Skoltech

MASTER’S THESIS

Information Bottleneck Analysis of Deep Neural Networks

Master’s Educational Program: Data Science

Student: ____________ Aleksandr Tolmachev
signature

Research Advisor: ____________ Alexey Frolov
signature
Dr.Sc., Professor

Moscow 2024

Skoltech

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Оценки взаимной информации и их применение для анализа процесса обучения нейронных сетей

Магистерская образовательная программа: Науки о данных

Студент: ____________ Александр Толмачев
подпись

Научный руководитель: ____________ Алексей Фролов
подпись
д.ф.-м.н., профессор

Москва 2024

Information Bottleneck Analysis of Deep Neural Networks

Aleksandr Tolmachev

Submitted to the Skolkovo Institute of Science and Technology on June 8, 2024

ABSTRACT

The Information Bottleneck (IB) principle offers an information-theoretic framework for analyzing the training process of deep neural networks (DNNs). Its essence lies in tracking the dynamics of two mutual information (MI) values: between the hidden layer output and the DNN input/target. According to the hypothesis put forth by Shwartz-Ziv, Tishby (2017), the training process consists of two distinct phases: fitting and compression. The latter phase is believed to account for the good generalization performance exhibited by DNNs. Due to the challenging nature of estimating MI between high-dimensional random vectors, this hypothesis was only partially verified for NNs of tiny sizes or specific types, such as quantized NNs. In this work, we introduce a framework for conducting IB analysis of general NNs. Our approach leverages the stochastic NN method proposed by Goldfeld et. al.(2019) and incorporates a compression step to overcome the obstacles associated with high dimensionality. In other words, we estimate the MI between the compressed representations of high-dimensional random vectors. The proposed method is supported by both theoretical and practical justifications. Notably, we demonstrate the accuracy of our estimator through synthetic experiments featuring predefined MI values and comparison with MINE (Belghazi et. al., 2018). Finally, we perform IB analysis on a close-to-real-scale convolutional DNN, which reveals new features of the MI dynamics.

Keywords: Information bottleneck, lossy compression, information theory

Research advisor:
Name: Alexey Frolov
Degree, title: Dr.Sc., Professor

Оценки взаимной информации и их применение для анализа процесса обучения нейронных сетей

Александр Толмачев

Представлено в Сколковский институт науки и технологий 8 июня 2024 года

Аннотация

Принцип Information Bottleneck (IB) заключается в применении теоретико-информационного подхода для анализа процесса обучения глубоких нейронных сетей. Его суть заключается в исследовании взаимосвязи двух значений взаимной информации: между выходом скрытого слоя и входом/выходом всей нейронной сети. Согласно гипотезе, выдвинутой Тишби и Шварцем-Зивом (2017), процесс обучения состоит из двух отдельных этапов: подгонки и сжатия. Авторы гипотезы полагают, что последний этап отвечает за хорошую обобщающую способность, демонстрируемую глубокими нейронными сетями. Ранее известные методы оценки взаимной информации имели большую погрешность измерений в случае многомерных случайных величин, поэтому данная гипотеза была лишь частично подтверждена для нейронных сетей небольших размеров или определенного вида, например квантованных сетей. В данной работе представлен метод оценки взаимной информации для многомерных случайных величин, а также подход для проведения IB-анализа для произвольных нейронных сетей. Ключевой идеей предлагаемого подхода является использование сжатия с потерями, а также стохастических нейронных сетей, как вспомогательных обьектов для анализа исходной сети. Разработанный метод оценивает взаимную информацию между сжатыми (маломерными) представлениями многомерных случайных векторов. Точность предлагаемого подхода подтверждается теоретическими обоснованиями и экспериментами с оценкой взаимной информации между синтетическими наборами данных с известным значениями взаимной информации между ними. Кроме того, в работе представлен IB-анализ для сверточной нейронной сети, близкой к реальным размерам, который позволяет установить новые особенности динамики значений взаимной информации, связанных с Information Bottleneck.

Ключевые слова: взаимная информация, сжатие с потерями, теория информации

Научный руководитель:
Имя: Алексей Фролов
Ученое звание, степень: д.ф.-м.н., профессор

Contents

Chapter 1
Introduction

Relevance. The analysis of deep neural networks (DNNs) based on information theory is an emerging field in the theory of deep learning, which may offer a reliable and interpretable way to evaluate the performance of deep models during training and inference. This type of network analysis could complement current opaque meta-optimization algorithms for architecture search, such as ENAS [40], DARTS [35, 59, 23], evolutionary algorithms [16], and others. [47]. Additionally, this method may provide new approaches to explainable AI through the estimation of information flows in neural networks [55, 60, 21, 52, 3] or via independence testing [5, 48], as opposed to existing methods of local analysis of a model [43, 50, 44, 27] or methods based on complex manipulations with data [34, 62, 14]. Information-theoretic quantities can also be considered as regularization terms or training objectives [55, 11, 4].

The information-theoretic analysis of DNNs relies on the *Information Bottleneck* (IB) principle proposed in [54]. This concept was later developed in [55] and applied to DNNs in [49]. The major idea of the IB approach is to track the dynamics of two *mutual information* (MI) values: $I(X;L)$ between the hidden layer output (L) and the DNN input (X) and $I(Y;L)$ between the hidden layer output and the target of the model (Y). As a result of the IB analysis, the authors of the latter article put forth the so-called *fitting-compression* hypothesis, which states that the training process consists of two phases: a feature-extraction "fitting" phase (both MI values grow) and a representation compression phase ($I(Y;L)$ grows while $I(X;L)$ decreases). The authors conjectured the compression phase to account for the good generalization performance exhibited by DNNs. However, it is still debated whether empirical confirmations of the compression phase are related to improper mutual information estimators, activation function choice, or other implementation details. For a more complete overview of current IB-related problems, we refer the reader to [19].

In the original work [49], a quantization (or binning) approach was proposed to estimate the mutual information. However, this approach encountered two primary challenges. Firstly, the MI estimate was highly sensitive to the bin size selection. Secondly, for a fixed training epoch, when the training weights are held constant, L becomes a deterministic function of X, resulting in the MI being independent of the DNN parameters (and infinite for practically all regimes of interest if we speak about continuous case and reasonable activation functions, see e.g., [2]). The subsequent papers addressed the aforementioned problems.

To tackle the infinite MI problem it was proposed to consider

- stochastic NNs [1, 21, 53],
- quantized NNs [36],
- a mixture of them [10].

Simple and inconsistent binning entropy estimators have been replaced with estimators more appropriate for continuous random variables [18, 21, 20, 1].

However, the high-dimensional problem still holds, as the sample complexity (the least number of samples required for an estimation within a fixed additive gap) of any entropy estimator is proven to depend on the dimension exponentially [20, 37]. Due to the challenging nature of estimating MI between high-dimensional random vectors, the fitting-compression hypothesis has only been verified for tiny NNs or special classes of models with tractable information-theoretic quantities (e.g., [18, 36]). Some existing works on IB-analysis of large networks also exhibit signs of the curse of dimensionality [21, 1]. We mention papers that suggest using lower bounds or other surrogate objectives [4, 15, 41, 13, 28, 30, 37], advanced binning [39] or even other definitions of entropy [58, 61] in order to perform IB-analysis of large networks. It may be assumed that these methods can partially overcome the curse of dimensionality via utilizing the internal data structure implicitly, or simply from the fact that non-conventional information theory might be less prone to the curse of dimensionality.

Main purpose of the research. Main goal of the current research is to develop methods for more accurate MI estimation and apply it to explore the Information Bottleneck principle of deep neural networks.

In contrast to the approaches mentioned above, we propose a solution to the curse of dimensionality problem by explicitly compressing the data. Since most datasets exhibit internal structure (according to the *Manifold Hypothesis* [17]), it is usually sufficient to estimate information-theoretic quantities using compressed or latent representations of the data. This enables the application of conventional and well-established information-theoretic approaches to real-world machine learning problems. In the recent work of [9], the compression was used to obtain the upper bound of the random vector entropy. However, it is necessary to precisely estimate or at least bound from both sides the entropy alternation under compression in order to derive the MI estimate. In the work of [22], two-sided bounds are obtained, but only in the special case of linear compression and smoothed distributions. Our work heavily extends these ideas by providing new theoretical statements and experimental results for MI estimation via compression-based entropy estimation.

Scientific novelty. Our contribution is as follows. We introduce a comprehensive framework for conducting IB analysis of general NNs. Our approach leverages the stochastic NN method proposed in [21] and incorporates a compression step to overcome the obstacles associated with high dimensionality. In other words, we estimate the MI between the compressed representations of high-dimensional random vectors. We provide a theoretical justification of MI estimation under lossless and lossy compression. The accuracy of our estimator is demonstrated through synthetic experiments featuring predefined MI values and comparison with MINE [4]. Finally, the experiment with convolutional DNN classifier of the MNIST handwritten digits dataset [33] is performed. The experiment shows that there may be several compression/fitting phases during the training process. It may be concluded that phases revealed by information plane plots are connected to different regimes of learning (i.e. accelerated, stationary, or decelerated drop of loss function).

It is important to note that stochastic NNs serve as proxies for analyzing real NNs. This is because injecting small amounts of noise have negligible effects on outputs of layers, and the introduced randomness allows for reasonable estimation of information-theoretic quantities that depend on NN parameters. We also mention that injecting noise during training is provenerformance and generalization capabilities [25, 51].

Statements for defense. The following proposals are submitted for the Master's thesis defense:

- development of the method for the Mutual Information estimation via lossy compression
- theoretical and practical justifications of the proposed approach
- participation in the development of the framework for mutual information (MI) estimation using synthetic data and for the Information Bottleneck analysis of deep neural networks

Chapter 2
Author contribution

This thesis is primarily based on the paper [8] (see next chapter for additional information), written by the author of the work with co-authors. The work on this study was carried out in the Project Center for Next Generation Wireless and IoT at the Skolkovo Institute of Science and Technology. This work have recently been presented at the Twelfth International Conference on Learning Representations (Vienna, Austria, May 7-11 2024) during the poster session. It should be noted that it is planned to continue work on this topic during author's PhD degree.

The author of this thesis was one of the two main authors of this paper and proposed some key ideas underlying the research methodology and applications. The author implemented the part of the code that covers this research (experiments with synthetic images, several modules of the general framework for NN analysis) as well as significantly contributed to the development of theoretical statements and practical justifications of the proposed approach. Additionally, the draft manuscript preparation, its editing and review during the rebuttal period were mostly done by the first two authors of this paper.

