On the Information Plane of Autoencoders

Nicolás I. Tapia
Department of Electrical Engineering
Universidad de Chile
Santiago, Chile
nicolas.tapia@ug.uchile.cl

Pablo A. Estévez
Department of Electrical Engineering
Universidad de Chile
Santiago, Chile
pestevez@cec.uchile.cl

***Abstract*—The training dynamics of hidden layers in deep learning are poorly understood in theory. Recently, the Information Plane (IP) was proposed to analyze them, which is based on the information-theoretic concept of mutual information (MI). The Information Bottleneck (IB) theory predicts that layers maximize relevant information and compress irrelevant information. Due to the limitations in MI estimation from samples, there is an ongoing debate about the properties of the IP for the supervised learning case. In this work, we derive a theoretical convergence for the IP of autoencoders. The theory predicts that ideal autoencoders with a large bottleneck layer size do not compress input information, whereas a small size causes compression only in the encoder layers. For the experiments, we use a Gram-matrix based MI estimator recently proposed in the literature. We propose a new rule to adjust its parameters that compensates scale and dimensionality effects. Using our proposed rule, we obtain experimental IPs closer to the theory. Our theoretical IP for autoencoders could be used as a benchmark to validate new methods to estimate MI in neural networks. In this way, experimental limitations could be recognized and corrected, helping with the ongoing debate on the supervised learning case.**

I. Introduction

The complexity of deep learning compared with traditional machine learning methods has not allowed a full theoretical understanding of its properties. In particular, it is poorly understood how each hidden layer evolves during training to achieve the end goal of the learning setting. An approach to understand deep neural networks using information-theoretic concepts was first proposed in [1] and further developed in [2]. The key quantity in this framework is mutual information (MI), derived from the concept of entropy [3].

The Shannon entropy $H(X)$, or simply entropy, of a discrete random variable (RV) $X \in \mathcal{X}$ with probability mass function p_X is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p_X(x) \log(p_X(x)). \tag{1}$$

Then, the MI between X and another discrete RV $Y \in \mathcal{Y}$ is given by

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{2}$$

Nicolás Tapia acknowledges financial support from the National Agency for Research and Development (ANID) / Scholarship Program / MAGISTER NACIONAL/2019 - 22191803. The authors acknowledge financial support from ANID-Chile through grant FONDECYT 1171678. Additionally, the authors acknowledge financial support from the Department of Electrical Engineering at Universidad de Chile.

When X is a continuous RV with probability density function f_X, its differential entropy $h(X)$ is defined as

$$h(X) = -\int_{x \in \mathcal{X}} f_X(x) \log(f_X(x)) dx. \tag{3}$$

Similarly, the differential MI between X and another continuous RV Y is given by

$$I(X;Y) = h(X) - h(X|Y) = h(Y) - h(Y|X). \tag{4}$$

The MI measures the dependency between X and Y, and attains its minimum, equal to zero, if they are independent.

Let A, B and C form a Markov chain $A \to B \to C$, which means that C is conditionally independent of A given B. Then, they satisfy the Data Processing Inequality (DPI) [3]:

$$I(A;B) \geq I(A;C). \tag{5}$$

Essentially, it means that the information that B contains about A cannot be increased through any transformation of B.

Let X and Y be the input and the desired output of a neural network, and let T be an intermediate hidden layer. According to [1], they form a Markov chain $Y \to X \to T$, satisfying a DPI. The Information Bottleneck (IB) theory predicts that T transforms X so that it maximizes the relevant information about Y while minimizing the information about X [1]. Motivated by this insight, the Information Plane (IP) was proposed to analyze training dynamics.

Definition 1 (Information Plane): The Information Plane (IP) is the space with coordinate axes $I(X;T)$ and $I(T;Y)$ at which a hidden layer T in a given training iteration is mapped onto a single point, describing a trajectory during training.

According to the IB theory, the learning trajectories of each layer should move towards the point of maximum $I(T;Y)$ and minimum $I(X;T)$. It was experimentally found in [2] that neural networks exhibit two phases: fitting and compression. The former corresponds to increasing both $I(T;Y)$ and $I(X;T)$, whereas the latter corresponds to decreasing $I(X;T)$ while $I(T;Y)$ increases or stays the same. After compression, each layer stabilizes in a theoretical IB bound.

The existence of these two phases, and their dependence on the activation functions and the MI estimators have been a topic of ongoing research [4]–[6]. This motivates the study of classes of neural networks with simpler theoretical behaviors. In this way, an intermediate step can be obtained to recognize

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

undesired experimental limitations and validate new methods to obtain the IP in more general cases.

The IP for a class of neural networks called autoencoders was studied in [7]. An autoencoder outputs a reconstruction X' of its input X. It has two components: an encoder and a decoder. The encoder is a first stack of layers that maps X to an encoding Z, that is, $Z = \mathrm{Encoder}(X)$. The decoder is a second stack of layers that reconstructs the input from this encoding, that is, $X' = \mathrm{Decoder}(Z)$. Generally, Z has the smallest dimensionality, so the last layer of the encoder is called the *bottleneck layer*.

For simplicity, let assume that both the encoder and the decoder have L layers. Let $\{T_i^E\}_{i=1}^{L-1}$ and $\{T_i^D\}_{i=1}^{L-1}$ be the intermediate layers of the encoder and the decoder, respectively. Then, the autoencoder is represented by

$$X \to T_1^E \to \cdots \to T_{L-1}^E \to Z \to T_1^D \to \cdots \to T_{L-1}^D \to X'. \tag{6}$$

Fig. 5 of Sec. IV-B illustrates this representation for the specific autoencoder used in our experiments.

