Convex Formulation of Overparameterized Deep Neural Networks

Cong Fang, Yihong Gu, Weizhong Zhang, and Tong Zhang, *Fellow, IEEE*

***Abstract*— The analysis of over-parameterized neural networks has drawn significant attention in recent years. It was shown that such systems behave like convex systems under various restricted settings, such as for two-layer neural networks, and when learning is only restricted locally in the so-called neural tangent kernel space around specialized initializations. However, there is a lack of powerful theoretical techniques that can analyze fully trained deep neural networks under general conditions. This paper considers this fundamental problem by investigating such overparameterized deep neural networks when fully trained. Specifically, we characterize a deep neural network by its features' distributions and propose a metric to intuitively measure the usefulness of feature representations. Under certain regularizers that bounds the metric, we show deep neural networks can be reformulated as a *convex* optimization and the system can guarantee effective feature representations in terms of the metric. Our new analysis is more consistent with empirical observations that deep neural networks are capable of learning efficient feature representations. Empirical studies confirm that predictions of our theory are consistent with results observed in practice.**

***Index Terms*— Deep learning, convex reformulation, feature representation.**

I. Introduction

DEEP Neural Networks (DNNs) have achieved great successes in numerous real applications, such as image classification [1]–[3], face recognition [4], video understanding [5], neural language processing [6], etc. Compared to the empirical successes, the theoretical understanding of DNNs falls far behind. Part of the reasons might be a general perception that DNNs are highly non-convex learning models. However, it is observed in practice that with the help of a number of tricks, DNNs can be reliably trained from random initialization with reproducible results. The solutions from proper training procedures behave well and consistently. In other words, two different random initializations (using the same initialization and training strategy) generally lead to models that give similar predictions on test data. Therefore, we may conclude that proper training leads to similar solutions for neural networks. This behavior resembles that of convex optimization, instead of generic non-convex optimization that tends to get stuck in suboptimal local stationary solutions. This empirical observation appears to be rather mysterious and requires further understanding.

In recent years, there have been significant breakthroughs [7]–[10] in analyzing over-parameterized Neural Networks (NNs), which are NNs with many neurons in the hidden layer(s). It is observed from empirical studies that such NNs are easy to train [11]. And it was noted that under some restrictive settings, such as two-layer NNs [7], [8] and when learning is only restricted locally in the neural tangent kernel space around certain initializations [9], [10], NNs behave like convex systems when the number of the hidden neurons goes to infinity. Unfortunately, existing studies failed to analyze fully trained DNNs under general settings. Moreover, the analysis cannot explain how DNNs can learn discriminative features specifically for the underlying learning task, as observed in real applications and argued to be one of the contributors for the success of deep learning [12], [13].

To remedy the gap between the existing theories and practical observations, this paper develops a new theoretical tool that can be used to study fully trained DNNs. Our results show that under suitable conditions, in the limit of an infinite number of hidden neurons, DNNs are infinite-dimensional *convex* learning models with appropriate re-parameterization. In our framework, given a DNN, the hidden layers are regarded as features and the model output is given by a simple linear model using features of the top hidden layer. The output of a DNN, in the limit of an infinite number of hidden neurons, depends on the distributions that can represent features, the weights that connect the layers, and the final linear model. We show the closeness of continuous DNN with its discretization and further study the variance of such approximation. The variance motivates us to propose a new metric that can intuitively measure the usefulness of feature representations. Then the feature learning process of a deep neural network is characterized by a class of regularizers that upper bound our proposed metric under mild conditions. We show under suitable re-parameterization, the objective function can be largely simplified. Especially, when such regularizers are imposed,

Manuscript received 30 June 2021; revised 3 January 2022; accepted 3 March 2022. Date of publication 30 March 2022; date of current version 13 July 2022. The work of Cong Fang was supported by the WuDao Research Foundation. The work of Tong Zhang was supported by the General Research Fund (GRF) under Grant 16201320. An earlier version of this paper was presented at the Neurips 2020. *(Corresponding author: Tong Zhang.)*

Cong Fang was with the Shenzhen Research Institute of Big Data, Shenzhen 518100, China. He is now with the Department of Machine Intelligence, Peking University, Beijing 100871, China (e-mail: fangcong@pku.edu.cn).

Yihong Gu was with the Shenzhen Research Institute of Big Data, Shenzhen 518100, China. He is now with the Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: yihongg@princeton.edu).

Weizhong Zhang is with the Department of Mathematics, The Hong Kong University of Science and Technology (HKUST), Hong Kong (e-mail: zhangweizhongzju@gmail.com).

Tong Zhang is with the Department of Mathematics and Computer Science, The Hong Kong University of Science and Technology (HKUST), Hong Kong, and also with Google Research, New York, NY 10011 USA (e-mail: tongzhang0@gmail.com).

Communicated by R. Venkataramanan, Associate Editor for Machine Learning and Statistics, Communications, Signal Processing and Source Coding.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TIT.2022.3163341.

Digital Object Identifier 10.1109/TIT.2022.3163341

the overall program under a special re-parameterization can be convex. Unlike the Neural Tangent Kernel approach, our theoretical framework for DNN does not require the variables to be confined in an infinitesimal region.

More concretely, the paper is organized as follows. Section II discusses the relationship of this paper to earlier studies, especially recent works on the analysis of over-parameterized NNs. Section III introduces the basic definition of discrete DNNs. Section IV describes the continuous DNN when the number of hidden nodes goes to infinity in the discrete DNN. In this formulation, each hidden layer is represented by a distribution over an abstract space that can represent features, i.e. function values on the hidden units of the input. We further interpret a discrete DNN as a random sample of hidden nodes from a continuous DNN at each layer, and then study the variance of such random discretization. The variance formula motivates the study of a class of regularization conditions for DNNs that measure the usefulness of the feature representations. Section V studies the DNN training problem, in which we specialize the abstract space by $\mathbb{R}^N$ which are feature spaces for the training samples. We show that the overall program can be re-parameterized into a convex optimization problem. Section VI discusses the implications of the convex reformulation. We explain why (noisy) gradient descent finds a global minimum. In particular, we show the critical points of the continuous DNN in a particular space attains the global minimum. Section VII presents experiments to demonstrate that our theory is consistent with empirical observations.

The main contributions of this work can be described as follows.

- We propose a new framework for analyzing over-parameterized DNNs. We show that DNNs can be re-parameterized as a convex optimization problem under certain conditions.
- We propose a metric that can be intuitively used to measure the usefulness of feature representations. We then propose a class of regularizers, and we show that under certain conditions, such regularizers are approximately an upper bound of such a metric.
- We conduct empirical studies to validate the consistency of our theory with practice.

A. Notations

For a vector $x \in \mathbb{R}^d$, we denote $\|\cdot\|_1$ and $\|\cdot\|$ to be its ℓ_1 and ℓ_2 norms, respectively. We let $x^\top$ be the transpose of x and x_k be the value of k-th dimension of x with $k = 1, \ldots, d$. Let $[m] = \{1, 2, \ldots, m\}$ for a positive integer m. For a function $f(x) : \mathbb{R}^d \to \mathbb{R}$, we denote $\nabla_x f$ to be the gradient of f with respect to x. For two real valued numbers a and b, we denote $a \vee b$ to be $\max(a, b)$. If μ and ν are two measures on the same measurable space, we denote $\mu \ll \nu$ if μ is absolutely continuous with respect to ν, and $\mu \sim \nu$ if $\mu \ll \nu$ and $\nu \ll \mu$.

II. Related Work

There have been a number of significant developments to obtain better theoretical understandings of over-parameterized NNs, which are NNs that contain a large number of hidden nodes. The motivation of overparameterization comes from the empirical observation that over-parameterized DNNs are much easier to train and often achieve better performances [11].

In some earlier works, a number of researchers [14]–[19] studied the landscape of NNs under special conditions either for input data or for NN architectures. By carefully characterizing the geometric landscapes of the objective function, these early works showed that some special NNs satisfy the so-called strict saddle property [20]. One can then use some recent results in nonconvex optimization [21]–[23] to show that first-order algorithms for such NNs can efficiently escape saddle points and converge to some local mimima.

In the generic setting, one popular approach to study two-layer NNs is the mean-field analysis. This point of view models the continuous NN as a distribution over the NN's parameters, and it studies the evolution of the distribution as a Wasserstein gradient flow during the training process [7], [8], [24]–[27]. The process can be represented by a partial differential equation, which can be further studied mathematically. For two-layer continuous NNs, it is known that the objective function with respect to the distribution of parameters is convex in the continuous limit. And it was shown that (noisy) Gradient Decent can find globally optimal solutions under certain conditions. However, the mean-field analysis relies on the special observation that a two-layer continuous NN is naturally a linear model with respect to the distribution of NN parameters, and this observation is not applicable to multi-layer architectures. Consequently, it is difficult to generalize the view to analyze DNNs. In fact, in the recent attempts [28], the technique of mean-field analysis is applied to DNNs. Though they proved the 0 loss phenomenon for the Gradient Descent algorithm, their analysis is under restricted conditions and no regularizer can be incorporated during training.

One remarkable direction to extend the mean-field view to analyze DNNs is to restrict the DNN parameters in an infinitesimal region around the initial values. With proper scaling and random initialization, DNN can be regarded as a linear model in this infinitesimal region, which makes the training dynamics solvable. This point of view is referred to as the Neural Tangent Kernel (NTK) view [9], [10], [29]–[37], since the evolution of the trainable parameters, when restricted to the infinitesimal region, can be characterized by a kernel in the tangent space. However, NTK essentially approximates a nonlinear NN by a linear model of the infinite-dimensional random features associated with the NTK, and so NTK view cannot explain how NNs learn the feature.

Another line of research [38]–[42] study DNNs by studying the duality of the DNNs. They proposed a special convex duality and show under suitable conditions the strong duality holds. In this way, NNs can be learned by semi-defined programming in polynomial computational complexities under suitable conditions. Compared with our work, these works mainly study the standard ℓ_2 regularizers. They novelly learn the NNs in the dual space. In contrast, we study $\ell_{1,p}$ norm ($p \geq 1$) and show the critical points of the continuous DNN under our reformulation attains the global minimum.

The most related work is [43], which was a short conference paper focusing on preliminary and mainly empirical studies of loss landscape for convolutional neural networks, evolved from an earlier version of the current paper. The major differences between the present paper and [43] are summarized as follows. First, the starting points of the two papers are different. [43] only aims to simplify the landscape of NNs. In contrast, the present paper provides a thorough study for DNNs by understanding DNNs from the view of feature representation: We treat a discrete DNN as a collection of hidden nodes samples from a continuous DNN with weights connecting them, thus the study of the variance for such random discretization leads to a new type of regularizers. Therefore, under our framework, not only the optimization of DNNs can be re-parameterized as a convex program, but also we can explain that DNNs are capable of learning effective features in terms of representation for the learning task. Second, the experimental goal in [43] is to validate the fact that the solution paths with different random initialization are essentially the same during the training. In contrast, the experiments in this paper focus on the consistency of our theory with practice and involve detailed analysis including the verification of optimal condition, the role of variance of approximation, etc. Overall the current paper is a deeper and more comprehensive investigation of neural network convexification and its consequences.

