

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 the deep linear neural network models. After clarifying 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. Using these upper bounds and utilizing a numerical algebraic geometry method, we find *all* stationary points for modest depth and matrix size. We show that in the presence of the non-zero regularization, deep linear networks indeed possess local minima which are not the global minima. Our computational results further clarify certain aspects of the loss surfaces of deep linear networks and provides novel insights.

I. 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 is yet to be 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 Hamiltonian of a physics model called the spherical p -spin model and analyzed using random matrix theory and statistical physics techniques. It was 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 may 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 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 down the learning. This is a typical nature of 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 [38], [12], [39].

The ability of a minimization routine to find the global minimum also depends to the structure of the loss surface. In [20], the loss surface of a single hidden layer feedforward neural network was shown to have a *single funnel* structure, i.e.,

D. Mehta is with the Autonomous and Intelligent Systems Department, United Technologies Research Center, East Hartford, CT, USA. e-mail: dhagashmehta@gmail.com

T. Chen is with the Department Mathematics and Computer Science, Auburn University at Montgomery, Montgomery, AL, USA e-mail: ti@nranchen.org.

T. Tang and J.D. Hauenstein are with Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN, USA. e-mail: {ttang,hauenstein}@nd.edu

asymmetric barrier heights between two adjacent local minima, ensuring that the routine quickly relaxes downhill towards the global minimum [40].

There are analytical and numerical results that either achieve the global minimum or construct necessary and sufficient conditions for a point to be the global minimum for restricted classes of deep networks [8], [41], [42], [43], [44], [45].

In [4], a detailed mathematical analysis of a class of a simpler model, the deep linear networks, was performed. Since then, the model has become one of the ideal test-grounds of ideas in artificial neural networks and deep learning [46]. Below we first briefly describe the formulation of the model and then describe the previous results.

A. 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 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], [46]

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

with

$$\bar{\mathcal{L}}(W) = \frac{1}{2} \sum_{i=1}^m \left\| (W_{H+1} W_H \cdots W_1 X)_{\cdot, i} - Y_{\cdot, i} \right\|_2^2, \quad (2)$$

where $\|\cdot\|$ is the vector norm, W_i is the weight matrix for the i^{th} layer with hidden layers from $i = 1, \dots, H$ 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 $W_1 \in \mathbb{R}^{d_1 \times d_x}$, $W_2 \in \mathbb{R}^{d_2 \times d_1}$, \dots , $W_{H+1} \in \mathbb{R}^{d_y \times d_H}$. We also denote $k = \min(d_H, \dots, d_1)$, i.e., the number of neurons in the hidden layer with the smallest width. The number of weights, or variables, is $n = d_x d_1 d_2 \cdots d_H d_y$.

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.

B. Earlier Works on Loss Surfaces of Deep Linear Networks

Almost all the existing results for deep linear networks are for the $\lambda = 0$ case. For this case, deep linear network

with $H = 1$, under the assumptions that (1) XX^T and XY^T are invertible matrices, (2) $\Sigma = YX^T(XX^T)^{-1}XY^T$ has d_y distinct eigenvalues, (3) $d_x = d_y$, i.e., an autoencoder, and (4) $k < d_x, d_y$, 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], [47], [48], for deep linear complex-valued autoencoders with one hidden layer [49], as well as for deep linear residual networks [50], [21]). Additionally, [43] provides several necessary and sufficient conditions on global optimality based on rank conditions on the W_i matrices for deep linear networks.

In [51] (cf. [52]), analytical forms of the stationary points (including minima) characterizing the values of the loss function were presented for deep linear 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 [53]) and was concluded that the learning speed can remain finite even in the $H \rightarrow \infty$ limit for a special class of initial conditions on the weights, likely due to no local minima present in the landscape.

The $\lambda > 0$ case is considered in [54] 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. Here, the 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.

C. Our Contribution

The main conceptual contribution in this paper is to identify solving the gradients of deep networks as a computational algebraic geometry [55], [56] problem. 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 on various definitions of *flat* minima, and distinguish the geometric definition of flat minima from the other definitions. We then show their existence in the unregularized landscapes of deep linear networks.
- 2) Then, we prove that a straightforward extension of L_2 regularization can guarantee to remove all these 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.

- 3) We take up a novel question on deep learning loss surfaces: how many isolated stationary 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. Obviously, these upper bounds provide strict upper bounds on the number of local minima.
- 4) Next, we custom-make a numerical algebraic geometry method which guarantees to find all stationary points of the deep linear networks for modest size systems. With all stationary points at hand, we obtain further novel insights on the loss landscapes of the deep linear networks, in addition to explicitly showing that the model exhibits local minima which are *not* global minima for the regularized case.

The remainder of the paper is organized as follows: in Section II, a brief introduction of algebraic geometry is provided, and a relation between algebraic geometry and deep linear networks is discussed. We also put our approach in perspective with respect to other attempts to apply algebraic geometry methods to machine learning. In Section III, we show how the flat stationary points of unregularized gradient equations can be removed using a generalized regularization. In Section IV, we provide algebraic geometry based upper bounds on number of stationary points of the gradient equations. In Section V, we introduce and apply the polynomial homotopy continuation method and provide results for modest size systems. In Section VI, we discuss our findings in more details and conclude.

II. ALGEBRAIC GEOMETRIC INTERPRETATION OF DEEP LINEAR NETWORKS

In this section, we argue that solving the gradient equations of deep linear networks can be viewed as an algebraic geometry problem, and briefly introduced algebraic geometry terminologies while distinguishing our algebraic geometry interpretation of the problem with previous attempts.

