The Landscape of Multi-Layer Linear Neural Network From the Perspective of Algebraic Geometry

Yang Xiuyi
fwinlee@gmail.com

Abstract
The clear understanding of the non-convex landscape of neural network is a complex incomplete problem. This paper studies the landscape of linear (residual) network, the simplified version of the nonlinear network. By treating the gradient equations as polynomial equations, we use algebraic geometry tools to solve it over the complex number field, the attained solution can be decomposed into different irreducible complex geometry objects. Then three hypotheses are proposed, involving how to calculate the loss on each irreducible geometry object, the losses of critical points have a certain range and the relationship between the dimension of each irreducible geometry object and strict saddle condition. Finally, numerical algebraic geometry is applied to verify the rationality of these three hypotheses which further clarify the landscape of linear network and the role of residual connection.

1 Introduction
The commonly used deep neural network with non-convex loss surface brings significant improvement to many practical applications[12]. The difficulty of non-convex optimization was manifest in the practical development of early neural networks [4]. In the past few years, with the introduction of some new structures, such as residual connection [8], normalization techniques [9, 1] and so on, non-convex neural networks optimized by stochastic gradient descent(SGD) and its variants can often get very low loss value in practice. The different non-convex landscapes caused by different components are the key to understand why SGD and its variants works.

It has been demonstrated that the linear network is similar to the nonlinear network in many aspects. The work of [17] analyzed the learning dynamics of linear network and non-linear network have similar patterns. Components of nonlinear networks such as residual connection and normalization method can also be used to solve the gradient vanishing or explosion problem of linear networks. These similarities are the first reason why we regard linear network as an ideal model for the research of non-linear neural network. Secondly, general results of nonlinear neural network are often difficult to derive analytically, whereas the linear network can often be analyzed in mathematical detail [2]. So the main topic in present paper is exploring the loss surface of linear (residual) network.

These results that every local minimum is a global minimum and every critical point that is not a global minimum is a saddle point are attained under different assumptions [11] [15] [13, 2]. The works of [21], [22] present conditions for a critical point of the risk function to be a global minimum. These results demonstrated that saddle points of linear network are the biggest barrier for SGD. We will use algebraic geometry to zoom in the critical points of linear network to explain the following experimental phenomena.

Suppose input and output data are fitted perfectly by a linear network, the parameters are initialized with random Gaussian with zero means, SGD is used to minimize squared error risk. When the depth of neural network is not so large, the loss of final solution of optimization is close to zero. When the gradient disappears due to the increase of depth, the residual connection can make the loss of the final solution of linear network close to zero. Three questions arise here. One is what structure near point 0 causes the gradient to disappear, another is how the residual connection resolves the dying gradient issue, and the final one is what structure the saddle points have to explain the saddle points is not an obstacle to SGD?

2 Linear (residue) network
In this section, we describe notations for multi-layer linear neural networks and multi-layer linear residue networks and the problem formulation.

Let’s say that there are m training data, \( x_i, y_i \) is the ith input data and the corresponding output. \( X \in \mathbb{R}^{d_x \times m} \) be the data matrix and \( Y \in \mathbb{R}^{d_y \times m} \) be the target matrix, \( d_x, d_y \) are the number of input and output units respectively. Since the multi-layer linear network (number of layers greater than 1) is a forward neural network with identity mapping activation function, if do not consider the residue connection, normalization methods, etc., and use square error loss, the objective function to be optimized is

\[
L(W) = \frac{1}{2} \sum_{i=1}^{m} \left\| \prod_{k=1}^{H+1} W_k x_i - y_i \right\|_F^2
\]

