applied sciences

MDPI

Article

Variable Selection Using Deep Variational Information Bottleneck with Drop-Out-One Loss

Junlong Pan [1], Weifu Li [1,2], Liyuan Liu [1], Kang Jia [3], Tong Liu [3] and Fen Chen [4,5,*]

[1] College of Science, Huazhong Agricultural University, Wuhan 430070, China
[2] Engineering Research Center of Intelligent Technology for Agriculture, Ministry of Education, Wuhan 430070, China
[3] Beijing Jiyun Intelligent Technology Co., Ltd., Beijing 100096, China
[4] School of Finance, Hubei University of Economics, Wuhan 430205, China
[5] Hubei Financial Development and Financial Security Research Center, Hubei University of Economics, Wuhan 430205, China
* Correspondence: fenfen_chen@hbue.edu.cn

Abstract: The information bottleneck (IB) model aims to find the optimal representations of input variables with respect to the response variable. While it has been widely used in the machine-learning community, research from the perspective of the information-theoretic method has been rarely reported regarding variable selection. In this paper, we investigate DNNs for variable selection through an information-theoretic lens. To be specific, we first state the rationality of variable selection with IB and then propose a new statistic to measure the variable importance. On this basis, a new algorithm based on a deep variational information bottleneck is developed to calculate the statistic, in which we consider the Gaussian distribution and the exponential distribution to estimate the Kullback–Leibler divergence. Empirical evaluations on simulated and real-world data show that the proposed method performs better than classical variable-selection methods. This confirms the feasibility of the variable selection from the perspective of IB.

Keywords: information bottleneck; drop-out-one loss; variable selection; deep learning

Citation: Pan, J.; Li, W.; Liu, L.; Jia, K.; Liu, T.; Chen, F. Variable Selection Using Deep Variational Information Bottleneck with Drop-Out-One Loss. *Appl. Sci.* **2023**, *13*, 3008. https://doi.org/10.3390/app13053008

Academic Editors: José Salvador Sánchez Garreta and Jan Egger

Received: 2 December 2022
Revised: 21 February 2023
Accepted: 22 February 2023
Published: 26 February 2023

1. Introduction

In many problems of interest, researchers always aim to understand which part of the input variables (features) explain the response variable (output), such as in biomedicine and financial engineering. The variable selection, in addition to reducing the computational complexity, also provides insight into the complex relationship between the inputs and response. Hence, its value is only increasing. To address this problem, many traditional machine-learning models have been studied for decades. For linear models, numerous variable-selection algorithms have been proposed, among which, the most well-known ones include Lasso [1], GroupLasso [2] and Smoothly Clipped Absolute Deviation Penalty (SCAD) [3]. To be specific, Lasso [1] realized the variable selection by embedding the L_1 regularizer into the linear model. GroupLASSO [2] considered the problem of selecting grouped variables for accurate prediction in regression. SCAD [3] extended the thresholding penalty function and satisfied the mathematical conditions for unbiasedness, sparsity and continuity.

In these models, the variable importance is measured by the coefficient of each variable. As an extension of the linear model, the additive models have been proposed to deal with nonlinear variable selection problems, such as Sparse Additive Models (SpAM) [4,5] and GroupSpAM [6]. They, respectively, extended the Lasso and GroupLasso with additive nonparametric regression. Instead of selecting the variable directly, Mukherjee et al. [7] proposed to learn the gradient at each point in the instance space. On this basis, several variants of the gradient-learning (GL) model have been devoted to developing alternatives

for individual purposes, such as sparse [8] and robust [9,10] gradients. Ye and Xie [8] proposed to learn sparse gradients for variable selection and dimension reduction. To resist the outliers or heavy-tailed noise, Feng et al. [9] proposed robust loss, and Liu et al. [10] adopted tilted loss to learn the gradient function. In these models, the variable importance is measured by the corresponding function. To control the false discovery rate (FDR) of selected variables, Barber et al. [11] introduced a new variable-selection procedure called knockoff filter and achieved exact FDR control in the statistical linear model. On this basis, Candes et al. proposed model-X knockoffs [12] and provided valid inference in a general setting that the conditional distribution of the response is arbitrary and completely unknown. In these models, the variable importance is measured by the difference of scores between the original variable and its knockoff variable. In addition, there also exist several methods proposed from the perspective of information gain, such as decision tree and random forest (RF) [13].

Despite the aforementioned studies having achieved satisfactory performance on variable selection, they are usually limited by the fitting accuracy due to the arrival of the era of big data. In contrast, deep neural networks (DNNs) have achieved great success on a wide variety of tasks due to their strong representation and approximation ability. However, they are largely treated as black-box tools with little interpretability. Therefore, various attempts have been made to uncover the mysteries of neural networks, among which a large class of methods is focused on designing new DNNs to uncover the key features. For example, Lemhadri et al. [14] extended the Lasso and proposed a corresponding DNN framework, LassoNet. It achieved feature sparsity by allowing a feature to participate in a hidden unit only if its linear representative is active. Feng et al. [15] fit the DNN using the sparse group lasso penalty on the first-layer input weights, which resulted in a DNN that only used a subset of the original features. Agarwal et al. [16] likewise proposed a neural additive model (NAM), which effectively combined the expressivity of DNN with the inherent intelligibility of generalized additive models. This model learned a linear combination of neural networks that each attended to a single input feature. Lu et al. [17] designed a new DNN architecture (called DeepPINK) with filter-integrated knockoffs, which improved the interpretability and reproducibility of the DNN by incorporating the idea of feature selection with a controlled error rate. On the other hand, recent works also focused on understanding the learning dynamics of DNNs through an information-theoretic lens. An information-theoretic paradigm for deep learning based on the IB framework has caused a great deal of concern as well. Schwartz and Tishby [18] suggested that the goal of the DNN was to optimize the information bottleneck (IB) [19] tradeoff between compression and prediction, successively, for each layer. They claimed that there existed two different and distinct phases: the compression phase and fitting phase in stochastic gradient descent (SGD) optimization. In addition, it was found that most of the training epochs in standard deep learning were spent on compression of the input to efficient representation and not on fitting the training labels.