In this paper, we identify solving the gradient equations of deep learning as an algebraic geometry [55], [56] problem. There have been various attempts to relate algebraic geometry and machine learning, e.g., an abstract relation between statistical learning methods and algebraic geometry has been extensively investigated [57]. Machine learning methods have been used to improve computational algebraic geometry methods such as in computing cylindrical algebraic decomposition [58] and to find roots of certain polynomials [59], [60], [61]. Neural networks have also been shown to effectively learn data whose target function is a polynomial [10] (see also [62], [63]). In the present paper, on the other hand, 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, as well as for

arbitrary data and size of all the matrices, equally well. 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.

A. The Gradient Equations are Algebraic Equations

The critical points of the objective function L are points at which all partial derivatives are equal to zero, i.e., satisfy the gradient equations $\nabla L = \mathbf{0}$. This 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.

Let $W = W_{H+1} \dots W_1$, $U_i^\top = \prod_{j=i+1}^{H+1} W_j^\top$, 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 . Hence,

$$\frac{\partial L}{\partial W_i} = U_i^\top \left(W \left(\sum_{k=1}^m \mathbf{x}_k \mathbf{x}_k^\top \right) - \left(\sum_{k=1}^m \mathbf{y}_k \mathbf{x}_k^\top \right) \right) V_i^\top + \frac{\lambda}{2} W_i. \quad (3)$$

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

B. 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 [55], [56] 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 $f(x) = (f_1(x), \dots, f_m(x))$, where $x = (x_1, \dots, x_n)$ in \mathbb{C}^n is

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

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 algebraic sets $\emptyset \subsetneq B_1, B_2 \subsetneq A$ 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 dimension 0 irreducible algebraic set A 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 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*.

C. 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 [64], [65], [66], [67], [68] and have gained renewed interest in deep learning [69], [70], [71], [72], [73] for its use in simultaneously modelling phase and amplitude data. In particular, back-propagation for complex-valued neural networks was developed in [67]. In [74], it was shown that the XOR data which cannot be solved with a single real-valued neuron in the hidden layer, can be solved with a complex-valued network. Such complex-valued neural networks were shown to have better generalization characteristics [75], and faster learning [76], in addition to biological motivations [69].

We note that in the aforementioned formulation of the 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 and 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 for the computational analysis purpose.

III. FLAT STATIONARY POINTS AND REGULARIZATION

In this section, we briefly review the existing literature on flat minima in deep learning. In [77], [78], 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 [79], it was empirically shown that SGD based methods tend to converge to sharp (flat) minima with large (small) batch sizes. In [80], [81], 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 [82], an entropy-SGD was proposed that actively bias the optimization towards flat minima of specific widths (cf. [83], [84], [85], [86]). However, later on, in [87], 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 the exactly flat saddles and minima, i.e., the connected components of the stationary points on which the loss function is precisely constant, whose existence is well-known since the works of [88], [89], [90], [91] (see [92] for a review). Such degenerate regions, sometimes referred to as *neuromanifolds*, are quite common [93] not only in artificial neural networks loss landscapes due to

various symmetries [94], [89], [95], [95] of the corresponding loss functions. At such solutions, the Fischer information matrix tends to be singular and the 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 at 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 a 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 [96], 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 moving in which leads to no change in the loss function. In [97], 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 [97] between almost flat minima within which the loss function is almost constant, and the flat minima within which the loss function is precisely constant. The former should be referred to as the “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 because in the next section where we devise a method to remove positive-dimensional stationary solutions, we remove all complex and real positive-dimensional stationary points.

We present a few examples to show some explicit results. The first example arise 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 satisfies $\nabla(f) = 0$, 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, one has $f(x, y, z) = -1$.

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 solutions of Eqs. (3) consists of three irreducible components: a point is an isolated saddle, a curve consisting of flat saddle points, and a curve consisting of flat minima described as follows. 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

$$W_1 = \alpha \cdot \widehat{W}_1 \quad \text{and} \quad W_2 = \alpha^{-1} \cdot \widehat{W}_2$$

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

$$\widehat{W}_1 = [1 \quad 9.6330] \quad \text{and} \quad \widehat{W}_2 = \begin{bmatrix} 0.0206 \\ -0.0180 \end{bmatrix}$$

while the flat minima approximately have

$$\widehat{W}_1 = [1 \quad 0.0696] \quad \text{and} \quad \widehat{W}_2 = \begin{bmatrix} 0.2664 \\ 0.3045 \end{bmatrix}.$$

Remark 1. *This example generalizes to all critical points in the unregularized case, i.e., $\lambda = 0$. That is, if (W_1, \dots, W_{H+1}) is a critical point, then so is $(A_1 W_1, A_2 W_2 A_1^{-1}, \dots, W_{H+1} A_H^{-1})$. 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 λ (Eq. 1) is also not necessarily enough to remove the the flat stationary points. The following provides a simple illustrative example showing that this need not be the case.

Example-3: For $H = 1$, $m = 3$, and $d_x = d_y = d_1 = 2$, we consider

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

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

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

where

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

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

A. 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 and tanh activation functions, 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 [98], a spin glass model called the XY model was found to exhibit residual

continuous symmetries, and a generalized regularization term was used to remove the continuous symmetries.

As outlined above, the existence of flat or degenerate critical set is a very common phenomenon in the study of deep linear and nonlinear networks in general. 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 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 introduces many unnecessary difficulties: For example, a simple solver based on Newton's iterations will 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 non-degenerate (isolated) critical points. That is, such perturbation would remove the flatness from all critical points.