The work of [44] proposed after our work rigorously studies the Gradient Descent dynamic equation. Based on our continuous framework, they show that with the number of hidden units for the discrete NN going to infinity and the step size going to 0, the Gradient Descent algorithm converges to a continuous dynamic called neural feature flow. Moreover, under certain conditions, neural feature flow converges to 0 training loss. In their analysis, no regularizer is considered.

III. Preliminary: Discrete Deep Neural Networks

In this section, we introduce the discrete fully connected DNNs. We consider a DNN $\hat{f}$ with L hidden layers. For input $x \in \mathbb{R}^d$, we can define $\hat{f}$ recursively as a function of x.

Specifically, we define $\hat{f}_j^{(0)}(x) = x_j$ for $j \in [d]$. Let $m^{(0)} = d$. Then for $\ell \in [L]$, we define the nodes in the ℓ-th layer as

$$\hat{f}_j^{(\ell)}(x) = h^{(\ell)}\left(\hat{g}_j^{(\ell)}(x)\right), \tag{1}$$

with

$$\hat{g}_j^{(\ell)}(x) = \frac{1}{m^{(\ell-1)}} \sum_{k=1}^{m^{(\ell-1)}} w_{j,k}^{(\ell)} \hat{f}_k^{(\ell-1)}(x), \quad j \in [m^{(\ell)}],$$

where $h^{(\ell)}$ is the activation function and $w^{(\ell)} \in \mathbb{R}^{m^{(\ell)} \times m^{(\ell-1)}}$ is the weight matrix of the ℓ-th layer comprised of $m^{(\ell)}$ rows, i.e., $w_j^{(\ell)} = [w_{j,1}^{(\ell)}, \ldots, w_{j,m^{(\ell-1)}}^{(\ell)}] \in \mathbb{R}^{m^{(\ell-1)}}$ with $j \in [m^{(\ell)}]$. At the top layer, we define the output $\hat{f}(x)$ as

$$\hat{f}(x) = \frac{1}{m^{(L)}} \sum_{j=1}^{m^{(L)}} u_j \hat{f}_j^{(L)}(x), \tag{2}$$

where $u_j \in R^K$ is a vector. This defines an $(L+1)$-layer fully connected deep neural network with $m^{(\ell)}$ nodes in each layer.

IV. Continuous Deep Neural Networks

In this section, we introduce the continuous DNN formulation according to the definition of discrete DNN in Section III and then show the relationship between the continuous and discrete DNNs.

A. Continuous DNN Formulation

In the case of continuous DNN, for each $\ell \in [L]$, we first consider an abstract space $\mathcal{Z}^{(\ell)}$ equipped with a probability measure $\rho^{(\ell)}$ to represent the states of the hidden units at layer ℓ. Then we introduce function $w^{(\ell)} : \mathcal{Z}^{(\ell-1)} \times \mathcal{Z}^{(\ell)} \to \mathbb{R}$ to represent the weights that connected from layer $\ell - 1$ to ℓ. We let $\rho = \{\rho^{(0)}, \ldots, \rho^{(L)}\}$ and $w = \{w^{(1)}, \ldots, w^{(L)}\}$. The details for the continuous DNN formulation are presented below.

At 0-th layer, we define $\mathcal{Z}^{(0)} := [d]$ as the node space corresponding to the d components of the input x. Let $\rho^{(0)}$ be the uniform distribution on $\mathcal{Z}^{(0)}$. For each node $z^{(0)} \in \mathcal{Z}^{(0)}$, we let

$$f^{(0)}(\rho, w, z^{(0)}; x) = x_{z^{(0)}}.$$

Consider the ℓ-th layer with $\ell \in [L]$. Because $z^{(\ell)}$ can be regarded as a hidden node in layer ℓ, and $z^{(\ell-1)}$ as a hidden node in the $(\ell-1)$-th layer, thus $w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)})$ is the analogy of $w_{i,j}^{(\ell)}$ in discrete DNN, which is the weight connecting node i and j in layer ℓ and $\ell - 1$. Using this notation, we define the function associated with the node $z^{(\ell)}$ in ℓ-th layer of the continuous NN as follows:

$$f^{(\ell)}(\rho, w, z^{(\ell)}; x) = h^{(\ell)}\left(g^{(\ell)}(\rho, w, z^{(\ell)}; x)\right), \tag{3}$$

where $h^{(\ell)}(\cdot)$ is the activation function of the ℓ-th layer, and

$$g^{(\ell)}(\rho, w, z^{(\ell)}; x) = \int w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)}) f^{(\ell-1)}(\rho, z^{(\ell-1)}; x)\, d\rho^{(\ell-1)}(z^{(\ell-1)}).$$

Moreover, we let $\rho^{(\ell)}$ be a probability measure over $\mathcal{Z}^{(\ell)}$.[1]

Finally, for the output layer, let $u(\cdot) : \mathcal{Z}^{(L)} \to \mathbb{R}^K$ be a K-dimensional vector valued function on $\mathcal{Z}^{(L)}$, then we can define the final output of continuous DNN as

$$f(\rho, w, u; x) = \int u(z^{(L)}) f^{(L)}(\rho, w, z^{(L)}; x)\, d\rho^{(L)}(z^{(L)}). \tag{4}$$

We will establish the relationship between the continuous and discrete DNNs.

B. Assumptions

Before presenting the result, we specify the necessary assumptions first. These assumptions are mild.

Assumption 1 (Bounded Gradient Condition): We assume the activation function is differentiable, and its derivative is

[1] $f^{(\ell)}$ and $g^{(\ell)}$ depend only on the components of ρ up until the ℓ-th layer.

bounded. That is, there exists a constant $c_0 > 0$, such that

$$\left|\nabla h^{(\ell)}(x)\ \right| \le c_0, \quad \ell \in [L].$$

Assumption 2 (Continuous Gradient Condition): We further assume that there exist two constants $\alpha > 0$ and $c_1 > 0$ such that

$$\left|\nabla h^{(\ell)}(x)\ -\nabla h^{(\ell)}(y)\right| \le c_1|x-y|^{\alpha}, \quad \ell \in [L].$$

Assumption 2 is a special type of modulus of continuity for $\nabla h^{(\ell)}(x)$. When $\alpha = 1$, Assumption 2 is the standard L-smooth condition for the activation function $h^{(\ell)}(x)$. When $\alpha < 1$, it holds more generally in the local region. When proving Theorem 2, our moment condition in Assumption 3 depends on α. We note that most commonly-used activation functions, e.g. sigmoid, tanh, and smooth relu, admit this assumption for all $0 < \alpha \le 1$.

Assumption 3 ((q_0, q_1)-Bounded Moment Condition): We assume for all $\ell \in [2 : L]$, we have

$$\mathbb{E}_{z^{(\ell-1)},\ldots,z^{(L)}}\Bigg|\|u^{(L)}(z^{(L)})\| \vee 1 \prod_{i=\ell+1}^{L}\left(|w^{(i)}(z^{(i)}, z^{(i-1)})| \vee 1\right)$$
$$\left[w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)}) f^{(\ell-1)}(\rho, w, z^{(\ell-1)}; x) - g^{(\ell)}(\rho, w, z^{(\ell)}; x)\right]\Bigg|^{q_0}$$
$$\le c_M.$$

Moreover, we assume

$$\mathbb{E}_{z^{(L)}}\left\|u^{(L)} f^{(L)}(\rho, z^{(L)}; x) - f(\rho, u; x)\right\|^{q_1} \le c_{M_1}.$$

The constants q_0 and q_1 in Assumption 3 will be specified later based on our theorem statements.

C. Relationship Between Discrete and Continuous DNNs

We now investigate the relationship between the discrete and continuous DNNs. Given a continuous DNN, we can construct a discrete one by sampling hidden nodes from the probability measure sequence ρ. The detailed procedure is as follows:

1) Keep the input layer of the discrete DNN identical to that of the continuous DNN.
2) For each hidden layer $\ell \in [L]$, draw $m^{(\ell)}$ i.i.d. samples $\{z_i^{(\ell)} : i \in [m^{(\ell)}], z_i^{(\ell)} \in \mathcal{Z}^{(\ell)}\}$, which is denoted as $\hat{\mathcal{Z}}^{(\ell)}$, from $\rho^{(\ell)}$ of continuous DNN, and set the weights
$$w_{i,j}^{(\ell)} = w^{(\ell)}(z_i^{(\ell)}, z_j^{(\ell-1)})$$
with $i \in [m^{(\ell)}]$ and $j \in [m^{(\ell-1)}]$.
3) For the top layer, set
$$u_j = u(z_j^{(L)}),$$
for each sampled $z_j^{(L)}$ with $j = [m^{(L)}]$.

The following result shows when $m^{(\ell)} \to \infty$ for all $\ell \in [L]$, the final output converges to that of the continuous DNN in L^1. All the proofs in this paper are left to the appendices.

Theorem 1 (Consistence of Discretization): Given a continuous NN, under Assumptions 1 and 3 with $q_0 = q_1 = 1 + c_\varepsilon$ for any $c_\varepsilon > 0$, suppose there is a discrete NN constructed from the continuous DNN using the procedure above, then for any input x, we have

(i) for any $k \ge 2$ and $j \in [m^{(k)}]$, given $z_j^{(k)}$,
$$\lim_{m^{(\ell)}\to\infty, \ell\in[k-1]} \mathbb{E}\left[\hat{f}_j^{(k)}(x) - f^{(k)}(\rho, w, z_j^{(k)}; x)\right] \mid \sigma(z_j^{(k)}) = 0,$$
(ii)
$$\lim_{m^{(\ell)}\to\infty, \ell\in[L]} \mathbb{E}\left\|\hat{f}(x) - f(\rho, u, w; x)\right\| = 0.$$

The part (i) of the theorem above does not cover the case of $k = 1$, since it is trivial to show that $f^{(1)}(\rho, w, z_j^{(1)}; x) \equiv \hat{f}_j^{(1)}(x)$ holds for all $j \in [m^{(0)}]$.

Conversely, given a discrete DNN, we may ask if there is a continuous DNN that can construct the discrete one by sampling the hidden nodes. Consider the discrete DNN that is initialized by some standard initialization strategy and is trained by (Scaled) Gradient Descent which may be of our central interest. Its hidden units are correlated since they all depend on a common set of outputs from the previous layer and have interactions during training. However, they are actually "nearly independent" when the number of hidden units is large enough in the sense that there is always an approximated discrete DNN that can be constructed from the procedure above for a certain continuous DNN. Similar to Theorem 1, the two DNNs approximate to each other with the increase of hidden units. The result was shown in [44] by a similar argument from the propagation of chaos. Therefore, in most cases, the discrete DNN can be interpreted as a random sample of hidden nodes from a continuous DNN at each layer.

D. Variance of Discrete Approximation

We further estimate the variance of such approximation with a slightly strong condition.

