TOWARD MINIMAL-SUFFICIENCY IN REGRESSION TASKS: AN APPROACH BASED ON A VARIATIONAL ESTIMATION BOTTLENECK

Zhaoyan Lyu[†], Gholamali Aminian[§] and Miguel R. D. Rodrigues

Department of Electronic and Electrical Engineering, University College London, United Kingdom
{z.lyu.17, g.aminian, m.rodrigues}@ucl.ac.uk

ABSTRACT

We propose a new *variational estimation bottleneck* based on a mean-squared error metric to aid regression tasks. In particular, this bottleneck – which draws inspiration from a *variational information bottleneck* for classification counterparts – consists of two components: (1) one captures the notion of $\mathcal{V}_r$–sufficiency that quantifies the ability for an *estimator* in some class of estimators $\mathcal{V}_r$ to infer the quantity of interest ; (2) the other component appears to capture a notion of $\mathcal{V}_r$-minimality that quantifies the ability of the *estimator* to generalize to new data. We demonstrate how to train this bottleneck for regression problems. We also conduct various experiments in image denoising and deraining applications showcasing that our proposed approach can lead to neural network regressors offering better performance without suffering from overfitting.

Index Terms— Deep learning, Information Bottleneck, regression, information theory

1. INTRODUCTION

Deep learning currently offers state-of-the-art performance in many signal & image processing tasks such as image denoising [1], image super-resolution [2], deraining [3–5], and many more. Therefore, it has been increasingly relevant to cast insight onto the performance of these powerful learning models.

The information bottleneck (IB) [6] is one of the popular analysis frameworks that has been used to study the behavior of deep neural networks. It suggests that a well-trained neural network performing some classification task processes data such that useful information about the underlying target (label) is preserved, whereas irrelevant information is compressed. However, in spite of the fact that the IB has been shown to be useful in various works e.g. [7], it also suffers from several drawbacks. First, the information-theoretic quantity underlying the IB – mutual information (MI) – is very difficult to estimate in the typical high-dimensional setting. Indeed, the existing MI estimators for high-dimensional random variables usually come with either a high bias or variance [8, 9]. Second, the MI between the neural network input and the neural network representation is not well-defined for deterministic neural networks [10]. Finally, the IB framework applies primarily for classification problems in *lieu* of regression ones [7, 11–14].

Inspired by the recent decodable IB principle [15], we alleviate these issues by introducing a *variational estimation bottleneck* framework applicable to regression problems. Our framework uses the quadratic loss function applicable to regression tasks in *lieu* of the log-loss relevant to classification problems; it also leads to new $\mathcal{V}_r$-estimation quantities that are the counterpart of $\mathcal{V}$-information in [15]; and it also leads to quantities that appear to connect to the notions of sufficiency and minimality in representation learning. Our framework, in addition, also involves quantities that can be more easily estimated than the standard minimum mean-squared error (MMSE) that has been recently adopted in IB frameworks such a [16] and [17], due to the restriction of the estimators to a class of estimators $\mathcal{V}_r$.

Finally, our framework also leads to an immediate approach to training a neural network for regression tasks.

In summary, our contributions are as follows:

- We introduce the *variational estimation bottleneck* principle in terms of mean square error: this variational estimation bottleneck framework applicable to practical regression-like problems is the counterpart of the variational IB framework applicable to classification tasks.
- We demonstrate how to implement our variational estimation bottleneck with deep neural networks.
- We also demonstrate that the variational estimation bottleneck framework can lead to neural network regressors that outperform existing ones within the context of image denoising and deraining tasks.

We adopt the following notation in the sequel. Uppercase letters denote random variables (e.g., Z), lower-case letters denote random variable realizations (e.g. z), and calligraphic letters denote sets (e.g. $\mathcal{Z}$). The distribution of the

[†]The first author is supported by China Scholarship Council.

[§]The second author is supported by the Royal Society Newton International Fellowship, grant no. NIF\R1 \192656 .

978-1-7281-6338-3/21/$31.00 ©2021 IEEE

random variable Z is denoted by P_Z and the joint distribution of two random variables (Z_1, Z_2) is denoted by $P_{Z_1 Z_2}$. If P and Q are probability measures over space $\mathcal{X}$, and P is absolutely continuous with respect to Q, the Kullback-Leibler (KL) divergence between P and Q is given by $D(P\|Q) \triangleq \int_{\mathcal{X}} \log\left(\frac{dP}{dQ}\right) dP$. The mutual information between two random variables X and Y is defined as the KL divergence between the joint distribution and product-of-marginal distribution $I(X;Y) \triangleq D(P_{X,Y}\|P_X \otimes P_Y)$.