Recall that for a smooth function $f : \mathbf{R}^n \rightarrow \mathbf{R}^n$, $\mathbf{v} \in \mathbf{R}^n$ is said to be a *regular value* if for each $\mathbf{x} \in \mathbf{R}^n$ such that $f(\mathbf{x}) = \mathbf{v}$, the Jacobian matrix Df is nonsingular at \mathbf{x} . **Sard's Theorem** [99] states that almost all $\mathbf{v} \in \mathbf{R}^n$ 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 discussions: Let $f(\mathbf{a}, \mathbf{x}) : \mathbf{R}^m \times \mathbf{R}^n \rightarrow \mathbf{R}^n$ be a smooth function. **Generalized Sard's Theorem** [99] states that if $\mathbf{0}$ is a regular value for f , then for almost all $\mathbf{a} \in \mathbf{R}^m$, $\mathbf{0}$ 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 the Generalized Sard's Theorem, we devise a regularization for the deep linear networks: Given $H + 1$ matrices with positive real entries $\Lambda = (\Lambda_1, \dots, \Lambda_{H+1})$ with each Λ_i having the same size as W_i , we can consider a generalized Tikhonov regularization of $\bar{\mathcal{L}}$ given by

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

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 $\lambda_{i,j,k}$'s, the entries of Λ_i 's, are small positive real numbers that serve as penalty coefficients. Therefore the minimization problem for \mathcal{L}^Λ 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

$$\frac{\partial \mathcal{L}^\Lambda}{\partial W_i} = U_i^\top (W X X^\top - Y X^\top) V_i^\top + \Lambda_i \circ W_i. \quad (4)$$

When 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 will be 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} . Here, we include the parameters — coefficients in Λ as variables.

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

The Jacobian matrix of F is a $m \times 2m$ 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 $F : (\mathbb{R}^*)^m \times \mathbb{R}^m \rightarrow \mathbb{R}^m$. Then by the Generalized Sard’s Theorem [99], for almost all choices of $\Lambda \in \mathbb{R}^m$, $\mathbf{0}$ is a regular value for the map $F_\Lambda : (\mathbb{R}^*)^m \rightarrow \mathbb{R}^m$ 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 *isolated* (a.k.a. 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 becomes only 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, and the dense cases of Theorem 1 will be the special case that require all entries to be 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 a $m \times 2m$ 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 the 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 method* [100] 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 zero, certain coordinates in some of the critical point of $\bar{\mathcal{L}}^\Lambda$ grow unboundedly.

Remark 3. *More rigorous description of this phenomenon of diverging solutions can be given in terms of projective space [56], [55] which encapsulate 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 [101] 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 example 1.

Example-4: The gradient of $f(x, y, z) = 2xy + 2xz - 2x - y - z + (\frac{2}{1000}x^2 + \frac{1}{1000}y^2 + \frac{3}{1000}z^2)$ is $\nabla(f) = \{\frac{x}{250} + 2y + 2z - 2, 2x + \frac{y}{500} - 1, 2x + \frac{3}{500}z - 1\}$. The set of stationary points, which satisfies $\nabla(f) = 0$, is $x \sim 0.49925, y \sim 0.74925, z \sim 0.24975$, and the dimension of the solution is 0.

Remark 4. *There have been various methods proposed which escapes 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 [38], [12], [39]. With the generalized L_2 -regularization, only the former set of methods may be required to escape saddles and achieve a deeper minimum.*

IV. ESTIMATES ON THE NUMBER OF ISOLATED SOLUTIONS

Now that after 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).

A. Upper Bounds on the Number of Isolated Stationary Points

Algebraic geometry interpretation of the gradient systems of the deep linear networks allows us to utilize different theorems on root-counts of the number of complex solutions to estimate number of stationary points of the system. To that end, suppose that $f(x) = (f_1(x), \dots, f_n(x))$ is a polynomial system where $x \in \mathbb{C}^n$, i.e., f is a *square* system of polynomials.

1) *Classical Bézout Bound:* The simplest upper bound on the number of isolated complex stationary points in $V(f)$ is called 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 $\tilde{\mathcal{L}}^\Lambda$, we can see that the leading terms in each polynomial are formed by the product of $2H + 1$ matrices, therefore each polynomial is of degree $2H + 1$. The CBB is therefore the product of these degrees:

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

The CBB is sharp when the system of equations is completely dense, i.e., each polynomial contains all the possible monomials with degrees equal or less than the degree of the polynomial. The systems arising from real-world applications, same as Eqs. (4), are however sparse. Sparse systems have in general significantly smaller number of complex isolated solutions than the CBB. Hence, we need a root-count which takes the sparsity structure of the polynomial systems such as the Bernshtein-Kushnirenko-Khovanskii bound.

2) *BKK Bound:* Another refinement of the Bézout bound takes into consideration the geometry of the “monomial structure” of the polynomial system f .

This number is known as the *Bernshtein-Kushnirenko-Khovanskii Bound*, or simply *BKK Bound* [102], [103], and 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 is given for number of solutions in $\mathbb{C}^n - \{\mathbf{0}\}$ but can also be extended to a bound on number of isolated solutions in \mathbb{C}^n [104], [105].

The BKK bound provides a much tighter refinement on this upper bound. As shown in Table I, the BKK bound the gradient system of $\tilde{\mathcal{L}}^\Lambda$ is much lower than the Bézout number.

B. 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 [106], or for upper bounds on the number of isolated real solutions of systems of polynomial equations [107], [108], [109], [110], [111], [112], [113].

To gain further insight on the number of real stationary points of Eq.(1) (with entries of X and Y picked randomly from a random 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:

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

with n being the number of variables and $(2H + 2)$ is the highest degree of the monomials. $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ is a multi-integer with $|\alpha| = \alpha_1 + \dots + \alpha_n$. Here, a_α are random coefficients i.i.d. drawn from the Gaussian distribution with mean 0 and variance 1. In [106], 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 $i = 1, \dots, n$, is:

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

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.