The abstract viewpoint of IB also helps to better understand the field of representation learning, which is an active research area in machine learning that focuses on identifying and uncovering potential explanatory factors [20]. It has attracted increasing attention in many applications, such as out-of-distribution generalization [21], sparse code extraction [22], semi-supervised classification [23] and geometric clustering [24,25]. However, research from the perspective of IB has rarely been reported for variable selection. In principle, the compression and fitting phases in IB are essential to eliminate useless information. This motivates us to use IB for variable selection. To this end, we investigate the deep variational information bottleneck (DeepVIB) [26] combined with Drop-Out-One loss [27] to realize the variable selection. To better highlight the novelty of the proposed method, we summarize the properties of different variable-selection methods in Table 1, where these methods are categorized as coefficient-based (the variable importance is measured by a coefficient, such as Lasso), function-based (the variable importance is measured by a function, such as additive models and GL), IB-based (through an IB lens, such as the

proposed method) and DNN-based (through DNNs, such as Lasso Net and the proposed method). The main contributions of this paper are summarized as below.

- **Originality.** To our knowledge, we are the first to investigate DNNs for variable selection through an IB lens.
- **New algorithm.** A new algorithm based on the DeepVIB and Drop-Out-One loss is designed to realize the variable selection, in which we provide the estimation of the Kullback–Leibler divergence by considering the cases of Gaussian distribution and exponential distribution. The source code is publicly accessed on 21 February 2023 at Github (https://github.com/ZeonlungPun/vib_drop_out_one_loss/tree/main).
- **Empirical performance.** Empirical evaluations on simulated and real-world data show that our method performs better than classical variable-selection methods. This confirms the feasibility of variable selection from the perspective of IB.

Table 1. Properties of different methods ($\surd$ means satisfying the given information, and $\times$ means not).

Methods	Lasso	SpAM	Knockoff	GL	Lasso Net	NAM	DeepPINK	Ours
Coefficient-based	$\surd$	$\times$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$
Function-based	$\times$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$	$\times$
IB-based	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$
DNN-based	$\times$	$\times$	$\times$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$

The rest of this paper is organized as follows. Section 2 recalls the background of IB and VIB. Section 3 states the rationality of variable selection with IB and presents its computing algorithm with DeepVIB. Section 4 reports the experimental analysis of our approach. Finally, Section 5 presents our conclusions.

2. Preliminaries

In this section, we recall the necessary definitions and notations involved in the IB and variational information bottleneck.

2.1. Information Bottleneck

The IB [19] can be viewed as a rate-distortion problem [28], with a distortion function $KL(p(y|x)|p(y|z))$ that measures how well Y is predicted from a compressed representation Z compared to its direct prediction from X. This means that specifying the features of X is required. The problem can be also formalized as finding a short code Z for X that preserves the maximum information about Y. For example, we hope to extract the most useful information in the speech sounds about the words spoken. The IB principle is formulated as the maximization

$$I(Z,Y) - \beta I(X,Z), \tag{1}$$

where the mutual information $I(X,Z)$ measures the dependence between random variables X and Z as follows:

$$I(X,Z) = \iint p(x,z) \log \frac{p(x,z)}{p(x)p(z)} dxdz. \tag{2}$$

Intuitively, $I(Z,Y)$ encourages Z to predict Y, and $I(X,Z)$ forces Z to "forget" X. The hyperparameter $\beta > 0$ controls the trade-off between the compression (complexity) and fitting. From another perspective, β can be also used to control the bias–variance trade-off in cases where the distribution function is not known and the mutual information is only estimated from a finite number of samples [29]. The complexity here is measured by the minimum description length (or rate) at which the observation is compressed.

When $\beta \to +\infty$, (1) leads to a trivial representation Z that is independent of X, and it recovers a maximum likelihood objective while $\beta \to 0$ [30]. In summary, IB seeks the right

balance between data fit and generalization by using the mutual information as both a cost function and a regularizer, which is vital to variable selection [20].

2.2. Deep Variational Information Bottleneck

Although the IB principle is appealing, the difficulties of computing mutual information have hindered its application to DNNs. A variational inference to IB (called VIB) was proposed to overcome this problem by estimating the Kullback–Leibler divergence [26,31], in which $q(y|z)$ and $r(z)$ are variational approximations to the conditional distribution $p(y|z)$ and marginal distribution $p(z)$. VIB provided a lower bound of (1) as follows

$$L = \iiint p(x)p(y|x)p(z|x)\log q(y|z)dxdydz - \beta \iint p(x)p(z|x)\log \frac{p(z|x)}{r(z)}dxdz. \quad (3)$$

Then, the objective is changed to the maximization of L. Contrary to the original IB, VIB obviates the need for full prior knowledge of the joint distribution $p(x, y)$. Instead, it works based on a finite sample set, which is easier to estimate.

3. Proposed Method

In this section, we first introduce the rationality of variable selection with IB and then provide the computational algorithm DeepVIB. The network structure of DeepVIB is presented in Figure 1.

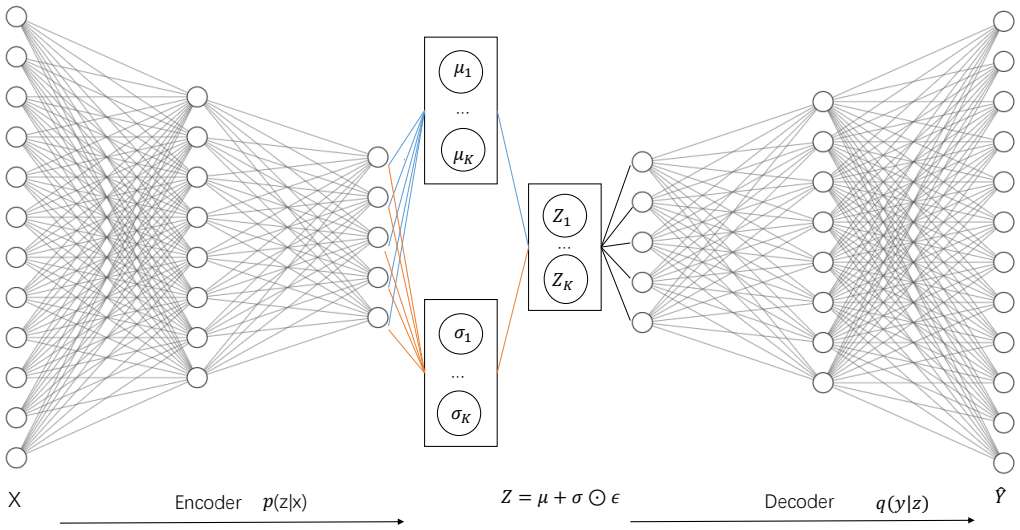

Figure 1. The network structure of DeepVIB. The encoder $p(z|x)$ learns a representation Z from X, in which the reparameterization trick $z_i = \mu_i + \sigma_i \odot \epsilon$ is implemented, and $\odot$ means the element-wise product. The decoder $q(y|z)$ uses Z to predict Y.