2. RELATED WORK

The well-known IB has been first proposed in [18] for classification tasks. In particular, for the Markov chain $Y_c \to X_c \to Z_c$, the IB is defined as follows:

$$\mathcal{L}_{IB}\left(P_{Z_c|X_c}\right) \triangleq I(X_c; Z_c) - \gamma I(Y_c; Z_c), \tag{1}$$

where γ is the Lagrange multiplier. The IB re-invigorated important notions in statistics such as sufficiency and minimality [19]; it has also been used to analyze various learning algorithms [6]. The only closed form solution to minimization IB problem over $P_{Z_c|X_c}$ distributions, (1), by considering jointly Gaussian (X_c, Y_c) is shown in [20]. Different applications of IB in machine learning are provided in [21].

The IB principle has also been used more recently within the context of deep learning. In particular, building upon pioneering work [6], the IB has been used as a tool to analyze [22, 23] or design [7, 10, 11] deep neural networks. For example, leveraging variational bounds of Shannon information, [7] has used a new IB based loss function to train deep neural networks.

Variations of the IB principle have also been introduced in recent years. For example, [17] introduced a generalized IB based on f-divergence. The authors also proposed an estimation bottleneck based on χ^2-information, but this quantity is difficult to estimate in practice, preventing its applicability in various problems. [16] in turn introduced a robust IB based on minimum means-squared error and Fisher information metrics to construct a classifier which is more robust to small perturbations.

More recently, inspired by [15], [14] introduced a new IB – so-called the $\mathcal{V}$-information bottleneck – that articulates about the amount of useful information a representation embodies about a target usable by a classifier drawn from a family of classifiers $\mathcal{V}$. This bottleneck reduces to the classical IB if the family of classifiers $\mathcal{V}$ corresponds to the universe of possible classifiers $\mathcal{U}$. This $\mathcal{V}$-information bottleneck has been proven to be a useful design tool, leading to neural networks exhibiting state-of-the-art performance properties [14].

Nevertheless, the majority of the work to date has focused solely on IBs relevant to classification rather than regression tasks, involving metrics or loss functions such a Shannon information or log loss [7, 11–14], so existing IB based analysis and design principles do not carry over immediately to regression problems.

Our work fills this gap by introducing a (variational) estimation bottleneck. It draws inspiration from the $\mathcal{V}$-information bottleneck in [14], but it departs from this work in various ways: (a) it uses loss functions relevant for regression problems, (b) it uses quantities that appear to link better with sufficiency and minimality notions relevant for regression problems, and (c) it also proposes new mechanisms to design neural networks targeting our proposed (variational) estimation bottleneck.

3. REGRESSION PROBLEM

We focus exclusively on regression problems. We let Y_r denote our target (clean data), X_r denote the data (corrupted data), and Z_r denote a data representation. The target-data pair are drawn from the joint distribution $P_{Y_r,X_r} = P_{Y_r} \cdot P_{Y_r|X_r}$ The data representation Z_r is in turn generated from the data X_r using the distribution $P_{Z_r|X_r}$. Therefore, the following Markov chain holds:

$$Y_r \to X_r \to Z_r \tag{2}$$

We also let $\hat{Y}_r = f(Z_r)$ represent the target prediction from the data representation based on the use of a decoder $f \in \mathcal{V}_r$ drawn from a family of decoders $\mathcal{V}_r$. Note that Y_r, X_r, Z_r and $\hat{Y}_r$ are continuous-valued random variables in the same domain space, $\mathbb{R}^n$. This therefore applies to regression tasks such as denoising and deraining.

We will measure the discrepancy between the target prediction and the original target using the standard quadratic loss given by:

$$L(y_r, z_r) = \|y_r - f(z_r)\|_2^2 \tag{3}$$

Therefore, we can write the average (population) risk as follows:

$$\begin{aligned} R(f, Z_r) &= \mathbb{E}_{P_{X_r,Y_r}}\left[\mathbb{E}_{P_{Z_r|X_r}}[L(Y_r, f(Z_r))]\right] \\ &= \mathbb{E}_{P_{X_r,Y_r}}\left[\mathbb{E}_{P_{Z_r|X_r}}[\|Y_r - f(Z_r)\|_2^2]\right] \end{aligned} \tag{4}$$

We next introduce our proposed bottleneck framework leading up to data representations appropriate for regression tasks.