C. A Few Words on the Equations for the Zero Training Error

In this subsection, we briefly consider 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 [114]. In [114], deep nonlinear networks with rectified linear units (ReLU) were considered and the ReLU were approximated with polynomials of certain degree. Then, the classical Bézout theorem was applied to conclude that for the case when there are more weights than the number of data points, there always are infinitely many global minima expected.

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

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

for all $i = 1, \dots, m$. 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, such minima 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, Eqs. (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 (Eqs. (7)) has no isolated solutions. For the case when $m = n$, the CBB is $(H + 1)^n$ complex isolated solutions.

We emphasize 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. However, except for the special case of $m = n$, these bounds do not provide any detailed information about number of flat stationary points.*

D. Symmetrical Solutions

In this subsection, we show the existence of some symmetries in the solutions of the gradient equations of the deep linear networks.

Proposition 4. *When $H = 1$, if W_0^* and W_1^* form a solution to system (4), then the vector formed by simultaneously reversing the signs of the i -th row of W_0^* and i -th column of W_1^* is also a solution for $i = 1, \dots, d_1$.*

Proof: Let

$$W_0^* = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{d_1} \end{pmatrix}, \text{ and } W_1^* = (c_1, c_2, \dots, c_{d_1}),$$

where r_i represents the i -th row of W_0^* and c_i represents the i -th column of W_1^* . Then system (4) can be rewritten as

$$\begin{pmatrix} c_1^T \\ \vdots \\ c_{d_1}^T \end{pmatrix} \left(\left(\sum_{i=1}^{d_1} c_i r_i \right) X X^T - Y X^T \right) + \Lambda \circ W_0^* = 0 \quad (8)$$

$$\left(\sum_{i=1}^{d_1} c_i r_i \right) X X^T - Y X^T \begin{pmatrix} r_1 \\ \vdots \\ r_{d_1} \end{pmatrix} + \Lambda \circ W_1^* = 0 \quad (9)$$

For (8), use the property that the rows of product of two matrix are linear combinations of rows of the right side matrix, we

have the following

$$c_i^T \left(\left(\sum_{i=1}^{d_1} c_i r_i \right) X X^T - Y X^T \right) + \Lambda(i, :) \circ r_i = 0, \quad (10)$$

where $\Lambda(i, :)$ represents the i -th row of Λ . It is immediately clear that when the signs of c_i and r_i changes simultaneously, (10) remains true. Thus, (8) holds. Using similar technique on the transpose of (9), we can show that (9) also holds. Hence, we conclude the proof. ■

It follows from Prop 5 that, for $H = 1$ and $d_1 = n$ if (4) has a solutions such that all entries of W_0^* and W_1^* are nonzero, then it has at least 2^n solutions. This proof can be generalized to $H = n$ to show that given a solution W_i^* , when reversing the sign of the i -th row of W_j^* and i -th column of W_{j+1} for $i < d_j$ and $j < H + 1$, it still forms a solution.

V. NUMERICALLY FINDING ALL THE STATIONARY SOLUTIONS OF THE DEEP LINEAR NETWORKS

Though solving systems of non-linear equations is a prohibitively difficult task, identifying the system (4) as a system of polynomial equations, several sophisticated computational algebraic geometry techniques can be employed to find all isolated complex solutions of the system. The purely real solutions can then be trivially sorted out from the complex solutions. Symbolic methods such as the Gröbner basis [56], [55] and real algebraic geometry [115] techniques can be used to solve these systems, though they may severely suffer from algorithmic complexity issues. Homotopy continuation methods have been applied to find minima and stationary points of artificial neural networks in the literature [116], [117], [118], [119], [120], [121]. However, though these traditional homotopy continuation methods perform well in finding multiple solutions (and often guarantee to 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) [122], [123] 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.

A. The NPHC Method

For a system of polynomial equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ where $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))$ and $\mathbf{x} = (x_1, \dots, x_n)$, with d_i s being the degree of $f_i(\mathbf{x})$ for all $i = 1, \dots, n$, first, one estimates an upper bound, such as the ones described in Sec. IV, on the number of isolated complex solutions. Then, another system $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ with $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_n(\mathbf{x}))$ is constructed such that (1) the number of complex solutions of the new system is exactly the same as the upper bound of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, and (2) the new system is easy to solve. For the CBB, a straightforward choice for the new system $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ can be $\mathbf{g}(\mathbf{x}) = (x_1^{d_1} - 1, \dots, x_n^{d_n} - 1)$. For tighter upper bounds, constructing the new system may turn out to be more involved and the reader is referred to [124], [125] for further details.

Then, a parametrized system, $\mathbf{h}(\mathbf{x}; t) = 0$, is formed such that $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ and $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ are specific parameter points of $\mathbf{h}(\mathbf{x}; t)$:

$$\mathbf{h}(\mathbf{x}; t) = (1 - t)\mathbf{f}(\mathbf{x}) + \gamma t\mathbf{g}(\mathbf{x}) = \mathbf{0}. \quad (11)$$

Here, $t \in \mathbf{R}$ such that we have $\mathbf{h}(\mathbf{x}; 1) = \mathbf{g}(\mathbf{x}) = \mathbf{0}$ and $\mathbf{h}(\mathbf{x}; 0) = \mathbf{f}(\mathbf{x}) = \mathbf{0}$. $\gamma \in \mathbf{C}$ is a generic complex number. The system $\mathbf{h}(\mathbf{x}; t) = 0$ is also called a homotopy, more specifically, a polynomial homotopy.

Then, each complex isolated solution of $\mathbf{h}(\mathbf{x}; 1) = \mathbf{g}(\mathbf{x}) = \mathbf{0}$, all of which are known by construction, is evolved from $t = 1$ to $t = 0$ using a numerical predictor-corrector method. As long as γ is a generic complex number, all the complex isolated solutions of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ can be reached starting from $\mathbf{g}(\mathbf{x}) = \mathbf{0}$. Specifically, it is proven [126] that each of such solution paths only exhibits either of the two characteristics: (1) the path converges to $t = 0$ and hence a solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ is found, or (2) the path diverges to infinity, i.e., the 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 achieve 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 Example-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.

