \boldsymbol{o}_{i,l}(y_{CNN}(\mathbf{K},\boldsymbol{x}_j)-y_j)\sigma'(\boldsymbol{k}_i^T\boldsymbol{x}_{j,l})\boldsymbol{x}_{j,l}$$

We will show that a CNN can be transformed into a FNN combined with a subspace constraint. This will allow us to apply Theorem 3.5 to CNNs which will yield near optimal local convergence guarantees. We start by writing convolutional model as a fully-connected network.

Definition 4.1 (Convolutional weight matrix structure)**.** *Set $h = kr$. Given kernels $\{\boldsymbol{k}_i\}_{i=1}^k$, we construct the fully-connected weight matrix $\boldsymbol{W} = \mathbf{FC}(\mathbf{K}) \in \mathbb{R}^{h\times p}$ as follows: Representing $\{1,\dots,h\}$ as cartesian product of $\{1,\dots,k\}$ and $\{1,\dots,r\}$, define the $h = kr$ rows $\{\boldsymbol{w}_{i,j}\}_{(i,j)=(1,1)}^{(k,r)}$ of the weight matrix $\boldsymbol{W}$ as $\boldsymbol{w}_{i,j} = \mathbf{map}_j(\boldsymbol{k}_i)$ for $1 \le i \le r$ and $1 \le l \le k$. Finally let $\mathcal{C}$ be the space of all convolutional weight matrices defined as*

$$\mathcal{C} = \{\mathbf{FC}(\mathbf{K})\ \big|\ \{\boldsymbol{k}_i\}_{i=1}^k \in \mathbb{R}^b\}.$$

This model yields a matrix $\boldsymbol{W}$ that has double structure:

- Each row of $\boldsymbol{W}$ has at most $b = p/r$ nonzero entries.
- For fixed i, the weight vectors $\{\boldsymbol{w}_{i,l}\}_{l=1}^r$ are just shifted copies of each other.

This implies the total degrees of freedom is same as $\{\boldsymbol{k}_i\}_{i=1}^k$ and convolutional constraint $\mathcal{C}$ is a kb dimensional subspace.

Next, given $\boldsymbol{W} = \mathbf{FC}(\mathbf{K})$, observe the equality of the predictions i.e.

$$y_{\mathbf{FC}}(\boldsymbol{W},\boldsymbol{x}_j) = \sum_{i=1}^k\sum_{l=1}^r \boldsymbol{o}_{i,l}\sigma(\boldsymbol{w}_{i,l}^T\boldsymbol{x}) = y_{CNN}(\mathbf{K},\boldsymbol{x}_j)$$

Similarly for $\boldsymbol{W} = \mathbf{FC}(\mathbf{K})$, one can also show the equality of the CNN gradient and projected FNN gradient. Considering the following CNN and FNN gradient iterations

$$\hat{\mathbf{K}} = \mathbf{K} - \frac{\mu}{r}\nabla\mathcal{L}_{CNN}(\mathbf{K}),\ \hat{\boldsymbol{W}} = \mathcal{P}_{\mathcal{C}}(\boldsymbol{W} - \mu\nabla\mathcal{L}_{FC}(\boldsymbol{W})),$$

we have the equality $\mathbf{FC}(\hat{\mathbf{K}}) = \hat{\boldsymbol{W}}$. This relation yields the following corollary of Theorem 3.5.

Corollary 4.2. *Let $\{\boldsymbol{x}_i\}_{i=1}^n$ be i.i.d. $\mathcal{N}(0, \boldsymbol{I}_p)$ data points. Given k kernels $\mathbf{K}^\star = [\boldsymbol{k}_1^\star \ \dots \ \boldsymbol{k}_k^\star]^T$ and generate the labels*

$$y_j = y_{CNN}(\mathbf{K}^\star, \boldsymbol{x}_j)$$

Assume $\mathbf{FC}(\mathbf{K}^\star)$ is full row-rank and let $\Theta, \Omega, \bar{\upsilon}, q, \mu, \rho, P$ be same as in Theorem 3.5 defined with respect to the matrix $\mathbf{FC}(\mathbf{K}^\star)$. Suppose $n \geq \mathcal{O}\left((\sqrt{kb} + t)^2/\bar{\upsilon}^4\right)$ and consider the convolutional iteration

$$\hat{\mathbf{K}} = \mathbf{K} - \frac{\mu}{r}\nabla\mathcal{L}_{CNN}(\mathbf{K}).$$

