An Information Theoretic Interpretation to Deep Neural Networks

Shao-Lun Huang
DSIT Research Center
Tsinghua-Berkeley Shenzhen Institute
Shenzhen, China 518055
Email: shaolun.huang@sz.tsinghua.edu.cn

Xiangxiang Xu
Dept. of Electronic Engineering
Tsinghua University
Beijing, China 100084
Email: xuxx14@mails.tsinghua.edu.cn

Lizhong Zheng, Gregory W. Wornell
Dep. of Electrical & Computer Eng.
Massachusetts Institute of Technology
Cambridge, MA 02139-4307
Email: {lizhong, gww}@mit.edu

***Abstract*—It is commonly believed that the hidden layers of deep neural networks (DNNs) attempt to extract informative features for learning tasks. In this paper, we formalize this intuition by showing that the features extracted by DNN coincide with the result of an optimization problem, which we call the "universal feature selection" problem, in a local analysis regime. We interpret the weights training in DNN as the projection of feature functions between feature spaces, specified by the network structure. Our formulation has direct operational meaning in terms of the performance for inference tasks, and gives interpretations to the internal computation results of DNNs. Results of numerical experiments are provided to support the analysis.**

I. Introduction

Due to the striking performance of deep learning in various fields, deep neural networks (DNNs) have gained great attentions in modern computer science. While it is a common understanding that the features extracted from the hidden layers of DNN are "informative" for learning tasks, the mathematical meaning of informative features in DNN is generally not clear. There have been numerous research efforts towards this direction [1]. For instance, the information bottleneck [2] employs the mutual information as the metric to quantify the informativeness of features in DNN, and other information metrics, such as the Kullback-Leibler (K-L) divergence [3] and Weissenstein distance [4] are also used in different problems. However, because of the complicated structure of DNNs, there is a disconnection between these information metrics and the performance objectives of the inference tasks that DNNs want to solve. Therefore, it is in general difficult to match the DNN learning with the optimization of a particular information metric.

In this paper, our first contribution is to propose a learning framework, called universal feature selection, which connects the information metric of features and the performance evaluation of inference problems. Specifically for a pair of data variables X and Y, the goal of universal feature selection is to select features from X to infer about a targeted attribute V of Y, where V is only assumed with a rotationally uniform prior over the attribute space of Y, but the precise statistical model between V and X is unknown. Thus, the selected features have to be good for solving multiple inference problems, and should be generally "informative" about Y. We show that in a local analysis regime, the averaged performance of inferring V by a selected feature of X is measured via a linear projection of this feature, which leads to an information metric to features, and the optimal features can be computed from the singular value decomposition (SVD) of this linear projection.

More importantly, we show that in the local analysis regime, the optimal features selected in DNNs from log-loss optimization coincide with the solutions of universal feature selection. Therefore, the information metric developed in universal feature selection can be used to understand the operations in DNNs. As a result, we observe that the DNN weight updates in general can be interpreted as projecting features between the feature spaces of data and label for extracting the most correlated aspects between them, and the iterative projections can be viewed as computing the SVD of a linear projection between these feature spaces. Moreover, our results also give an explicit interpretation of the goal and the procedures of the BackProp/SGD operations in deep learning. Finally, the theoretic results are validated via numerical experiments.

Notations: Throughout this paper, we use X, $\mathcal{X}$, P_X, and x to represent a discrete random variable, the range, the probability distribution, and the value of X. In addition, for any function $s(X) \in \mathbb{R}^k$ of X, we use μ_s to denote the mean of $s(X)$, and "$\tilde{\ }$" to denote the mean removed version of a variable; e.g., $\tilde{s}(X) = s(X) - \mu_s$. Finally, we use $\|\cdot\|$ and $\|\cdot\|_{\mathrm{F}}$ to denote the ℓ_2-norm and the Frobenius norm, respectively.

II. Preliminary and Definition

Given a pair of discrete random variables X, Y with the joint distribution $P_{XY}(x, y)$, the $|\mathcal{Y}| \times |\mathcal{X}|$ matrix $\tilde{\mathbf{B}}$ is defined as

$$\tilde{\mathbf{B}}(y, x) \triangleq \frac{P_{XY}(x, y) - P_X(x)P_Y(y)}{\sqrt{P_X(x)P_Y(y)}}, \tag{1}$$

where $\tilde{\mathbf{B}}(y, x)$ is the (y, x)th entry of $\tilde{\mathbf{B}}$. The matrix $\tilde{\mathbf{B}}$ is referred to as the canonical dependence matrix (CDM). The SVD of $\tilde{\mathbf{B}}$ has the following properties [3].