B. Computational Details

In the following, we show the numerical results for the effect of changing Λ , d_x , d_y , m and H on number of isolated real solutions. 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. We i.i.d. draw each λ_i from $\Lambda \in [0, \Lambda_{\max}]^{d_x d_1 + \sum_{i=1}^{H-1} d_i d_{i+1} + d_H d_y}$, i.e., from the uniform distribution between 0 and Λ_{\max} . For every case, all isolated solutions to each of 1000 samples are computed using the software *Bertini* [125] which is an efficient implementation of the NPHC method. We explore how the change of any of the five variables affect the solutions of (1) for only modest size systems using *Bertini* as we are restricted both in terms of computational resources as well as the number of starting solutions blowing up exponentially as a function of the system size.

C. Results

In this subsection, we provide results of solving Eqs. (4) for the cases described in section V-B.

1) *Enumeration of Complex and Real Solutions:* First, to compare the numerical results with the upper bounds on the number of solutions discussed in section IV, we list the bounds for various cases in Table I. In Table I, we record the number of weights n , CBB, BKK, mean number of complex solutions $\mathcal{N}_{\mathbf{C}}$, the Didieu-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 $d_1 = \dots = d_H = 2$. The CBB grows exponentially with the number of variables. Though the BKK count grows rapidly as well, it is significantly smaller than CBB. However, the average 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 Didieu-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).

In Table II, we record numerical results for mean number of complex solutions, $\mathcal{N}_{\mathbf{C}}$, maximum number of real solutions out of all samples $\max(\mathcal{N}_{\mathbf{R}})$ and the maximum index, $\max(I)$ out of all real solutions of all samples. Note that $m = 1$ is a pathological case as it refers to only one data point case, but it still provides nonlinearity to the gradient equations to have non-trivial solutions. Moreover, the value of m does not change the order of the polynomials but only the monomial structure of the polynomials. When $m = 1$, the matrix XX^T in (4) is singular and of rank 1, which implies there are additional structure. For $m > 1$ and $d_x \leq m$, XX^T is nonsingular with probability 1, the polynomial system has the very same structure yielding a constant number of complex solutions for generic values of X and Y .

2) *Distribution of Number of Real Solutions:* To see the impact of the regulation term Λ , we change the maximum value of the interval on which Λ is uniformly distributed. In Figure 1, we show how the distribution of $\mathcal{N}_{\mathbf{R}}$ changes as the range of Λ changes: the mean number of real solutions decreases as the range of Λ increases, which yields the phenomenon of topology trivialization [127], [128], [129], [130], [131]. As Λ values increase beyond 1, there are more samples with no real solution. Also, as Λ s approach to zero, the mean number of real solutions becomes relatively stable, though the condition number of the real solutions begin to increase which is expected because system (4) tends to be the unregularized case.

Figure 2 demonstrates the impact on average number of real solutions of d_x , d_y and m . It yields that increase of any of the three parameters increases the mean number of real solution. Combining Figure 2 and Table I, one notices that the more data points there are, the more real solutions to (4), on an average, there are.

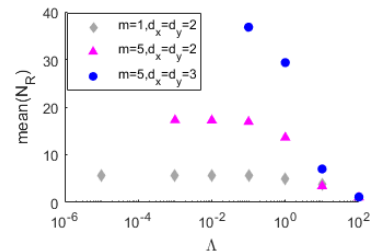


Fig. 1: The mean of $\mathcal{N}_{\mathbf{R}}$ as a function of range of Λ values, for $H = 1$ and $d_1 = 2$.

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

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

TABLE II: 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 I, \mathcal{N}_{DM} is independent of the choice of Λ . For the other values $\mathcal{N}_{\mathbb{C}}$, $\text{mean}(\mathcal{N}_{\mathbb{R}})$, and $\max(I)$, we run 1000 samples for each case with $\Lambda \in [0, 1]$ and $d_i = 2$ for all integers i .

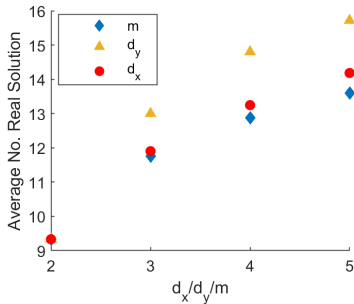


Fig. 2: The mean of $\mathcal{N}_{\mathbb{R}}$ for different value of d_x (hexagon), d_y (triangles) and m (diamonds). For the cases represented by diamonds, $H = 1$, and $d_x = d_y = d_1 = 2$. For the cases represented by triangles, $H = 1$, $m = 2$ and $d_x = d_1 = 2$. For the cases represented by circles, $H = 1$, $m = 2$ and $d_x = d_1 = 2$. The range of Λ is $[0, 1]$ for all cases.

3) *Index-resolved Number of Real Solutions*: For each case and each sample, we compute the ratio of number of real solution with index I , \mathcal{N}_I , to the total number of real solutions.

Then we calculate the index distribution by taking the mean of the ratios for 1000 samples for each case. For the samples for which there are no solutions, we set the ratio to be 0.

Figure 3 and 4 demonstrate the index distribution for different d_x , range of Λ , and m respectively. From table II and right figure of Figure 3, we notice that, for the cases $H = 1$, $m = 2$, and $d_y = d_1 = 2$, even though the number of variable increases from 8 to 14 as d_x increasing from 2 to 5, the number of isolated solution remains the same. It also shows that when $H = 1$, $m = 2$, $d_y = d_1 = 2$, and $\Lambda \in [0, 1]$, the highest index is 4 and the probability of a solutions to (4) is not an extrema of \mathcal{L}^Λ increases as d_x increases.