Where \( \prod_{k=1}^{H+1} W_k = W_{H+1} W_H \cdots W_1, n > 1, \) for \( k = 1, \ldots, H + 1, \) \( W_k \in \mathbb{R}^{d_k \times d_{k-1}} \) is the weights between adjacent layers. For notational simplicity we let \( d_0 = d_x \) and \( d_{H+1} = d_y \) and the width of net is defined as \( k = \min(d_0, d_1, \ldots, d_{H+1}) \). The number of weights, or variables, \( n = d_0 \times d_1 + d_1 \times d_2 + \cdots + d_H \times d_{H+1} \).
For linear residual networks, we only consider the case that the units of each layer are equal and residue connection only skip one layer as in\cite{7}. The empirical risk in this case is as follows,

$$L_{e}(W') = \frac{1}{2} \sum_{i=1}^{m} \prod_{k=1}^{H+1} (I + W_{k}^{i})x_{i} - y_{i}^{i}$$ \quad (2)

It is easy to conclude that the loss surface of linear residual network is the translation of the corresponding linear network because that the objective function equation (2) is obtained by reparameterization $W_{k} = I + W_{k}'$ from equation (1). Therefore, this conclusion that there are only global minima and saddle points holds for linear residual networks.

Let the partial derivative of $L(W)$ with respect to $W_{k}$ equals to 0,

$$\frac{\partial L(W)}{\partial W_{i}} = (\prod_{k=i+1}^{H+1} W_{k})^{T}(\prod_{k=1}^{H+1} W_{k}X - Y)X^{T}(\prod_{k=i+1}^{H+1} W_{k})^{T} = 0$$ \quad (3)

for $i = 1, \ldots, H + 1$.

Thus, this is a set of polynomial equations in the entries of $W_{1}, \ldots, W_{H+1}$, so is linear residual networks.\cite{16} remove all the flat stationary points by adding an extension of $L_{2}$-regularization to the loss function (1), then pick out real solutions from all isolated complex points and find that there are indeed local minima which are not global minima contrary to the available conclusions in the unregularized case \cite{11}. As proved in \cite{20}, regularization alters the loss surface of linear critical points.

Not only the above gradient equations (4) can be transformed into polynomial equations, but also some others results about linear networks can be transformed into polynomial equations. A recent major work on linear networks\cite{21}, gives the condition that the critical points are globally optimal when the width of linear network equals to the number of neuron of input or output layer. This condition can be considered from the perspective of polynomial equations.

**Theorem 1** (\cite{22}) If $k = \min\{d_{x}, d_{y}\}$, define the following set $V_{1} := \{(W_{1}, \ldots, W_{H+1}) : \text{rank}(W_{H+1} \cdots W_{1}) = k\}$. Then, every critical point of $L(W)$ in $V_{1}$ is a global minimum. Moreover, every critical point of $L(W)$ in $V_{1}$ is a saddle point.

Here the solution set of rank$(W_{H+1} \cdots W_{1}) = k$ is complementary set of solution set of rank$(W_{H+1} \cdots W_{1}) < k$. The latter is equivalent to the determinants of all the $k \times k$ sub-matrix of $W_{H+1} \cdots W_{1}$ equal to zero. So saddle points are solutions of polynomial equations. When solutions of gradient equations and these polynomial equations lives in $\mathbb{C}^{n}$, they are called complex saddle points. The corresponding global minima are called complex global minima. These saddle points are further divided into different subsets each of which satisfies rank$(W_{n} \cdots W_{1}) = i, i = 0, \ldots, k - 1$. Likewise, these different subsets can be computed through rank$(W_{n} \cdots W_{1}) < i, i = 1, \ldots, k$. There must have a trivial solution that is saddle point 0 when rank$(W_{n} \cdots W_{1}) = 0$ and the losses at these saddle points of rank zero equal to the loss at saddle point 0.

Since global minima are critical points which meet the condition rank$(W_{H+1} \cdots W_{1}) = k$, we have following hypothesis,

**Hypothesis 1** When the width of nets equals to the number of input or output units, the larger the rank of the product of the chained product of weight matrices, the smaller the loss of the corresponding critical points.

The critical points under condition rank$(W_{n} \cdots W_{1}) = i$ may include many stationary points or none. So the comparison of the loss of critical points which satisfy rank$(W_{n} \cdots W_{1}) = i$ and rank$(W_{n} \cdots W_{1}) = i - 1$ is based on the rule: the loss of any point in one set is greater than or equal to or less than or equal to that of any point in the other set. In the case of one of them is empty set, the comparison is trivially true.