The IP of Definition 1 is not suitable for autoencoders as the desired output is the input itself, reducing the plane to a line. It was noted in [7] that an autoencoder satisfies two DPIs analogous to the supervised learning case: the forward DPI

$$I(X;T_1^E) \geq \cdots \geq I(X;T_{L-1}^E) \geq I(X;Z), \tag{7}$$

and the backward DPI

$$I(T_{L-1}^D;X') \geq \cdots \geq I(T_1^D;X') \geq I(Z;X'). \tag{8}$$

Both DPIs can be extended to the output and the input layers, respectively. Based on these DPIs, a modified IP was proposed.

Definition 2 (Information Plane of an Autoencoder): The IP of an autoencoder is the space with coordinate axes $I(X;T)$ and $I(T;X')$ at which a hidden layer T is mapped as in Definition 1. For readability, $I(X;T)$ is called the *input MI* and $I(T;X')$ is called the *output MI* of the layer T.

It was postulated in [7] that the IP curves show two distinct patterns in the form of a bifurcation point depending on whether the bottleneck layer size is larger or smaller than the intrinsic dimensionality of the input data. The authors experimentally found that the IP curves show a compression phase after some critical value for the bottleneck layer size, and that this effect intensifies as the bottleneck gets larger. Our replication of this result is shown in Fig. 6 of Sec. IV-B.

Having a compression phase with a large bottleneck means that input information is lost, similar to the supervised learning setting. However, large bottlenecks allow near perfect reconstruction to be achieved. The experimental finding in [7] is conflicting with perfect reconstruction, which requires that all input information is contained at the output layer.

We hypothesize that autoencoders with a large bottleneck layer size do not allow compression. In this work, we theoretically derive the convergence of the IP of autoencoders for different bottleneck layer sizes. Next, we study the limitations of the estimator used in [7] that could have lead to their reported results. The main contributions of this work are the following: a) a theoretical IP of autoencoders with ideal convergence values for the input MI and output MI; and b) an improved adjustment rule for the parameter of the MI estimator used in [7] that allows better agreement between estimations and expected theoretical behaviors.

II. Theoretical Information Plane of an Autoencoder

We assume the common premise that the size K of the bottleneck layer Z restricts the information that can be transferred from the encoder to the decoder. In this section, we derive the theoretical limit of the input MI and the output MI using ideal autoencoders.

Definition 3 (Ideal Autoencoder): An ideal autoencoder minimizes the distance between X' and X (reconstruction error) as much as allowed by its bottleneck layer size K.

A well-trained autoencoder of enough capacity approximates an ideal autoencoder at the end of training. Therefore, this theoretical limit provides an ideal convergence for each IP curve during training. The specific trajectory followed from initialization to convergence cannot be derived from this analysis because it depends on the optimization process. As a result, we can only provide a sketch of the theoretical IP.

A. Mutual Information Analysis

Let X and T be the input and an arbitrary hidden layer of a neural network, respectively. At any given training iteration, the layer T is a deterministic function of X. Generally, X has an absolutely continuous component. It was proved in [8] that, in this case, $I(X;T)$ is infinite for almost any selection of weights. In the literature, this problem is avoided either by discretizing X or by measuring MI after adding noise [5]. In the following, we consider both approaches.

Definition 4 (Discretization Approach): Let A and B be two continuous RVs. Let A_q and B_q be their discretized versions obtained using a suitable quantization method. Then, $I(A;B)$ is replaced by $\hat{I}_D(A;B) := I(A_q;B_q)$.

Definition 5 (Noise Addition Approach): Let A and B be two continuous RVs. Let R be an independent additive noise. Then, $I(A;B)$ is replaced by $\hat{I}_C(A;B) := I(A;B+R)$.

Definition 6 (Unified Approach): Let A and B be two continuous RVs. Then, $I(A;B)$ is replaced by $\hat{I}(A;B)$, with

$$\hat{I}(A;B) := \begin{cases} \hat{I}_D(A;B) \text{ if discretization is used,} \\ \hat{I}_C(A;B) \text{ if noise addition is used.} \end{cases} \tag{9}$$

Using $\hat{I}(A;B)$ to analyze autoencoders allows us to prove the following lemma for the output MI.

Lemma 1 (Output MI): $\hat{I}(T;X') = \hat{I}(X;X')$.

Proof: The discretization approach implies

$$\hat{I}(T;X') = I(T_q;X'_q) = H(X'_q) - H(X'_q|T_q), \tag{10}$$

and the noise addition approach implies

$$\hat{I}(T;X') = I(T;X'+R) = h(X'+R) - h(X'+R|T). \tag{11}$$

Since X' is a deterministic function of any layer T, $H(X'_q|T_q)$ equals zero and $h(X'+R|T)$ equals $h(R)$. In both approaches,

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

the output MI is the same for any hidden layer T, and equal to $\hat{I}(X;X')$ by taking the particular case $T = X$. ■

Let $\lambda = \lambda(K)$ be an increasing function of the bottleneck layer size K that represents the maximum amount of information that can be transferred from the encoder to the decoder. The following lemma can be proved for the input MI.

Lemma 2 (Input MI): An ideal autoencoder satisfies $\hat{I}(X;X') = \hat{I}(X;Z) = \min\{\lambda,\ \hat{I}(X;X)\}$

Proof: The forward DPI implies that

$$\hat{I}(X;X) \geq \hat{I}(X;Z) \geq \hat{I}(X;X'). \tag{12}$$

In particular, $X' = X$ achieves the upper bound of $\hat{I}(X;X')$. In this sense, an ideal autoencoder maximizes $\hat{I}(X;X')$ to minimize the reconstruction error. Since the transfer of information is restricted only on the bottleneck layer, the decoder contributes to the maximization of $\hat{I}(X;X')$ by achieving $\hat{I}(X;Z) = \hat{I}(X;X')$ in (12). Additionally, the encoder contributes to the maximization of $\hat{I}(X;X')$ by maximizing $\hat{I}(X;Z)$, which is bounded by $\hat{I}(X;X)$ in (12). Due to the bottleneck restriction, $\hat{I}(X;Z)$ is also bounded by λ, implying that the achievable maximum is $\min\{\lambda,\ \hat{I}(X;X)\}$. ■