The left figure of Figure 3 reveals that as the range of regulation term Λ 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 solution with index 2. Similar as in the cases with different d_x , the probability of a solution to (4) is a saddle point of \mathcal{L}^Λ increases sub-linearly as the number of data points increases.

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

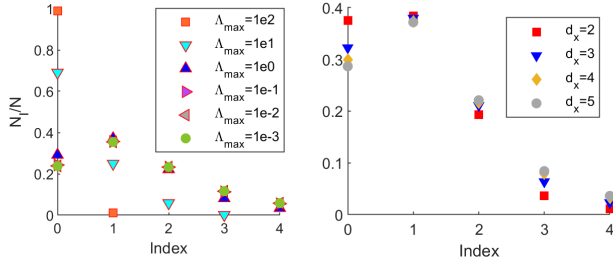


Fig. 3: The index distribution for different d_x and range of Λ . For the figure on the left side, the parameters are set to be $H = 1$, $m = 5$, $d_x = d_y = d_1 = 2$, and $\Lambda \in [0, \Lambda_{\max}]$. For the figure on the right side, the other parameters are set to be $H = 1$, $m = 2$, $d_y = d_1 = 2$, and $\Lambda \in [0, 1]$.

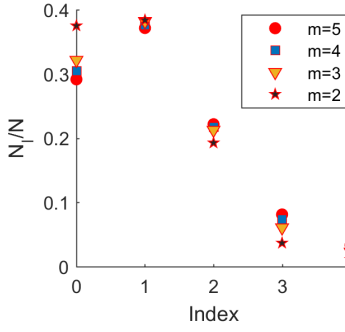


Fig. 4: The index distribution for different m . The other parameters are set to be $H = 1$, $d_x = 2$, $d_y = 2$ and $\Lambda \in [0, 1]$.

being zero is always a solution of Eqs. (4). For the cases where $H = 1$ and $m = 1$, we have total of 13,000 samples consisting of 13 different scenarios listed in Table I, II 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 total of 15000 samples consisting of 15 different scenarios as listed in Table I, II and Figure 1. Here, we observe instances where there exist local minima which are not global minima. One such instance is given in Table III and the parameters for the system are given in equation (12).

For the cases where $H = 1$ and $m = 5$, we notice that all sample runs with $\Lambda \in [0, 100]$ exhibit all local minima are global minima, though, combining with the observation from Figure 1, this may only be an artifact of the topology trivialization.

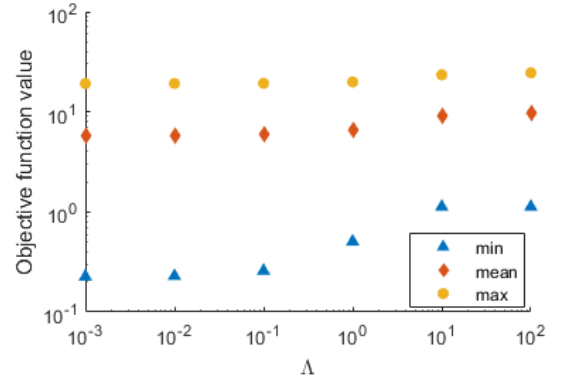


Fig. 5: The minimum, mean and maximum of global minimum loss function value at real solutions of 1000 samples with different range of Λ . The other parameters are set as $H = 1$, $M = 5$, $d_1 = d_x = d_y = 2$.

$$\begin{aligned}
 X &= \begin{bmatrix} -0.1297 & 0.5236 & -2.1491 & 0.3252 & 0.7313 \\ -1.0135 & -1.4616 & -1.6352 & -0.4289 & -0.8680 \\ 0.2523 & 1.8664 & 1.2240 & 0.0116 & 0.9282 \end{bmatrix} \\
 Y &= \begin{bmatrix} 0.6973 & -0.6288 & 1.0285 & -0.9793 & 1.0402 \\ -0.0452 & -0.8566 & -0.2397 & -1.1334 & 1.2315 \\ 0.1912 & -0.3887 & -0.4516 & 0.0221 & 0.5602 \end{bmatrix} \\
 \Lambda_0 &= \begin{bmatrix} 0.383 & 0.6917 & 0.9245 \\ 0.298 & 0.8805 & 0.0813 \end{bmatrix}, \Lambda_1 = \begin{bmatrix} 0.4827 & 0.884 \\ 0.1283 & 0.1963 \\ 0.2529 & 0.1214 \end{bmatrix} \quad (12)
 \end{aligned}$$

5) *Loss Function at Real Solutions:* To see how the range of Λ affect the loss function value, we compute the global minimum of each sample and plot the minimum, mean and maximum of 1000 samples of global minimum for different Λ range. In Figure 5, the log log plot of the minimum, mean, and maximum of 1000 samples for different Λ are shown. We observe that as the range of Λ approaches zero, the mean and minimum of the global minimum approach to nonzero constants. This implies that, for a generic case with 5 data points and two one hidden layer, there is no parameter values of W_0 and W_1 such that the loss function achieves global minimum of 0, i.e., the zero training error minima.

VI. CONCLUSIONS AND DISCUSSION

Understanding non-convexities of the optimization problems in deep learning and their implications in learning are an active area of research. Deep linear networks have served as an ideal test ground of ideas as they qualitatively captures certain features of the deep non-linear 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 the algebraic geometry point of view. Our approach is to provide practicable results from algebraic geometry rather than abstract ones, by invoking computational and numerical algebraic geometry methods.

w_{11}^0	w_{21}^0	w_{12}^0	w_{22}^0	w_{13}^0	w_{23}^0	w_{11}^1	w_{21}^1	w_{31}^1	w_{12}^1	w_{22}^1	w_{32}^1	\mathcal{L}^Λ
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 III: An instance where some local minimum are not global minimum. All real solutions with hessian index 0 are listed. The case settings are $H = 1, m = 5, d_x = d_y = 3, d_1 = 2$, and $\Lambda \in [0, 1]$.