### 3 Methodology

A brief introduction of algebraic geometry is provided in subsection 3.1. Then in next subsection 3.2, the reason why complex field is used to solve equations of critical points is explained. In subsection 3.3, a very simple network that is known all the stationary points is used as an example to explain the algebraic geometry terms in present paper. This example leads to two other hypotheses related to the properties of critical points. Finally in subsection 3.4, three methods in numerical algebraic geometry are introduced, which will be used to compute complex critical points.

#### 3.1 A brief introduction to algebraic geometry

The fundamental goal of algebraic geometry is to study solution sets of systems of polynomial equations in several variables. For a subset $S \subset \mathbb{C}[x_{1}, \ldots, x_{n}]$ of polynomials we define $V(S) := \{x \in \mathbb{C}^{n} : f(x) = 0 \text{ for all } f \in S\}$ closed affine varieties of $\mathbb{C}^{n}$.

In this paper, the closed affine varieties to be studied are, the complex solutions of gradient equations of linear (residue) network (complex critical points) and its subsets. In view of the importance of the critical point in the study of the loss surface of linear network, we name complex critical points as linear neural variety. The word closed appears in the definition because it can be shown that the closed affine varieties of $\mathbb{C}^{n}$ satisfy the axioms to be the closed subsets for a topology on $\mathbb{C}^{n}$. This topology is called the Zariski topology on $\mathbb{C}^{n}$. The induced topology on a closed affine variety $V$ of $\mathbb{C}^{n}$ is called the Zariski topology on $V$. The Zariski topology is coarser than the usual complex topology. Irregular (non-smooth or local dimension greater than the dimension of variety) points are measure 0 in variety $V$ is deduced from that smooth points on this variety $V$ are zariski dense in it. A generic property is a property which is true for almost every point of a variety and a generic point of a variety is a point at which all generic properties are true. For example, a generic point of a variety is a smooth point.
One of main goals in present paper is decomposing a variety into finite union of irreducible varieties. Let \( V \) be a topological space. We say that \( V \) is reducible if it can be written as \( V = V_1 \cup V_2 \) for closed subsets \( V_1, V_2 \subseteq V \). Otherwise \( V \) is called irreducible. The finite irreducible decomposition of an affine variety is \( V = V_1 \cup V_2 \cdots \cup V_r \) of irreducible closed subsets, up to permutation. \( V_i, i = 1, \ldots, r \) are called the irreducible components of \( V \). These irreducible closed subsets can be written as the zero locus of finitely many polynomials by Hilbert’s Basis Theorem. If the components of linear neural variety belong to complex global minima they are called global minimum components, if belongs to complex saddle points, then are called saddle components. Each algebraic component has a well-defined dimension and degree. Every irreducible dimension 0 algebraic component is a set of single point. A irreducible algebraic curve has dimension 1 and so on. The degree of an affine variety of dimension \( n \) is the number of intersection points of the variety with \( n \) generic hyperplanes. The number of intersection is counted with intersection multiplicity. Complex irreducible component \( V_i \) has a important property that if \( V_i \) is an irreducible algebraic variety over \( \mathbb{C} \), then \( V_i(\mathbb{C}) \) is connected.

3.2 Why linear neural variety lives in \( \mathbb{C}^n \)

From the definition of linear neural variety, it can be seen that real stationary points are the intersection of linear neural variety and \( \mathbb{R}^n \), that is real solutions of gradient system. Symbolic computational methods such as the Gröbner basis [5, 6] and cylindrical algebraic decomposition [10] in semi-algebraic geometry can be used to solve this system by identifying it as polynomial equations. But Gröbner basis and cylindrical algebraic decomposition may have a worst case complexity doubly exponential in the number of solutions of the polynomial system and the size of the input, respectively. Due to algorithmic complexity issues of these two methods, [16] employ the numerical homotopy continuation method [18, 3] to sort these purely real solutions out from the complex solutions through removing flat stationary points by a generalized \( L_2 \) penalty term. However, the loss surface is changed after the introducing of this penalty term. Therefore, in present paper, we consider the linear neural variety over the complex field and studying the properties of critical points by studying the corresponding linear neural variety.