Borrowing the notion from the discrete case, we can interpret $\hat{I}(X;T)$ as the information that layer T preserves from the input X. The forward DPI, completed as

$$\begin{aligned}\hat{I}(X;X) \geq \hat{I}(X;T_1^E) \geq \cdots \geq \hat{I}(X;Z) \geq \cdots \\ \geq \hat{I}(X;T_{L-1}^D) \geq \hat{I}(X;X'),\end{aligned} \tag{13}$$

implies that the information is decreased or at most preserved from input to output. For random weights, we expect a significant information loss through the layers. Therefore, when initializing practical autoencoders, we expect a strict inequality in (13) and a small value of $\hat{I}(X;X')$. After training, as stated before, practical autoencoders approximate ideal ones. Lemmas 1 and 2 allow us to prove our main result for ideal autoencoders to complete our characterization of the IP.

Theorem 1 (Two Patterns): Consider an ideal autoencoder with a bottleneck layer size K. Let $\lambda = \lambda(K)$ be the maximum amount of information that can be transferred through the bottleneck layer:

a) If $\lambda > \hat{I}(X;X)$, then the output MI and the input MI are equal to $\hat{I}(X;X)$ for every hidden layer T.

b) If $\lambda < \hat{I}(X;X)$, then the output MI is equal to λ for every hidden layer T. Moreover, the encoder has input MIs satisfying

$$\hat{I}(X;X) \geq \hat{I}(X;T_1^E) \geq \cdots \geq \hat{I}(X;Z) = \lambda, \tag{14}$$

and the decoder has input MIs satisfying

$$\hat{I}(X;Z) = \hat{I}(X;T_1^D) = \cdots = \hat{I}(X;X') = \lambda. \tag{15}$$

Proof: Case a): Suppose $\lambda > \hat{I}(X;X)$. This implies $\hat{I}(X;X') = \hat{I}(X;X)$ according to Lemma 2. From Lemma 1, this implies $\hat{I}(T;X') = \hat{I}(X;X)$, proving the result for the output MI. On the other hand, $\hat{I}(X;X') = \hat{I}(X;X)$ implies that the forward DPI in (13) achieves equality with $\hat{I}(X;T) = \hat{I}(X;X)$, proving the result for the input MI.

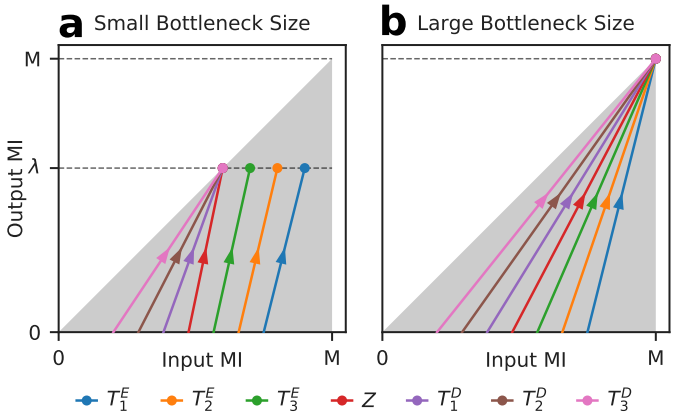

Fig. 1. Theoretical IP when the bottleneck layer size is (**a**) small and (**b**) large. The convergence is highlighted with circles, and the feasible region corresponds to the shaded area. The trajectories to convergence depend on the optimization algorithm, so they were arbitrarily drawn as straight lines.

Case b): Suppose $\lambda < \hat{I}(X;X)$. According to Lemma 2, it implies $\hat{I}(X;X') = \hat{I}(X;Z) = \lambda$. From Lemma 1, $\hat{I}(X;X') = \lambda$ implies $\hat{I}(T;X') = \lambda$, proving the result for the output MI. In addition, the equality $\hat{I}(X;Z) = \lambda$ and the forward DPI directly imply (14), proving the result for the input MI in the encoder. Finally, the equality $\hat{I}(X;X') = \hat{I}(X;Z) = \lambda$ implies that the forward DPI in the decoder achieves equality as in (15), proving the result for the input MI in the decoder. ■

B. Consequences in the Information Plane

For simplicity, we define M as the total information available at the input, i.e., $M = \hat{I}(X;X)$. A direct consequence of the forward DPI in (13) is the feasible region of the IP. Since $\hat{I}(X;X') \leq \hat{I}(X;T)$ and $\hat{I}(T;X') = \hat{I}(X;X')$ (Lemma 1), the curves are restricted to the region below the bisector $\hat{I}(X;T) = \hat{I}(T;X')$. Furthermore, $\hat{I}(X;X) \geq \hat{I}(X;T)$, implying that the curves are restricted to the left of the vertical line $\hat{I}(X;T) = M$. In summary, the feasible region is the one contained in the triangle of vertices $(0,0)$, $(M,0)$ and (M,M), shown as a shaded area in Fig. 1.

Every layer has the same output MI, equal to $\hat{I}(X;X')$, at each iteration according to Lemma 1. Because $\hat{I}(X;X')$ is generally small at initialization, the IP curves will start with input MIs satisfying the inequality of the forward DPI in (13) and with low output MIs, as sketched in Fig. 1. As the reconstruction error is minimized, $\hat{I}(X;X')$ will grow, implying that the output MIs will grow as well.

Theorem 1 allows us to derive the convergence of each layer in the IP. For $\lambda > \hat{I}(X;X) = M$, every IP curve will converge to the same input MI and output MI, both equal to M. As a result, the IP curves will converge to the point (M,M), at the edge of the feasible region of the IP (see Fig.1b). On the other hand, for $\lambda < \hat{I}(X;X) = M$, the input MIs of the encoder will converge to a decreasing sequence up to the bottleneck layer were $\hat{I}(X;Z) = \lambda$, and the input MIs of the decoder will converge to the same value $\hat{I}(X;T^D) = \lambda$. The output

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

MI will converge to λ, so the bottleneck layer and the decoder layers will reach the bisector $\hat{I}(X;T) = \hat{I}(T;X')$, whereas the encoder layers will converge to the interior of the feasible region (see Fig.1a).

In summary, our theoretical analysis predicts the existence of two distinct patterns in the IP depending on the input information M and the bottleneck layer size K in agreement with [7]. However, we predict neither a bifurcation point nor a compression phase that intensifies will larger K. If K is large ($\lambda > M$), all layers converge together on the bisector because they contain all the input information. Therefore, no input information is compressed, which relates to perfect reconstruction. Otherwise, if K is small ($\lambda < M$), some information is compressed through the encoder to achieve the allowed information λ at the bottleneck. Then, it is transferred through the decoder preserving as much as possible, without further compression, to minimize the reconstruction error.

The IP of each regime is sketched in Fig. 1 with both the encoder and the decoder having four layers as this is the number of layers used in our experiments. In this sketch, we have drawn the IP curves as straight lines reaching the theoretical convergence. However, these trajectories are arbitrary. The analysis predicts that the output MIs are always equal, but the evolution of the input MIs cannot be deduced from it. Instead, they depend on the optimization algorithm.

The critical value of K that marks the transition between these patterns, also marks the point after which the information is no longer compressed. Therefore, it could be estimated by measuring the MI at the end of training for a range of bottleneck layer sizes. According to [7], this critical K would approximate the intrinsic dimensionality of the data.

III. Estimation of Mutual Information

In practice, the theoretical MI cannot be obtained because the data distribution is unknown, so the MI has to be estimated from samples. Following [7], we estimate MI during training using the matrix-based estimator proposed in [9]. It estimates Renyi's α-entropy [10], defined for an RV $X \in \mathcal{X}$ by

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int_{\mathcal{X}} f_X^\alpha(x) dx. \tag{16}$$

The standard Shannon entropy is the limit $\alpha \to 1$. In this section, we briefly describe the estimator and our proposed method to adjust its kernel width.

A. Matrix–Based Mutual Information Estimator

Let X be an RV and $x_i \in \mathcal{X}, i = 1, \ldots, N$ be N realizations of it. Let $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be an infinitely divisible positive definite kernel that defines a Gram matrix $K \in \mathbb{R}^{N \times N}$ as $K_{ij} = \kappa(x_i, x_j)$. The normalized Gram matrix A is

$$A_{ij} = \frac{1}{N} \frac{K_{ij}}{\sqrt{K_{ii} K_{jj}}}. \tag{17}$$

Let $\lambda_i(A)$ be the i-th eigenvalue of A. In [9], an estimator of the α-entropy of X is defined as

$$S_\alpha(A) := \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{N} \lambda_i(A)^\alpha \right). \tag{18}$$

Let Y be another RV with normalized Gram matrix B. Using the element-wise product $A \circ B$, the joint-entropy estimator is defined as

$$S_\alpha(A, B) := S_\alpha \left(\frac{A \circ B}{\mathrm{tr}(A \circ B)} \right). \tag{19}$$

From (18) and (19), an MI estimator is given by:

$$I_\alpha(A; B) := S_\alpha(A) + S_\alpha(B) - S_\alpha(A, B). \tag{20}$$

Both $S_\alpha(A)$ and $I_\alpha(A, B)$ are restricted to $[0,\ \log(N)]$.

As in [7], we set $\alpha = 1.01$ to approximate Shannon entropy and choose a gaussian kernel G_σ with width σ, given by

$$G_\sigma(x_i, x_j) = \beta \exp \left(-\frac{||x_i - x_j||^2}{2\sigma^2} \right), \tag{21}$$

where β is a constant whose value is irrelevant because it is canceled out in the normalized Gram matrix (17).

B. Kernel Width Selection

The value of the kernel width σ is central in the performance of the estimator described in Sec. III-A. The following properties hold for the gaussian kernel:

$$\lim_{\sigma \to 0} S_\alpha(A) = \log(N), \tag{22}$$

$$\lim_{\sigma \to 0} I_\alpha(A, B) = \log(N), \tag{23}$$

$$\lim_{\sigma \to \infty} S_\alpha(A) = 0, \tag{24}$$

$$\lim_{\sigma \to \infty} I_\alpha(A, B) = 0. \tag{25}$$

They imply that the value of σ controls the operating point of the estimator relative to the bounds because a value too large or too small saturates $S_\alpha(A)$ and $I_\alpha(A; B)$ to 0 and $\log(N)$, respectively. This saturation has to be avoided to have discriminative estimates. Therefore, a suitable value of σ has to be determined for an RV X of d dimensions and N samples.

A common rule for the Gaussian kernel is the Silverman's rule of thumb [11] that comes from the literature of density estimation. For the j-th dimension of X, it is given by

$$\sigma_j = \left(\frac{4}{2+d} \right)^{1/(4+d)} \hat{\sigma}_j N^{-1/(4+d)}, \tag{26}$$

where $\hat{\sigma}_j$ is the empirical standard deviation of the j-th dimension. Since $0.92 \leq (4/(2+d))^{1/(4+d)} \leq 1.06$, this term can be safely discarded for this application. To study autoencoders in [7], this rule was further simplified to

$$\sigma = \gamma N^{-1/(4+d)}, \tag{27}$$

where $\gamma > 0$ is an empirically determined constant.