Algebraic Geometry Interpretation:- In the present paper, after reviewing the existing results on the 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 emphasize that the algebraic geometric interpretation of gradient equations is not restricted only to the deep linear networks: The 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 (ReLU), leaky ReLUs, Heaviside, etc. activation functions are, or can be transformed to, form 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 and their Regularization:- We then reviewed the current understanding of the "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 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 strictly constant when evaluated over 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, though the hessian matrices evaluated at both of which are singular.

For deep linear networks, in the present paper, 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 possess no local minima which are not

global minima. Our results then yield that the loss surface of unregularized deep linear network consists of *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.

Then, using the 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 (SGD) descent method and its variants rely only on the first order (gradients) information while searching for a minimum, they have to pass through saddle points of higher index. The number of saddle points of higher index is usually exponentially more than the number of minima in such a high dimensional and nonlinear loss landscapes. In addition, if there are flat saddle points present in the system, the SGD may encounter further issues such as the computation getting stuck on the flat saddle point, in turn performance plateauing 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) [25], [36], [37], [38], [12], [39]. An alternative way to evade the 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 and hence the wide minima escaping methods can 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 the loss landscapes quantitatively (and in some cases even qualitatively) changes 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 the 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 [40]. The generalized L_2 regularization term essentially perturbs the system to leave only isolated solutions in the system. e.g., in [132], 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 relation between the generalized L_2 regularization and skipped connections. One can also project the constant zero modes of the hessian in the computation [40]. 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.

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 complex space, there are many upper bounds, such as the CBB and BKK bounds, on the number of isolated complex solutions for systems of polynomial equations available in the literature which we can employ to gain insight into our systems. For the deep linear networks, the CBB and BKK for modest size networks are given in Table I.

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 drawn from the Gaussian distribution with mean 0 and variance 1 and λ_i s from uniform distributions between $[0, 1]$, $[0, 0.01]$, \dots , $[0, 0.00001]$ to investigate the effect of different magnitudes of λ_i s. The average number of complex and real solutions over all samples are compared with the CBB and BKK in Table I. We compared these bounds with the available analytical result on the average number of real stationary points of random polynomial cost function. The average number of complex and real solutions of these systems are orders of magnitudes smaller than these bounds confirming that the deep linear systems are very *sparse*. This conclusion *may or may not* necessarily extend to deep nonlinear network as the structure of the corresponding polynomials may differ.

We showed that the average number of real solutions reduces as λ_i s vanishes, the phenomenon called topology trivialization [127], [128], [129], [130], [131], and singular solutions appear as the system tends to the unregularized case where the solutions are flat.

We sorted the stationary points in terms of index (number of negative eigenvalues) of 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. [54] is the only result available for the linear networks with non-zero

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 not necessarily be a phenomenon for the 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 the working of the deep learning to fit 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 saddles escaping methods will be an important breakthrough on these theoretical insights.