Chapter 3
List of publications

The results of this research are presented in the following papers and conferences:

Journal publications

- I. Butakov, A. Tolmachev, S. Malanchuk, A. Neopryatnaya, A. Frolov, K. Andreev **Information Bottleneck Analysis of Deep Neural Networks via Lossy Compression** (published at the *ICLR 2024*, Poster, A^* Core conference)
- I.D. Butakov, S.V. Malanchuk, A.M. Neopryatnaya, A. D. Tolmachev, K. V. Andreev, S. A. Kruglik, E. A. Marshakov, A. A. Frolov **High-Dimensional Dataset Entropy Estimation via Lossy Compression** // *Journal of Communications Technology and Electronics*, 2021, № 66, pp. 764–768

Preprints

- I. Butakov, A. Tolmachev, S. Malanchuk, A. Neopryatnaya, A. Frolov, K. Andreev **Mutual Information Estimation via Normalizing Flows** //arXiv preprint. [2024]. arXiv: 2403.02187v2, submitted to *NeurIPS 2024*

Conferences

- The Twelfth International Conference on Learning Representations, Vienna Austria, May 7-11, 2024 (poster session)
- 66th All-Russian Scientific Conference of MIPT, April 2024 (oral talk)
- All-Russian Summer School on Machine Learning SMILES-2023, Altai, August 20-31, 2023 (poster session, received *“Best poster“ prize*)
- 65th All-Russian Scientific Conference of MIPT, April 2023 (oral talk)
- AIRI Conference on Artificial Intelligence, Sirius University of Science and Technology, July 18-26, 2022 (poster session)

Chapter 4
Literature review

In this chapter, we provide a brief overview of modern entropy and MI estimators that achieve a relative practical success in dealing with the curse of dimensionality. We provide reasoning why we choose MINE [4] as the only modern MI estimator among the mentioned in the Introduction to compare our results with.

- **MINE** is widely considered as a good benchmark estimator and featured in several recent works [41, 28, 37, 38]. As MINE is a neural estimator, it is theoretically able to grasp latent structure of data, thus performing compression implicitly.
- **Other lower/upper bounds and surrogate objectives.**
  - According to [37], not many methods in question outperform MINE. In fact, among the other methods mentioned in [37], only the difference of entropies (DoE) estimator achieves good results during a standard correlated Gaussians test. Unfortunately, DoE requires good parametrized and differentiable (by parameters) estimates of two PDFs, which is difficult to achieve in the case of complex multidimensional distributions.
  - According to an another overview [41], the methods in question have various significant trade-offs. Some of them require parts of the original distribution (like $\rho_{Y|X}$) or even some information-theoretic quantities (like $h(X)$) to be tractable, which is not achievable without utilizing special kinds of stochastic NNs. The others heavily rely on fitting a critic function to partially reproduce the original distribution, which leads to a poor bias-variance trade-off (it is illustrated by the results of these estimators in a standard correlated Gaussians test, see Figure 2 in [41]).
  - Compared to autoencoders, critic networks in methods in question are usually unstable and hard to train, see experiments in [41, 37]. We also have witnessed this instability while conducting experiments with MINE, see the attached source code.

  We, however, note that all these methods are of great use for building information-theoretic training objectives (as they are differentiable and usually represent upper or lower bounds).

  In addition to the limitations mentioned above, we would like to note that the relative practical success of the modern NN-based MI estimators might be attributed to the data compression being performed implicitly.

  In the work [41] it has been shown that other complex parametric NN-based estimators (NJW, JS, InfoNCE, etc.) exhibit poor performance during the estimation of MI between a pair of 20-dimensional incompressible (i.e., not lying along a manifold) synthetic vectors. These vectors, however, are of much simpler structure than the synthetic datasets used in our work (Gaussian vectors and $x \mapsto x^3$ mapping applied to Gaussian vectors in [41] versus high-dimensional images of geometric shapes and functions in our work). We interpret this phenomenon as a practical manifestation of the universal problem of MI estimation, which also affects the performance of modern NN-based MI estimators in the case of hard-to-compress data.

- **EDGE** [39] is a generalized version of the original binning estimator proposed in [49]: the binning operation is replaced by a more general hashing. We suppose that this method suffers from the same problems revealed in [21], unless a special hashing function admitting manifold-like or cluster-like structure of complex high-dimensional data is used.

- **Other definitions of entropy**. We are interested in fitting-compression hypothesis [55, 49] which is formulated for the classical mutual information, so other definitions are not appropriate for this particular task. We also note that the classical theory of information is well-developed and provides rigorous theoretical results (e.g., data processing inequality, which we used to prove Statement 3).

- We also mention the approach proposed in [1], where $h(L \mid X)$ is computed via a closed-form formula for Gaussian distribution and Monte-Carlo sampling. However, we note the following drawbacks of this method: (a) a closed-form formula is applicable to the entropy estimation only for the first stochastic NN layer, (b) a general-case estimator still has to be utilized to estimate $h(L)$ (in the work [1], the plug-in estimator from [21] is used; this estimator also suffers from the curse of dimensionality).

Chapter 5
Problem statement

Let $X : \Omega \to \mathbb{R}^n$ and $Y : \Omega \to \mathbb{R}^m$ be random vectors, where Ω represents the sample space. Let us suppose that these random vectors have absolutely continuous probability density functions (PDF) denoted as $\rho(x)$, $\rho(y)$, and $\rho(x, y)$, respectively, where the latter refers to the joint probability density function. The differential entropy of X is defined as follows

$$h(X) = -\mathbb{E}\log\rho(x) = -\int\limits_{\operatorname{supp} X} \rho(x)\log\rho(x)\,dx,$$

where $\operatorname{supp} X \subseteq \mathbb{R}^n$ represents the *support* of X, and $\log(\cdot)$ denotes the natural logarithm. Similarly, we define the joint differential entropy as $h(X, Y) = -\mathbb{E}\log\rho(x, y)$ and conditional differential entropy as $h(X \mid Y) = -\mathbb{E}\log\rho\,(X|Y) = -\mathbb{E}_Y\left(\mathbb{E}_{X|Y=y}\log\rho(X \mid Y = y)\right)$. Finally, the Mutual Information (MI) is given by

$$I(X; Y) = h(X) - h(X \mid Y),$$

and the following equivalences hold

$$I(X; Y) = h(X) - h(X \mid Y) = h(Y) - h(Y \mid X), \tag{5.1}$$

$$I(X; Y) = h(X) + h(Y) - h(X, Y). \tag{5.2}$$

Note that $\operatorname{supp} X$ or $\operatorname{supp} Y$ (density supports) may have measure zero, indicating a singular distribution. In such cases, if the supports are manifolds, probability density functions can be treated as induced probability densities, and dx and dy can be seen as area elements of the corresponding manifolds. Hence, all the above mentioned definitions remain valid.

Alternatively, Kullback-Leibler divergence $D_{KL}(\,\cdot\,)$ can be used to define the Mutual Information:

$$I(X; Y) = \int \rho(x, y)\log\frac{\rho(x, y)}{\rho(x)\rho(y)}dxdy = D_{KL}(\rho(x, y)||\rho(x)\rho(y)).$$

In this discussion, we employ a crucial property of the mutual information: its ability to remain unchanged under nonsingular mappings between smooth manifolds. Next statement demonstrates the possibility to measure the mutual information between compressed representations of random vectors.

Statement 1. *Let $\xi\colon \Omega \to \mathbb{R}^{n'}$ be an absolutely continuous random vector, and let $f\colon \mathbb{R}^{n'} \to \mathbb{R}^n$ be an injective piecewise-smooth mapping with Jacobian J_f,satisfying $n \geq n'$ and* $\det\left(J_f^T J_f\right) \neq 0$ *almost everywhere. Let either η be a discrete random variable, or (ξ, η) be an absolutely continuous random vector. Then*

$$I(\xi; \eta) = I\left(f(\xi); \eta\right) \tag{5.3}$$

Remark 1. *In what follows by $\xi\colon \Omega \to \mathbb{R}^{n'}$ we denote the compressed representation of X, $n' \leq n$.*

Proof. For any function f, let us denote $\sqrt{\det\left(J_f^T(x)J_f(x)\right)}$ (area transformation coefficient) by

$\alpha_f(x)$ where it exists.

Foremost, let us note that in both cases, $\rho_\xi(x \mid \eta)$ and $\rho_{f(\xi)}(x' \mid \eta) = \rho_\xi(x \mid \eta)/\alpha_f(x)$ exist. Hereinafter, we integrate over $\operatorname{supp} \xi \cap \{x \mid \alpha_f(x) \neq 0\}$ instead of $\operatorname{supp} \xi$; as $\alpha_f \neq 0$ almost everywhere by the assumption, the values of the integrals are not altered.

Let us use definitions of the differential entropy and the conditional differential entropy:

$$\begin{aligned} h(f(\xi)) &= -\int \frac{\rho_\xi(x)}{\alpha_f(x)} \log\left(\frac{\rho_\xi(x)}{\alpha_f(x)}\right) \alpha_f(x)\, dx = \\ &= -\int \rho_\xi(x) \log\left(\rho_\xi(x)\right) dx + \int \rho_\xi(x) \log\left(\alpha_f(x)\right) dx = \\ &= h(\xi) + \mathbb{E} \log \alpha_f(\xi). \end{aligned}$$

$$\begin{aligned} h(f(\xi) \mid \eta) &= \mathbb{E}_\eta \left(-\int \frac{\rho_\xi(x \mid \eta)}{\alpha_f(x)} \log\left(\frac{\rho_\xi(x \mid \eta)}{\alpha_f(x)}\right) \alpha_f(x)\, dx\right) = \\ &= \mathbb{E}_\eta \left(-\int \rho_\xi(x \mid \eta) \log\left(\rho_\xi(x \mid \eta)\right) dx + \int \rho_\xi(x \mid \eta) \log\left(\alpha_f(x)\right) dx\right) = \\ &= h(\xi \mid \eta) + \mathbb{E} \log \alpha_f(\xi) \end{aligned}$$

Finally, by the MI definition,

$$I(f(\xi); \eta) = h(f(\xi)) - h(f(\xi) \mid \eta) = h(\xi) - h(\xi \mid \eta) = I(\xi; \eta).$$

□

Recall that we utilize the stochastic neural network (NN) approach to address the problem of infinite mutual information $I(X; f(X))$ for a deterministic mapping f. As shown in [21], introducing stochasticity enables proper MI estimation between outputs of neural network layers. The stochastic modification of a network serves as a proxy to determine the information-theoretic properties of the original model.

A conventional feedforward neural network can be defined as an acyclic computational graph that can be topologically sorted:

$$L_0 \triangleq X, \quad L_1 := f_1(L_0), \quad L_2 := f_2(L_0, L_1), \quad \ldots, \quad \hat{Y} \triangleq L_n := f_n(L_0, \ldots, L_{n-1}),$$

where $L_0, \ldots, L_n$ denote the outputs of the network's layers. The stochastic modification is defined similarly, but using the Markov chain stochastic model:

Definition 1. *The sequence of random vectors $L_0, \ldots, L_n$ is said to form a* stochastic neural network *with input X and output $\hat{Y}$, if $L_0 \triangleq X$, $\hat{Y} \triangleq L_n$, and*

$$L_0 \longrightarrow (L_0, L_1) \longrightarrow \ldots \longrightarrow (L_0, \ldots, L_n)$$

is a Markov chain; L_k represents outputs of the k-th layer of the network.

Our primary objective is to track $I(L_i; L_j)$ during the training process. In the subsequent sections, we assume the manifold hypothesis to hold for X. In such case, under certain additional circumstances (continuity of f_k, small magnitude of injected stochasticity) this hypothesis can also be assumed for L_k, thereby justifying the proposed method.

Hence, the main goals of this research are following:

- create the approach for the MI estimation between high-dimensional random variables
- provide the Information Bottleneck analysis for real neural networks via the proposed method

Chapter 6
Methodology

In this section, we explore the application of lossless and lossy compression to estimation of MI between high-dimensional random vectors. We mention the limitations of conventional MI estimators, propose and theoretically justify a complementary lossy compression step to address the curse dimensionality, and derive theoretical bounds on the MI estimate under lossy compression.

