The Loss Surface Of Deep Linear Networks Viewed Through The Algebraic Geometry Lens

Dhagash Mehta, Tianran Chen, Tingting Tang and Jonathan D. Hauenstein

Abstract—By using the viewpoint of modern computational algebraic geometry, we explore properties of the optimization landscapes of deep linear neural network models. After providing clarification on the various definitions of "flat" minima, we show that the geometrically flat minima, which are merely artifacts of residual continuous symmetries of the deep linear networks, can be straightforwardly removed by a generalized L_2-regularization. Then, we establish upper bounds on the number of isolated stationary points of these networks with the help of algebraic geometry. Combining these upper bounds with a method in numerical algebraic geometry, we find *all* stationary points for modest depth and matrix size. We demonstrate that, in the presence of the non-zero regularization, deep linear networks can indeed possess local minima which are not global minima. Finally, we show that even though the number of stationary points increases as the number of neurons (regularization parameters) increases (decreases), higher index saddles are surprisingly rare.

Index Terms—Deep linear network, global optimization, regularization, numerical algebraic geometry

1 Introduction

Advancement in both computational algorithms and computer hardware has led a surge in applied and theoretical research activities for deep learning techniques. Though the applied side of the research has been remarkably successful with applications in such areas as computer vision, natural language processing, machine translation, object recognition, speech and audio recognition, stock market analysis, bioinformatics, and drug analysis [1], [2], a thorough theoretical understanding of the techniques have not yet been achieved.

One of the urgent theoretical issues that is of particular interest to the present work is the highly non-convex nature of the underlying optimization problems that the techniques bring with them: the cost function (also called the loss function) of a typical deep learning task, such as the mean squared error between the observed data and predicted data from the deep network, is known to have numerous local minima. Finding a minimum which possesses a desired characteristic is usually a daunting task, especially for a high-dimensional problem, and most of the times it turns out to be an NP hard problem [3]. Nonetheless, in practice, for a typical deep learning task, a reasonably good minimization algorithm, such as a stochastic gradient descent (SGD) based method, converges to a minimum that performs well. This observation, along with several empirical results [4], [5], [6], [7], [8], [9], [10], [11], [12], has led to the belief that there is no bad minima in the loss functions of *deep* networks. In [13] (cf., [14], [15]), the loss function of a typical dense feed-forward neural network with rectified linear (ReLu) units was approximated by the Hamiltonion of a physics model called the spherical p-spin model and analyzed using random matrix theory and statistical physics techniques. It concluded that for this approximate model, the number of minima and saddle points at which the value of the loss function is beyond certain threshold vanishes as the number of hidden layers increases (cf. [16]) supporting the "no bad minima" scenario, though the assumptions made to bring the deep network to the spin glass model were unrealistic.

The specific characteristics of the minima that numerical minimization algorithms may be looking for cab play a crucial role in determining if and why the algorithm finds them so efficiently [17]. In the literature, the distance of a minimum from the global minimum has been the defining characteristic of the "goodness" of minima, i.e., if the difference between the loss function at the local minimum and that at the global minimum is within certain threshold, then the minimum is good enough for the task. There are recent examples of artificial neural networks with such suboptimal minima for deep nonlinear networks [18], [19], [20], [21] (and [22] for neural networks without hidden layers). In [23], good and bad minima are distinguished based not only in terms of the the performance of the network on the training data but also on the testing data, and it is empirically shown that the volume of basin of attraction of good minima dominates over that of bad minima (cf., [17], [24] for discussions on good and bad minima). In [17], the shape and size of the decision boundaries as well as size of the effective network (measured in terms of number of non-zero weights) are shown to provide further metrics of goodness of minima.

Another scenario that was proposed in [18], [25], and further confirmed in [26], is that the loss function of a deep network is typically proliferated by the large number of saddle points (and degenerate saddles [27]) compared to minima. Gradient based optimization algorithms may get stuck at a saddle point rather than a minimum which slows

- *D. Mehta is with The Vanguard Group, Valley Forge, PA, USA. E-mail: dhagashbmehta@gmail.com*
- *T. Chen is with the Department of Mathematics, Auburn University at Montgomery, Montgomery, AL, USA. E-mail: ti@nranchen.org.*
- *T. Tang is with the Department of Mathematics and Statistics, San Diego State University, Imperial Valley Campus, Calexico, CA, USA. E-mail: ttang2@sdsu.edu.*
- *J.D. Hauenstein is with Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN, USA. E-mail: hauenstein@nd.edu.*

Manuscript received XXXX;

down the learning. This is typical for the types of nonlinear multivariate cost functions one encounters in physics and chemistry [28], [29], [30], [31], [32], [33], [34], [35]. Several ways to escape from saddle points provided no singular saddle points exist have been developed [25], [36], [37], and also in the presence of singular saddle points in certain specific cases [12], [38], [39]. It is argued that the probability of converging to a saddle point is small enough to be ignored empirically and theoretically. In addition, regularization is often used to avoid saddle points and to improve global optimal convergence.

In [4], a detailed mathematical analysis of simpler models called deep linear networks was performed. Since then, the model has become one of the ideal testing grounds for ideas in artificial neural networks and deep learning [40]. Here, we use the framework of a deep linear network to study the effect of regularization on the minima by posing the problem as an algebraic geometry problem. Below, we first briefly describe the formulation of the model and then describe some previous results.

1.1 Deep Linear Networks

A deep linear network is an artificial neural network with multiple hidden layers with each neuron having a linear activation function. It is the linearity of the activation functions that separates deep linear networks from the deep nonlinear networks used in practice in which each neuron has a nonlinear (or, at least, piecewise linear) activation function. The mean squared error for the deep linear networks with the usual L_2-regularization is defined to be [14], [40]

$$L(W) = \bar{\mathcal{L}}(W) + \frac{\lambda}{2}\sum_{i=1}^{H+1}\|W_i\|_2^2, \tag{1}$$

with

$$\bar{\mathcal{L}}(W) = \frac{1}{2}\sum_{i=1}^{m}\left\|(W_{H+1}W_H\cdots W_1X)_{\cdot,i} - Y_{\cdot,i}\right\|_2^2, \tag{2}$$

where $\|.\|$ is the vector norm, W_i is the weight matrix for the i^{th} layer with hidden layers from $i = 1,\ldots,H$ and output layer $H+1$, and $\lambda \geq 0$ is the regularization parameter. For m data points in the training set, d_x input dimensions, and d_y output dimensions, the dimensions of X and Y are $d_x \times m$ and $d_y \times m$, respectively. Then, with d_i hidden neurons in the i^{th} hidden layer, matrix multiplication yields that $W_1 \in \mathbb{R}^{d_1\times d_x}, W_2 \in \mathbb{R}^{d_2\times d_1},\ldots,W_{H+1}\times\mathbb{R}^{d_y\times d_H}$. We also denote $k = \min(d_1,\ldots,d_H)$, i.e., the number of neurons in the hidden layer with the smallest width. The number of weights, or variables, is $n = d_xd_1d_2\cdots d_Hd_y$.

The simplicity of the deep linear network yields that it can approximate functions which are linear in X and Y though nonlinear in weights, whereas the real-world data may also possess nonlinearity in X. However, these networks contain most of the basic ingredients of a typical deep nonlinear networks. Due to the network architecture, the loss function of the deep linear networks (Eqs. (1) and (2)) are still non-convex and non-trivial to analyze in a general setting. Understanding the loss surfaces of the deep linear networks also may enhance our understanding of the same for deep nonlinear networks.

1.2 Earlier Works on Loss Surfaces of Deep Linear Networks

Almost all the existing results for deep linear networks are for networks without regularization, i.e., for $\lambda = 0$. For a deep linear network with $\lambda = 0$ and $H = 1$ under the assumptions that (1) $XX^\top$ and $XY^\top$ are invertible matrices, (2) $\Sigma = YX^\top(XX^\top)^{-1}XY^\top$ has d_y distinct eigenvalues, (3) $d_x = d_y$, i.e., an autoencoder, and (4) $k < \min\{d_x, d_y\}$, it was shown in [4] that:

1) $\bar{\mathcal{L}}(W)$ is convex if either W_1 or W_2 are fixed, and the entries of the other vary;
2) every local minimum is a global minimum.

Moreover, [4] also conjectured the following upon dropping the $H = 1$ condition but retaining the other assumptions:

1) $\bar{\mathcal{L}}(W)$ is convex if the entries of one W_i vary while the others are fixed;
2) every local minimum is a global minimum.

This conjecture was proven in more general settings of deep linear networks in [14], [41], [42], for deep linear complex-valued autoencoders with one hidden layer [43], as well as for deep linear residual networks [21], [44]. Additionally, [45] provides several necessary and sufficient conditions on global optimality based on rank conditions on the W_i matrices for deep linear networks.

In [46] (cf. [47]), analytical forms of the stationary points (including minima) characterizing the values of the loss function were presented for deep linear networks as well as for certain limited cases of unregularized deep nonlinear networks. The aforementioned necessary and sufficient conditions for global optimality were also reformulated with the help of the analytical form of the critical points.

Layer-wise training of deep linear networks was investigated from the dynamical systems point of view in [7] (see also [48]) and was concluded that the learning speed can remain finite even in the $H \to \infty$ limit for a special class of initial conditions on the weights, likely due to having no local minima present in the landscape.

The $\lambda > 0$ case, i.e., with regularization, is considered in [49] by modeling a linear networks (though, not a deep linear network) with L_2-regularization term as a continuous time optimal control problem. The problem of characterizing the critical points of the deep linear networks was reduced to solving a finite-dimensional nonlinear matrix-valued equation. Continuous time is essentially a surrogate index for layers and the final weight matrix was assumed to be square. It was shown that for a special case of the model, even for small amount of regularization, saddle points emerge. Moreover, [50, Prop. 2.2] shows that there are no bad minima for 2-layer deep linear networks under certain constraints.

1.3 Our Contribution

The main conceptual contribution in this paper is to identify solving the gradients of deep networks as a problem in computational algebraic geometry, e.g., see [51], [52]. We review the existing literature related to the optimization landscape and put our algebraic geometry point of view into perspective. The other key contributions from this viewpoint are summarized as follows:

1) We clarify various definitions of *flat* minima, and distinguish the geometric definition of flat minima from the other definitions (Sec. 3.1). We then show their existence in the unregularized landscapes of deep linear networks (Remark 1, Sec. 3.1),

2) We prove that a straightforward extension of L_2-regularization can guarantee to remove all flat minima: these flat minima are only an artifact of the underlying residual symmetries of the equations and can be removed using, for example, the generalized L_2-regularization (Theorems 1 and 2, Sec. 3.2).

3) We take up a novel question on deep learning loss surfaces: how many isolated stationary (also called critical) points and, more specifically, minima are there in a typical deep learning loss surface? With the help of algebraic geometry, we provide the first results in this direction on upper bounds on the number of stationary points for deep linear networks (Propositions 3 and 4, Sec. 4). Obviously, these upper bounds provide strict upper bounds on the number of local minima.

4) We design a numerical algebraic geometric method which guarantees to find *all* stationary points of the deep linear networks which is applicable to networks of modest size (Sec. 5.1; Tables 1 and 2, Sec. 5.3.1). With all stationary points at hand, we explicitly show that the model exhibits local minima which are *not* global minima for the regularized case (Table 3 and Figure 5, Sec. 5.3.4 and 5.3.5).

5) We show that the number of stationary points increases with the number of neurons and hidden layers, and decreases when the regularization parameter is increased (Figures 1 and 2, Sec. 5.3.2).

6) We show that stationary points of higher index are surprisingly rare, if any, in the landscape even though the total number of stationary points may be a plenty, at least for the cases at hand (Figures 3 and 4, Sec. 5.3.3).

7) As a simple real-world application, we apply our approach to train a deep linear network to learn the Boston housing data and confirm the above results using the proposed methods and pointing out some differences (Sec. 5.3.6).

The remainder of the paper is organized as follows. Section 2 provides a brief introduction to algebraic geometry and discusses a relation between algebraic geometry and deep linear networks. We also put our approach in perspective with respect to other attempts to apply algebraic geometry methods to machine learning. Section 3 shows how flat stationary points of unregularized gradient equations can be removed using a generalized regularization. Section 4 provides upper bounds on number of stationary points of the gradient equations based on algebraic geometry. Section 5 introduces and applies homotopy continuation to provide results for modest size systems. Section 6 discusses our findings in more details.

2 Algebraic Geometric Interpretation of Deep Linear Networks

In this section, we show that solving the gradient equations of deep linear networks can be viewed as a problem in computational algebraic geometry [51], [52], and briefly introduce algebraic geometry terminologies while distinguishing our algebraic geometry interpretation of the problem with previous attempts.

An abstract relation between statistical learning methods and algebraic geometry has been extensively investigated [53]. Machine learning methods have been used to improve computational algebraic geometry methods such as in computing cylindrical algebraic decomposition [54] and to find roots of certain polynomials [55], [56], [57]. Neural networks have also been shown to effectively learn data whose target function is a polynomial [10] (see also [58], [59]).