The rule in (27) has three main limitations when applied to neural networks. The first one is that an appropriate value of γ has to be found experimentally and it can change significantly between variables. This means that a different γ could be needed at different layers and even at different iterations as the layers change during training. The second one is that the rule is affected by linear transformations of X, whereas Shannon MI is not. Indeed, let X be scaled by $a \in \mathbb{R}$. In (21), this is

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

equivalent to keep the unscaled variable X and to replace σ by σ/a, changing the estimation. This is problematic because neural networks often contain normalization layers such as batch normalization [12], and neural layers change their variance during training. In particular, the MI can be overestimated or underestimated depending on whether the variance increases or decreases, respectively. The third limitation is that the rule is affected by dimensionality. To see this, let X have zero mean and unit variance dimension-wise, and let x_1 and x_2 be two i.i.d. samples. Then,

$$\mathbb{E}[||x_1 - x_2||^2] = 2d. \tag{28}$$

Therefore, the mean square distance is proportional to the number of dimensions. In (21), this means that higher dimensions decrease the effective kernel width on average, increasing the estimated MI value. As a result, neural layers with more units tend to show an overestimated MI.

The need of improving the adjustment of the kernel width was acknowledged in [6]. They proposed a method for the supervised learning setting by leveraging the structure induced by the labels at each layer and at each training iteration. This method is not applicable in the general case, and in particular to the case of autoencoders.

C. Proposed Rule for Kernel Width Selection

We propose a new rule for the kernel width σ that alleviates the aforementioned limitations. Our rule can be understood as augmenting the constant γ with variance and dimensionality dependencies. First, we normalize the variable X dimension-wise as

$$X_j \leftarrow \frac{X_j}{\sqrt{\hat{\sigma}_j^2 + \epsilon}},\ j = 1, \ldots, d, \tag{29}$$

where $\hat{\sigma}_j$ is the estimated standard deviation for the j-th dimension and ϵ is a small constant to avoid division by zero. This change effectively makes the kernel width different for each dimension and proportional to its standard deviation, returning to the Silverman's rule given by (26), so that the kernel is not affected by changes in scale. If X has subsets of components with shared statistics, like channels in an image, then the normalization should be performed by aggregating the statistics of each group, as done in the batch normalization technique. Unlike the Silverman's rule, we additionally modify the rule in (27) to

$$\sigma = \gamma \sqrt{d} N^{-1/(4+d)}, \tag{30}$$

so that the kernel width compensates the dimensionality dependency of the mean square distance.

IV. Experiments

In this section, we present the results of two sets of experiments. First, we validate the proposed kernel width selection rule with a toy problem called *correlated gaussians*. Next, we estimate the IPs of the same autoencoder used in [7] and compare them with the theoretical result of Sec. II. We tried $\gamma \in [0.1,\ 10]$ and selected the value where the dynamics were best shown, although they could be observed in most of them. We used $\epsilon = 10^{-8}$ and logarithms of base 2.

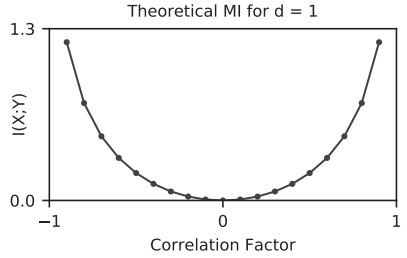

Fig. 2. Theoretical MI in the *correlated gaussians* problem.

A. Toy Problem: Correlated Gaussians

This problem was used in [13] to compare MI estimators, and it is defined as follows. Let $X \sim \mathcal{N}(0, I)$ and $Y \sim \mathcal{N}(0, I)$ be two multivariate normal RVs with d dimensions and with component-wise correlation $\text{corr}(X_i, Y_j) = \delta_{ij}\rho\ \forall i, j \in \{1, \ldots, d\}$, where $\rho \in (-1, 1)$ is the correlation factor and δ_{ij} is Kronecker's delta. The problem consists of estimating $I(X;Y)$ from N samples, whose theoretical value is given by

$$I(X;Y) = -\frac{d}{2}\log(1 - \rho^2). \tag{31}$$

This theoretical MI is illustrated in Fig. 2 for $d = 1$. For other dimensions, it is scaled by d according to (31).

We compare the performance of the MI estimator when selecting the kernel width using the previous rule (27) and the rule proposed in Sec. III-C. In this problem, the normalization of the variables has no effect because X and Y are standard gaussians. Therefore, the difference lies in whether we add the extra term $\sqrt{d}$ as in (30). We evaluate the cases $d \in \{10,\ 100,\ 1000\}$ to cover a range that is commonly found in neural networks, and we set $N = 128$ samples. We set $\gamma = 2$ for the proposed rule and $\gamma = 2\sqrt{10}$ for the rule (27), so that they are equivalent when $d = 10$. The results are shown in Fig. 3 with the mean and standard deviation of 50 independent runs.

Neither rule can properly approximate the theoretical value. Moreover, the previous rule (27) is affected by the saturation effect described in Sec. III-B because the estimations grow close to $\log_2(128) = 7$ for $d = 100$ and almost exactly for $d = 1000$. Even with the proposed rule, the estimations do not grow linearly with d as in the theoretical MI, and the minimum is not zero. However, it can compensate the dimensionality effect to approximate the expected shape of the curve. Therefore, we can expect to approximate tendencies in the IP with this estimator rather than exact values.

The resolution of the estimation, i.e., the observed range of values, decreases for larger dimensions, but it can be controlled by the number of samples. The estimated MI for $d = 100$ using the proposed rule is shown in Fig. 4 after increasing the number of samples from 128 to 256 and 512. The resolution progressively increases, improving the confidence

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

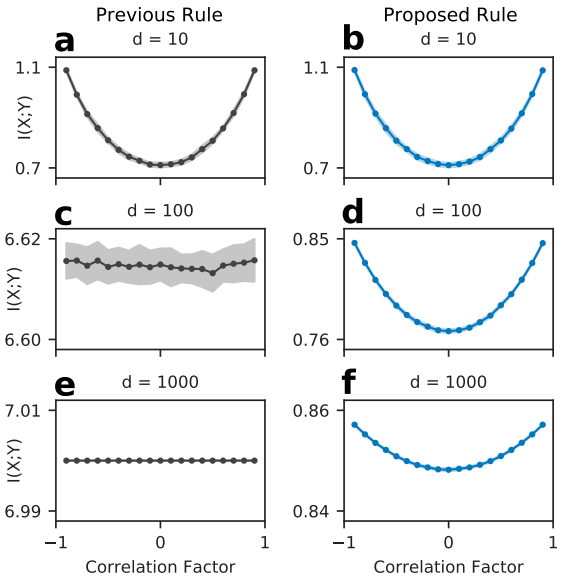

Fig. 3. Comparison between (**a**,**c**,**e**) the previous rule and (**b**,**d**,**f**) the proposed rule for the kernel width selection of the MI estimator using the *correlated gaussians* problem with different dimensions.

