

---

# Fast Convergence of Natural Gradient Descent for Overparameterized Neural Networks

---

Guodong Zhang<sup>1,2</sup>, James Martens<sup>3</sup>, Roger Grosse<sup>1,2</sup>

University of Toronto<sup>1</sup>, Vector Institute<sup>2</sup>, DeepMind<sup>3</sup>  
{gdzhang, rgrosse}@cs.toronto.edu, jamesmartens@google.com

## Abstract

Natural gradient descent has proven effective at mitigating the effects of pathological curvature in neural network optimization, but little is known theoretically about its convergence properties, especially for *nonlinear* networks. In this work, we analyze for the first time the speed of convergence of natural gradient descent on nonlinear neural networks with squared-error loss. We identify two conditions which guarantee efficient convergence from random initializations: (1) the Jacobian matrix (of network’s output for all training cases with respect to the parameters) has full row rank, and (2) the Jacobian matrix is stable for small perturbations around the initialization. For two-layer ReLU neural networks, we prove that these two conditions do in fact hold throughout the training, under the assumptions of nondegenerate inputs and overparameterization. We further extend our analysis to more general loss functions. Lastly, we show that K-FAC, an approximate natural gradient descent method, also converges to global minima under the same assumptions, and we give a bound on the rate of this convergence.

## 1 Introduction

Because training large neural networks is costly, there has been much interest in using second-order optimization to speed up training [Becker and LeCun, 1989, Martens, 2010, Martens and Grosse, 2015], and in particular natural gradient descent [Amari, 1998, 1997]. Recently, scalable approximations to natural gradient descent have shown practical success in a variety of tasks and architectures [Martens and Grosse, 2015, Grosse and Martens, 2016, Wu et al., 2017, Zhang et al., 2018a, Martens et al., 2018]. Natural gradient descent has an appealing interpretation as optimizing over a Riemannian manifold using an intrinsic distance metric; this implies the updates are invariant to transformations such as whitening [Ollivier, 2015, Luk and Grosse, 2018]. It is also closely connected to Gauss-Newton optimization, suggesting it should achieve fast convergence in certain settings [Pascanu and Bengio, 2013, Martens, 2014, Botev et al., 2017].

Does this intuition translate into faster convergence? Amari [1998] provided arguments in the affirmative, as long as the cost function is well approximated by a convex quadratic. However, it remains unknown whether natural gradient descent can optimize neural networks faster than gradient descent — a major gap in our understanding. The problem is that the optimization of neural networks is both nonconvex and non-smooth, making it difficult to prove nontrivial convergence bounds. In general, finding a global minimum of a general non-convex function is an NP-complete problem, and neural network training in particular is NP-complete [Blum and Rivest, 1992].

However, in the past two years, researchers have finally gained substantial traction in understanding the dynamics of gradient-based optimization of neural networks. Theoretically, it has been shown that gradient descent starting from a random initialization is able to find a global minimum if the network is wide enough [Li and Liang, 2018, Du et al., 2018b,a, Zou et al., 2018, Allen-Zhu et al., 2018, Oymak and Soltanolkotabi, 2019]. The key technique of those works is to show that neural

networks become well-behaved if they are largely overparameterized in the sense that the number of hidden units is polynomially large in the size of the training data. However, most of these works have focused on standard gradient descent, leaving open the question of whether similar statements can be made about other optimizers.

Most convergence analysis of natural gradient descent has focused on simple convex quadratic objectives (e.g. [Martens, 2014]). Very recently, the convergence properties of NGD were studied in the context of linear networks [Bernacchia et al., 2018]. While the linearity assumption simplifies the analysis of training dynamics [Saxe et al., 2013], linear networks are severely limited in terms of their expressivity, and it’s not clear which conclusions will generalize from linear to nonlinear networks.

In this work, we analyze natural gradient descent for *nonlinear networks*. We give two simple and generic conditions on the Jacobian matrix which guarantee efficient convergence to a global minimum. We then apply this analysis to a particular distribution over two-layer ReLU networks which has recently been used to analyze the convergence of gradient descent [Li and Liang, 2018, Du et al., 2018a, Oymak and Soltanolkotabi, 2019]. We show that for sufficiently high network width, NGD will converge to the global minimum. We give bounds on the convergence rate of two-layer ReLU networks that are much better than the analogous bounds that have been proven for gradient descent [Du et al., 2018b, Wu et al., 2019, Oymak and Soltanolkotabi, 2019], while allowing for much higher learning rates. Moreover, in the limit of infinite width, and assuming a squared error loss, we show that NGD converges in just *one iteration*. The main contributions of our work are summarized as follows:

- We provide the first convergence result for natural gradient descent in training randomly-initialized overparameterized neural networks where the number of hidden units is polynomially larger than the number of training samples. We show that natural gradient descent gives an  $\mathcal{O}(\lambda_{\min}(\mathbf{G}^\infty))$  improvement in convergence rate given the same learning rate as gradient descent, where  $\mathbf{G}^\infty$  is a Gram matrix that depends on the data.
- We show that natural gradient enables us to use a much larger step size, resulting in an even faster convergence rate. Specifically, the maximal step size of natural gradient descent is  $\mathcal{O}(1)$  for (polynomially) wide networks.
- We show that K-FAC [Martens and Grosse, 2015], an approximate natural gradient descent method, also converges to global minima with linear rate, although this result requires a higher level of overparameterization compared to GD and exact NGD.
- We analyze the generalization properties of NGD, showing that the improved convergence rates *provably* don’t come at the expense of worse generalization.

## 2 Related Works

Recently, there have been many works studying the optimization problem in deep learning, i.e., why in practice many neural network architectures reliably converge to global minima (zero training error). One popular way to attack this problem is to analyze the underlying loss surface [Hardt and Ma, 2016, Kawaguchi, 2016, Kawaguchi and Bengio, 2018, Nguyen and Hein, 2017, Soudry and Carmon, 2016]. The main argument of those works is that there are no bad local minima. It has been proven that gradient descent can find global minima [Ge et al., 2015, Lee et al., 2016] if the loss surface satisfies: (1) all local minima are global and (2) all saddle points are strict in the sense that there exists at least one negative curvature direction. Unfortunately, most of those works rely on unrealistic assumptions (e.g., linear activations [Hardt and Ma, 2016, Kawaguchi, 2016]) and cannot generalize to practical neural networks. Moreover, Yun et al. [2018] shows that small nonlinearity in shallow networks can create bad local minima.

Another way to understand the optimization of neural networks is to directly analyze the optimization dynamics. Our work also falls within this category. However, most work in this direction focuses on gradient descent. Bartlett et al., Arora et al. [2019a] studied the optimization trajectory of deep linear networks and showed that gradient descent can find global minima under some assumptions. Previously, the dynamics of linear networks have also been studied by Saxe et al. [2013], Advani and Saxe [2017]. For nonlinear neural networks, a series of papers [Tian, 2017, Brutzkus and Globerson, 2017, Du et al., 2017, Li and Yuan, 2017, Zhang et al., 2018b] studied a specific class of shallow two-layer neural networks together with strong assumptions on input distribution as well as realizability of labels, proving global convergence of gradient descent. Very recently, there are

some works proving global convergence of gradient descent [Li and Liang, 2018, Du et al., 2018b,a, Allen-Zhu et al., 2018, Zou et al., 2018, Gao et al., 2019] or adaptive gradient methods [Wu et al., 2019] on overparameterized neural networks. More specifically, Li and Liang [2018], Allen-Zhu et al. [2018], Zou et al. [2018] analyzed the dynamics of weights and showed that the gradient cannot be small if the objective value is large. On the other hand, Du et al. [2018b,a], Wu et al. [2019] studied the dynamics of the outputs of neural networks, where the convergence properties are captured by a Gram matrix. Our work is very similar to Du et al. [2018b], Wu et al. [2019]. We note that these papers all require the step size to be sufficiently small to guarantee the global convergence, leading to slow convergence.

To our knowledge, there is only one paper [Bernacchia et al., 2018] studying the global convergence of natural gradient for neural networks. However, Bernacchia et al. [2018] only studied deep linear networks with infinitesimal step size and squared error loss functions. In this sense, our work is the first one proving global convergence of natural gradient descent on nonlinear networks.

There have been many attempts to understand the generalization properties of neural networks since Zhang et al. [2016]’s seminal paper. Researchers have proposed norm-based generalization bounds [Neyshabur et al., 2015, 2017, Bartlett and Mendelson, 2002, Bartlett et al., 2017, Golowich et al., 2017], compression bounds [Arora et al., 2018] and PAC-Bayes bounds [Dziugaite and Roy, 2017, 2018, Zou et al., 2018]. Recently, overparameterization of neural networks together with good initialization has been believed to be one key factor of good generalization. Neyshabur et al. [2019] empirically showed that wide neural networks stay close to the initialization, thus leading to good generalization. Theoretically, researchers did prove that overparameterization as well as linear convergence jointly restrict the weights to be close to the initialization [Du et al., 2018b,a, Allen-Zhu et al., 2018, Zou et al., 2018, Arora et al., 2019b]. The most closely related paper is Arora et al. [2019b], which shows that the optimization and generalization phenomenon can be explained by a Gram matrix. The main difference is that our analysis is based on natural gradient descent, which converges faster and provably generalizes as well as gradient descent.

Concurrently and independently, Cai et al. [2019] showed that natural gradient descent (they call it Gram-Gauss-Newton) enjoys quadratic convergence rate guarantee for overparameterized networks on regression problems. Additionally, they showed that it is much cheaper to precondition the gradient in the output space when the number of data points is much smaller than the number of parameters.

### 3 Convergence Analysis of Natural Gradient Descent

We begin our convergence analysis of natural gradient descent – under appropriate conditions – for the neural network optimization problem. Formally, we consider a generic neural network  $f(\boldsymbol{\theta}, \mathbf{x})$  with a single output and squared error loss  $\ell(u, y) = \frac{1}{2}(u - y)^2$  for simplicity<sup>1</sup>, where  $\boldsymbol{\theta} \in \mathbb{R}^m$  denotes all parameters of the network (i.e. weights and biases). Given a training dataset  $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ , we want to minimize the following loss function:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \ell(f(\boldsymbol{\theta}, \mathbf{x}_i), y_i) = \frac{1}{2n} \sum_{i=1}^n (f(\boldsymbol{\theta}, \mathbf{x}_i) - y_i)^2. \quad (1)$$

One main focus of this paper is to analyze the following procedure:

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \eta \mathbf{F}(\boldsymbol{\theta}(k))^{-1} \frac{\partial \mathcal{L}(\boldsymbol{\theta}(k))}{\partial \boldsymbol{\theta}(k)}, \quad (2)$$

where  $\eta > 0$  is the step size, and  $\mathbf{F}$  is the Fisher information matrix associated with the network’s predictive distribution over  $y$  (which is implied by its loss function and is  $\mathcal{N}(f(\boldsymbol{\theta}, \mathbf{x}_i), 1)$  for the squared error loss) and the dataset’s distribution over  $\mathbf{x}$ .

As shown by Martens [2014], the Fisher  $\mathbf{F}$  is equivalent to the generalized Gauss-Newton matrix, defined as  $\mathbb{E}_{\mathbf{x}_i} [\mathbf{J}_i^\top \mathbf{H}_\ell \mathbf{J}_i]$  if the predictive distribution is in the exponential family, such as categorical distribution (for classification) or Gaussian distribution (for regression).  $\mathbf{J}_i$  is the Jacobian matrix of  $\mathbf{u}_i$  with respect to the parameters  $\boldsymbol{\theta}$  and  $\mathbf{H}_\ell$  is the Hessian of the loss  $\ell(\mathbf{u}, \mathbf{y})$  with respect to the network prediction  $\mathbf{u}$  (which is  $\mathbf{I}$  in our setting). Therefore, with the squared error loss, the Fisher

<sup>1</sup>It is easy to extend to multi-output networks and other loss functions, here we focus on single-output and quadratic just for notational simplicity.

matrix can be compactly written as  $\mathbf{F} = \mathbb{E} [\mathbf{J}_i^\top \mathbf{J}_i] = \frac{1}{n} \mathbf{J}^\top \mathbf{J}$  (which coincides with classical Gauss-Newton matrix), where  $\mathbf{J} = [\mathbf{J}_1^\top, \dots, \mathbf{J}_n^\top]^\top$  is the Jacobian matrix for the whole dataset. In practice, when the number of parameters  $m$  is larger than number of samples  $n$  we have, the Fisher information matrix  $\mathbf{F} = \frac{1}{n} \mathbf{J}^\top \mathbf{J}$  is surely singular. In that case, we take the generalized inverse [Bernacchia et al., 2018]  $\mathbf{F}^\dagger = n \mathbf{J}^\top \mathbf{G}^{-1} \mathbf{G}^{-1} \mathbf{J}$  with  $\mathbf{G} = \mathbf{J} \mathbf{J}^\top$ , which gives the following update rule:

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \eta \mathbf{J}^\top (\mathbf{J} \mathbf{J}^\top)^{-1} (\mathbf{u} - \mathbf{y}), \quad (3)$$

where  $\mathbf{u} = [\mathbf{u}_1, \dots, \mathbf{u}_n]^\top = [f(\boldsymbol{\theta}, \mathbf{x}_1), \dots, f(\boldsymbol{\theta}, \mathbf{x}_n)]^\top$  and  $\mathbf{y} = [y_1, \dots, y_n]^\top$ .

We now introduce two conditions on the network  $f_\theta$  that suffice for proving the global convergence of NGD to a minimizer which achieves zero training loss (and is therefore a global minimizer). To motivate these two conditions we make the following observations. First, the global minimizer is characterized by the condition that the gradient in the output space is zero for each case (i.e.  $\nabla_{\mathbf{u}} \mathcal{L}(\boldsymbol{\theta}) = \mathbf{0}$ ). Meanwhile, local minima are characterized by the condition that the gradient with respect to the parameters  $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$  is zero. Thus, one way to avoid finding local minima that aren't global is to ensure that the parameter gradient is zero if and only if the output space gradient (for each case) is zero. It's not hard to see that this property holds as long as  $\mathbf{G}$  remains non-singular throughout optimization (or equivalently that  $\mathbf{J}$  always has full row rank). The following two conditions ensure that this happens, by first requiring that this property hold at initialization time, and second that  $\mathbf{J}$  changes slowly enough that it remains true in a big enough neighborhood around  $\boldsymbol{\theta}(0)$ .

**Condition 1** (Full row rank of Jacobian matrix). *The Jacobian matrix  $\mathbf{J}(0)$  at the initialization has full row rank, or equivalently, the Gram matrix  $\mathbf{G}(0) = \mathbf{J}(0)\mathbf{J}(0)^\top$  is positive definite.*

**Remark 1.** *Condition 1 implies that  $m \leq n$ , which means the Fisher information matrix is singular and we have to use the generalized inverse except in the case where  $m = n$ .*

**Condition 2** (Stable Jacobian). *There exists  $0 \leq C < \frac{1}{2}$  such that for all parameters  $\boldsymbol{\theta}$  that satisfy  $\|\boldsymbol{\theta} - \boldsymbol{\theta}(0)\|_2 \leq \frac{3\|\mathbf{y} - \mathbf{u}(0)\|_2}{\sqrt{\lambda_{\min}(\mathbf{G}(0))}}$ , we have*

$$\|\mathbf{J}(\boldsymbol{\theta}) - \mathbf{J}(0)\|_2 \leq \frac{C}{3} \sqrt{\lambda_{\min}(\mathbf{G}(0))}. \quad (4)$$

This condition shares the same spirit with the Lipschitz smoothness assumption in classical optimization theory. It implies (with small  $C$ ) that the network is close to a linearized network [Lee et al., 2019] around the initialization and therefore natural gradient descent update is close to the gradient descent update in the output space. Along with Condition 1, we have the following theorem.

**Theorem 1** (Natural gradient descent). *Let Condition 1 and 2 hold. Suppose we optimize with NGD using a step size  $\eta \leq \frac{1-2C}{(1+C)^2}$ . Then for  $k = 0, 1, 2, \dots$  we have*

$$\|\mathbf{u}(k) - \mathbf{y}\|_2^2 \leq (1 - \eta)^k \|\mathbf{u}(0) - \mathbf{y}\|_2^2. \quad (5)$$

To be noted,  $\|\mathbf{u}(k) - \mathbf{y}\|_2^2$  is the squared error loss up to a constant. Due to space constraints we only give a short sketch of the proof here. The full proof is given in Appendix B.