In particular, there is a theorem in algebraic geometry relates complex solutions of polynomial equations with real solutions. That is,

**Theorem 2** [19] Let \( V_i \subset \mathbb{C}^n \) be an irreducible variety defined by real polynomials. If \( V \) has a smooth real point, then \( V_i(\mathbb{R}) = V_i \cap \mathbb{R}^n \) is Zariski dense in \( V \).

Training data in present paper are over the real number, so if linear neural variety can be decomposed into finite irreducible components, each of which are defined by finite real polynomials and if each irreducible component has a smooth real point, then all algebraic and geometric information about this component is already captured in real critical points, and vice-versa. So, if the complex dimension of each irreducible component are known, then corresponding real counterpart is with the same real dimension. At the same time, each irreducible component is regarded as an individual just like the isolated critical point, its pseudo loss and pseudo eigenvalues of Hessian matrix should be studied. The word pseudo appears here due to that each irreducible closed variety is solved over complex number, yet loss and eigenvalues of Hessian are only meaningful for real point of a component at first glance. It will be seen that loss and eigenvalues of real critical points \( V_i(\mathbb{R}) \) can be reflected by corresponding irreducible component \( V_i \). After all, both complex critical points and real critical points correspond to the same polynomial equations, only one are solved over complex field, the other are solved over real number.

3.3 Example: a very simple linear neural network

This section will first examine the linear variety of simple network with width 1, then the loss and eigenvalues of Hessian matrix of each irreducible component are derived analytically, this leads to two other main hypotheses in present paper.

the gradient polynomial system with respect to the loss function is:

\[
\frac{\partial l(w)}{\partial w} = -(y - (\prod_{k=1}^{H+1} w_k)x)((\prod_{k=1}^{i-1} w_k)x \times \prod_{k=i+1}^{H+1} w_k) = 0 \tag{4}
\]

The quadratic loss is globally minimized by solutions of the equation \( (\prod_{k=1}^{H+1} w_k)x - y = 0 \). On the contrary, if \( (\prod_{k=1}^{H+1} w_k)x - y \neq 0 \), then the locus of

\[
\prod_{k=1}^{i-1} w_k x \times \prod_{k=i+1}^{H+1} w_k = 0, i = 1, \ldots, n; \quad \tag{5}
\]

be composed of

\[
w_j = 0, w_j = 0, i, j \in \{1, \ldots, H + 1\}, i \neq j. \tag{6}
\]

With terms in algebraic geometry, above is equivalent to the linear variety defined by equation (4) decomposed as the union of global minimum variety and \( (H + 1) \) saddle components defined by equations (5) and (6) respectively. The global minimum component is a complex surface with complex dimension \( H = n - 1 \), each irreducible component is a complex plane with complex dimension \( H - 1 = n - 2 \) and these saddle component intersect at saddle point 0. Note that the real dimension is consistent with the complex dimension, these components can be treated as a real surface and a real plane with real dimension \( H = n - 1 \) and \( H - 1 = n - 2 \) respectively.

Observe that the pseudo loss of each irreducible component is constant at every complex point, this pseudo loss is the same as the actual loss at real critical point. So, there are only two loss values, one is global minimum 0 at global minimum component, the other is saddle loss \( \frac{1}{2} \sum_{j=1}^{m} (y_j)^2 \) at each saddle component. The loss surface of 2 units neural network is seen as in figure [1]
neural variety. The loss function of multi-layer linear network is constant on each irreducible algebraic component of its linear neural variety.

Hypothesis 2

The only non constant zero eigenvalues of Hessian at a point on global minimum component is \( \lambda^r, f(W, \lambda) = \lambda^r g(W, \lambda), r \) is the dimension of this component. The coefficients of \( f \) and \( g \) are (rational) polynomials of parameters of network. If some point of this component makes the constant term of \( g \) vanish, then the number of constant zero eigenvalues is \( r + 1 \). These points is rare due to they are the closed subvariety of this component. When several components meet at a point, the number of constant zero eigenvalues may increase due to different component correspond to different eigenvalues of Hessian matrix as above network example. So, we have the following corollary,

Corollary 1

For every point of each irreducible component, the number of constant zero eigenvalues of Hessian at this point equals to the dimension of this component.