Theorem 2 (Variance of Discrete Approximation): We denote $\frac{\partial f(\rho,w,u;x)}{\partial z^{(L)}} = u(z^{(L)})$ and

$$\frac{\partial f(\rho, w, u; x)}{\partial z^{(\ell)}} = \mathbb{E}_{z^{(\ell+1)}}\left[A^{(\ell+1)}\frac{\partial f(\rho, w, u; x)}{\partial z^{(\ell+1)}}\right]$$

where $A^{(\ell+1)} = w^{(\ell+1)}(z^{(\ell+1)}, z^{(\ell)})\nabla h^{(\ell+1)}\left(g(\rho, w, z^{(\ell+1)}; x)\right)$ and $\ell \in [L-1]$. Then under Assumptions 1, 2, and 3 with $q_0 = 2(1+\alpha)^L$, $q_1 = 2$ and treating c_0, c_1, α, c_M, c_{M_1}, and L as constants, we have

$$\begin{aligned}
&\mathbb{E}\left\|\hat{f}(x) - f(\rho, w, u; x)\right\|^2 \qquad (5)\\
&= \sum_{\ell=1}^{L-1}\frac{1}{m^{(\ell)}}\mathbb{E}_{z^{(\ell)}}\left|\mathbb{E}_{z^{(\ell+1)}}\frac{\partial f(\rho, w, u; x)}{\partial z^{(\ell+1)}}\bar{\Delta}_{\ell+1}\right|^2\\
&+ \frac{1}{m^{(L)}}\mathbb{E}_{z^{(L)}}\left\|\bar{\Delta}_{L+1}\right\|^2 + \mathcal{O}\left(\sum_{\ell=1}^{L}(m^{(\ell)})^{-2}\right)\\
&+ \mathcal{O}\left(\sum_{\ell=1}^{L}(m^{(\ell)})^{-(1+\alpha/2)}\right),
\end{aligned}$$

where

$$\bar{\Delta}_{L+1} = f^{(L)}(\rho, w, z^{(L)}; x)u(z^{(L)}) - f(\rho, w, u; x),$$

and

$$\bar{\Delta}_\ell = f^{(\ell-1)}(\rho, w, z^{(\ell)}; x) w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)}) - g^{(\ell)}(\rho, w, z^{(\ell)}; x),$$

for $\ell \in [2:L]$.

In Theorem 2, for $\ell \in [L-1]$, we can choose $\alpha = \mathcal{O}\left(\frac{1}{L^{1+\nu}}\right)$ with $\nu \geq 0$. Thus, the assumption only requires the bounded $2 + \mathcal{O}\left(\frac{1}{L^\nu}\right)$-th moment.

Theorem 2 shows that the variance of the approximation decreases with a rate of $\sum_{\ell=1}^{L} O(1/m^{(\ell)})$. We can use the terms on the right hand side of (5), i.e. $\left|\mathbb{E}_{z^{(\ell+1)}} \frac{\partial f(\rho,w,u;x)}{\partial z^{(\ell+1)}} \bar{\Delta}_{\ell+1}\right|^2$ ($\ell \in [L-1]$) and $\mathbb{E}_{z^{(L)}} \left\|\bar{\Delta}_{L+1}\right\|^2$ as a metric to measure the usefulness of the feature representations. That is, if the variance of the corresponding discrete approximation is small, then a discrete DNN with a small number of hidden neurons can represent the target f. This inspires us to impose an appropriate regularization condition to guarantee the usefulness of the feature representations. Specifically, if we assume that both $|f^{(\ell)}(\rho, w, z^{(\ell)}; x) \frac{\partial f(\rho,w,u;x)}{\partial z^{(\ell+1)}}|$ and $|f^{(L)}(\rho, z^{(L)}; x)|$ are bounded, then in order to minimize the variance, Theorem 2 implies that we can minimize the following regularization (see the proofs of (7) and (8) in Appendix B.4):

$$R^{(w,u)} = \sum_{\ell=1}^{L} \lambda^{(\ell)} R_w^{(\ell)}(\rho, w) + \lambda^{(u)} R^{(u)}(\rho, u), \tag{6}$$

with

$$R_w^{(\ell)}(\rho, w) \tag{7}$$
$$= \int \left(\int |w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)})| d\rho^{(\ell)}(z^{(\ell)}) \right)^2 d\rho^{(\ell-1)}(z^{(\ell-1)}),$$

$$R^{(u)}(\rho, u) = \int \left\| u(z^{(L)}) \right\|^2 d\rho^{(L)}(z^{(L)}), \tag{8}$$

the parameters $\lambda^{(1)}, \ldots, \lambda^{(L)}$ and $\lambda^{(u)}$ being non-negative. Thus, we propose a new regularizer that controls the efficacy of the learned feature distributions in terms of efficient representation under random sampling. Under the regularizers, the variance decreases with $\sum_{\ell=1}^{L} O(1/m^{(\ell)})$. We remark that our unexpected result in Section V will show that with this regularizer, the learning problem is convex under suitable reparameterization. Moreover, in the discrete formulation, the regularizer is the simple $\ell_{1,2}$ norm regularizer if we write $w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)})$ as a matrix with the (j,i)-th entry being $w^{(\ell)}(z^{(i)}, z^{(j)})$, which is a convex regularizer.

V. Convexify DNN

In this section, we study the training problem of DNNs. We show the learning program can be reparameterized as a convex optimization.

A. Problem Setup

Consider training a DNN, where we are given N training pairs $\{x^i, y^i\}_{i=1}^N$ with $x^i \in \mathbb{R}^d$ and $y \in \mathbb{R}^k$ and our target is to minimize the loss:

$$\hat{Q}(w, u) = \frac{1}{N} \sum_{i=1}^{N} \phi(\hat{f}(x^i), y^i) + \hat{R}(w, u), \tag{9}$$

where $\phi(\cdot, \cdot)$ is the loss function used to measure the quality of prediction and $\hat{R}$ is the regularizer. We assume that loss is convex in the first argument and consider the $\ell_{1,2}$ norm regularizer written as:

$$\hat{R}(w, u) = \left[\sum_{\ell=1}^{L} \lambda^{(\ell)} \hat{R}_w^{(\ell)}(\rho, w) + \lambda^{(u)} \hat{R}^{(u)}(\rho, u) \right], \tag{10}$$

where

$$\hat{R}_w^{(\ell)}(w^{(\ell)}) = \frac{1}{m^{(\ell-1)}} \sum_{k=1}^{m^{(\ell-1)}} \left| \frac{1}{m^{(\ell)}} \sum_{j=1}^{m^{(\ell)}} \left| w_{j,k}^{(\ell)} \right| \right|^2, \tag{11}$$

$$\hat{R}^{(u)}(u) = \frac{1}{m^{(L)}} \sum_{j=1}^{m^{(L)}} \|u_j\|^2, \tag{12}$$

the parameters $\lambda^{(1)}, \ldots, \lambda^{(L)}$ and $\lambda^{(u)}$ are non-negative. The regularizer is not necessary for our convex argument, but is capable of controlling the efficiency of the feature under our proposed metric.

B. Specialize $\mathcal{Z}^{(\ell)}$ as Feature Space

Observe that the feature for the N-training samples can be written as N-dimensional vector. We specify $\mathcal{Z}^{(\ell)}$ as $\mathbb{R}^N$ for $\ell \in [2:L]$. In this way, a continuous DNN is characterized by the distributions and functions of the features. The overall learning program becomes a constrained optimization, where the complicated recursive composition structure of the forward propagation is simplified as independent constraints. We present the details below, following a similar argument in Section IV.

(1) At 0-th layer, we denote $X = \left[x^1, x^2, \ldots, x^N\right]^\top \in \mathbb{R}^{N \times d}$.

(2) At 1-st layer, because each hidden node (before the activation function) is computed by a linear mapping of the input data, each node can be indexed by the weights connecting it to the input. We define $\mathcal{Z}^{(1)}$ as $\mathbb{R}^d$ and let $\rho^{(1)}$ be the probability measure over $\mathbb{R}^d$,[2] then for all $z^{(1)} \in \mathrm{supp}(\rho^{(1)})$, the outputs of the nodes $z^{(1)}$ satisfy:

$$\theta_1(z^{(1)}) := \frac{1}{d} X z^{(1)}. \tag{13}$$

(3) At 2-nd layer, recall that the output of each node, i.e., the feature, for the training samples is a N-dimensional vector. We define $\mathcal{Z}^{(2)}$ as $\mathbb{R}^N$ and let $\rho^{(2)}$ be the probability measure over $\mathbb{R}^N$ that describes the distribution of features in the second layer. Moreover, we introduce function $w^{(2)} : \mathbb{R}^d \times \mathbb{R}^N \to \mathbb{R}$ to denote the weights on the connections from layer 1 to 2. We have for any $z^{(2)} \in \mathrm{supp}(\rho^{(2)})$,

$$\int w^{(2)}(z^{(1)}, z^{(2)})\ \dot{h}\left(\theta_1(z^{(1)})\right) d\rho^{(1)}(z^{(1)}) = z^{(2)}, \tag{14}$$

[2]The state of the first layer can also be characterized by the output of the linear mapping.

where $\dot{h}:\mathbb{R}^N\to\mathbb{R}^N$ is the entrywise operation of the activation function h such that $\dot{h}(\mathbf{a})_i=h(\mathbf{a}_i)$ for $i\in[N]$ and $\mathbf{a}\in\mathbb{R}^N$.

(4) Similarly, for $\ell\in[3:L]$, let $\mathcal{Z}^{(\ell)}=\mathbb{R}^N$ be the space of the features and let $\rho^{(\ell)}$ be the probability measure over $\mathbb{R}^N$ that describes the distribution of the features in the ℓ-th layer. We also introduce function $w^{(\ell)}:\mathbb{R}^N\times\mathbb{R}^N\to\mathbb{R}$ to denote the weights on the connections from layer $\ell-1$ to ℓ. Then for any $z^{(\ell)}\in\mathrm{supp}(\rho^{(\ell)})$, we have

$$\int w^{(\ell)}(z^{(\ell-1)},z^{(\ell)})\ \dot{h}\left(z^{(\ell-1)}\right)d\rho^{(\ell-1)}(z^{(\ell-1)})=z^{(\ell)}. \tag{15}$$

(5) Finally, introducing $u:\mathbb{R}^N\to\mathbb{R}^k$, we have

$$\int u(z^{(L)})\ \dot{h}\left(z^{(L)}\right)d\rho^{(L)}(z^{(L)})=z^{(L+1)}. \tag{16}$$

Therefore the overall learning problem for a continuous DNN (ρ,w,u) can be formulated as the following constrained optimization program:

$$\begin{aligned}\min_{\rho,w,u}\ & Q(\rho,w,u)=\frac{1}{N}\sum_{n=1}^{N}\phi\left(z_n^{(L+1)},y^n\right)+R^{(w,u)}(\rho,w,u)\\ \text{s.t. }& \text{for all } z^{(2)}\in\mathrm{supp}(\rho^{(2)}),\\ &\int w^{(2)}(z^{(1)},z^{(2)})\ \dot{h}\left(\theta_1(z^{(1)})\right)d\rho^{(1)}(z^{(1)})=z^{(2)},\\ &\text{for all } z^{(\ell)}\in\mathrm{supp}(\rho^{(\ell)}),\ \ell\in[3:L],\\ &\int w^{(\ell)}(z^{(\ell-1)},z^{(\ell)})\ \dot{h}\left(z^{(\ell-1)}\right)d\rho^{(\ell-1)}(z^{(\ell-1)})=z^{(\ell)},\\ &\int u(z^{(L)})\ \dot{h}\left(z^{(L)}\right)d\rho^{(L)}(z^{(L)})=z^{(L+1)}.\end{aligned} \tag{17}$$

C. Convex Reformulation

In Program (17), $z^{(\ell)}$ can be arbitrary element in $\mathrm{supp}(\rho^{\ell})$. Thus there is an infinite number of constraints. The optimization variables are ρ, w, and u. The program is still non-convex since there are multiplications of variables in the constraints. We show it is possible to re-parameterize the program as a convex optimization, which will lead to a great simplification for the landscape analysis of DNNs.