3.1. Variable Selection with IB

Let $\mathcal{X} \subset \mathbb{R}^p$ be a compact input space and $\mathcal{Y} \subset \mathbb{R}$ be an output space. Denote (X, Y) as the pair of explanatory and response variables taking values in $\mathcal{X} \times \mathcal{Y}$. Assume

$$Y = f^*(X) + \epsilon, \quad (4)$$

where $f^* : \mathcal{X} \to \mathcal{Y}$ is an intrinsic target function and ϵ is a random noise satisfying some certain conditions, e.g., the zero-mean noise assumption or the zero-mode noise condition. Let X_j be the j-th feature of X. We also assume that $f^*(X)$ depends on X only through $\{X_j : j \in T\}$, where T is an index set of relevant variables and $|T| < p$ is the cardinality of T. Let $T^c = \{1, 2, \cdots, p\} - T$ be an index set of irrelevant variables.

In another perspective [18], the IB objective is looking for a sufficient statistic Z_0, which is a map of X, and extracts all the information that X has on Y, i.e., satisfying

$I(Z_0, Y) = I(X, Y)$. Since the exact sufficient statistics only exist for very special distributions, the IB framework only provides for finding approximate sufficient statistics. That means $I(Z_0, Y)$ captures as much as possible of $I(X, Y)$—not necessarily all of it [18]. Let Z_0 be the sufficient statistic, and this means

$$Z_0 = \arg\max_Z \{I(Z, Y) - \beta I(X, Z)\}.$$

Given a Markov chain $Y \to X \to Z_0$ [19], the elementary data processing inequality ensures

$$I(Z_0, X) \geq I(Z_0, Y).$$

Combining the above inequalities, it is trivial to find that

$$\{I(Z_0, Y) - \beta I(Z_0, X)\} \leq (1 - \beta) I(X, Y),\ 0 < \beta < 1. \tag{5}$$

Inequality (5) provides the intuition that the IB objective can be approximated by $I(X, Y)$. Denote X_{-j} as the rest of $p - 1$ variables after removing the j-th variable. Note that the $I(X_j, Y)$ and conditional mutual information $I(X_j, Y|X_{-j})$ can be assumed as 0 for any $j \in T^c$; the equation $I(X, Y) = I(X_{-j}, Y) + I(X_j, Y|X_{-j})$ informs us $I(X, Y) = I(X_{-j}, Y)$. Similarly, the assumptions $I(X_k, Y) > 0$ and $I(X_k, Y|X_{-k}) > 0$ imply that $I(X, Y) > I(X_{-k}, Y)$ for any $k \in T$. This means that $I(X, Y)$ will decrease after removing the relevant variable, while it will not change for irrelevant variables. Together with inequality (5), a natural idea is that the IB objective will decrease when dropping the critical variable. Therefore, we propose a new statistic to measure the importance of the j-th variable

$$S_j = \max_Z \{I(Z, Y) - \beta I(X, Z)\} - \max_Z \{I(Z, Y) - \beta I(X_{-j}, Z)\}, j = 1, 2, \cdots, p. \tag{6}$$

A small S_j means the j-th variable X_j tends to be irrelevant.

Remark 1. *Intuitively, $I(X, Z)$ measures the useful information of the inputs extracted by the encoder. It is inversely related with the compression, i.e., the greater the compression, the smaller the $I(X, Z)$. Combined with $I(Z, Y)$ denoting predictive power, the IB can be seen as extracting the key information for predicting Y from X. Therefore, utilizing the "information" in the IB to identify the irrelevant variables is sensible.*

3.2. Variable Selection with DeepVIB

To calculate the statistic S_j, we approximate the lower bound in (3) using the empirical data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, which is independent and identically distributed (i.i.d.) drawn from (4). We first place all input variables into DeepVIB for training. After convergence, the resulting empirical VIB w_0 is calculated as follows:

$$w_0 = \frac{1}{N} \sum_{i=1}^{N} \left[\int p(z|\mathbf{x}_i) \log q(y_i|z) - \beta p(z|\mathbf{x}_i) \log \frac{p(z|\mathbf{x}_i)}{r(z)} dz \right]. \tag{7}$$

Here, the $p(z|\mathbf{x}_i)$ and $q(y_i|z)$ in the first term can be easily obtained by the encoder and decoder of DeepVIB, respectively. As for the second term, the KL-divergence $KL(p(z|\mathbf{x}_i)|r(z))$ are determined by the distribution of the latent variable $r(z)$ and its posterior $p(z|x)$. In this paper, we consider the Gaussian distribution and the exponential distribution. The detailed proofs of estimating the $KL(p(z|\mathbf{x}_i)|r(z))$ are provided in the Appendix A.

Denoting $D_j = \{(\mathbf{x}_{i,-j}, y_i)\}_{i=1}^N$ as the data after removing the j-th variable, we similarly retrain the DeepVIB model and calculate the empirical VIB w_j as follows:

$$w_j = \frac{1}{N}\sum_{i=1}^{N}\left[\int p(z|\mathbf{x}_{i,-j})\log q(y|z) - \beta p(z|\mathbf{x}_{i,-j})\log\frac{p(z|\mathbf{x}_{i,-j})}{r(z)}dz\right],\ j = 1, 2, \cdots, p.$$

During the training of DNN, there are several reasons, such as inconsistent initialization of the weight parameters and randomly chosen samples used in each batch, that can cause fluctuations over epochs. To eliminate the randomness and make the statistics stable, a common solution is to calculate the w_0 and w_j in the later epochs. We denote the corresponding empirical VIB values at the k-th epoch as w_0^k and w_j^k, the final VIB values are averaged by the last L (called the "reserved length") epochs, i.e.,

$$\overline{w_0} = \frac{1}{L}\sum_{k=E-L+1}^{E} w_0^k,\ \ \overline{w_j} = \frac{1}{L}\sum_{k=E-L+1}^{E} w_j^k,$$

where E means the number of epochs. Then, the empirical statistics approximating (6) are calculated by

$$\hat{S}_j = \overline{w_0} - \overline{w_j},\ j = 1, 2, \cdots, p.$$

The detailed training steps of the proposed method are also summarized in Algorithm 1.