Suppose the initial point $\mathbf{K} = [\boldsymbol{k}_1 \ \dots \ \boldsymbol{k}_k]^T$ satisfies $\|\mathbf{K} - \mathbf{K}^\star\|_F \leq \mathcal{O}\left(\frac{\|\mathbf{K}^\star\|_F}{q\sqrt{h\Omega\log p}\bar{\upsilon}^4}\right)$. Then, with $1 - P$ probability,

$$\|\hat{\mathbf{K}} - \mathbf{K}^\star\|_F^2 \leq \rho\|\mathbf{K} - \mathbf{K}^\star\|_F^2.$$

This corollary can be combined with the results of (Zhong et al., 2017a) to obtain a globally convergent CNN learning algorithm using $n \sim \mathcal{O}(\text{poly}(k, t, \log p))$ samples. In particular, for local convergence (Zhong et al., 2017a) needed $n \geq \mathcal{O}(p)$ samples whereas we show that there is no dependence on the data length p.

5. Numerical Results

To support our theoretical findings, we present numerical performance of sparsity and convolutional constraints for neural network training. We consider synthetic simulations where $\boldsymbol{o}$ is a vector of all ones and weight matrix $\boldsymbol{W}^\star \in \mathbb{R}^{h\times p}$ is sparse or corresponds to a CNN.

5.1. Sparsity Constraint

We generate $\boldsymbol{W}^\star$ matrices with exactly s nonzero entries at each row and nonzero pattern is distributed uniformly at random. Each entry of $\boldsymbol{W}^\star$ is $\mathcal{N}(0, \frac{p}{hs})$ to ensure $\mathbb{E}[\|\boldsymbol{W}^\star\boldsymbol{x}\|_{\ell_2}^2] = \|\boldsymbol{x}\|_{\ell_2}^2$. We set the learning rate to $\mu = 5$. We verified that smaller learning rate leads to similar results with slower convergence. We declare the estimate $\hat{\boldsymbol{W}}$ to be the output of PGD algorithm after 2000 iterations. We consider two sets of simulations using ReLU activations.

- **Good initialization:** We set $\boldsymbol{W}_0 = \boldsymbol{W}^\star + \boldsymbol{Z}$ where $\boldsymbol{Z}$ has i.i.d. $\mathcal{N}(0, \frac{1}{h})$ entries. Note that noise $\boldsymbol{Z}$ satisfies $\mathbb{E}[\|\boldsymbol{Z}\|_F^2] = \mathbb{E}[\|\boldsymbol{W}^\star\|_F^2]$.

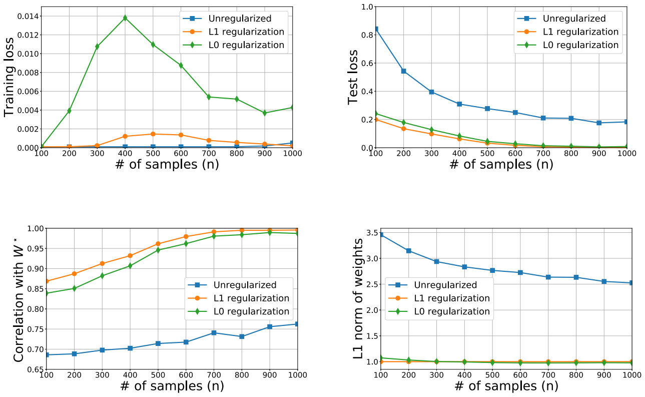

Figure 1: Experiments with good initialization $\boldsymbol{W}_0 = \boldsymbol{W}^\star + \boldsymbol{Z}$.

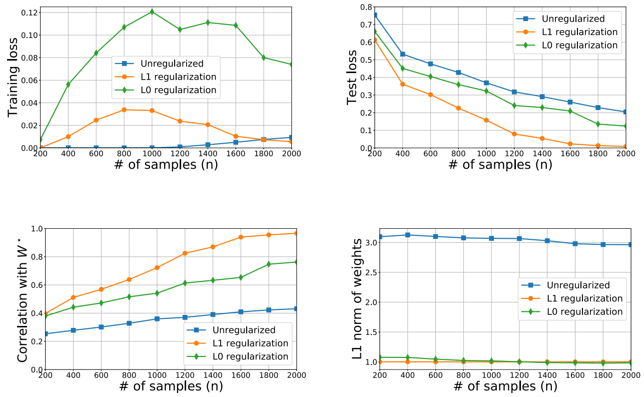

Figure 2: Experiments with random initialization $\boldsymbol{W}_0 = \boldsymbol{Z}$.