In the present paper, we explore the loss landscape by interpreting solving the gradient systems of deep learning as an algebraic geometry problem. The algebraic geometry interpretation allows us to investigate the gradient equations for both the regularized and unregularized cases and for arbitrary data and size of all the matrices. Though we focus on deep linear networks in this paper, the deep learning problem can also be cast as an algebraic geometry problem in the presence of all the conventional activation functions.

2.1 The Gradient Equations are Algebraic Equations

The critical points of the objective function L in (1) are points at which all partial derivatives are equal to zero, i.e., satisfy the gradient equations $\nabla L = \mathbf{0}$. These gradient equations form a system of equations which is nonlinear in its variables, i.e., the entries of W_i. This system is naturally an algebraic system since each equation is polynomial in the variables.

Consider the matrix $W = W_{H+1} \cdots W_1$ and define $U_i^\top = \prod_{j=i+1}^{H+1} W_j^\top$ and $V_i^\top = \prod_{j=1}^{i-1} W_j^\top$. Then, $\frac{\partial L}{\partial W_i}$ is a matrix whose (j,k) entry is the partial derivative of L with respect to the (j,k) entry of W_i. In particular, one has

$$\frac{\partial L}{\partial W_i} = U_i^\top \left(W \left(\sum_{k=1}^{m} \mathbf{x}_k \mathbf{x}_k^\top \right) - \left(\sum_{k=1}^{m} \mathbf{y}_k \mathbf{x}_k^\top \right) \right) V_i^\top + \lambda W_i. \quad (3)$$

Therefore, each partial derivative is polynomial in the entries of $W_1, \ldots, W_{H+1}$. Hence, studying the critical points of L is equivalent to studying the solution set to a system of polynomial equations, namely, the gradient equations, which is the central question in the field of *algebraic geometry*.

2.2 A Brief Introduction to Algebraic Geometry

In the context of deep linear networks, critical points are *real* solutions to the gradient equations. It is common in algebraic geometry [51], [52] to simplify the problem by computing all solutions over the complex numbers since the complex numbers form an algebraically closed field, i.e., every univariate polynomial equation with complex coefficients has at least one complex solution.

An *algebraic set* is the solution set of a collection of polynomial equations. That is, the algebraic set associated to the polynomial system $f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, where $\mathbf{x} = (x_1, \ldots, x_n)$ in $\mathbb{C}^n$ is

$$V(f) = \{\mathbf{x} \in \mathbb{C}^n \mid f_i(\mathbf{x}) = 0, i = 1, \ldots, m\}.$$

The real points in $V(f)$ is simply $V_{\mathbb{R}}(f) = V(f) \cap \mathbb{R}^n$. An algebraic set A is *reducible* if there exists nonempty algebraic sets $B_1, B_2 \subsetneq A$ such that $A = B_1 \cup B_2$, otherwise, A is said to be *irreducible*. Every algebraic set can be presented uniquely, up to reordering, as a finite union of irreducible algebraic sets yielding its *irreducible decomposition*.

Each irreducible algebraic set A has a well-defined *dimension*. Every irreducible algebraic set A of dimension 0 is a singleton, i.e., of the form $A = \{p\}$ in which case p is called an *isolated solution* to the corresponding polynomial system. A positive-dimensional irreducible algebraic set consists of infinitely many points, e.g., a curve has dimension 1 and a surface has dimension 2. In the context of the gradient equations, isolated solutions correspond with *isolated stationary points* and positive-dimensional algebraic sets consist of *flat stationary points*.

2.3 Difference Between Complexifying the Gradient Equations and Complex Loss Functions

We note that neural networks with complex-valued weights (and complex-valued inputs and outputs) have been studied in the past [60], [61], [62], [63], [64] and have gained renewed interest in deep learning [65], [66], [67], [68], [69] for use in simultaneously modeling phase and amplitude data. In particular, back-propagation for complex-valued neural networks was developed in [63]. In [70], it was shown that the XOR data which cannot be solved with a single real-valued neuron in the hidden layer, but can be solved with a complex-valued network. Such complex-valued neural networks were shown to have better generalization characteristics [71] and faster learning [72] in addition to biological motivations [65].

The aforementioned formulation of deep complex networks consider complex weights, inputs, and outputs, and hence the corresponding loss function is also complex-valued. On the other hand, in the present paper, we start from the conventional real-valued weights, inputs, and outputs, with the loss function also being real-valued. Then, we merely complexify the gradient equations in that we assume weights living in the complex space with inputs and outputs living in the real space. In other words, the former is fundamentally a complex-valued set up whereas, in the latter case, the weights are complexified only for computational analysis purposes.

3 Flat Stationary Points and Regularization

In this section, we briefly review the existing literature on flat minima in deep learning and propose our approach to remove them.

3.1 Flat Critical Sets

In [73], [74], an algorithm to search for some (but not provably all) *acceptable* (i.e., almost) flat minima, which are large connected regions of minima at which the training error was below a threshold, was proposed. Such acceptable flat minima correspond to weights many of which may be specified with low precision (hence, with fewer bits of information). In these references, it was also argued that these minima also correspond to low complexity networks.

In [75], it was empirically shown that SGD based methods tend to converge to sharp (flat) minima with large (small) batch sizes. In [76], [77], it was argued that higher (lower) ratio between learning rate and batch size pushes the SGD towards flatter (sharper) minima, and that the flatter minima generalized better than sharper minima. In [78], an entropy-SGD was proposed that actively bias the optimization towards flat minima of specific widths (cf. [79], [80], [81], [82]). However, later on, in [83], the above definitions of flatness of minima were formalized and it was then argued that deep networks do not necessarily generalize better when they converge to "flat" minima (as defined above) than sharp minima because one can reparametrize the loss function that correspond to equivalent models but possessing arbitrarily sharp minima.

In the current paper, we are interested in exactly flat saddles and minima, i.e., the components of the stationary points on which the loss function is precisely constant, whose existence is well-known since the works of [84], [85], [86], [87] (see [88] for a review). Such degenerate regions, sometimes referred to as *neuromanifolds*, are quite common [89] in various loss landscapes, not just artificial neural networks, due to various symmetries [85], [90], [91] of the corresponding loss functions. At such solutions, the Fischer information matrix tends to be singular and traditional gradient descent algorithms are known to slow down.

To be sure, the Hessian matrix of the loss function can be singular at either isolated singular solutions (i.e., multiple roots) as well as on a non-isolated degenerate solution region. In [27], it was shown using numerical experiments for modest size deep neural networks that the available SGD based optimization routine converged to degenerate saddle points at which the Hessian matrix not only has many positive and negative eigenvalues but also multiple zero eigenvalues. Moreover, they showed that the number of zero eigenvalues increases with increasing depth. It was argued that for good training, it is enough that deep neural network models converge at degenerate saddle points as long as the training error is low. Whereas, in [92], by computing the eigenvalues of the Hessian of deep nonlinear networks after training as well as at random points in the configuration space, it was shown that a vast number of eigenvalues were zero. Hence, most of the directions in the weight space of these networks are flat leading to no change in the loss function.

In [93], it was shown that though small and large batch gradient descent appeared to converge to seemingly different minima, a straight line interpolation between the two did not contain any barrier, implying that the two regions may be in the same basin of attraction. In the present paper, we make a distinction between isolated singular solutions and flat minima. We also carry forward the distinction made in [93] between almost flat minima within which the loss function is almost constant and flat minima within which the loss function is precisely constant. The former should be referred to as *wide* minima.

In terms of algebraic geometry, a stationary point is flat if it is not an isolated solution of the gradient equations. Hence, each flat stationary point lies on a positive-dimensional component. For the purpose of this paper, we focus on complex positive-dimensional stationary points which may include real positive-dimensional solutions. In the next section, we device a method to remove positive-dimensional stationary solutions which removes complex and real positive-dimensional stationary points.

We present a few examples to show some explicit results. The first example arises in [39].

Example 1. *The gradient of* $f(x,y,z) = 2xy+2xz-2x-y-z$ *is* $\nabla(f) = \{2y+2z-2, 2x-1, 2x-1\}$. *The set of stationary points which satisfy* $\nabla(f)=0$ *is the line defined by* $x-1/2 = y+z-1 = 0$, *i.e., in the complex space, the solution has dimension 1. At every point on this line,* $f(x,y,z) = -1$.

Example 2. *For* $H=1$, $m=5$, $d_x = d_y = 2$, *and* $d_1 = 1$ *with* $\lambda = 0$, *we consider the data matrices*

$$X = \begin{bmatrix} 7 & -8 & 3 & -5 & 10 \\ -7 & 10 & 6 & -2 & 6 \end{bmatrix},\ Y = \begin{bmatrix} 9 & 9 & -8 & 1 & 10 \\ 10 & 3 & -8 & 9 & 10 \end{bmatrix}.$$

The stationary points of Eqs. (3) *consist of three irreducible components: a point that is an isolated saddle, a curve consisting of flat saddle points, and a curve consisting of flat minima. The point is* $W_1 = 0 \in \mathbb{R}^{1\times 2}$ *and* $W_2 = 0 \in \mathbb{R}^{2\times 1}$. *The flat saddle points and flat minima have the form*

$$W_1 = \alpha \cdot \widehat{W_1} \ \text{ and } \ W_2 = \alpha^{-1} \cdot \widehat{W_2}$$

for any $\alpha \neq 0$. *For example, the flat saddle points approximately have*

$$\widehat{W_1} = \begin{bmatrix} 1 & 9.6330 \end{bmatrix} \ \text{ and } \ \widehat{W_2} = \begin{bmatrix} 0.0206 \\ -0.0180 \end{bmatrix}$$

while the flat minima approximately have

$$\widehat{W_1} = \begin{bmatrix} 1 & 0.0696 \end{bmatrix} \ \text{ and } \ \widehat{W_2} = \begin{bmatrix} 0.2664 \\ 0.3045 \end{bmatrix}.$$

Remark 1. *This example generalizes to all critical points in the unregularized case, i.e.,* $\lambda = 0$. *That is, if* $(W_1, \ldots, W_{H+1})$ *is a critical point, then so is* $(A_1 W_1, A_2 W_2 A_1^{-1}, \ldots, W_{H+1} A_H^{-1})$. *Hence, if there is a critical point with some* $W_i \neq 0$, *then there are always flat critical points in the unregularized case.*

The traditional L_2-regularization with single parameter λ as in (1) is not necessarily enough to remove the flat stationary points as shown in the following.

Example 3. *For* $H=1$, $m=3$, *and* $d_x = d_y = d_1 = 2$, *let*

$$X = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \ \text{ and } \ Y = \begin{bmatrix} 1 & 2 & 3 \\ 1 & -3 & 2 \end{bmatrix}.$$

For any $\lambda \geq 0$ *and* $a \geq 0$, *the following is a family of flat critical points:*

$$W_1 = \begin{bmatrix} a & a \\ \gamma(a,\lambda) & \gamma(a,\lambda) \end{bmatrix}, W_2 = \sqrt{\tfrac{2}{197}} \begin{bmatrix} 14a & 14\gamma(a,\lambda) \\ a & \gamma(a,\lambda) \end{bmatrix}$$

where

$$\gamma(a,\lambda) = \sqrt{\sqrt{394}/56 - a^2 - \lambda/28}.$$

If $0 < \lambda < \sqrt{\frac{197}{2}}$, *then this component consists of flat minima which are real for* $0 \leq a \leq \sqrt{7(\sqrt{197/2} - \lambda)}/14$.

3.2 Regularization of flat critical sets

We begin the discussion of removing flat minima from the landscapes of loss functions by pointing out two observations. First, in [20], [17], where the goal of the study was to numerically investigate the loss landscape of a deep nonlinear neural network with one hidden layer with the $\tanh$ activation function, it was noted that the constant zero eigenvalues disappeared as soon as the L_2-regularization term was non-zero. Here, all the weights including the bias weights were regularized [17]. However, this observation may not directly apply in general because the continuous symmetries present in more complex systems may depend on the network architectures, activation functions, data, etc. Second, in [94], a spin glass model called the XY model was found to exhibit residual continuous symmetries and a generalized regularization term was used to remove them.

As outlined above, the existence of flat or degenerate critical set is a very common phenomenon in the general study of deep linear and nonlinear networks. At any point in a flat critical set of $\bar{\mathcal{L}}$, the Hessian matrix of $\bar{\mathcal{L}}$ has at least one zero eigenvalue. Such a zero eigenvalue of the Hessian matrix signifies a certain degree of freedom in the weight matrices. That is, there are directions in which weights infinitesimally change without violating the gradient equations.

From a computational point of view, flat critical sets introduce many unnecessary difficulties. For example, a simple solver based on Newton's iterations may encounter numerical instabilities near a flat critical set. From a purely theoretical point of view, flat critical sets indicate the training data set and the network structure are not sufficient to determine the optimal configuration of the weights. In this section, we outline a "regularization" technique that could perturb the loss function $\bar{\mathcal{L}}(W)$ ever so slightly so that all the critical points become nondegenerate (isolated) critical points. That is, such a perturbation would remove the flatness from all critical points.