Corollary 2

For a generic point of each irreducible component, the number of constant zero eigenvalues of Hessian at a critical points as long as we sample a complex or real point so that we can get the number of constant zero eigenvalues, this will be seen in section 4, so the losses of these several components will be equal to the loss of point 0.

Hypothesis 3

For a generic point of each irreducible component, the number of constant zero eigenvalues of Hessian at this point equals to the dimension of this component.

This hypothesis has a very obvious corollary,

Corollary 1

The loss function of multi-layer linear network is constant on finite many connected algebraic components of its linear neural variety.

This hypothesis has a very obvious corollary,

Corollary 1

The loss function of multi-layer linear network is constant on finite many connected algebraic components of its linear neural variety.

This hypothesis has a very obvious corollary,

Corollary 1

The loss function of multi-layer linear network is constant on finite many connected algebraic components of its linear neural variety.

This hypothesis has a very obvious corollary,

Corollary 1

The loss function of multi-layer linear network is constant on finite many connected algebraic components of its linear neural variety.

This hypothesis has a very obvious corollary,

Corollary 1

The loss function of multi-layer linear network is constant on finite many connected algebraic components of its linear neural variety.

This hypothesis has a very obvious corollary,

Corollary 1

The loss function of multi-layer linear network is constant on finite many connected algebraic components of its linear neural variety.
contains three part \((S, L, S \cap L)\). The first part \(S\) is a system of polynomial equations which define the studied variety, the second part \(L\) is a generic slicing plane the dimension of which is codimension of \(V\), the last part \(S \cap L\) is witness points (the intersection points of slicing plane \(L\) and variety \(V\)) the number of which equals to the degree of variety \(V\).

Wit set are been used to compute anything of interest in numerical algebraic geometry such as the dimension and degree of a variety, the intersection of two varieties, etc. In the present paper, we are interested in three methods using witness sets: numerical irreducible decomposition, component membership testing, and component sampling. **Numerical irreducible decomposition** is computing a witness set for each of the irreducible components, **component membership testing** method tests whether a given point is on a variety and **component sampling** method is used to sample a generic point on a variety. These three methods are implemented in Bertini [3].

With these three methods of numerical algebraic geometry, we can compute the loss and the number of constant zero eigenvalues of a irreducible component,

1. First represent linear neural variety as the union of irreducible components by numerical irreducible decomposition method.
2. Then sample a generic point on a component to compute the loss and eigenvalues at this point, this will lead to results we want.

### 4 Experiments

In this section, we will compute linear neural varieties of some networks, then compute loss on each irreducible component. In order to verify the rational of the hypothesis and its corollary, we sample at least two points to compute their losses to get the final loss of a component. We use component membership testing to test point 0 is actually the intersection point of several components in some cases of nets and the losses of these intersecting components indeed have the same loss. Then the rational of the hypothesis is obvious by solving the saddle points under different ranks. The checking of reasonableness of hypothesis and its corollary is also obvious after we computed each irreducible component. Finally, the validity of the loss surface of linear residue network is the translation of corresponding linear network by checking whether two linear neural varieties are equal after translate one linear neural variety, component membership testing method will be used during checking the relationship between two sets.

Experimental settings are as follows. Due to we have limited computational resource, we only compute width and depth of nets less than Four. Algebraic geometry method is suitable for linear network trained with any data, so we don’t make any assumptions about the amount of data or the distribution of data. But in order to check whether the loss value of global minimum component is zero, the input and output data can be perfectly fitted by linear network. For convenience and the loss surface is non-convex regardless of the number of training data, the training data is up to three.

#### Linear neural varieties and losses

The results of numerical irreducible decomposition of linear neural varieties and losses are summarized in Table 1, 2, 3, 4, 5. We observe that in all of these results, there is only one global minimum component and the losses of components which saddle point 0 is on are larger than that of any other component.