Algorithm 1: Calculation algorithm of DeepVIB

Input : Dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, parameter β, the epochs E, reserved length L, a sequence of scores sets $W_j = \varnothing, j = 1, 2, \cdots, p$

Output: p empirical statistics

for *k in* $\{1, 2, \cdots, E\}$ **do**

  Calculate the empirical VIB with dataset D:

  $w_0^k = \frac{1}{N}\sum_{i=1}^{N}\left[\int p(z|\mathbf{x}_i)\log q(y_i|z) - \beta p(z|\mathbf{x}_i)\log\frac{p(z|\mathbf{x}_i)}{r(z)}dz\right]$

  Store w_0^k in this epoch: $W_0 = W_0 \cup w_0^k$

end

Calculate the benchmark VIB: $\overline{w_0} = \frac{1}{L}\sum_{k=E-L+1}^{E} w_0^k$

for *each j in* $\{1, 2, \cdots, p\}$ **do**

  Construct the dataset $D_j = \{(\mathbf{x}_{i,-j}, y_i)\}_{i=1}^N$ by dropping the j-th variable

  for *k in* $\{1, 2, \cdots, E\}$ **do**

    Calculate the empirical VIB with dataset D_j:

    $w_j^k = \frac{1}{N}\sum_{i=1}^{N}\left[\int p(z|\mathbf{x}_{i,-j})\log q(y|z) - \beta p(z|\mathbf{x}_{i,-j})\log\frac{p(z|\mathbf{x}_{i,-j})}{r(z)}dz\right]$

    Store w_j^k in this epoch: $W_j = W_j \cup w_j^k$

  end

end

for *each j in* $\{1, 2, \cdots, p\}$ **do**

  Calculate the average VIB in the L later epochs: $\overline{w_j} = \frac{1}{L}\sum_{k=E-L+1}^{E} w_j^k$

  Score of importance for the j-th feature: $\hat{S}_j = \overline{w_0} - \overline{w_j}$

end

4. Experimental Analysis

In this section, we perform several experiments on both synthetic and real datasets in the nonlinear variable selection problem. Our purpose here is to verify the effectiveness of the proposed method and to give some empirical comparisons with other classical variable-selection algorithms, including Lasso [1], Elastic Net [32], SCAD [3], RF [13], AdaBoost [33], GL [7], SpAM [4], Knockoff [11] and Lasso Net [14].

4.1. Simulated Experiments

For the simulated data, we consider the ϵ in (4) following the Gaussian distribution $\mathcal{N}(0,1)$ scaled by 0.05 and the same target functions f^* as adopted in previous papers [9,34]. Among them, f^* is non-additive in Examples 1, 2 and 4 and is additive in Example 3.

Example 1. *The non-additive model (from [9])*

$$f^*(\mathbf{x}) = x_1x_2 + (2x_3 - 0.5)^2 + x_4 + x_5,$$

where $X = [X_1, X_2, \cdots, X_{10}]$ *are i.i.d. and drawn from the uniform distribution on* $[0,1]$ *with* $N = 100$.

Example 2. *The non-additive model (from [9])*

$$f^*(\mathbf{x}) = \frac{x_1}{0.5 + (1.5 + x_2)^2}.$$

In this situation, $X = [X_1, X_2, \cdots, X_{10}]$ *are drawn from* $\mathcal{N}(0, \Sigma)$ *with* $N = 100$, *where* Σ *denotes the covariance matrix with the* (i,j)*th entry given by* $\Sigma_{i,j} = 0.5^{|i-j|}$.

Example 3. *The additive model (from [9])*

$$f^*(\mathbf{x}) = -2\tan(0.5x_1) + x_2 + x_3 + \exp(-x_4).$$

Here, we consider three cases $(N,p) = (100,10), (100,20)$ *and* $(200,50)$ *and both the correlated and uncorrelated variables, i.e.,* $X \sim \mathcal{N}(0, I)$ *and* $X \sim \mathcal{N}(0, \Sigma)$.

Example 4. *The non-additive model (from [34])*

$$f^*(\mathbf{x}) = (2x_1 - 1)(2x_2 - 1).$$

In this case, $x_{ij} = \frac{w_{ij} + \eta u_i}{1+\eta}$, *where* w_{ij} *and* u_i *are independently sampled from* $\mathcal{N}(0,1)$. *When* $\eta = 0$, *the input variables* x_1 *and* x_2 *are independent, whereas* $\eta \neq 0$ *means they are correlated. We also consider three cases* $(N,p) = (100,10), (100,20)$ *and* $(200,50)$ *with different choices* $\eta = 0, 0.2, 0.5$.

As mentioned above, three parameters, including E, L and β, are involved in our algorithm when performing the variable selection. In the experiment, we generally set E ranging over $[1200, 1500]$, L ranging over $[100, 500]$ and β ranging over $[0.015, 0.8]$. For the compared algorithms, the corresponding parameters are tuned near the default values. The final values are chosen by cross validation. Finally, each algorithm will give each variable a statistic that measures the importance.

For simplicity, we assume that the number of relevant variables p_{true} is known, and the first p_{true} variables with largest statistics are selected. Three common metrics are adopted to measure the performance of each algorithm, including TP (the average number of selected truly relevant variables), FP (the average number of selected truly irrelevant variables) and STP (the standard deviation of TP). Generally, a higher TP with a lower FP and STP indicates a better variable-selection algorithm.

We repeated the experiments 20 times with the observation set generated in each circumstance. The average variable selection results of each algorithm are presented in Tables 2–4. The optimal results in each circumstance are marked in bold, where DeepVIB (G) means that $p(z)$ obeys the Gaussian distribution, while DeepVIB (E) indicates the exponential distribution. The above-reported results tell us the following.

Table 2. The averaged performance of variable selection (Examples 1 and 2).

Models	Example 1			Example 2		
	TP	**FP**	**STP**	**TP**	**FP**	**STP**
Lasso [1]	4.35	0.65	0.85	1.30	0.70	0.45
Elastic Net [32]	4.55	0.45	0.80	1.50	0.50	0.50
SCAD [3]	4.90	0.10	0.30	1.30	0.70	0.45
RF [13]	4.50	0.50	0.74	**2.00**	**0.00**	**0.00**
AdaBoost [33]	4.75	0.25	0.62	1.90	0.10	0.30
SpAM [4]	**5.00**	**0.00**	**0.00**	1.35	0.65	0.47
GL [7]	4.80	0.20	0.50	1.20	0.800	0.40
Knockoff [11]	4.90	0.10	0.40	1.60	0.40	0.48
Lasso Net [14]	**5.00**	**0.00**	**0.00**	1.85	0.15	0.35
DeepVIB (G)	**5.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
DeepVIB (E)	**5.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**

Table 3. The averaged performance of variable selection (Example 3).

(N, p)	Models	$\Sigma = I$			$\Sigma_{i,j} = 0.5^{\lvert i-j \rvert}$		
		TP	**FP**	**STP**	**TP**	**FP**	**STP**
$(100, 10)$	Lasso [1]	3.50	0.50	1.11	2.50	1.50	1.16
	Elastic Net [32]	3.70	0.30	0.78	3.50	0.50	1.07
	SCAD [3]	3.85	0.15	0.47	3.50	0.50	1.02
	RF [13]	3.90	0.10	0.30	3.75	0.25	0.50
	AdaBoost [33]	3.65	0.35	0.65	3.35	0.65	0.96
	SpAM [4]	**4.00**	**0.00**	**0.00**	**4.00**	**0.00**	**0.00**
	GL [7]	3.45	0.55	0.97	3.20	0.80	1.02
	Knockoff [11]	3.70	0.30	0.95	3.55	0.45	1.11
	Lasso Net [14]	**4.00**	**0.00**	**0.00**	**4.00**	**0.00**	**0.00**
	DeepVIB (G)	**4.00**	**0.00**	**0.00**	**4.00**	**0.00**	**0.00**
	DeepVIB (E)	**4.00**	**0.00**	**0.00**	**4.00**	**0.00**	**0.00**
$(100, 20)$	Lasso [1]	2.65	1.35	1.68	2.10	1.90	1.33
	Elastic Net [32]	3.5	0.5	0.97	3.3	0.7	1.14
	SCAD [3]	3.55	0.45	1.07	3.5	0.5	0.97
	RF [13]	3.65	0.35	0.72	3.65	0.35	0.72
	AdaBoost [33]	3.1	0.9	0.88	2.9	1.1	1.09
	SpAM [4]	**4.00**	**0.00**	**0.00**	**4.00**	**0.00**	**0.00**
	GL [7]	3.40	0.60	0.91	2.85	1.15	1.23
	Knockoff [11]	3.35	0.65	1.23	3.30	0.70	1.42
	Lasso Net [14]	3.85	0.15	0.65	3.80	0.87	0.20
	DeepVIB (G)	**4.00**	**0.00**	**0.00**	**4.00**	**0.00**	**0.00**
	DeepVIB (E)	**4.00**	**0.00**	**0.00**	**4.00**	**0.00**	**0.00**
$(200, 50)$	Lasso [1]	2.00	2.00	1.76	0.85	3.15	0.85
	Elastic Net [32]	3.35	0.65	1.31	3.10	0.90	1.09
	SCAD [3]	3.35	0.65	1.42	3.70	0.30	0.90
	RF [13]	3.55	0.45	0.86	3.40	0.60	0.73
	AdaBoost [33]	2.8	1.20	0.40	2.85	1.15	0.90
	SpAM [4]	**4.00**	**0.00**	**0.00**	**4.00**	**0.00**	**0.00**
	GL [7]	3.25	0.75	1.08	2.65	1.35	1.11
	Knockoff [11]	3.00	1.00	1.34	3.05	0.95	1.65
	Lasso Net [14]	3.75	0.25	0.88	3.65	0.35	0.96
	DeepVIB (G)	**4.00**	**0.00**	**0.00**	3.75	0.25	0.53
	DeepVIB (E)	3.80	0.20	0.50	2.00	2.00	0.67

Table 4. The averaged performance of variable selection (Example 4).

		$\eta = 0$			$\eta = 0.2$			$\eta = 0.5$		
(N, p)	**Models**	**TP**	**FP**	**STP**	**TP**	**FP**	**STP**	**TP**	**FP**	**STP**
$(100, 10)$	Lasso [1]	1.80	0.20	0.50	1.85	0.15	0.35	1.85	0.15	0.47
	Elastic Net [32]	1.95	0.05	0.21	1.90	0.10	0.30	1.90	0.10	0.30
	SCAD [3]	**2.00**	**0.00**	**0.00**	1.95	0.05	0.21	1.95	0.05	0.21
	RF [13]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
	AdaBoost [33]	1.95	0.05	0.20	1.90	0.10	0.30	1.90	0.10	0.30
	SpAM [4]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
	GL [7]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	1.95	0.05	0.20
	Knockoff [11]	1.90	0.10	0.43	1.95	0.05	0.20	1.90	0.10	0.43
	Lasso Net [14]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
	DeepVIB (G)	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
	DeepVIB (E)	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
$(100, 20)$	Lasso [1]	1.80	0.20	0.40	1.75	0.25	0.62	1.15	0.85	0.96
	Elastic Net [32]	1.90	0.10	0.43	1.90	0.10	0.43	1.90	0.10	0.43
	SCAD [3]	1.90	0.10	0.30	1.90	0.10	0.43	1.80	0.20	0.50
	RF [13]	2.00	0.00	0.00	1.95	0.05	0.21	1.85	0.15	0.47
	AdaBoost [33]	1.95	0.05	0.21	1.85	0.15	0.47	1.75	0.25	0.62
	SpAM [4]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
	GL [7]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
	Knockoff [11]	1.85	0.15	0.35	1.85	0.15	0.47	1.90	0.10	0.43
	Lasso Net [14]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
	DeepVIB (G)	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
	DeepVIB (E)	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
$(200, 50)$	Lasso [1]	1.50	0.50	0.67	1.25	0.75	0.88	0.80	1.20	0.92
	Elastic Net [32]	0.95	1.05	0.66	1.10	0.90	0.62	1.05	0.95	0.49
	SCAD [3]	1.90	0.10	0.30	1.85	0.15	0.47	1.75	0.25	0.62
	RF [13]	1.85	0.15	0.35	1.85	0.15	0.47	1.85	0.15	0.47
	AdaBoost [33]	1.95	0.05	0.21	1.85	0.15	0.47	1.80	0.20	0.60
	SpAM [4]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
	GL [7]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	1.95	0.05	0.21
	Knockoff [11]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	1.95	0.05	0.21
	Lasso Net [14]	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**	**2.00**	**0.00**	**0.00**
	DeepVIB (G)	**2.00**	**0.00**	**0.00**	1.90	0.10	0.30	1.90	0.10	0.30
	DeepVIB (E)	1.85	0.15	0.47	1.80	0.20	0.60	1.40	0.60	0.50

(1) Table 2 shows that the TPs of the proposed DeepVIB (G) and DeepVIB (E) in Examples 1 and 2 always equal the p_{true}, while the FPs and STPs are always 0. This means that the proposed method can perfectly select all the relevant variables in each experiment. For comparison, other algorithms always select the irrelevant variables in Example 1 or 2. This supports the superiority of DeepVIB for variable selection.