**Proof Sketch.** *Our proof relies on the following insights. First, if the Jacobian matrix has full row rank, this guarantees linear convergence for infinitesimal step size. The linear convergence property restricts the parameters to be close to the initialization, which implies the Jacobian matrix is always full row rank throughout the training, and therefore natural gradient descent with infinitesimal step size converges to global minima. Furthermore, given the network is close to a linearized network (since the Jacobian matrix is stable with respect to small perturbations around the initialization), we are able to extend the proof to discrete time with a large step size.*

In summary, we prove that NGD exhibits linear convergence to the global minimizer of the neural network training problem, under Conditions 1 and 2. We believe our arguments in this section are general (i.e., architecture-agnostic), and can serve as a recipe for proving global convergence of natural gradient descent in other settings.

### 3.1 Other Loss Functions

We note that our analysis can be easily extended to more general loss function class. Here, we take the class of functions that are  $\mu$ -strongly convex with  $L$ -Lipschitz gradients as an example. Note that

strongly convexity is a very mild assumption since we can always add  $L_2$  regularization to make the convex loss strongly convex. Therefore, this function class includes regularized cross-entropy loss (which is typically used in classification) and squared error (for regression). For this type of loss, we need a strong version of Condition 2.

**Condition 3** (Stable Jacobian). *There exists  $0 \leq C < \frac{1}{1+\kappa}$  such that for all parameters  $\theta$  that satisfy  $\|\theta - \theta(0)\|_2 \leq \frac{3(1+\kappa)\|\mathbf{y}-\mathbf{u}(0)\|_2}{2\sqrt{\lambda_{\min}(\mathbf{G}(0))}}$  where  $\kappa = \frac{L}{\mu}$*

$$\|\mathbf{J}(\theta) - \mathbf{J}(0)\|_2 \leq \frac{C}{3} \sqrt{\lambda_{\min}(\mathbf{G}(0))}. \quad (6)$$

**Theorem 2.** *Under Condition 1 and 3, but with  $\mu$ -strongly convex loss function  $\ell(\cdot, \cdot)$  with  $L$ -Lipschitz gradient ( $\kappa = \frac{L}{\mu}$ ), and we set the step size  $\eta \leq \frac{2}{\mu+L} \frac{1-(1+\kappa)C}{(1+C)^2}$ , then we have for  $k = 0, 1, 2, \dots$*

$$\|\mathbf{u}(k) - \mathbf{y}\|_2^2 \leq \left(1 - \frac{2\eta\mu L}{\mu + L}\right)^k \|\mathbf{u}(0) - \mathbf{y}\|_2^2. \quad (7)$$

The key step of proving Theorem 2 is to show if  $m$  is large enough, then natural gradient descent is approximately gradient descent in the output space. Thus the results can be easily derived according to standard bounds for convex optimization. Due to the page limit, we defer the proof to the Appendix C.

**Remark 2.** *In Theorem 2, the convergence rate depends on the condition number  $\kappa = \frac{L}{\mu}$ , which can be removed if we take into the curvature information of the loss function. In other words, we expect that the bound has no dependency on  $\kappa$  if we use the Fisher matrix rather than the classical Gauss-Newton (assuming Euclidean metric in the output space [Luk and Grosse, 2018]) in Theorem 2.*

## 4 Optimizing Overparameterized Neural Networks

In Section 3, we analyzed the convergence properties of natural gradient descent, under the abstract Conditions 1 and 2. In this section, we make our analysis concrete by applying it to a specific type of overparameterized network (with a certain random initialization). We show that Conditions 1 and 2 hold with high probability. We therefore establish that NGD exhibits linear convergence to a global minimizer for such networks.

### 4.1 Notation

We let  $[m] = \{1, 2, \dots, m\}$ . We use  $\otimes, \odot$  to denote the Kronecker and Hadamard products. And we use  $*$  and  $\star$  to denote row-wise and column-wise Khatri-Rao products, respectively. For a matrix  $\mathbf{A}$ , we use  $\mathbf{A}_{ij}$  to denote its  $(i, j)$ -th entry. We use  $\|\cdot\|_2$  to denote the Euclidean norm of a vector or spectral norm of a matrix and  $\|\cdot\|_{\mathbb{F}}$  to denote the Frobenius norm of a matrix. We use  $\lambda_{\max}(\mathbf{A})$  and  $\lambda_{\min}(\mathbf{A})$  to denote the largest and smallest eigenvalue of a square matrix, and  $\sigma_{\max}(\mathbf{A})$  and  $\sigma_{\min}(\mathbf{A})$  to denote the largest and smallest singular value of a (possibly non-square) matrix. For a positive definite matrix  $\mathbf{A}$ , we use  $\kappa_{\mathbf{A}}$  to denote its condition number, i.e.,  $\lambda_{\max}(\mathbf{A})/\lambda_{\min}(\mathbf{A})$ . We also use  $\langle \cdot, \cdot \rangle$  to denote the standard inner product between two vectors. Given an event  $E$ , we use  $\mathbb{I}\{E\}$  to denote the indicator function for  $E$ .