#### Critical points arrangement

When the width of linear network is one and \(k = \min\{d_x, d_y\}\), there are only global minimum component and saddle components passing through point 0. If we calculate rank zero saddle points, the result is saddle components passing through point 0. For example, for net with structure "2-2-2, 2 training data", after we calculate gradient formula accompany by saddle condition rank \((W_3W_2W_1) < 1\), the results is in Table 10. Component membership testing is used to determine whether these saddle components equal to saddle components obtained before.

When the width of linear network is greater than one and \(k = \min\{d_x, d_y\}\), we take net with structure "2-2-2, 2 training data” as example. The saddle points under rank condition rank \((W_2W_1) < 2\) are in Table 11 and the saddle points under rank condition rank \((W_2W_1) < 1\) are just components passing through 0. It can be seen that these results are consistent with Hypothesis [1].

The number of constant zero eigenvalues on a component

After we have all the linear neural varieties of these nets, we can calculate the eigenvalues of a sampled generic point. We list three typical results in Table 2, 3, 4. In all three cases, the number of constant zero eigenvalues of a generic point is identical with the dimension of component. Table 2 shows that saddle point 0 is totally flat (all eigenvalues are zero). Table 3 is a example that saddle point 0 is strict saddle but has more zero eigenvalues than any other point of component and Table 4 demonstrates that point 0 is a generic point if it is not a intersection point.

#### The case of linear residual network

Due to the landscape translation relation of linear residual network and linear network, the solution of the linear neural variety of linear residual network has components with the same dimensions and degrees as corresponding counterpart. This is shown in Table 15.

#### 5 Conclusion and discussion

In this paper, we use numerical algebraic geometry to get some complex solutions to gradient formulas for multi-layer linear networks although the actual critical points are over the real number field. Three hypotheses are putting forward. The reasonableness of these three hypotheses is verified by numerical algebraic geometry computation. Hypothesis [3] means that each...
### Table 1. The linear neural variety of 1-1-1-1-1 linear net, with 2 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|-------------------------------|
| Dim 3, deg 4, 1 component | 0    | No                            |
| Dim 2, deg 1, 6 components | 10   | Yes                           |

### Table 2. The linear neural variety of 2-2-2-1 linear net, with 1 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|-------------------------------|
| Dim 9, deg 3, 1 component | 0    | No                            |
| Dim 6, deg 1, 1 component | 12.5 | Yes                           |
| Dim 6, deg 3, 2 components | 12.5 | Yes                           |

### Table 3. The linear neural variety of 2-2-2-1 linear net, with 2 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|-------------------------------|
| Dim 8, deg 9, 1 component | 0    | No                            |
| Dim 6, deg 1, 1 component | 13   | Yes                           |
| Dim 6, deg 3, 1 component | 13   | Yes                           |
| Dim 6, deg 3, 1 component | 13   | Yes                           |

### Table 4. The linear neural variety of 2-2-2-1 linear net, with 3 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|-------------------------------|
| Dim 8, deg 9, 1 component | 0    | No                            |
| Dim 6, deg 1, 1 component | 13.5 | Yes                           |
| Dim 6, deg 3, 1 component | 13.5 | Yes                           |
| Dim 6, deg 3, 1 component | 13.5 | Yes                           |

### Table 5. The linear neural variety of 2-2-2 linear net, with 2 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|-------------------------------|
| Dim 4, deg 8, 1 component | 0    | No                            |
| Dim 3, deg 2, 1 component | 5.3  | No                            |
| Dim 3, deg 2, 1 component | 1.69 | No                            |
| Dim 0, deg 1, 1 component | 7    | Yes                           |

### Table 6. The linear neural variety of 2-2-3 linear net, with 2 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|-------------------------------|
| Dim 4, deg 8, 1 component | 0    | No                            |
| Dim 4, deg 4, 1 component | 2    | No                            |
| Dim 4, deg 4, 1 component | 7    | No                            |
| Dim 2, deg 1, 1 component | 9    | Yes                           |

### Table 7. The linear neural variety of 2-1-2-2 linear net, with 2 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|-------------------------------|
| Dim 5, deg 1, 1 component | 13   | Yes                           |
| Dim 5, deg 2, 1 component | 13   | Yes                           |
| Dim 5, deg 3, 1 component | 13   | Yes                           |
| Dim 5, deg 6, 1 component | 0    | No                            |
| Dim 4, deg 1, 1 component | 13   | Yes                           |

### Table 8. The linear neural variety of 3-2-3 linear net, with 2 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|-------------------------------|
| Dim 6, deg 4, 1 component | 8.17 | No                            |
| Dim 6, deg 4, 1 component | 97.82 | No                           |
| Dim 6, deg 8, 1 component | 0    | No                            |
| Dim 4, deg 1, 1 component | 106  | Yes                           |

### Table 9. The linear neural variety of 2-1-2 linear net, with 2 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|-------------------------------|
| Dim 1, deg 1, 2 components | 13   | Yes                           |
| Dim 1, deg 2, 1 component | 0    | No                            |
**Table 10.** The saddle components of 2-2-2-1 linear net, with 2 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|------------------------------|
| Dim 6, deg 1, 1 component | 13   | Yes                          |
| Dim 6, deg 3, 1 component | 13   | Yes                          |

**Table 11.** The saddle components of 2-2-2 linear net, with 2 training data under condition rank($W_2 W_1$) < 2

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|------------------------------|
| Dim3, deg2,1 comp      | 5.3  | No                           |
| dim3, deg2,1 comp      | 1.69 | No                           |
| dim0, deg1,1 comp      | 7    | Yes                          |

**Table 12.** The eigenvalues of a generic point on global minimum component, one of saddle components and saddle point 0

| Dim 3, deg 4, 1 component | Dim 2, deg 1, 6 components | Saddle point 0 |
|---------------------------|-----------------------------|----------------|
| -1.9149e+01 - 4.4317e+01i | -5.0525e+01 - 7.6442e+01i | 0              |
| 6.3600e-07 + 2.4911e-06i  | 1.0105e+01 + 1.5288e+01i   | 0              |
| -2.7642e-07 + 1.6268e-07i | 3.8473e-33 - 3.2474e-33i | 0              |
| -5.0118e-07 - 2.7757e-08i | -7.6581e-34 - 6.0445e-34i | 0              |

**Table 13.** The eigenvalues of a generic point on global minimum component, one of saddle components and saddle point 0

| Dim 1, deg 1, 2 components | Dim 1, deg 2, 1 component | Saddle point 0 |
|----------------------------|---------------------------|----------------|
| 9.0735e+00 + 1.6091e-01i   | -1.8925e+01 + 2.4843e+00i | -9.0554        |
| -9.0345e+00 + 1.6022e-01i  | -3.5563e-01 + 1.0326e-01i | -0.0000        |
| -2.3606e+16 + 4.7020e-17i  | -1.0784e-06 - 2.4537e-07i | 0              |
| 3.8988e-02 + 3.2113e-01i   | -5.6025e+00 - 1.7950e+00i | 9.0554         |

**Table 14.** The eigenvalues of a generic point on global minimum component, one of saddle components and saddle point 0

| Dim 4, deg 8, 1 component | Dim 3, deg 2, 2 components | Saddle point 0 |
|----------------------------|-----------------------------|----------------|
| -2.6274e+00 - 1.6162e+01i  | -9.5240e+00 + 1.4045e+00i  | -4.6056        |
| -7.5948e+00 - 8.0849e+00i  | -8.8515e+00 - 7.4800e+00i  | -4.6056        |
| 5.1586e+00 - 5.7933e+00i   | -1.7240e-01 - 8.4029e+00i  | -2.6056        |
| 1.9124e-01 + 2.2839e+00i   | 1.3592e+00 - 9.8640e-01i   | -2.6056        |
| 1.6486e-06 + 1.8576e-06i   | -5.1422e-01 + 5.0474e-01i  | 2.6056         |
| -4.1784e-07 + 6.3060e-07i  | -4.7988e-07 + 4.8486e-07i  | 2.6056         |
| -2.2429e-07 - 2.2350e-07i  | 4.2581e-08 - 5.4055e-07i   | 4.6056         |
| -8.0787e-07 - 1.5855e-06i  | 4.7988e-07 - 4.8486e-07i   | 4.6056         |

**Table 15.** The linear neural variety of 2-2-2 linear residual net, with 2 training data

| Irreducible components | Loss | Is saddle 0 on this component? |
|------------------------|------|------------------------------|
| Dim 4, deg 8, 1 component | 0     | No                           |
| Dim 3, deg 2, 1 component | 5.3  | No                           |
| Dim 3, deg 2, 1 component | 1.69 | No                           |
| Dim 0, deg 1, 1 component | 7    | Yes                          |
irreducible saddle component is regarded as a high-dimensional generalization of an isolated saddle point. In addition, every point of a dense subset of each component has at least one strictly negative eigenvalue, that is to say, almost every saddle point is not a problem with SGD by work of [14]. Its Corollary 2 means that the loss surface of most linear networks will be flat near the saddle point 0, and the residual connection will make the path of SGD avoid passing through this area. Hypothesis 2 makes it possible to define the loss of each individual component. Hypothesis 1 implies that the losses of all irreducible components are bounded by the losses of saddle point 0 and global minimum component. According these three hypotheses we can think of loss surface as a concave surface, and the existence of saddle point 0 leads to the bottom bulge as Figure 1. All other components are in the bottom bulge area, which represents that their losses have upper and lower bound.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

[2] Pierre F Baldi and Kurt Hornik. Learning in linear neural networks: A survey. IEEE Transactions on neural networks, 6(4):837–858, 1995.

[3] Daniel J Bates, Andrew J Sommese, Jonathan D Hauenstein, and Charles W Wampler. Numerically solving polynomial systems with Bertini. SIAM, 2013.

[4] Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. Neural Networks, 5(4):117–127, 1992.

[5] David Cox, John Little, and Donal OShea. Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra. Springer Science & Business Media, 2013.

[6] David A Cox, John Little, and Donal O’shea. Using algebraic geometry, volume 185. Springer Science & Business Media, 2006.

[7] Moritz Hardt and Tengyu Ma. Identity matters in deep learning. arXiv preprint arXiv:1611.04231, 2016.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[10] Mats Jirstrand. Cylindrical algebraic decomposition—an introduction. Linköping University, 1995.

[11] Kenji Kawaguchi. Deep learning without poor local minima. Advances in neural information processing systems, 29:586–594, 2016.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25:1097–1105, 2012.

[13] Thomas Laurent and James Brecht. Deep linear networks with arbitrary loss: All local minima are global. In International conference on machine learning, pages 2902–2907. PMLR, 2018.

[14] Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent converges to minimizers. arXiv preprint arXiv:1602.04915, 2016.

[15] Haihao Lu and Kenji Kawaguchi. Depth creates no bad local minima. arXiv preprint arXiv:1702.08580, 2017.

[16] Dhagash Mehta, Tianran Chen, Tingting Tang, and Jonathan D Hauenstein. The loss surface of deep linear networks viewed through the algebraic geometry lens. arXiv preprint arXiv:1810.07716, 2018.

[17] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv preprint arXiv:1312.6120, 2013.

[18] Andrew J Sommese, Charles W Wampler, et al. The Numerical solution of systems of polynomials arising in engineering and science. World Scientific, 2005.

[19] Frank Sottile. Real algebraic geometry for geometric constraints. Handbook of geometric constraint systems principles, Discrete Math. Appl.(Boca Raton), pages 273–285, 2016.

[20] Amirhossein Taghvaei, Jin W Kim, and Prashant Mehta. How regularization affects the critical points in linear networks. In Advances in neural information processing systems, pages 2502–2512, 2017.

[21] Chuilhee Yun, Suvrit Sra, and Ali Jadbabaie. Global optimality conditions for deep neural networks. arXiv preprint arXiv:1707.02444, 2017.

[22] Yi Zhou and Yingbin Liang. Critical points of neural networks: Analytical forms and landscape properties. arXiv preprint arXiv:1710.11205, 2017.