6.1 Mutual information estimation

Let $\{(x_k, y_k)\}_{k=1}^N$ be a sequence of samples from the joint distribution of random vectors X and Y. Our goal is to estimate the mutual information between X and Y, denoted as $I(X;Y)$, based on these samples. The most straightforward way to achieve this is to estimate all the components in equation 5.1 or equation 5.2 via entropy estimators.

In this work, we make the assumption of the manifold hypothesis [17], which posits that data lie along or close to some manifold in multidimensional space. This hypothesis is believed to hold for a wide range of structured data, and there are datasets known to satisfy this assumption precisely (e.g., photogrammetry datasets, as all images are parametrized by camera position and rotation). In this study, we adopt a simplified definition of the manifold hypothesis:

Definition 2. *A random vector $X \colon \Omega \to \mathbb{R}^n$ strictly satisfies the manifold hypothesis iff there exist $\xi \colon \Omega \to \mathbb{R}^{n'}$ and $f \colon \mathbb{R}^{n'} \to \mathbb{R}^n$ satisfying the conditions of Statement 1, such that $X = f(\xi)$. A random vector $X' \colon \Omega \to \mathbb{R}^n$ loosely satisfies the manifold hypothesis iff $X' = X + Z$, where X strictly satisfies the manifold hypothesis, and Z is insignificant in terms of some metric.*

To overcome the curse of dimensionality, we propose learning the manifold with autoencoders [32, 26] and applying conventional estimators to the compressed representations.To address the issue of measure-zero support, we consider the probability measure induced on the manifold.

Let us consider an absolutely continuous X, compressible via autoencoder $A = D \circ E$.

Corollary 1. *Let $E^{-1} \colon \mathbb{R}^{n'} \supseteq E(\operatorname{supp} X) \to \mathbb{R}^n$ and $E(X) \colon \Omega \to \mathbb{R}^{n'}$ exist, let $(E^{-1} \circ E)(X) \equiv X$, let X, Y, $E(X)$ and E^{-1} satisfy conditions of the Statement 1. Then*

$$I(X;Y) = I(E(X);Y).$$

Proof.

$$I(X;Y) = \underbrace{I\left((E^{-1} \circ E)(X);Y\right) = I\left(E(X);Y\right)}_{\text{from the Statement 1}}$$

□

In case of absolutely continuous (X,Y), the mutual information estimate can be defined as follows:

$$\hat{I}(X;Y) \triangleq \hat{h}(E(X)) + \hat{h}(Y) - \hat{h}(E(X),Y) \tag{6.1}$$

In case of absolutely continuous X and discrete Y, it is impractical to use equation 5.2, as the (induced) joined probability distribution is neither absolutely continuous nor discrete. However, equation 5.1 is still valid:

$$h(X \mid Y) = \sum_{y \in \operatorname{supp} Y} p_Y(y) \cdot \underbrace{\left[- \int \rho_X(x \mid Y = y) \log \left(\rho_X(x \mid Y = y) \right) \, dx \right]}_{h(X \mid Y = y)}$$

Probabilities p_Y can be estimated using empirical frequencies: $\hat{p}_Y(y) = \frac{1}{N} \cdot |\{k \mid y_k = y\}|$. Conditional entropy $h(X \mid Y = y)$ can be estimated using corresponding subsets of $\{x_k\}_{k=1}^N$: $\hat{h}(X \mid Y = y) = \hat{h}(\{x_k \mid y_k = y\})$. The mutual information estimate in this case can be defined as follows:

$$\hat{I}(X;Y) \triangleq \hat{h}(E(X)) - \sum_{y \in \operatorname{supp} Y} \hat{p}_Y(y) \cdot \hat{h}(E(X) \mid Y = y) \tag{6.2}$$

According to the strong law of large numbers, $\hat{p} \xrightarrow{\text{a.s.}} p$. That is why the convergence of the proposed MI estimation methods solely relies on the convergence of the entropy estimator used in equation 6.1 and equation 6.2. Note that this method can be obviously generalized to account for compressible Y.

6.2 Bounds for mutual information estimate

It can be shown that it is not possible to derive non-trivial bounds for $I(E(X);Y)$ in general case if the conditions of Corollary 1 do not hold. Let us consider a simple linear autoencoder that is optimal in terms of mean squared error, such as principal component analysis-based autoencoder. The following statement demonstrates cases where the proposed method of estimating mutual information through lossy compression fails.

Statement 2. *For any given $\varkappa \geq 0$ there exist random vectors $X\colon \Omega \to \mathbb{R}^n$, $Y\colon \Omega \to \mathbb{R}^m$, and a non-trivial linear autoencoder $A = D \circ E$ with latent space dimension $n' < n$ that is optimal in terms of minimizing mean squared error $\mathbb{E}\,\|X - A(X)\|^2$, such that $I(X;Y) = \varkappa$ and $I(E(X);Y) = 0$.*

Proof. Let us consider the following three-dimensional Gaussian vector $(X_1, X_2, Y) \triangleq (X, Y)$:

$$X \sim \mathcal{N}\left(0, \begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix}\right), \qquad Y \sim \mathcal{N}(0,1) \qquad (X_1, X_2, Y) \sim \mathcal{N}\left(0, \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sigma & a \\ 0 & a & 1 \end{bmatrix}\right),$$

where $\operatorname{cov}(X_2, Y) = a \triangleq \sqrt{1 - e^{-2\varkappa}}$, $\operatorname{cov}(X_1, Y) = 0$ (so X_1 and Y are independent). Let the intrinsic dimension be $n' = 1$, and $\sigma < 1$. According to principal component analysis, the optimal linear encoder is defined up to a scalar factor by the equality $E(X) = X_1$. However, $I(X;Y) = -\frac{1}{2}\ln(1 - a^2) = \varkappa$ (see the Statement 8), but $I(E(X);Y) = 0$, as X_1 and Y are independent. □

This statement demonstrates that an arbitrary amount of information can be lost through compression of the data. It arises from the fact that “less significant” in terms of metric spaces does not align with “less significant” in terms of information theory. However, with additional assumptions, a more useful theoretical result can be obtained.

Statement 3. *Let X, Y, and Z be random variables such that $I(X;Y)$ and $I((X,Z);Y)$ are defined. Let f be a function of two arguments such that $I(f(X,Z);Y)$ is defined. If there exists a function g such that $X = g(f(X,Z))$, then the following chain of inequalities holds:*

$$I(X;Y) \leq I(f(X,Z);Y) \leq I((X,Z);Y) \leq I(X;Y) + h(Z) - h(Z \mid X, Y)$$

Proof. According to data processing inequality [12], $I\left(f(X,Z);Y\right) \leq I\left(X,Z;Y\right)$. As $I(X;Y) = I\left(g(f(X,Z));Y\right)$, $I(X;Y) \leq I\left(f(X,Z);Y\right)$.

Note that as DPI is optimal, additional assumptions on f, X, Y and Z are required to tighten the bounds.

The last inequality is derived via the following equations from [12]:

$$
\begin{aligned}
I(X,Z;Y) &= I(X;Y) + I(Y;Z \mid X) \\
I(X,Y;Z) &= I(X;Z) + I(Y;Z \mid X)
\end{aligned}
$$

As $I(X;Z) \geq 0$,

$$
\begin{aligned}
I(X;Y) + I(X,Y;Z) = I(X;Y) + I(Y;Z \mid X) + I(X;Z) \geq \\
\geq I(X;Y) + I(Y;Z \mid X) = I(X,Z;Y)
\end{aligned}
$$

Finally, recall that $I(X,Y;Z) = h(Z) - h(Z \mid X,Y)$. □

In this context, $f(X,Z)$ can be interpreted as compressed noisy data, X as denoised data, and g as a perfect denoising decoder. The term $h(Z)$ can be upper-bounded via entropy of Gaussian distribution of the same variance, $h(Z \mid X,Y)$ can be lower-bounded in special cases (e.g., when Z is a sum of independent random vectors, at least one of which is of finite entropy); see Section 6.3 of this chapter for details. We also note the special case where the data lost by compression can be considered as independent random noise.

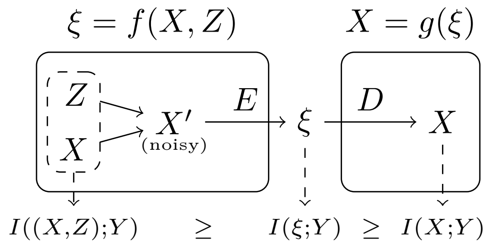


Figure 1: Conceptual scheme of Statement 3 in application to lossy compression with autoencoder $A = D \circ E$.

Corollary 2. *Let X, Y, Z, f, and g satisfy the conditions of the Statement 3. Let also random variables (X,Y) and Z be independent. Then $I(X;Y) = I\left(f(X,Z);Y\right)$.*

Proof. Since (X,Y) and Z are independent, $I\left(X,Z;Y\right) = I(X;Y)$, which implies $I(X;Y) = I\left(f(X,Z);Y\right)$ according to the Statement 3. □

We note that (a) the presented bounds cannot be further improved unless additional assumptions are made (e.g., linearity of f in [22]); (b) additional knowledge about the connection between X, Y, and Z is required to properly utilize the bounds. Other bounds can also be derived [46, 4, 41], but they do not take advantage of the compression aspect.

The provided theoretical analysis and additional results from Section 6.3 of this chapter show that the proposed method allows for tracking the true value of MI within the errors of a third-party estimator ran on compressed data and the derived bounds imposed by the compression itself.

6.3 Entropy bounds

In this section, we provide several theoretical results that complement the bounds proposed in Section 6.2. The following inequalities can be used to bound the entropy terms in Statement 3.

Statement 4 ([12], Theorem 8.6.5)**.** *Let X be a random vector with covariance matrix R. Then $h(X) \leq h(\mathcal{N}(0,R))$.*

Statement 5. *Let $X, Z \colon \Omega \to \mathbb{R}^n$ be independent random vectors. Then $h(X+Z) \geq h(Z)$.*

Proof. Recall that

$$h(X, X+Z) = h(X+Z) + h(X \mid X+Z) = h(X) + h(X+Z \mid X),$$

from which the following is derived:

$$h(X+Z) = h(X) + h(X+Z \mid X) - h(X \mid X+Z)$$

Note that $h(X+Z \mid X) = \mathbb{E}_X\, h(x+Z \mid X=x) = h(Z \mid X)$. As X and Z are independent, $h(Z \mid X) = h(Z)$. Thus, we derive the following:

$$h(X+Z) = h(X) + h(Z) - h(X \mid X+Z) = h(Z) + \underbrace{I(X; X+Z)}_{\geq 0} \geq h(Z)$$

□

Statement 6. *Let $X\colon \Omega \to \mathbb{R}^{n\times n}$ and $Z\colon \Omega \to \mathbb{R}^n$ be a random matrix and vector, correspondingly. Let X and Z be independent. Then $h(X \cdot Z) \geq h(Z) + \mathbb{E}\left(\ln|\det X|\right)$.*

Proof. Note that $h(X \cdot Z \mid X) = \mathbb{E}_X\, h(x \cdot Z \mid X = x) = h(Z \mid X) + \mathbb{E}\left(\ln|\det X|\right)$. The rest of the proof is the same as for Statement 5:

$$\begin{aligned} h(X \cdot Z) = h(X) + h(Z) + \mathbb{E}\left(\ln|\det X|\right) - h(X \mid X \cdot Z) = \\ = h(Z) + \underbrace{I(X; X \cdot Z)}_{\geq 0} + \mathbb{E}\left(\ln|\det X|\right) \geq h(Z) + \mathbb{E}\left(\ln|\det X|\right). \end{aligned}$$

□

Corollary 3. *Let $X, Z\colon \Omega \to \mathbb{R}^n$ be independent random vectors. Then $h(X \odot Z) \geq h(Z) + \sum_{i=1}^n \mathbb{E}\left(\ln|X_i|\right)$, where $\odot$ is an element-wise product.*

Proof. Note that $X \odot Z = \mathrm{diag}(X) \cdot Z$, and $\log|\det \mathrm{diag}(X)| = \sum_{i=1}^n \ln|X_i|$. We then apply Statement 6. □

Note that entropy terms in Statements 5 and 6 can be conditioned. The independence requirement should then be replaced by independence under corresponding conditions.