4. $\mathcal{V}_R$-ESTIMATION BOTTLENECK

We first introduce a new notion – $\mathcal{V}_r$-MSE – corresponding to the best achievable mean-squared error when we restrict the estimator to be drawn from a family of estimators $\mathcal{V}_r$. This is given by:

$$\mathrm{MSE}_{\mathcal{V}_r}(Y_r \to Z_r) \triangleq \min_{f \in \mathcal{V}_r} \mathbb{E}_{P_{Y_r,Z_r}}[\|Y_r - f(Z_r)\|_2^2] \tag{5}$$

Note that this quantity is a natural generalization of the minimum mean-squared error (MMSE) given by:

$$\mathrm{MMSE}(Y_r|Z_r) \triangleq \mathbb{E}_{P_{Y_r,Z_r}}[\|Y_r - \mathbb{E}_{P_{Y_r|Z_r}}(Y_r|Z_r)\|_2^2] \tag{6}$$

because we can recover the standard MMSE from the $\mathcal{V}_r$-MSE provided that the family of estimators $\mathcal{V}_r$ includes the conditional mean estimator. A key advantage of the $\mathcal{V}_r$-MSE in relation to the standard MMSE is that it is much easier to estimate because we do not often have access to the posterior $P_{Y_r|Z_r}$ required to compute the conditional mean $\mathbb{E}_{P_{Y_r|Z_r}}[Y_r|Z_r]$.

We can now introduce our new bottleneck applicable to regression problems building upon the work by [6], [14]. Our bottleneck framework – which we refer to by $\mathcal{V}_r$-estimation bottleneck (EB) – is based on the objective given by:

$$\mathcal{L}_{EB}(P_{Z_r|X_r}) \triangleq \text{MSE}_{\mathcal{V}_r}(Y_r \to Z_r) - \beta \times \text{MSE}_{\mathcal{V}_r}(X_r \to Z_r) \tag{7}$$

where $\beta > 0$ is a parameter trading-off the effect of the first and second components. In particular, our EB objective acts as a proxy to learn data representation mechanisms (i.e. $P_{Z_r|X_r}$) that possess two key properties: (1) first, such a mechanism ought to allow reliable reconstruction of the data target Y_r given the data representation Z_r using some estimator $f(\cdot)$ drawn from the family of estimators $\mathcal{V}_r$; (2) second, such a mechanism ought to prevent reliable reconstruction of the (corrupted) data X_r given the data representation Z_r using the estimator $f(\cdot)$ selected from the family of estimators $\mathcal{V}_r$. Therefore, together, these two components of our EB objective are meant to offer both good statistical performance (first component) and good generalization performance (second component).

We can indeed prove that a representation mechanism that minimizes the first term of the EB leads to the best possible statistical risk as follows:

$$\begin{aligned} p^{\star}_{Z_r|X_r} &= \arg \min_{p_{Z_r|X_r}} \text{MSE}_{\mathcal{V}_r}(Y_r \to Z_r) \\ &= \arg \min_{p_{Z_r|X_r}} \min_{f \in \mathcal{V}_r} \mathbb{E}_{P_{Y_r,Z_r}}[\|y_r - f(z_r)\|_2^2] \\ &= \arg \min_{p_{Z_r|X_r}} \min_{f \in \mathcal{V}_r} \mathbb{E}_{P_{Y_r}}[\mathbb{E}_{P_{Z_r|Y_r}}[\|y_r - f(z_r)\|_2^2]] \\ &= \arg \min_{p_{Z_r|X_r}} \min_{f \in \mathcal{V}_r} \mathbb{E}_{P_{X_r,Y_r}}[\mathbb{E}_{P_{Z_r|X_r}}[\|y_r - f(z_r)\|_2^2]] \\ &= \arg \min_{p_{Z_r|X_r}} \min_{f \in \mathcal{V}_r} R(f, Z_r) \end{aligned} \tag{8}$$

We also conjecture that a representation mechanism that in addition also maximizes the second term of the EB leads to good generalization behaviour in the presence of new data. Overall, the proposed estimation bottleneck differs from the existing classical IB and the recently introduced variational IB in view of the fact that it relies on quantities that are more adequate for regression problems.