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, “Generalization 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] D. J. Wales, *Energy Landscapes*. Cambridge: Cambridge University Press, 2003.
- [41] B. D. Haeffele and R. Vidal, “Global optimality in tensor factorization, deep learning, and beyond,” *arXiv preprint arXiv:1506.07540*, 2015.
- [42] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, “Understanding deep neural networks with rectified linear units,” *arXiv preprint arXiv:1611.01491*, 2016.
- [43] C. Yun, S. Sra, and A. Jadbabaie, “Global optimality conditions for deep neural networks,” *arXiv preprint arXiv:1707.02444*, 2017.
- [44] S. Feizi, H. Javadi, J. Zhang, and D. Tse, “Porcupine neural networks:(almost) all local optima are global,” *arXiv preprint arXiv:1710.02196*, 2017.
- [45] Z. Zhang, Y. Wu, and G. Wang, “Bpgrad: Towards global optimality in deep learning via branch and pruning,” *arXiv preprint arXiv:1711.06959*, 2017.
- [46] 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.
- [47] H. Lu and K. Kawaguchi, “Depth creates no bad local minima,” *arXiv preprint arXiv:1702.08580*, 2017.
- [48] 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.
- [49] P. Baldi and Z. Lu, “Complex-valued autoencoders,” *Neural Networks*, vol. 33, pp. 136–147, 2012.
- [50] M. Hardt and T. Ma, “Identity matters in deep learning,” *arXiv preprint arXiv:1611.04231*, 2016.
- [51] Y. Zhou and Y. Liang, “Critical points of neural networks: Analytical forms and landscape properties,” *arXiv preprint arXiv:1710.11205*, 2017.
- [52] Y. Zhou and Y. Liang, “Characterization of gradient dominance and regularity conditions for neural networks,” *arXiv preprint arXiv:1710.06910*, 2017.
- [53] 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.
- [54] 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.
- [55] 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.
- [56] D. A. Cox, J. Little, and D. O’Shea, *Using Algebraic Geometry*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998.
- [57] S. Watanabe, *Algebraic geometry and statistical learning theory*, vol. 25. Cambridge University Press, 2009.
- [58] 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.
- [59] 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.
- [60] 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.
- [61] 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.
- [62] 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.
- [63] C. Knoll and F. Pernkopf, “On loopy belief propagation—local stability analysis for non-vanishing fields,” in *Uncertainty in Artificial Intelligence*, 2017.
- [64] 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.
- [65] 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.
- [66] T. Kim and T. Adalı, “Approximation by fully complex multilayer perceptrons,” *Neural computation*, vol. 15, no. 7, pp. 1641–1666, 2003.
- [67] T. Nitta, “An extension of the back-propagation algorithm to complex numbers,” *Neural Networks*, vol. 10, no. 8, pp. 1391–1415, 1997.
- [68] H. Akira, *Complex-valued neural networks: theories and applications*, vol. 5. World Scientific, 2003.
- [69] D. P. Reichert and T. Serre, “Neuronal synchrony in complex-valued deep networks,” *arXiv preprint arXiv:1312.6115*, 2013.
- [70] N. Guberman, “On complex valued convolutional neural networks,” *arXiv preprint arXiv:1602.09046*, 2016.
- [71] I. Danihelka, G. Wayne, B. Uribe, N. Kalchbrenner, and A. Graves, “Associative long short-term memory,” *arXiv preprint arXiv:1602.03032*, 2016.
- [72] 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.
- [73] 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.
- [74] 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.
- [75] 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.
- [76] M. Arjovsky, A. Shah, and Y. Bengio, “Unitary evolution recurrent neural networks,” in *International Conference on Machine Learning*, pp. 1120–1128, 2016.
- [77] S. Hochreiter and J. Schmidhuber, “Flat minima,” *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997.
- [78] S. Hochreiter and J. Schmidhuber, “Simplifying neural nets by discovering flat minima,” in *Advances in neural information processing systems*, pp. 529–536, 1995.
- [79] 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.
- [80] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey, “Finding flatter minima with sgd,” 2018.
- [81] S. Jastrzebski, 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.
- [82] 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.
- [83] 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.
- [84] 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.
- [85] 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.
- [86] 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.
- [87] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, “Sharp minima can generalize for deep nets,” *arXiv preprint arXiv:1703.04933*, 2017.
- [88] R. Brockett, “Some geometric questions in the theory of linear systems,” *IEEE Transactions on Automatic Control*, vol. 21, no. 4, pp. 449–455, 1976.
- [89] 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.
- [90] D. Saad and S. A. Solla, “On-line learning in soft committee machines,” *Physical Review E*, vol. 52, no. 4, p. 4225, 1995.
- [91] V. Kůrková and P. C. Kainen, “Functionally equivalent feedforward neural networks,” *Neural Computation*, vol. 6, no. 3, pp. 543–558, 1994.
- [92] 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.
- [93] S. Watanabe, “Almost all learning machines are singular,” in *Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symposium on*, pp. 383–388, IEEE, 2007.
- [94] 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.
- [95] 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.
- [96] L. Sagun, L. Bottou, and Y. LeCun, “Eigenvalues of the hessian in deep learning: Singularity and beyond,” *arXiv preprint arXiv:1611.07476*, 2016.
- [97] L. Sagun, U. Evci, V. U. Güneş, Y. Dauphin, and L. Bottou, “Empirical analysis of the hessian of over-parametrized neural networks,” *arXiv preprint arXiv:1706.04454*, 2017.
- [98] 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.
- [99] R. H. Abraham and J. W. Robbin, *Transversal Mappings and Flows*. Benjamin, 1967.
- [100] E. L. Allgower and K. Georg, *Introduction to Numerical Continuation Methods*. John Wiley & Sons, New York, 1979.
- [101] 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.
- [102] 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.
- [103] B. Huber and B. Sturmfels, “A polyhedral method for solving sparse polynomial systems,” *Mathematics of Computation*, vol. 64, no. 212, pp. 1541–1555, 1995.
- [104] T. Y. Li and X. Wang, “The BKK root count in C^n ,” *Mathematics of Computation*, vol. 65, no. 216, pp. 1477–1485, 1996.

- [105] 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.
- [106] 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.
- [107] 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.
- [108] 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.
- [109] 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.
- [110] L. Blum, "F. c u c k e r, m. s h u b, and s. s m a l e. complexity and real computation;" 1998.
- [111] 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.
- [112] 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.
- [113] 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.
- [114] Q. Liao and T. Poggio, "Theory ii: Landscape of the empirical risk in deep learning," *arXiv preprint arXiv:1703.09833*, 2017.
- [115] S. Basu, R. Pollack, and M. F. Roy, *Algorithms in Real Algebraic Geometry*. Springer, 2003.
- [116] 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.
- [117] 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.
- [118] 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.
- [119] 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.
- [120] 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.
- [121] A. Anandkumar, Y. Deng, R. Ge, and H. Mobahi, "Homotopy analysis for tensor pca," in *Conference on Learning Theory*, pp. 79–104, 2017.
- [122] A. Sommese and C. Wampler, *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific Publishing, Hackensack, NJ, 2005.
- [123] T. Y. Li, "Solving polynomial systems by the homotopy continuation method," *Handbook of numerical analysis*, vol. XI, pp. 209–304, 2003.
- [124] A. J. Sommese and C. W. Wampler, *The numerical solution of systems of polynomials arising in Engineering and Science*. World Scientific Publishing Company, 2005.
- [125] D. Bates, J. Hauenstein, A. Sommese, and C. Wampler, *Numerically solving polynomial systems with Bertini*, vol. 25. SIAM, 2013.
- [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. Niernerg, N. J. Simm, and D. A. Stariolo, "Energy Landscape of the Finite-Size Mean-field 2-Spin Spherical Model and Topology Trivialization," 2014.
- [131] P. Chaudhari and S. Soatto, "Trivializing the energy landscape of deep networks," *arXiv preprint arXiv:1511.06485*, 2015.
- [132] A. E. Orhan and X. Pitkow, "Skip connections eliminate singularities," *arXiv preprint arXiv:1701.09175*, 2017.