Lemma 1. *The SVD of $\tilde{\mathbf{B}}$ can be written as $\tilde{\mathbf{B}} = \sum_{i=1}^{K} \sigma_i \, \boldsymbol{\psi}_i^Y \left(\boldsymbol{\psi}_i^X\right)^{\mathrm{T}}$, where $K \triangleq \min\{|\mathcal{X}|, |\mathcal{Y}|\}$, and σ_i denotes the ith singular value with the ordering $1 \geq \sigma_1 \geq \cdots \geq \sigma_K = 0$, and $\boldsymbol{\psi}_i^Y$ and $\boldsymbol{\psi}_i^X$ are the corresponding left and right singular vectors with $\psi_K^X(x) = \sqrt{P_X(x)}$ and $\psi_K^Y(y) = \sqrt{P_Y(y)}$.*

This SVD decomposes the feature spaces of X, Y into maximally correlated features. To see that, consider the generalized canonical correlation analysis (CCA) problem:

$$\max_{\substack{\mathbb{E}[f_i(X)]=\mathbb{E}[g_i(Y)]=0 \\ \mathbb{E}\left[f_i(X)\,f_j(X)\right]=\mathbb{E}\left[g_i(Y)\,g_j(Y)\right]=\mathbb{1}_{i=j}}} \sum_{i=1}^{k} \mathbb{E}\left[f_i(X)\,g_i(Y)\right].$$

It can be shown that for any $1 \leq k \leq K-1$, the optimal features are $f_i(x) = \psi_i^X(x)/\sqrt{P_X(x)}$, and $g_i(y) = \psi_i^Y(y)/\sqrt{P_Y(y)}$, for $i = 0, \ldots, K-1$, where $\psi_i^X(x)$ and $\psi_i^Y(y)$ are the xth and yth entries of $\boldsymbol{\psi}_i^X$ and $\boldsymbol{\psi}_i^Y$, respectively [3]. The special case $k = 1$ corresponds to the HGR maximal correlation [5]–[7], and the optimal features can be computed from the ACE algorithm [8].

Moreover, in this paper we focus on a particular analysis regime described as follows.

Definition 1 (ϵ-Neighborhood)**.** *Let $\mathcal{P}^{\mathcal{X}}$ denote the space of distributions on some finite alphabet $\mathcal{X}$, and let $\operatorname{relint}(\mathcal{P}^{\mathcal{X}})$ denote the subset of strictly positive distributions. For a given $\epsilon > 0$, the ϵ-neighborhood of a distribution $P_X \in \operatorname{relint}(\mathcal{P}^{\mathcal{X}})$ is defined by the χ^2-divergence as*

$$\mathcal{N}_\epsilon^{\mathcal{X}}(P_X) \triangleq \left\{ P \in \mathcal{P}^{\mathcal{X}} \colon \sum_{x\in\mathcal{X}} \frac{\left(P(x) - P_X(x)\right)^2}{P_X(x)} \leq \epsilon^2 \right\}.$$

Definition 2 (ϵ-Dependence)**.** *The random variables X, Y is called ϵ-dependent if $P_{XY} \in \mathcal{N}_\epsilon^{\mathcal{X}\times\mathcal{Y}}(P_X P_Y)$.*

Definition 3 (ϵ-Attribute)**.** *A random variable U is called an ϵ-attribute of X if $P_{X|U}(\cdot|u) \in \mathcal{N}_\epsilon^{\mathcal{X}}(P_X)$, for all $u \in \mathcal{U}$.*

Throughout this paper, we focus on the small ϵ regime, where we refer to as the local analysis regime. In addition, for any $P \in \mathcal{P}^{\mathcal{X}}$, we define the *information vector* ϕ and *feature function* $L(x)$ corresponding to P, with respect to a reference distribution $P_X \in \operatorname{relint}(\mathcal{P}^{\mathcal{X}})$, as

$$\phi(x) \triangleq \frac{P(x) - P_X(x)}{\sqrt{P_X(x)}}, \quad L(x) \triangleq \frac{\phi(x)}{\sqrt{P_X(x)}}. \tag{2}$$

This gives a three way correspondence $P \leftrightarrow \phi \leftrightarrow L$ for all distributions in $\mathcal{N}_\epsilon^{\mathcal{X}}(P_X)$, which will be useful in our derivations.

Due to the space limitations, we omit the proofs of the lemmas and theorems in the rest of this paper, but refer the readers to the extended version of this paper [9] for the detailed proofs.