5. IMPLEMENTATION

We now illustrate how to develop data representations conforming to the proposed variational estimation bottleneck in a data driven manner[1] for a simple regression task involving the recovery of a noiseless image from a noisy one. The approach – which involves a forward-propagation and a back-propagation step as described next – is succinctly depicted in Fig.1.

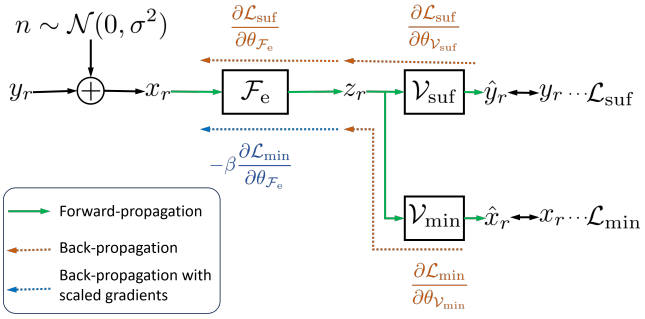

Fig. 1: Variational Estimation Bottleneck implementation within the context of a denoising task. y_r denotes the samples of clean image, n denotes the additive Gaussian noise and x_r is the noisy image. $\hat{y}_r$ and $\hat{x}_r$ are predictions delivered by the sufficiency decoder and minimality decoder respectively.

5.1. Key Modules

Unlike standard image processing neural networks involving the use of e.g. an end-to-end deep neural network, we construct an encoder and two decoders that target the various terms of the variational estimation bottleneck objective.

In particular, the encoder $\mathcal{F}_{\text{e}}$ converts a noisy image x_r onto some representation z_r in high-dimensional space. Then, the decoder $\mathcal{V}_{\text{suf}}$ converts the high-dimensional representation z_r onto an estimate $\hat{y}_r$ of the original noiseless image y_r. This decoder, $\mathcal{V}_{\text{suf}}$, is optimized by targeting $\text{MSE}_{\mathcal{V}_r}(Y_r \to Z_r)$, which is denoted as $\mathcal{L}_{\text{suf}}$. And the decoder $\mathcal{V}_{\text{min}}$ in turn converts the high-dimensional representation z_r onto an estimate $\hat{x}_r$ of the noisy image x_r. This decoder, $\mathcal{V}_{\text{min}}$, is optimized by targeting $\text{MSE}_{\mathcal{V}_r}(X_r \to Z_r)$, which is denoted as $\mathcal{L}_{\text{min}}$.

The encoder is parameterized via a series of parameters $\theta_{\mathcal{F}_e}$ In turn, the decoders $\mathcal{V}_{\text{suf}}$ and $\mathcal{V}_{\text{min}}$ are parameterized via the parameters $\theta_{\mathcal{V}_{\text{suf}}}$ and $\theta_{\mathcal{V}_{\text{min}}}$, respectively. We will often refer to the decoders $\mathcal{V}_{\text{suf}}$ and $\mathcal{V}_{\text{min}}$ as the statistical sufficiency and statistical minimality decoders in view of the fact these are trying to enforce the first and second terms of the EB objectives that we have conjectured to correspond to representation sufficiency and minimality respectively.

5.2. Optimization

The models are trained as follows: The parameters in the sufficiency decoder $\theta_{\mathcal{V}_{\text{suf}}}$ and the minimality decoder $\theta_{\mathcal{V}_{\text{min}}}$ are updated based on the gradients generated by $\mathcal{L}_{\text{suf}}$ and $\mathcal{L}_{\text{min}}$ respectively. The gradient from the minimality branch on encoder parameters $\theta_{\mathcal{F}_{\text{e}}}$, however, will be reversed and multiplied with a scaling factor β before being applied on the pa-

[1] That is, the various terms appearing in the objective of the variational estimation bottleneck are approximated using samples drawn from the relevant distribution.

rameters of encoder. The gradient on $\theta_{\mathcal{F}_e}$ generated by $\mathcal{L}_{\min}$ will be applied normally.

5.3. Deployment

The models are finally deployed after training as follows: We retain the encoder that provides a mechanism to map the noisy data onto some high-dimensional representation; we also retain the sufficiency decoder that provides a mechanism to map the high-dimensional representation onto a estimate of the noiseless data; however, we discard the minimality decoder whose sole purpose is to allow us to learn encoder-decoder pairs targeting the variational estimation bottleneck objective.

