INFORMATION PLANE ANALYSIS FOR DROPOUT NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

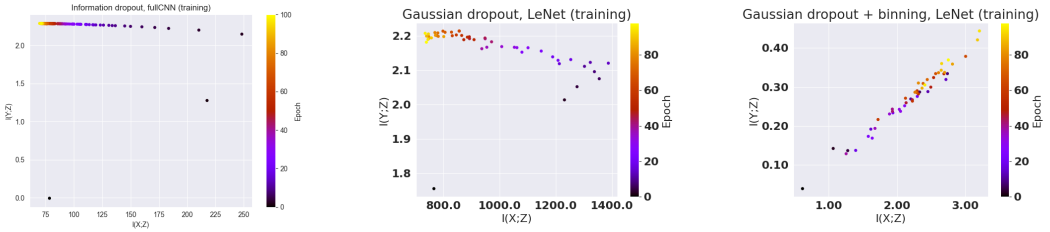
The information-theoretic framework promises to explain the predictive power of neural networks. In particular, the information plane analysis, which measures mutual information (MI) between input and representation as well as representation and output, should give rich insights into the training process. This approach, however, was shown to strongly depend on the choice of estimator of the MI especially for deterministic networks, since in these the MI between input and representation is often infinite. Thus, the estimated values depend strongly on the different approaches for estimation, but do not adequately present the training process from an information-theoretic perspective. In this work, we show that dropout with continuously distributed multiplicative noise ensures that MI is finite. We demonstrate in a range of experiments that this subsequently enables a meaningful information plane analysis for a class of dropout neural networks that is widely used in practice.

1 INTRODUCTION

The information bottleneck hypothesis for deep learning conjectures two phases of training feed-forward neural networks (Shwartz-Ziv and Tishby, 2017): the fitting phase and the compression phase. The former corresponds to extracting information from the input into the learned representations, and is characterized by an increase of mutual information (MI) between inputs and hidden representations. The latter corresponds to forgetting information that is not needed to predict the target, which is reflected in a decrease of the MI between learned representations and inputs, while MI between representations and targets stays the same or grows. The phases can be observed via an information plane analysis, i.e., by analyzing the development of MI between inputs and representations and between representations and targets during training (see Fig. 1 for an example). For an overview we refer the reader to Geiger (2021).

While being elegant and plausible, the information bottleneck hypothesis is challenging to investigate empirically. As shown by Amjad and Geiger (2019, Th. 1), the MI between input and the representations learned by a deterministic neural network is infinite, if the input distribution is continuous. The standard approach is therefore to assume the input distribution to be discrete (e.g., equivalent to the empirical distribution of the dataset S at hand) and to discretize the real-valued hidden representations by binning to allow for non-trivial measurements, i.e. to avoid that the MI always takes the maximum value of $\log(|S|)$ (Shwartz-Ziv and Tishby, 2017). In this discrete setting the MI theoretically gets equivalent to the Shannon entropy. However, considering the effect of binning, the descrease of MI is essentially equivalent to geometrical compression (Basirat et al., 2021). Moreover, the binning-based estimate highly dependents on the chosen bin size (Ross, 2014). To instead work with continuous input distributions, Goldfeld et al. (2019) suggest to replace deterministic neural networks by stochastic ones via adding Gaussian noise to each of the hidden representations. This kind of stochastic networks is rarely used in practice, which limits the insights brought by the analysis.

In contrast, dropout, which is a source of stochasticity, is heavily used in practice due to its effective regularizing properties. The core questions investigated in this work therefore are: i) Can we obtain accurate and meaningful MI estimates in neural networks with dropout noise? ii) And if so, do information planes built for dropout networks confirm the information bottleneck hypothesis? Our main contributions answer these questions and can be summarized as follows: We present a theoretical



(a) Information dropout (b) Gaussian dropout (our) (c) Gaussian dropout (binning)

Figure 1: Information planes for the representation with information and Gaussian dropout at one layer of a LeNet network. In contrast to the information plane based on estimating MI using binning, our estimates (both with Gaussian and information dropout) clearly show compression. This suggests that even if MI is finite, the binning estimator fails to converge to the true MI.

analysis showing that binary dropout does not prevent the MI from being infinite due to the discrete nature of the noise. In contrast, we prove that dropout noise with any continuous distribution not only results in finite MI, but also provides an elegant way to estimate it, thus addressing both questions i) and ii). This in particular holds for Gaussian dropout which is known to benefit generalization even more than binary dropout (Srivastava et al., 2014). We empirically analyze the quality of the MI estimation in the setup with Gaussian dropout in a range of experiments on benchmark neural networks and datasets.

2 Mutual Information Estimation for Neural Networks

We use the following notations: $H(A)$ denotes the Shannon entropy of a discrete random variable (RV) A whose distribution is denoted p_A; $h(B)$ is the differential entropy of a continuous RV B whose distribution is described by the probability density function p_B; $I(A;B)$ is the MI between RVs A and B; $X \in \mathcal{X} \subseteq \mathbb{R}^n$ and $Y \in \mathcal{Y}$ are the RVs describing inputs to a neural network and corresponding targets; $f(X)$ is the result of the forward pass of the input through the network to the hidden layer of interest; Z is an N-dimensional RV describing the hidden representations that are being analyzed. Note that while in a deterministic neural network we have $Z = f(X)$, here we consider representations Z that contain additional stochasticity, e.g., via additive or multiplicative noise. Lower-case letters shall denote realizations of RVs, e.g., b may be a realization of RV B.