We also note that Statement 4 can utilize autoencoder reconstruction error (via error covariance matrix), and Statements 5, 6 – magnitude of random vector and injected noise, which is of particular use, as this information is easily accessible in a typical experimental setup.

Practical use cases include using Statement 5 when stochasticity is introduced via additive noise (e.g., [21]) and Corollary 3 when stochasticity is introduced via multiplicative noise (e.g., [1]).

6.4 Classical entropy estimators

In this section, we provide definitions of conventional entropy estimators used to conduct the experiments, as well as provide proofs that these estimators fail in case of high-dimensional data.

6.4.1 Kernel density estimation

In the proposed method of mutual information estimation, the estimation of the probability density function for codes c_k in the latent space plays a crucial role. There are many methods(e.g., [57, 31, 6]) available for probability density function estimation, including kernel density estimation (KDE).

KDE is a non-parametric method that uses a kernel function to estimate the probability distribution of a given dataset. The kernel function assigns a weight to each data point based on its proximity to other points in the dataset. By summing these weights over all data points, we can obtain an estimate of the underlying probability density function.

The choice of kernel function and bandwidth parameter is critical for KDE. Different kernels and bandwidths can result in different estimates of the density function, affecting the accuracy and robustness of the results. In the context of entropy estimation, KDE can be used to estimate the joint probability density of two or more random variables, which is essential for calculating mutual information.

To apply KDE to entropy estimation, we need to choose a suitable kernel function and determine the optimal bandwidth. Common choices for kernel functions include Gaussian, Epanechnikov, and triangular kernels. The bandwidth parameter controls the smoothness of the estimated density function and should be chosen carefully to balance between underfitting and overfitting.

Once the kernel function and bandwidth are selected, we can use KDE to estimate the joint density of the two random variables involved in the mutual information calculation. This allows us to approximate the entropy of one variable conditioned on another, which is one of key steps in estimating mutual information.

Let $\rho_{X,k}(x)$ be a density estimate at a point x, which is obtained from the sampling $\{x_k\}_{k=1}^{N}$ without the k-th element by KDE with the kernel K. We get the following expression for the density:

$$\hat{\rho}_{b,-k}(x) = \frac{1}{b^n\,(N-1)} \sum_{\substack{l=0 \\ l\neq k}}^{N} K\left(\frac{x - x_l}{b}\right) \tag{6.3}$$

Here, $K(x) = \dfrac{1}{(2\pi)^{n/2}} \exp\left(-\dfrac{\|x\|^2}{2}\right)$ is a standard Gaussian kernel. Hereinafter, it is possible to use any kernel with infinite support as K, but the Gaussian one is preferable because of its light tails and infinite differentiability.

The entropy estimate is obtained via Leave-One-Out method. Densities at each sample x_k are calculated according to the formula 6.3.

$$\hat{H}(X) = \frac{1}{N} \sum_{k=1}^{N} \log \hat{\rho}_{b,-k}(x_k) \tag{6.4}$$

Maximum-likelihood

The optimal bandwidth can be selected in accordance with the minimization of the Kullback-Leibler divergence between the estimated distributions and the empirical one ($\hat{\rho}_{\text{emp}}(x) = \frac{1}{N}\sum_{k=1}^{N} \delta(x - x_k)$). This is equivalent to selecting the bandwidth as a maximum likelihood estimate:

$$\hat{b} = \arg\max_{b} \hat{H}(X) = \arg\max_{b} \sum_{k=1}^{N} \log \hat{\rho}_{b,-k}(x_k) \tag{6.5}$$

The experiments have shown that this method tends to underestimate mutual information, and the difference increases with an increasing true value of mutual information.

Asymptotic: *the entropy estimation and bandwidth selection take $\mathcal{O}\left(n \log n\right)$, compression takes $\mathcal{O}\left(n\right)$, resulting in a total time complexity of $\mathcal{O}\left(n \log n\right)$*

Least Squares Error

Now let us consider the Least Square Cross Validation method (see [56, 45]). In this method, bandwidth selection is based on the minimization of the mean squared error between the exact density and the corresponding kernel density estimate. We minimize the following expression:

$$ISE(b) = \int\limits_{\mathbb{R}^n} (\hat{\rho}_b(x) - \rho(x))^2 \, dx$$

Here, ρ is the true probability density function, and $\hat{\rho}_b(x)$ is the estimate with the bandwidth b:

$$\hat{\rho}_b(x) = \frac{1}{b^n N} \sum_{k=1}^{N} K\left(\frac{x - x_k}{b}\right)$$

Since the true distribution is unknown, we substitute ρ with $\hat{\rho}_{\text{emp}}$. This leads to the following objective function to be minimized:

$$\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} J_b(x_i - x_j) - \frac{2}{N} \sum_{i=1}^{N} \hat{\rho}_{b,-i}(x_i),$$

where

$$J_b(\xi) = \int\limits_{\mathbb{R}^n} \frac{1}{b^{2n}} K\left(\frac{x}{b}\right) K\left(\frac{x - \xi}{b}\right) dx,$$

which can be computed via the Fourier transform.

Asymptotic: *The entropy estimation takes $\mathcal{O}(n \log n)$, compression takes $\mathcal{O}(n)$, same as KDE ML. However, the optimal bandwidth selection takes $\mathcal{O}(n^2)$ due to the quadratic complexity of the minimized objective. Therefore, this algorithm has a total time complexity of $\mathcal{O}(n^2)$, making KDE LSE asymptotically the slowest algorithm implemented within this research.*

6.4.2 Kozachenko-Leonenko

There is another method of entropy estimation, which was proposed by Kozachenko and Leonenko in [31]. The main feature of this method is that it uses k-nearest neighbor density estimation instead of kernel density estimation (KDE). In this section, we consider two variations of this nearest neighbors based method.

Non-weighted Kozachenko-Leonenko

Let $\{x_k\}_{k=1}^N \subseteq \mathbb{R}^n$ be the sampling of random vector X. Let us denote $\hat{r}(x) = \min\limits_{1 \le k \le N} r(x, x_k)$ the distance to the nearest neighbour using the metric r (by default, r is Euclidean metric).

According to [31], the density estimation at x is given by:

$$\hat{\rho}(x) = \frac{1}{\gamma \cdot \hat{r}(x)^n \cdot c_1(n) \cdot (N-1)},$$

where $c_1(n) = \pi^{n/2} / \Gamma(n/2 + 1)$ is a unit n-dimensional ball volume and γ is a constant which makes the entropy estimate unbiased ($\ln \gamma = c_2 \approx 0.5772$ is the Euler constant).

Asymptotic: *the entropy estimation takes $\mathcal{O}(n \log n)$, compression takes $\mathcal{O}(n)$, resulting in a total time complexity of $\mathcal{O}(n \log n)$.*

Weighted Kozachenko-Leonenko

The main drawback of the conventional Kozachenko-Leonenko estimator is the bias that occurs in dimensions higher than 3. This issue can be addressed by using weighted nearest neighbors estimation. A modified estimator is proposed in [6]:

$$\hat{H}_N^w = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} w_i \log \xi_{(j),i}$$

where w is the weight vector, $\xi_{(j),i} = e^{-\Psi(j)} \cdot c_1(n) \cdot (N-1) \cdot \rho_{(j),i}^d$, Ψ denotes the digamma function. We choose the weight vector $w = (w_1, \ldots, w_k)$ as follows. For $k \in \mathbb{N}$ let

$$\mathcal{W}^{(k)} = \Bigg\{ w \in \mathbb{R}^k : \sum_{j=1}^{k} w_j \cdot \frac{\Gamma\left(j + \frac{2\ell}{n}\right)}{\Gamma(j)} = 0 \text{ for } \ell = 1, \ldots, \left\lfloor \frac{n}{4} \right\rfloor, \\ \sum_{j=1}^{k} w_j = 1 \text{ and } w_j = 0 \text{ if } j \notin \left\{ \left\lfloor \frac{k}{n} \right\rfloor, \left\lfloor \frac{2k}{n} \right\rfloor, \ldots, k \right\} \Bigg\}$$

and let the w be a vector from $\mathcal{W}^{(k)}$ with the least l_2-norm.

Asymptotic: *the entropy estimation takes $\mathcal{O}\left(n \log n\right)$, weight selection – $\mathcal{O}\left(k^3\right) = \mathcal{O}\left(1\right)$, compression – $\mathcal{O}\left(n\right)$, resulting in a total time complexity of $\mathcal{O}\left(n \log n\right)$.*

Chapter 7
Numerical experiments

7.1 Synthetic dataset generation

In order to test the proposed mutual information estimator, we developed an universal method for synthetic dataset generation with defined information-theoretic properties. This method yields two random vectors, X and Y, with a predefined value of mutual information $I(X;Y)$. The method requires X and Y to be images of normally distributed vectors under known nonsingular smooth mappings. The generation consists of two steps. First, a normal vector $(\xi, \eta) \sim \mathcal{N}(0, M)$ is considered, where $\xi \sim \mathcal{N}(0, I_{n'})$, $\eta \sim \mathcal{N}(0, I_{m'})$, and n', m' are dimensions of ξ and η, respectively. The covariance matrix M is chosen to satisfy $I(\xi;\eta) = \varkappa$, where $\varkappa$ is an arbitrary non-negative constant.