6. EXPERIMENTAL RESULTS[2]

We have applied our framework to various image denoising and deraining tasks.

6.1. MNIST Denoising

Image denoising is a classical regression task that can be formulated by the Markov chain in (2). Thus, we perform a MNIST image with Gaussian noise denoising task based on a convolutional neural network. We use the original training and validation split in the standard MNIST dataset [24], and input samples are generated by applying additional Gaussian noise with standard deviation taking value in $\{0.6, 0.8, 1.0, 1.2\}$. Fig.2 shows some examples drawn from this dataset.

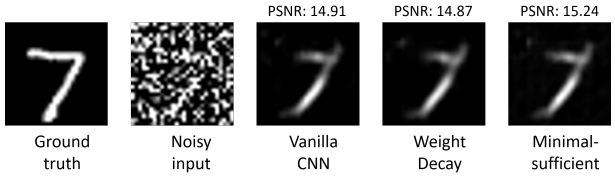

Fig. 2: Visualization of ground truth, noisy samples with $\sigma = 1.0$, and denoised results using different training methods.

We test the proposed minimal-sufficient denoising framework on MNIST dataset with additive Gaussian noise. The MNIST dataset is normalized and each pixel takes value between 0 and 1. The additive noises on each pixel are generated based on i.i.d. Gaussian distributions with 0 mean and standard deviation taking value in $\{0.6, 0.8, 1.0, 1.2\}$.

The encoder module is a 4-layer convolutional neural network (CNN) with 64, 3×3 filters per layer. Each convolutional layer is followed by a ReLU activation. The $\mathcal{V}_{\text{suf}}$ and $\mathcal{V}_{\min}$ modules are 1-layer linear convolutional neural networks with filter shape 3×3. For the sake of comparison, we test the minimal-sufficient framework, vanilla CNN network without minimality branch and vanilla CNN with weight decay regularization. Each setup is trained using Adam optimizer with a learning rate of $1\text{e}{-3}$ for 100 epochs. We choose β based on a grid search and the best results are obtained with $\beta = 0.1$, and each experiment is repeated 5 times to get the average performance.

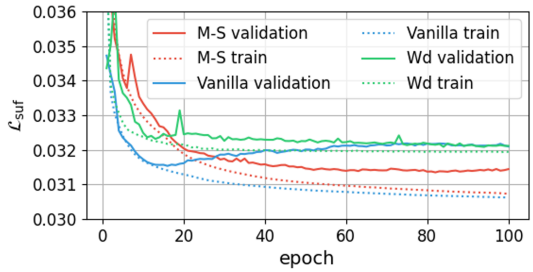

Fig. 3: Training curves for MNIST denoising $\mathcal{V}_r$-sufficiency $\mathcal{L}_{\text{suf}}$ versus epochs.

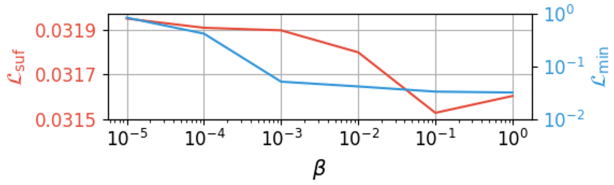

Fig. 4: $\mathcal{L}_{\text{suf}}$ and $\mathcal{L}_{\min}$ when choosing different β.

Results: The $\mathcal{L}_{\text{suf}}$, peak signal-to-noise ratio (PSNR) and structured similarity (SSIM) for different noise levels are summarized in Table. 1. We can see that the proposed minimal-sufficient framework outperforms other training strategies in terms of $\mathcal{L}_{\text{suf}}$ and PSNR, especially when the noise level is high. However, the SSIM for minimal-sufficient framework is less stable and sometimes worse than others. This is not entirely surprising because our framework attempts to optimize MSE performance rather than other metrics.

We can also find that the minimal-sufficiency approach generalizes better than competing approaches. In particular, Fig. 3 demonstrates that our approach overfits less to the training data in comparison with standard training. Meanwhile, although weight decay regularization prevents the network from overfitting, the validation loss is much higher than minimal-sufficient model and even slightly worse than vanilla training.

Finally, we also plot the sufficiency and minimality terms on the validation set when using different β in Fig.4. It can be observed that in a certain range, promoting the minimality term (increasing β) during training will result in a better sufficiency loss, which shows the benefit of minimal-sufficiency training again.