(2) Tables 3 and 4 show the performance of all variable-selection algorithms in Examples 3 and 4. Generally, the performance degrades when p increases or when the input variables become correlated. This is consistent with the previous phenomenon in Examples 1 and 2. In particular, the proposed DeepVIB (G) and DeepVIB (E) can select all the relevant variables in the cases $(N, p) = (100, 10)$ and $(100, 20)$, which is better performance compared with the other variable-selection algorithms.

(3) From the empirical results in Tables 3 and 4, we also note that DeepVIB (G) provides consistently larger TPs compared with DeepVIB (E) in the case $(N, p) = (200, 50)$. This indicates that DeepVIB can achieve better variable selection results when $p(z)$ obeys the Gaussian distribution. An empirical reason may be that the Gaussian distribution can generate a larger range on Z compared to the exponential distribution, providing the neural

network more choices when mapping Y from Z. Hence, we prefer the Gaussian distribution when estimating the $KL(p(z|x)|r(z))$ in DeepVIB.

(4) As for other classical variable-selection algorithms, the experimental results rely on their properties. Methods, such as Elastic Net and SCAD, are proposed for linear models, and their performance is generally inferior to methods for nonlinear models, such as Lasso Net and RF, in nonlinear examples. We also note that Lasso Net performed perfectly in Example 4 but not in Example 3. The reason may be that the input data are drawn from different distributions and that the target function in Example 3 is more complex than the one in Example 4. In addition, we can see that SpAM performs well except in Example 2. An underlying reason is that SpAM was originally proposed with an additional assumption on the target function for sparse variable selection.

In fact, Example 2 contains x_2 in the denominator, which is not coincident with the additive assumption. In contrast, Examples 1 and 4 can be more like an additive model except for the interaction term between x_1 and x_2. Different from SpAM, Lasso Net was proposed based on DNNs and, thus, does not rely on a specific assumption of the target function. The strong representation and approximation ability of DNNs ensure that Lasso Net can achieve satisfactory performance on the four examples. This also supports our motivation to investigate DeepVIB for variable selection.

To highlight the influence of parameter β in the proposed method, we define a new static by empirical statistics:

$$gap = \min_{k \in T} \hat{S}_k - \max_{j \in T^c} \hat{S}_j,$$

where a large gap means better selection results. Performing experiments on Example 2, the gap vs. β is shown in Figure 2. The experimental results show that the gap achieves the maximum when $\beta = 0.5$ in this situation. It performs better than a small β close to zero (in cases of total compression).

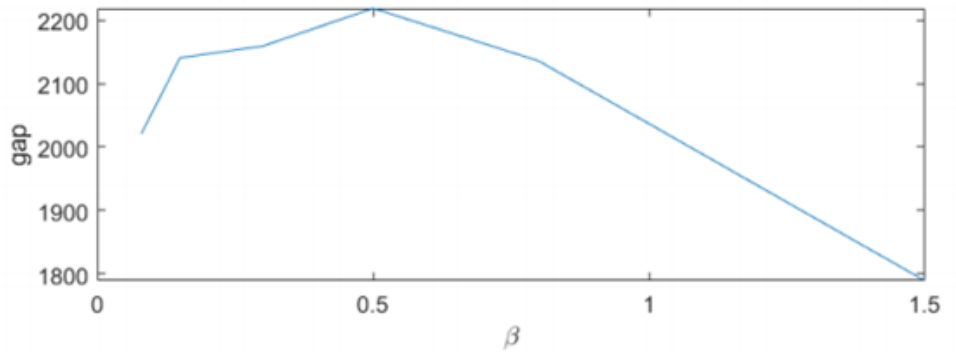

Figure 2. The gap vs. β in Example 2.

4.2. Real Experiments

We also conducted experiments on three real-world datasets for regression problems. These data were selected from the UCI machine-learning datasets, including Boston Housing Price (BHP), California Housing Price (CHP) and Diabetes.

To be specific, the BHP dataset contains 506 observations with 13 input variables, including the per capita crime rate by town (CRIM), proportion of residential land zoned for lots (ZN), proportion of non-retail business acres per town (INDUS), Charles River dummy variable (CHAS), nitric oxides concentration (NOX), average number of rooms per dwelling (RM), AGE, weighted distances to five Boston employment centers (DIS), index of accessibility to radial highways (RAD), TAX, pupil–teacher ratio by town (PTRATIO), the proportion of black people by town (B) and lower status of the population (LSTAT). These variables are considered as relevant, and 13 irrelevant variables are generated from the distribution $\mathcal{N}(0, 1)$, which are denoted as *irre1*, . . . , *irre13*.

The dataset CHP consists of 20,640 instances with eight variables, including the median household income (MedInc), median age of home (HouseAge), average number of rooms

(AveRooms), average number of bedrooms (AveBedrms), population, average occupancy (AveOccup), latitude and longitude. Similarly, we generated 22 additional irrelevant variables for the same operation. The irrelevant variables, denoted as *irre1*, . . . , *irre22*, are added analogously. For the dataset Diabetes, 442 instances were collected with 10 attributes, including age, sex, BMI, BP, s1, s2, s3, s4, s5 and s6. An additional 20 irrelevant variables, denoted as *irre1*, . . . , *irre20*, are generated analogously.

Since the truly informative variables are unknown, Table 5 presents the two most-relevant variables and two most-irrelevant variables in BHP, CHP and Diabetes as selected by each algorithm. From Table 5, we can see that the proposed algorithm considered the pseudo-variables as irrelevant and the real variables as relevant in the three datasets. The selected relevant variables generally coincide with the results selected by other algorithms.