The caveats of different approaches with respect to measuring MI were discussed widely in the literature (Saxe et al., 2019; Geiger, 2021). These caveats do not appear for the MI measured between representations Z and targets Y, since the target is in most cases a discrete (class) RV, for which MI is always finite. (But see the work of Kolchinsky et al. (2019) for problems with the information bottleneck hypothesis in this setting.) We therefore concentrate on $I(X;Z)$, considering a deterministic neural network with $Z = f(X)$.

One option for estimating this MI is to assume the input to be drawn from a discrete distribution. This view makes it easy to use a finite dataset S to describe the distribution and is supported by the finiteness of the accuracy of the used computational resources (Lorenzen et al., 2021). More precisely, the distribution of X is assumed uniform on the dataset S, and the discretization of Z is performed at a fixed bin size (e.g., corresponding to the computer precision). In this case, Z is discrete as well and the MI between X and Z is computed as $I(X;Z) = H(Z) - H(Z|X) = H(Z) - H(f(X)|X) = H(Z) - 0 = H(Z)$, since the network forward function $f(\cdot)$ is deterministic. Thus, the estimated MI between input and representation essentially corresponds to the entropy of the representation, which is equal to the entropy $H(Z) = H(X) = \log |S|$ of the empirical distribution on the dataset, unless f maps some of the different data points from the dataset to the same value in latent space.

A different option is to assume X to be drawn from a continuous distribution. This is more aligned to the common description of real-world data and allows us to cover the setting where data is drawn from a data generating distribution $p_{X,Y}$ over the whole space of $\mathcal{X} \times \mathcal{Y}$. In this case, if the net-

work transformation $f(\cdot)$ results in a discrete distribution of the representations Z, one can use the decomposition $I(X, Z) = H(Z) - H(Z|X) = H(Z)$ to estimate MI based on Shannon entropy, provided that the sample size is sufficiently large (note that the dimension N of Z may be large, and that therefore the estimation of $H(Z)$ may suffer from the curse of dimensionality). As shown in Theorem 1 of Amjad and Geiger (2019) this is however not the case for neural networks with commonly used activation functions. If $f(\cdot)$ is deterministic and Z is not purely discrete, the MI between X and Z is infinite. By binning, i.e., by quantizing Z to a discrete RV $\hat{Z}$, the MI $I(X; \hat{Z}) = H(\hat{Z})$ remains finite, but the qualitative behavior of this entropy will be defined by properties of activation functions and selected bin size (Saxe et al., 2019).

From the discussion above it follows that estimating $I(X; Z)$ in deterministic neural networks is an ill-posed problem, and that the estimates reveal not an information-theoretic picture, but often rather a geometric one that is determined by the properties of the chosen estimators. As a solution to the discussed challenges, several authors have suggested to investigate the information planes of stochastic neural networks instead (Amjad and Geiger, 2019; Goldfeld et al., 2019). For example, Goldfeld et al. (2019) proposed to add zero-mean Gaussian noise to the representations during training. This transforms a deterministic neural network into a stochastic one, but was shown to yield similar training results and predictive abilities of the model. Such setup guarantees a finite MI[1] and therefore allows for estimating MI using Monte Carlo sampling with bounds on the estimation error. The addition of Gaussian noise to representations links the information-theoretic perspective of the information plane with geometric effects taking place in latent space. Indeed, when MI between input and representation is decreasing, it means that noise-induced Gaussians centered at $f(x^{(j)})$, the image of the data point $x^{(j)}$, are overlapping more strongly. Thus, it is becoming harder to distinguish between inputs of the same class based on their representations, which translates into lower MI between representation and input.

As discussed above, both the information planes of deterministic neural networks as well as of stochastic neural networks with additive noise at the representations show a geometric picture (and in the former case the geometric picture is the only valid one, since MI is infinite in this case). We therefore, in this work, study the estimation of MI in networks with dropout layers, i.e., in settings where the stochasticity is introduced by multiplicative, rather than additive noise. In what follows we will investigate the requirements on the multiplicative noise for MI to remain finite, and whether the resulting information planes confirm the information bottleneck hypothesis. Finally, we will compare the information planes with quantities reflecting the geometric effects in latent space, effectively determining whether the information-theoretic compression due to multiplicative noise is intrinsically different from geometric compression.

3 Mutual Information in Dropout Networks

As discussed in the previous section, the MI between inputs and hidden representations of deterministic networks is infinite, if we assume the input distribution to be described by some continuous probability density function. To overcome this problem, some form of stochasticity has to be introduced. While adding noise to activations (Goldfeld et al., 2019) indeed allows to compute MI, this practice is not used in most contemporary neural networks. In contrast, neural networks with dropout are one of the most popular classes of neural networks used in practice and are indeed stochastic in nature: Adding a dropout layer to a neural network corresponds to multiplying the hidden representation with some form of random noise. Formally, denoting the random noise by a RV D of the same dimension as $f(X)$, the hidden representation is $Z = f(X) \circ D$, where $\circ$ denotes element-wise multiplication. In the most basic form D follows a Bernoulli distribution (Srivastava et al., 2014). Currently, such binary dropout is widely used since it allows us to intuitively understand it as "turning off" a fraction of neurons during training. There is a variety of other dropout schemes, including multiplicative Gaussian noise, fast dropout (Wang and Manning, 2013), or variational dropout (Kingma et al., 2015). Information dropout (Achille and Soatto, 2018) is a variant that allows us to approximate MI using a closed form expression—while the approximation itself is rather crude, it was shown to help training a network and was therefore proposed as a weighted reg-

[1] At least when the distribution of X and f are such that $f(X)$ has finite variance, then the finiteness of MI follows from the result about the capacity of the additive Gaussian noise channel, cf. (Cover Thomas and Thomas Joy, 1991).

ularization term complementing the cross-entropy loss. In order to obtain such closed form, dropout noise is sampled from a log-normal distribution, and the prior distribution on representations is chosen depending on the activation function (ReLU or Softplus). We provide details on the derivation in Appendix A.2.

