NEURAL NETWORK PERFORMS SUFFICIENT DIMENSION REDUCTION UNDER RANK REGULARIZATION

A PREPRINT

Shuntuo Xu
KLATASDS-MOE
School of Statistics
East China Normal University
Shanghai, China
oaksword@163.com

Zhou Yu *
KLATASDS-MOE
School of Statistics
East China Normal University
Shanghai, China
zyu@stat.ecnu.edu.cn

January 19, 2024

ABSTRACT

This paper investigates the interpretability of neural networks under rank regularization. It is demonstrated that, in regression tasks, the optimal solution of a neural network automatically achieves sufficient dimension reduction when using least squares loss. Specifically, the weights in the first layer of the optimal neural network asymptotically span a linear space that contains the central mean space, and the corresponding statistical consistency is established. This suggests that neural networks are well-suited for addressing tasks related to sufficient dimension reduction. Numerical experiments further illustrate the strong performance of our proposed method based on neural networks.

Keywords Interpretability · Neural network · Rank regularization · Sufficient dimension reduction

1 Introduction

Neural networks have achieved significant success in various applications [Lee and Abu-El-Haija, 2017, Silver et al., 2018, Jumper et al., 2021], and the associated interpretation has been widely discussed. Specifically, a feedforward neural network is constructed by a series of linear transformations and nonlinear activations. To be more mathematical, a function f implemented by a feedforward neural network of L layers can be represented as

$$f(x) = \phi_L \circ \sigma_{L-1} \circ \phi_{L-1} \circ \cdots \circ \sigma_1 \circ \phi_1(x). \tag{1}$$

Here, $\circ$ is the functional composition operator, $\phi_i(z) = W_i^{\mathrm{T}} z + b_i$ denotes the linear transformation and σ_i represents the elementwise nonlinear activation function $(i = 1, \ldots, n)$. Intuitively, the formula (1) provides limited insight into how the information within the input data is processed by the neural network. As a result, there have been various efforts to unravel the interpretability of neural networks.

The precise definition of interpretability needs to be specified, as varying definitions indicate different perspectives [Doshi-Velez and Kim, 2017, Guidotti et al., 2018, Zhang et al., 2021]. For instance, Post-hoc interpretability [Lipton, 2018] focuses on the predictions generated by neural networks, while disregarding the detailed mechanism and feature importance. Ghorbani et al. [2019] highlighted the fragility of this type of interpretation, as indistinguishable perturbations could result in completely different interpretations. In this paper, the interpretability is defined as the fundamental structure acquired by the neural network, in the regression task with the presence of intrinsic structures within the input data.

Let $x \in \mathbb{R}^p$ represent the covariates and $y \in \mathbb{R}$ represent the response. Consider the model that

$$y = f_0(B_0^{\mathrm{T}} x) + \epsilon, \tag{2}$$

where $B_0 \in \mathbb{R}^{p \times d}$ is a nonrandom matrix with $d \leq p$, $f_0 : \mathbb{R}^d \to \mathbb{R}$ is an unknown function, and ϵ is the noise such that $E(\epsilon \mid x) = 0$ and $\mathrm{var}(\epsilon \mid x) = \nu^2$ for some constant ν. In other words, the equation (2) constrains that only certain

linear combinations of the covariates directly contribute to the response. Numerous studies have demonstrated the ability of neural networks to approximate continuous functions [Hornik et al., 1989, Barron, 1993, Yarotsky, 2017, Shen et al., 2021]. Nevertheless, it is of interest to investigate whether neural networks are capable of identifying the intrinsic structure encapsulated in B_0.

Our study was inspired by the observation that the the weight matrix in the first layer, i.e., W_1, can accurately detect the presence of B_0 in a toy data set, with (or without) the rank regularization. Specifically, for the simulated setting $y = (B_0^{\mathrm{T}} x)^3 + \epsilon$ where $x \sim \mathrm{Uniform}([0,1]^{10})$, $B_0 = (1, -2, 0, \ldots, 0)^{\mathrm{T}}$ and $\epsilon \sim \mathrm{Normal}(0, 0.1^2)$, we trained a neural network using the least squares loss with $W_1 = W_{11} W_{12}$ where $W_{11} \in \mathbb{R}^{10 \times q}$ and $W_{12} \in \mathbb{R}^{q \times 64}$ for $q = 1, \ldots, 10$. Evidently, the rank of W_1 does not exceed q. It was then observed that for each q, (i) B_0 was closely contained within the space spanned by the columns of W_{11}, and (ii) the eigenvector corresponding to the largest eigenvalue of $W_1 W_1^{\mathrm{T}}$ closely aligned with B_0 (see Fig. 1). This observation indicates that the first layer of the neural network may potentially uncover the underlying structure within the input data. Specifically, when $q = d$, W_{11} recovers B_0.