6.2. Rain12 Deraining

The other experiment we present is image deraining, which attempt to remove raining streaks from a raining picture. [3–5]

In particular, we use the Rain12 dataset provided by [3]. This dataset contains 12 image samples with synthetic rain streaks. We split the training, validation and test set from Rain12 with a ratio of 50%, 25% and 25%. Each picture is

[2]Code to reproduce experiments is to be found at https://github.com/iiml-ucl/rib

Table 1: $\mathcal{L}_{\text{suf}}$, PSNR and SSIM on test set for MNIST denoising with different noise levels. (Wd: weight decay; M-S: minimal-sufficient training)

	$\mathcal{L}_{\text{suf}}$				PSNR				SSIM			
σ	0.6	0.8	1.0	1.2	0.6	0.8	1.0	1.2	0.6	0.8	1.0	1.2
Vanilla	1.34e-3	2.25e-3	3.20e-3	4.02e-3	18.70	16.47	14.93	13.95	**0.83**	**0.71**	**0.58**	0.47
Wd	1.44e-3	2.32e-3	3.21e-3	4.01e-3	18.40	16.34	14.93	13.96	0.81	0.68	0.57	**0.48**
M-s	**1.34e-3**	**2.23e-3**	**3.14e-3**	**3.94e-3**	**18.71**	**16.52**	**15.03**	**14.04**	**0.83**	0.64	0.46	**0.48**

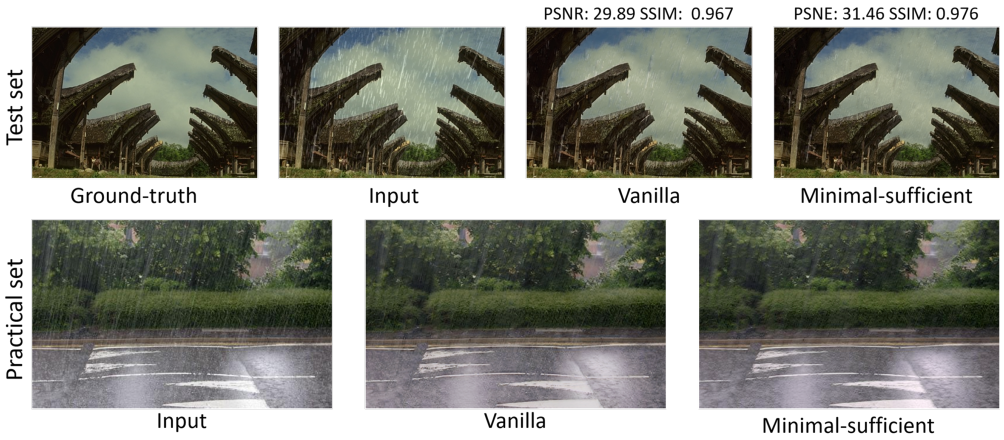

Fig. 5: Visualization of deraining task: (top) sample from test set, and (bottom) sample from practical set.

cropped into 128×128 square patches with a stride of 64. In addition, we also use the practical dataset from [5] for testing. Unlike the Rain12 dataset, the practical dataset are real life raining images without labels.

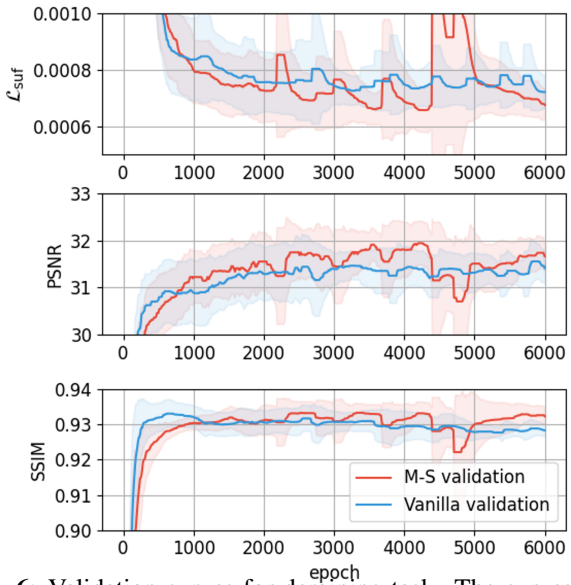

Fig. 6: Validation curves for deraining task. The curves are averaged from 5 individual runs and the shadows indicate the standard deviation. Top-down: (1) $\mathcal{L}_{\text{suf}}$ (2) PSNR, and (3) SSIM.