Statement 7. *For every $\varkappa \geq 0$ and every $n', m' \in \mathbb{N}$ exists a matrix $M \in \mathbb{R}^{(n'+m') \times (n'+m')}$ such that $(\xi, \eta) \sim \mathcal{N}(0, M)$, $\xi \sim \mathcal{N}(0, I_{n'})$, $\eta \sim \mathcal{N}(0, I_{m'})$ and $I(\xi;\eta) = \varkappa$.*

Proof. We divide the proof of this statement into the following statements:

Statement 8. *Let $(\xi, \eta) \sim \mathcal{N}(0, M)$ be a Gaussian pair of (scalar) random variables with unit variance such that $I(\xi;\eta) = \varkappa$. Then*

$$M = \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix}, \qquad a = \sqrt{1 - e^{-2\varkappa}} \tag{7.1}$$

Proof. Differential entropy of multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ is $h = \frac{1}{2} \ln\left(\det\left(2\pi e \cdot \Sigma\right)\right)$. This and equation 5.2 leads to the following:

$$\varkappa = I(\xi;\eta) = \frac{1}{2}\ln(2\pi e) + \frac{1}{2}\ln(2\pi e) - \frac{1}{2}\ln\left((2\pi e)^2 \cdot (1 - a^2)\right) = -\frac{1}{2}\ln\left(1 - a^2\right)$$

$$a = \sqrt{1 - e^{-2\varkappa}}$$

□

Statement 9. *Let ξ and η be independent random variables. Then $I(\xi;\eta) = 0$, $h(\xi, \eta) = h(\xi) + h(\eta)$.*

Proof. We consider only the case of absolutely continuous ξ. As ξ and η are independent, $\rho_\xi(x \mid \eta = y) = \rho_\xi(x)$. That is why $I(\xi;\eta) = h(\xi) - h(\xi \mid \eta) = h(\xi) - h(\xi) = 0$, according to the definition of MI. The second equality is derived from equation 5.2. □

Corollary 4. *Let ξ_1, ξ_2 and η_1, η_2 be random variables, independent in the following tuples: (ξ_1, ξ_2), (η_1, η_2) and $((\xi_1, \eta_1), (\xi_2, \eta_2))$. Then $I\left((\xi_1, \xi_2); (\eta_1, \eta_2)\right) = I(\xi_1;\eta_1) + I(\xi_2;\eta_2)$*

Proof. From equation 5.2 and Statement 9 the following chain of equalities is derived:

$$\begin{aligned} I\left((\xi_1, \xi_2); (\eta_1, \eta_2)\right) &= h(\xi_1, \xi_2) + h(\eta_1, \eta_2) - h(\xi_1, \xi_2, \eta_1, \eta_2) = \\ &= h(\xi_1) + h(\xi_2) + h(\eta_1) + h(\eta_2) - h(\xi_1, \eta_1) - h(\xi_2, \eta_2) = \\ &= I(\xi_1; \eta_1) + I(\xi_2; \eta_2) \end{aligned}$$

□

The Statement 8 and Corollary 4 provide us with a trivial way of generating dependent normal random vectors with a defined mutual information. Firstly, we consider $\Xi \sim \mathcal{N}(0, M')$, where M' is a $(n' + m') \times (n' + m')$ block-diagonal matrix with blocks from equation 7.1. The number of blocks is $k = \min\{n', m'\}$ (other diagonal elements are units). The parameter $\varkappa$ for each block equals $I(\xi; \eta)/k$, where $I(\xi; \eta)$ is the desired mutual information of the resulting vectors. The components of Ξ are then rearranged to get $(\xi, \eta) \sim \mathcal{N}(0, M)$, where $\xi \sim \mathcal{N}(0, I_{n'})$ and $\eta \sim \mathcal{N}(0, I_{m'})$. The final structure of M is as follows:

$$M = \begin{array}{c} \left[\begin{array}{ccc|ccc} 1 & & & a & & \\ & 1 & & & a & \\ & & \ddots & & & \ddots \\ \hline a & & & 1 & & \\ & a & & & 1 & \\ & & \ddots & & & \ddots \end{array}\right] \\ \underbrace{\hphantom{1\quad 1\quad \ddots}}_{n'}\ \underbrace{\hphantom{a\quad a\quad \ddots}}_{m'} \end{array} \tag{7.2}$$

□

After generating the correlated normal random vectors (ξ, η) with the desired mutual information, we apply smooth nonsingular mappings to obtain $X = f(\xi)$ and $Y = g(\eta)$. According to Statement 1, this step preserves the mutual information, so $I(\xi; \eta) = I(X; Y)$.

7.2 Comparison of the entropy estimators

The MI estimate is acquired according to Subsection 6.1. To estimate the entropy terms in equation 5.1 or equation 5.2, we leverage conventional entropy estimators, such as kernel density-based [56, 46, 45] and Kozachenko-Leonenko estimators (original [31] and weighted [6] versions). To test the accuracy of these approaches, we use datasets sampled from synthetic random vectors with known MI. We generate these datasets in accordance with Section 7.1.

To examine the impact of the compression step proposed in Subsection 6.1, we utilize a special type of synthetic datasets. Synthetic data lies on a manifold of small dimension. This is achieved by generating a low-dimensional dataset and then embedding it into a high-dimensional space by a smooth mapping (so the Statement 1 can be applied). Then, the acquired datasets are compressed via autoencoders. Finally, the obtained results are fed into a mutual information estimator.

Algorithm 1 and Figure 1 describe the proposed mutual information estimation quality measurement. We run several experiments with f and g mapping normal distributions to rasterized images of geometric shapes (e.g., rectangles) or 2D plots of smooth functions (e.g., Gaussian functions).[1] The results are presented in Figures 2 and 3. The blue and green curves correspond to the estimates of MIs marked by the corresponding colors in Figure 1. Thus, we see that the compression

[1] Due to the high complexity of the used f and g, we do not define these functions in the main text; instead, we refer to the source code published along with the paper.

step does not lead to bad estimation accuracy, especially for the weighted Kozachenko-Leonenko (WKL) estimator, which demonstrates the best performance. Note that we do not plot estimates for uncompressed data, as all the four tested classical estimators completely fail to correctly estimate MI for such high dimensions; for more information, we refer to Section 7.3. We also conduct experiments with MINE (without compression), for which we train the critic network of the same complexity, as we use for the autoencoder.

Algorithm 1 Measure mutual information estimation quality on high-dimensional synthetic datasets

```
1: Generate two datasets of samples from normal vectors $\xi$ and $\eta$ with given mutual information
   as described in section7.1 – $\{(x_k, y_k)\}_{k=1}^N$.
2: Choose functions $f$ and $g$ satisfying conditions of the Statement 1 (so $I(\xi;\eta) = I(f(\xi);g(\eta))$)
   and obtain datasets for $f(\xi)$ and $g(\eta)$ – $\{f(x_k)\}_{k=1}^N$, $\{g(y_k)\}_{k=1}^N$.
3: Train autoencoders $A_X = D_X \circ E_X$, $A_Y = D_Y \circ E_Y$ on $\{f(x_k)\}$, $\{g(y_k)\}$ respectively.
4: Obtain datasets for $(E_X \circ f)(\xi)$ and $(E_Y \circ g)(\eta)$.
   We assume that $E_X$, $E_Y$ satisfy conditions of the Corollary 1, so we expect

       $I(\xi;\eta) = I(f(\xi);g(\eta)) = I\big((E_X \circ f)(\xi);(E_Y \circ g)(\eta)\big)$

5: Estimate $I\big((E_X \circ f)(\xi);(E_Y \circ g)(\eta)\big)$ and compare the estimated value with the exact one.
```

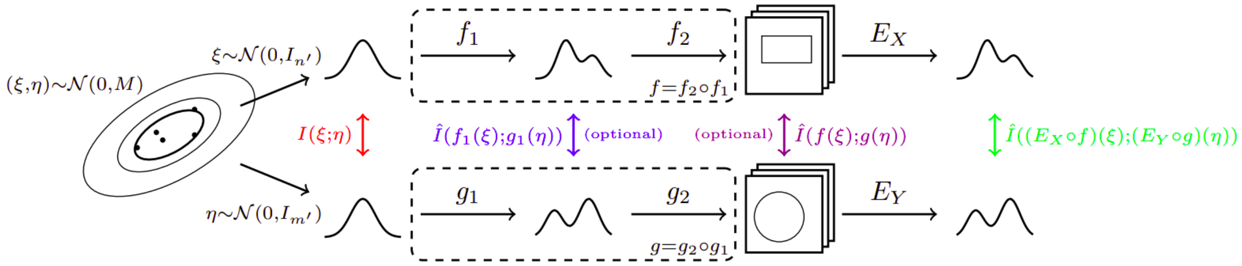
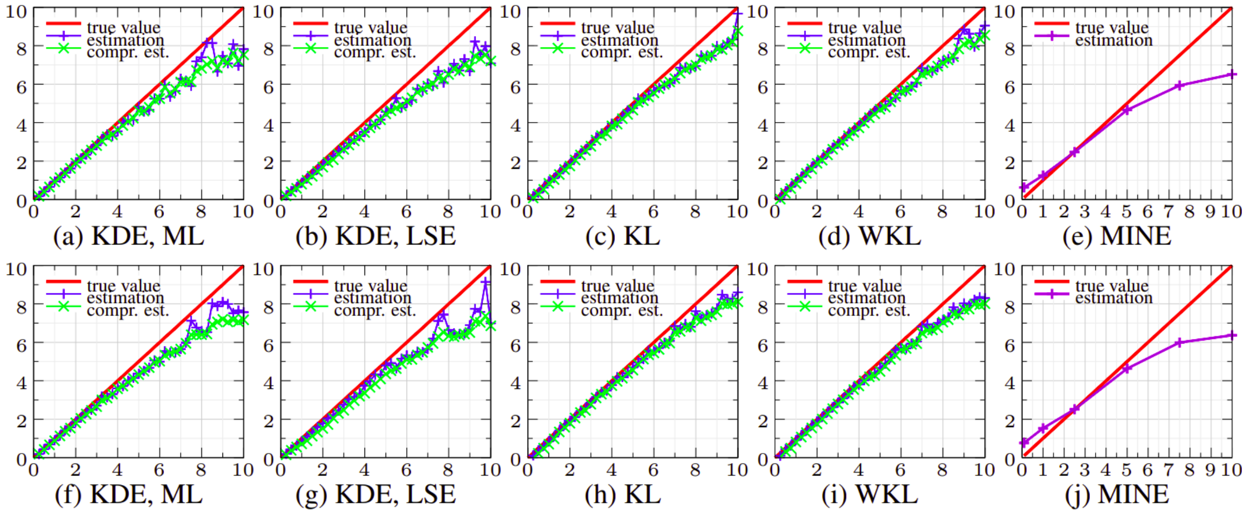
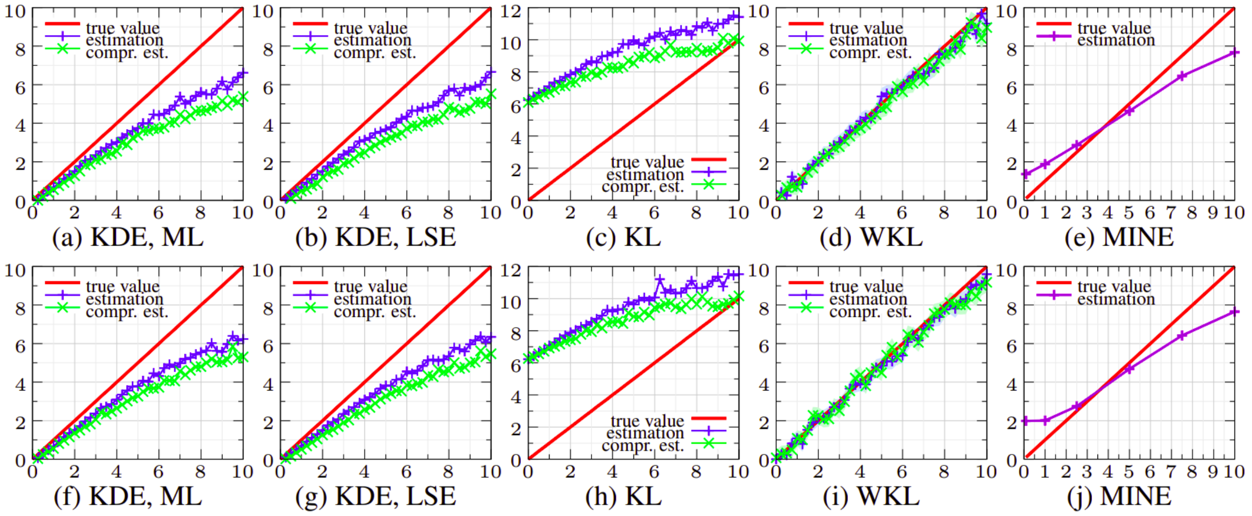

$(\xi,\eta)\sim\mathcal{N}(0,M)$
$\xi\sim\mathcal{N}(0,I_{n'})$
$\eta\sim\mathcal{N}(0,I_{m'})$
f_1
f_2
$f=f_2\circ f_1$
E_X
$I(\xi;\eta)$
$\hat{I}(f_1(\xi);g_1(\eta))$
(optional)
(optional)
$\hat{I}(f(\xi);g(\eta))$
$\hat{I}((E_X\circ f)(\xi);(E_Y\circ g)(\eta))$
g_1
g_2
$g=g_2\circ g_1$
E_Y

Figure 1: Conceptual scheme of Algorithm 1. In order to observe and quantify the loss of information caused by the compression step, we split $f\colon \mathbb{R}^{n'} \to \mathbb{R}^n$ into two functions: $f_1\colon \mathbb{R}^{n'} \to \mathbb{R}^{n'}$ maps ξ to a structured latent representation of X (e.g., parameters of geometric shapes), and $f_2\colon \mathbb{R}^{n'} \to \mathbb{R}^n$ maps latent representations to corresponding high-dimensional vectors (e.g., rasterized images of geometric shapes). The same goes for $g = g_2 \circ g_1$. Colors correspond to the Figures 2 and 3. For a proper experimental setup, we require f_1, f_2, g_1, g_2 to satisfy the conditions of Statement 1.


(a) KDE, ML (b) KDE, LSE (c) KL (d) WKL (e) MINE

(f) KDE, ML (g) KDE, LSE (h) KL (i) WKL (j) MINE

Figure 2: Maximum-likelihood and Least Squares Error KDE, Non-weighted and Weighted Kozachenko-Leonenko, MINE for 16×16 (first row) and 32×32 (second row) images of 2D Gaussians ($n' = m' = 2$), $5 \cdot 10^3$ samples. Along x axes is $I(X;Y)$, along y axes is $\hat{I}(X;Y)$.


(a) KDE, ML (b) KDE, LSE (c) KL (d) WKL (e) MINE

(f) KDE, ML (g) KDE, LSE (h) KL (i) WKL (j) MINE

Figure 3: Maximum-likelihood and Least Squares Error KDE, Non-weighted and Weighted Kozachenko-Leonenko, MINE for 16×16 (first row) and 32×32 (second row) images of rectangles ($n' = m' = 4$), $5 \cdot 10^3$ samples. Along x axes is $I(X;Y)$, along y axes is $\hat{I}(X;Y)$.

7.3 Limitations of classical entropy estimators

Although the entropy estimation is an example of a classical problem, it is still difficult to acquire estimates for high-dimensional data, as the estimation requires an exponentially (in dimension) large number of samples (see [20, 37]). As the mutual information estimation is tightly connected to the entropy estimation, this problem also manifests itself in our task. Although this difficulty affects every MI estimator, classical estimators may be assumed to be more prone to the curse of dimensionality, as they are usually too basic to grasp a manifold-like low-dimensional structure of high-dimensional data.

In this section, we provide experimental proofs of classical estimators' inability to yield correct MI estimates in the high-dimensional case. We utilize the same tests with images of 2D Gaussians used in Section 7.2 Figure 2. However, due to computational reasons, the size of the images is reduced to 4×4 and 8×8 (so the data is of even smaller dimension compared to Section 7.2). The results are presented int Table 1. For a comparison we also provide the results for

WKL estimator fed with the PCA-compressed data.

Table 1: MSE (in nats) of classical MI estimation methods ran on $5 \cdot 10^3$ images of 2D Gaussians. The character "–" denotes cases, in which the method failed to work due computational reasons (numerical overflows, ill-conditioned matrices, etc.).

Images size	KDE ML	KDE LSE	KL	WKL	WKL, PCA-compressed
4×4	$4.1 \cdot 10^3$	$1,95 \cdot 10^1$	$9,7$	$2,9 \cdot 10^1$	$1,87$
8×8	$2.8 \cdot 10^3$	–	$6,6 \cdot 10^1$	$7,5 \cdot 10^1$	$1,71$
16×16	–	–	–	–	$0,67$

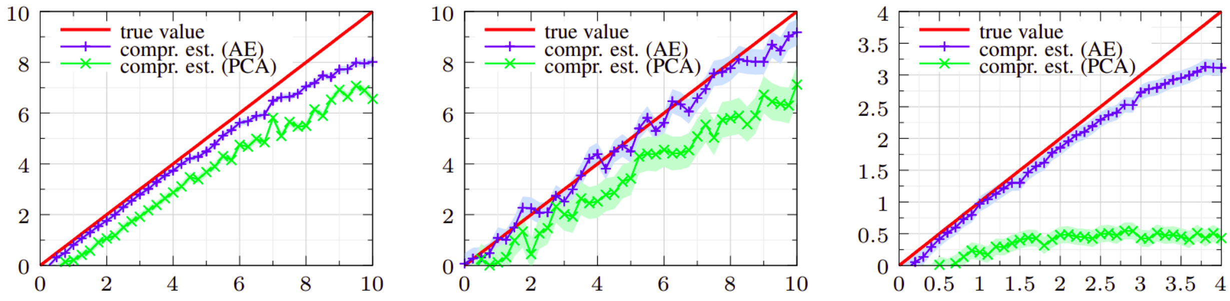
Note that although WKL estimator performs better in Section 7.2 due to lower bias, it is outperformed by the original KL estimator in the case of uncompressed data due to lower variance. However, this observation is not of great importance, as all the four methods perform extremely poorly in case of 8×8 images and bigger.

7.4 Limitations of the linear/nonlinear compression

This section devoted to the confirmation of the novelty of our contribution to the problem of high-dimensional MI estimation.

The linearity of the encoder E is one of the limitations of the proposed approach. The paper by the author, which forms the basis for this thesis, includes theoretical bounds for the absolute error of the estimated mutual information *only in the case of the linear compression* (see Section C in Appendix of [8]). Although the possibility of extending the approach to nonlinear dimensionality reduction methods is mentioned in this thesis, it is not clear whether the derived bounds can be directly applied to the nonlinear case. To achieve this, one needs to propose a nonlinear generalization of explained variance and provide a more general analysis of entropy alteration through discarding nonlinear components.

In this section, we conduct tests using synthetic data to demonstrate that the autoencoder-based approach outperforms the PCA-based method when dealing with nonlinear manifolds (see Figure 4). Although the difference is relatively small for the datasets used in Section 7.1, it is possible to provide an example of a highly nonlinear manifold where linear compression results in significant loss of accuracy (see the part (c) in Figure 4). The code for generating the synthetic data is provided in our code [7].



(a) 32×32 images of 2D Gaussians ($n' = m' = 2$)

(b) 32×32 images of rectangles ($n' = m' = 4$)

(c) Highly-nonlinear manifold in $\mathbb{R}^{32}$ ($n' = m' = 2$)

Figure 4: Comparison of nonlinear AE and linear PCA performance in task of MI estimation via lossy compression: $5 \cdot 10^3$ samples. Along x axes is $I(X;Y)$, along y axes is $\hat{I}(X;Y)$. WKL entropy estimator is used in these experiments

The experiments mentioned above confirm that the non-linearity of the encoder E is more versatile compared to the the linear compression. However, more accurate theoretical estimates have been obtained for the linear case.

7.5 Information flow in deep neural networks

This section is dedicated to the information flow estimation in DNNs via the proposed method. We estimate the information flow in a convolutional classifier of the MNIST handwritten digits dataset. This neural network is simple enough to be quickly trained and tested, but at the same time, is complex enough to suffer from the curse of dimensionality. The dataset consists of images of size $28 \times 28 = 784$ pixels. It was shown in [24] that these images have a relatively low latent space dimension, approximately 12–13. If the preservation of only the main features is desired, the latent space can even be narrowed down to 3–10. Although the proposed experimental setup is nowadays considered to be toy and small, it is still problematic for the IB-analysis, as it was shown in [21].

It can be concluded from the previous section that the weighted Kozachenko-Leonenko estimator is superior to the other methods tested in this paper. That is why it is used in experiments with the DNN classifier described in the current section. The analyzed network is designed to return the output of every layer. To avoid the problem of a deterministic relationship between input and output, we apply Gaussian dropout with a small variance after each layer. This allows for the better generalization during the training [51] and finite values of MI during the IB-analysis [1]. Lossy compression of input images X is performed via a convolutional autoencoder with a latent dimension of d_X^{latent}. Lossy compression of layer outputs L_i is performed via principal component analysis with $d_{L_i}^{\text{latent}}$ as the number of principal components, as it showed to be faster and not significantly worse than general AE approach in this particular case. The general algorithm is described in Algorithm 2.

Algorithm 2 Estimate information flow in the neural network during training

```
Compress the input dataset $\{x_k\}_{k=1}^N$ via the input encoder $E_X$: $c_k^X = E_X(x_k)$.
for epoch : $1, \ldots,$ number of epochs do
    for $L_i$ : layers do
        Collect outputs of the layer $L_i$: $y_k^{L_i} = L_i(x_k)$. Each layer must be noisy/stochastic.
        Compress the outputs via the layer encoder $E_{L_i}$: $c_k^{L_i} = E_{L_i}(y_k^{L_i})$.
        Estimate $I(E_X(X); E_{L_i}(L_i))$ and $I(E_{L_i}(L_i); Y(X))$, where $Y$ maps inputs to true targets.
    end for
    Perform one-epoch training step of the network.
end for
```