Recall that for a smooth function $f : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{v} \in \mathbb{R}^n$ is said to be a *regular value* if, for each $\mathbf{x} \in \mathbb{R}^n$ such that $f(\mathbf{x}) = \mathbf{v}$, the Jacobian matrix Df is nonsingular at $\mathbf{x}$. **Sard's Theorem** [95] states that almost all $\mathbf{v} \in \mathbb{R}^n$ are regular values (in the sense of Lebesgue measure). This result can be generalized into a stronger result on parametric systems that fits our current situation. Let $f(\mathbf{a}, \mathbf{x}) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ be a smooth function. **Generalized Sard's Theorem** [95], [96] states that if $\mathbf{0}$ is a regular value for f, then for almost all $\mathbf{a} \in \mathbb{R}^m$, $\mathbf{0}$ is a regular value of the function $f_{\mathbf{a}}(\mathbf{x}) = f(\mathbf{a}, \mathbf{x})$ with the parameter $\mathbf{a}$ fixed. In the following, we adapt this idea to the context of deep linear networks.

Motivated by the aforementioned observations and Generalized Sard's Theorem, we devise a regularization for the deep linear networks. Given $H+1$ matrices with positive real entries $\Lambda = (\Lambda_1, \ldots, \Lambda_{H+1})$ with each Λ_i having the same size as W_i, we can consider a generalized Tikhonov regularization of $\bar{\mathcal{L}}$ given by

$$\bar{\mathcal{L}}^{\Lambda} = \bar{\mathcal{L}}(W) + \frac{1}{2}(\|\Lambda_1 \circ W_1\|_F^2 + \cdots + \|\Lambda_{H+1} \circ W_{H+1}\|_F^2),$$

where $\Lambda_i \circ W_i$ denotes the Hadamard product (entrywise product) between Λ_i and W_i. That is, each term in $\Lambda_i \circ W_i$ is of the form of $\lambda_{i,j,k} w_{i,j,k}$, where each $\lambda_{i,j,k}$, the (j,k) entry of Λ_i, is a small positive real numbers that serve as a penalty coefficient. Therefore, the minimization problem for $\bar{\mathcal{L}}^{\Lambda}$ attempts to minimize $\bar{\mathcal{L}}$ and, at the same time, minimize each entries of the weight matrices. Note here that the penalty on each entry of the weight matrices could potentially be different. It is straightforward to verify that

$$\frac{\partial \bar{\mathcal{L}}^{\Lambda}}{\partial W_i} = U_i^{\top}(WXX^{\top} - YX^{\top})V_i^{\top} + \Lambda_i \circ W_i. \qquad (4)$$

When the entries of Λ_i's are small positive real numbers, we can see the above gradient system is a slightly perturbed

version of the original gradient system $\nabla\bar{\mathcal{L}}$. In the following, we demonstrate that this construction is sufficient to turn flat critical set of $\bar{\mathcal{L}}$ into isolated nondegenerate critical points. That is, the flatness of the critical points is removed.

First, we shall show the above regularization technique is sufficient to "desingularize" all *dense critical points*. Here, a dense critical point of $\bar{\mathcal{L}}^\Lambda$ is a (real) solution to $\frac{\partial\bar{\mathcal{L}}^\Lambda}{\partial W_i}=0$ for each i for which W_i contains no zero entries, i.e., all weight matrices are dense matrices.

Theorem 1 (Regularity of dense critical points)**.** *For almost all choices of Λ, all dense (real) critical points of $\bar{\mathcal{L}}^\Lambda$ are isolated and nondegenerate.*

Proof. Let $W=(W_1,\dots,W_{H+1})$ collect all the weight matrices and let m be the total number of entries in all these matrices. Consider the open set $(\mathbb{R}^*)^m=(\mathbb{R}\setminus\{0\})^m$. Let $F(W_1,\dots,W_{H+1},\Lambda)=(\frac{\partial\bar{\mathcal{L}}^\Lambda}{\partial W_i})_{i=1}^{H+1}$ be the gradient of $\bar{\mathcal{L}}^\Lambda$ with respect to $W_1,\dots,W_{H+1}$. Note that we treat the parameters, i.e., the entries in the Λ_i's, as variables with

$$\frac{\partial F}{\partial\lambda_{i,j,k}}=w_{i,j,k}$$

The Jacobian matrix of F is an $m\times 2m$ matrix. Since $\frac{\partial F}{\partial\Lambda}$ is a diagonal matrix whose diagonal entries are $w_{i,j,k}\neq 0$, we can conclude that the Jacobian matrix is of rank m, i.e., full row rank. Therefore $\mathbf{0}$ is a regular value for the map $F:(\mathbb{R}^*)^m\times\mathbb{R}^m\to\mathbb{R}^m$. By Generalized Sard's Theorem [95], for almost all choices of $\Lambda\in\mathbb{R}^m$, $\mathbf{0}$ is a regular value for the map $F_\Lambda:(\mathbb{R}^*)^m\to\mathbb{R}^m$ given by $F_\Lambda=F(\cdot,\Lambda)$. Consequently, for any W such that $F_\Lambda(W)=F(W,\Lambda)=\mathbf{0}$, the square Jacobian matrix $\frac{\partial F_\Lambda}{\partial W}$ must be of full column rank, i.e., nonsingular, which implies W must be a nonsingular solution of the equation. By the Inverse Function Theorem, such a solution must also be geometrically isolated. □

Here, a critical point is considered (geometrically) *isolated* if there is a neighborhood in which it is the only critical point. An isolated critical point is considered *nondegenerate* when the Hessian matrix at this point is nonsingular. The "almost all choices" in the above statement is to be interpreted in the sense of Lebesgue measure. It is also sufficient to take a probabilistic interpretation: if the entries of Λ are chosen at random, then with probability one, the above theorem holds.

Remark 2. *Instead of randomly drawing each λ_{ijk} separately, one can also consider $\lambda_{ijk}=\lambda+\rho_{ijk}$. Then, the ρ_{ijk}'s are drawn from a random distribution once for all and adjusting the regularization again reduces to a one parameter problem.*

The above regularization result can also be generalized to "sparse" cases which are desired in actual application. For instance, in convolutional neural networks, the first layer is generally highly structured and very sparse as it represents the application of convolution matrices. Similarly, many real world applications have specific sparsity pattern in mind. We therefore generalize the above result with respect to certain sparsity pattern. A sparsity pattern for the weight matrices is a set $\mathcal{N}$ of indices of the form (i,j,k) specifying the nonzero positions. We say the matrices $(W_1,\dots,W_{H+1})$ have the sparsity pattern $\mathcal{N}$ if for each $(i,j,k)\in\mathcal{N}$, the (j,k) entry of W_i is nonzero while all other entries are zero. We can generalize the above theorem to weight matrices having a given sparsity pattern in which the dense case considered in Theorem 1 is a special case in which all entries are nonzero.

Theorem 2 (Regularity of sparse solutions)**.** *Given a sparsity pattern $\mathcal{N}$, for almost all choices of Λ, all (real) solutions of the gradient system $\nabla\bar{\mathcal{L}}^\Lambda=\mathbf{0}$ having the sparsity pattern $\mathcal{N}$ are geometrically isolated and nonsingular.*

Proof. Let $W^{\mathcal{N}}$ be the set of all $w_{i,j,k}$'s for which $(i,j,k)\in\mathcal{N}$. That is, $W^{\mathcal{N}}$ collect all the nonzero entries in the weight matrices. By fixing all the remaining entries to zero, the gradient equations $\frac{\partial\bar{\mathcal{L}}^\Lambda}{\partial W_i}$ for $i=1,\dots,H+1$ under the regularization can be considered as a system in $W^{\mathcal{N}}$ only.

Following the previous proof, we can define $m=|W^{\mathcal{N}}|$ and $F(W^{\mathcal{N}},\Lambda^{\mathcal{N}})$ to be the system of gradient equations with $\Lambda^{\mathcal{N}}$ (entries in Λ corresponding to $W^{\mathcal{N}}$) also considered to be variables. Then, as in the previous case, the Jacobian matrix of F is an $m\times 2m$ matrix with $\partial F/\partial\Lambda^{\mathcal{N}}$ being a diagonal matrix with nonzero diagonal entries $w_{i,j,k}$ for $(i,j,k)\in\mathcal{N}$. Consequently, this Jacobian matrix also has full row rank. By Generalized Sard's Theorem, we can conclude that for almost all choices of $\Lambda^{\mathcal{N}}\in\mathbb{R}^m$, all solutions to $F_{\Lambda^{\mathcal{N}}}(W^{\mathcal{N}})=F(W^{\mathcal{N}},\Lambda^{\mathcal{N}})=\mathbf{0}$ must be geometrically isolated and nonsingular. □

Note that the regularization $\bar{\mathcal{L}}^\Lambda$ is constructed as a perturbation of the original loss function $\bar{\mathcal{L}}$ with small penalty terms added to also minimize the magnitude of each weight coefficient. The theory of *homotopy continuation* [97], [98] also guarantees that for sufficiently small perturbation, this process can be reversed. The following is an immediate consequence of the Implicit Function Theorem.

Proposition 1. *For sufficiently small regularization coefficients Λ, as all entries of Λ shrink to 0 uniformly, the critical points of $\bar{\mathcal{L}}^\Lambda$ also move smoothly and either converge to regular critical sets of $\bar{\mathcal{L}}$ or diverge to infinity.*

Here, "diverge to infinity" means as the perturbation coefficients in Λ shrink to zero, certain coordinates of the critical point of $\bar{\mathcal{L}}^\Lambda$ are unbounded.

Remark 3. *More rigorous description of this phenomenon of diverging solutions can be given in terms of* projective space *(e.g., see [51], [52]) which encapsulates infinity as an actual place in the space. In that sense, certain critical points of $\bar{\mathcal{L}}^\Lambda$ may converge to "saddle points at infinity" of $\bar{\mathcal{L}}$.*

Another important observation from the homotopy point of view is that while this perturbation slightly alters the loss landscape, any global minimum will survive in the following sense. The following is an immediate consequence of [99] as well as the Implicit Function Theorem.

Proposition 2. *For sufficiently small regularization coefficients Λ, as all entries of Λ shrink to 0 uniformly, there is at least one critical point of $\bar{\mathcal{L}}^\Lambda$ that will converge to a global minimum of $\bar{\mathcal{L}}$.*

Below, we show the regularization technique implemented for Ex. 1.

Example 4. *A regularization of the polynomial in Ex. 1 is*

$$f(x,y,z)=2xy+2xz-2x-y-z+\tfrac{1}{1000}\left(2x^2+y^2+3z^2\right)$$

with gradient

$$\nabla(f) = \{\tfrac{x}{250} + 2y + 2z - 2, 2x + \tfrac{y}{500} - 1, 2x + \tfrac{3}{500}z - 1\}.$$

The critical point system $\nabla(f) = 0$ *defines a zero-dimensional solution set. In fact, there is a unique stationary point, which is approximately* $(0.49925, 0.74925, 0.24975)$.

Remark 4. *There have been various methods proposed for escaping flat saddle points (in the wide minima sense) in the absence of singular saddles [25], [36], [37]. Recent attempts have also been made to extend such methods in the presence of singular saddle points in limited cases [12], [38], [39]. Using the generalized* L_2*-regularization, only the former set of methods may be required to escape saddles and achieve a deeper minimum.*

3.3 Role of the Regularization

The following emphasizes the role of regularization and, specifically, that of the proposed generalized L_2-regularization. Regularization is a frequently used technique in machine learning models using neural networks in practice, such as LASSO. It benefits the resulting model using machine learning methods in several ways such as avoiding the problem of overfitting and ill-conditioned optimization stage. Here, we adopted a generalized L_2-regularization technique that provides two additional benefits.

From a computational point of view, the generalized L_2-regularization ensures all critical points are isolated. This allows the use of efficient, scalable, and stable numerical methods such as homotopy continuation methods [100], [101], [102], [103], to locate *all* critical points.

From the perspective of training, another important role that the generalized L_2-regularization plays is in ensuring a certain form of robustness of the training process. The regularity of the critical points established in Theorem 2 combined with Generalized Sard's Theorem ensures that sufficiently small perturbations in the training data causes small changes in the critical points of $\bar{\mathcal{L}}^\Lambda$ and preserves their indices.

4 Estimating Number of Solutions

By using the generalized L_2-regularization, we are left with only isolated stationary points. In this section, we focus on estimates on the number of *isolated* solutions of Eqs. (4).

4.1 Upper Bounds on Number of Solutions

The algebraic geometry interpretation of the gradient system of deep linear networks allows us to utilize different bounds on the number of complex solutions to bound the number of stationary points. To that end, suppose that $f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x}))$ is a polynomial system where $\mathbf{x} \in \mathbb{C}^n$, i.e., f is a *square* system of polynomials.