In this section, we investigate whether neural networks with dropout have indeed finite MI between input X and representation Z. While we first show a negative result by proving that binary dropout still leads to $I(X;Z) = \infty$, our Theorem 3.3 shows that dropout with continuous multiplicative noise keeps MI finite. This fact will then allow us to estimate MI for such neural networks in Sections 4 and 5.

3.1 Binary Dropout

We start by analysing binary dropout, which forces individual neurons to be "turned off" with some probability. More formally, the output of each neuron is multiplied with an independent Bernoulli RV that is equal to 1 with a predefined probability p. The following theorem shows that this kind of (combinatorial) stochasticity is insufficient to prevent MI from becoming infinite.

Theorem 3.1. *In the setting of (Amjad and Geiger, 2019, Th. 1), let the output $f(\cdot)$ of a hidden layer be parameterized as a deterministic neural network with N neurons, let $B \in \{0,1\}^N$ be the set of independent Bernoulli RVs characterizing the dropout pattern, and let $Z = f(X) \circ B$. Then it holds that $I(X;Z) = \infty$.*

Proof. Using the chain rule of MI, we have

$$\begin{aligned} I(X;Z) &= I(X;Z,B) - I(B;X|Z) \\ &= I(X;Z|B) + I(B;X) - I(B;X|Z) \geq I(X;Z|B) - H(B) \end{aligned}$$

where the last line follows from dropping $I(B;X)$ since they are independent and the fact that $I(B;X|Z) \leq H(B)$. Having $B \in \{0,1\}^N$ as a discrete RV, it immediately follows that $H(B) \leq N \log 2$. Now note that

$$I(X;Z|B) = \sum_{b \in \{0,1\}^N} \mathbb{P}(B=b) I(X;Z|B=b).$$

Since the Bernoulli RVs are independent, positive probability mass is assigned to $B = (1,1,\dots,1)$, i.e., to the case where all neurons are active. Evidently, when $b = (1,1,\dots,1)$ it follows that $Z = f(X)$. Thus, with (Amjad and Geiger, 2019, Th. 1)

$$I(X;Z|B) \geq \mathbb{P}(B = (1,1,\dots,1)) I(X; f(X)) = \infty$$

and thus $I(X;Z) = \infty$. □

While the Bernoulli distribution guarantees that $B = (1,1,\dots,1)$ always has non-zero probability, other distributions over $\{0,1\}^N$ might not have this property. Theorem 3.1 can however be generalized to arbitrary distributions over $\{0,1\}^N$:

Theorem 3.2. *In the setting of (Amjad and Geiger, 2019, Th. 1), let the output $f(\cdot)$ of a hidden layer be parameterized as a deterministic neural network with N neurons, let $B \in \{0,1\}^N$ be the binary RV characterizing the dropout pattern, and let $Z = f(X) \circ B$. Then, it either holds that $I(X;Z) = \infty$ or that $I(X;Z) = 0$ if the dropout patterns almost surely disrupt information flow through the network.*

Proof. If the binary dropout is such that nonzero probability is assigned to the dropout mask $B = (1,1,\dots 1)$, then the statement of the theorem follows as in the proof of the previous theorem.

Assume now that B is such that zero mass is assigned to $B = (1,1,\dots,1)$. To treat this case, we suppose that the distribution of X has a portion with a continuous probability density function on a compact set and that the neural network has activation functions that are either bi-Lipschitz or continuously differentiable with a strictly positive derivative (following the requirements of Amjad and Geiger (2019, Th. 1)). Then, we obtain $I(X; f(X)) = \infty$ from (Amjad and Geiger, 2019, Th. 1) for almost all parameterizations of the neural network. Under this setting, $f(X) \circ b$ is again a neural

network with activation functions that are either bi-Lipschitz or continuously differentiable with a strictly positive derivative. Assuming that b is such that the input of the network is not completely disconnected from the considered layer, for this pattern we have $I(X;Z|B=b)=\infty$. Otherwise, we obviously have $I(X;Z|B=b)=0$. The statement of the theorem follows from taking the expectation over all patterns b. □

Thus, for neural networks with binary dropout, any finite estimate of MI is "infinitely wrong", and the resulting information plane does not permit an information-theoretic interpretation. Essentially, the stochasticity added by binary dropout is combinatorial, and hence cannot mitigate the "continuous" stochasticity available in the input X.

3.2 Dropout with Continuous Noise

As proposed by Srivastava et al. (2014), dropout can also be implemented using continuous Gaussian noise with mean vector $\mu = \mathbf{1}$ and diagonal covariance matrix $I\sigma^2$ with fixed variance σ^2. Achille and Soatto (2018), in contrast, proposed log-normally distributed dropout noise, the variance of which depends on the input sample $x^{(j)}$ (called Information Dropout). Generalizing both Gaussian and Information Dropout, in this section we consider continuously distributed multiplicative noise D. In contrast to binary noise sampled from a discrete distribution, multiplicative Gaussian noise turns the joint distribution of (Z,X) to be absolutely continuous with respect to the marginals of Z and X allowing for finite values of MI between the input X and the hidden representation.