The encoder and decoder modules are similar to those in MNIST denoising task. But compared with the setup for MNIST, the encoder for Rain12 deraining has four layers and the decoder has an extra convolutional layer with ReLU nonlinearity. The whole model is optimized by Adam optimizer with learning rate of 0.001. The parameter β is set to 10 to give best sufficiency loss on validation set, which is found by a grid search. Because the training set is small, The network are trained for 6000 epochs.

Results: As shown in Fig.6, compared with vanilla training on validation set, the proposed minimal-sufficient training outperforms in terms of $\mathcal{L}_{\text{suf}}$, PSNR and SSIM. Note that we do not compare with weight decay regularizer here because in this regression task and current network setup, weight decay regularization makes the network seriously under-fit. The output of our minimal-sufficient training is visually better as visualized in Fig 5.

7. CONCLUSION

In this work, we propose a new bottleneck framework for regression applications which is based on variational mean square error. Our $\mathcal{V}_r$-estimation bottleneck consists of two quantities including $\mathcal{V}_r$-sufficiency and $\mathcal{V}_r$-minimality. We empirically show that $\mathcal{V}_r$-estimation bottleneck framework could improve performance in image denoising and deraining applications in comparison to other denoising algorithms based on neural networks.

For future work, we will extend this framework to settings where the target output quantity and the input one may live in Euclidean spaces with different dimensions, such as in image super-resolution tasks.

8. REFERENCES

[1] Chunwei Tian, Yong Xu, Lunke Fei, and Ke Yan, "Deep learning for image denoising: a survey," in *International Conference on Genetic and Evolutionary Computing*. Springer, 2018, pp. 563–572.

[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*. Springer, 2014, pp. 184–199.

[3] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown, "Rain streak removal using layer priors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2736–2744.

[4] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2017.

[5] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan, "Deep joint rain detection and removal from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1357–1366.

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

[7] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[8] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker, "On variational bounds of mutual information," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5171–5180.

[9] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm, "Mine: mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[10] Rana Ali Amjad and Bernhard C Geiger, "Learning representations for neural network-based classification using the information bottleneck principle," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 9, pp. 2225–2239, 2019.

[11] Artemy Kolchinsky, Brendan D Tracey, and David H Wolpert, "Nonlinear information bottleneck," *Entropy*, vol. 21, no. 12, pp. 1181, 2019.

[12] Alexander A Alemi, Ian Fischer, and Joshua V Dillon, "Uncertainty in the variational information bottleneck," *arXiv preprint arXiv:1807.00906*, 2018.

[13] Pradeep Kr Banerjee and Guido Montúfar, "The variational deficiency bottleneck," *arXiv preprint arXiv:1810.11677*, 2018.

[14] Yann Dubois, Douwe Kiela, David J Schwab, and Ramakrishna Vedantam, "Learning optimal representations with the decodable information bottleneck," *arXiv preprint arXiv:2009.12789*, 2020.

[15] Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon, "A theory of usable information under computational constraints," in *International Conference on Learning Representations*, 2019.

[16] Ankit Pensia, Varun Jog, and Po-Ling Loh, "Extracting robust and accurate features via a robust information bottleneck," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 131–144, 2020.

[17] Hsiang Hsu, Shahab Asoodeh, Salman Salamatian, and Flavio P Calmon, "Generalizing bottleneck problems," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 531–535.

[18] Naftali Tishby, Fernando C Pereira, and William Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[19] Thomas M Cover, *Elements of information theory*, John Wiley & Sons, 1999.

[20] Gal Chechik, Amir Globerson, Naftali Tishby, Yair Weiss, and Peter Dayan, "Information bottleneck for gaussian variables.," *Journal of machine learning research*, vol. 6, no. 1, 2005.

[21] Ziv Goldfeld and Yury Polyanskiy, "The information bottleneck problem and its applications in machine learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 19–38, 2020.

[22] Ravid Shwartz-Ziv and Naftali Tishby, "Opening the black box of deep neural networks via information," *arXiv preprint arXiv:1703.00810*, 2017.

[23] Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox, "On the information bottleneck theory of deep learning," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2019, no. 12, pp. 124020, 2019.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.