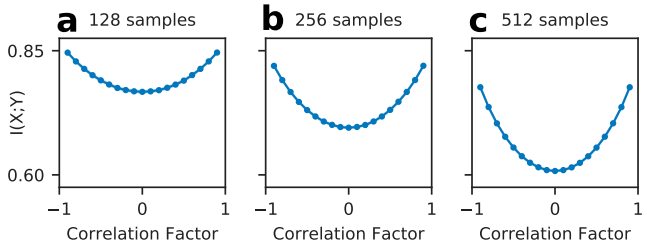

Fig. 4. Change in the estimated MI when the number of samples is varied, using the *correlated gaussians* problem with 100 dimensions and the proposed rule for the kernel width selection of the MI estimator.

in the observed differences. The maximum number of samples is limited by memory constraints because $O(N^2)$ memory is needed to compute the eigenvalues of the Gram matrix.

B. Estimated Information Planes of an Autoencoder

Following the experiment reported in [7], we train an autoencoder to reconstruct grayscale images of handwritten digits using MNIST [14]. This dataset contains 60000 training images and 10000 testing images of 28×28 pixels. The pixels of each image are normalized to the interval $[0, 1]$.

We use a fully-connected autoencoder, shown in Fig. 5, with the same architecture and training process described in [7]. The bottleneck layer size K is varied throughout the experiments. The activation function is sigmoid except for the bottleneck layer where it is linear. The model is trained to minimize the MSE between X and X' using stochastic gradient descent with learning rate 0.1, momentum 0.5, batch size 100, and 100 epochs. To estimate MI at each iteration, we average the results obtained from 10 batches of 512 samples of the testing set. To improve readability, we plot the encoder and the decoder separately in all the experiments. To reduce noise

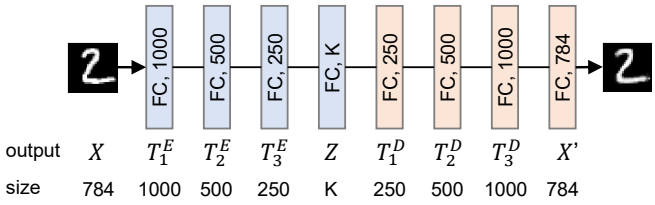

Fig. 5. Architecture of the autoencoder used in the experiments with fully-connected layers and a variable bottleneck layer size K.

and overplotting in the IPs, we first smooth the estimations by sliding a Hanning window that spans 500 iterations and then we plot a logarithmically spaced subset of iterations.

The IPs of the autoencoder for $K = 2$ and $K = 32$ are computed. We replicate the result of [7] in Fig. 6 using the previous rule for the kernel width selection with $\gamma = 25$ for the bottleneck layer and $\gamma = 5$ for the other layers. For a small K the layers do not get close to the bisector, whereas for a large K the layers show a compression phase towards the bisector. Then, we recompute the IPs in Fig. 7 using our proposed rule with $\gamma = 0.8$ for all the layers. In this case, the results follow more closely the theoretical reference shown in Fig. 1. In particular, there is no compression phase for a large K. For $K = 2$, there is a visible restriction on the information that can be transferred through the bottleneck layer that results in the compression observed in the sequence of encoder layers.

Comparing the results shown in Figs. 6 and 7 there are two notable differences. The first one is that the layers show different relative magnitudes. For example, in Fig. 6d, the layer T_3^D has a significantly higher input information than Z, which is theoretically impossible because it violates the DPI. Conversely, this situation is less significant in the correction shown in Fig. 7d, probably because the dimensionality effect has been compensated.

The second difference is the existence of the compression phase in Figs. 6c-d and the absence of it in Figs. 7c-d. Based on the theoretical analysis described in Sec. III-B, changes in scale, such as variance, can affect the estimation. A correlation between variance and the estimations was found, which can be observed in the average variance evolution of each layer shown in Fig. 8. For $K = 2$, the variance always increases, which correlates with the behavior observed in the IPs of Figs. 6a-b. On the other hand, for $K = 32$, the variance starts to decrease after an initial increasing phase, which correlates with the compression phase in Figs. 6c-d. The bottleneck layer variance is not shown in Fig. 8 because its variance magnitude is too large compared to the other layers, but the same observations apply. Finally, the variance of T_3^D does not decrease in Fig. 8d, which correlates with the absence of compression for T_3^D in the IP of Fig. 6d. This suggests that the compression phase observed for large K might be caused by variance changes, but an analysis using causal inference would be needed to draw further conclusions.