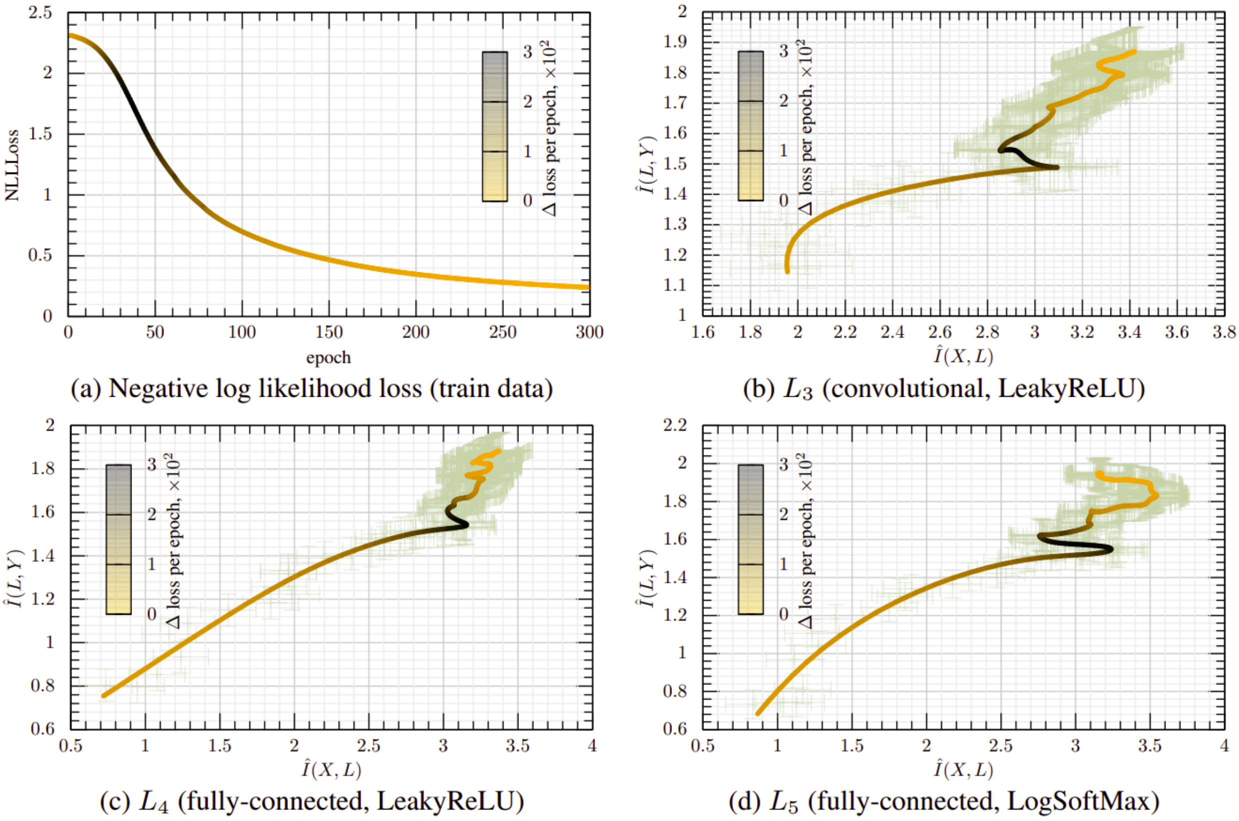
We use the architecture of the classification network provided in Table 2. We train our network with a learning rate of 10^{-4} using the Nvidia Titan RTX. We use $d_X^{\text{latent}} = d_{L_i}^{\text{latent}} = 4$. For other hyperparameters, we refer to Section 7.6 of this chapter and to the source code [7].

L_1:	Conv2d(1, 8, ks=3), LeakyReLU(0.01)
L_2:	Conv2d(8, 16, ks=3), LeakyReLU(0.01)
L_3:	Conv2d(16, 32, ks=3), LeakyReLU(0.01)
L_4:	Dense(32, 32), LeakyReLU(0.01)
L_5:	Dense(32, 10), LogSoftMax

Table 2: The architecture of the MNIST convolution-DNN classifier used in this paper.

The acquired information plane plots are provided in Figure 5. As the direction of the plots with respect to the epoch can be deduced implicitly (the lower left corner of the plot corresponds to the first epochs), we color the lines according to the dynamics of the loss function per epoch. We do this to emphasize one of the key observations:

the first transition from fitting to the compression phase coincides with an acceleration of the loss function decrease. It is also evident that there is no clear large-scale compression phase. Moreover, it seems that the number of fitting and compression phases can vary from layer to layer.



(a) Negative log likelihood loss (train data)

(b) L_3 (convolutional, LeakyReLU)

(c) L_4 (fully-connected, LeakyReLU)

(d) L_5 (fully-connected, LogSoftMax)

Figure 5: Information plane plots for the MNIST classifier. The lower left parts of the plots correspond to the first epochs of training. We use 95% asymptotic CIs for the MI estimates acquired from the compressed data. The colormap represents the difference of losses between two consecutive epochs.

7.6 Technical details

In this section, we describe the technical details of our experimental setup: architecture of the neural networks, hyperparameters, etc.

For the tests described in Section 7.2, we use architectures listed in Table 3. The autoencoders are trained via Adam [29] optimizer on $5 \cdot 10^3$ images with a batch size $5 \cdot 10^3$, a learning rate 10^{-3} and MAE loss for $2 \cdot 10^3$ epochs. The MINE critic network is trained via Adam optimizer on $5 \cdot 10^3$ images with a batch size 512, a learning rate 10^{-3} for $5 \cdot 10^3$ epochs.

For the experiments described in Section 7.5, we use architectures listed in Table 4. The input data autoencoder is trained via Adam optimizer on $5 \cdot 10^4$ images with a batch size 1024, a learning rate 10^{-3} and MAE loss for $2 \cdot 10^2$ epochs; the latent dimension equals 4. The convolutional classifier is trained via Adam optimizer on $5 \cdot 10^4$ images with a batch size 1024, a learning rate 10^{-4} and NLL loss for 100 epochs; Outputs of the layers are compressed via PCA into 4-dimensional vectors. Mutual information is estimated via WKL estimator with 5 nearest neighbours.

Here we do not define f_i and g_i used in the tests with synthetic data, as these functions smoothly map low-dimensional vectors to high-dimensional images and, thus, are very complex. A Python implementation of the functions in question is available in the code for experimets [7] (see the file `source/source/python/mutinfo/utils/synthetic.py`).

Table 3: The NN architectures used to conduct the synthetic tests in Section 7.2.

NN		Architecture
AEs, 16×16 (32×32) images	$\times 1$:	Conv2d(1, 4, ks=3), BatchNorm2d, LeakyReLU(0.2), MaxPool2d(2)
	$\times 1$:	Conv2d(4, 8, ks=3), BatchNorm2d, LeakyReLU(0.2), MaxPool2d(2)
	$\times 2(3)$:	Conv2d(8, 8, ks=3), BatchNorm2d, LeakyReLU(0.2), MaxPool2d(2)
	$\times 1$:	Dense(8, dim), Tanh, Dense(dim, 8), LeakyReLU(0.2)
	$\times 2(3)$:	Upsample(2), Conv2d(8, 8, ks=3), BatchNorm2d, LeakyReLU(0.2)
	$\times 1$:	Upsample(2), Conv2d(8, 4, ks=3), BatchNorm2d, LeakyReLU(0.2)
	$\times 1$:	Conv2d(4, 1, ks=3), BatchNorm2d, LeakyReLU(0.2)
MINE, critic NN, 16×16 (32×32) images	$\times 1$:	[Conv2d(1, 8, ks=3), MaxPool2d(2), LeakyReLU(0.01)]$^{\times 2 \text{ in parallel}}$
	$\times 1(2)$:	[Conv2d(8, 8, ks=3), MaxPool2d(2), LeakyReLU(0.01)]$^{\times 2 \text{ in parallel}}$
	$\times 1$:	Dense(128, 100), LeakyReLU(0.01)
	$\times 1$:	Dense(100, 100), LeakyReLU(0.01)
	$\times 1$:	Dense(100, 1)

Table 4: The NN architectures used to conduct the information plane experiments in Section 7.5.

NN		Architecture
Input data AE, 24×24 images	$\times 1$:	Dropout(0.1), Conv2d(1, 8, ks=3), MaxPool2d(2), LeakyReLU(0.01)
	$\times 1$:	Dropout(0.1), Conv2d(8, 16, ks=3), MaxPool2d(2), LeakyReLU(0.01)
	$\times 1$:	Dropout(0.1), Conv2d(16, 32, ks=3), MaxPool2d(2), LeakyReLU(0.01)
	$\times 1$:	Dense(288, 128), LeakyReLU(0.01)
	$\times 1$:	Dense(128, dim), Sigmoid
CNN classifier	L_1:	Conv2d(1, 8, ks=3), LeakyReLU(0.01)
	L_2:	Conv2d(8, 16, ks=3), LeakyReLU(0.01)
	L_3:	Conv2d(16, 32, ks=3), LeakyReLU(0.01)
	L_4:	Dense(32, 32), LeakyReLU(0.01)
	L_5:	Dense(32, 10), LogSoftMax

Chapter 8
Discussion and conclusion

An information-theoretic approach to explainable artificial intelligence and deep neural network analysis seems promising, as it is interpretable, robust, and relies on well-developed information theory. However, the direct application of information-theoretic analysis still poses some problems.

We have shown that it is possible to apply information analysis to compressed representations of datasets or models' outputs. To justify our approach, we have acquired several theoretical results regarding mutual information estimation under lossless and lossy compression. These results suggest that this approach is applicable to real datasets. Although it has been shown that an arbitrary amount of information can be lost due to compression, the information required for optimal decompression is still preserved.

We have also developed a framework to test conventional mutual information estimators complemented with the proposed lossy compression step. This framework allows the generation of pairs of high-dimensional datasets with small internal (latent) dimensions and a predefined quantity of mutual information. The conducted numerical experiments have shown that the proposed method performs well, especially when entropy estimation is done via a weighted Kozachenko-Leonenko estimator. Other methods tend to underestimate or overestimate mutual information.

Finally, an information plane experiment with the MNIST dataset classifier has been carried out. This experiment has shown that the dynamics of information-theoretic quantities during the training of DNNs are indeed non-trivial. However, it is not clear whether the original fitting-compression hypothesis holds, as there is no clear large-scale compression phase after the fitting. We suggest that there may be several compression/fitting phases during the training of real-scale neural networks.

An interesting observation has also been made: the first compression phase coincides with the rapid decrease of the loss functions It may be concluded that the phases revealed by information plane plots are connected to different regimes of learning (i.e., accelerated, stationary, or decelerated drop of the loss function). However, we note that this observation is not the main contribution of our work, and further investigation has to be carried out in order to support this seeming connection with more evidence and theoretical basis.

Future work. In our further research, we explored the use of normalizing flows [42] to enhance our approach. Normalizing flows are invertible smooth transformations that allow for lossless and information-preserving compression. They can be employed to transform the joint distribution into a Gaussian distribution, facilitating mutual information estimation. Based on this concept of normalizing flows, we developed an approach and submitted a corresponding paper to the NeurIPS 2024 conference. Additionally, we plan to apply our methods to various large neural networks and conduct a corresponding analysis of the information plane.

Analyzing neural networks through information-theoretic methods is a promising area related to explainable AI. The approaches created (based on lossy compression, normalizing flows et al.) rapidly expand the number of networks that can be analyzed using information plane analysis.

To summarize, the general framework outlined in this thesis (and in the related paper [8]) expands the applicability of Information Bottleneck analysis to deep neural networks that are close to real-world scale. This enables the discovery of new mutual information-based dependencies and relationships.

Acknowledgements

The author would like to express their gratitude to his scientific colleagues from the Skoltech research group for the fruitful and exciting work! The author thanks Alexey Frolov for the supervision and guidance throughout this project.