III. Universal Feature Selection

Suppose that given random variables X, Y with joint distribution P_{XY}, we want to infer about an attribute V of Y from observed i.i.d. samples $x_1, \ldots, x_n$ of X. When the statistical model $P_{X|V}$ is known, the optimal decision rule is the log-likelihood ratio test, where the log-likelihood function can be viewed as the optimal feature for inference. However, in many practical situations [3], it is hard to identify the model of the targeted attribute, and is necessary to select low-dimensional informative features of X for inference tasks before knowing the model. We call this universal feature selection problem. To formalize this problem, for an attribute V, we refer to $\mathcal{C}_{\mathcal{Y}} = \left\{ \mathcal{V},\ \{P_V(v),\ v \in \mathcal{V}\},\ \{\boldsymbol{\phi}_v^{Y|V},\ v \in \mathcal{V}\} \right\}$, as the *configuration* of V, where $\phi_v^{Y|V} \leftrightarrow P_{Y|V}(\cdot|v)$ is the information vector specifying the corresponding conditional distribution $P_{Y|V}(\cdot|v)$. The configuration of V models the statistical correlation between V and Y. In the sequel, we focus on the local analysis regime, for which we assume that all the attributes V of our interests to detect are ϵ-attributes of Y. As a result, the corresponding configuration satisfies $\|\phi_v^{Y|V}\| \leq \epsilon$, for all $v \in \mathcal{V}$. We refer to this as the *ϵ-configurations*. The configuration of V is unknown in advance, but assumed to be generated from a *rotational invariant ensemble (RIE)*.

Definition 4 (RIE)**.** *Two configurations $\mathcal{C}_{\mathcal{Y}}$ and $\tilde{\mathcal{C}}_{\mathcal{Y}}$ defined as*

$$\mathcal{C}_{\mathcal{Y}} = \left\{ \mathcal{V},\ \{P_V(v),\ v \in \mathcal{V}\},\ \{\boldsymbol{\phi}_v^{Y|V},\ v \in \mathcal{V}\} \right\},$$
$$\tilde{\mathcal{C}}_{\mathcal{Y}} \triangleq \left\{ \mathcal{V},\ \{P_V(v),\ v \in \mathcal{V}\},\ \{\tilde{\boldsymbol{\phi}}_v^{Y|V},\ v \in \mathcal{V}\} \right\}$$

are called rotationally equivalent, if there exists a unitary matrix $\mathbf{Q}$ such that $\tilde{\boldsymbol{\phi}}_v^{Y|V} = \mathbf{Q}\,\boldsymbol{\phi}_v^{Y|V}$, for all $v \in \mathcal{V}$. Moreover, a probability measure defined on a set of configurations is called an RIE, if all rotationally equivalent configurations have the same measure.

The RIE can be interpreted as assigning a uniform measure to the attributes with the same level of distinguishability. To infer about the attribute V, we construct a k-dimensional feature vector $h^k = (h_1, \ldots, h_k)$, for some $1 \leq k \leq K-1$, of the form $h_i = \frac{1}{n}\sum_{l=1}^{n} f_i(x_l),\ i = 1, \ldots, k$, for some choices of feature functions f_i. Our goal is to determine the f_i such that the optimal decision rule based on h^k achieves the smallest possible error probability, where the performance is averaged over the possible $\mathcal{C}_{\mathcal{Y}}$ generated from an RIE. In turn, we denote $\xi_i^X \leftrightarrow f_i$ as the corresponding information vector, and define the matrix $\boldsymbol{\Xi}^X \triangleq [\boldsymbol{\xi}_1^X\ \cdots\ \boldsymbol{\xi}_k^X]$.

Theorem 1 (Universal Feature Selection)**.** *For $v, v' \in \mathcal{V}$, let $E_{h^k}(v, v')$ be the error exponent associated with the pairwise error probability distinguishing v and v' based on h^k, then the expectation of the error exponent over a given RIE defined on the set of ϵ-configuration is given by*

$$\mathbb{E}\left[E_{h^k}(v, v')\right] = \frac{\mathbb{E}\left[\left\|\boldsymbol{\phi}_v^{Y|V} - \boldsymbol{\phi}_{v'}^{Y|V}\right\|^2\right]}{8|\mathcal{Y}|} \left\|\tilde{\mathbf{B}}\boldsymbol{\Xi}^X\left(\left(\boldsymbol{\Xi}^X\right)^{\mathrm{T}}\boldsymbol{\Xi}^X\right)^{-\frac{1}{2}}\right\|_{\mathrm{F}}^2 + o(\epsilon^2), \tag{3}$$

where the expectations are taken over this RIE.