- **Random initialization:** We set $\boldsymbol{W}_0 = \boldsymbol{Z}$ where $\boldsymbol{Z}$ has i.i.d. $\mathcal{N}(0, \frac{1}{h})$ entries.

Each set of experiments consider three algorithms.

- **Unconstrained:** Only uses gradient descent.
- **ℓ_1-regularization:** Projects $\boldsymbol{W}$ to ℓ_1 ball scaled by the ℓ_1 norm of $\boldsymbol{W}^\star$.
- **ℓ_0-regularization:** Projects $\boldsymbol{W}$ to set of sh sparse matrices.

For our experiments, we picked $p = 80$, $h = 20$ and $s = p/10 = 8$. For training, we use n data points which varies from 100 to 1000. Test error is obtained by averaging $n_{\text{test}} = 1000$ independent data points. For each point in the plots, we averaged the outcomes of 20 random trials. The total degrees of freedom is the number of nonzeros equal to $sh = 160$. Our theorems imply good estimation via $\mathcal{O}(sh\log p/s)$ data points when initialization is sufficiently close. Figure 1 summarizes the outcome of the experiments with good initialization. Suppose y is the label and $\hat{y}$ is

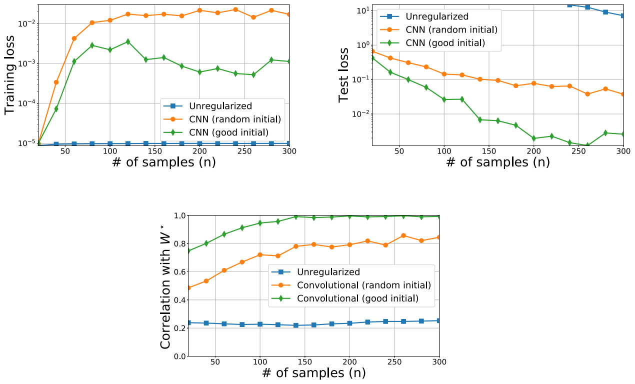

Figure 3: Experiments for convolutional constraint.

the prediction. We define the (normalized) test and train losses as the ratio of empirical variances that approximates the population $\frac{\textbf{var}[y-\hat{y}]}{\textbf{var}[y]}$. Centering (i.e. variance) is used to eliminate the contribution of trivial but large $\mathbb{E}[y]$ term due to nonnegative ReLU outputs. First, we observe that ℓ_1 is slightly better than ℓ_0 constraint however both approach ≈ 0 test loss when $n \geq 600$. Unregularized model has significant test error for all $100 \leq n \leq 1000$ while perfectly overfitting training set for all n values. We also consider the recovery of ground truth $\boldsymbol{W}^\star$. Since there is permutation invariance (permuting rows of $\boldsymbol{W}$ doesn't change the prediction), we define the correlation between $\boldsymbol{W}^\star$ and $\hat{\boldsymbol{W}}$ as follows,

$$\text{corr}(\boldsymbol{W}^\star, \hat{\boldsymbol{W}}) = \frac{1}{h}\sum_{i=1}^{h} \max_{1\leq j\leq h} \frac{\langle \boldsymbol{w}_i^\star, \hat{\boldsymbol{w}}_j\rangle}{\|\boldsymbol{w}_i^\star\|_{\ell_2}\|\hat{\boldsymbol{w}}_j\|_{\ell_2}}$$

where $\boldsymbol{w}_i$ is the ith row of $\boldsymbol{W}$. In words, each row of $\boldsymbol{W}^\star$ is matched to the highest correlated row from $\hat{\boldsymbol{W}}$ and correlations are averaged over h rows. Observe that, if $\hat{\boldsymbol{W}}$ and $\boldsymbol{W}^\star$ have matching permutations, $\text{corr}(\boldsymbol{W}^\star, \hat{\boldsymbol{W}}) = 1$. We see that $\text{corr}(\boldsymbol{W}^\star, \hat{\boldsymbol{W}}) \approx 1$ once $n \geq 600$ which is the moment test error hits 0.