In contrast, the Knockoff and SCAD algorithms incorrectly identified one real variable as irrelevant in the dataset BHP. The Lasso and Elastic Net algorithms identified one real variable, while the Knockoff algorithm identified two real variables as irrelevant in the dataset CHP. The SCAD, RF, AdaBoost and SpAM algorithms incorrectly identified one real variable as irrelevant in the Diabetes dataset. This also supports the superiority of DeepVIB for variable selection.

Table 5. Variable selection results on real data.

Dataset	Method	Most Relevant Variables	Most Irrelevant Variables
BHP	Lasso [1]	LSTAT, DIS	*irre*7, *irre*8
	Elastic Net [32]	CRIM, RM	*irre*9, *irre*11
	SCAD [3]	CRIM, RM	AGE, *irre*3
	RF [13]	CRIM, RM	*irre*6 , *irre*9
	AdaBoost [33]	CRIM, RM	*irre*6, *irre*9
	GL [7]	CRIM, ZN	*irre*9, *irre*13,
	SpAM [4]	RM, AGE	*irre*1, *irre*3,
	Knockoff [11]	RM, PTRATIO	*irre*11, INDUS
	Lasso Net [14]	NOX, DIS	*irre*12, *irre*13
	DeepVIB (G)	LSTAT, PTRATO	*irre*13, *irre*4
	DeepVIB (E)	LSTAT, CRIM	*irre*9, *irre*5
CHP	Lasso [1]	MedInc, AveBedrms	*irre*7, Population
	Elastic Net [32]	MedInc, Latitude	*irre*3, Population
	SCAD [3]	AveRooms, AveBedrms	*irre*6, *irre*7
	RF [13]	MedInc, AveOccup	*irre*6, *irre*9
	AdaBoost [33]	MedInc, Longitude	*irre*5, *irre*6
	GL [7]	MedInc, Longitude	*irre*1, *irre*4
	SpAM [4]	MedInc, AveRooms	*irre*18, *irre*20
	Knockoff [11]	MedInc, Latitude	AveRooms, AveBedrms
	Lasso Net [14]	MedInc, HouseAge	*irre*7, *irre*20
	DeepVIB (G)	MedInc, Latitude	*irre*3, *irre*6
	DeepVIB (E)	MedInc, AveOccup	*irre*5, *irre*12
Diabetes	Lasso [1]	BMI, BP	*irre*15, *irre*19
	Elastic Net [32]	BMI, BP	*irre*4, *irre*16
	SCAD [3]	s1, s5	age, *irre*1
	RF [13]	BMI, s5	sex, s4
	AdaBoost [33]	BMI, s5	age, *irre*18
	GL [7]	BMI, s6	*irre*7, *irre*12
	SpAM [4]	BMI, s4	s3, *irre*14
	Knockoff [11]	BMI, s5	*irre*17, *irre*18
	Lasso Net [14]	BMI, BP	*irre*5, *irre*20
	DeepVIB (G)	BMI, sex	*irre*3, *irre*5
	DeepVIB (E)	BMI, s5	*irre*7, *irre*9

5. Conclusions

In this paper, we investigated DNNs for variable selection through an information-theoretic lens. First, we demonstrated the rationality of variable selection with IB and then proposed a new statistic to measure the variable importance. On this basis, a new algorithm based on DeepVIB was designed to compute the statistic where the Kullback–Leibler divergence was estimated in cases of Gaussian distribution and exponential distribution. Finally, the experimental results indicated the superiority of DeepVIB over classical variable-selection methods.

Although the proposed algorithm achieved satisfactory performance, there are still many problems. For example, the proposed algorithm requires a loop depending on p, which is time-consuming. In addition, the current variable selection results rely on the assumption that the number of relevant variables p_{true} is known, which is rare in real scenes. In the future, we will focus on processing the empirical statistics (such as normalization) and mapping them over a fixed range. In this way, it will be easier to find the optimal threshold for variable selection.

Author Contributions: All authors made great contributions to the work. Methodology, J.P., W.L., K.J., L.L. and T.L.; investigation, J.P. and L.L.; writing—original draft preparation, J.P. and W.L.; writing—review and editing, J.P., W.L., L.L., K.J. and T.L.; visualization, J.P.; supervision, W.L. and F.C.; project administration, F.C.; and funding acquisition, F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities of China (2662020LXQD002, 2662022YJ005), Knowledge Innovation Program of Wuhan-Shuguang Project (2022010801020234) and by the Doctoral Scientific Research Foundation (XJ19BS09).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In this appendix, we will provide the proofs of the estimation of $KL(p(z|x)|r(z))$ in VIB. The parameters in $p(z|x)$ are learned by the encoder, and $r(z)$ is assumed as the standard distribution.

Case 1: $r(z)$ and $p(z|x)$ obey the Gaussian distribution

$$r(z) \sim \mathcal{N}(0, I), \quad p(z|x) \sim \mathcal{N}(\mu, \sigma^2 I).$$

In [35], Kingma et al. assumed that the K dimensions of the true posterior $Z|X$ are approximately independent. Hence, the resulting estimator is

$$KL(p(z|x)|r(z)) = -\frac{1}{2}\sum_{j=1}^{K}(1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2), \tag{A1}$$

where μ_j and σ_j^2 are the expectation and variance of the j-th dimension of $Z|X$, respectively.

Case 2: $r(z)$ and $p(z|x)$ obey the exponential distribution

$$r(z) = \exp(-z)\mathbf{1}(z \geq 0), \quad p(z|x) = \frac{1}{\sigma}\exp(-\frac{z-\mu}{\sigma})\mathbf{1}(z \geq \mu), \tag{A2}$$

where $\mathbf{1}$ is an indicator function. For convenience, we first discuss the case $K = 1$. The KL-divergence between $p(z|x)$ and $r(z)$ can be written as:

$$\begin{aligned} KL(p(z|x)|r(z)) &= \int \frac{1}{\sigma}\exp\left(-\frac{z-\mu}{\sigma}\right)\log\frac{\frac{1}{\sigma}\exp\left(-\frac{z-\mu}{\sigma}\right)}{\exp\left(-z\right)}dz \\ &= -\log\sigma + \int \frac{\sigma-1}{\sigma}z\frac{1}{\sigma}\exp\left(-\frac{z-\mu}{\sigma}\right)dz + \int \frac{\mu}{\sigma}\frac{1}{\sigma}\exp\left(-\frac{z-\mu}{\sigma}\right)dz \\ &= -\log\sigma + \sigma - 1 + \mu. \end{aligned}$$

According to the assumption that the K dimensions of $Z|X$ are approximately independent [35], the equation above can approximately transform into following:

$$KL(p(z|x)|r(z)) = \sum_{j=1}^{K}(-\log\sigma_j + \sigma_j - 1 + \mu_j), \tag{A3}$$

where μ_j and σ_j are the location parameter and scale parameter, respectively, of the j-th dimension of $Z|X$.