4.1.1 Classical Bézout Bound

The simplest upper bound on the number of isolated complex points in $V(f)$ is the *classical Bézout bound* (CBB) which is simply the product of the degrees of the polynomials in f, namely $\prod_{i=1}^n \deg f_i$. In fact, this bound, and all others discussed below, are generically sharp with respect to the structure that they capture.

From (3) and the definition of $\bar{\mathcal{L}}^\Lambda$, we can see that the leading terms in each polynomial are formed by the product of $2H+1$ matrices. Therefore, each polynomial has degree $2H+1$. The CBB is therefore the product of these degrees:

Proposition 3. *The regularized loss function* $\bar{\mathcal{L}}^\Lambda$ *has at most* $(2H+1)^n$ *complex isolated critical points where* n *is the total number of weights.*

Refinements of CBB can be accomplished by utilizing additional structure such as the multihomogeneous structure resulting in the multihomogeneous Bézout bound. Since the next bound under consideration is always no larger than the multihomogeneous Bézout bound, we will not consider multihomogeneous structure here.

4.1.2 BKK Bound

Since the systems arising from real-world applications are typically sparse, a refinement of the CBB based on the sparsity structure is the *Bernshtein-Kushnirenko-Khovanskii Bound*, or simply *BKK Bound* [104], [105]. It is given by a geometric invariant defined on the monomial structure — the *mixed volume* of the convex bodies created by the set of monomials appear in f (i.e., the *Newton polytopes* of f). This bound was originally proposed for bounding the number of solutions in $(\mathbb{C}^*)^n = (\mathbb{C} \setminus \{\mathbf{0}\})^n$ but can also be extended to a bound on number of isolated solutions in $\mathbb{C}^n$ [106], [107].

By exploiting the sparsity structure, the BKK bound for the gradient system of $\bar{\mathcal{L}}^\Lambda$ is much lower than the Bézout number as shown in Table 1.

4.2 Analytical Results for Mean Number of Real Solutions of Random Polynomial Systems

There are only a handful of results known for the upper bounds on the number of isolated real stationary points of polynomial loss functions [108] and for upper bounds on the number of isolated real solutions of systems of polynomial equations [109], [110], [111], [112], [113], [114], [115].

To gain further insight on the number of real stationary points of (1) (with entries of X and Y picked randomly from some probability distribution), we compare the existing analytical results for the mean number of real stationary points of random polynomial cost functions. The most general random polynomial cost function is written as:

$$F(\mathbf{x}) = \sum_{|\alpha| \leq 2H+2} a_\alpha x_1^{\alpha_1} \ldots x_n^{\alpha_n}, \tag{5}$$

with n being the number of variables and $2H+2$ is the highest degree of the monomials, $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ is a multi-integer with $|\alpha| = \alpha_1 + \cdots + \alpha_n$. Here, a_α are random coefficients i.i.d. drawn from the Gaussian distribution with mean 0 and variance 1. In [108], it was shown that the mean number of real stationary points of this cost function, i.e., the mean number of real solutions of the corresponding gradient system $\frac{\partial F(\mathbf{x})}{\partial x_i} = 0$ for $i = 1, \ldots, n$, is:

$$\mathcal{N}_{DM}(H, n) = \sqrt{2}(2H+1)^{\frac{n+1}{2}}, \tag{6}$$

i.e., the mean number of real stationary points of the random polynomial of the same degree and number of variables as the loss function of the deep linear network. This result also

yields that the mean number of real stationary points of such a dense random polynomial function is significantly smaller than the corresponding CBB as expected since CBB bounds the number of complex solutions.

4.3 Equations for the Zero Training Error

This subsection briefly considers the problem of finding a special type of minima, called the *zero training error minima*, as these were recently studied for certain class of deep learning models. In [116], deep nonlinear networks with rectified linear units (ReLUs) were considered and the ReLUs were approximated with polynomials of certain degree. When there are more weights than the number of data points, there always are infinitely many global minima expected.

For deep linear networks, zero training error minima satisfy $\bar{\mathcal{L}}(W) = 0$, i.e.,

$$(W_{H+1}W_H \cdots W_1 X)_{\cdot,i} - Y_{\cdot,i} = 0, \tag{7}$$

for all $i = 1, \ldots, m$. It must be emphasized that these minima may only exist if the model can fit *all* the training data perfectly well. Except for some special cases, it is also difficult to know if such minima exist for a chosen model *a priori*. Clearly, if such minima exist, they are the global minima of the model for the specific dataset. Here, we *assume* that such zero training error minima do exist for our deep linear networks for the given data matrices. Then, (7) is again a system of m polynomial equations in n variables. In short, for $H \geq 1$ and $Y \neq 0$, the zero training error minima system (7) has no isolated solutions. For the case when $m = n$, the CBB is $(H+1)^n$ complex isolated solutions.

We emphasis that the assumption that such zero training error minima do exist is a very strong one as it means that each data point is exactly fit, which either may not occur in practice or may be a case of over-fitting.

Remark 5. *For the underdetermined systems, the CBB and BKK are actually bounds on the number of connected components (flat stationary points). The existence of positive-dimensional components reduces the maximum number of isolated solutions. In fact, even for an apparently underdetermined system, it may be possible to have only isolated stationary points over the real numbers. However, except for the special case of $m = n$, these bounds do not provide any detailed information about flat stationary points.*

4.4 Symmetrical Solutions

This subsection shows the existence of symmetries in the solutions of the gradient equations of deep linear networks.

Proposition 4. *When $H = 1$, if W_1^* and W_2^* form a solution to system* (4)*, then simultaneously switching the signs of the i^{th} row of W_1^* and i^{th} column of W_2^* also yields a solution for $i = 1, ...d_1$.*

Proof. Suppose that r_i is the i^{th} row of W_1^* and c_i is the i^{th} column of W_2^*. Thus, (4) can be rewritten as

$$(W_2^*)^\top \left(\left(\sum_{i=1}^{d_1} c_i r_i \right) XX^\top - YX^\top \right) + \Lambda_1 \circ W_1^* = 0 \tag{8}$$

$$\left(\left(\sum_{i=1}^{d_1} c_i r_i \right) XX^\top - YX^\top \right) W_1^* + \Lambda_2 \circ W_2^* = 0 \tag{9}$$

The result is immediately seen from (8) and (9). □

If $H = 1$ and $d_1 = n$, Prop. 4 shows that if (4) has a solution such that all entries of W_1^* and W_2^* are nonzero, then it has at least 2^n solutions. This can be generalized to arbitrary H to show that a natural sequence of simultaneous sign switching will lead to additional solutions.

5 Numerically Finding All the Stationary Solutions of the Deep Linear Networks

Although solving systems of nonlinear equations can be a prohibitively difficult task, identifying (4) as a system of polynomial equations allows for several sophisticated computational algebraic geometry techniques to be employed to find all isolated complex solutions of the system. Once all complex solutions have been computed, the real solutions can then be trivially identified. Symbolic methods such as using Gröbner basis techniques [52], [51] and techniques in real algebraic geometry [117] could be used to solve these systems, they may severely suffer from algorithmic complexity issues. The approach we utilize is homotopy continuation which has already been applied to find minima and stationary points of artificial neural networks in the literature [118], [119], [120], [121], [122], [123]. Local real homotopy methods perform well in finding solutions (and often guarantee global convergence to a solution), they do not guarantee to find all isolated solutions. In this section, we describe a sophisticated method called the numerical homotopy continuation (NPHC) [124], [125] method which guarantees to find *all* complex isolated solutions of systems of multivariate polynomial equations. Then, we present our results for the deep linear networks using the NPHC method.

5.1 The NPHC Method

For a square system of polynomial equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, i.e., $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x}))$ and $\mathbf{x} = (x_1, \ldots, x_n)$, one first determines an upper bound, such as the ones described in Sec. 4, on the number of isolated complex solutions. Then, another square polynomial system $\mathbf{g}(\mathbf{x})$ is constructed such that satisfying the following two properties:

1) $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ is easy to solve, and
2) the number of complex solutions $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ is equal to the corresponding upper bound.

For the CBB from Sec. 4.1.1, a straightforward choice is $\mathbf{g}(\mathbf{x}) = (x_1^{d_1} - 1, \ldots, x_n^{d_n} - 1)$ where $d_i = \deg f_i$. For tighter upper bounds, constructing $\mathbf{g}(\mathbf{x})$ can be more involved and the reader is referred to [100], [103] for further details.

Next, a parametrized system $\mathbf{h}(\mathbf{x}; t)$ is formed connecting $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ creating a so-called polynomial homotopy. A homotopy that linearly interpolates between $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ is

$$\mathbf{h}(\mathbf{x}; t) = (1-t)\mathbf{f}(\mathbf{x}) + \gamma t \mathbf{g}(\mathbf{x}) = \mathbf{0} \tag{10}$$

where $t \in [0, 1]$ such that $\mathbf{h}(\mathbf{x}; 1) = \mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x}; 0) = \mathbf{f}(\mathbf{x})$, and $\gamma \in \mathbf{C}$ is a generic complex number.

For each complex isolated solution of $\mathbf{h}(\mathbf{x}; 1) = \mathbf{g}(\mathbf{x}) = \mathbf{0}$, all of which are known by construction, a numerical predictor-corrector method evolves the solution of $\mathbf{h}(\mathbf{x}; t) = 0$ from $t = 1$ to $t = 0$. As long as γ is a generic complex number,

every complex isolated solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ can be reached starting from a solution of $\mathbf{g}(\mathbf{x}) = \mathbf{0}$. Specifically, it is proven [126] that each of such solution path can only exhibit either of the two characteristics:

1) the path converges at $t = 0$ and hence a solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ is found, or
2) the path diverges to infinity as $t \to 0^+$.

In particular, every solution path is regular over $t \in (0, 1]$ and yields no bifurcation, singularities, path-crossing, etc. Hence, after tracking all possible solution paths (as many as the estimated upper bound), we obtain all complex isolated solutions of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. Moreover, the method is embarrassingly parallelizable since each solution path can be tracked independent of each other.

Example 5. *Using the data from Ex. 2, if we utilize* $\lambda = 0.01$, *there are only* 13 *isolated stationary points,* 5 *of which are real. Two are local minima that are both global minima and the other three are saddles.*

5.2 Computational Details

The setup for the results described next, which demonstrate the effect of changing Λ, d_x, d_y, m, and H on number of isolated real solutions, is as follows. For each case, we take each entry of the data matrices $X \in \mathbb{R}^{d_x \times m}$ and $Y \in \mathbb{R}^{d_y \times m}$ i.i.d. drawn from the Gaussian distribution with mean 0 and variance 1. Each entry of the matrices Λ_i are drawn i.i.d from the uniform distribution between 0 and $\Lambda_{\max} > 0$. For every case, 1000 trials are run with all isolated solutions computed using the software *Bertini* [103] which provides an efficient implementation of the NPHC method. We explore how the change of any of the five variables impact the solutions of (4) for modest size systems.

5.3 Results

Using the setup from Sec. 5.2, the following summarizes results of solving (4).

5.3.1 Enumeration of Complex and Real Solutions

The first experiment is to compare the upper bounds on the number of solutions discussed in Sec. 4 with results from numerical experiments.

Table 1 records the number of weights n, CBB, BKK, mean number of complex solutions $\mathcal{N}_{\mathbb{C}}$, the Dedieu-Malajovich number of average real solutions of random polynomial cost function $\mathcal{N}_{DM}$, the maximum number of real solutions (for all Λ-values), and the maximum index among all the solutions over all samples for various values of H, m, d_x, and d_y while fixing $d_1 = \cdots = d_H = 2$. As expected, the CBB grows exponentially with the number of variables. Though the BKK count grows rapidly as well, it is significantly smaller than CBB. Nonetheless, the actual number of complex solutions computed using the NPHC method is even smaller compared to these two bounds. Moreover, the maximum number of real solution is also smaller than Dedieu-Malajovich number. Both these observations yield that our gradient system is highly sparse and structured compared to that of the dense polynomial cost functions (5).

Table 2 records numerical results for the mean number of complex solutions $\mathcal{N}_{\mathbb{C}}$, maximum number of real solutions out of all samples $\max(\mathcal{N}_{\mathbb{R}})$, and the maximum index out of all real solutions of all samples $\max(I)$.

Note that $m = 1$ is a pathological case as it refers to only one data point case, but it still creates nonlinearity in the gradient equations yielding nontrivial solutions. Moreover, the value of m does not change the degree of the polynomials but only the monomial structure of the polynomials. When $m = 1$, the matrix $XX^\top$ in (4) is singular and of rank 1, which imposes additional structure. For $m > 1$ and $d_x \leq m$, $XX^\top$ is nonsingular with probability 1 so that the polynomial system has the very same structure yielding a constant number of complex solutions for generic values of X and Y.