### 4.2 Problem Setup

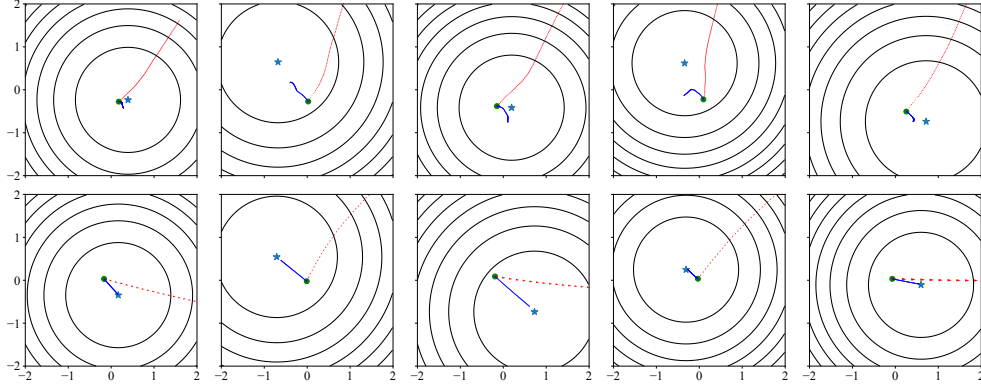
Formally, we consider a neural network of the following form:

$$f(\mathbf{w}, \mathbf{a}, \mathbf{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^m a_r \phi(\mathbf{w}_r^\top \mathbf{x}), \quad (8)$$

where  $\mathbf{x} \in \mathbb{R}^d$  is the input,  $\mathbf{w} = [\mathbf{w}_1^\top, \dots, \mathbf{w}_r^\top]^\top \in \mathbb{R}^{md}$  is the weight matrix (formed into a vector) of the first layer,  $a_r \in \mathbb{R}$  is the output weight of hidden unit  $r$  and  $\phi(\cdot)$  is the ReLU activation function (acting entry-wise for vector arguments). For  $r \in [m]$ , we initialize the weights of first layer  $\mathbf{w}_r \sim \mathcal{N}(\mathbf{0}, \nu^2 \mathbf{I})$  and output weight  $a_r \sim \text{unif}[\{-1, +1\}]$ .

Following Du et al. [2018b], Wu et al. [2019], we make the following assumption on the data.





**Figure 1:** Visualization of **natural gradient** update and **gradient descent** update in the output space (for a randomly initialized network). We take two classes (4 and 9) from MNIST [LeCun et al., 1998] and generate the targets (denoted as star in the figure) by  $f(x) = x - 0.5 + 0.3 \times \mathcal{N}(0, \mathbf{I})$  where  $x \in \mathbb{R}^2$  is one-hot target. We get natural gradient update by running 100 iterations of conjugate gradient [Martens, 2010]. **The first row:** a MLP with two hidden layers and 100 hidden units in each layer. **The second row:** a MLP with two hidden layers and 6000 hidden units in each layer. In both cases, ReLU activation function was used. We interpolate the step size from 0 to 1. For the over-parameterized network (in the second row), natural gradient descent (implemented by conjugate gradient) matches output space gradient well.

**Assumption 1.** For all  $i$ ,  $\|\mathbf{x}_i\|_2 = 1$  and  $|y_i| = \mathcal{O}(1)$ . For any  $i \neq j$ ,  $\mathbf{x}_i \not\parallel \mathbf{x}_j$ .

This very mild condition simply requires the inputs and outputs have standardized norms, and that different input vectors are distinguishable from each other. Datasets that do not satisfy this condition can be made to do so via simple pre-processing.

Following Du et al. [2018b], Oymak and Soltanolkotabi [2019], Wu et al. [2019], we only optimize the weights of the first layer<sup>2</sup>, i.e.,  $\boldsymbol{\theta} = \mathbf{w}$ . Therefore, natural gradient descent can be simplified to

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \eta \mathbf{J}^\top (\mathbf{J}\mathbf{J}^\top)^{-1} (\mathbf{u} - \mathbf{y}). \quad (9)$$

Though this is only a shallow fully connected neural network, the objective is still non-smooth and non-convex [Du et al., 2018b] due to the use of ReLU activation function. We further note that this two-layer network model has been useful in understanding the optimization and generalization of deep neural networks [Xie et al., 2016, Li and Liang, 2018, Du et al., 2018b, Arora et al., 2019b, Wu et al., 2019], and some results have been extended to multi-layer networks [Du et al., 2018a].

Following Du et al. [2018b], Wu et al. [2019], we define the *limiting Gram matrix* as follows:

**Definition 1** (Limiting Gram Matrix). The limiting Gram matrix  $\mathbf{G}^\infty \in \mathbb{R}^{n \times n}$  is defined as follows. For  $(i, j)$ -entry, we have

$$\mathbf{G}_{ij}^\infty = \mathbb{E}_{\mathbf{w} \sim \mathcal{N}(0, \nu^2 \mathbf{I})} [\mathbf{x}_i^\top \mathbf{x}_j \mathbb{I} \{ \mathbf{w}^\top \mathbf{x}_i \geq 0, \mathbf{w}^\top \mathbf{x}_j \geq 0 \}] = \mathbf{x}_i^\top \mathbf{x}_j \frac{\pi - \arccos(\mathbf{x}_i^\top \mathbf{x}_j)}{2\pi}. \quad (10)$$

This matrix coincides with neural tangent kernel [Jacot et al., 2018] for ReLU activation function. As shown by Du et al. [2018b], this matrix is positive definite and we define its smallest eigenvalue  $\lambda_0 \triangleq \lambda_{\min}(\mathbf{G}^\infty) > 0$ . In the same way, we can define its finite version  $\mathbf{G}(t) = \mathbf{J}(t)\mathbf{J}(t)^\top$  with  $(i, j)$ -entry  $\mathbf{G}_{ij}(t) = \frac{1}{m} \mathbf{x}_i^\top \mathbf{x}_j \sum_{r \in [m]} \mathbb{I} \{ \mathbf{w}_r(t)^\top \mathbf{x}_i \geq 0, \mathbf{w}_r(t)^\top \mathbf{x}_j \geq 0 \}$ .

### 4.3 Exact Natural Gradient Descent

In this subsection, we present our result for this setting. The main difficulty is to show that Conditions 1 and 2 hold. Here we state our main result.