References

1. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. B Methodol.* **1996**, *58*, 267–288. [CrossRef]
2. Yuan, M.; Lin, Y. Model selection and estimation in regression with grouped variables. *J. Roy. Statist. Soc. B Methodol.* **2006**, *68*, 49–67. [CrossRef]
3. Fan, J.; Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.* **2001**, *96*, 1348–1360. [CrossRef]
4. Ravikumar, P.; Lafferty, J.; Liu, H.; Wasserman, L. Sparse additive models. *J. Roy. Statist. Soc. B Methodol.* **2009**, *71*, 1009–1030. [CrossRef]
5. Chen, H.; Wang, Y.; Zheng, F.; Deng, C.; Huang, H. Sparse modal additive model. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 2373–2387. [CrossRef] [PubMed]
6. Yin, J.; Chen, X.; Xing, E.P. Group sparse additive models. In Proceedings of the International Conference on Machine Learning, Edinburgh, Scotland, 26 June–1 July 2012; pp. 871–878.
7. Mukherjee, S.; Zhou, D.X.; Shawe-Taylor, J. Learning coordinate covariances via gradients. *J. Mach. Learn. Res.* **2006**, *7*, 519–549.
8. Ye, G.B.; Xie, X. Learning sparse gradients for variable selection and dimension reduction. *Mach. Learn.* **2012**, *87*, 303–355. [CrossRef]
9. Feng, Y.; Yang, Y.; Suykens, J.A. Robust gradient learning with applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 822–835. [CrossRef]
10. Liu, L.; Song, B.; Pan, Z.; Yang, C.; Xiao, C.; Li, W. Gradient learning under tilted empirical risk minimization. *Entropy* **2022**, *24*, 956. [CrossRef] [PubMed]
11. Barber, R.F.; Candès, E.J. A knockoff filter for high-dimensional selective inference. *Ann. Stat.* **2019**, *47*, 2504–2537. [CrossRef]
12. Candes, E.; Fan, Y.; Janson, L.; Lv, J. Panning for gold: 'model-X' knockoffs for high dimensional controlled variable selection. *J. Roy. Statist. Soc. B Methodol.* **2018**, *80*, 551–577. [CrossRef]
13. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Routledge: New York, NY, USA, 2017; p. 368.
14. Lemhadri, I.; Ruan, F.; Tibshirani, R. Lassonet: Neural networks with feature sparsity. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Virtual, 13–15 April 2021; pp. 10–18.
15. Feng, J.; Simon, N. Sparse-input neural networks for high-dimensional nonparametric regression and classification. *arXiv* **2017**, arXiv:1711.07592.
16. Agarwal, R.; Melnick, L.; Frosst, N.; Zhang, X.; Lengerich, B.; Caruana, R.; Hinton, G.E. Neural additive models: Interpretable machine learning with neural nets. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021; Volume 34, pp. 4699–4711.
17. Lu, Y.; Fan, Y.; Lv, J.; Stafford Noble, W. DeepPINK: Reproducible feature selection in deep neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31.
18. Shwartz-Ziv, R.; Tishby, N. Opening the black box of deep neural networks via information. *arXiv* **2017**, arXiv:1703.00810.
19. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. In Proceedings of the Annual Allerton Conference on Communication, Control, and Computing, Monticello, VA, USA, 22–25 September 2000; pp. 368–377.
20. Zaidi, A.; Estella-Aguerri, I.; Shamai, S. On the information bottleneck problems: Models, connections, applications and information theoretic views. *Entropy* **2020**, *22*, 151. [CrossRef]
21. Ahuja, K.; Caballero, E.; Zhang, D.; Gagnon-Audet, J.C.; Bengio, Y.; Mitliagkas, I.; Rish, I. Invariance principle meets information bottleneck for out-of-distribution generalization. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021; Volume 34, pp. 3438–3450.

22. Chalk, M.; Marre, O.; Tkacik, G. Relevant sparse codes with variational information bottleneck. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29, pp. 1957–1965.
23. Voloshynovskiy, S.; Taran, O.; Kondah, M.; Holotyak, T.; Rezende, D. Variational information bottleneck for semi-supervised classification. *Entropy* **2020**, *22*, 943. [CrossRef]
24. Strouse, D.; Schwab, D.J. The information bottleneck and geometric clustering. *Neural Comput.* **2019**, *31*, 596–612. [CrossRef] [PubMed]
25. Still, S.; Bialek, W.; Bottou, L. Geometric clustering using the information bottleneck method. In Proceedings of the Advances in Neural Information Processing Systems, Xi'an, China, 5 November 2003; Volume 16, pp. 596–612.
26. Alemi, A.A.; Fischer, I.; Dillon, J.V.; Murphy, K. Deep variational information bottleneck. *arXiv* **2016**, arXiv:1612.00410.
27. Ye, M.; Sun, Y. Variable selection via penalized neural network: A drop-out-one loss approach. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5620–5629.
28. Davisson, L. Rate distortion theory: A mathematical basis for data compression. *IEEE Trans. Commun.* **1972**, *20*, 1202. [CrossRef]
29. Hafez-Kolahi, H.; Kasaei, S. Information bottleneck and its applications in deep learning. *arXiv* **2019**, arXiv:1904.03743.
30. Wu, T. Intelligence, physics and information–the tradeoff between accuracy and simplicity in machine learning. *arXiv* **2020**, arXiv:2001.03780.
31. Zamanzade, E.; Mahdizadeh, M. Entropy estimation from ranked set samples with application to test of fit. *Rev. Colomb. Estad.* **2017**, *40*, 223–241. [CrossRef]
32. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. Roy. Statist. Soc. B Methodol.* **2005**, *67*, 301–320. [CrossRef]
33. Quinlan, J.R. Bagging, boosting, and C4.5. In Proceedings of the National Conference on Artificial Intelligence, Portland, OR, USA, 4–8 August 1996; pp. 725–730.
34. Yang, L.; Lv, S.; Wang, J. Model-free variable selection in reproducing kernel Hilbert space. *J. Mach. Learn. Res.* **2016**, *17*, 2885–2908.
35. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.