5.3.2 Distribution of Number of Real Solutions

To see the impact of the regulation term Λ, we change the maximum value $\Lambda_{\max}$ of the interval on which Λ is uniformly distributed. Figure 1 shows how the distribution of $\mathcal{N}_{\mathbb{R}}$ changes as a function of $\Lambda_{\max}$. In particular, the mean number of real solutions decrease as $\Lambda_{\max}$ increases which yields the phenomenon of topology trivialization [127], [128], [129], [130], [131]. As $\Lambda_{\max}$ increase beyond 1, there are more samples with no real solutions. Furthermore, as $\Lambda_{\max}$ approaches zero, the mean number of real solutions becomes relatively stable but the condition number of the real solutions begin to increase. This is expected since the system (4) is converging to the unregularized case.

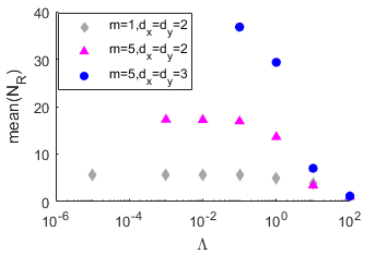


Fig. 1. The mean of $\mathcal{N}_{\mathbb{R}}$ as a function of $\Lambda_{\max}$ for $H = 1$ and $d_1 = 2$.

Figure 2 demonstrates the impact on the average number of real solutions as a function of d_x, d_y, and m. It shows that increasing any of these three parameters leads to an increase in the mean number of real solutions. Combining Figure 2 and Table 1, one notices that the more data points there are, the more real solutions to (4) there are, on average.

5.3.3 Index-resolved Number of Real Solutions

The next experiment compares the ratio of number of real solution with index I, $\mathcal{N}_I$, to the total number of real solutions using 1000 samples for each case. For the samples for which there are no real solutions, we set the ratio to be 0.

Figure 3 and 4 demonstrate the index distribution as a function of d_x, $\Lambda_{\max}$, and m, respectively. From Table 2 and the right plot of Figure 3, we notice that, for the cases $H = 1$, $m = 2$, and $d_y = d_1 = 2$, even though the number of variables increased from 8 to 14 as d_x increased from 2 to 5, the number of isolated solutions remained the same. It also shows that when $H = 1$, $m = 2$, $d_y = d_1 = 2$, and $\Lambda_{\max} = 1$, the highest index is 4 and the probability of a solution to (4) is not an extrema of $\mathcal{L}^\Lambda$ increases as d_x increases.

TABLE 1
Upper bounds on the number solutions for (4) based on CBB and BKK, with comparison to the Dedieu-Malajovich number $\mathcal{N}_{DM}$ which are independent of the parameter values. When the network has more than one layer, $d_i = 2$ for all integers i. The number of isolated complex solutions with generic parameters is $\mathcal{N}_{\mathbb{C}}$ while max$(\mathcal{N}_{\mathbb{R}})$ and max(I) the maximum number of real solutions and the highest index of a real solution found among all the samples, respectively.

H	m	d_x	d_y	n	CBB	BKK	$\mathcal{N}_{\mathbb{C}}$	$\mathcal{N}_{DM}(H,n)$	max$(\mathcal{N}_{\mathbb{R}})$	max(I)
1	1	2	2	8	$3^8 = 6561$	1024	33	199	9	2
1	1	3	2	10	3^{10}	5184	33	592	9	2
1	1	4	2	12	3^{12}	16384	33	1786	9	2
1	1	5	2	14	3^{14}	40000	33	5357	9	2
1	1	10	2	24	3^{24}	640000	33	1301759	9	2
1	1	2	3	10	3^{10}	5184	73	592	9	2
1	1	2	4	12	3^{12}	16384	129	1786	9	2
1	1	2	5	14	3^{14}	40000	201	5357	9	2
2	1	2	2	12	$5^{12} = 152587890625$	770048	641	6250000	65	3

TABLE 2
Computational results of $\mathcal{N}_{\mathbb{C}}$, $\mathcal{N}_{\mathbb{R}}$, $\mathcal{N}_{DM}$, and $\max(I)$ for the cases $m > 1$. As in Table 1, $\mathcal{N}_{DM}$ and $\mathcal{N}_{\mathbb{C}}$ are independent of the choice of Λ. The values max$(\mathcal{N}_{\mathbb{R}})$ and max(I) are based on 1000 samples for each case with $\Lambda \in [0, 1]$ and $d_i = 2$ for all integers i.

H	m	d_x	d_y	n	$\mathcal{N}_{\mathbb{C}}$	$\mathcal{N}_{DM}(H,n)$	max$(\mathcal{N}_{\mathbb{R}})$	max(I)
1	2	2	2	8	225	199	29	4
1	2	3	2	10	225	592	29	4
1	3	2	2	8	225	199	29	4
1	4	2	2	8	225	199	29	4
1	5	2	2	8	225	199	29	4
1	20	2	2	8	225	199	29	4
1	5	3	3	12	2537	1786	73	6

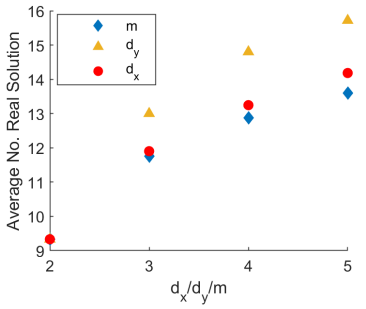


Fig. 2. The mean of $\mathcal{N}_{\mathbb{R}}$ for different value of d_x (circles), d_y (triangles) and m (diamonds) where $\Lambda_{\max} = 1$. For circles: $H = 1$, $m = 2$, and $d_y = d_1 = 2$. For triangles: $H = 1$, $m = 2$, and $d_x = d_1 = 2$. For diamonds: $H = 1$ and $d_x = d_y = d_1 = 2$.

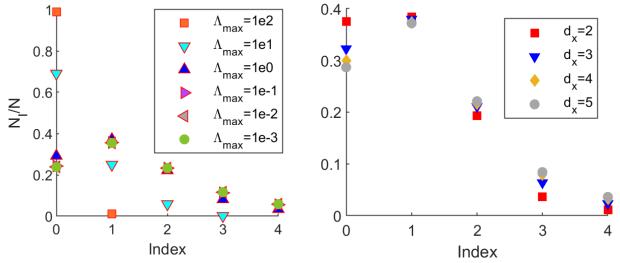


Fig. 3. The index distribution for different d_x and $\Lambda_{\max}$. For the left plot, $H = 1$, $m = 5$, $d_x = d_y = d_1 = 2$, and $\Lambda \in [0, \Lambda_{\max}]$. For the right plot, $H = 1$, $m = 2$, $d_y = d_1 = 2$, and $\Lambda_{\max} = 1$.

The left plot of Figure 3 reveals that as $\Lambda_{\max}$ approaches 0, the index distribution reach an equilibrium. Figure 4 informs us that, for cases where $H = 1$ and $d_x = d_y = d_1 = 2$, the highest solution index is 4 and the peak frequency is always reached by solutions with index 2. Similar as in the cases with different d_x, the probability that a solution to (4) is a saddle point of $\mathcal{L}^{\Lambda}$ increases sub-linearly as the number of data points increases.

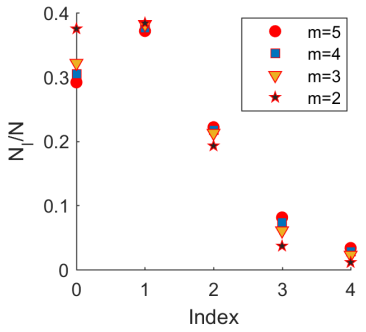


Fig. 4. The index distribution for different m where $H = 1$, $d_x = 2$, $d_y = 2$, and $\Lambda_{\max} = 1$.

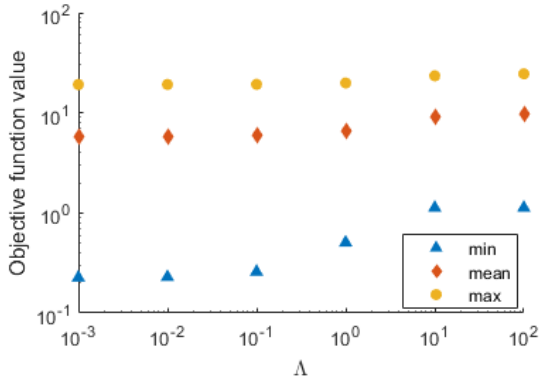


Fig. 5. The minimum, mean, and maximum of global minimum loss function value at real solutions of 1000 samples with different $\Lambda_{\max}$. The other parameters are $H = 1$, $m = 5$, and $d_1 = d_x = d_y = 2$.

5.3.4 Minima

In the following, we take a closer look at the structure of all the real solutions for each sample. It is straightforward to verify that the configuration with all weights being zero is always a solution of (4). For the cases where $H = 1$ and $m = 1$, we have a total of 13,000 samples consisting of 13 different scenarios listed in Tables 1 and 2 and Figure 1. For these cases, all local minima are global minima, and the absolute values of all the local minima are the same, i.e., all the minima are symmetrically related to each other.

For the cases where $H > 1$ or $m > 1$, we have a total of 15,000 samples consisting of 15 different scenarios as listed in Tables 1 and 2 and Figure 1. Here, we observe instances where there exist local minima which are not global minima. One such instance is given in Table 3 and the parameters for the system are

$$X = \begin{bmatrix} -0.1297 & 0.5236 & -2.1491 & 0.3252 & 0.7313 \\ -1.0135 & -1.4616 & -1.6352 & -0.4289 & -0.8680 \\ 0.2523 & 1.8664 & 1.2240 & 0.0116 & 0.9282 \end{bmatrix},$$
$$Y = \begin{bmatrix} 0.6973 & -0.6288 & 1.0285 & -0.9793 & 1.0402 \\ -0.0452 & -0.8566 & -0.2397 & -1.1334 & 1.2315 \\ 0.1912 & -0.3887 & -0.4516 & 0.0221 & 0.5602 \end{bmatrix}, \quad (11)$$
$$\Lambda_0 = \begin{bmatrix} 0.383 & 0.6917 & 0.9245 \\ 0.298 & 0.8805 & 0.0813 \end{bmatrix}, \Lambda_1 = \begin{bmatrix} 0.4827 & 0.884 \\ 0.1283 & 0.1963 \\ 0.2529 & 0.1214 \end{bmatrix}.$$

When $H = 1$ and $m = 5$, we notice that all sample runs with $\Lambda_{\max} = 100$ exhibit all local minima are global minima. However, combining with the observation from Figure 1, this may only be an artifact of the topology trivialization.

5.3.5 Loss Function at Real Solutions

To see how the value of $\Lambda_{\max}$ impacts the loss function value, we compute the global minimum of each sample and plot the minimum, mean, and maximum of 1000 samples for different values of $\Lambda_{\max}$ and summarized in Figure 5. We observe that as $\Lambda_{\max}$ approaches zero, the mean and minimum of the global minima approaches nonzero constants. This implies that, for a generic case with 5 data points and one hidden layer, there can be no values of W_1 and W_2 such that the loss function achieves a global minimum of 0, i.e., not possible to have a zero training error minima.

5.3.6 A simple application to the Boston Housing data

To better explain the implication of the algebraic theory on the loss surface of a deep linear network, we apply the above described method to the Boston housing data which can be found at https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html. There are 13 variable factors documented which may contribute to the price of a house. For the purpose of this example, we remove the two categorical variables from the data. We train a neural network with linear activation function to predict the price of Boston houses. The loss function for the 2-layer deep linear network is given as

$$\|\text{Price} - W_2 \times W_1 \times \text{factors}\| + \|\Lambda_1 \circ W_1\| + \|\Lambda_2 \circ W_2\|. \quad (12)$$

The *Price* is a scalar indicating the predicted price of a house using the 11 *factors*. The hidden layer W_1 has size 2×11 and the layer W_2 has size 1×2. Here, when Λ_1 and Λ_2 are zero, there exist only flat minima. By Theorem 1, when Λ_1 and Λ_2 are randomly selected, there are finitely many critical points for (12). Consequently, there are finitely many local minima.

Here, we find all local minima and saddle points of (12) using *Bertini* with Λ_1 and Λ_2 randomly selected uniformly in $[0, 1]$. There are 5 sets of real solutions including the trivial (zero) solution. The 4 non-trivial isolated solutions have the following format:

$$\text{Solution 1} \quad \underbrace{\begin{bmatrix} 0 & \square \end{bmatrix}}_{W_2} \times \overbrace{\underbrace{\begin{bmatrix} 0 & \cdots & 0 \\ \square & \cdots & \square \end{bmatrix}}_{W_1}}^{11 \text{ columns}}$$

$$\text{Solution 2} \quad \underbrace{\begin{bmatrix} 0 & -\square \end{bmatrix}}_{W_2} \times \overbrace{\underbrace{\begin{bmatrix} 0 & \cdots & 0 \\ -\square & \cdots & -\square \end{bmatrix}}_{W_1}}^{11 \text{ columns}}$$

$$\text{Solution 3} \quad \underbrace{\begin{bmatrix} \square & 0 \end{bmatrix}}_{W_2} \times \overbrace{\underbrace{\begin{bmatrix} \square & \cdots & \square \\ 0 & \cdots & 0 \end{bmatrix}}_{W_1}}^{11 \text{ columns}}$$

$$\text{Solution 4} \quad \underbrace{\begin{bmatrix} -\square & 0 \end{bmatrix}}_{W_2} \times \overbrace{\underbrace{\begin{bmatrix} -\square & \cdots & -\square \\ 0 & \cdots & 0 \end{bmatrix}}_{W_1}}^{11 \text{ columns}}$$

In particular, Solutions 1 and 2 share the same values on the non-zero elements as does Solutions 3 and 4. Following the notation in Table 3, we present the optimal solutions below in Table 4.

Following a similar procedure as in Section 5, we obtain multiple sets of solutions while decreasing the magnitude of Λ_1 and Λ_2 to 0. These four solutions yield the same minimum loss function value of 112.6 while the zero solution has a higher loss function value at 547.5. Figure 6 shows the loss function near Solution 1 and the zero solution. The trivial solution is a saddle point and not a minimum. Here, all local minima are global minima.

To compare the solution of the deep linear network with traditional linear regression, we compute the linear

TABLE 3
All local minima arising from (11) with $H = 1$, $m = 5$, $d_x = d_y = 3$, and $d_1 = 2$ such that there are local minima which are not global minima.

w^0_{11}	w^0_{21}	w^0_{12}	w^0_{22}	w^0_{13}	w^0_{23}	w^1_{11}	w^1_{21}	w^1_{31}	w^1_{12}	w^1_{22}	w^1_{32}	$\mathcal{L}^\Lambda$
0.42959	0.36758	0.30899	-0.10019	-0.01419	-0.23650	-0.50336	0.33655	-0.01843	-0.11969	-0.14925	0.54928	7.13717
-0.42959	0.36758	-0.30899	-0.10019	0.01419	-0.23650	0.50336	-0.33655	0.01843	-0.11969	-0.14925	0.54928	7.13717
-0.42959	-0.36758	-0.30899	0.10019	0.01419	0.23650	0.50336	-0.33655	0.01843	0.11969	0.14925	-0.54928	7.13717
0.42959	-0.36758	0.30899	0.10019	-0.01419	0.23650	-0.50336	0.33655	-0.01843	0.11969	0.14925	-0.54928	7.13717
0.54286	-0.05927	0.22411	-0.05389	-0.04306	0.26254	-0.51936	0.16009	0.25030	0.05058	-0.17838	-0.07580	7.16775
0.54286	0.05927	0.22411	0.05389	-0.04306	-0.26254	-0.51936	0.16009	0.25030	-0.05058	0.17838	0.07580	7.16775
-0.54286	-0.05927	-0.22411	-0.05389	0.04306	0.26254	0.51936	-0.16009	-0.25030	0.05058	-0.17838	-0.07580	7.16775
-0.54286	-0.05927	-0.22411	-0.05389	0.04306	0.26254	0.51936	-0.16009	-0.25030	0.05058	-0.17838	-0.07580	7.16775

TABLE 4
The nontrivial optima of (12). Solutions 1 and 2 are negatives of each other and so are Solutions 3 and 4.

Solution No.	w^1_{12}	w^0_{21}	w^0_{22}	w^0_{23}	w^0_{24}	w^0_{25}	w^0_{26}	w^0_{27}	w^0_{28}	w^0_{19}	$w^0_{2\,10}$	$w^0_{2\,11}$
1(2)	2.3803	-0.0305	0.0187	-0.0139	-0.9908	2.4981	-0.0030	-0.4401	-0.0008	-0.1770	0.0059	-0.1823
	w^1_{11}	w^0_{11}	w^0_{12}	w^0_{13}	w^0_{14}	w^0_{15}	w^0_{16}	w^0_{17}	w^0_{18}	w^0_{19}	$w^0_{1\,10}$	$w^0_{1\,11}$
3(4)	2.6526	-0.0273	-0.0168	-0.0125	-0.8891	2.2416	-0.0027	-0.3950	-0.0007	-0.1588	0.0053	-0.1635

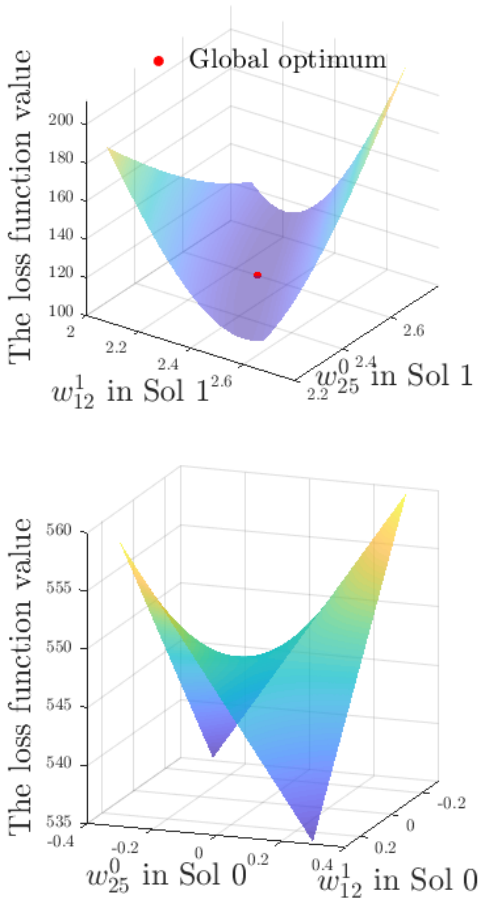


Fig. 6. The loss surfaces near two solutions found by *Bertini*. The top figure shows the loss surfaces for Solution 1 against variable w^1_{12} and w^0_{25}. The bottom figure shows the loss surfaces for zero solution against variable w^1_{12} and w^0_{25}

regression model for the Boston housing data with the following loss function:

$$\|Price - W \times \text{factors}\|$$

where W is a vector of size 1×11. For the Boston house data, the L_2-difference between W at the global (and the only) minimum for the linear regression without any regularization and the product W_2W_1 computed from the global minima for the deep linear network as Λ vanishes is numerically zero. Thus, the linear regression model is correctly reproduced.

6 Conclusions and Discussion

Understanding nonconvexities of optimization problems arising in deep learning and their implications are an active area of research. Deep linear networks have served as an ideal test ground of ideas as they qualitatively capture certain features of deep nonlinear networks yet simple enough for analytical and numerical investigations. In the present paper, we have initiated an ambitious plan to understand the loss landscapes of deep networks from an algebraic geometric point of view. Our approach is to provide practicable results from algebraic geometry rather than abstract ones by invoking computational and numerical methods in algebraic geometry.

Algebraic Geometry Interpretation:- In the present paper, after reviewing existing results on deep learning loss surfaces as well as for deep linear loss surfaces, we observed that the system of gradient equations of the deep linear networks is an algebraic system and argued that by complexifying the equations brings the problem of solving this system into the complex algebraic geometry domain. In turn, we can utilize many of the mature results and methods from algebraic geometry to gain insights into the optimization landscapes of these systems.

We emphasis that the algebraic geometric interpretation of gradient equations is not restricted only to the deep linear networks: classes of deep nonlinear networks which

obviously fall under the algebraic geometry paradigm are deep polynomial networks and deep complex networks. While any other activation functions can be approximated by polynomials of finite degrees, the gradient systems for most of the contemporary activation functions used for deep nonlinear networks in practice such as hyperbolic tangent, sigmoid, rectified linear units (ReLUs), leaky ReLUs, Heaviside, etc. activation functions are, or can be transformed to, algebraic systems. Hence, the results and methods can also be applied, after appropriate modifications, to investigate loss landscapes of deep nonlinear networks.

Flat Stationary Points:- We reviewed the current understanding of "flat" minima in deep learning and provided a distinction among different definitions of "flat" minima and other stationary points. In particular, a flat stationary point in our case is a point on a connected component in the weight space such that each of the points on this component are solutions of the gradient equations and that the loss function remains constant on the whole component. Such flat stationary points also called positive-dimensional solutions where the dimension refers to the (real or complex) dimension of the component. Such a flat minimum over the real space is distinct from an isolated stationary point in the real space even though the Hessian matrices evaluated at both of which are singular.

For deep linear networks, we showed that there do exist positive-dimensional components when no regularization is used. In the existing literature, the deep linear networks are shown to posses no local minima which are not global minima. Our results then yield that the loss surface of unregularized deep linear network consists of flat minima forming *minima lakes*, each of which are at the same level as the global minimum. In fact, the landscape also consists of *stationary point lakes* with the Hessian matrix having higher index at these solutions.

Regularizing Flat Stationary Points and Role of Regularization:- Using Generalized Sard's Theorem, we showed that when an extension of L_2-regularization is added to the loss function, all (complex and real) stationary points become isolated, i.e., no flat stationary points exist. In addition, this regularization also removes isolated singular solutions.

Since the stochastic gradient descent (SGD) method and its variants rely only on first order (gradient) information while searching for a minimum, they have to pass near or through saddle points of higher index. The number of saddle points of higher index is usually exponentially more than the number of minima in high dimensional and nonlinear loss landscapes. In addition, if there are flat saddle points present in the system, SGD may encounter further issues such as the computation getting stuck at the flat saddle point which, in turn, results in performance plateaus for many epochs. Recently, a few attempts have been made to devise methods that escape from wide minima in the absence of singular solutions (and in presence of singular solutions in limited cases) [12], [25], [36], [37], [38], [39]. An alternative way to evade singular solutions (both flat and degenerate) may be to use the proposed regularization which eliminates flat stationary points and minima right from the beginning of the SGD computation. Hence, wide minima escaping methods could then be applied to achieve better training.

The existence and implications of flat minima have been discussed in the existing literature. In particular, it has been argued that networks trained on flat minima generalize more than when trained on sharp minima. On the other hand, it is also argued that flat minima can be easily converted to sharp minima using a reparametrization. Our results confer the former argument, though in the paradigm of the definition of flatness in the algebraic geometry sense. We also argue that since, in general, loss landscapes quantitatively (and in some cases even qualitatively) change with respect to data unless the "flatness" (however defined) of the minima is an invariant of the data, the existence of flat stationary points may not be crucial for the generalization ability of the network. On the other hand, the existence of an invariance of flatness of minima and saddle points, if proven, may turn out to be crucial in understanding generalization properties.

It should be noted that the existence of flat stationary points directly corresponds to continuous symmetries in the system. Various ways to break these continuous symmetries have been investigated in the literature [132]. The generalized L_2-regularization term essentially perturbs the system to leave only isolated solutions in the system. For example, in [133], it is argued that skip connections in neural networks eliminate singularities as it removes certain symmetries from the system. It may be interesting to study if there is a relation between the generalized L_2-regularization and skipped connections. One can also project the constant zero modes of the Hessian in the computation [132]. From the computational point of view though, the generalized L_2-regularization approach may be the most straightforward way to implement in the current deep learning suites.

By leaving only isolated critical points, the generalized L_2-regularization also opens the door to utilizing efficient and scalable numerical methods, including homotopy methods [100], [103], to locate *all* critical points. Its implication on training robustness is also noteworthy as it ensures that sufficiently small perturbations in training data only cause small changes in critical points while preserving their indices.

Upper Bounds on the Number of Stationary Points and Numerical Results:- Once all the flat stationary points are removed from the gradient equations, the next question we addressed is how many isolated stationary points are there? When the gradient equations are treated as defined over the complex numbers, one can employ many upper bounds on the number of complex solutions for systems of polynomial equations available in the literature, such as the CBB and BKK bounds, to gain insight into the systems. For deep linear networks, the CBB and BKK bounds for modest size networks are given in Table 1.

Using these upper bounds, we employed a numerical algebraic geometry method called the numerical polynomial homotopy continuation method which guarantees to find all isolated complex solutions of such polynomial systems. In our experiments, we generated data matrices X and Y by drawing each of their entries independently from the Gaussian distribution with mean 0 and variance 1 and λ_i's from uniform distributions on $[0, \Lambda_{\max}]$ for various values of $\Lambda_{\max}$ to investigate their impact. Table 1 provides some insight relating bounds on the number of solutions with the actual number of solutions. Moreover, by comparing the

average number of real stationary points with an analytical result, we see that all bounds are orders of magnitudes larger than the actual results showing that deep linear systems are very *sparse*. This conclusion *may or may not* extend to deep nonlinear network as the structure of the corresponding polynomials may different.

Moreover, we showed that the average number of real solutions reduces as perturbation parameters λ_i's vanish, a phenomenon called topology trivialization [127], [128], [129], [130], [131]. They limit to stationary points of the unregularized problem which has flat minima.

We sorted the stationary points by the index (number of negative eigenvalues) of the Hessian matrix and showed that for some samples, *there are indeed local minima which are not global minima* contrary to the available results in the unregularized case. This result is a first for the complete deep linear networks in the regularized case (in fact, Ref. [49], [50] are the only result available for the linear networks with nonzero regularization in a restricted case). There exist many discrete symmetries among solutions, i.e., the value of the loss function at the symmetrically related solutions is equal. We also notice that the stationary points with higher index are rare, which may be due the linearity of the activation functions but may or may not necessarily be a phenomenon when using nonlinear activation functions.