Figure 2 summarizes the outcome of the experiments with random initialization. In this case, we vary n from 200 to 2000 but the rest of the setup is the same. We observe that unlike good initialization, ℓ_0 test error and $1 - \text{corr}(\boldsymbol{W}^\star, \hat{\boldsymbol{W}})$ does not hit 0 and ℓ_1 approaches 0 only at $n = 2000$. On the other hand, both metrics demonstrate the clear benefit of sparsity regularization. The performance gap between ℓ_1 and ℓ_0 is surprisingly high however it is consistent with Theorem 3.5 which only applies to convex regularizers. The performance difference between good and random initialization implies that initialization indeed plays a big role not only for finding the ground truth solution $\boldsymbol{W}^\star$ but also for achieving good test errors.

5.2. Convolutional Constraint

For the CNN experiment, we picked the following configuration. Problem parameters are input dimension $p = 81$, kernel width $b = 15$, stride $s = 6$, number of kernels $k = 4$ and learning rate $\mu = 1$. We did not use zero-padding hence $r = (p - b)/s + 1 = 12$. This implies $kr = 48$ hidden layers for fully-connected representation. The subspace dimension and degrees of freedom is $kb = 60$. We generate kernel entries with i.i.d. $\mathcal{N}(0, \frac{p}{hb})$ and the random matrix $\boldsymbol{Z}$ with i.i.d. $\mathcal{N}(0, \frac{p}{bk})$ entries. The noise variance is chosen higher to ensure $\mathbb{E}[\|\mathcal{P}_\mathcal{C}(\boldsymbol{Z})\|_F^2] = \mathbb{E}[\|\mathbf{FC}(\mathbf{K})\|_F^2]$ i.e. $\boldsymbol{Z}$ projected onto convolutional space has the same variance as the kernel matrix. We compare three models.

- Unconstrained model with $\boldsymbol{W}_0 = \boldsymbol{Z}$ initialization: Uses only gradient descent.
- CNN subspace constraint with $\boldsymbol{W}_0 = \boldsymbol{Z}$ initialization: Weights are shared via CNN backpropagation.
- CNN subspace constraint with with $\boldsymbol{W}_0 = \boldsymbol{W}^\star + \boldsymbol{Z}$ initialization.

Figures 1 illustrates the outcome of CNN experiments. Unconstrained model barely makes it into the test loss figure due to low signal-to-noise ratio. Focusing on CNN constraints, we observe that good initialization greatly helps and quickly achieves ≈ 0 test error. However random initialization has respectable test and correlation performance and gracefully improves as the data amount n increases.

6. Conclusions

In this work, we studied neural network regularization in order to reduce the storage cost and to improve generalization properties. We introduced covering dimension to quantify the impact of regularization and the richness of the constraint set. We proposed projected gradient descent algorithms to efficiently learn compact neural networks and showed that, if initialized reasonably close, PGD linearly converges to the ground truth while requiring minimal amount of training data. The sample complexity of the algorithm is governed by the covering dimension. We also specialized our results to convolutional neural nets and demonstrated how CNNs can be efficiently learned within our framework. Numerical experiments support the substantial benefit of regularization over training fully-connected neural nets.

Global convergence of the projected gradient descent appears to be a more challenging problem. In Section 5, we observed that gradient descent with random initialization can get stuck at local minima. For fully-connected networks, this is a well-known issue and the best known global convergence results are based on tensor initialization (Zhong et al., 2017b; Safran & Shamir, 2017; Janzamin et al., 2015). Interesting future directions include developing data-efficient initialization algorithms that can take advantage of the network priors (weight-sharing, sparsity, low-rank) and studying the properties of PGD from random initialization.

References

Aghasi, A., Abdi, A., Nguyen, N., and Romberg, J. Net-trim: Convex pruning of deep neural networks with performance guarantee. In *Advances in Neural Information Processing Systems*, pp. 3180–3189, 2017.

Arora, S., Bhaskara, A., Ge, R., and Ma, T. Provable bounds for learning some deep representations. In *International Conference on Machine Learning*, pp. 584–592, 2014.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pp. 6241–6250, 2017.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Brutzkus, A. and Globerson, A. Globally optimal gradient descent for a convnet with gaussian inputs. *arXiv preprint arXiv:1702.07966*, 2017.