To begin with, we first introduce some basic distributions $\rho_0=\{\rho_0^{(1)},\dots,\rho_0^{(L)}\}$. In general, $\rho_0^{(\ell)}$ can be set as any distribution. We consider narrowing down the search space of the optimization variable to $\Theta_{\rho_0}=\{(\rho,w,u):\rho^{(\ell)}\sim\rho_0^{(\ell)},\ell\in[L]\}$. This does not degenerate the problem much as we can pick ρ_0 as standard gaussian distributions for example. We then define

$$p^{(\ell)}(z^{(\ell)})=\frac{d\rho^{(\ell)}}{d\rho_0^{(\ell)}}(z^{(\ell)}),\quad \ell\in[L],$$

as the Radon-Nikodym derivative of $\rho^{(\ell)}$ with respect to $\rho_0^{(\ell)}$ for all $\ell\in[L]$. Then in the search space $\{(\rho,w,u):\rho^{(\ell)}\sim\rho_0^{(\ell)},\ell\in[L]\}$, we have $0<p^{(\ell)}(z^{(\ell)})<\infty$ $\rho_0^{(\ell)}$-a.e.. Furthermore, we consider a change of variables below to eliminate the explicit multiplication between variables:

$$\tilde{w}^{(2)}(z^{(1)},z^{(2)})=w^{(2)}(z^{(1)},z^{(2)})p^{(1)}(z^{(1)})p^{(2)}(z^{(2)}), \tag{18}$$

$$\text{for } \ell\in[2:L-1],\quad \tilde{w}^{(\ell+1)}(z^{(\ell)},z^{(\ell+1)})=w^{(\ell+1)}(z^{(\ell)},z^{(\ell+1)})p^{(\ell)}(z^{(\ell)})p^{(\ell+1)}(z^{(\ell+1)}), \tag{19}$$

$$\tilde{u}(z^{(L)})=u(z^{(L)})p^{(L)}(z^{(L)}). \tag{20}$$

This re-parameterization can significantly simplify the program in the sense that all the "hardness" of the problem is now wrapped into the regularizer. Specifically, by plugging (18), (19), and (20) into Program (17), we have that

$$\begin{aligned}\min_{p,\tilde{w},\tilde{u}}\ & \tilde{Q}(p,\tilde{w},\tilde{u})=\frac{1}{N}\sum_{n=1}^{N}\phi\left(z_n^{(L+1)},y^n\right)+\tilde{R}(p,\tilde{w},\tilde{u})\\ \text{s.t. }& \text{for all } z^{(2)}\in\mathbb{R}^N,\ z^{(2)}p^{(2)}(z^{(2)})\\ &=\int\tilde{w}^{(2)}(z^{(1)},z^{(2)})\ \dot{h}\left(\theta_1(z^{(1)})\right)d\rho_0^{(1)}(z^{(1)}),\\ &\text{for all } z^{(\ell)}\in\mathbb{R}^N,\ \ell\in[3:L],\ z^{(\ell)}p^{(\ell)}(z^{(\ell)})\\ &=\int\tilde{w}^{(\ell)}(z^{(\ell-1)},z^{(\ell)})\ \dot{h}\left(z^{(\ell-1)}\right)d\rho_0^{(\ell-1)}(z^{(\ell-1)}),\\ &\int\tilde{u}(z^{(L)})\ \dot{h}\left(z^{(L)}\right)d\rho_0^{(L)}(z^{(L)})=z^{(L+1)},\end{aligned} \tag{21}$$

where

$$\tilde{R}(p,\tilde{w},\tilde{u})=\sum_{\ell=1}^{L}\lambda^{(\ell)}\tilde{R}_w^{(\ell)}(p,\tilde{w})+\lambda^{(u)}\tilde{R}^{(u)}(p,\tilde{u}), \tag{22}$$

with

$$\tilde{R}_w^{(\ell)}(p,\tilde{w})=\int\frac{\left(\int|\tilde{w}^{(\ell)}(z^{(\ell)},z^{(\ell-1)})|d\rho_0^{(\ell)}(z^{(\ell)})\right)^2}{p^{(\ell-1)}(z^{(\ell-1)})}d\rho_0^{(\ell-1)}(z^{(\ell-1)}), \tag{23}$$

$$\tilde{R}^{(u)}(p,\tilde{u})=\int\frac{\left\|\tilde{u}(z^{(L)})\right\|^2}{p^{(L)}(z^{(L)})}d\rho_0^{(L)}(z^{(L)}). \tag{24}$$

Program (21) only has linear constraints. Next, we show one of the core results of this paper which states that for $\ell_{1,2}$ norm regularizer, Program (21) is convex.

Theorem 3: Assume $\phi(\cdot;\cdot)$ is convex on the first argument, then Program (21) is joint convex.

Remark 4: Theorem 3 can hold more generally. For $\ell_{1,r}$ norm regularizer with $r\geq 2$, Program (21) is convex.

Observe that in the space $\{(\rho,w,u):\rho^{(\ell)}\sim\rho_0^{(\ell)},\ell\in[L]\}$, our new reparameterization $(p,\tilde{w},\tilde{u})$ has one-to-one correspondence with the original parameterization (ρ,w,u). Since the objective function is convex under the reparameterization, we conclude that a local solution of NN in the original parameterization is a global solution.

Theorem 5: Under the assumptions of Theorem 3, in the space $\{(\rho,w,u):\rho^{(\ell)}\sim\rho_0^{(\ell)},\ell\in[L]\}$, if (ρ_*,w_*,u_*) is a critical point (17), then (ρ_*,w_*,u_*) attains the global minimum of (17).

Theorem 5 sheds light on the landscape of the continuous DNN, which shows the non-existence of bad local minima in the space $\{(\rho,w,u):\rho^{(\ell)}\sim\rho_0^{(\ell)},\ell\in[L]\}$. We may impose an additional appropriate convex regularizer $R'(\rho)$ to restrict the

variables into the desired space. For example, we may consider $D_{\mathrm{KL}}\left(\rho^{(\ell)}\|\rho_0^{(\ell)}\right)+D_{\mathrm{KL}}\left(\rho_0^{(\ell)}\|\rho^{(\ell)}\right)$, where $D_{\mathrm{KL}}(\cdot\|\cdot)$ denotes relative entropy and $\ell\in[L]$. Then the nonequivalence of $\rho^{(\ell)}$ and $\rho_0^{(\ell)}$ would push the objective function to $+\infty$.

VI. Implications

A. Convex Feature Learning

In the view of NTK, DNNs in effect only exploit the random features. In contrast, we show under $\ell_{1,2}$ norm, features are learned from the underlying task. Specifically, by the convexity argument in Theorem 3 with the relationship of the re-parameterization built in Theorem 5, we can establish specific properties satisfied by the optimal solutions of DNNs.

Proposition 1: Under the assumptions of Theorem 3, in the space $\{(\rho,w,u):\rho^{(\ell)}\sim\rho_0^{(\ell)},\ell\in[L]\}$, if (ρ_*,w_*,u_*) is an optimal solution of the DNN, and $\rho_*^{(\ell)}\sim\rho_0^{(\ell)}$, then there exists a real number sequence $\{\Lambda\}_{\ell\in[L]}$, i.e. $\Lambda^{(\ell)}\in\mathbb{R}$ for all $\ell\in[L]$, so that the following equations hold: (1) for all $\ell\in[L-1]$ and $\check{z}^{(\ell)}\in\mathcal{Z}^{(\ell)}$, we have

$$\begin{aligned}&\lambda^{(\ell+1)}\left(\int\left|w^{(\ell+1)}(\check{z}^{(\ell+1)},\check{z}^{(\ell)})\right|d\rho_*^{\ell+1}(\check{z}^{(\ell+1)})\right)^2\\ =&2\lambda^{\ell}B(\check{z}^{(\ell)})+\Lambda^{(\ell)};\end{aligned}$$

(2) for all $\check{z}^{(L)}\in\mathcal{Z}^{(L)}$, we have

$$\Lambda^{(L)}+2\lambda^{(L)}B(\check{z}^{(L)})=\lambda^{(u)}\left\|u_*(\check{z}^{(L)})\right\|^2,$$

and

$$\frac{1}{N}\sum_{n=1}^{N}\phi'(z_n^{(L+1)},y^n)[\dot{h}(\check{z}^{(L)})]_n=-2\lambda^{(u)}u_*(\check{z}^{(L)}),$$

where

$$\begin{aligned}B(\check{z}^{(\ell)})=&\iint\left|w^{(\ell)}(z^{(\ell)},\check{z}^{(\ell-1)})\right|d\rho_*^{\ell}(z^{(\ell)})\\&\left|w^{(\ell)}(\check{z}^{(\ell)},\check{z}^{(\ell-1)})\right|d\rho_*^{(\ell-1)}(\check{z}^{(\ell-1)})\end{aligned}$$

for $\ell\in[2:L]$. In the above formulas, we abbreviate brackets in the superscripts.

Proposition 1 shows that the optimal feature distribution sequence ρ_* relies on (w_*,u_*). There are lots of triples (ρ,w,u) that can reach the same layer of the training loss as (ρ_*,w_*,u_*) does, whereas (ρ_*,w_*,u_*) is the one that achieves the minimum $\ell_{1,2}$ norm regularization value under this equivalent class. Since the $\ell_{1,2}$ norm regularization upper bounds the variance of discrete approximation of the continuous DNN in Theorem 2, a small $\ell_{1,2}$ norm implies that a small number of hidden units are needed to represent $f(\rho_*,w_*,u_*;\cdot)$ in the randomly sampled discrete DNN. This means that $\ell_{1,2}$ norm regularization leads to efficient feature representations. Proposition 1 will be validated in our experiment.

B. Relationship With Gradient Descent

We study the relationship of the convex reformulation and Gradient Descent in the original weight space in this section. We consider the Scaled Mini-norm (Sub)-Gradient Descent algorithm with the meta-algorithm shown in Algorithm 1. The sub-gradient $\hat{\mathcal{G}}$ is composed of the part from the loss and the regularizers. The loss part is differentiated and can be obtained by the standard backward-propagation algorithm, whereas the $\ell_{1,2}$ norm is not differentiated. However, it has sub-gradients at every point thanks to its convexity. We choose the sub-gradient that has the minimum norm (see (25) and (26)) as $\hat{\mathcal{G}}$. We also study a different step size for layers compared with the standard Gradient Descent algorithm. These appropriate step sizes (time scales) for the parameters match the scale in the continuous limit and are also adopted in all existing mean-field theory of DNNs [28], [44], [45].

When the step size goes to 0, we can define the corresponding gradient flow for training discrete NNs by:

$$\begin{aligned}\frac{dw_{i,j}^{(\ell),t}}{dt}&=-\left[m_{\ell-1}m_\ell\right]\hat{\mathcal{G}}^t_{w_{i,j}^{(\ell)}},\ \ell\in[L],\ i\in[m_{\ell-1}],\ j\in[m_\ell],\\ \frac{du_i^t}{dt}&=-\left[m_L\right]\hat{\mathcal{G}}^t_{u_i}\quad i\in[m_L],\end{aligned}$$

where the mini-norm sub-gradient $\hat{\mathcal{G}}^t$ satisfies that

$$\begin{aligned}\hat{\mathcal{G}}^t=&\nabla\hat{L}(w^t,u^t)\\&+\operatorname*{argmin}_{\xi}\left\{\left\|\nabla\hat{L}(w^t,u^t)+\xi\right\|:\xi\in\partial\hat{R}(w^t,u^t)\right\}\end{aligned}\tag{25}$$

and $\hat{\mathcal{G}}^t_{w_{i,j}^{(\ell)}}$ and $\hat{\mathcal{G}}^t_{u_i}$ are the corresponding elements of $\hat{\mathcal{G}}^t$.

Note the above gradient flow exists a unique solution, which can be obtained by the same argument in Proposition 2.3 in [8] (also see Section 2.1 in [46]) due to the choice of the mini-norm sub-gradient for the regularizers. And the standard results from gradient flow indicate the monotonic non-increase of the objective. That is

$$\frac{d\hat{Q}^t}{dt}=-\sum_{\ell=1}^{L}\sum_{i=1}^{m_{\ell-1}}\sum_{j=1}^{m_\ell}m_{\ell-1}m_\ell\left\|\hat{\mathcal{G}}^t_{w_{i,j}^{(\ell)}}\right\|^2-\sum_{i=1}^{m_L}m_L\left\|\hat{\mathcal{G}}^t_{u_i}\right\|^2.$$

Now consider the continuous limit where all the number of hidden units diverges. When there is no regularizer, it was shown in [44] that the discrete gradient flow provably converges to a continuous gradient flow called neural feature flow, which is a solution of an infinitely dimensional non-linear dynamic ODE system under suitable conditions. Generally speaking, neural feature flow tracks the trajectories for the weights and the features to capture the learning process of the continuous DNNs.

In the continuous DNN formulation (17), the network is parameterized with respect to (ρ,w,u). However, the gradient flow only modifies the network parameters (w,u) but does not modify ρ directly. But it is clear that changing (w,u) modifies ρ accordingly. Specifically, we can write down the continuous sub-gradients. Specifically, by back-propagation, for each training sample $n\in[N]$, we introduce intermediate terms[3]:

$$\begin{aligned}&D_n^{(L+1)}(\rho,u;z^{(L+1)})=\nabla_1\phi(z_n^{(L+1)},y^n),\\&D_n^{(L)}(\rho,w,u;z^{(L)})h'(z_n^{(L)})\left(u(z^{(L)})^\top D_n^{(L+1)}(\rho,u;z^{(L+1)})\right),\end{aligned}$$

[3] $D_n^{(\ell)}$ depend only on the components of ρ and w after ℓ-th layer.

Algorithm 1 Scaled Gradient Descent for Training a DNN

1: Input the data $\{x^i, y^i\}_{i=1}^N$, step size η, and initial weights (w_0, u_0).
2: **for** $k = 0, 1, \ldots, K-1$ **do**
3: Perform forward-propagation to compute the loss.
4: Perform backward-propagation to compute the gradient $(\hat{\mathcal{G}}_{w_k}, \hat{\mathcal{G}}_{u_k})$.
5: Perform scaled Gradient Descent:

$$\begin{aligned} w_{k+1,i,j}^{(\ell)} &= w_{k,i,j}^{(\ell)} - \big[\eta m_{\ell-1} m_\ell\big] \hat{\mathcal{G}}_{w_{k,i,j}^{(\ell)}} \quad \ell \in [L],\ i \in [m_{\ell-1}],\ j \in [m_\ell], \\ u_{k+1,i} &= u_{k,i} - \big[\eta m_L\big] \hat{\mathcal{G}}_{u_{k,i}} \quad i \in [m_L]. \end{aligned}$$

6: **end for**
7: Output the weights (w_K, u_K).

$D_n^{(\ell)}(\rho, w, u; z^{(\ell)}) = h'(z_n^{(\ell)}))\cdot$
$\Big(\int w^{(\ell+1)}(\check{z}^{(\ell+1)}, z^{(\ell)} D_n^{(\ell+1)}(\rho, w, u; \check{z}^{(\ell+1)}) d\rho^{(\ell+1)}(\check{z}^{(\ell+1)})\Big)$,

where $\nabla_1 \phi : \mathbb{R}^k \to \mathbb{R}^k$ is the gradient of ϕ on the first argument, $z_n^{(\ell)}$ is n-th coordinate of $z^{(\ell)} \in \mathbb{R}^n$. Then we have for all $\ell \in [L]$

$$\begin{aligned} \mathcal{G}^{(\ell)}(\rho, w, u; z^{(\ell)}, z^{(\ell-1)}) = & \frac{1}{N} \sum_{n=1}^N D_n^{(\ell)}(\rho, w, u; z^{(\ell)}) h(z_n^{(\ell-1)}) \\ & + 2\lambda^{(\ell)} V^{(\ell)}(z^{(\ell-1)}) |w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)})|', \end{aligned}$$

and

$$\begin{aligned} & \mathcal{G}^{(u)}(\rho, u; z^{(L)}) \\ = & \frac{1}{N} \sum_{n=1}^N D_n^{(L+1)}(\rho, u; z^{(L+1)}) h(z_n^{(L)}) + 2\lambda^{(u)} u(z^L), \end{aligned}$$

where $V^{(\ell)}(z^{(\ell-1)}) = \int \big|w^{(\ell)}(\check{z}^{(\ell)}, z^{(\ell-1)})\big| \, d\rho^{(\ell)}(\check{z}^{(\ell)})$ and $|w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)})|'$ is properly set to ensure that $\mathcal{G}^{(\ell)}(\rho, w, u; z^{(\ell)}, z^{(\ell-1)})$ has the mini-norm. That is

$$\begin{aligned} & \Big|w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)})\Big|' \qquad (26) \\ = & \begin{cases} 1, & \text{if } w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)}) > 0, \\ 0, & \text{if } w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)}) = 0,\ V^{(\ell)}(z^{(\ell-1)}) = 0, \\ -1, & \text{if } w^{(\ell)}(z^{(\ell)}, z^{(\ell-1)}) < 0 \\ \max(\min(-\frac{\frac{1}{N}\sum_{n=1}^N D_n^{(\ell)}(\rho, w, u; z^{(\ell)}) h(z_n^{(\ell-1)})}{2\lambda^{(\ell)} V^{(\ell)}(z^{(\ell-1)})}, 1), -1), & \text{else.} \end{cases} \end{aligned}$$

When there are suitable regularizers, if the solution of the neural feature flow also exists, then analogously to discrete gradient flow, we have

$$\begin{aligned} & \frac{dQ^t}{dt} \\ = & -\sum_{\ell=1}^L \int \Big\| \mathcal{G}^{(\ell)}(\rho_t, w_t, u_t; z^{(\ell)}, z^{(\ell-1)}) \Big\|^2 \\ & \times d\rho_t^{(\ell)}(z^{(\ell)}) d\rho_t^{(\ell-1)}(z^{(\ell-1)}) \\ & - \int \Big\| \mathcal{G}^{(u)}(\rho_t, u_t; z^{(L)}) \Big\|^2 d\rho_t^{(L)}(z^{(L)}). \qquad (27) \end{aligned}$$

Now we give intuitions on why this continuous gradient flow achieves a global minimum based on our convex reformulation. Suppose there is a point (ρ_*, w_*, u_*). We show if $\mathcal{G}^{(\ell)}(\rho_*, w_*, u_*; z^{(\ell-1)}, z^{(\ell)}) = 0$ and $\mathcal{G}^{(u)}(\rho_*, w_*, u_*; z^{(L)}) = 0$ holds almost surely, it converges to the solution of the convex reformulation. Note that because the gradient flow only directly updates the neural network original parameters (w, u), it is not clear that (ρ_*, w_*, u_*) will be a local minima in (17). Hence Theorem 5 is not directly applicable.

Theorem 6: Under the assumptions of Theorem 3, in the space $\{(\rho, w, u) : \rho^{(\ell)} \sim \rho_0^{(\ell)}, \ell \in [L]\}$, for all $\ell \in [L]$, if $\mathcal{G}^{(\ell)}(\rho_*, w_*, u_*; z^{(\ell-1)}, z^{(\ell)}) = 0$ and $\mathcal{G}^{(u)}(\rho_*, w_*, u_*; z^{(L)}) = 0$ holds almost surely, then (ρ_*, w_*, u_*) achieves the global minimal of Programs (17) and (21).

Theorem 6 only provides some intuitions on why this continuous gradient flow achieves a global minimum. We should mention that there are still several challenges to rigorously prove necessary results. The main challenges are listed below and left as open questions for future research.

1) The biggest challenge is to prove that
$\int \Big\| \mathcal{G}^{(\ell)}(\rho_t, w_t, u_t; z^{(\ell)}, z^{(\ell-1)}) \Big\|^2 d\rho_0^{(\ell)}(z^{(\ell)}) d\rho_0^{(\ell-1)}(z^{(\ell-1)})$
$\to 0$ (and $\int \big\| \mathcal{G}^{(u)}(\rho_t, u_t; z^{(L)}) \big\|^2 d\rho_0^{(L)}(z^{(L)}) \to 0$) with $t \to \infty$. One important step is to prove that $\frac{d\rho_t^{(\ell)}}{d\rho_0^{(\ell)}}(z^{(\ell)})$ will be strictly bounded away from 0 for any $t > 0$. Note that from (27), for some region $R \in \mathbb{R}^N$, if $\rho_t^{(\ell)}(R) = 0$ for $t > t_0$, then $\big\| \mathcal{G}^{(\ell)}(\rho_t, w_t, u_t; z^{(\ell)}, z^{(\ell-1)}) \big\|^2$ may not be 0 on R. As we mentioned before, one possible direction to deal with the issue is to impose additional regularizers, such as relative entropy regularizers, and then to show that $\rho_*^{(\ell)}$ would satisfy some useful properties, such as the Logarithmic Sobolev property. However, we still lack some technique to rigorously achieve this result. Moreover, adding a non-trivial regularizer also brings more difficulties to prove the existence of the solution for neural feature flow. See (3).
2) (27) only ensures (ρ_t, w_t, u_t) converges in a weak sense. Even after we overcome challenge (1), one still cannot show that $Q(\rho_t, w_t, u_t)$ converges to its minimal value (directly using Theorem 6). Nevertheless, we believe this is not a fundamental problem, and a method to deal with this difficulty is to properly smooth (the amount of smoothing can be arbitrarily small) the parameters so that the slightly smoothed distribution has point-wise convergence.