The DPIs of the autoencoder with the proposed rule are better approximated than in [7], but they are not fully satisfied

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

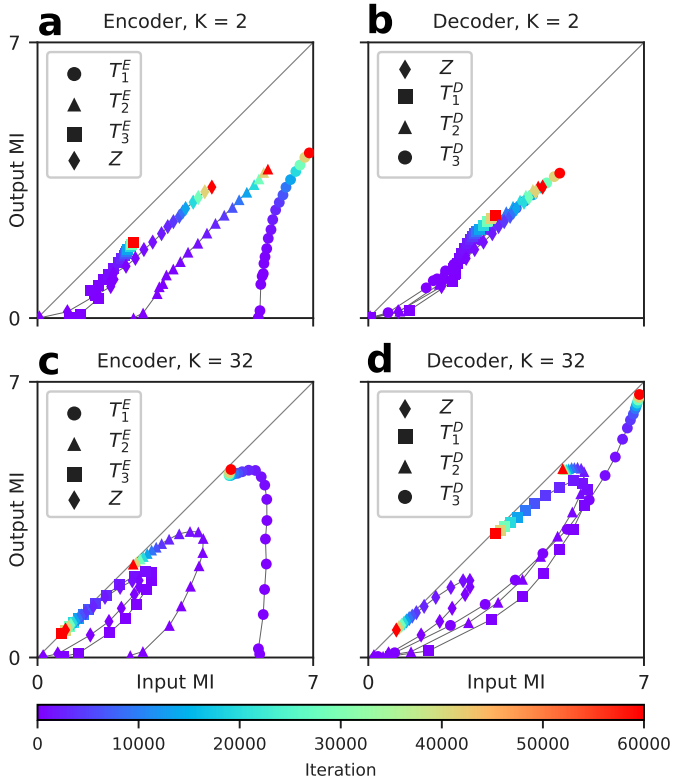

Fig. 6. Information planes for (**a**,**b**) $K = 2$ and (**c**,**d**) $K = 32$ using the previous rule for the kernel width selection (replication of [7]).

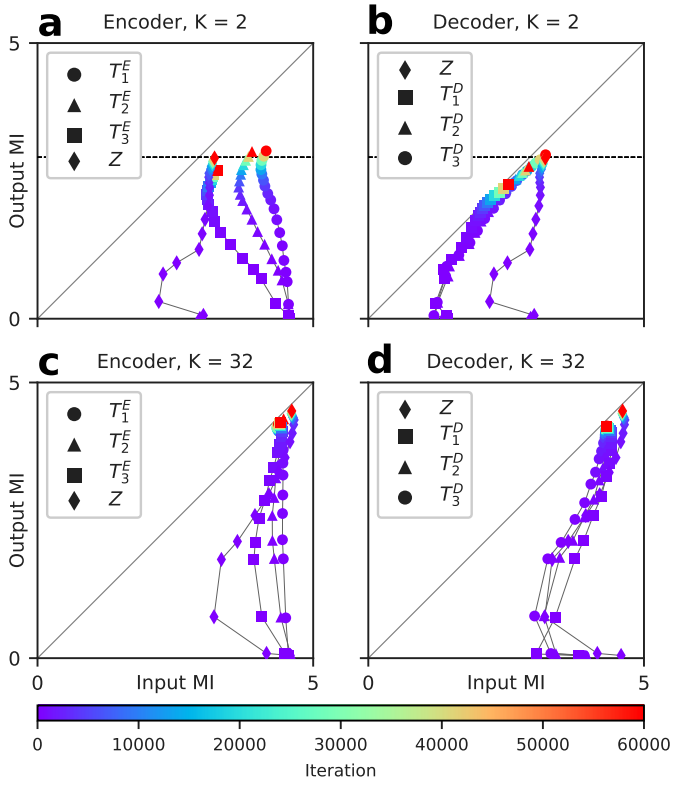

Fig. 7. Information planes for (**a**,**b**) $K = 2$ and (**c**,**d**) $K = 32$ using the proposed rule for the kernel width selection. The restriction imposed by the bottleneck layer has been highlighted with a horizontal line in the case $K = 2$.

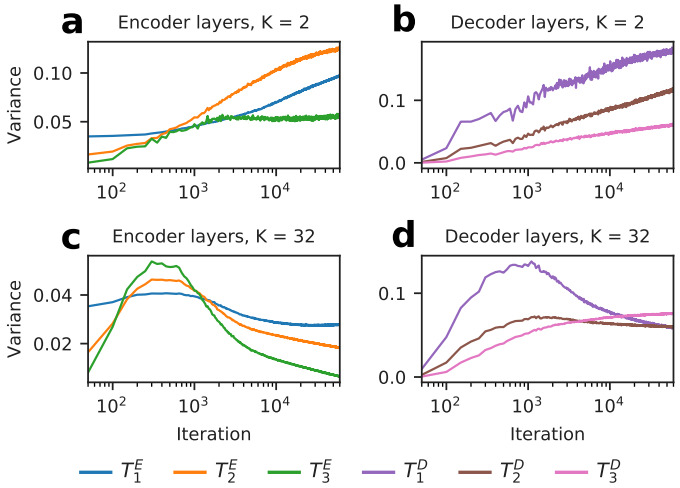

Fig. 8. Evolution of the average variance of each hidden layer during training for (**a**,**b**) $K = 2$ and (**c**,**d**) $K = 32$.

yet. For example, the ordering of the layers does not follow strictly the theoretical order even when averaging to compensate for noise in the measurements. Moreover, the theoretical analysis predicts that all layers have the same output MI at every iteration, and this was particularly violated in Fig. 7b.

Using our proposed kernel width selection rule, we analyze the effect of a range of values of K. The input MI and output MI achieved at the end of training by each layer is shown in Fig. 9. Our theoretical prediction is approximated by the experimental results. There are differences in Figs. 9b-d where the MIs should be the same for any K, specially for the estimated MI of the bottleneck layer. The MI grows with K, which agrees with the premise that larger K allows more information to be transferred. In addition, the result in Fig. 9a approximates the decreasing sequence of the encoder for small K, where the input information is decreased from its maximum value to the bottleneck layer value. This sequence shrinks as K increases, except for the bottleneck layer after $K = 2$. In Fig. 9a, the compression made by the encoder layers mostly disappears after $K = 13$. Beyond this size, all layers are mostly stabilized in their maximum value, where the input MI and the output MI are equal. Hence, $K = 13$ is an approximation of the intrinsic dimensionality of the data.