**Theorem 3** (Natural Gradient Descent for overparameterized Networks). Under Assumption 1, if we i.i.d initialize  $\mathbf{w}_r \sim \mathcal{N}(\mathbf{0}, \nu^2 \mathbf{I})$ ,  $a_r \sim \text{unif}\{-1, +1\}$  for  $r \in [m]$ , we set the number of hidden

<sup>2</sup>We fix the second layer just for simplicity. Based on the same analysis, one can also prove global convergence for jointly training both layers.

nodes  $m = \Omega\left(\frac{n^4}{\nu^2 \lambda_0^4 \delta^3}\right)$ , and the step size  $\eta = \mathcal{O}(1)$ , then with probability at least  $1 - \delta$  over the random initialization we have for  $k = 0, 1, 2, \dots$

$$\|\mathbf{u}(k) - \mathbf{y}\|_2^2 \leq (1 - \eta)^k \|\mathbf{u}(0) - \mathbf{y}\|_2^2. \quad (11)$$

Even though the objective is non-convex and non-smooth, natural gradient descent with a constant step size enjoys a linear convergence rate. For large enough  $m$ , we show that the learning rate can be chosen up to 1, so NGD can provably converge within  $\mathcal{O}(1)$  steps. Compared to analogous bounds for gradient descent [Du et al., 2018a, Oymak and Soltanolkotabi, 2019, Wu et al., 2019], we improve the maximum allowable learning rate from  $\mathcal{O}(1/n)$  to  $\mathcal{O}(1)$  and also get rid of the dependency on  $\lambda_0$ . Overall, NGD (Theorem 3) gives an  $\mathcal{O}(\lambda_0/n)$  improvement over gradient descent.

Our strategy to prove this result will be to show that for the given choice of random initialization, Condition 1 and 2 hold with high probability. For proving Condition 1 hold, we used matrix concentration inequalities. For Condition 2, we show that  $\|\mathbf{J} - \mathbf{J}(0)\|_2 = \mathcal{O}(m^{-1/6})$ , which implies the Jacobian is stable for wide networks. For detailed proof, we refer the reader to the Appendix D.1.

#### 4.4 Approximate Natural Gradient Descent with K-FAC

Exact natural gradient descent is quite expensive in terms of computation or memory. In training deep neural networks, K-FAC [Martens and Grosse, 2015] has been a powerful optimizer for leveraging curvature information while retaining tractable computation. The K-FAC update rule for the two-layer ReLU network is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \eta \underbrace{[(\mathbf{X}^\top \mathbf{X})^{-1} \otimes (\mathbf{S}(k)^\top \mathbf{S}(k))^{-1}]}_{\mathbf{F}_{\text{K-FAC}}^{-1}} \mathbf{J}(k)^\top (\mathbf{u}(k) - \mathbf{y}). \quad (12)$$

where  $\mathbf{X} \in \mathbb{R}^{n \times d}$  denotes the matrix formed from the  $n$  input vectors (i.e.  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^\top$ ), and  $\mathbf{S} = [\phi'(\mathbf{X}\mathbf{w}_1), \dots, \phi'(\mathbf{X}\mathbf{w}_m)] \in \mathbb{R}^{n \times m}$  is the matrix of pre-activation derivatives. Under the same argument as the Gram matrix  $\mathbf{G}^\infty$ , we get that  $\mathbf{S}^\infty \mathbf{S}^{\infty \top}$  is strictly positive definite with smallest eigenvalue  $\lambda_{\mathbf{S}}$  (see Appendix D.3 for detailed proof).

We show that for sufficiently wide networks, K-FAC does converge linearly to a global minimizer. We further show, with a particular transformation on the input data, K-FAC does match the optimization performance of exact natural gradient for two-layer ReLU networks. Here we state the main result.

**Theorem 4 (K-FAC).** *Under the same assumptions as in Theorem 3, plus the additional assumption that  $\text{rank}(\mathbf{X}) = d$ , if we set the number of hidden units  $m = \mathcal{O}\left(\frac{n^4}{\nu^2 \lambda_{\mathbf{S}}^4 \kappa_{\mathbf{X}^\top \mathbf{X}}^4 \delta^3}\right)$  and step size  $\eta = \mathcal{O}(\lambda_{\min}(\mathbf{X}^\top \mathbf{X}))$ , then with probability at least  $1 - \delta$  over the random initialization, we have for  $k = 0, 1, 2, \dots$*

$$\|\mathbf{u}(k) - \mathbf{y}\|_2^2 \leq \left(1 - \frac{\eta}{\lambda_{\max}(\mathbf{X}^\top \mathbf{X})}\right)^k \|\mathbf{u}(0) - \mathbf{y}\|_2^2. \quad (13)$$

The key step in proving Theorem 4 is to show

$$\mathbf{u}(k+1) - \mathbf{u}(k) \approx [(\mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top) \odot \mathbf{I}] (\mathbf{y} - \mathbf{u}(k)). \quad (14)$$

**Remark 3.** *The convergence rate of K-FAC is captured by the condition number of the matrix  $\mathbf{X}^\top \mathbf{X}$ , as opposed to gradient descent [Du et al., 2018b, Oymak and Soltanolkotabi, 2019], for which the convergence rate is determined by the condition number of the Gram matrix  $\mathbf{G}$ .*

**Remark 4.** *The dependence of the convergence rate on  $\kappa_{\mathbf{X}^\top \mathbf{X}}$  in Theorem 4 may seem paradoxical, as K-FAC is invariant to invertible linear transformations of the data (including those that would change  $\kappa_{\mathbf{X}^\top \mathbf{X}}$ ). But we note that said transformations would also make the norms of the input vectors non-uniform, thus violating Assumption 1 in a way that isn't repairable. Interestingly, there exists an invertible linear transformation which, if applied to the input vectors and followed by normalization, produces vectors that simultaneously satisfy Assumption 1 and the condition  $\kappa_{\mathbf{X}^\top \mathbf{X}} = 1$  (thus improving the bound in Theorem 4 substantially). See Appendix A for details. Notably, K-FAC is not invariant to such pre-processing, as the normalization step is a nonlinear operation.*