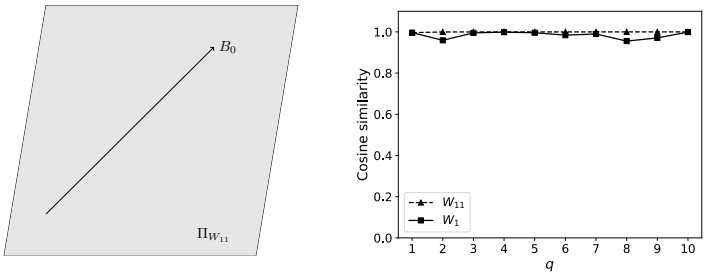

Figure 1: Results of the toy data set. The left picture demonstrates that the true direction B_0 lies in the space spanned by the columns of W_{11}, denoted as $\Pi_{W_{11}}$. The right line plot shows the cosine similarity between (i) B_0 and its projection on $\Pi_{W_{11}}$, (ii) B_0 and the eigenvector corresponding to the largest eigenvalue of $W_1 W_1^{\mathrm{T}}$.

In statistical learning, the model (2) has been extensively studied [Kong and Xia, 2007, Ma and Zhu, 2013, Meng et al., 2020]. Particularly, when x is independent from ϵ, the model attains the property that $y \perp\!\!\!\perp x \mid B_0^{\mathrm{T}} x$, which is the condition of interest in sufficient dimension reduction (SDR). Here, $\perp\!\!\!\perp$ denotes statistical independence. SDR aims to estimate the space spanned by the columns of B_0 in (2), which is called the central mean space [Lee et al., 2013]. Relevant estimation methods include sliced inverse regression (SIR, Li [1991]), sliced average variance estimation (SAVE, Cook and Weisberg [1991]), principal Hessian directions (PHD, Li [1992]), minimum average variance estimation (MAVE, Xia et al. [2009]) and generalized kernel-based dimension reduction (GKDR, Xie and Zhu [2020]), among others.

This paper investigates the interpretability of neural networks through the lens of SDR. We show that, with suitable rank regularization, the first layer of a feedforward neural network conducts SDR in a regression task, wherein $d(W_{11}, B_0) \to 0$ in probability for some metric $d(\cdot, \cdot)$. Additionally, numerical experiments provide empirical evidence for this result and demonstrate the effective performance of neural networks in addressing the SDR problem.

Throughout this paper, we use $\|v\|_2$ to represent the Euclidean norm of a vector v. For a matrix A, $\|A\|_F$ is the Frobenius norm of A, $\pi_A = A(A^{\mathrm{T}}A)^{-}A^{\mathrm{T}}$ denotes the projection matrix of A where A^{-} is the generalized inverse of A, and Π_A stands for the linear space spanned by the columns of A. For a measurable function $f : \mathcal{X} \to \mathbb{R}$, $\|f\|_{L^2(\mu)}$ represents the L^2 norm of f with respect of a given probability measure μ, and $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)|$ represents the supreme norm of f. $\mathcal{B}(\mathcal{X})$ is the unit ball induced by a set $\mathcal{X}$ such that $\mathcal{B}(\mathcal{X}) \subset \mathcal{X}$ and $\|v\|_2 \leq 1$ for any $v \in \mathcal{X}$.

2 Consistency

2.1 Main theorem

Suppose the true intrinsic dimension d defined in model (2) is known and the covarites $x \in \mathcal{B}([0,1]^p)$. For sake of identifiability, it is assumed without loss of generality that $B_0^{\mathrm{T}} B_0 = I_d$ where I_d is the identity matrix with d rows. By defining $\Psi_d = \{B \in \mathbb{R}^{p \times d} : B^{\mathrm{T}} B = I_d\}$, we have $B_0 \in \Psi_d$. In this paper, we consider the following neural network

function class

$$\mathcal{F}_{\mathcal{L},\mathcal{M},\mathcal{S},\mathcal{R}} = \Big\{ f(x) = \phi_L \circ \sigma_{L-1} \circ \phi_{L-1} \circ \cdots \circ \sigma_1 \circ \phi_1(B^{\mathrm{T}} x) : \\ B \in \Psi_d, \phi_i(z) = W_i^{\mathrm{T}} z + b_i, W_i \in \mathbb{R}^{d_i \times d_{i+1}}, i = 1, \ldots, L, \\ \text{with } L \le \mathcal{L}, \max_{i=1,\ldots,L+1} d_i \le \mathcal{M}, \sum_{i=1}^{L} (d_i + 1) d_{i+1} \le \mathcal{S}, \|f\|_\infty \le \mathcal{R} \Big\}.$$