Overall, the estimations using the proposed rule for the kernel width followed reasonably well the expected curves, both in the *correlated gaussians* toy problem and in the information planes of the autoencoder. However, in those cases we had a theoretical reference to compare against the experimental results. We do not know if the estimation errors are small enough to study other problems. As with all estimators, more samples can improve the results, but the memory constraints did not allow to use more than 512 samples at a time. Despite this limitation, the MI estimator used was able to capture expected behaviors for a very high-dimensional setting, which is not the case for other estimators in the literature.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

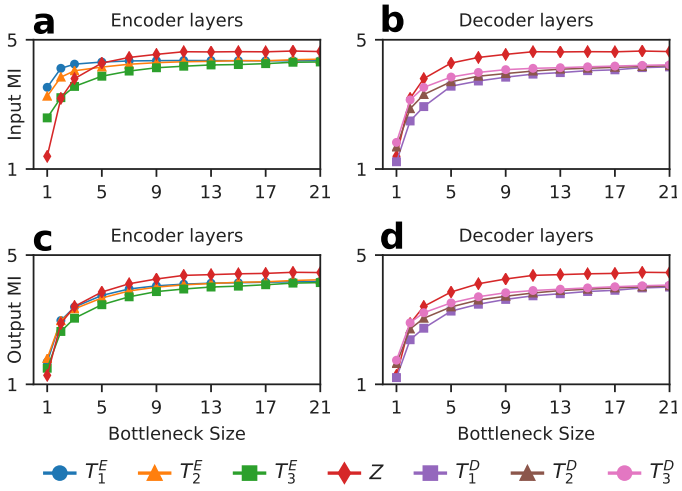

Fig. 9. Information contained in each hidden layer at the end of training for different bottleneck layer sizes, estimated with the rule proposed in Sec. III-C. (**a**,**b**) MI with the input layer. (**c**,**d**) MI with the output layer.

V. Conclusions

A particular class of neural networks, the autoencoder, allowed us to obtain theoretical convergences for the IP. They predict that the layers of an autoencoder maximize the information they contain from the input data subject to the restriction imposed by the bottleneck layer size K in the form of a maximum amount of information that can be transferred from the encoder to the decoder. As a result, two patterns appear in the IP depending on whether the bottleneck size is sufficiently large. This is in agreement with what was postulated in [7], but, contrary to their experimental findings, compression was not observed when K was large enough to allow near perfect reconstruction. Instead, compression was observed for small K and only on the encoder layers, which was linked to the loss of information imposed by the small bottleneck size. To solve this contradiction, we proposed a new rule to adjust the kernel width of the MI estimator used in [7] that compensates for variance effects, as in the original Silverman's rule, and dimensionality effects. This rule allowed us to obtain experimental results that supported our theoretical claims. As future work, these findings have to be further validated using more architectures and datasets.

The absence of information compression was explained by the fact that perfect reconstruction is impossible if any information is lost. However, there exists geometric compression because the number of dimensions is decreased in the bottleneck layer and the dispersion of the variables changes during training as observed in Fig. 8. Because the MI is invariant under bijections, it is inappropriate to measure geometric changes that do not affect the information content. On the contrary, neural networks are sensitive to these transformations, as the ultimate goal for a classification task is to transform the input variable to an output variable that admits a simple linear decision function. These other dimensions of learning that are not captured by the IB theory have been already acknowledged by [8]. It is left as future work to find another measure that captures these other phenomena to complement the theoretical analysis of neural networks.

In agreement with previous works in the IP, estimating MI in neural networks was challenging. We were not able to fully satisfy the theory in the experiments, so more work has to be done in this area. Therefore, our theoretical IP for the autoencoder might serve as a benchmark to assess new approaches to estimate MI in neural networks. In this way, an approach can be validated before using it in the supervised learning scenario where there exists an ongoing discussion on the training dynamics.

Acknowledgment

We would like to thank Shujian Yu for providing replication details and José Príncipe for useful discussions. Additionally, we would like to thank Rodrigo Carrasco, Jhon Intriago, and Leon Garcia for their idea of summarizing the maximum MI achieved for each bottleneck layer size in a single plot.

References

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

[2] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," *Why & When Deep Learning works: looking inside Deep Learning (ICRI-CI paper bundle)*, 2017.

[3] T. Cover and J. Thomas, *Elements of information theory*, 2nd ed. John Wiley & Sons, 2006.

[4] A. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. Tracey, and D. Cox, "On the information bottleneck theory of deep learning," in *Int. Conf. Learning Representations (ICLR)*, 2018.

[5] I. Chelombiev, C. Houghton, and C. O'Donnell, "Adaptive estimators show information compression in deep neural networks," in *Int. Conf. Learning Representations (ICLR)*, 2019.

[6] K. Wickstrøm, S. Løkse, M. Kampffmeyer, S. Yu, J. Principe, and R. Jenssen, "Information plane analysis of deep neural networks via matrix-based Renyi's entropy and tensor kernels," *arXiv preprint arXiv:1909.11396*, 2019.

[7] S. Yu and J. Principe, "Understanding autoencoders with information theoretic concepts," *Neural Networks*, vol. 117, pp. 104–123, 2019.

[8] R. Amjad and B. Geiger, "Learning representations for neural network-based classification using the information bottleneck principle," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019, doi: 10.1109/TPAMI.2019.2909031.

[9] L. Sanchez, M. Rao, and J. Principe, "Measures of entropy from data using infinitely divisible kernels," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 535–548, 2015.

[10] A. Rényi, "On measures of entropy and information," in *Proc. 4th Berkeley Symp. on Math. Statist. and Prob.*, vol. 1, 1961, pp. 547–561.

[11] D. Henderson and C. Parmeter, "Normal reference bandwidths for the general order, multivariate kernel density derivative estimator," *Statistics & Probability Letters*, vol. 82, no. 12, pp. 2198–2205, 2012.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Machine Learning (ICML)*, 2015, pp. 448–456.

[13] M. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *Proc. 35th Int. Conf. Machine Learning (ICML)*, 2018, pp. 531–540.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.