To quantify the degree of overparameterization (which is a function of the network width  $m$ ) required to achieve global convergence under our analysis, we must estimate  $\lambda_S$ . To this end, we observe that  $\mathbf{G} = \mathbf{X}\mathbf{X}^\top \odot \mathbf{S}\mathbf{S}^\top$ , and then apply the following lemma:

**Lemma 1.** [Schur [1911]] For two positive definite matrices  $\mathbf{A}$  and  $\mathbf{B}$ , we have

$$\begin{aligned}\lambda_{\max}(\mathbf{A} \odot \mathbf{B}) &\leq \max_i \mathbf{A}_{ii} \lambda_{\max}(\mathbf{B}) \\ \lambda_{\min}(\mathbf{A} \odot \mathbf{B}) &\geq \min_i \mathbf{A}_{ii} \lambda_{\min}(\mathbf{B})\end{aligned}\tag{15}$$

The diagonal entries of  $\mathbf{X}\mathbf{X}^\top$  are all 1 since the inputs are normalized. Therefore, we have  $\lambda_0 \geq \lambda_S$  according to Lemma 1, and hence K-FAC requires a slightly higher degree of overparameterization than exact NGD under our analysis.

#### 4.5 Bounding $\lambda_0$

As pointed out by Allen-Zhu et al. [2018], it is unclear if  $1/\lambda_0$  is small or even polynomial. Here, we bound  $\lambda$  using matrix concentration inequalities and harmonic analysis. To leverage harmonic analysis, we have to assume the data  $\mathbf{x}_i$  are drawn i.i.d. from the unit sphere<sup>3</sup>.

**Theorem 5.** Under this assumption on the training data, with probability  $1 - n \exp(-n^\beta/4)$ ,

$$\lambda_0 \triangleq \lambda_{\min}(\mathbf{G}^\infty) \geq n^\beta/2, \text{ where } \beta \in (0, 0.5)\tag{16}$$

Basically, Theorem 5 says that the Gram matrix  $\mathbf{G}^\infty$  should have high chance of having large smallest eigenvalue if the training data are uniformly distributed. Intuitively, we would expect the smallest eigenvalue to be very small if all  $\mathbf{x}_i$  are similar to each other. Therefore, some notion of diversity of the training inputs is needed. We conjecture that the smallest eigenvalue would still be large if the data are  $\delta$ -separable (i.e.,  $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \geq \delta$  for any pair  $i, j \in [n]$ ), an assumption adopted by Li and Liang [2018], Allen-Zhu et al. [2018], Zou et al. [2018].

## 5 Generalization analysis

It is often speculated that NGD or other preconditioned gradient descent methods (e.g., Adam) perform worse than gradient descent in terms of generalization [Wilson et al., 2017]. In this section, we show that NGD achieves the same generalization bounds which have been proved for GD, at least for two-layer ReLU networks.

Consider a loss function  $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ . The expected risk over the data distribution  $\mathcal{D}$  and the empirical risk over a training set  $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$  are defined as

$$\mathcal{L}_{\mathcal{D}}(f) = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\ell(f(\mathbf{x}), y)] \quad \text{and} \quad \mathcal{L}_{\mathcal{S}}(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i), y_i)\tag{17}$$

It has been shown [Neyshabur et al., 2019] that the Rademacher complexity [Bartlett and Mendelson, 2002] for two-layer ReLU networks depends on  $\|\mathbf{w} - \mathbf{w}(0)\|_2$ . By the standard Rademacher complexity generalization bound, we have the following bound (see Appendix E.1 for proof):

**Theorem 6.** Given a target error parameter  $\epsilon > 0$  and failure probability  $\delta \in (0, 1)$ . Suppose  $\nu = \mathcal{O}(\epsilon\sqrt{\lambda_0\delta})$  and  $m \geq \nu^{-2} \text{poly}(n, \lambda_0^{-1}, \delta^{-1}, \epsilon^{-1})$ . For any 1-Lipschitz loss function, with probability at least  $1 - \delta$  over random initialization and training samples, the two-layer neural network  $f(\mathbf{w}, \mathbf{a})$  trained by NGD for  $k \geq \Omega\left(\frac{1}{\eta} \log \frac{1}{\epsilon\delta}\right)$  iterations has expected loss  $\mathcal{L}_{\mathcal{D}}(f(\mathbf{w}, \mathbf{a})) = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\ell(f(\mathbf{w}, \mathbf{a}, \mathbf{x}), y)]$  bounded as:

$$\mathcal{L}_{\mathcal{D}}(f(\mathbf{w}, \mathbf{a})) \leq \sqrt{\frac{2\mathbf{y}^\top (\mathbf{G}^\infty)^{-1} \mathbf{y}}{n}} + 3\sqrt{\frac{\log(6/\delta)}{2n}} + \epsilon\tag{18}$$

which matches the bound for gradient descent in Arora et al. [2019b]. For detailed proof, we refer the reader to the Appendix E.1.

<sup>3</sup>This assumption is not too stringent since the inputs are already normalized. Moreover, we can relax the assumption of unit sphere input to separable input, which is used in Li and Liang [2018], Allen-Zhu et al. [2018], Zou et al. [2018]. See Oymak and Soltanolkotabi [2019] (Theorem I.1) for more details.