Investigating the loss landscapes when X and Y are correlated, instead of choosing their values from random distributions, may exhibit interesting characteristics of the optimization landscapes as well as impacting the fit of a deep network for a given data set. Extending the algebraic geometry interpretation to deep nonlinear networks will shed further novel insights into the optimization landscapes of these models. Computational implications of the proposed regularization approach specially together with the saddle escaping methods will be an important breakthrough on these theoretical insights.

Finally, the NPHC method we have employed to solve the gradient equations in this paper is known to be embarrassingly parallelizeable, meaning larger systems of equations can still be straightforwardly tackled given enough computational resources. Moreover, the highly structured nature of the stationary points shown in Table 3 and Section 5.3.6 suggest that efficient methods could be developed to exploit the structure in the solution set. The combination of parallelizability and a custom-made homotopy exploiting structure can then be deployed on larger-scale models comparable to the ones being used in real-world applications recently.

ACKNOWLEDGMENT

The first author was at United Technologies Research Center, USA, when a major part of the work for this paper was completed. The second author acknowledges support from NSF grant DMS-1923099. The last two authors acknowledge support from grants NSF CCF 1812746 and ONR N00014-16-1-2722. The last author also acknowledges support from ARO W911NF-20-2-0218.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning," *MIT Press.*, 2015.

[3] A. Blum and R. Rivest, "Training a 3-node neural network is np-complete," in *Proceedings of the 1st International Conference on Neural Information Processing Systems*, pp. 494–501, MIT Press, 1988.

[4] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural networks*, vol. 2, no. 1, pp. 53–58, 1989.

[5] M. Gori and A. Tesi, "On the problem of local minima in backpropagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 1, pp. 76–86, 1992.

[6] X.-H. Yu and G.-A. Chen, "On the local minima free condition of backpropagation learning," *IEEE Transactions on Neural Networks*, vol. 6, no. 5, pp. 1300–1303, 1995.

[7] A. Saxe, J. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *arXiv preprint arXiv:1312.6120*, 2013.

[8] Q. Nguyen and M. Hein, "The loss surface of deep and wide neural networks," *arXiv preprint arXiv:1704.08045*, 2017.

[9] I. Goodfellow, O. Vinyals, and A. Saxe, "Qualitatively characterizing neural network optimization problems," *arXiv preprint arXiv:1412.6544*, 2014.

[10] A. Andoni, R. Panigrahy, G. Valiant, and L. Zhang, "Learning polynomials with neural networks," in *International Conference on Machine Learning*, pp. 1908–1916, 2014.

[11] D. Soudry and Y. Carmon, "No bad local minima: Data independent training error guarantees for multilayer neural networks," *arXiv preprint arXiv:1605.08361*, 2016.

[12] J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, and B. Recht, "First-order Methods Almost Always Avoid Saddle Points," *ArXiv e-prints*, Oct. 2017.

[13] A. Choromanska, M. Henaff, M. Mathieu, G. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," *arXiv preprint arXiv:1412.0233*, 2014.

[14] K. Kawaguchi, "Deep learning without poor local minima," *arXiv preprint arXiv:1605.07110*, 2016.

[15] D. Soudry and E. Hoffer, "Exponentially vanishing sub-optimal local minima in multilayer neural networks," *arXiv preprint arXiv:1702.05777*, 2017.

[16] L. Sagun, V. Guney, G. Arous, and Y. LeCun, "Explorations on high dimensional landscapes," *arXiv preprint arXiv:1412.6615*, 2014.

[17] D. Mehta, X. Zhao, E. A. Bernal, and D. J. Wales, "The loss surface of xor artificial neural networks," *Preprint*, 2018.

[18] F. M. Coetzee and V. L. Stonick, "488 solutions to the xor problem," *Advances in Neural Information Processing Systems*, pp. 410–416, 1997.

[19] G. Swirszcz, W. M. Czarnecki, and R. Pascanu, "Local minima in training of neural networks," *stat*, vol. 1050, p. 17, 2017.

[20] A. Ballard, S. Martiniani, D. Mehta, J. Stevenson, and D. J. Wales, "Energy landscapes for machine learning," *Phys. Chem. Chem. Phys.*, vol. 19, pp. 12585–12603, 2017.

[21] C. Yun, S. Sra, and A. Jadbabaie, "A critical view of global optimality in deep learning," *arXiv preprint arXiv:1802.03487*, 2018.

[22] E. D. Sontag and H. J. Sussmann, "Backpropagation can give rise to spurious local minima even for networks without hidden layers," *Complex Systems*, vol. 3, no. 1, pp. 91–106, 1989.

[23] L. Wu, Z. Zhu, *et al.*, "Towards understanding generalization of deep learning: Perspective of loss landscapes," *arXiv preprint arXiv:1706.10239*, 2017.

[24] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generaliation in deep learning," *arXiv preprint arXiv:1710.05468*, 2017.

[25] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in neural information processing systems*, pp. 2933–2941, 2014.

[26] D. J. Im, M. Tao, and K. Branson, "An empirical analysis of the optimization of deep network loss surfaces," *arXiv preprint arXiv:1612.04010*, 2016.

[27] A. R. Sankar and V. N. Balasubramanian, "Are saddles good enough for deep learning?," *arXiv preprint arXiv:1706.02052*, 2017.

[28] J. P. K. Doye and D. J. Wales, "Saddle points and dynamics of lennard-jones clusters, solids, and supercooled liquids," *J. Chem. Phys.*, vol. 116, no. 9, pp. 3777–3788, 2002.

[29] J. P. K. Doye and D. J. Wales, "Saddle points and dynamics of Lennard-Jones clusters, solids, and supercooled liquids," *Journal of Chem. Phys.*, vol. 116, pp. 3777–3788, 2002.

[30] D. J. Wales, "Some further applications of discrete path sampling to cluster isomerization," *Mol. Phys.*, vol. 102, pp. 891–908, 2004.

[31] D. Mehta, "Finding All the Stationary Points of a Potential Energy Landscape via Numerical Polynomial Homotopy Continuation Method," *Phys.Rev.*, vol. E84, p. 025702, 2011.

[32] D. Mehta, C. Hughes, M. Kastner, and D. Wales, "Potential energy landscape of the two-dimensional xy model: Higher-index stationary points," *The Journal of chemical physics*, vol. 140, no. 22, p. 224503, 2014.

[33] C. Huges, D. Mehta, and D. J. Wales, "An Inversion-Relaxation Approach for Sampling Stationary Points of Spin Model Hamiltonians," *J. Chem. Phys.*, vol. 140, p. 194104, 2014.

[34] D. Mehta, D. A. Stariolo, and M. Kastner, "Energy landscape of the finite-size spherical three-spin glass model," *Phys.Rev.*, vol. E87, no. 5, p. 052143, 2013.

[35] D. Mehta, N. S. Daleo, F. Dörfler, and J. D. Hauenstein, "Algebraic geometrization of the kuramoto model: Equilibria and stability analysis," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 5, p. 053103, 2015.

[36] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points — online stochastic gradient for tensor decomposition," in *Proceedings of The 28th Conference on Learning Theory* (P. Grünwald, E. Hazan, and S. Kale, eds.), vol. 40 of *Proceedings of Machine Learning Research*, (Paris, France), pp. 797–842, PMLR, 03–06 Jul 2015.

[37] Y. Nesterov and B. T. Polyak, "Cubic regularization of newton method and its global performance," *Mathematical Programming*, vol. 108, no. 1, pp. 177–205, 2006.

[38] A. Anandkumar and R. Ge, "Efficient approaches for escaping higher order saddle points in non-convex optimization," in *Conference on Learning Theory*, pp. 81–102, 2016.

[39] I. Panageas and G. Piliouras, "Gradient descent converges to minimizers: The case of non-isolated critical points," *CoRR, abs/1605.00405*, 2016.

[40] P. F. Baldi and K. Hornik, "Learning in linear neural networks: A survey," *IEEE Transactions on neural networks*, vol. 6, no. 4, pp. 837–858, 1995.

[41] H. Lu and K. Kawaguchi, "Depth creates no bad local minima," *arXiv preprint arXiv:1702.08580*, 2017.

[42] Z. Zhu, D. Soudry, Y. C. Eldar, and M. B. Wakin, "The Global Optimization Geometry of Shallow Linear Neural Networks," *ArXiv e-prints*, May 2018.

[43] P. Baldi and Z. Lu, "Complex-valued autoencoders," *Neural Networks*, vol. 33, pp. 136–147, 2012.

[44] M. Hardt and T. Ma, "Identity matters in deep learning," *arXiv preprint arXiv:1611.04231*, 2016.

[45] C. Yun, S. Sra, and A. Jadbabaie, "Global optimality conditions for deep neural networks," *arXiv preprint arXiv:1707.02444*, 2017.

[46] Y. Zhou and Y. Liang, "Critical points of neural networks: Analytical forms and landscape properties," *arXiv preprint arXiv:1710.11205*, 2017.

[47] Y. Zhou and Y. Liang, "Characterization of gradient dominance and regularity conditions for neural networks," *arXiv preprint arXiv:1710.06910*, 2017.

[48] Y. Bansal, M. Advani, D. D. Cox, and A. M. Saxe, "Minnorm training: an algorithm for training over-parameterized deep neural networks," *ArXiv e-prints*, June 2018.

[49] A. Taghvaei, J. W. Kim, and P. Mehta, "How regularization affects the critical points in linear networks," in *Advances in Neural Information Processing Systems*, pp. 2499–2509, 2017.

[50] C. D. Freeman and J. Bruna, "Topology and geometry of half-rectified network optimization," *arXiv preprint arXiv:1611.01540*, 2016.

[51] D. A. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[52] D. A. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998.

[53] S. Watanabe, *Algebraic geometry and statistical learning theory*, vol. 25. Cambridge University Press, 2009.

[54] Z. Huang, M. England, D. Wilson, J. H. Davenport, and L. C. Paulson, "Using machine learning to improve cylindrical algebraic decomposition," *arXiv preprint arXiv:1804.10520*, 2018.

[55] D.-S. Huang, H. H. Ip, and Z. Chi, "A neural root finder of polynomials based on root moments," *Neural Computation*, vol. 16, no. 8, pp. 1721–1762, 2004.

[56] S. Perantonis, N. Ampazis, S. Varoufakis, and G. Antoniou, "Constrained learning in neural networks: Application to stable factorization of 2-d polynomials," *Neural Processing Letters*, vol. 7, no. 1, pp. 5–14, 1998.

[57] B. Mourrain, N. G. Pavlidis, D. K. Tasoulis, and M. N. Vrahatis, "Determining the number of real roots of polynomials through neural networks," *Computers & Mathematics with Applications*, vol. 51, no. 3-4, pp. 527–536, 2006.

[58] C. Knoll, D. Mehta, T. Chen, and F. Pernkopf, "Fixed points of belief propagation—an analysis via polynomial homotopy continuation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 2124–2136, Sept 2018.

[59] C. Knoll and F. Pernkopf, "On loopy belief propagation–local stability analysis for non-vanishing fields," in *Uncertainty in Artificial Intelligence*, 2017.

[60] G. M. Georgiou and C. Koutsougeras, "Complex domain backpropagation," *IEEE transactions on Circuits and systems II: analog and digital signal processing*, vol. 39, no. 5, pp. 330–334, 1992.

[61] R. S. Zemel, C. K. Williams, and M. C. Mozer, "Lending direction to neural networks," *Neural Networks*, vol. 8, no. 4, pp. 503–512, 1995.

[62] T. Kim and T. Adalı, "Approximation by fully complex multilayer perceptrons," *Neural computation*, vol. 15, no. 7, pp. 1641–1666, 2003.

[63] T. Nitta, "An extension of the back-propagation algorithm to complex numbers," *Neural Networks*, vol. 10, no. 8, pp. 1391–1415, 1997.

[64] H. Akira, *Complex-valued neural networks: theories and applications*, vol. 5. World Scientific, 2003.

[65] D. P. Reichert and T. Serre, "Neuronal synchrony in complex-valued deep networks," *arXiv preprint arXiv:1312.6115*, 2013.

[66] N. Guberman, "On complex valued convolutional neural networks," *arXiv preprint arXiv:1602.09046*, 2016.

[67] I. Danihelka, G. Wayne, B. Uria, N. Kalchbrenner, and A. Graves, "Associative long short-term memory," *arXiv preprint arXiv:1602.03032*, 2016.

[68] S. Wisdom, T. Powers, J. Hershey, J. Le Roux, and L. Atlas, "Full-capacity unitary recurrent neural networks," in *Advances in Neural Information Processing Systems*, pp. 4880–4888, 2016.

[69] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," *arXiv preprint arXiv:1705.09792*, 2017.