Chandrasekaran, V., Recht, B., Parrilo, P. A., and Willsky, A. S. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., and Bengio, Y. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

Covington, P., Adams, J., and Sargin, E. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 191–198. ACM, 2016.

Cun, Y. L., Denker, J. S., and Solla, S. A. Optimal brain damage. In Touretzky, D. S. (ed.), *Advances in Neural Information Processing Systems 2*, pp. 598–605. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1-55860-100-7. URL http://dl.acm.org/citation.cfm?id=109230.109298.

Denton, E. L., Zaremba, W., Bruna, J., LeCun, Y., and Fergus, R. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in neural information processing systems*, pp. 1269–1277, 2014.

Dong, X., Chen, S., and Pan, S. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In *Advances in Neural Information Processing Systems*, pp. 4860–4874, 2017.

Du, S. S., Lee, J. D., and Tian, Y. When is a convolutional filter easy to learn? *arXiv preprint arXiv:1709.06129*, 2017a.

Du, S. S., Lee, J. D., Tian, Y., Poczos, B., and Singh, A. Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. *arXiv preprint arXiv:1712.00779*, 2017b.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Ge, R., Lee, J. D., and Ma, T. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017.

Graves, A., Mohamed, A.-r., and Hinton, G. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pp. 6645–6649. IEEE, 2013.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015a.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pp. 1135–1143, 2015b.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

Hassibi, B. and Stork, D. G. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in neural information processing systems*, pp. 164–171, 1993.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Janzamin, M., Sedghi, H., and Anandkumar, A. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

Jin, X., Yuan, X., Feng, J., and Yan, S. Training skinny deep neural networks with iterative hard thresholding methods. *arXiv preprint arXiv:1607.05423*, 2016.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323, 2013.

Kawaguchi, K., Kaelbling, L. P., and Bengio, Y. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Konstantinos, P., Davies, M., and Vandergheynst, P. Pac-bayesian margin bounds for convolutional neural networks-technical report. *arXiv preprint arXiv:1801.00171*, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Mei, S., Bai, Y., and Montanari, A. The landscape of empirical risk for non-convex losses. *arXiv preprint arXiv:1607.06534*, 2016.

Mendelson, S. Learning without concentration. *arXiv preprint arXiv:1401.0304*, 2014.

Negahban, S. and Wainwright, M. J. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13 (May):1665–1697, 2012.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Oymak, S. Learning compact neural networks with regularization. *arXiv preprint arXiv:1802.01223*, 2018.

Oymak, S. and Soltanolkotabi, M. End-to-end learning of a convolutional neural network via deep tensor decomposition. *arXiv preprint arXiv:1805.06523*, 2018.

Oymak, S., Recht, B., and Soltanolkotabi, M. Sharp time–data tradeoffs for linear inverse problems. *IEEE Transactions on Information Theory*, 2017.

Panigrahy, R., Rahimi, A., Sachdeva, S., and Zhang, Q. Convergence results for neural networks via electrodynamics. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 94. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.

Safran, I. and Shamir, O. Spurious local minima are common in two-layer relu neural networks. *arXiv preprint arXiv:1712.08968*, 2017.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Soltanolkotabi, M. Learning relus via gradient descent. *arXiv preprint arXiv:1705.04591*, 2017.

Soltanolkotabi, M., Javanmard, A., and Lee, J. D. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *arXiv preprint arXiv:1707.04926*, 2017.

Talagrand, M. *The generic chaining: upper and lower bounds of stochastic processes*. Springer Science & Business Media, 2006.

Tian, Y. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. *arXiv preprint arXiv:1703.00560*, 2017.

Wang, H., Wang, N., and Yeung, D.-Y. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1235–1244. ACM, 2015.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Zhong, K., Song, Z., and Dhillon, I. S. Learning non-overlapping convolutional neural networks with multiple kernels. *arXiv preprint arXiv:1711.03440*, 2017a.

Zhong, K., Song, Z., Jain, P., Bartlett, P. L., and Dhillon, I. S. Recovery guarantees for one-hidden-layer neural networks. *arXiv preprint arXiv:1706.03175*, 2017b.