Theorem 3.3. *Let X be bounded, $f(\cdot)$ be parameterized by a deterministic neural network with Lipschitz activation functions, and let $Z = f(X) \circ D(X)$, where the components of $D(x) = (D_1(X),\dots,D_N(X))$ are conditionally independent given X and have essentially bounded differential entropy and second moments, i.e., $\mathbb{E}[D_i(x)^2] \le M < \infty$, for some M and all $i = 1,\dots,N$. Then, under assumption that p_z is bounded, $I(X,Z) < \infty$.*

Proof. We first restrict our attention to representations Z for which all dimensions are different from zero. This can be done w.l.o.g. as the following argument shows. Specifically, suppose that $Z = (Z_1,\dots,Z_N)$ and that $B = (B_1,\dots,B_N)$ with $B_i = 0$ if $Z_i = 0$ and $B_i = 1$ otherwise. Clearly, B is a function of Z, hence $I(X;Z) = I(X;Z,B) = I(B;X) + I(Z;X|B)$. Since B is binary, we have that $I(X;B) \le H(B) \le n\ln 2$. Letting $Z_B = (Z_i|i\colon B_i = 1)$ denote the sub-vector of non-zero elements of Z, we thus obtain

$$I(X;Z) \le n\ln 2 + \sum_b \mathbb{P}(B=b) I(Z_b;X). \tag{1}$$

Therefore, $I(X;Z)$ is finite iff $I(Z_b;X) = I(Z;X|B=b)$ is finite B-almost surely. We thus now fix an arbitrary $B=b$ and continue the proof for $Z = Z_b$.

Since the distribution of Z is continuous, we consider the following decomposition of MI into differential entropies: $I(X;Z) = h(Z) - h(Z|X)$. The entropy of the representations $h(Z)$ is upper-bounded by the entropy of a Gaussian RV with the same covariance matrix Σ as the distribution of $Z = (Z_1,\dots,Z_N)$, i.e., by $\frac{N}{2}\ln(2\pi) + \frac{1}{2}\ln(\det(\Sigma)) + \frac{N}{2}$, where N is the dimensionality of Z. From Hadamard's inequality and since Σ is positive semidefinite it follows that $\det(\Sigma) \le \prod_{i=1}^n \sigma_{ii}^2$, where σ_{ii}^2 are diagonal elements of the covariance matrix, i.e., $\sigma_{ii}^2 = Var[Z_i]$. This variance can be bounded from above. Specifically, since X_i is bounded and $f(\cdot)$ is a composition of Lipschitz functions, $f(X)_i$ is bounded as well. Recalling that $\mathbb{E}[D_i(x)^2] \le M$ holds X-almost surely, this yields

$$Var[Z_i] \le \mathbb{E}_x[f(X)_i^2 D_i(X)^2] = \mathbb{E}_x[f(X)_i^2 \mathbb{E}_d[D_i(X)^2 \mid X]] \tag{2}$$

$$\le M\mathbb{E}_x[f(X)_i^2] \le M\|f(X)_i\|_\infty^2. \tag{3}$$

It remains to show that the $h(Z|X) > -\infty$. Due to the conditional independence of D_i and D_j given X, for all $i \ne j$, the conditional differential entropy of Z factorises in the sum of conditional differential entropy of its components, i.e., $h(Z|X) = \sum_{i=1}^N h(Z_i|X)$. We write this conditional

entropy as an expectation over X and obtain using (Cover Thomas and Thomas Joy, 1991)

$$h(Z_i|X) = \mathbb{E}_x[h(Z_i|X = X)] = \mathbb{E}_x[h(D_i(X)|f(X)_i||X = X)] \tag{4}$$
$$= \mathbb{E}_x[h(D_i(X)|X = X) + \log(|f(X)_i|)] = h(D_i(X)|X) + \mathbb{E}_x[\log(|f(X)_i|)] \tag{5}$$

The first term is finite by assumption. The second term is greater than $-\infty$ on the condition of $p_z \leq C < \infty$. By the law of unconscious statistician:

$$\mathbb{E}[\ln(|f(X)_i|) \mid |f(X)_i| > 0]$$
$$= \int_{f(\mathcal{X})_i} \ln(z) p_z(z) \mathrm{d}z \tag{6}$$
$$= \underbrace{\int_0^1 \ln(z) p_z(z) \mathrm{d}z}_{I_1} + \underbrace{\int_1^{\sup(|f(\mathcal{X})_i|)} \ln(z) p_z(z) \mathrm{d}z}_{I_2} \tag{7}$$

It is obvious that I_2 is positive and finite. Also, $I_1 \geq C \int_0^1 \ln(z) \mathrm{d}z = Cz(\ln(z) - 1)|_0^1 = -C$, thus larger than $-\infty$.

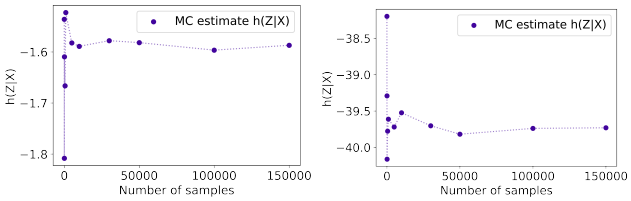
This completes the proof. □

Theorem 3.3 can be instantiated for Gaussian dropout, where $D_i(x) = D_i \sim \mathcal{N}(1, \sigma^2)$, and for Information Dropout, where $D_i(x) \sim \log \mathcal{N}(0, \sigma^2(x))$. Note the for Information Dropout we have to ensure that the (learned) variance stays bounded from above and below; in (Achille and Soatto, 2018), the upper bound is explicitly enforced in the experiments (i.e., $\sigma^2(x) \leq 0.7$). The lower bound on $\sigma^2(x)$ is required to ensure that the differential entropy of D_i is bounded from below.

In Theorem 3.3 we have shown that $I(X; Z)$ if dropout is applied to a the layer that defines Z and that the neural network is deterministic up to this layer. Letting Z_ℓ denote the representation of a subsequent layer, for feed-forward neural networks (e.g., excluding residual connections), the data processing inequality (Cover Thomas and Thomas Joy, 1991) ensures that $I(X; Z_\ell) < \infty$ holds as well.

4 Estimation of MI under Continuous Dropout

We now consider estimating MI in networks with continuously distributed dropout. We start with information dropout. As discussed by Achille and Soatto (2018), networks with information dropout are trained with cross-entropy loss ℓ_{ce} (which is involved in the known variational lower bound on $I(Z; Y) \geq H(Y) - \ell_{ce}$) and regularized using a variational upper bound on $I(X; Z)$. Therefore, estimates of the quantities displayed in the information plane are directly used in the training loss and, thus, easy to track, at least for softplus activation functions.[2]



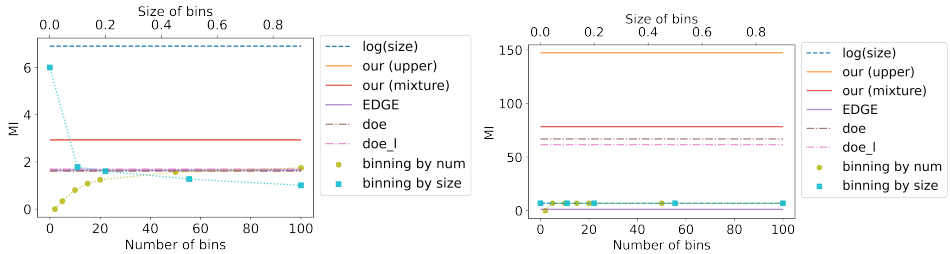
(a) $\sigma = 0.1, n = 2$ (b) $\sigma = 0.1, n = 50$

Figure 2: Independent of the dimensionality, MC estimation of $h(Z|X)$ stabilizes with the amount of samples.

In the case of Gaussian dropout, to estimate $I(X; Z)$ we approximate $h(Z)$ and $h(Z|X)$ separately.

For estimating $h(Z)$ we employ a Monte Carlo (MC) estimate, similar to the one proposed by Goldfeld et al. (2019). That is, we approximate the distribution of Z as a Gaussian mixture, where we draw samples $f(x^{(j)}), j = 1, \ldots, |S|$ and place Gaussians with a diagonal covariance matrix

[2] Indeed, for softplus activation functions, the variational approximation of $I(X; Z)$ is available in closed form, while for ReLU activation functions, the available expression is only useful for minimizing, rather than for computing, $I(X; Z)$.



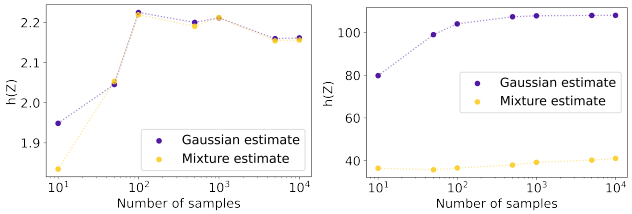
(a) $\sigma = 0.1, n = 1$ (b) $\sigma = 0.1, n = 50$

Figure 4: Comparison of various approaches to MI estimation for the toy example with multiplicative Gaussian noise. For low-dimensional X and Z, different bin sizes lead to different MI estimates of the binning estimator. For higher dimensions, the binning-based estimate is collapsing.

with variances $\sigma^2|f(x^{(j)})_i|^2, i = 1, \ldots, N$ on each sample of $f(x^{(j)})$. For a sanity check, we also compute an upper bound of $h(Z)$ given by the entropy of a Gaussian with the same covariance matrix as Z. Note that the estimation of the upper bound needs a sufficiently large number of samples to guarantee that sample covariance is not singular and that the resulting estimate is finite.

For each fixed $X = x$ the conditional distribution $p_{z|x}$ is a Gaussian distribution $\mathcal{N}(f(x), \text{diag}(\{\sigma^2|f(x)_i|)^2\})$. Moreover, when the input is fixed, the components of $Z|X = x$ are independent, since components of the noise are independent. This allows to compute $h(Z|X)$ as a sum of $h(Z_i|X)$ where Z_i is the i-th component of the representation vector. The computation of $h(Z_i|X)$ requires integration over the input space for computing the mathematical expectation $\mathbb{E}_x[h(Z_i|X = x)]$. This can be approximated via MC sampling. That is, we approximate $h(Z_i|X)$ by $1/|S| \sum_{j=1}^{|S|} h(Z_i|X = x^{(j)})$ where $h(Z_i|X = x^{(j)}) = \log(|f(x^{(j)})_i|\sigma\sqrt{2\pi e})$. More strict pseudocode is given in Algorithm 1 in Appendix A.3.

We consider a simple toy problem for validating our approach to estimating MI where the input X is generated from an n-dimensional standard normal distribution, modified with a function $f(X) = 2X + 0.5$, and then subjected to Gaussian dropout distributed according to $\mathcal{N}(1, \sigma^2)$. We investigate the convergence of our estimator for $h(Z|X)$ for increasing number of samples. For each input data point, we generate 10 noise masks, thus obtaining 10 samples of Z for each $x^{(j)}$. The results in Fig. 2 show that the estimation stabilizes with larger amount of samples for different dimensionality of the data. We also compare the estimate to the upper bound for $h(Z)$ in Fig 3.



(a) $\sigma = 0.1, n = 1$ (b) $\sigma = 0.1, n = 50$

Figure 3: Estimates of the differential entropy $h(Z)$ of the hidden representation Z. With growing dimensionality of X, the Gaussian upper bound becomes very loose, compared to the Gaussian mixture-based MC estimation.

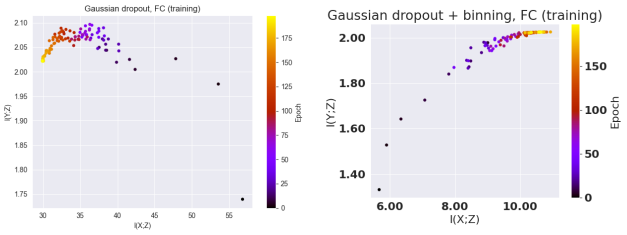
We finally compare our estimator to binning, the EDGE estimator (Noshad et al., 2019), and the lower bounds analyzed by McAllester and Stratos (2020). In the plot, doe stands for difference-of-entropies (DoE) estimator and doe_l stands for DoE with logistic parametrization McAllester and Stratos (2020). The results are shown in Fig. 4. With binning we underestimate MI when bin size is small and overestimate with large bin size (Ross, 2014), which can be clearly seen in the plots where bins are organized both by size and by amount. Moreover, with the high-dimensional data, binning hits the maximal possible value of $\log(|S|)$ very fast, not being able to overestimate the MI value. According to McAllester and Stratos (2020), lower bound-based MI estimators also need exponentially (in the true value of MI) many data points for correct value prediction, otherwise they

will always heavily underestimate the value. Further computations with different noise level are demonstrated in Appendix A.3.

5 Information Plane Analysis of Dropout Networks

We use the estimators described in the previous section for an information plane (IP) analysis of networks with Gaussian and information dropout. We always consider only the representation corresponding to the first dropout layer[3] and measure the MI in nats, e.g. use the natural logarithm.

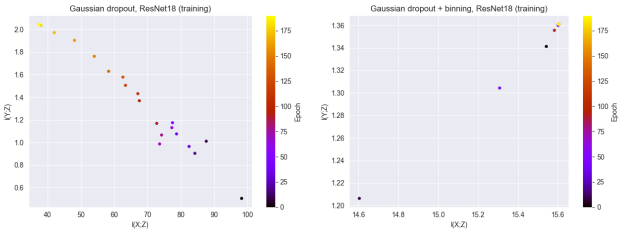
For estimating $I(Y;Z)$, we employ the EDGE estimator (Noshad et al., 2019) for our approach and binning for the binning IP-s. For estimating $I(X;Z)$, we make use of our Gaussian mixture-based estimator (see Section 4). The analysis on the MNIST dataset was performed for a LeNet network (LeCun et al., 1998) that achieves 99% accuracy and a simple fully connected (FC) network with three hidden layers ($28 \times 28 - 512 - 128 - 32 - 10$) and softplus activation functions achieving 97% accuracy. We analyze both information dropout and Gaussian dropout in the LeNet network and only Gaussian dropout in the FC network. We compare IPs based on binning estimators to IPs based on our estimators in Fig. 1 and Fig. 5. We also build the IPs for a ResNet18 trained on CIFAR10, where we added an additional bottleneck layer with 128 neurons and Gaussian dropout before the output layer, and which achieves an accuracy of 94%. The resulting IPs for our MC-based estimator and binning are shown in Fig. 6.



(a) Our estimator (b) Binning estimator

Figure 5: IPs for a FC network with Gaussian dropout trained on MNIST. Compared to the binning estimation of MI our approach reveals compression.

Interestingly, for all networks and datasets we observe significant compression for our estimator and a lack of compression for binning estimators (also for different bin sizes). This indicates that either the MI compression measured in dropout networks is different from purely geometrical compression, or that the number of samples $|S|$ is insufficient to reliably estimate $I(X;\hat{Z})$ by binning. While both options are interesting in their own regard and exhibit the limitations of binning-based estimators in the study of IPs, we tend to arguing for the former, since the binning-based estimates of $I(X;Z)$ are much lower than $\log|S|$ in Figs. 1 and 5.



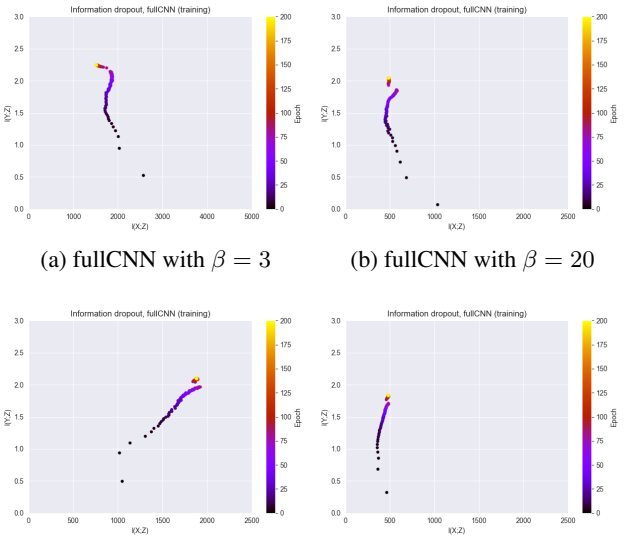
(a) Our estimator (b) Binning estimator

Figure 6: IPs for a ResNet18 network with Gaussian dropout trained on CIFAR10. In contrast to the binning-based estimator of MI our approach clearly shows compression.

In a second set of experiments, we analyze IPs in information dropout networks, with estimations as described before. To this end, we trained a fully convolutional neural network (fullCNN) on CIFAR10 using code provided by Achille and Soatto (2018). Training proceeded for 200 epochs using SGD with momentum and, different from the original setup, only one dropout layer after the third convolutional layer.

[3]This makes the MI estimation more efficient, since the previous part of the network is deterministic which allows for an analytical expression of $h(Z|X = x)$. Note however, that the estimation could be extended to higher layers as well since for those MI also remains finite.

The batch size was set to 100, the learning rate was initially set to 0.05 and was reduced by multiplying it with 0.1 after the 40, 80, and 120 epoch. The network was trained with different values of the regularization weight β and different amounts of filters in the convolutional layers. Also different from the original setup, we allowed noise variance to grow up to 0.95, i.e., that the information flow is nearly cut, in order to see the effect of the limited information between representation and input more pronounced.



(a) fullCNN with $\beta = 3$ (b) fullCNN with $\beta = 20$

(c) 0.25fullCNN with $\beta = 3$ (d) 0.25fullCNN with $\beta = 20$

Figure 7: IP demonstrate more (a), (c) and less (b), (d) compression of MI between input and representation.

Results are shown in Fig. 7 (a) and (b). It can be seen that regularizing $I(X;Z)$ is effective (i.e., larger values of β lead to smaller $I(X;Z)$), and that regularizing too strongly ($\beta = 20$) leads to worse performance (the error is 5% higher for the test set and 10% for the train set) We can further see stronger compression for smaller β and almost no compression for larger β. We conjecture that compressed can only become visible if sufficient information is permitted to flow through the network (which happens only for small β). We repeated experiments with the same convolutional network architecture where we reduced the number of filters in the hidden layers. Fig. 7 (c) and (d) show the IPs for a fullCNN that contains 25% of the filters of the original fullCNN on each of the layers. There it can be seen that the smaller network appears not to compress at all, but that $I(X;Z)$ rather increases throughout training until it is at the same level as in Fig. 7 (a). This indicates that β determines to which point in the information plane training converges, and that trajectory that is traversed during training depends on architectural properties and/or the overall capacity of the neural network.

Additional plots for the performed experiments can be found in Appendix A.4.

6 Conclusion

In this work, we analyzed the dropout neural networks from an information-theoretic perspective. A theoretically sound estimation of the MI between input and hidden representation of a neural network requires either discretization or the introduction of stochasticity into the network. While discretization has the disadvantage that it limits the analysis to the geometric properties of Z, artificially introducing stochasticity limits insights due to altering the model. Dropout neural networks, however, present a class of neural networks that are inherently stochastic and heavily used in practice. In our theoretical analysis, we thus investigate if different variants of dropout noise allow for finite values of MI if we assume a continuous input distribution. Our first result shows that binary dropout does not help to prevent the MI from getting infinite. We then prove, however, that Gaussian dropout—or more generally any form of dropout with noise from a continuous distribution with finite entropy—does. Based on this result we propose an MC based estimate of the MI in Gaussian dropout networks and perform an information plane analysis for different networks with Gaussian and information dropout (the latter offers a natural way to measure MI based on the MI based regularization term) on different datasets. This analysis demonstrates that the proposed estimates lead to substantially different observations than binning based approaches. Particularly, we observe compression for all studied networks and data sets and hypothesize that in dropout networks compression of MI is not driven only by geometrical compression. A more detailed information bottleneck analysis of dropout networks is left for the future work.

REFERENCES

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. IEEE transactions on pattern analysis and machine intelligence, 40 (12):2897–2905, 2018. 3, 5, 6, 8, 11

Rana Ali Amjad and Bernhard C Geiger. Learning representations for neural network-based classification using the information bottleneck principle. IEEE transactions on pattern analysis and machine intelligence, 42(9):2225–2239, 2019. 1, 3, 4

Mina Basirat, Bernhard C Geiger, and Peter M Roth. A geometric perspective on information plane analysis. Entropy, 23(6):711, 2021. 1

Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In International Conference on Machine Learning, pages 573–582. PMLR, 2019. 11

M Cover Thomas and A Thomas Joy. Elements of information theory. New York: Wiley, 3:37–38, 1991. 3, 6

Bernhard Claus Geiger. On information plane analyses of neural network classifiers – a review, 2021. accepted for publication in IEEE Trans. Neural Netw. Learn. Syst.; preprint available: `arXiv:2003.09671 [cs.LG]`. 1, 2

Ziv Goldfeld, Ewout van den Berg, Kristjan H Greenewald, Igor Melnyk, Nam Nguyen, Brian Kingsbury, and Yury Polyanskiy. Estimating information flow in deep neural networks. In ICML, 2019. 1, 3, 6

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. Advances in neural information processing systems, 28, 2015. 3

Artemy Kolchinsky, Brendan D Tracey, and Steven Van Kuyk. Caveats for information bottleneck in deterministic scenarios. ICLR, 2019. 2, 11

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998. 8

Stephan Sloth Lorenzen, Christian Igel, and Mads Nielsen. Information bottleneck: Exact analysis of (quantized) neural networks. arXiv preprint arXiv:2106.12912, 2021. 2

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In International Conference on Artificial Intelligence and Statistics, pages 875–884. PMLR, 2020. 7

Morteza Noshad, Yu Zeng, and Alfred O Hero. Scalable mutual information estimation using dependence graphs. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2962–2966. IEEE, 2019. 7, 8

Brian C Ross. Mutual information between discrete and continuous data sets. PloS one, 9(2): e87357, 2014. 1, 7

Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. Journal of Statistical Mechanics: Theory and Experiment, 2019(12):124020, 2019. 2, 3

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810, 2017. 1

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1):1929–1958, 2014. 2, 3, 5, 15

Sida Wang and Christopher Manning. Fast dropout training. In international conference on machine learning, pages 118–126. PMLR, 2013. 3

A MUTUAL INFORMATION ESTIMATION CHALLENGES

A.1 DISCUSSION ON INFINITE MUTUAL INFORMATION

MI is defined (among other definitions) as Kullback-Leibler divergence between the joint distribution and product of marginals for two random variables. According to definition

$$D_{KL}(A||B) = \mathbb{E}_A log\frac{dA}{dB}$$

where A should be absolutely continuous with respect to B (otherwise the value is defined to be infinity). So, for the validity of MI estimation we have to be sure that joint distribution is absolutely continuous with respect to the product of marginals, i.e., claiming that if $P_A \cdot P_B = 0$ it will imply that $P_{A,B} = 0$. Now let's consider the case of a deterministic neural network representation, i.e., put $X \in \mathbb{R}^d$ to be an input random variable, $Z = f(X)$, $Z \in \mathbb{R}^l$ to be an intermediate representation. While joint distribution will be defined by a pdf on $\mathbb{R}^{d+l}$ space, essentially it will have only d degrees of freedom. So the measure of the space defined by $(x, f(x)), x \in X$ will be 1 for the joint distribution pdf. But the measure of product of marginals in the same space $(x, f(x))$ will be 0, since the dimensionality of it is lower (e.g., 3-dimensional integral of a line in 2-dimensional space is 0). Like this, estimation of MI between input and representation for deterministic neural networks becomes a meaningless task, since theoretically it should be infinity.

Another side of the overall problem modeling is possibility that neural network is a bijective (one-to-one) function. Then the information passed through it does not change in terms of MI: According to the basic properties of MI $I(g(A); B) = I(A; B)$ when $g(\circ)$ is a bijective function. This results in $I(X; Z) = I(X; X)$, which is either entropy of the input or infinity, and is detrimental for measuring MI between representation and label, because here $I(Z; Y) = I(f(X); Y) = I(X; Y)$ meaning that MI will not reveal any knowledge about the properties of Z and neural network that produces it. Nevertheless, we know that training an invertable neural network is a hard task, pursued by, for example, metric learning (Behrmann et al., 2019). So, until specifically trained, neural network representations are not a bijection of the input space. On the other hand, the setup is justified by Kolchinsky et al. (2019) on the assumption that with a finite dataset in high dimension at hand the representations are mostly bijective and the information plane is not interesting for analysis of the training process. We nevertheless hold to the standard modeling paradigm in machine learning that there is a distribution defined over inputs and outputs $\mathcal{D}(\mathcal{X} \times \mathcal{Y})$, and even if the dataset at hand can be mapped to its labels one-to-one the overall space does not follow that deterministic trend, thus such analysis just omits important aspects of the underlying process in the training.

A.2 INFORMATION DROPOUT

Another type of dropout with continuous dropout noise is information dropout (Achille and Soatto, 2018). It is a technique that combines a variable dropout noise and a continuous dropout variables with dropout noise sampled from a log-normal distribution $p_{\alpha(x)}(\epsilon) = log\mathcal{N}(0, \alpha_\theta^2(x))$, where $\alpha(x)$ is a learnable parameter, and the introduction of a regularization term that is added to the standard cross entropy loss and is given by $\beta KL(p(z|x_i)|| \prod_{i=1}^{|z|} q_i(z_i))$.

This is based on another application of the information bottleneck theory to deep learning: training a neural network using the information bottleneck in the form of Lagrangian. Thus, rewriting IB-Lagrangian and adding a disentanglement term (i.e., we want each element of representation z to be independent of others) results in the used regularization term. Initial $I(x, z)$ is rewritten as $\mathbb{E}_{x\sim p(x)}[KL(p_\theta(z|x)||p_\theta(z)]$ and then mathematical expectation is replaced with empirical average.

In order to be able to optimize this regularization term in a closed form, dropout noise is sampled from log-normal distribution $p_{\alpha(x)}(\epsilon) = log\mathcal{N}(0, \alpha_\theta^2(x))$ where α is a learnable through back-propagation parameter. Additionally, it is proposed to use particular priors ($p(z)$) depending on the activation function (ReLU or Softplus) whose validity is empirically verified. Such priors and dropout noise allow for closed form of KL-divergence, which makes it very easy to directly track information plane while training.

In the following, we provide the closed form of the $I(x, z)$ as proposed by Achille and Soatto (2018)

$$I(x, z) = KL(p_\theta(x, z)||p_\theta(z) \circ p(x)) = \int p_\theta(x, z) log(\frac{p_\theta(x, z)}{p_\theta(z) \circ p(x)})$$

$$= \int p(x)p_\theta(z|x)log(\frac{p(x)p_\theta(z|x)}{p_\theta(z) \circ p(x)}) = \int p(x)KL(p_\theta(z|x)||p_\theta(z)) = \mathbb{E}_{x\sim p(x)}[KL(p_\theta(z|x)||p_\theta(z))]$$

Empirically we can compute this as

$$I(x;z) = \sum_{i=1}^{n} KL(p_\theta(z|x_i)||p_\theta(z))$$

If we fix prior distribution $p_\theta(z)$ to improper log-uniform (empirically valid for ReLU activations) and take into account that $z = \epsilon \circ f(x)$, where $\epsilon \sim log\mathcal{N}(0, \alpha^2(x))$, we can write KL-divergence as following

$$KL(p_\theta(z|x)||p_\theta(z)) =$$

considering only the case when $f(x)! = 0$ (which in turn means that $z! = 0$, since ