

A Local Geometric Interpretation of Feature Extraction in Deep Feedforward Neural Networks

Md Kamran Chowdhury Shisher, Tasmeen Zaman Ornee, and Yin Sun
 Department of Electrical and Computer Engineering
 Auburn University, Auburn, AL, 36849
 {mzs0153, tzo0017, yzs0078}@auburn.edu

Abstract—In this paper, we present a local geometric analysis to interpret how deep feedforward neural networks extract low-dimensional features from high-dimensional data. Our study shows that, in a local geometric region, the optimal weight in one layer of the neural network and the optimal feature generated by the previous layer comprise a low-rank approximation of a matrix that is determined by the Bayes action of this layer. This result holds (i) for analyzing both the output layer and the hidden layers of the neural network, and (ii) for neuron activation functions with non-vanishing gradients. We use two supervised learning problems to illustrate our results: neural network based maximum likelihood classification (i.e., softmax regression) and neural network based minimum mean square estimation. Experimental validation of these theoretical results will be conducted in our future work.

I. INTRODUCTION

In recent years, neural network based supervised learning has been extensively admired due to its emerging applications in a wide range of inference problems, such as image classification, DNA sequencing, natural language processing, etc. The success of deep neural networks depends heavily on its capability of extracting good low-dimensional features from high-dimensional data. Due to the complexity of deep neural networks, theoretical interpretation of feature extraction in deep neural networks has been challenging, with some recent progress reported in, e.g., [1]–[13].

In this paper, we analyze the training of deep feedforward neural networks for a class of empirical risk minimization (ERM) based supervised learning algorithms. A local geometric analysis is conducted for feature extraction in deep feedforward neural networks. Specifically, the technical contributions of this paper are summarized as follows:

- We first analyze the design of (i) the weights and biases in the output layer and (ii) the feature constructed by the last hidden layer. In a local geometric region, this design problem is converted to a low-rank matrix approximation problem, where the matrix is characterized by the Bayes action of the supervised learning problem. Optimal designs of the weights, biases, and feature are derived in the local geometric region (see Theorems 1-3).
- The above local geometric analysis can be readily applied to a hidden layer (see Corollaries 2-4), by considering another supervised learning problem for the hidden layer.

This work was supported in part by the NSF grant CCF-1813078 and the ARO grant W911NF-21-1-0244.

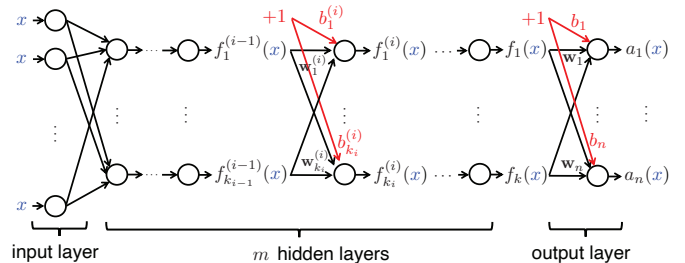


Fig. 1: A deep feedforward neural network.

The local geometric analyses of different layers are related to each other in an iterative manner: The optimal feature obtained from the analysis of one layer is the Bayes action needed for analyzing the previous layer. We use two supervised learning problems to illustrate our results.

A. Related Work

Due to the practical success of deep neural networks, there have been numerous efforts [1]–[13] to explain the feature extraction procedure of deep neural networks. Towards this end, researchers have used different approaches, for example, statistical learning theory approach [1], [2], information geometric approach [3]–[10], information theoretic approach [10]–[12], etc. The information bottleneck formulation in [11] suggested that the role of the deep neural network is to learn minimal sufficient statistics of the data for an inference task. The authors in [12] proposed that maximal coding rate reduction is a fundamental principle in deep neural networks. In [10], the authors formulated the problem of feature extraction by using KL-divergence, and provided a local geometric analysis by considering a weak dependency between the data and the label. Motivated by [10], we also consider the weak dependency. Compared to [10], our local geometric analysis can handle more general supervised learning problems and neuron activation functions, as explained in Section III.

II. MODEL AND PROBLEM

A. Deep Feedforward Neural Network Model

Consider the deep feedforward neural network illustrated in Figure 1, which consists of one input layer, m hidden layers,

and one output layer. The input layer admits an input variable $x \in \mathcal{X}$ and feeds a vector

$$\mathbf{f}^{(0)}(x) = [h_1^{(0)}(x), \dots, h_{k_0}^{(0)}(x)]^T \in \mathbb{R}^{k_0} \quad (1)$$

to the first hidden layer, where k_0 is the number of neurons in the input layer and $h_j^{(0)} : \mathcal{X} \mapsto \mathbb{R}$ is the activation function of the j -th neuron in the input layer. For all $i = 1, \dots, m$, the i -th hidden layer admits $\mathbf{f}^{(i-1)}(x) \in \mathbb{R}^{k_{i-1}}$ from the previous layer and constructs a vector $\mathbf{f}^{(i)}(x) \in \mathbb{R}^{k_i}$, usually called a *feature*, given by

$$\begin{aligned} & \mathbf{f}^{(i)}(x) \\ &= \left[h^{(i)}(\mathbf{w}_1^{(i)T} \mathbf{f}^{(i-1)}(x) + b_1^{(i)}), \dots, h^{(i)}(\mathbf{w}_{k_i}^{(i)T} \mathbf{f}^{(i-1)}(x) + b_{k_i}^{(i)}) \right]^T, \end{aligned} \quad (2)$$

where k_i is the number of neurons in the i -th hidden layer, $h^{(i)} : \mathbb{R} \mapsto \mathbb{R}$ is the activation function of each neuron in the i -th hidden layer, $\mathbf{w}_j^{(i)} \in \mathbb{R}^{k_{i-1}}$ and $b_j^{(i)} \in \mathbb{R}$ are the weight vector and bias of the j -th neuron in the i -th hidden layer, respectively. Denote $\mathbf{W}^{(i)} = [\mathbf{w}_1^{(i)}, \dots, \mathbf{w}_{k_i}^{(i)}]$ and $\mathbf{b}^{(i)} = [b_1^{(i)}, \dots, b_{k_i}^{(i)}]^T$, then (2) can be expressed compactly as

$$\mathbf{f}^{(i)}(x) = \mathbf{h}^{(i)} \left(\mathbf{W}^{(i)T} \mathbf{f}^{(i-1)}(x) + \mathbf{b}^{(i)} \right), \quad (3)$$

where $\mathbf{h}^{(i)} : \mathbb{R}^{k_i} \mapsto \mathbb{R}^{k_i}$ is a vector-valued function determined by (2). For notational simplicity, let us denote $k = k_m$ and $\mathbf{f}(x) = \mathbf{f}^{(m)}(x)$. The output layer admits $\mathbf{f}(x) \in \mathbb{R}^k$ from the last hidden layer and generates an output vector $\mathbf{a}(x) \in \mathcal{H}^n$, called an *action*, which is determined by

$$\begin{aligned} \mathbf{a}(x) &= \mathbf{h}(\mathbf{W}^T \mathbf{f}(x) + \mathbf{b}) \\ &= [h(\mathbf{w}_1^T \mathbf{f}(x) + b_1), \dots, h(\mathbf{w}_n^T \mathbf{f}(x) + b_n)]^T, \end{aligned} \quad (4)$$

where n is the number of neurons in the output layer, $h : \mathbb{R} \mapsto \mathcal{H}$ is the activation function of each neuron in the output layer, \mathcal{H} is the image set of h with $\mathcal{H} \subseteq \mathbb{R}$, $\mathbf{w}_j \in \mathbb{R}^k$ and $b_j \in \mathbb{R}$ are the weight vector and bias of the j -th neuron in the output layer, respectively, $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_n]$ and $\mathbf{b} = [b_1, \dots, b_n]^T$.

B. Neural Network based Supervised Learning Problem

The above deep feedforward neural network is used to solve a supervised learning problem. We focus on a class of popular supervised learning algorithms called *empirical risk minimization (ERM)*. In ERM algorithms, the weights and biases of the neural network are trained to construct a vector-valued function $\mathbf{a} : \mathcal{X} \mapsto \mathcal{A}$ that outputs an action $\mathbf{a}(x) \in \mathcal{A}$ for each $x \in \mathcal{X}$, where $\mathcal{A} \subseteq \mathcal{H}^n \subseteq \mathbb{R}^n$. Consider two random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, where \mathcal{X} and \mathcal{Y} are finite sets. The performance of an ERM algorithm is measured by a loss function $L : \mathcal{Y} \times \mathcal{A} \mapsto \mathbb{R}$, where $L(y, \mathbf{a}(x))$ is the incurred loss if action $\mathbf{a}(x)$ is generated by the neural network when $Y = y$. For example, in neural network based maximum likelihood classification, also known as *softmax regression*, the loss function is

$$L_{\log}(y, \mathbf{a}) = -\log \left(\frac{a_y}{\sum_{y' \in \mathcal{Y}} a_{y'}} \right), \quad (5)$$

which is the negative log-likelihood of a distribution Q_Y generated by the neural network, where $Q_Y(y) = a_y / \sum_{y' \in \mathcal{Y}} a_{y'}$, $a_y > 0$ for all $y \in \mathcal{Y}$, and the dimension of \mathbf{a} is $n = |\mathcal{Y}|$. In neural network based minimum mean-square estimation, the loss function is one half of the mean-square error between $\mathbf{y} \in \mathbb{R}^n$ and an estimate $\hat{\mathbf{y}} = \mathbf{a}(x) \in \mathbb{R}^n$ constructed by the neural network, i.e.,

$$L_2(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2} \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2. \quad (6)$$

Let $P_{X,Y}$ be the empirical joint distribution of X and Y in the training data, P_X and P_Y be the associated marginal distributions, which satisfies $P_X(x) > 0$ for all $x \in \mathcal{X}$ and $P_Y(y) > 0$ for all $y \in \mathcal{Y}$. The objective of ERM algorithms is to solve the following neural network training problem:

$$\min_{(\mathbf{W}, \mathbf{b}), (\mathbf{w}^{(i)}, \mathbf{b}^{(i)}), i=1, \dots, m} \mathbb{E}_{X,Y \sim P_{X,Y}} [L(Y, \mathbf{a}(X))], \quad (7)$$

where $\mathbf{a}(x)$ is subject to (1)-(4), because $\mathbf{a}(x)$ is the action generated by the neural network.

C. Problem Reformulation

Denote $\Phi = \{f : \mathcal{X} \mapsto \mathcal{A}\}$ as the set of all functions from \mathcal{X} to \mathcal{A} . Any action function $\mathbf{a}(x)$ produced by the neural network, i.e., any function satisfying (1)-(4), belongs to Φ , whereas some functions in Φ cannot be constructed by the neural network. By relaxing the set of feasible action functions in (7) as Φ , we derive the following lower bound of (7):

$$\begin{aligned} & \min_{\mathbf{a} \in \Phi} \mathbb{E}_{X,Y \sim P_{X,Y}} [L(Y, \mathbf{a}(X))] \\ &= \sum_{x \in \mathcal{X}} P_X(x) \min_{\mathbf{a}(x) \in \mathcal{A}} \mathbb{E}_{Y \sim P_{Y|X=x}} [L(Y, \mathbf{a}(x))], \end{aligned} \quad (8)$$

where (8) is decomposed into a sequence of separable optimization problems in (9), each optimizing the action $\mathbf{a}(x) \in \mathcal{A}$ for a given $x \in \mathcal{X}$. Let $\mathcal{A}_{P_Y} \subseteq \mathcal{A}$ denote the set of optimal solutions to the following problem:

$$\mathcal{A}_{P_Y} = \arg \min_{\mathbf{a} \in \mathcal{A}} \mathbb{E}_{Y \sim P_Y} [L(Y, \mathbf{a})] \quad (10)$$

and use \mathbf{a}_{P_Y} to denote an element of \mathcal{A}_{P_Y} , which is usually called a *Bayes action*. Define the discrepancy

$$D_L(\mathbf{a}_{P_Y} || \mathbf{a}) = \mathbb{E}_{Y \sim P_Y} [L(Y, \mathbf{a})] - \mathbb{E}_{Y \sim P_Y} [L(Y, \mathbf{a}_{P_Y})]. \quad (11)$$

According to (10) and (11), $D_L(\mathbf{a}_{P_Y} || \mathbf{a}) \geq 0$ for all $\mathbf{a} \in \mathcal{A}$, where equality is achieved if and only if $\mathbf{a} \in \mathcal{A}_{P_Y}$. When $\mathbf{a} = \mathbf{a}_{Q_Y}$, $D_L(\mathbf{a}_{P_Y} || \mathbf{a}_{Q_Y})$ is a generalized divergence between P_Y and Q_Y [14], [15].

By subtracting the lower bound (9) from (7), we obtain the following problem that is equivalent to (7):

$$\min_{(\mathbf{W}, \mathbf{b}), (\mathbf{w}^{(i)}, \mathbf{b}^{(i)}), i=1, \dots, m} \sum_{x \in \mathcal{X}} P_X(x) D_L(\mathbf{a}_{P_Y|X=x} || \mathbf{a}(x)), \quad (12)$$

where $\mathbf{a}_{P_Y|X=x} \in \mathcal{A}_{P_Y|X=x}$ is a Bayes action associated with the conditional empirical distribution $P_{Y|X=x}$ and $\mathbf{a}(x)$ is subject to (1)-(4).

III. MAIN RESULTS: FEATURE EXTRACTION IN DEEP FEEDFORWARD NEURAL NETWORKS

A. Local Geometric Analysis of the Output Layer

We consider the following reformulation of (12) that focuses on the training of the output layer:

$$\min_{\substack{\mathbf{W} \in \mathbb{R}^{k \times n} \\ \mathbf{b} \in \mathbb{R}^n, \mathbf{f} \in \Lambda}} \sum_{x \in \mathcal{X}} P_X(x) D_L(\mathbf{a}_{P_Y|X=x} \| \mathbf{h}(\mathbf{W}^T \mathbf{f}(x) + \mathbf{b})), \quad (13)$$

where Λ is the set of feature functions created by the input and hidden layers of the neural network.

Recall that \mathcal{H}^n is the image set of the vector-valued activation function $\mathbf{h}(\mathbf{b})$ of the output layer. Because $\mathcal{A}_{P_Y} \subseteq \mathcal{A} \subseteq \mathcal{H}^n$, for any Bayes action $\mathbf{a}_{P_Y} \in \mathcal{A}_{P_Y}$ that solves (10), there exists a bias $\tilde{\mathbf{b}} = [\tilde{b}_1, \dots, \tilde{b}_n]^T \in \mathbb{R}^n$ such that

$$\mathbf{h}(\tilde{\mathbf{b}}) = [h(\tilde{b}_1), \dots, h(\tilde{b}_n)]^T = \mathbf{a}_{P_Y}. \quad (14)$$

The following assumption is needed in our study.

Assumption 1. For each $i = 1, \dots, n$, there exist $\delta > 0$ and $K > 0$ such that for all $z \in (b_i - \delta, \tilde{b}_i + \delta)$, the activation function h satisfies

$$\left| h(z) - h(\tilde{b}_i) \right| \geq K \left| z - \tilde{b}_i \right|. \quad (15)$$

Lemma 1. If h is strictly increasing and continuously differentiable, then h satisfies Assumption 1.

Due to space limitation, all the proofs are relegated to our technical report [21]. It is easy to see that the leaky ReLU activation function [16, pp. 187-188] satisfies Assumption 1. In addition, the hyperbolic tangent function and the sigmoid function [16, p. 189] also satisfy Assumption 1, because they are strictly increasing and continuously differentiable.

Let $\mathcal{P}^{\mathcal{Y}}$ be the set of all probability distributions on \mathcal{Y} and $\text{relint}(\mathcal{P}^{\mathcal{Y}})$ be the relative interior of the set $\mathcal{P}^{\mathcal{Y}}$.

Assumption 2. If two distributions $P_Y, Q_Y \in \text{relint}(\mathcal{P}^{\mathcal{Y}})$ are close to each other such that

$$\sum_{y \in \mathcal{Y}} (P_Y(y) - Q_Y(y))^2 \leq \gamma^2, \quad (16)$$

then for any $\mathbf{a}_{P_Y} \in \mathcal{A}_{P_Y}$, there exists an $\mathbf{a}_{Q_Y} \in \mathcal{A}_{Q_Y}$ such that

$$\|\mathbf{a}_{P_Y} - \mathbf{a}_{Q_Y}\|_2 = O(\gamma). \quad (17)$$

Assumption 2 characterizes the differentiability of the Bayes action \mathbf{a}_{P_Y} with respect to P_Y . The loss functions in (5) and (6) satisfy Assumption 2, as explained later in Section III-C.

Because of the universal function approximation properties of deep feedforward neural networks [16]–[18], we make the following assumption.

Assumption 3. For given $\epsilon > 0$ and $\mathbf{a}_{P_Y|X=x} \in \mathcal{A}_{P_Y|X=x}$ with $x \in \mathcal{X}$, there exists an optimal solution $(\mathbf{f}, \mathbf{W}, \mathbf{b})$ to (13) such that for all $x \in \mathcal{X}$

$$\|\mathbf{a}_{P_Y|X=x} - \mathbf{h}(\mathbf{W}^T \mathbf{f}(x) + \mathbf{b})\|_2^2 \leq \epsilon^2. \quad (18)$$

By Assumption 3, the neural network can closely approximate the vector-valued function $x \mapsto \mathbf{a}_{P_Y|X=x}$.

Definition 1. For a given $\epsilon > 0$, two random variables X and Y are called ϵ -dependent, if the χ^2 -mutual information $I_{\chi^2}(X; Y)$ is no more than ϵ^2 , given by

$$I_{\chi^2}(X; Y) = D_{\chi^2}(P_{X,Y} \| P_X \otimes P_Y) \leq \epsilon^2, \quad (19)$$

where

$$D_{\chi^2}(P_X \| Q_X) = \int_{\mathcal{X}} \frac{(P(x) - Q(x))^2}{Q^2(x)} dQ(x) \quad (20)$$

is Neyman's χ^2 -divergence [19].

Motivated by the seminal work [10] and [20], we consider the following assumption.

Assumption 4. For a given $\epsilon > 0$, X and Y are ϵ -dependent.

By using the above assumptions, we can find a local geometric region (21) that is useful for our analysis.

Lemma 2. For a sufficiently small $\epsilon > 0$, if Assumptions 1-4 hold, then there exists an optimal solution $(\mathbf{f}, \mathbf{W}, \mathbf{b})$ to (13) such that for all $x \in \mathcal{X}$ and $i = 1, \dots, n$

$$\mathbf{w}_i^T \mathbf{f}(x) + b_i - \tilde{b}_i = O(\epsilon). \quad (21)$$

For any feature $\mathbf{f}(x) \in \mathbb{R}^k$, define a matrix $\Xi_{\mathbf{f}} \in \mathbb{R}^{k \times |\mathcal{X}|}$ as

$$\Xi_{\mathbf{f}} = [\xi_{\mathbf{f}}(1), \dots, \xi_{\mathbf{f}}(|\mathcal{X}|)], \quad (22)$$

where

$$\xi_{\mathbf{f}}(x) = \sqrt{P_X(x)} (\mathbf{f}(x) - \boldsymbol{\mu}_{\mathbf{f}}), \quad (23)$$

$$\boldsymbol{\mu}_{\mathbf{f}} = \sum_{x \in \mathcal{X}} P_X(x) \mathbf{f}(x). \quad (24)$$

In addition, define the following matrix $\mathbf{B} \in \mathbb{R}^{n \times |\mathcal{X}|}$ based on the Bayes actions $\mathbf{a}_{P_Y|X=x}$ for $x \in \mathcal{X}$:

$$\mathbf{B} = [\beta_Y(1), \dots, \beta_Y(|\mathcal{X}|)], \quad (25)$$

where

$$\beta_Y(x) = \sqrt{P_X(x)} (\mathbf{a}_{P_Y|X=x} - \boldsymbol{\mu}_{\mathbf{a}}), \quad (26)$$

$$\boldsymbol{\mu}_{\mathbf{a}} = \sum_{x \in \mathcal{X}} P_X(x) \mathbf{a}_{P_Y|X=x}. \quad (27)$$

Assumption 5. The function $\mathbf{a} \mapsto L(y, \mathbf{a})$ is twice continuously differentiable.

The Hessian matrix \mathbf{M}_L of the function $\mathbf{a} \mapsto \mathbb{E}_{Y \sim P_Y} [L(Y, \mathbf{a})]$ at the point $\mathbf{a} = \mathbf{a}_{P_Y}$ is

$$\mathbf{M}_L = \frac{\partial^2 \mathbb{E}_{Y \sim P_Y} [L(Y, \mathbf{a})]}{\partial \mathbf{a} \partial \mathbf{a}^T} \Big|_{\mathbf{a} = \mathbf{a}_{P_Y}}. \quad (28)$$

Because \mathbf{a}_{P_Y} is an optimal solution to (10), \mathbf{M}_L is positive semi-definite. Hence, it has a Cholesky decomposition $\mathbf{M}_L = \mathbf{R}_L^T \mathbf{R}_L$. The Jacobian matrix of $\mathbf{h}(\mathbf{b})$ at the point $\mathbf{b} = \tilde{\mathbf{b}}$ is

$$\mathbf{J} = \frac{\partial \mathbf{h}(\mathbf{b})}{\partial \mathbf{b}^T} \Big|_{\mathbf{b} = \tilde{\mathbf{b}}}. \quad (29)$$

Lemma 3. *If Assumptions 2, 4, and 5 are satisfied, then in the local analysis regime (21), the objective function in (13) can be expressed as*

$$\begin{aligned} & \sum_{x \in \mathcal{X}} P_X(x) D_L(\mathbf{a}_{P_Y|X=x} \|\mathbf{h}(\mathbf{W}^T \mathbf{f}(x) + \mathbf{b})) \\ &= \frac{1}{2} \|\tilde{\mathbf{B}} - \Xi_{\mathbf{W}} \Xi_{\mathbf{f}}\|_F^2 + \frac{1}{2} \eta(\mathbf{d}, \mathbf{f}) + o(\epsilon^2), \end{aligned} \quad (30)$$

where $\tilde{\mathbf{B}} = \mathbf{R}_L \mathbf{B}$,

$$\Xi_{\mathbf{W}} = \mathbf{R}_L \mathbf{J} \mathbf{W}^T, \quad (31)$$

$$\mathbf{d} = \mathbf{b} - \tilde{\mathbf{b}}, \quad (32)$$

$$\begin{aligned} \eta(\mathbf{d}, \mathbf{f}) &= (\mathbf{a}_{P_Y} - \boldsymbol{\mu}_{\mathbf{a}} + \mathbf{J} \mathbf{d} + \mathbf{J} \mathbf{W}^T \boldsymbol{\mu}_{\mathbf{f}})^T \mathbf{M}_L \\ &\quad \times (\mathbf{a}_{P_Y} - \boldsymbol{\mu}_{\mathbf{a}} + \mathbf{J} \mathbf{d} + \mathbf{J} \mathbf{W}^T \boldsymbol{\mu}_{\mathbf{f}}). \end{aligned} \quad (33)$$

In the local analysis regime, the training of $(\mathbf{f}, \mathbf{W}, \mathbf{b})$ in (13) can be expressed as the following optimization problem of $(\Xi_{\mathbf{W}}, \Xi_{\mathbf{f}}, \boldsymbol{\mu}_{\mathbf{f}}, \mathbf{d})$:

$$\min_{\Xi_{\mathbf{W}}, \Xi_{\mathbf{f}}, \boldsymbol{\mu}_{\mathbf{f}}, \mathbf{d}} \frac{1}{2} \|\tilde{\mathbf{B}} - \Xi_{\mathbf{W}} \Xi_{\mathbf{f}}\|_F^2 + \frac{1}{2} \eta(\mathbf{d}, \mathbf{f}). \quad (34)$$

When $(\Xi_{\mathbf{f}}, \boldsymbol{\mu}_{\mathbf{f}})$ are fixed, the optimal $(\Xi_{\mathbf{W}}^*, \mathbf{d}^*)$ are determined by

Theorem 1. *For fixed $\Xi_{\mathbf{f}}$ and $\boldsymbol{\mu}_{\mathbf{f}}$, the optimal $\Xi_{\mathbf{W}}^*$ to minimize (34) is given by*

$$\Xi_{\mathbf{W}}^* = \tilde{\mathbf{B}} \Xi_{\mathbf{f}}^T (\Xi_{\mathbf{f}} \Xi_{\mathbf{f}}^T)^{-1}, \quad (35)$$

and the optimal bias \mathbf{d}^* is expressed as

$$\mathbf{d}^* = -\mathbf{W}^T \boldsymbol{\mu}_{\mathbf{f}} + \mathbf{J}^{-1} (\boldsymbol{\mu}_{\mathbf{a}} - \mathbf{a}_{P_Y}). \quad (36)$$

By Theorem 1, the rows of $\Xi_{\mathbf{W}}^*$ are obtained by projecting the rows of $\tilde{\mathbf{B}}$ on the subspace spanned by the rows of $\Xi_{\mathbf{f}}$. The optimal bias \mathbf{d}^* cancels out the effects of the mean feature $\boldsymbol{\mu}_{\mathbf{f}}$ and the mean difference $\boldsymbol{\mu}_{\mathbf{a}} - \mathbf{a}_{P_Y}$ between the Bayes actions $\mathbf{a}_{P_Y|X=x}$ and \mathbf{a}_{P_Y} . The optimal weight \mathbf{W}^* and bias \mathbf{b}^* can be derived by using (31)-(32) and (35)-(36).

When $(\Xi_{\mathbf{W}}, \mathbf{d})$ are fixed and the hidden layers have sufficient expression power, the optimal $(\Xi_{\mathbf{f}}^*, \boldsymbol{\mu}_{\mathbf{f}}^*)$ are given by

Theorem 2. *For fixed $\Xi_{\mathbf{W}}$ and \mathbf{d} , the optimal $\Xi_{\mathbf{f}}^*$ to minimize (34) is given by*

$$\Xi_{\mathbf{f}}^* = (\Xi_{\mathbf{W}}^T \Xi_{\mathbf{W}})^{-1} \Xi_{\mathbf{W}}^T \tilde{\mathbf{B}}, \quad (37)$$

and the optimal mean $\boldsymbol{\mu}_{\mathbf{f}}^*$ is given by

$$\boldsymbol{\mu}_{\mathbf{f}}^* = -(\Xi_{\mathbf{W}}^T \Xi_{\mathbf{W}})^{-1} \Xi_{\mathbf{W}}^T (\mathbf{a}_{P_Y} - \boldsymbol{\mu}_{\mathbf{a}} + \mathbf{J} \mathbf{d}). \quad (38)$$

By Theorem 2, the columns of $\Xi_{\mathbf{f}}^*$ are obtained by projecting the columns of $\tilde{\mathbf{B}}$ on the subspace spanned by the columns of $\Xi_{\mathbf{W}}$. The optimal feature $\boldsymbol{\mu}_{\mathbf{f}}^*$ cancels out the effects of \mathbf{d} and $\boldsymbol{\mu}_{\mathbf{a}} - \mathbf{a}_{P_Y}$. The optimal feature $\mathbf{f}^*(x)$ can be derived by using (22)-(24) and (37)-(38).

The singular value decomposition of $\tilde{\mathbf{B}}$ can be written as

$$\tilde{\mathbf{B}} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T, \quad (39)$$

where $\boldsymbol{\Sigma} = \text{Diag}(\sigma_1, \dots, \sigma_K)$ is a diagonal matrix with $K = \min(n, |\mathcal{X}|)$ singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_K = 0$, \mathbf{U}

and \mathbf{V} are composed by the K leading left and right singular vectors of $\tilde{\mathbf{B}}$, respectively. Denote

$$\sqrt{\mathbf{p}_X} = [\sqrt{P_X(1)}, \dots, \sqrt{P_X(|\mathcal{X}|)}]^T. \quad (40)$$

Because $\tilde{\mathbf{B}} \sqrt{\mathbf{p}_X} = 0$ and $\|\sqrt{\mathbf{p}_X}\|_2 = 1$, $\sqrt{\mathbf{p}_X}$ is the right singular vector of $\tilde{\mathbf{B}}$ for the singular value $\sigma_K = 0$. When $(\Xi_{\mathbf{f}}, \boldsymbol{\mu}_{\mathbf{f}}, \Xi_{\mathbf{W}}, \mathbf{d})$ are all designable, the optimal solutions are characterized in the following theorem.

Theorem 3. *If $k \leq \min(n, |\mathcal{X}|)$, then any $(\Xi_{\mathbf{f}}^*, \Xi_{\mathbf{W}}^*)$ satisfying (41) jointly minimizes (34):*

$$\Xi_{\mathbf{W}}^* \Xi_{\mathbf{f}}^* = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T, \quad (41)$$

where $\boldsymbol{\Sigma}_k = \text{Diag}(\sigma_1, \dots, \sigma_k)$, $\mathbf{U}_k = [\mathbf{u}_1, \dots, \mathbf{u}_k]$, and $\mathbf{V}_k = [\mathbf{v}_1, \dots, \mathbf{v}_k]$. Moreover, any bias \mathbf{d}^* and mean $\boldsymbol{\mu}_{\mathbf{f}}^*$ satisfying (42) jointly minimizes (34):

$$\mathbf{J}(\mathbf{d}^* + \mathbf{W}^T \boldsymbol{\mu}_{\mathbf{f}}^*) = \boldsymbol{\mu}_{\mathbf{a}} - \mathbf{a}_{P_Y}. \quad (42)$$

According to Theorem 3, the optimal $(\Xi_{\mathbf{f}}^*, \Xi_{\mathbf{W}}^*)$ are given by the low-rank approximation of $\tilde{\mathbf{B}}$, which can be derived by using the power iteration algorithm [22], or equivalently, by executing (35) and (37) iteratively. The optimal $(\mathbf{d}^*, \boldsymbol{\mu}_{\mathbf{f}}^*)$ cancel out the effect of $\boldsymbol{\mu}_{\mathbf{a}} - \mathbf{a}_{P_Y}$.

The optimal $\Xi_{\mathbf{f}}^*$ in Theorems 2-3 can be achieved only when the hidden layers have sufficient expression power. Nonetheless, $\Xi_{\mathbf{f}}^*$ plays an important role in the analysis of the hidden layers, as explained in the next subsection.

B. Local Geometric Analysis of Hidden Layers

Next, we provide a local geometric analysis for each hidden layer. To that end, let us consider the training of the i -th hidden layer for fixed weights and biases in the subsequent layers. Define a loss function $L^{(i)}$ for the i -th hidden layer

$$L^{(i)}(y, \mathbf{a}^{(i)}) = L\left(y, \mathbf{g} \circ \mathbf{g}^{(m)} \circ \dots \circ \mathbf{g}^{(i+1)}(\mathbf{a}^{(i)})\right), \quad (43)$$

where for $k = i, \dots, m-1$

$$\mathbf{g}^{(k+1)}(\mathbf{a}^{(k)}) = \mathbf{h}^{(k+1)}(\mathbf{W}^{(k+1)T} \mathbf{a}^{(k)} + \mathbf{b}^{(k+1)}), \quad (44)$$

$$\mathbf{g}(\mathbf{a}^{(m)}) = \mathbf{h}(\mathbf{W}^T \mathbf{a}^{(m)} + \mathbf{b}). \quad (45)$$

Given $(\mathbf{W}^{(k)}, \mathbf{b}^{(k)})$ for $k = i+1, \dots, m$ and (\mathbf{W}, \mathbf{b}) , the training problem of the i -th hidden layer is formulated as

$$\min_{\substack{\mathbf{W}^{(i)}, \\ \mathbf{b}^{(i)}, \\ \mathbf{f}^{(i-1)} \in \Lambda^{i-1}}} \sum_{x \in \mathcal{X}} P_X(x) D_{L^{(i)}}(\mathbf{a}_{P_Y|X=x}^{(i)} \|\mathbf{h}^{(i)}(\mathbf{W}^{(i)T} \mathbf{f}^{(i-1)}(x) + \mathbf{b}^{(i)}\|), \quad (46)$$

where Λ^{i-1} is the set of all feature functions that can be created by the first $(i-1)$ hidden layers. We adopt several assumptions for the i -th hidden layer that are similar to Assumptions 1-5. Let $\mathbf{a}_{P_Y}^{(i)}$ denote the Bayes action associated to the loss function $L^{(i)}$ and distribution P_Y . According to Lemma 2, there exists a bias $\tilde{\mathbf{b}}^{(i)}$ and a tuple $(\mathbf{f}^{(i-1)}, \mathbf{W}^{(i)}, \mathbf{b}^{(i)})$ such that (i) $\mathbf{h}^{(i)}(\tilde{\mathbf{b}}^{(i)}) = \mathbf{a}_{P_Y}^{(i)}$ is a Bayes action associated to the loss function $L^{(i)}$ and distribution P_Y , (ii) $(\mathbf{f}^{(i-1)}, \mathbf{W}^{(i)}, \mathbf{b}^{(i)})$

is an optimal solution to (46), and (iii) for all $x \in \mathcal{X}$ and $j = 1, \dots, k_i$

$$\mathbf{w}_j^{(i)\top} \mathbf{f}^{(i-1)}(x) + b_j^{(i)} - \tilde{b}_j^{(i)} = O(\epsilon). \quad (47)$$

Similar to (22)-(29), we define $\Xi_{\mathbf{f}^{(i)}}$, $\mathbf{B}^{(i)}$, $\boldsymbol{\mu}_{\mathbf{f}^{(i)}}$, $\boldsymbol{\mu}_{\mathbf{a}^{(i)}}$, $\mathbf{M}_{L^{(i)}}$, $\mathbf{R}_{L^{(i)}}$, and $\mathbf{J}^{(i)}$ for the i -th hidden layer. Due to space limitations, these definitions are relegated to our technical report [21]. The following result is an immediate corollary of Lemma 3.

Corollary 1. *In the local analysis regime (47), the objective function in (46) can be expressed as*

$$\begin{aligned} & \sum_{x \in \mathcal{X}} P_X(x) D_{L^{(i)}}(\mathbf{a}_{P_Y|X=x}^{(i)} \|\mathbf{h}^{(i)}(\mathbf{W}^{(i)\top} \mathbf{f}^{(i-1)}(x) + \mathbf{b}^{(i)})\| \\ &= \frac{1}{2} \|\tilde{\mathbf{B}}^{(i)} - \Xi_{\mathbf{W}^{(i)}} \Xi_{\mathbf{f}^{(i-1)}}\|_F^2 + \frac{1}{2} \eta(\mathbf{d}^{(i)}, \mathbf{f}^{(i-1)}) + o(\epsilon^2), \end{aligned} \quad (48)$$

where $\tilde{\mathbf{B}}^{(i)} = \mathbf{R}_{L^{(i)}} \mathbf{B}^{(i)}$, $\Xi_{\mathbf{W}^{(i)}} = \mathbf{R}_{L^{(i)}} \mathbf{J}^{(i)} \mathbf{W}^{(i)\top}$, $\mathbf{d}^{(i)} = \mathbf{b}^{(i)} - \tilde{\mathbf{b}}^{(i)}$, and

$$\begin{aligned} & \eta(\mathbf{d}^{(i)}, \mathbf{f}^{(i)}) \\ &= (\mathbf{a}_{P_Y}^{(i)} - \boldsymbol{\mu}_{\mathbf{a}^{(i)}} + \mathbf{J}^{(i)} \mathbf{d}^{(i)} + \mathbf{J}^{(i)} \mathbf{W}^{(i)\top} \boldsymbol{\mu}_{\mathbf{f}^{(i-1)}})^{\top} \mathbf{M}_{L^{(i)}} \\ & \quad \times (\mathbf{a}_{P_Y}^{(i)} - \boldsymbol{\mu}_{\mathbf{a}^{(i)}} + \mathbf{J}^{(i)} \mathbf{d}^{(i)} + \mathbf{J}^{(i)} \mathbf{W}^{(i)\top} \boldsymbol{\mu}_{\mathbf{f}^{(i-1)}}). \end{aligned} \quad (49)$$

In the local analysis regime, the training of $(\Xi_{\mathbf{W}^{(i)}}$, $\Xi_{\mathbf{f}^{(i-1)}}$, $\mathbf{d}^{(i)}$, $\boldsymbol{\mu}_{\mathbf{f}^{(i-1)}}$) in (46) can be expressed as the following optimization problem:

$$\min_{\substack{\Xi_{\mathbf{W}^{(i)}}, \Xi_{\mathbf{f}^{(i-1)}} \\ \mathbf{d}^{(i)}, \boldsymbol{\mu}_{\mathbf{f}^{(i-1)}}}} \frac{1}{2} \|\tilde{\mathbf{B}}^{(i)} - \Xi_{\mathbf{W}^{(i)}} \Xi_{\mathbf{f}^{(i-1)}}\|_F^2 + \frac{1}{2} \eta(\mathbf{d}^{(i)}, \mathbf{f}^{(i-1)}). \quad (50)$$

Similar to Theorems 1-3, we can get

Corollary 2. *For fixed $\Xi_{\mathbf{f}^{(i-1)}}$ and $\boldsymbol{\mu}_{\mathbf{f}^{(i-1)}}$, the optimal $\Xi_{\mathbf{W}^{(i)}}$ to minimize (50) is given by*

$$\Xi_{\mathbf{W}^{(i)}}^* = \tilde{\mathbf{B}}^{(i)} \Xi_{\mathbf{f}^{(i-1)}}^{(i-1)\top} (\Xi_{\mathbf{f}^{(i-1)}} \Xi_{\mathbf{f}^{(i-1)}}^{\top})^{-1}, \quad (51)$$

and the optimal bias $\mathbf{d}^{(i)*}$ is expressed as

$$\mathbf{d}^{(i)*} = -\bar{\mathbf{W}}^{(i)\top} \boldsymbol{\mu}_{\mathbf{f}^{(i-1)}} + (\mathbf{J}^{(i)})^{-1} (\boldsymbol{\mu}_{\mathbf{a}^{(i)}} - \mathbf{a}_{P_Y}^{(i)}). \quad (52)$$

Corollary 3. *For fixed $\Xi_{\mathbf{W}^{(i)}}$ and $\mathbf{d}^{(i)}$, the optimal $\Xi_{\mathbf{f}^{(i-1)}}$ to minimize (50) is given by*

$$\Xi_{\mathbf{f}^{(i-1)}}^* = (\Xi_{\mathbf{W}^{(i)}}^{\top} \Xi_{\mathbf{W}^{(i)}})^{-1} \Xi_{\mathbf{W}^{(i)}}^{\top} \tilde{\mathbf{B}}^{(i)}, \quad (53)$$

and the optimal mean $\boldsymbol{\mu}_{\mathbf{f}^*}$ is given by

$$\boldsymbol{\mu}_{\mathbf{f}^{(i-1)}}^* = -(\Xi_{\mathbf{W}^{(i)}}^{\top} \Xi_{\mathbf{W}^{(i)}})^{-1} \Xi_{\mathbf{W}^{(i)}}^{\top} (\mathbf{a}_{P_Y}^{(i)} - \boldsymbol{\mu}_{\mathbf{a}^{(i)}} + \mathbf{J}^{(i)} \mathbf{d}^{(i)}). \quad (54)$$

Corollary 4. *If $k_{i-1} \leq \min(k_i, |\mathcal{X}|)$, then any $(\Xi_{\mathbf{f}^{(i)}}^*, \Xi_{\mathbf{W}^{(i)}}^*)$ satisfying (55) jointly minimizes (50):*

$$\Xi_{\mathbf{W}^{(i)}}^* \Xi_{\mathbf{f}^{(i)}}^* = \mathbf{U}_{k_{i-1}}^{(i)} \boldsymbol{\Sigma}_{k_{i-1}}^{(i)} \mathbf{V}_{k_{i-1}}^{(i)\top}, \quad (55)$$

where $\boldsymbol{\Sigma}_{k_{i-1}}^{(i)} = \text{Diag}(\sigma_1^{(i)}, \dots, \sigma_{k_{i-1}}^{(i)})$ is a diagonal matrix associated with k_{i-1} leading singular values of $\tilde{\mathbf{B}}^{(i)}$, $\mathbf{U}_{k_{i-1}}^{(i)}$

and $\mathbf{V}_{k_{i-1}}^{(i)}$ are composed by the corresponding left and right singular vectors of $\tilde{\mathbf{B}}^{(i)}$, respectively. Moreover, any bias $\mathbf{d}^{(i)*}$ and mean $\boldsymbol{\mu}_{\mathbf{f}^{(i)}}^*$ satisfying (56) jointly minimizes (50):

$$\mathbf{J}^{(i)} \left(\mathbf{d}^{(i)*} + \mathbf{W}^{(i)\top} \boldsymbol{\mu}_{\mathbf{f}^{(i)}}^* \right) = \boldsymbol{\mu}_{\mathbf{a}^{(i)}} - \mathbf{a}_{P_Y}^{(i)}. \quad (56)$$

Compared to the local geometric analysis for softmax regression in [10], Theorems 1-3 and Corollaries 2-4 could handle more general loss functions and activation functions. In addition, our results can be applied to multi-layer neural networks in the following iterative manner: For fixed (\mathbf{W}, \mathbf{b}) in the output layer, the Bayes action $\mathbf{a}_{P_Y|X=x}^{(m)}$ needed for analyzing the m -th hidden layer is the optimal feature $\mathbf{f}^*(x)$ provided by Theorem 2. Similar results hold for the i -th hidden layer. For fixed weights and biases in subsequent layers, the Bayes action $\mathbf{a}_{P_Y|X=x}^{(i-1)}$ needed for analyzing the $(i-1)$ -th hidden layer is the optimal feature $\mathbf{f}^{(i-1)*}(x)$ in Corollary 3. Hence, the optimal features obtained in Theorem 2 and Corollary 3 are useful for the local geometric analysis of earlier layers.

C. Two Examples

1) *Neural Network based Maximum Likelihood Classification (Softmax Regression):* The Bayes actions \mathbf{a}_{P_Y} associated to the loss function (5) are non-unique. The set of all Bayes actions is $\mathcal{A}_{P_Y} = \{\alpha P_Y : \alpha > 0\}$, which satisfies Assumption 2. By choosing one Bayes action $\mathbf{a}_{P_Y} = P_Y$, one can derive the matrices \mathbf{M}_L and \mathbf{B} used in Theorems 1-3: The (y, y') -th element of \mathbf{M}_L is

$$(\mathbf{M}_L)_{y, y'} = \frac{\delta(y, y')}{P_Y(y)} - 1, \quad (57)$$

where $\delta(y, y') = 1$, if $y = y'$; and $\delta(y, y') = 0$, if $y \neq y'$. The (y, x) -th element of \mathbf{B} is

$$(\mathbf{B})_{y, x} = \sqrt{P_X(x)} (P_Y|X=x(y|x) - P_Y(y)). \quad (58)$$

To make our analysis applicable to the softmax activation function [16, Eq. (6.29)], we have used a loss function (5) that is different from the log-loss function in [10], [14]. As a result, our local geometric analysis with (57) and (58) is different from the results in [10].

2) *Neural Network based Minimum Mean-square Estimation:* Consider the minimum mean-square estimation of a random vector $\mathbf{Y} = [Y_1, \dots, Y_n]^{\top}$. The Bayes action associated to the loss function (6) is $\mathbf{a}_{P_Y} = \mathbb{E}[\mathbf{Y}]$, which satisfies Assumption 2 because $\mathbb{E}[\mathbf{Y}]$ is a linear function of P_Y . One can show that $\mathbf{M}_L = \mathbf{I}$ is an identity matrix and the (j, x) -th element of \mathbf{B} is

$$(\mathbf{B})_{j, x} = \sqrt{P_X(x)} (\mathbb{E}[Y_j|X=x] - \mathbb{E}[Y_j]). \quad (59)$$

IV. CONCLUSION

In this paper, we have analyzed feature extraction in deep feedforward neural networks in a local region. We will conduct experiments to verify these results in our future work.

REFERENCES

- [1] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian, “Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks,” *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 2285–2301, 2019.
- [2] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.
- [3] R. Karakida, S. Akaho, and S.-i. Amari, “Universal statistics of fisher information in deep neural networks: Mean field approach,” pp. 1032–1041, 2019.
- [4] S. Mei, A. Montanari, and P.-M. Nguyen, “A mean field view of the landscape of two-layer neural networks,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 33, pp. E7665–E7671, 2018.
- [5] S. Goldt, M. Mézard, F. Krzakala, and L. Zdeborová, “Modeling the influence of data structure on learning in neural networks: The hidden manifold model,” *Physical Review X*, vol. 10, no. 4, p. 041044, 2020.
- [6] A. Jacot, F. Gabriel, and C. Hongler, “Neural tangent kernel: Convergence and generalization in neural networks,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [7] N. Lei, D. An, Y. Guo, K. Su, S. Liu, Z. Luo, S.-T. Yau, and X. Gu, “A geometric understanding of deep learning,” *Engineering*, vol. 6, no. 3, pp. 361–374, 2020.
- [8] M. Geiger, L. Petrini, and M. Wyart, “Landscape and training regimes in deep learning,” *Physics Reports*, vol. 924, pp. 1–18, 2021.
- [9] K. N. Quinn, M. C. Abbott, M. K. Transtrum, B. B. Machta, and J. P. Sethna, “Information geometry for multiparameter models: New perspectives on the origin of simplicity,” 2021, arXiv:2111.07176.
- [10] X. Xu, S.-L. Huang, L. Zheng, and G. W. Wornell, “An information theoretic interpretation to deep neural networks,” *Entropy*, vol. 24, p. 135, Jan. 2022.
- [11] N. Tishby and N. Zaslavsky, “Deep learning and the information bottleneck principle,” *IEEE Information Theory Workshop (ITW)*, pp. 1–5, 2015.
- [12] Y. Yu, K. H. R. Chan, C. You, C. Song, and Y. Ma, “Learning diverse and discriminative representations via the principle of maximal coding rate reduction,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9422–9434, 2020.
- [13] S. Arora, S. Du, W. Hu, Z. Li, and R. Wang, “Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks,” vol. 97, pp. 322–332, Jun 2019.
- [14] F. Farnia and D. Tse, “A minimax approach to supervised learning,” *Advances in Neural Information Processing Systems*, vol. 29, pp. 4240–4248, 2016.
- [15] P. D. Grünwald and A. P. Dawid, “Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory,” *the Annals of Statistics*, vol. 32, no. 4, pp. 1367–1433, 2004.
- [16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [17] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [18] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [19] Y. Polyanskiy and Y. Wu, “Lecture notes on information theory,” *Lecture Notes for MIT (6.441), UIUC (ECE 563), Yale (STAT 664)*, no. 2012-2017, 2014.
- [20] S.-L. Huang, A. Makur, G. W. Wornell, and L. Zheng, “On universal features for high-dimensional learning and inference,” 2019, arXiv:1911.09105.
- [21] M. K. C. Shisher, T. Z. Ornee, and Y. Sun, “A local geometric interpretation of feature extraction in deep feedforward neural networks,” *Technical Report*, 2022, <http://webhome.auburn.edu/~yzs0078/PaperISIT2022.pdf>.
- [22] R. Bulirsch, J. Stoer, and J. Stoer, *Introduction to Numerical Analysis*. Springer, 2002.