Innovations

Mutual information evaluation can be useful in various areas of machine learning and applied statistics, making it a versatile tool with an great potential for innovative applications. This chapter present some possible directions for the new technologies in which the results of this work could be integrated:

1. **Feature selection:** Mutual information can help identify the most informative features in a dataset, reducing the dimensionality of the data and improving the performance of machine learning models.
2. **Anomaly detection** By measuring the mutual information between the feature and the target variable, some anomalies in data can be detected. This allows us to identify features associated with anomalous behavior.
3. **Clustering:** Mutual information serves as a similarity measure in clustering algorithms. Using mutual information to measure the similarity between data points allows for more meaningful clusters.
4. **Dimensionality reduction:** Identifying features with the highest mutual information with the target variable helps reduce the number of features while preserving relevant information.
5. **Reinforcement learning:** In reinforcement learning, mutual information guides the exploration-exploitation tradeoff. Measuring the mutual information between the current state and the next state helps determine whether to explore new actions or exploit known ones.
6. **Recommending systems:** Mutual information helps recommend items to users by measuring the mutual information between user preferences and item attributes.
7. **Natural Language Processing (NLP):** Analyzing the relationships between words in natural language text involves measuring the mutual information between them. This helps understand semantic relationships.
8. **Time series forecasting:** Predicting future values in time series data involves measuring the mutual information between past and future values. This leads to more accurate predictions.

These few examples shows that mutual information estimation can be applied in various domains and problems. Therefore the results of this research could lead to novel and innovative applications in machine learning and statistics.

One of the most promising and valuable directions for future innovation is the application of the developed approach to large-scale neural networks for different purposes such as:

- search for instabilities/anomalies in the learning process of a neural network via the Information Bottleneck principle.
- choice of the optimal neural network architecture according to MI-based relationships
- usage of mutual information as a proxy metric to control the neural network training process.

The research on this topic is still ongoing, and the author along with his colleagues are actively working on developing new methods for mutual information (MI) estimation. These new methods are expected to be faster and more accurate than existing approaches. Furthermore, some of above mentioned innovative ideas mentioned may be implemented using these new methods. This could lead to even more advanced applications in machine learning and statistics, making mutual information evaluation an even more powerful tool for researchers and practitioners alike.

Bibliography

[1] Adilova, L., Geiger, B. C., and Fischer, A. Information plane analysis for dropout neural networks, 2023.

[2] Amjad, R. A., and Geiger, B. Learning representations for neural network-based classification using the information bottleneck principle. *(submitted to) IEEE Transactions on Pattern Analysis and Machine Intelligence PP* (02 2018).

[3] Amjad, R. A., Liu, K., and Geiger, B. C. Understanding neural networks and individual neuron importance via information-ordered cumulative ablation. *IEEE Transactions on Neural Networks and Learning Systems 33*, 12 (2022), 7842–7852.

[4] Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning* (07 2018), J. Dy and A. Krause, Eds., vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 531–540.

[5] Berrett, T., and Samworth, R. Nonparametric independence testing via mutual information. *Biometrika 106* (11 2017).

[6] Berrett, T. B., Samworth, R. J., and Yuan, M. Efficient multivariate entropy estimation via k-nearest neighbour distances. *Ann. Statist. 47*, 1 (02 2019), 288–318.

[7] Butakov, I., Tolmachev, A., Malanchuk, S., Neopryatnaya, A., Frolov, A., and Andreev, K. Package for information-theoretic data analysis: https://github.com/VanessB/Information-v3.

[8] Butakov, I., Tolmachev, A., Malanchuk, S., Neopryatnaya, A., Frolov, A., and Andreev, K. Information bottleneck analysis of deep neural networks via lossy compression. In *The Twelfth International Conference on Learning Representations* (2024).

[9] Butakov, I. D., Malanchuk, S. V., Neopryatnaya, A. M., Tolmachev, A. D., Andreev, K. V., Kruglik, S. A., Marshakov, E. A., and Frolov, A. A. High-dimensional dataset entropy estimation via lossy compression. *Journal of Communications Technology and Electronics 66*, 6 (7 2021), 764–768.

[10] Cerrato, M., Köppel, M., Esposito, R., and Kramer, S. Invariant representations with stochastically quantized neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence 37* (06 2023), 6962–6970.

[11] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (2016), D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Eds., pp. 2172–2180.

[12] Cover, T. M., and Thomas, J. A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[13] Darlow, L. N., and Storkey, A. What information does a resnet compress?, 2020.

[14] Datta, A., Sen, S., and Zick, Y. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE Symposium on Security and Privacy (SP)* (2016), pp. 598–617.

[15] Elad, A., Haviv, D., Blau, Y., and Michaeli, T. Direct validation of the information bottleneck principle for deep nets. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (2019), pp. 758–762.

[16] Fan, Z., Wei, J., Zhu, G., Mo, J., and Li, W. Evolutionary neural architecture search for retinal vessel segmentation, 2020.

[17] Fefferman, C., Mitter, S., and Narayanan, H. Testing the manifold hypothesis. *Journal of the American Mathematical Society 29* (10 2013).

[18] Gabrié, M., Manoel, A., Luneau, C., barbier, j., Macris, N., Krzakala, F., and Zdeborová, L. Entropy and mutual information in models of deep neural networks. In *Advances in Neural Information Processing Systems* (2018), S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc.

[19] Geiger, B. C. On information plane analyses of neural network classifiers—a review. *IEEE Transactions on Neural Networks and Learning Systems 33*, 12 (12 2022), 7039–7051.

[20] Goldfeld, Z., Greenewald, K., Niles-Weed, J., and Polyanskiy, Y. Convergence of smoothed empirical measures with applications to entropy estimation. *IEEE Transactions on Information Theory 66*, 7 (2020), 4368–4391.

[21] Goldfeld, Z., van den Berg, E., Greenewald, K. H., Melnyk, I. V., Nguyen, N. H., Kingsbury, B., and Polyanskiy, Y. Estimating information flow in deep neural networks. In *ICML* (2019).

[22] Greenewald, K. H., Kingsbury, B., and Yu, Y. High-dimensional smoothed entropy estimation via dimensionality reduction. In *IEEE International Symposium on Information Theory, ISIT 2023, Taipei, Taiwan, June 25-30, 2023* (2023), IEEE, pp. 2613–2618.

[23] He, W., Wu, Y., Liang, P., and Hao, G. Using darts to improve mold id recognition model based on mask r-cnn. *Journal of Physics: Conference Series 1518* (04 2020), 012042.

[24] Hein, M., and Audibert, J.-Y. Intrinsic dimensionality estimation of submanifolds in r^d. In *Proceedings of the 22nd international conference on Machine learning* (2005), pp. 289–296.

[25] Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv* (07 2012).

[26] Hinton, G. E., and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science 313*, 5786 (2006), 504–507.

[27] Ivanovs, M., Kadikis, R., and Ozols, K. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters 150* (2021), 228–234.

[28] Jónsson, H., Cherubini, G., and Eleftheriou, E. Convergence behavior of dnns with mutual-information-based regularization. *Entropy 22*, 7 (2020).

[29] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization, 2017.

[30] Kirsch, A., Lyle, C., and Gal, Y. Unpacking information bottlenecks: Unifying information-theoretic objectives in deep learning, 2021.

[31] Kozachenko, L. F., and Leonenko, N. N. Sample estimate of the entropy of a random vector. *Problems Inform. Transmission 23* (1987), 95–101.

[32] Kramer, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal 37*, 2 (1991), 233–243.

[33] LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist 2* (2010).

[34] Lipovetsky, S., and Conklin, M. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry 17* (10 2001), 319 – 330.

[35] Liu, H., Simonyan, K., and Yang, Y. Darts: Differentiable architecture search. *ArXiv abs/1806.09055* (2019).

[36] Lorenzen, S. S., Igel, C., and Nielsen, M. Information bottleneck: Exact analysis of (quantized) neural networks. In *International Conference on Learning Representations* (2022).

[37] McAllester, D., and Stratos, K. Formal limitations on the measurement of mutual information. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* (08 2020), S. Chiappa and R. Calandra, Eds., vol. 108 of *Proceedings of Machine Learning Research*, PMLR, pp. 875–884.

[38] Mroueh, Y., Melnyk, I., Dognin, P., Ross, J., and Sercu, T. Improved mutual information estimation. *Proceedings of the AAAI Conference on Artificial Intelligence 35*, 10 (May 2021), 9009–9017.

[39] Noshad, M., Zeng, Y., and Hero, A. O. Scalable mutual information estimation using dependence graphs. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 2962–2966.

[40] Pham, H., Guan, M. Y., Zoph, B., Le, Q. V., and Dean, J. Efficient neural architecture search via parameter sharing. In *ICML* (2018).

[41] Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *Proceedings of the 36th International Conference on Machine Learning* (06 2019), K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97 of *Proceedings of Machine Learning Research*, PMLR, pp. 5171–5180.

[42] Rezende, D. J., and Mohamed, S. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (2015), ICML'15, JMLR.org, p. 1530–1538.

[43] Ribeiro, M., Singh, S., and Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 1135–1144.

[44] Rs, R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision 128* (02 2020).

[45] Sain, S. R. *Adaptive kernel density estimation.* PhD thesis, Rice University, 1994.

[46] Sayyareh, A. A new upper bound for kullback-leibler divergence. *Applied Mathematical Sciences 5* (01 2011), 3303–3317.

[47] Scheidegger, F., Benini, L., Bekas, C., and Malossi, A. C. I. Constrained deep neural network architecture search for iot devices accounting for hardware calibration. In *Advances in Neural Information Processing Systems* (2019), H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc.

[48] Sen, R., Suresh, A. T., Shanmugam, K., Dimakis, A. G., and Shakkottai, S. Model-powered conditional independence test. In *Advances in Neural Information Processing Systems* (2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.

[49] Shwartz-Ziv, R., and Tishby, N. Opening the black box of deep neural networks via information, 2017.

[50] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net, 2015.

[51] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, 56 (2014), 1929–1958.

[52] Steinke, T., and Zakynthinou, L. Reasoning About Generalization via Conditional Mutual Information. In *Proceedings of Thirty Third Conference on Learning Theory* (09–12 Jul 2020), J. Abernethy and S. Agarwal, Eds., vol. 125 of *Proceedings of Machine Learning Research*, PMLR, pp. 3437–3452.

[53] Tang Nguyen, T., and Choi, J. Markov information bottleneck to improve information flow in stochastic neural networks. *Entropy 21*, 10 (2019).

[54] Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing* (1999), pp. 368–377.

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

[56] Turlach, B. Bandwidth selection in kernel density estimation: A review. *Technical Report* (02 1999).

[57] Weglarczyk, S. Kernel density estimation and its application. *ITM Web of Conferences 23* (01 2018), 00037.

[58] Wickstrøm, K., Løkse, S., Kampffmeyer, M., Yu, S., Principe, J., and Jenssen, R. Information plane analysis of deep neural networks via matrix-based renyi's entropy and tensor kernels, 2019.

[59] Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y., and Keutzer, K. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 10726–10734.

[60] Xu, A., and Raginsky, M. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems* (2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.

[61] Yu, S., Wickstrøm, K., Jenssen, R., and Príncipe, J. C. Understanding convolutional neural networks with information theory: An initial exploration. *IEEE Transactions on Neural Networks and Learning Systems 32*, 1 (2021), 435–442.

[62] Štrumbelj, E., and Kononenko, I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems 41* (12 2013), 647–665.