## 6 Conclusion

We’ve analyzed for the first time the rate of convergence to a global optimum for (both exact and approximate) natural gradient descent on nonlinear neural networks. Particularly, we identified two conditions which guarantee the global convergence, i.e., the Jacobian matrix with respect to the parameters has full row rank and stable for perturbations around the initialization. Based on these insights, we improved the convergence rate of gradient descent by a factor of  $\mathcal{O}(\lambda_0/n)$  on two-layer ReLU networks by using natural gradient descent. Beyond that, we also showed that the improved convergence rates don’t come at the expense of worse generalization.

## Acknowledgements

We thank Jeffrey Z. HaoChen, Shengyang Sun and Mufan Li for helpful discussion. RG acknowledges support from the CIFAR Canadian AI Chairs program and the Ontario MRIS Early Researcher Award.

## References

- Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.
- Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.
- Shun-ichi Amari. Neural learning in structured parameter spaces-natural riemannian gradient. In *Advances in neural information processing systems*, pages 127–133, 1997.
- Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.
- Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*, 2018.
- Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations*, 2019a. URL <https://openreview.net/forum?id=SkMQg3C5K7>.
- Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019b.
- Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.
- Peter L Bartlett, David P Helmbold, and Philip M Long. Gradient descent with identity initialization efficiently learns positive-definite linear transformations by deep residual networks. *Neural computation*, pages 1–26.
- Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.
- S. Becker and Y. LeCun. Improving the convergence of backpropagation learning with second order methods. In *Proceedings of the 1988 Connectionist Models Summer School*, 1989.
- Alberto Bernacchia, Mate Lengyel, and Guillaume Hennequin. Exact natural gradient in deep linear networks and its application to the nonlinear case. In *Advances in Neural Information Processing Systems*, pages 5945–5954, 2018.
- Avrim Blum and Ronald L. Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5:117–127, 1992.
- Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton optimisation for deep learning. In *International Conference on Machine Learning*, 2017.

- Stephen Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2004.
- Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 605–614. JMLR. org, 2017.
- Tianle Cai, Ruiqi Gao, Jikai Hou, Siyu Chen, Dong Wang, Di He, Zhihua Zhang, and Liwei Wang. A gram-gauss-newton method learning overparameterized deep neural networks for regression problems. *arXiv preprint arXiv:1905.11675*, 2019.
- Simon S Du, Jason D Lee, and Yuandong Tian. When is a convolutional filter easy to learn? *arXiv preprint arXiv:1709.06129*, 2017.
- Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018a.
- Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018b.
- Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.
- Gintare Karolina Dziugaite and Daniel M Roy. Data-dependent pac-bayes priors via differential privacy. In *Advances in Neural Information Processing Systems*, pages 8440–8450, 2018.
- Jürgen Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65(4):612–625, 2002.
- Weihao Gao, Ashok Makkuva, Sewoong Oh, and Pramod Viswanath. Learning one-hidden-layer neural networks under general input distributions. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1950–1959, 2019.
- Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.
- Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017.
- Roger Grosse and James Martens. A Kronecker-factored approximate Fisher matrix for convolution layers. In *International Conference on Machine Learning*, 2016.
- Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.
- Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8580–8589, 2018.
- Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in neural information processing systems*, pages 586–594, 2016.
- Kenji Kawaguchi and Yoshua Bengio. Depth with nonlinearity creates no bad local minima in resnets. *arXiv preprint arXiv:1810.09038*, 2018.
- Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural network of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.
- Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent converges to minimizers. *arXiv preprint arXiv:1602.04915*, 2016.

- Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pages 8168–8177, 2018.
- Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.
- Kevin Luk and Roger Grosse. A coordinate-free construction of scalable natural gradient. *arXiv preprint arXiv:1808.1340*, 2018.
- James Martens. Deep learning via hessian-free optimization. In *ICML*, volume 27, pages 735–742, 2010.
- James Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.
- James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417, 2015.
- James Martens, Jimmy Ba, and Matt Johnson. Kronecker-factored curvature approximations for recurrent neural networks. In *International Conference on Learning Representations*, 2018.
- Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. 2018.
- Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.
- Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.
- Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=BygfgHAcYX>.
- Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2603–2612. JMLR. org, 2017.
- Yann Ollivier. Riemannian metrics for neural networks I: Feedforward networks. *Information and Inference*, 4:108–153, 2015.
- Samet Oymak and Mahdi Soltanolkotabi. Towards moderate overparameterization: global convergence guarantees for training shallow neural networks. *arXiv preprint arXiv:1902.04674*, 2019.
- Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.
- Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.
- Jssai Schur. Bemerkungen zur theorie der beschränkten bilinearformen mit unendlich vielen veränderlichen. *Journal für die reine und Angewandte Mathematik*, 140:1–28, 1911.
- Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.
- Yuandong Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3404–3413. JMLR. org, 2017.
- Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

- Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pages 4148–4158, 2017.
- Xiaoxia Wu, Simon S Du, and Rachel Ward. Global convergence of adaptive gradient methods for an over-parameterized neural network. *arXiv preprint arXiv:1902.07111*, 2019.
- Yuhuai Wu, Elman Mansimov, Shun Liao, Roger Grosse, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. In *Neural Information Processing Systems*, 2017.
- Bo Xie, Yingyu Liang, and Le Song. Diverse neural network learns true target functions. *arXiv preprint arXiv:1611.03131*, 2016.
- Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small nonlinearities in activation functions create bad local minima in neural networks. *arXiv preprint arXiv:1802.03487*, 2018.
- Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.
- Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pages 5847–5856, 2018a.
- Xiao Zhang, Yaodong Yu, Lingxiao Wang, and Quanquan Gu. Learning one-hidden-layer relu networks via gradient descent. *arXiv preprint arXiv:1806.07808*, 2018b.
- Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018.