The activation functions $\sigma_i (i = 1, \ldots, L-1)$ utilized are the rectified linear units. We emphasize that $\mathcal{F}_{\mathcal{L},\mathcal{M},\mathcal{S},\mathcal{R}}$ incorporates rank regularization of d in the first layer. For arbitrary $f \in \mathcal{F}_{\mathcal{L},\mathcal{M},\mathcal{S},\mathcal{R}}$, we define $\mathcal{T}(f)$ to present the component $B \in \Psi_d$ in the first layer of f.

In the regression task, the asymptotic property is highly related to the smoothness of underlying conditonal mean function [Yang and Barron, 1999]. Here, we make the following assumptions of model (2).

Assumption 1 (Smoothness)**.** *f_0 is a Hölder continuous function of order $\alpha \in (0, 1]$ with constant λ, i.e., $|f_0(x) - f_0(z)| \le \lambda \|x - z\|_2^\alpha$ for any $x, z \in [0, 1]^d$. Additionally, $\|f_0\|_\infty \le \mathcal{R}_0$ for some constant $\mathcal{R}_0 > 0$.*

Assumption 2 (Sharpness)**.** *For any scalar $\delta > 0$ and $B \in \Psi_d$, $\|\pi_B - \pi_{B_0}\|_F > \delta$ implies $E[\mathrm{var}\{f_0(B_0^{\mathrm{T}} x) \mid B^{\mathrm{T}} x\}] > M(\delta)$ for some $M(\delta) > 0$.*

Assumption 3. *y is sub-exponentially distributed; that is, there exists $\tau > 0$ such that $E\{\exp(\tau|y|)\} < \infty$.*

Assumption 4. *The density of $B_0^{\mathrm{T}} x$ is bounded by a positive constant κ .*

Assumption 1 is a technical condition to utilize the approximation capability of neural networks [Shen et al., 2019]. Alternatively, other functional spaces such as the Sobolev space can also be employed for this purpose [Abdeljawad and Grohs, 2022, Shen et al., 2022]. Furthermore, Assumption 2 imposes a restriction on the sharpness of f_0. For instance, if f_0 is a constant function of zero, a trivial neural network by setting all the parameters except B as zero can perfectly fit f_0, regardless of the value of B. With assumption 2, it becomes difficult to accurately capture the overall behavior of $f_0(B_0^{\mathrm{T}} x)$ using a biased B. Therefore, it is essential to impose such a constraint to avoid trivial cases. Assumptions 3 and 4 are common conditions applied in tackling the empirical process and concentration inequalities [Van der Vaart, 2000, Zhu et al., 2022].

Theorem 1. *Suppose that Assumptions 1 and 2 hold. Let*

$$f^* = \operatorname*{argmin}_{f \in \mathcal{F}_{\mathcal{L},\mathcal{M},\mathcal{S},\mathcal{R}}} E\{y - f(x)\}^2.$$

Then, we have $\Pi_{\mathcal{T}(f^*)} = \Pi_{B_0}$ when $\mathcal{R}$ is sufficiently large and $\mathcal{L} = \mathcal{M} = \infty$.

Theorem 1 constructs a bridge that links neural networks and SDR, by illustrating that the solution in the neural network function class meets the target of SDR in population level. The details of the proof are provided in the Supplementary Material. Additionally, Theorem 1 suggests that neural networks can be utilized to perform SDR with a minor adjustment to the neural network class.