3) When there are non-trivial regularizers, the existence of the solution for the neural feature flow is not rigorously proven. [44] achieves the result only when there are no regularizers by using Picard-Lindelöf theorem (see Theorem 4 in [44]). It is still open how to analyze the evolution of DNNs with non-trivial regularizers such as $\ell_{1,2}$ norm and relative entropy regularizer.
4) Finally, even if we prove the convergence of the continuous dynamics, it still requires some treatment to show the convergence of the discrete dynamics. We suspect that this difficulty can be solved by using the argument from the propagation of chaos, (see e.g. Theorem 6 in [44]).

Despite the challenges, we believe that our explanation would bring some value in understanding the Gradient Descent algorithm to train the DNNs.

VII. Experiments

The experiments are designed to qualitatively verify the following.

1) *Optimality condition*: We demonstrate that fully trained overparameterized DNNs are consistent with our convex reformulation argument by verifying the optimality condition in Proposition 1. Here we consider the relationship between

$$u_j^{(\ell)} = \frac{\lambda^{(\ell)}}{m^{(\ell)} m^{(\ell-1)}} \sum_{k=1}^{m^{(\ell-1)}} \big(\sum_{j'=1}^{m^{(\ell)}} |w_{k,j'}^{(\ell)}| \big) |w_{k,j}^{(\ell)}| \quad (28)$$

and

$$v_j^{(\ell)} = \begin{cases} \lambda^{(\ell+1)} \big(\frac{1}{m^{(\ell+1)}} \sum_{i=1}^{m^{(\ell+1)}} |w_{j,i}^{(\ell+1)}| \big)^2 & \ell \in [L-1] \\ \lambda^{(u)} \|u_j\|^2 & \ell = L \end{cases} \quad (29)$$

for one neuron j in layer $\ell \in [L]$, which are the estimates of

$$\lambda^{(\ell)} \iint |w(\tilde{z}^{(\ell)}, z^{(\ell-1)})| d\rho(\tilde{z}^{(\ell)}) |w(z^{(\ell)}, z^{(\ell-1)})| d\rho(z^{(\ell-1)})$$

and

$$\begin{cases} \lambda^{(\ell+1)} \big(\int |w(z^{(\ell+1)}, z^{(\ell)})| d\rho(z^{(\ell+1)}) \big)^2 & \ell \in [L-1] \\ \lambda^{(u)} \big\| u(z^{(L)}) \big\|^2 & \ell = L, \end{cases}$$

respectively.
2) *Deep versus Shallow Networks*: We show that by increasing L, the number of hidden layers, fully connected NN can learn hierarchical feature representations that can reduce the variance of approximation described in Theorem 2. This verifies the benefit of using deeper networks for certain problems.
3) *Compactness*: We show that compared with other regularizers, the proposed regularizer can learn better (more compact) feature representations.
4) *Convex Landscape*: We visualize the landscape of neural network in our formulation and show it is convex.

Note that similar to [47], we use the approximation variance of discretization $V(w,u)$ to measure the effectiveness of feature representation, based on the theoretical findings of Theorem 2:

$$V(w,u) = \frac{1}{(m^{(L)})^2} \sum_{j=1}^{m^{(L)}} \Big\| u_j \hat{f}_j^{(L)}(x) - \hat{f}(x) \Big\|^2 +$$

$$\mathbb{E}_x \sum_{\ell=2}^{L} \frac{1}{(m^{(\ell-1)})^2} \sum_{j=1}^{m^{(\ell-1)}} \left(\sum_{i=1}^{m^{(\ell)}} a_i(x) \ \big(\hat{f}_j^{(\ell-1)}(x) w_{i,j}^{(\ell)} - \hat{g}_i^{(\ell)}(x) \big) \right)^2,$$

where $a_i(x) = \frac{\partial \hat{f}(x)}{\partial \hat{g}_i^{(\ell)}(x)}$.

A. Synthetic 1-D Regression Task

We begin to empirically validate our claims in a synthetic 1-D regression task. Since the feature representation $f_j^{(\ell)}(x)$ corresponding to each neuron in each layer $\ell \in [L]$ is a single-variable function, it can be easily visualized.

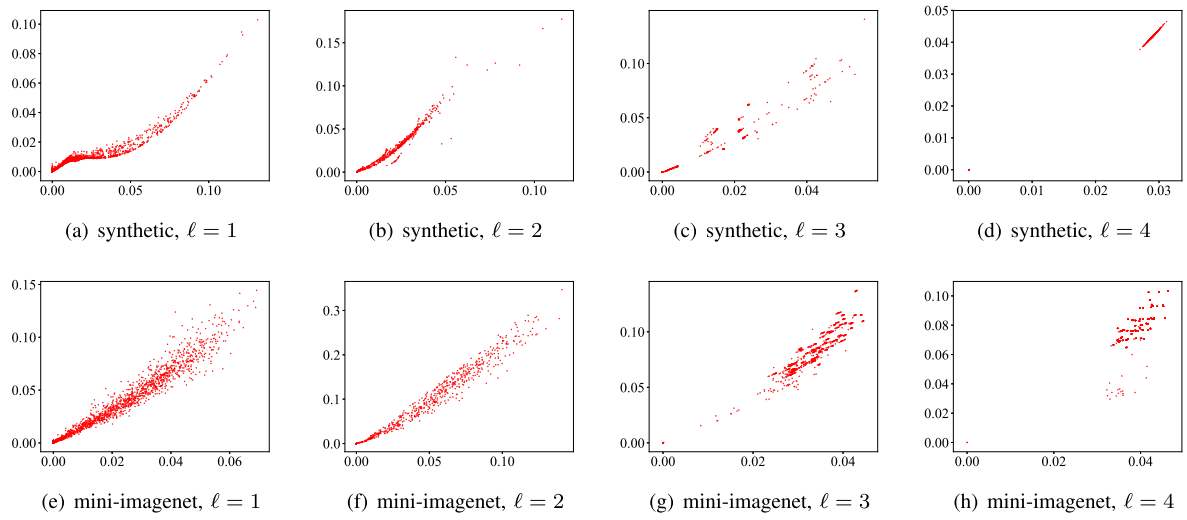
Here we consider the function $f(x) = 2(2\cos^2(x) - 1)^2 - 1$ introduced by [48]. We draw 60k training samples and 60k test samples uniformly from $[-2\pi, 2\pi]$ for x and set $y = f(x)$. We use a fully-connected NN with $m^{(\ell)} = 1000 \times 2^{L-\ell}$ hidden units in each hidden layer ℓ to learn this target function. We take $L = \{1, \cdots, 4\}$, and use the Adam optimizer with an initial learning rate $1e\text{-}4$ in our experiments, and let the activation function be $\sigma(x) = \tanh(x)$. For fair comparison, we tune the hyper-parameters of the weight of regularizer so that for different L, the NN could reach training RMSE of $1e\text{-}4$ at converge. This controls the representation power of the NN.

We first validate that fully trained overparameterized NN satisfies the optimality condition of Proposition 1. Here we consider the case of $L = 4$, and the top row in Fig. 1 plots the estimated quantities $u_j^{(\ell)}$ and $v_j^{(\ell)}$. We can see that these two quantities are approximately linearly correlated, as predicted by Proposition 1.

To compare the performance of shallow versus deep networks, Fig. 2 (a) reports how the approximated variance changes when L increases. It demonstrates that the approximated variance decreases as L increases. Moreover, the approximate variance gap between $L = 2$ and $L = 3$ is very large while that between $L = 3$ and $L = 4$ is small. This is consistent with the fact that the hierarchical composition of the target function $f(x)$ has depth 3 (i.e. $f(x) = h(h(\cos x))$, where $h(u) = u^2 - 1$). At the same time, we can observe that for larger L, the regularizer in subplot (b) and training RMSE in subplot (c) decrease much faster, which also demonstrate the effectiveness of increasing L for this target function.

Fig. 3 shows representative features (as 1D-functions) at each layer after convergence. We reach the following conclusion from visualization of different L: DNN is able to learn hierarchical feature representations when we take optimization process into consideration. To be more specific, the layer next to the input layer tends to learn low-frequency signals while the upper layers take these lower-frequency signals to form higher-frequency signals.

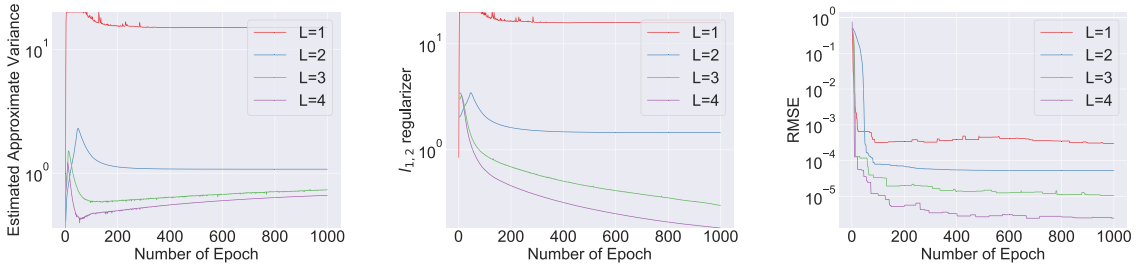
We further compare the compactness of different regularizers. Here we use the notation $\ell_{a,b}$ to represent the regularizer



(a) synthetic, $\ell = 1$ (b) synthetic, $\ell = 2$ (c) synthetic, $\ell = 3$ (d) synthetic, $\ell = 4$

(e) mini-imagenet, $\ell = 1$ (f) mini-imagenet, $\ell = 2$ (g) mini-imagenet, $\ell = 3$ (h) mini-imagenet, $\ell = 4$

Fig. 1. Property of optimal solution for $L = 4$ hidden-layer fully connected NN. The top (synthetic 1-d regression) and bottom (mini-imagenet) rows are scatter-plots of the estimated quantities for each sampled neuron after convergence. For each subplot, one point (x, y) represents one sampled neuron $j \in [m^{(\ell)}]$, where $x = u_j^{(\ell)}$ and $y = v_j^{(\ell)}$ defined in (28) and (29), respectively. The correlation coefficient of these points for the top four figures are 0.88, 0.82, 0.92 and 0.99 from left to right, and that for the bottom four figures are 0.87, 0.96, 0.92 and 0.86 from left to right, which demonstrates the existence of strong linear association between $u_j^{(\ell)}$ and $v_j^{(\ell)}$.



(a) (b) (c)

Fig. 2. The visualization of optimization process when training multilayer NNs with different L.

in the form of $r_1(x) = |x|^a$ and $r_2(x) = |x|^b$. $\ell_{1,2}$ regularizer is the proposed regularizer, which is an upper-bound of the approximation variance. $\ell_{2,1}$ regularizer is the traditional L_2 regularizer (i.e. ℓ_2 weight decay). From Fig. 4, we found that the proposed regularizer leads to sparser weights, and thus has a more compact representation. We have also tried the $\ell_{1/2,4}$ regularizer, and found that the sparsity of $\ell_{1/2,4}$ regularizer isn't significantly better than the proposed $\ell_{1,2}$ regularizer. This verifies the effectiveness of the proposed regularizer to obtain sparse weights because by decreasing parameter from 1 to $1/2$ cannot significantly increase the sparisity of weights as shown in Fig. 4.

B. Mini-Imagenet Classification Task

We have also performed experiments on real data. Mini-Imagenet dataset is a simplified version of ILSVRC'12 dataset [49], which consists of 600 84 $\times$ 84 $\times$ 3 images sampled from 100 classes each. Here we consider the data split introduced by [50], which consists of 64 classes and 38.4k images as our full dataset. We divide the dataset into train/valid/test split by 7:1:2.

Since fully-connected NNs do not have the capacity to deal with such image data, we first train a base CNN embedding network with a four block architecture as in [51]. We then take the 1600-dimensional output of the embedding layer and feed it to an L layer NN for classification. The training configurations and network architectures are the same as those for the synthetic 1-D experiment, except that we tune the regularization parameters to achieve the best validation accuracy. Since the feature function of this task is hard to visualize, we only consider the *optimality condition*, *shallow versus deep networks* and *compactness*.

Similar to the results in the synthetic 1-D experiment, the sub-figures in the bottom row of Fig. 1 show that the two quantities we care about are also linearly correlated in each layer, which is consistent with our theory. Fig. 5 reports how approximation variance, test RMSE, and train RMSE change during the model training procedure. We can see that the approximation variance decreases as L increases, and the gap between $L = 1$ and $L = 2$ is very large. This demonstrates the benefits of using deeper neural network. Moreover, the generalization performance also increases as L increases.

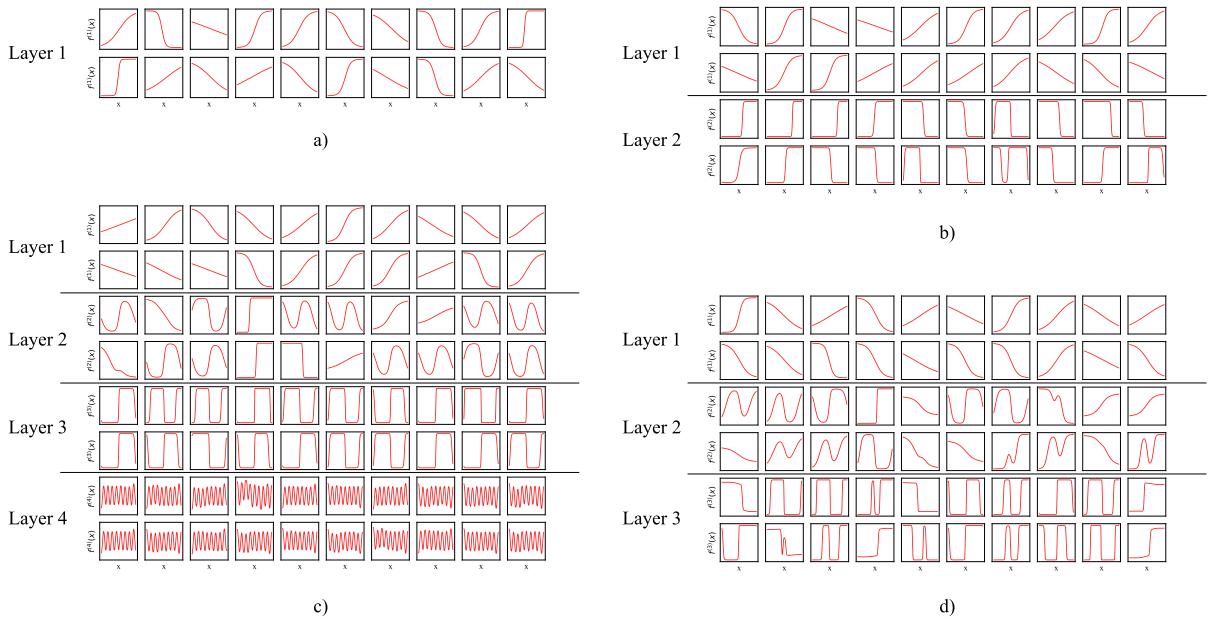
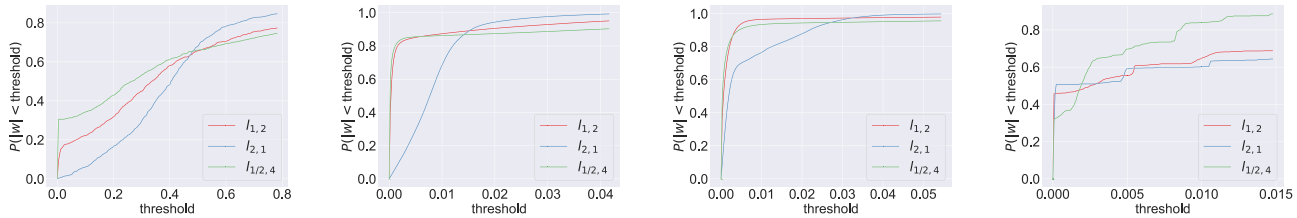
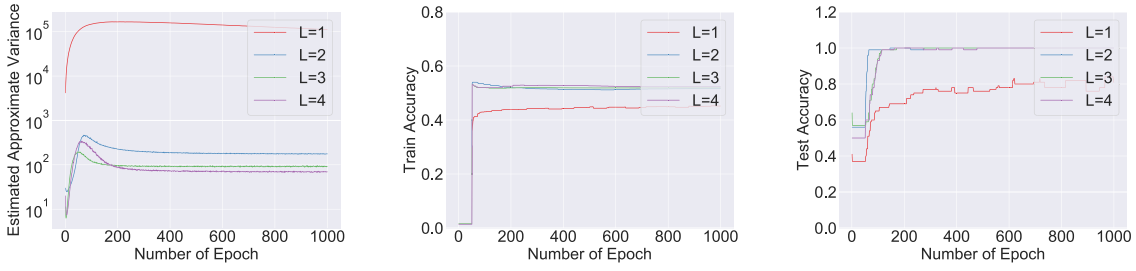


Fig. 3. Representative features as 1-D functions for fully-connected NNs with different hidden-layers: a) 1-hidden-layer b) 2-hidden-layers c) 3-hidden-layers d) 4-hidden-layers. For each architecture, we sample 20 neurons from each layer and plot the single-variable feature function $f_j^{(\ell)}(x)$ of the sampled neurons.



(a) weights of $w^{(1)}$ (b) weights of $w^{(2)}$ (c) weights of $w^{(3)}$ (d) weights of u

Fig. 4. The comparison of sparsity of weights in different layers between different regularizers. The i-th figure demonstrate the sparsity of weights in i-th layer for different regularizers. The point (x, y) on one curve suggests that there are $100 \times y\%$ weights in such layer lies in $[-x, x]$. We could see that the red curve is lower-bounded by the blue one, which suggests that our proposed regularizer will result in more sparse representations compared with traditional regularizer.



(a) (b) (c)

Fig. 5. Visualization of the optimization process when training multilayer NNs with different L.

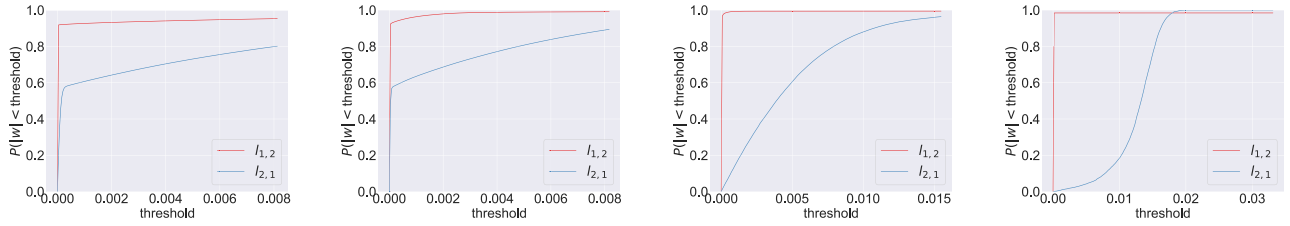
Fig. 6 shows that the proposed $\ell_{1,2}$ regularizer leads to more compact weights for each layer than the traditional $\ell_{2,1}$.

C. Visualization of Landscape

Given three discrete neural network θ_1, θ_2, and θ_3. In this section, we conduct experiments to show that the feature distribution linear interpolation between these three neural networks, i.e., the function

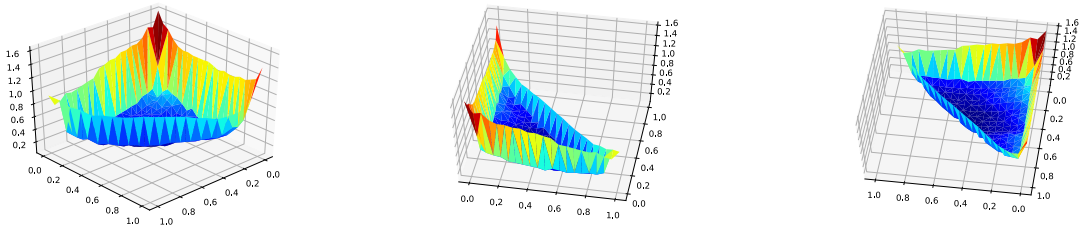
$$L(\alpha, \beta) = Q(\breve{\rho}_{\alpha,\beta}, \breve{w}_{\alpha,\beta}, \breve{u}_{\alpha,\beta})$$

is convex. Here the interpolation neural network $\breve{\theta}_{\alpha,\beta} = (\breve{\rho}_{\alpha,\beta}, \breve{w}_{\alpha,\beta}, \breve{u}_{\alpha,\beta})$ satisfies $\rho_{\alpha,\beta}^{(\ell)} = \alpha\rho_{\theta_1}^{(\ell)} + \beta\rho_{\theta_2}^{(\ell)} + (1-\alpha-\beta)\rho_{\theta_3}^{(\ell)}$, which means its feature distribution at each layer is close to the mixture of empirical feature distributions of θ_1,



(a) weights of $w^{(1)}$ (b) weights of $w^{(2)}$ (c) weights of $w^{(3)}$ (d) weights of u

Fig. 6. Comparison of weight sparsity at different layers for different regularizers on the Mini-Imagenet classification task. The i-th figure demonstrates the weight sparsity at layer i for different regularizers. A point (x, y) indicates that there are $100 \times y\%$ weight values lying in $[-x, x]$. We can see that the red curve dominates the blue one, which suggests that the proposed regularizer leads to sparser solutions than those of the traditional ℓ_2 regularizer.



(a) (b) (c)

Fig. 7. Different views of landscape of neural network in feature distribution view: $L(\alpha, \beta)$: for any point (x, y, z) on the surface, (x, y) represents coefficient $(\alpha, \beta) = (x, y)$ and z represents the loss value of $\breve{\theta}_{\alpha,\beta}$.

θ_2, and θ_3 with coefficient α, β, and $1 - \alpha - \beta$, and

$$\breve{w}_{\alpha,\beta}, \breve{u}_{\alpha,\beta} = \text{argmin}_{w,u}\{Q(\breve{\rho}_{\alpha,\beta}, w, u) : (\breve{\rho}_{\alpha,\beta}, w, u) \text{ satisfies constraints}\}$$

chosen to satisfy all constraints and attain the minimal loss value. This supports our claim that the loss function with respect to the feature distributions at each layer is convex in our feature distribution view.

We use same dataset and similar training configurations as what we do for 1-D synthetic regression task. To make three neural network θ_1, θ_2, and θ_3 have different and meaningful feature distributions, we let θ_1, θ_2, and θ_3 be neural network that was fully trained on data whose features x lies in three disjoint sets D_1, D_2, and D_3, respectively. So they all learn parts of but not perfect features about the regression task. The intuitive idea of constructing the interpolation neural network $\breve{\theta}_{\alpha,\beta}$ is to make sure its pre-activation set close to the mixture of that of θ_1, θ_2 and θ_3 with corresponding coefficients, where the pre-activation set of θ is defined as

$$\hat{\mathcal{V}}_\theta^{(\ell)} = \{v_j^{(\ell)} = [\hat{g}_j^{(\ell)}(\theta; x_1), \cdots, \hat{g}_j^{(\ell)}(\theta; x_N)] : j \in [m^{(\ell)}]\}.$$

A layerwise optimization-based algorithm is designed to construct $\breve{\theta}_{\alpha,\beta}$ in Appendix D. The result is shown in Fig. 7. For any point (x, y, z) on the surface, (x, y) represents coefficient $(\alpha, \beta) = (x, y)$ and z represents the loss value of $\breve{\theta}_{\alpha,\beta}$. The three corners are corresponding to the loss of neural network θ_1, θ_2 and θ_3 and the other points in the simplex in (x, y) plane are corresponding to the loss of neural network $\breve{\theta}_{\alpha,\beta}$ with specific interpolation coefficients. We can see from the image that the landscape of neural network in feature distribution view is unimodal and it looks convex.

VIII. Conclusion

This paper analyzed over-parameterized DNNs and showed that it is possible to reformulate overparameterized DNNs as convex systems. Moreover, when fully trained, DNNs learn effective feature representations suitable for the underlying learning task via regularization. Our analysis is consistent with empirical observations. Our newly introduced method paves the way for establishing global convergence results of standard optimization algorithms such as (noisy) gradient descent for overparameterized DNNs. We will leave the study as a future work.

Acknowledgment

The authors would like to thank Jason Lee, Xiang Wang, and Pengkun Yang for very helpful discussions.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[4] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.

[5] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4694–4702.

[6] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.

[7] S. Mei, A. Montanari, and P.-M. Nguyen, "A mean field view of the landscape of two-layer neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 33, pp. E7665–E7671, 2018.

[8] L. Chizat and F. Bach, "On the global convergence of gradient descent for over-parameterized models using optimal transport," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 3036–3046.

[9] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1675–1685.

[10] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 242–252.
[11] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning(still) requires rethinking generalization," *Commun. ACM*, vol. 64, no. 3, pp. 107–115, 2021.
[12] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833.
[13] T. Wiatowski and H. Bölcskei, "A mathematical theory of deep convolutional neural networks for feature extraction," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1845–1866, Mar. 2018.
[14] M. Hardt and T. Ma, "Identity matters in deep learning," in *Proc. Int. Conf. Learn. Represent.*, 2016.
[15] C. D. Freeman and J. Bruna, "Topology and geometry of half-rectified network optimization," 2016, *arXiv:1611.01540*.
[16] A. Brutzkus and A. Globerson, "Globally optimal gradient descent for a ConvNet with Gaussian inputs," 2017, *arXiv:1702.07966*.
[17] R. Ge, J. D. Lee, and T. Ma, "Learning one-hidden-layer neural networks with landscape design," 2017, *arXiv:1711.00501*.
[18] A. Bakshi, R. Jayaram, and D. P. Woodruff, "Learning two layer rectified neural networks in polynomial time," 2018, *arXiv:1811.01885*.
[19] M. Soltanolkotabi, A. Javanmard, and J. D. Lee, "Theoretical insights into the optimization landscape of over-parameterized shallow neural networks," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 742–769, Feb. 2019.
[20] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points—Online stochastic gradient for tensor decomposition," in *Proc. Annu. Conf. Learn. Theory*, 2015, pp. 797–842.
[21] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1724–1732.
[22] C. Fang, C. J. Li, Z. Lin, and T. Zhang, "SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator," in *Advances in Neural Information Processing Systems*, vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, 2018.
[23] C. Fang, Z. Lin, and T. Zhang, "Sharp analysis for nonconvex SGD escaping from saddle points," in *Proc. Annu. Conf. Learn. Theory*, 2019, pp. 1192–1234.
[24] J. Sirignano and K. Spiliopoulos, "Mean field analysis of neural networks: A central limit theorem," *Stochastic Processes Their Appl.*, vol. 130, no. 3, pp. 1820–1852, Mar. 2020.
[25] G. M. Rotskoff and E. Vanden-Eijnden, "Trainability and accuracy of neural networks: An interacting particle system approach," May 2018, *arXiv:1805.00915*.
[26] S. Mei, T. Misiakiewicz, and A. Montanari, "Mean-field theory of two-layers neural networks: Dimension-free bounds and kernel limit," in *Proc. Annu. Conf. Learn. Theory*, 2019, pp. 2388–2464.
[27] C. Wei, J. D. Lee, Q. Liu, and T. Ma, "Regularization matters: Generalization and optimization of neural nets vs their induced kernel," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9712–9724.
[28] H. T. Pham and P.-M. Nguyen, "Global convergence of three-layer neural networks in the mean field regime," in *Proc. Int. Conf. Learn. Represent.*, 2020.
[29] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018.
[30] Y. Li and Y. Liang, "Learning overparameterized neural networks via stochastic gradient descent on structured data," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
[31] S. Arora, S. S. Du, W. Hu, Z. Li, and R. Wang, "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 322-332.
[32] L. Su and P. Yang, "On learning over-parameterized neural networks: A functional approximation prospective," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[33] J. Lee, J. Sohl-Dickstein, J. Pennington, R. Novak, S. Schoenholz, and Y. Bahri, "Deep neural networks as Gaussian processes," in *Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: https://openreview.net/forum?id=B1EA-M-0Z
[34] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
[35] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in over-parameterized neural networks, going beyond two layers," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[36] D. Zou, Y. Cao, D. Zhou, and Q. Gu, "Stochastic gradient descent optimizes over-parameterized deep relu networks," *Mach. Learn.*, vol. 109, pp. 467–492, 2018.
[37] T. V. Nguyen, R. K. W. Wong, and C. Hegde, "Benefits of jointly training autoencoders: An improved neural tangent kernel analysis," *IEEE Trans. Inf. Theory*, vol. 67, no. 7, pp. 4669–4692, Jul. 2021.
[38] M. Pilanci and T. Ergen, "Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7695–7705.
[39] T. Ergen and M. Pilanci, "Implicit convex regularizers of CNN architectures: Convex optimization of two- and three-layer networks in polynomial time," 2020, *arXiv:2006.14798*.
[40] T. Ergen and M. Pilanci, "Convex geometry and duality of over-parameterized neural networks," *J. Mach. Learn. Res.*, vol. 22, no. 212, pp. 1–63, 2021.
[41] B. Bartan and M. Pilanci, "Training quantized neural networks to global optimality via semidefinite programming," 2021, *arXiv:2105.01420*.
[42] T. Ergen and M. Pilanci, "Revealing the structure of deep neural networks via convex duality," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 3004–3014.
[43] Y. Gu, W. Zhang, C. Fang, J. D. Lee, and T. Zhang, "How to characterize the landscape of overparameterized convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3797–3807.
[44] C. Fang, J. D. Lee, P. Yang, and T. Zhang, "Modeling from features: A mean-field framework for over-parameterized deep neural networks," in *Proc. Annu. Conf. Learn. Theory*, 2021, pp. 1887–1936.
[45] D. Araújo, R. I. Oliveira, and D. Yukimura, "A mean-field limit for certain deep neural networks," 2019, *arXiv:1906.00193*.
[46] F. Santambrogio, "{Euclidean, metric, and Wasserstein} gradient flows: An overview," 2016, *arXiv:1609.03890*.
[47] C. Fang, H. Dong, and T. Zhang, "Over parameterized two-level neural networks can learn near optimal feature representations," Oct. 2019, *arXiv:1910.11508*.
[48] H. Mhaskar, Q. Liao, and T. Poggio, "When and why are deep networks better than shallow ones?" in *Proc. 31st AAAI Conf. Artif. Intell.*, vol. 31, no. 1, 2017.
[49] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[50] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. Int. Conf. Learn. Represent.*, 2017.
[51] O. Vinyals *et al.*, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3630–3638.
[52] M. J. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*, vol. 48. Cambridge, U.K.: Cambridge Univ. Press, 2019.
[53] R. Ibragimov and S. Sharakhmetov, "Analogues of Khintchine, Marcinkiewicz–Zygmund and Rosenthal inequalities for symmetric statistics," *Scandin. J. Statist.*, vol. 26, no. 4, pp. 621–633, Dec. 1999.

Cong Fang received the Ph.D. degree from Peking University in 2019. He was a Post-Doctoral Researcher with Princeton University in 2020 and the University of Pennsylvania in 2021. He is an Assistant Professor with Peking University, Beijing, China. His research interests include machine learning and optimization.

Yihong Gu received the B.Eng. degree in computer science from Tsinghua University in 2019. He is currently pursuing the Ph.D. degree in operation research and financial engineering with Princeton University. His research interests span statistics and machine learning.

Weizhong Zhang received the B.S. and Ph.D. degrees from Zhejiang University in 2012 and 2017, respectively. He is a Research Assistant Professor with the Department of Mathematics, The Hong Kong University of Science and Technology. His research interests include sparse neural networks training, robustness, and out-of-distribution generalization.

Tong Zhang (Fellow, IEEE) received the B.S. degree from Cornell University and the Ph.D. degree from Stanford University. He is a Chair Professor with the Mathematics and Computer Science Departments, The Hong Kong University of Science and Technology. His research interests include statistics, machine learning, optimization, and their applications.