[70] T. Nitta, "Solving the xor problem and the detection of symmetry using a single complex-valued neuron," *Neural Networks*, vol. 16, no. 8, pp. 1101–1105, 2003.

[71] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Transactions on Neural Networks and learning systems*, vol. 23, no. 4, pp. 541–551, 2012.

[72] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," in *International Conference on Machine Learning*, pp. 1120–1128, 2016.

[73] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997.

[74] S. Hochreiter and J. Schmidhuber, "Simplifying neural nets by discovering flat minima," in *Advances in neural information processing systems*, pp. 529–536, 1995.

[75] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[76] S. Jastrzk ebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. J. Storkey, "Finding flatter minima with sgd," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*, OpenReview.net, 2018.

[77] S. Jastrzębski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey, "Three factors influencing minima in sgd," *arXiv preprint arXiv:1711.04623*, 2017.

[78] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. T. Chayes, L. Sagun, and R. Zecchina, "Entropy-sgd: Biasing gradient descent into wide valleys," *CoRR*, vol. abs/1611.01838, 2016.

[79] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, "Local entropy as a measure for sampling solutions in constraint satisfaction problems," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2016, no. 2, p. 023301, 2016.

[80] C. Baldassi, F. Gerace, C. Lucibello, L. Saglietti, and R. Zecchina, "Learning may need only a few bits of synaptic precision," *Physical Review E*, vol. 93, no. 5, p. 052313, 2016.

[81] C. Baldassi, C. Borgs, J. T. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, "Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes," *Proceedings of the National Academy of Sciences*, vol. 113, no. 48, pp. E7655–E7662, 2016.

[82] Y. Zhang, A. M. Saxe, M. S. Advani, and A. A. Lee, "Energy–entropy competition and the effectiveness of stochastic gradient descent in machine learning," *Molecular Physics*, pp. 1–10, 2018.

[83] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," *arXiv preprint arXiv:1703.04933*, 2017.

[84] R. Brockett, "Some geometric questions in the theory of linear systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 4, pp. 449–455, 1976.

[85] A. M. Chen, H.-M. Lu, and R. Hecht-Nielsen, "On the geometry of feedforward neural network error surfaces," *Neural computation*, vol. 5, no. 6, pp. 910–927, 1993.

[86] D. Saad and S. A. Solla, "On-line learning in soft committee machines," *Physical Review E*, vol. 52, no. 4, p. 4225, 1995.

[87] V. Kůrková and P. C. Kainen, "Functionally equivalent feedforward neural networks," *Neural Computation*, vol. 6, no. 3, pp. 543–558, 1994.

[88] S.-I. Amari, H. Park, and T. Ozeki, "Singularities affect dynamics of learning in neuromanifolds," *Neural computation*, vol. 18, no. 5, pp. 1007–1065, 2006.

[89] S. Watanabe, "Almost all learning machines are singular," in *Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symposium on*, pp. 383–388, IEEE, 2007.

[90] H. J. Sussmann, "Uniqueness of the weights for minimal feedforward nets with a given input-output map," *Neural networks*, vol. 5, no. 4, pp. 589–593, 1992.

[91] B. Neyshabur, R. R. Salakhutdinov, and N. Srebro, "Path-sgd: Path-normalized optimization in deep neural networks," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 2422–2430, Curran Associates, Inc., 2015.

[92] L. Sagun, L. Bottou, and Y. LeCun, "Eigenvalues of the hessian in deep learning: Singularity and beyond," *arXiv preprint arXiv:1611.07476*, 2016.

[93] L. Sagun, U. Evci, V. U. Guney, Y. Dauphin, and L. Bottou, "Empirical analysis of the hessian of over-parametrized neural networks," *arXiv preprint arXiv:1706.04454*, 2017.

[94] R. Nerattini, M. Kastner, D. Mehta, and L. Casetti, "Exploring the energy landscape of XY models," *Phys.Rev.*, vol. E87, no. 3, p. 032140, 2013.

[95] R. H. Abraham and J. W. Robbin, *Transversal Mappings and Flows*. Benjamin, 1967.

[96] S. N. Chow, J. Mallet-Paret, and J. A. Yorke, "Finding zeroes of maps: homotopy methods that are constructive with probability one," *Mathematics of Computation*, vol. 32, no. 143, pp. 887–899, 1978.

[97] E. L. Allgower and K. Georg, *Introduction to Numerical Continuation Methods*. John Wiley & Sons, New York, 1979.

[98] A. P. Morgan and A. J. Sommese, "Coefficient-parameter polynomial continuation," *Applied Mathematics and Computation*, vol. 29, no. 2, pp. 123–160, 1989.

[99] D. Bates, D. A. Brake, J. Hauenstein, A. J. Sommese, and C. Wampler, "Homotopies for connected components of algebraic sets with application to computing critical sets," in *Mathematical Aspects of Computer and Information Sciences* (J. Blömer, I. Kotsireas, T. Kutsia, and D. Simos, eds.), (Cham), pp. 107–120, Springer International Publishing, 2017.

[100] A. J. Sommese and C. W. Wampler, *The numerical solution of systems of polynomials arising in Engineering and Science*. World Scientific Publishing Company, 2005.

[101] T.-Y. Li, "Numerical Solution of Polynomial Systems by Homotopy Continuation," in *Handbook of Numerical Analysis: Special Volume: Foundations of Computational Mathematics* (P. G. Ciarlet, ed.), vol. 11, p. 470, North-Holland, 2003.

[102] E. L. Allgower and K. Georg, *Introduction to numerical continuation methods*, vol. 45. Society for Industrial and Applied Mathematics, 2003.

[103] D. Bates, J. Hauenstein, A. Sommese, and C. Wampler, *Numerically solving polynomial systems with Bertini*, vol. 25. SIAM, 2013.

[104] D. N. Bernshtein, "The number of roots of a system of equations," *Functional Analysis and its Applications*, vol. 9, no. 3, pp. 183–185, 1975.

[105] B. Huber and B. Sturmfels, "A polyhedral method for solving sparse polynomial systems," *Mathematics of Computation*, vol. 64, no. 212, pp. 1541–1555, 1995.

[106] T. Y. Li and X. Wang, "The BKK root count in C^n," *Mathematics of Computation*, vol. 65, no. 216, pp. 1477–1485, 1996.

[107] M. J. Rojas and X. Wang, "Counting affine roots of polynomial systems via pointed Newton polytopes," *Journal of Complexity*, vol. 12, pp. 116–133, jun 1996.

[108] J.-P. Dedieu and G. Malajovich, "On the number of minima of a random polynomial," *Journal of Complexity*, vol. 24, no. 2, pp. 89–108, 2008.

[109] M. Kac, "On the average number of real roots of a random algebraic equation," *Bull. Am. Math. Soc.*, vol. 49, no. 938, p. 314–320, 1943.

[110] M. Kac, "On the average number of real roots of a random algebraic equation (ii)," *Proceedings of the London Mathematical Society*, vol. 2, no. 1, pp. 390–408, 1948.

[111] A. Edelman and E. Kostlan, "How many zeros of a random polynomial are real?," *Bulletin of the American Mathematical Society*, vol. 32, no. 1, pp. 1–37, 1995.

[112] L. Blum, "F. cuck er, m. shub, and s. smale. complexity and real computation," 1998.

[113] E. Kostlan, "On the expected number of real roots of a system of random polynomial equations," in *Foundations of computational mathematics*, pp. 149–188, World Scientific, 2002.

[114] J.-M. Azaïs and M. Wschebor, "On the roots of a random system of equations. the theorem of shub and smale and some extensions," *Foundations of Computational Mathematics*, vol. 5, no. 2, pp. 125–144, 2005.

[115] D. Armentano, M. Wschebor, *et al.*, "Random systems of polynomial equations. the expected number of roots under smooth analysis," *Bernoulli*, vol. 15, no. 1, pp. 249–266, 2009.

[116] Q. Liao and T. Poggio, "Theory ii: Landscape of the empirical risk in deep learning," *arXiv preprint arXiv:1703.09833*, 2017.

[117] S. Basu, R. Pollack, and M. F. Roy, *Algorithms in Real Algebraic Geometry*. Springer, 2003.

[118] F. M. Coetzee and V. Stonick, "Homotopy approaches for the analysis and solution of neural network and other nonlinear systems of equations," *Doctoral Thesis, Carnegie Mellon University, May*, 1995.

[119] F. M. Coetzee and V. L. Stonick, "On a natural homotopy between linear and nonlinear single-layer networks," *IEEE Transactions on Neural Networks*, vol. 7, pp. 307–317, March 1996.

[120] H. Ninomiya, C. Tomita, and H. Asai, "An efficient learning algorithm for finding multiple solutions based on fixed-point homotopy method," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, pp. 978–983 vol. 2, July 2005.

[121] J. C. Chow, L. Udpa, and S. Udpa, "New training algorithm for neural networks," in *Review of Progress in Quantitative Nondestructive Evaluation*, pp. 685–691, Springer, 1992.

[122] H. Mobahi and J. W. Fisher, "On the link between gaussian homotopy continuation and convex envelopes," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 43–56, Springer, 2015.

[123] A. Anandkumar, Y. Deng, R. Ge, and H. Mobahi, "Homotopy analysis for tensor pca," in *Conference on Learning Theory*, pp. 79–104, 2017.

[124] A. Sommese and C. Wampler, *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific Publishing, Hackensack, NJ, 2005.

[125] T. Y. Li, "Solving polynomial systems by the homotopy continuation method," *Handbook of numerical analysis*, vol. XI, pp. 209–304, 2003.

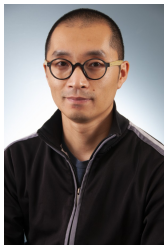
[126] A. Morgan and A. J. Sommese, "Computing all solutions to polynomial systems using homotopy continuation," *Applied Mathematics and Computation*, vol. 24, no. 2, pp. 115–138, 1987.

[127] M. Kastner and D. Mehta, "Phase Transitions Detached from Stationary Points of the Energy Landscape," *Phys.Rev.Lett.*, vol. 107, p. 160602, 2011.

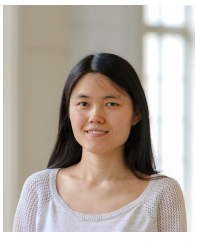
[128] D. Mehta, J. D. Hauenstein, and M. Kastner, "Energy-landscape analysis of the two-dimensional nearest-neighbor ϕ^4 model," *Phys. Rev. E*, vol. 85, p. 061103, Jun 2012.

[129] Y. V. Fyodorov, "High-dimensional random fields and random matrix theory," *arXiv preprint arXiv:1307.2379*, 2013.
[130] D. Mehta, J. D. Hauenstein, M. Niemerg, N. J. Simm, and D. A. Stariolo, "Energy landscape of the finite-size mean-field 2-spin spherical model and topology trivialization," *Phys. Rev. E*, vol. 91, p. 022133, Feb 2015.
[131] P. Chaudhari and S. Soatto, "Trivializing the energy landscape of deep networks," *arXiv preprint arXiv:1511.06485*, 2015.
[132] D. J. Wales, *Energy Landscapes*. Cambridge: Cambridge University Press, 2003.
[133] A. E. Orhan and X. Pitkow, "Skip connections eliminate singularities," *arXiv preprint arXiv:1701.09175*, 2017.

Dhagash Mehta was conferred his PhD degree in theoretical physics from the University of Adelaide, Australia, in 2011. He held various research postions at the National University of Ireland Maynooth, Syracuse University, The University of Cambridge, North Carolina State University, Simons Institute for the Theory of Computing at Berkeley and Fields Institute Toronto and the University of Notre Dame. He was a senior research scientist at the United Technologies Research Center, USA, before moving to The Vanguard Group, PA, USA, where he is a Senior Manager, Asset Allocation and Machine Learning. His research areas are machine learning, non-convex optimization, network science and applied algebraic geometry with their applications arising in finance, science and engineering areas.



Tianran Chen received his PhD degree in applied mathematics from the Michigan State University, USA, in 2012. He also worked as a postdoctoral research instructor at Michigan State University. He is currently an assistant professor in the Department of Mathematics at Auburn University at Montgomery. His research interests include numerical algebraic geometry, numerical analysis, and scientific computing.



Tingting Tang received her PhD degree in mathematics from University of Louisiana at Lafayette, USA, in 2017. She then spent two years at the University of Notre Dame as a postdoctoral researcher. She is currently an assistant professor with a joint position in the Department of Mathematics and Statistics at San Diego State University and their Imperial Valley Campus. Her research interests include math biology, numerical analysis, and the application of numerical algebraic geometry in optimizations.

Jonathan D. Hauenstein received the PhD degree in mathematics from the University of Notre Dame, USA, in 2009. He held various positions at the Fields Institute, Texas A&M University, Mittag-Leffler Institute, North Carolina State University, and the Simons Institute for the Theory of Computing before returning to the University of Notre Dame in 2014. He is currently a professor with the Department of Applied and Computational Mathematics and Statistics. His research interests include numerical algebraic geometry and its applications involving a variety of fields in science and engineering.