Empirically, suppose we obtain n sample observations $\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, where (X_i, Y_i) is an independent copy of (x, y) for $i = 1, \ldots, n$. The commonly used least squares loss results in the empirical loss function as

$$L_n(f) = \frac{1}{n} \sum_{i=1}^{n} \{Y_i - f(X_i)\}^2.$$

Let

$$\hat{f}_n = \operatorname*{argmin}_{f \in \mathcal{F}_{\mathcal{L},\mathcal{M},\mathcal{S},\mathcal{R}}} L_n(f).$$

To investigate the closeness of $\Pi_{\mathcal{T}}(\hat{f}_n)$ and Π_{B_0}, we define the metric $d(\cdot, \cdot)$ such that

$$d(f, m_0) = \min_{Q \in \mathcal{Q}} \|B_0 - \mathcal{T}(f)Q\|_2 \vee \|f_0 \circ Q - f \circ \mathcal{T}(f)\|_{L^2(\mu)}.$$

Here, $m_0 = f_0 \circ B_0^{\mathrm{T}}$, $a \vee b$ means $\max(a, b)$, $\mathcal{Q}$ is the collection of all orthogonal matrices in $\mathbb{R}^{d \times d}$, and μ is the probability distribution of $\mathcal{T}(f)^{\mathrm{T}} x$. Clearly, $d(f, m_0) = 0$ implies that there is a orthogonal matrix Q^* satisfying $B_0 = \mathcal{T}(f)Q^*$ and hence $\Pi_{B_0} = \Pi_{\mathcal{T}(f)}$. For any $B \in \Psi_d$, we define

$$d_1(B, B_0) = \min_{Q \in \mathcal{Q}} \|B_0 - BQ\|_2 .$$

We make the following assumption which is another view of Assumption 2.

Assumption 5. *For any positive scalar* δ , $d_1(B, B_0) > \delta$ *implies* $E[\mathrm{var}\{f_0(B_0^{\mathrm{T}}x) \mid B^{\mathrm{T}}x\}] > M_1(\delta)$ *for some* $M_1(\delta) > 0$.

Theorem 2. *Suppose the Assumptions 1, 3, 4 and 5 hold. It follows that* $d(\hat{f}_n, m_0) \to 0$ *in probability, indicating that* $d_1(\mathcal{T}(\hat{f}_n), B_0) \to 0$ *in probability.*

Theorem 2 demonstrates that the estimator $\hat{f}_n$ consistently approaches $m_0 = f_0 \circ B_0^{\mathrm{T}}$ in terms of the metric $d(\cdot, \cdot)$. Consequently, according to the definition of $d(\cdot, \cdot)$, we can infer that $\Pi_{\mathcal{T}(\hat{f}_n)}$ converges to Π_{B_0}. This illustrates that, under the least squares loss, the neural network, with appropriate rank regularization, achieves SDR in order to obtain an empirically optimal solution. Therefore, it is applicable to utilize the neural network function class $\mathcal{F}_{\mathcal{L},\mathcal{M},\mathcal{S},\mathcal{R}}$ to address SDR problems. The proof is presented in the Supplementary Material.

2.2 Discussion

The results presented in Theorem 1 and 2 are contingent on the availability of true intrinsic dimension d. In the case where d is unknown, a natural modification for $\mathcal{F}_{\mathcal{L},\mathcal{M},\mathcal{S},\mathcal{R}}$ is to set $B \in \mathbb{R}^{p\times p}$ without rank regularization. Under Assumptions 1 and 2, in this scenario, the minimum f^* at the population level still attains that $\Pi_{\mathcal{T}(f^*)}$ encompasses Π_{B_0}, with the sharpness of f_0 playing a crucial role.

In the regression task, our goal is to estimate the conditional mean function $E(y \mid x)$. We have shown that neural networks have the capability to recognize the underlying structure B_0 under the condition that $E(y \mid x) = E(y \mid B_0^{\mathrm{T}}x)$, and as a result, neural networks can be utilized to estimate the central mean space Π_{B_0}. In the context of SDR, a more general scenario arises when $p(y \mid x) = p(y \mid B_0^{\mathrm{T}}x)$, where $p(\cdot \mid \cdot)$ represents the conditional probability density function with a little abuse of notation. The linear space formed by the columns of B_0 is then called the central space. Following the work of Xia [2007], we can apply neural networks to estimate the central space by modifying the loss function.

An important proposition suggests that, under mild conditions, we have

$$E\{K_h(y - y_0) \mid x = x_0\} \to p(y_0 \mid B_0^{\mathrm{T}}x_0), \quad \text{as } h \to 0^+.$$

Here, x_0 and y_0 are fixed points, and $K_h(\cdot)$ is a kernel function with a bandwidth h. Based on this discovery, we define the loss function as

$$\tilde{L}_n(B, f) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\{K_h(y_j - y_i) - f(B^{\mathrm{T}}x_j, y_i)\}^2,$$

where f is a neural network function.

In the study of Xia [2007], the estimation of B_0 utilizes the outer products of gradients method or the minimum average variance estimation method, both of which involve matrix operations such as matrix inversion. These methods are effective for small sample sizes, but the computational burden escalates significantly as more data are gathered. By contrast, the use of deep learning offers a potential solution to circumvent matrix operations. Instead, the algorithm only processes a small batch of data, leading to reduced memory and time costs in cases with large samples.

3 Numerical Experiments

3.1 Simulation

We utilize simulated data sets to demonstrate that (i) the component B as defined in Equation (1) approaches the central mean space Π_{B_0} after optimization, (ii) the performance of neural network-based method in conducting SDR is comparable to classical methods, and in some cases, the neural network-based method can outperform classical methods, particularly when the latent intrinsic dimension d is large. Five commonly used methods, namely SIR, SAVE, PHD, MAVE, and GKDR as mentioned in Section 1, are selected as benchmarks. The SIR, SAVE, and PHD methods are implemented using the `dr` package in R, the MAVE method is implemented using the `MAVE` package in R, and the GKDR algorithm is implemented in MATLAB.

The neural network-based method is implemented in Python using PyTorch. Specifically, we utilize a linear layer without bias term to represent the matrix B (recall that we suppose d is known), which is further appended by a fully-connect feedforward neural network with neurons of $h - h/2 - 1$. Here, we set $h = 64$ when the sample size is less than 1000, and set $h = 128$ when sample size is between 1000 and 2000. Overall, the neural network architecture is $p - d - h - h/2 - 1$. To ensure the adequate optimization, we restart the optimization procedure multiple times and preserve the optimized neural network that yields the lowest empirical loss.

We focus on four scenarios, with two of them attaining small $d = 1, 2$ and the rest presenting $d \geq 3$. The scenarios are as follows.

Setting 1: $y = x_1^4 + \epsilon$ where $x \in \text{Normal}(0, I_p)$ and $\epsilon \in \text{t}_5$.

Setting 2: $y = \log(x_1 + x_1 x_2) + \epsilon$ where $x \in \text{Uniform}([0, 1]^{10})$ and $\epsilon \in \text{Normal}(0, \sigma^2)$.

Setting 3: $y = (1 + \beta_1^{\text{T}} x)^2 \exp(\beta_2^{\text{T}} x) + 5 \cdot 1(\beta_3^{\text{T}} x > 0) + \epsilon$ where $x \in \text{Uniform}([-1, 1]^6)$, $\epsilon \sim \chi_2^2 - 2$ and

$$(\beta_1, \beta_2, \beta_3) = \begin{pmatrix} -2 & -1 & 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 \end{pmatrix}^{\text{T}}.$$

Setting 4:

$$y = \sum_{k=1}^{d} \frac{e^{x_i}}{2 + \sin(\pi x_i)} + \epsilon,$$

where $\epsilon \sim \text{Normal}(0, 0.1^2)$,

$$x \sim \text{Normal}\left(1_{10}, I_{10} - 1_{10} 1_{10}^{\text{T}} / 20\right).$$

In setting 1, we test the combinations of $(n, p) = (100, 10), (200, 30)$ and $(300, 30)$, where n is the sample size. In setting 2, we fix $n = 200, p = 10$ and set $\sigma = 0.1, 0.2, 0.5$. In setting 3, we fix $p = 6$ and test $n = 200, 500, 1000$. In setting 4, we fix $p = 10$ and test $(n, d) = (1000, 4), (1500, 6)$ and $(2000, 8)$. Setting 1, 2, 4 are continuous situations, and setting 3 involves discontinuity.

In terms of evaluation, we calculate the metric $\|\pi_{\hat{B}} - \pi_{B_0}\|_F$ where $\hat{B}$ represents the estimate of B_0. Particularly, in the neural network-based method, $\hat{B} = \mathcal{T}(\hat{f}_n)$. Lower value of this metric indicates better performance. 100 replicates were run for each setting. The results are displayed in Table 1 and Fig. S1 in the Supplementary Material.

Table 1: The results of average and standard deviation of $\|\pi_{\hat{B}} - \pi_{B_0}\|_F$ on 100 replicates across different methods. NN, MAVE, GKDR, SIR, MAVE, PHD represent the neural network-based method, minimum average variance estimation, generalized kernel-based dimension reduction, sliced inverse regression, sliced average variance estimation and pricipal Hessian directions, respectively.

			NN	MAVE	GKDR	SIR	SAVE	PHD
setting 1	$(n, p) = (100, 10)$	mean	0.1345	0.1602	0.3882	0.8972	0.3150	0.6225
		std	0.0637	0.0557	0.1260	0.1864	0.0817	0.1498
	$(n, p) = (200, 30)$	mean	0.2756	0.2138	1.0124	0.9455	0.2978	0.8066
		std	0.2739	0.0508	0.2990	0.1134	0.0487	0.1074
	$(n, p) = (300, 30)$	mean	0.1204	0.1304	0.5416	0.9001	0.3371	0.7086
		std	0.0378	0.0292	0.1657	0.1357	0.0632	0.1082
setting 2	$\sigma = 0.1$	mean	0.2958	0.7304	1.1304	0.6649	0.3188	1.5500
		std	0.1257	0.3121	0.2714	0.1658	0.0761	0.1263
	$\sigma = 0.2$	mean	0.6284	0.8992	1.1546	0.7048	0.3376	1.5256
		std	0.2690	0.3287	0.2368	0.1492	0.0739	0.1380
	$\sigma = 0.5$	mean	1.1965	1.1871	1.2777	0.8299	0.3674	1.5665
		std	0.2126	0.2040	0.2000	0.1691	0.0718	0.1306
setting 3	$n = 200$	mean	0.6388	1.2458	1.7592	1.6693	1.7277	1.7400
		std	0.4184	0.2892	0.1486	0.2351	0.1361	0.2213
	$n = 500$	mean	0.2483	1.0760	1.7524	1.6495	1.6834	1.7214
		std	0.2418	0.3311	0.1246	0.2705	0.2006	0.2280
	$n = 1000$	mean	0.0753	0.9238	0.5542	1.6516	1.6781	1.7369
		std	0.0808	0.3823	0.3713	0.2593	0.2451	0.1910
setting 4	$n = 1000, d = 4$	mean	0.1271	0.2933	0.3682	1.3625	0.4288	0.9602
		std	0.1607	0.0501	0.0793	0.1671	0.0786	0.1775
	$n = 1500, d = 6$	mean	0.1438	0.4151	0.3857	1.5297	0.4673	0.9022
		std	0.0673	0.0762	0.0772	0.1533	0.0817	0.2059
	$n = 2000, d = 8$	mean	0.1399	0.3604	0.3443	1.4100	0.3639	0.7353
		std	0.0547	0.0837	0.0997	0.1626	0.0650	0.1751

In simple cases (setting 1 and 2), the neural network-based method shows similar performance to classical methods, with sliced average variance estimation being the most effective method in setting 2. However, as the complexity

increases, the neural network-based method significantly outperforms other methods. Specifically, in setting 3, as the sample size n increases, the metric $\|\pi_{\hat{B}} - \pi_{B_0}\|_F$ decreases rapidly, indicating that $\Pi_{\hat{B}}$ does indeed converge to Π_{B_0}. According to the results in setting 4, the neural network-based method is capable of handling high-dimensional scenarios.

3.2 Real data

We apply the neural network-based method which involves rank regularization to a real regression task. In practice, the precise intrinsic dimension d is unknown, and it is uncertain if a low-dimensional structure exists. Hence, we use cross validation to determine the appropriate d from the range of values $\{1, \ldots, p\}$. More specifically, the data set is divided into 80% training data and 20% testing data. The optimal value of d is determined through cross validation on the training data. Subsequently, the final model is fit using the selected d on the training data, and the mean squared error on the testing data is evaluated.

In order to reduce randomness, the aforementioned process is repeated 20 times and the resulting testing errors are recorded. Additionally, we conduct a comparative analysis between the neural network-based approach and alternative methods such as vanilla neural network without rank regularization, SIR-based regression, SAVE-based regression, and MAVE-based regression. For the latter three techniques, we initially execute SDR to acquire the embedded data, followed by the utilization of a fully-connected neural network for predictive purpose. The optimal value for d is determined through cross validation.

We utilize a data set of weather records from Seoul, South Korea during the summer months from 2013 to 2017 [Cho et al., 2020], which is available at `https://archive.ics.uci.edu/ml/datasets/Bias+correction+of+numerical+prediction+model+temperature+forecast`. The data set contains 7750 observations with 23 predictors and 2 responses, specifically the next-day maximum air temperature and next-day minimum air temperature. After excluding the variables for station and date, the data set is reduced to 21 predictors, which are further standardized using `StandardScaler` in `sklearn` package.

Table 2: The results of testing errors on 20 replicates across different methods. We report the average and standard deviation of testing errors, along with averaged optimal d determined through cross validation. NN-RR, NN-VN, MAVE, SIR, SAVE represent the neural network-based method with rank regularization, vanilla neural network regression, MAVE-based regression, SIR-based regression, and SAVE-based regression, respectively.

	NN-RR	NN-VN	MAVE	SIR	SAVE
mean	0.6021	0.7739	0.7429	1.3235	0.7724
std	0.0426	0.1158	0.1592	0.1903	0.1608
average best d	19.6	—	19.25	19.6	20.6

It is evident from Table 2 and Fig. S2 in the Supplementary Material that the neural network-based method with rank regularization outperforms other methods, demonstrating the effectiveness of the modification compared to the vanilla neural network, and the sound performance in reducing dimensions compared to other SDR methods. The presence of latent structures is partially supported by the averaged optimal d. It is possible that 19 or 20 combinations of raw predictors may be sufficient, as opposed to the original 21 predictors.

References

Joonseok Lee and Sami Abu-El-Haija. Large-scale content-only video recommendation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 987–995, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021.

Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688, 2019.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Neural network approximation: Three hidden layers are enough. *Neural Networks*, 141:160–173, 2021.

Efang Kong and Yingcun Xia. Variable selection for the single-index model. *Biometrika*, 94(1):217–229, 2007.

Yanyuan Ma and Liping Zhu. Efficient estimation in sufficient dimension reduction. *Annals of statistics*, 41(1):250, 2013.

Cheng Meng, Jun Yu, Jingyi Zhang, Ping Ma, and Wenxuan Zhong. Sufficient dimension reduction for classification using principal optimal transport direction. *Advances in Neural Information Processing Systems*, 33:4015–4028, 2020.

Kuang-Yao Lee, Bing Li, and Francesca Chiaromonte. A general theory for nonlinear sufficient dimension reduction: Formulation and estimation. *The Annals of Statistics*, 41(1):221–249, 2013.

Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86 (414):316–327, 1991.

R Dennis Cook and Sanford Weisberg. Sliced inverse regression for dimension reduction: Comment. *Journal of the American Statistical Association*, 86(414):328–332, 1991.

Ker-Chau Li. On principal hessian directions for data visualization and dimension reduction: Another application of stein's lemma. *Journal of the American Statistical Association*, 87(420):1025–1039, 1992.

Yingcun Xia, Howell Tong, Wai Keung Li, and Li-Xing Zhu. An adaptive estimation of dimension reduction space. In *Exploration of A Nonlinear World: An Appreciation of Howell Tong's Contributions to Statistics*, pages 299–346. World Scientific, 2009.

Chuanlong Xie and Lixing Zhu. Generalized kernel-based inverse regression methods for sufficient dimension reduction. *Computational Statistics & Data Analysis*, 150:106995, 2020.

Yuhong Yang and Andrew Barron. Information-theoretic determination of minimax rates of convergence. *Annals of Statistics*, pages 1564–1599, 1999.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation characterized by number of neurons. *arXiv preprint arXiv:1906.05497*, 2019.

Ahmed Abdeljawad and Philipp Grohs. Approximations with deep neural networks in sobolev time-space. *Analysis and Applications*, 20(03):499–541, 2022.

Guohao Shen, Yuling Jiao, Yuanyuan Lin, and Jian Huang. Approximation with cnns in sobolev space: with applications to classification. *Advances in Neural Information Processing Systems*, 35:2876–2888, 2022.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Banghua Zhu, Jiantao Jiao, and Michael I Jordan. Robust estimation for non-parametric families via generative adversarial networks. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 1100–1105. IEEE, 2022.

Yingcun Xia. A constructive approach to the estimation of dimension reduction directions. *The Annals of Statistics*, 35(6):2654 – 2690, 2007. doi:10.1214/009053607000000352. URL `https://doi.org/10.1214/009053607000000352`.

Dongjin Cho, Cheolhee Yoo, Jungho Im, and Dong-Hyun Cha. Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas. *Earth and Space Science*, 7(4):e2019EA000740, 2020.