As a result of (3), designing the $\boldsymbol{\xi}_i^X$ as the singular vectors $\boldsymbol{\psi}_i^X$ of $\tilde{\mathbf{B}}$, for $i = 1, \ldots, k$, optimizes (3) for all RIEs, pairs of (v, v'), and ϵ-configurations. Thus, the feature functions corresponding to $\boldsymbol{\psi}_i^X$ are *universally optimal* for inferring the unknown attribute V. Moreover, (3) naturally leads to an information metric $\|\tilde{\mathbf{B}}\boldsymbol{\Xi}^X\left(\left(\boldsymbol{\Xi}^X\right)^{\mathrm{T}}\boldsymbol{\Xi}^X\right)^{-\frac{1}{2}}\|_{\mathrm{F}}^2$ for any feature $\boldsymbol{\Xi}^X$ of X, measured by projecting the normalized $\boldsymbol{\Xi}^X$ through a linear projection $\tilde{\mathbf{B}}$. This information metric quantifies how informative a feature of X is when solving inference problems with respect to Y, and is optimized when designing features by singular vectors of $\tilde{\mathbf{B}}$. Thus, we can interpret the universal feature selection as solving the most informative features for

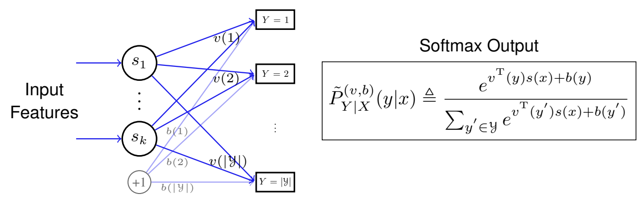


Fig. 1: A simple neural network with one layer of hidden nodes with softmax output.

data inferences via the SVD of $\tilde{\mathbf{B}}$, which also coincides with the maximally correlated features in Section II. Later on we will show that the feature selections in DNN share the same information metric as universal feature selection in the local analysis regime.

IV. Interpreting Softmax Regression

To begin, recall that for a data vector X and label Y with labeled samples (x_i, y_i), for $i = 1, \dots, n$, the softmax regression generally uses a discriminative model of the form

$$\tilde{P}^{(v,b)}_{Y|X}(y|x) \triangleq \frac{e^{v^{\mathrm{T}}(y)s(x)+b(y)}}{\sum_{y'\in\mathcal{Y}} e^{v^{\mathrm{T}}(y')s(x)+b(y')}} \tag{4}$$

to address the classification problems, where $s(x) \in \mathbb{R}^k$ is a k-dimensional representation of X used to predict the label, and $v(y) \in \mathbb{R}^k$ and $b(y) \in \mathbb{R}$ are the parameters required to be learned from

$$(v, b)^* = \operatorname*{arg\,max}_{(v,b)} \frac{1}{N} \sum_{i=1}^{N} \log \tilde{P}^{(v,b)}_{Y|X}(y_i|x_i). \tag{5}$$

As depicted in Fig. 1, the ordinary softmax regression corresponds to $s(x) = x$. More generally, $s(x)$ can be the output of the previous hidden layer of a neural network, i.e., the selected feature of x fed into the softmax regression. In the rest of this section, we will show that when X, Y are ϵ-dependent, the functions $s(x)$ and $v(y)$ coincide with the solutions of the universal feature selection.

First, we use P_{XY} to denote the joint empirical distribution of the labeled samples $(x_i, y_i), i = 1, \dots, N$, and P_X, P_Y to denote the corresponding marginal distributions. Then, the objective function in the optimization problem (5) is precisely the empirical average of the log-likelihood, i.e., $\frac{1}{N}\sum_{i=1}^{N} \log \tilde{P}^{(v,b)}_{Y|X}(y_i|x_i) = \mathbb{E}_{P_{XY}}\left[\log \tilde{P}^{(v,b)}_{Y|X}(Y|X)\right]$. Therefore, maximizing this empirical average is equivalent as minimizing the K-L divergence:

$$(v, b)^* = \operatorname*{arg\,min}_{(v,b)} \quad D(P_{XY} \| P_X\, \tilde{P}^{(v,b)}_{Y|X}). \tag{6}$$

This can be interpreted as finding the best fitting to empirical joint distribution P_{XY} by distributions of the form $P_X\, \tilde{P}^{(v,b)}_{Y|X}$. In our development, it is more convenient to denote the bias by $d(y) = b(y) - \log P_Y(y)$, for $y \in \mathcal{Y}$. Then, the following lemma illustrates the explicit constraint on the problem (6) in the local analysis regime.

Lemma 2. *If X, Y are ϵ-dependent, then the optimal v, d for (6) satisfy*

$$|\tilde{v}^{\mathrm{T}}(y)s(x) + \tilde{d}(y)| = O(\epsilon), \quad \textit{for all } x \in \mathcal{X},\ y \in \mathcal{Y}. \tag{7}$$

In turn, we take (7) as the constraint for solving the problem (6) in the local analysis regime. Moreover, we define the information vectors for zero-mean vectors $\tilde{s}$, $\tilde{v}$ as $\xi^X(x) = \sqrt{P_X(x)}\, \tilde{s}(x)$, $\xi^Y(y) = \sqrt{P_Y(y)}\, \tilde{v}(y)$, and define matrices

$$\mathbf{\Xi}^Y \triangleq \left[\xi^Y(1) \quad \cdots \quad \xi^Y(|\mathcal{Y}|)\right]^{\mathrm{T}},$$
$$\mathbf{\Xi}^X \triangleq \left[\xi^X(1) \quad \cdots \quad \xi^X(|\mathcal{X}|)\right]^{\mathrm{T}}.$$

Lemma 3. *The K-L divergence (6) in the local analysis regime (7) can be expressed as*

$$D(P_{XY} \| P_X\, \tilde{P}^{(v,b)}_{Y|X}) = \frac{1}{2}\left\|\tilde{\mathbf{B}} - \mathbf{\Xi}^Y (\mathbf{\Xi}^X)^{\mathrm{T}}\right\|_{\mathrm{F}}^2 + \frac{1}{2}\eta^{(v,b)}(s) + o(\epsilon^2), \tag{8}$$

where $\eta^{(v,b)}(s) \triangleq \mathbb{E}_{P_Y}\left[(\mu_s^{\mathrm{T}}\tilde{v}(Y) + \tilde{d}(Y))^2\right]$.

Eq. (8) reveals key insights for feature selection in neural networks, which are illustrated by the following three learning problems, depending on if the weights, input feature, or both can be trained from data.

A. Forward Feature Projection

For the case that s is fixed, we can optimize (8) with $\mathbf{\Xi}^X$ fixed and get the following optimal weights:

Theorem 2. *For fixed $\mathbf{\Xi}^X$ and μ_s, the optimal $\mathbf{\Xi}^{Y*}$ to minimize (8) is given by*

$$\mathbf{\Xi}^{Y*} = \tilde{\mathbf{B}}\, \mathbf{\Xi}^X \left((\mathbf{\Xi}^X)^{\mathrm{T}} \mathbf{\Xi}^X\right)^{-1}, \tag{9}$$

and the optimal weights $\tilde{v}^*$ and bias $\tilde{d}^*$ are

$$\tilde{v}^*(y) = \mathbb{E}_{P_{X|Y}}\left[\mathbf{\Lambda}^{-1}_{\tilde{s}(X)}\, \tilde{s}(X) \,\middle|\, Y = y\right], \quad \tilde{d}^*(y) = -\mu_s^{\mathrm{T}}\, \tilde{v}(Y). \tag{10}$$

where $\mathbf{\Lambda}_{\tilde{s}(X)}$ denotes the covariance matrix of $\tilde{s}(X)$.

Eq. (9) can be viewed as a projection of the input feature $\tilde{s}(x)$, to a feature $v(y)$ computable from the value of y, which is the most correlated feature to $\tilde{s}(x)$. The solution is given by left multiplying the $\tilde{\mathbf{B}}$ matrix. We call this the "forward feature projection".

Remark 1. *While we assume the continuous input $s(x)$ is a function of a discrete variable X, we only need the labeled samples between s and Y to compute the weights and bias from the conditional expectation (10), and the correlation between X and s is irrelevant. Thus, our analysis for weights and bias can be applied to continuous input networks by just ignoring X and taking s as the real network input.*

B. Backward Feature Projection

It is also useful to consider the "backward problem", which attempts to find informative feature $s^*(X)$ to minimize the loss (8) with given weights and bias.

Theorem 3. *For fixed $\tilde{v}$, $\mathbf{\Xi}^Y$, and $\tilde{d}$, the optimal $\mathbf{\Xi}^{X*}$ to minimize (8) is given by*

$$\mathbf{\Xi}^{X*} = \tilde{\mathbf{B}}^{\mathrm{T}}\, \mathbf{\Xi}^Y \left((\mathbf{\Xi}^Y)^{\mathrm{T}} \mathbf{\Xi}^Y\right)^{-1}, \tag{11}$$

and the optimal feature function s^*, which are decomposed to $\tilde{s}^*$ and μ_s^*, are given by

$$\tilde{s}^*(x) = \mathbb{E}_{P_{Y|X}}\left[\mathbf{\Lambda}^{-1}_{\tilde{v}(Y)}\, \tilde{v}(Y) \middle| X = x\right],$$

$$\mu_s^* = -\mathbf{\Lambda}_{\tilde{v}(Y)}^{-1}\,\mathbb{E}_{P_Y}\left[\tilde{v}(Y)\,\tilde{d}(Y)\right], \tag{12}$$

where $\mathbf{\Lambda}_{\tilde{v}(Y)}$ *denotes the covariance matrix of* $\tilde{v}(Y)$.

The solution of this backward feature projection is precisely symmetric to the forward one. Note we assumed here that the feature $s(X)$ can be selected as any desired function. This is only true in the ideal case where the previous hidden layers of the neural network have sufficient expressive power. That is, it can generate the desired feature function as given in (12). In general, however, the form of feature functions that can be generalized is often limited by the network structure. In the next section, we discuss such cases, where we do know the most desirable feature function as given in (12), and the question is how does a network with limited expressive power approximate this optimal solution.

C. Universal Feature Selection

When both s and (v, b) (and hence $\mathbf{\Xi}^X, \mathbf{\Xi}^Y, d$) can be designed, the optimal $(\mathbf{\Xi}^Y, \mathbf{\Xi}^X)$ corresponds to the low rank factorization of $\tilde{\mathbf{B}}$, and the solutions coincide with the universal feature selection.

Theorem 4. *The optimal solutions for weights and bias to maximize* (8) *are given by* $\tilde{d}(y) = -\mu_s^{\mathrm{T}}\tilde{v}(y)$, *and* $(\mathbf{\Xi}^Y, \mathbf{\Xi}^X)^*$ *chosen as the largest* k *left and right singular vectors of* $\tilde{\mathbf{B}}$.

Therefore, we conclude that the softmax regression, when both s and (v, b) are designable, is to extract the most correlated aspects of the input data X and the label Y that are informative features for data inferences from universal feature selection.

In the learning process of DNN, the BackProp procedure alternatively chooses the weights of the softmax layer and those on the previous layer(s). In each step, the weights on the rest of the network are fixed. This is equivalent as alternating between the forward and the backward feature projections, i.e. it alternates between (9) and (11). This is in fact the power method to solve the SVD for $\tilde{\mathbf{B}}$ [10], which is also known as the Alternating Conditional Expectation (ACE) algorithm [8].

V. Multi-Layer Network Analysis

From the previous discussions, the performance of the softmax regression not only depends on the weight and bias $(v(y), b(y))$, but the input feature $s(x)$ has to be informative. It turns out that the hidden layers of neural networks, which are known to have strong expressive power of features, are essentially extracting such informative features. For illustration, we consider the neural network with a hidden layer of k nodes, and a zero-mean continuous input $t = [t_1 \ \cdots \ t_m]^{\mathrm{T}} \in \mathbb{R}^m$ to this hidden layer, where t is assumed to be a function $t(x)$ of some discrete variable X[1]. Our goal is to analyze the weights and bias in this layer with labeled samples $(t(x_i), y_i)$. Assume the activation function of the hidden layer is a generally smooth

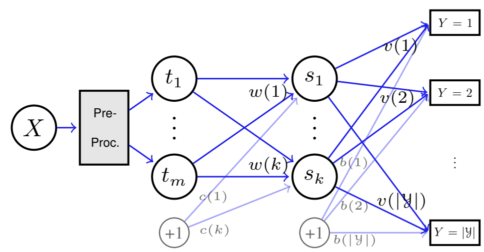


Fig. 2: A multi-layer network: all hidden layers previous to t are labeled as "pre-processing".

function $\sigma(\cdot)$, then the output $s_z(X)$ of the z-th hidden node is

$$s_z(x) = \sigma\left(w^{\mathrm{T}}(z)t(x) + c(z)\right), \quad \text{for } z = 1, \ldots, k, \ x \in \mathcal{X}, \tag{13}$$

where $w(z) \in \mathbb{R}^m$ and $c(z) \in \mathbb{R}$ are the weights and bias from input layer to hidden layer as shown in Fig. 2. We denote $s = [s_1 \ \cdots \ s_k]^{\mathrm{T}}$ as the input vector to the output softmax regression layer.

To interpret the feature selection in hidden layers, we fix $(v(y), b(y))$ at the output layer, and consider the problem of designing $(w(z), c(z))$ to minimize the loss function (6) of the softmax regression at the output layer. Ideally, we should have picked $w(z)$ and $c(z)$ to generate $s(x)$ to match $s^*(x)$ from (12), which minimizes the loss. However, here we have the constraint that $s(x)$ must take the form of (13), and intuitively the network should select $w(z), c(z)$ so that $s(x)$ is close to $s^*(x)$. Our goal is to quantify the notion of closeness in the local analysis regime.

To develop insights on feature selection in hidden layers, we again focus on the local analysis regime, where the weights and bias are assumed with the local constraint

$$\left|\tilde{v}^{\mathrm{T}}(y)s(x) + \tilde{d}(y)\right| = O(\epsilon), \ \left|w^{\mathrm{T}}(z)\tilde{t}(x)\right| = O(\epsilon), \ \forall x, y, z. \tag{14}$$

Then, since t is zero-mean, we can express (13) as

$$\begin{aligned} s_z(x) &= \sigma\left(w^{\mathrm{T}}(z)t(x) + c(z)\right) \\ &= w^{\mathrm{T}}(z)\tilde{t}(x) \cdot \sigma'\left(c(z)\right) + \sigma\left(c(z)\right) + o(\epsilon), \end{aligned}$$

Moreover, we define a matrix $\tilde{\mathbf{B}}_1$ with the (z, x)th entry $\tilde{\mathbf{B}}_1(z, x) = \frac{\sqrt{P_X(x)}}{\sigma'(c(z))}\tilde{s}_z^*(x)$, which can be interpreted as a generalized DTM for the hidden layer. Furthermore, we denote $\xi_1^X(x) = \sqrt{P_X(x)}\,\tilde{t}(x)$ as the information vector of $\tilde{t}(x)$ with the matrix $\mathbf{\Xi}_1^X$ defined as $\mathbf{\Xi}_1^X \triangleq \left[\xi_1^X(1) \ \cdots \ \xi_1^X(|\mathcal{X}|)\right]^{\mathrm{T}}$, and we also define

$$\mathbf{W} \triangleq [w(1), w(2), \cdots, w(k)]^{\mathrm{T}}$$
$$\mathbf{J} \triangleq \mathrm{diag}\{\sigma'(c(1)), \sigma'(c(2)), \cdots, \sigma'(c(k))\}.$$

The following theorem characterizes the loss (6).

Theorem 5. *Given the weights and bias* (v, b) *at the output layer, and for any input feature* s, *we denote* $\mathcal{L}(s)$ *as the loss* (6) *evaluated with respect to* (v, b) *and* s. *Then, with the constraints* (14)

$\mathcal{L}(s) - \mathcal{L}(s^*)$

[1] As discussed in Remark 1, X is assumed only for the convenience of analysis, and the computation of weights and bias only needs t, but not X. Moreover, the input t to the hidden layer can be either directly from data or the output of previous hidden layers in a DNN, which we model as "pre-processing" as shown in Fig. 2.

$$= \frac{1}{2}\left\|\Theta\tilde{\mathbf{B}}_1 - \Theta\mathbf{W}\left(\boldsymbol{\Xi}_1^X\right)^{\mathrm{T}}\right\|_{\mathrm{F}}^2 + \frac{1}{2}\kappa^{(v,b)}(s,s^*) + o(\epsilon^2), \quad (15)$$

where $\Theta \triangleq \left(\left(\boldsymbol{\Xi}^Y\right)^{\mathrm{T}}\boldsymbol{\Xi}^Y\right)^{1/2}\mathbf{J}$, *and the term* $\kappa^{(v,b)}(s,s^*) = (\mu_s - \mu_{s^*})^{\mathrm{T}}\boldsymbol{\Lambda}_{\tilde{v}(Y)}(\mu_s - \mu_{s^*})$.

Eq. (15) quantifies the closeness between s and s^* in terms of the loss (6). Then, our goal is to minimize (15), which can be separated to two optimization problems:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \left\|\Theta\tilde{\mathbf{B}}_1 - \Theta\mathbf{W}\left(\boldsymbol{\Xi}_1^X\right)^{\mathrm{T}}\right\|_{\mathrm{F}}^2, \quad (16)$$

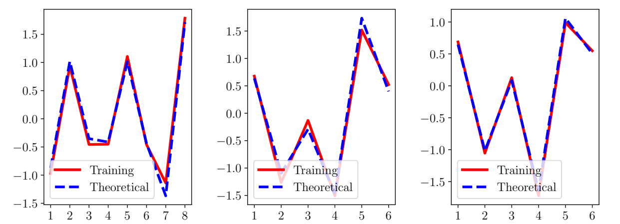
$$\mu_s^* = \arg\min_{\mu_s} \kappa^{(v,b)}(s,s^*). \quad (17)$$

First note that the optimization problem (16) is similar to the ordinary softmax regression depicted in Section IV, and the optimal solution is given by $\mathbf{W}^* = \tilde{\mathbf{B}}_1\boldsymbol{\Xi}_1^X\left(\left(\boldsymbol{\Xi}_1^X\right)^{\mathrm{T}}\boldsymbol{\Xi}_1^X\right)^{-1}$. Therefore, solving the optimal weights in the hidden layer can be interpreted as projecting $\tilde{s}^*(x)$ to the subspace of feature functions spanned by $t(x)$ to find the closest expressible function. Finally, the problem (17) is to choose μ_s (and hence the bias $c(z)$) to minimize the quadratic term similar to $\eta^{(v,b)}(s)$ in (8), and the optimal solution of (17) is referred to [9].

Overall, we observe the correspondence between (9), (12), and (16), (17), and interpret both operations as feature projections. Our argument can be generalized to any intermediate layer in a multi-layer network, with all the previous layers viewed as the fixed pre-processing that specifies $t(x)$, and all the layers after determining s^*. Then the iterative procedure in back-propagation can be viewed as alternating projection finding the fixed-point solution over the entire network. This final fixed-point solution, even under the local assumption, might not be the SVD solution as in Theorem 4. This is because the limited expressive power of the network often makes it impossible to generate the desired feature function. In such cases, the concept of feature projection can be used to quantify this gap, and thus to measure the quality of the selected features.

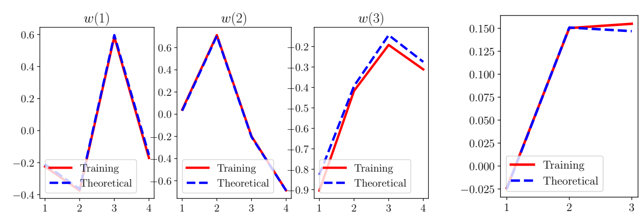
VI. Experimental Validation

We first validate the feature projection in Theorem 4. For this purpose, we construct the NN as shown in Fig. 1 with $k = 1$, $|\mathcal{X}| = 8$, and $|\mathcal{Y}| = 6$, and the input feature $s(X)$ is generated from a sigmoid layer with the one-hot encoded X as the input. Note that with proper weights in the sigmoid layer, $s(X)$ can express any desired function, up to scaling and shifting. To compare the result trained by the neural network and that in Theorem 4, we first randomly generate a distribution P_{XY}, and then generate $n = 100,000$ samples of (X,Y) pair. Using these data to train the neural network, the corresponding results of $s(x), v(y)$ and $b(y)$ are shown in Fig. 3 with a comparison to theoretical result, where the training results match our theory. In addition, we validate Theorem 5 by the NN depicted in Fig. 2, with the same setup of X, Y. The number of neurons in hidden layers are $m = 4$ and $k = 3$, and the input $t(X)$ is some randomly chosen function of X, and the activation $\sigma(\cdot)$ is the sigmoid function. We then fix the weights and bias at the output layer and train the weights $w(1), w(2), w(3)$, and



(a) $s(x)$ (b) $v(y)$ (c) $b(y)$

Fig. 3: The comparisons of the weights and bias in softmax regression.



(a) $w(1), w(2)$ and $w(3)$ (b) c

Fig. 4: The comparisons of the weights and bias in the hidden layer.

bias c in the hidden layer to optimize the Log-Loss. Fig. 3 shows the matching between our results and the experiment.

Acknowledgment

The research of Shao-Lun Huang was funded by the Natural Science Foundation of China 61807021, Shenzhen Science and Technology Research and Development Funds (JCYJ20170818094022586), and Innovation and entrepreneurship project for overseas high-level talents of Shenzhen (KQJSCX20180327144403783).

References

[1] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, ISBN 9780521642989, 2003.

[2] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Information Theory Workshop (ITW), 2015 IEEE*. IEEE, 2015, pp. 1–5.

[3] S.-L. Huang, A. Makur, L. Zheng, and G. W. Wornell, "On universal features for high-dimensional learning and inference," *submitted to IEEE Trans. Inform. Theory*, 2019. Preprint.

[4] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[5] H. O. Hirschfeld, "A connection between correlation and contingency," *Proc. Cambridge Phil. Soc.*, vol. 31, pp. 520–524, 1935.

[6] H. Gebelein, "Das statistische problem der korrelation als variations-und eigenwertproblem und sein zusammenhang mit der ausgleichungsrechnung," *Z. für angewandte Math., Mech.*, vol. 21, pp. 364–379, 1941.

[7] A. Rényi, "On measures of dependence," *Acta Mathematica Academiae Scientiarum Hungarica*, vol. 10, no. 3–4, pp. 441–451, 1959.

[8] L. Breiman and J. H. Friedman, "Estimating optimal transformations for multiple regression and correlation," *Journal of American Statistical Assocication*, vol. 80, no. 391, pp. 614–619, 1985.

[9] S.-L. Huang, X. Xu, L. Zheng, and G. W. Wornell, "An information theoretic interpretation to deep neural networks," *preprint*, 2019. [Online]. Available: https://xiangxiangxu.com/media/documents/isit2019a.pdf

[10] J. Stoer and R. Bulirsch, *Introduction to numerical analysis*. Springer Science & Business Media, 2013, vol. 12.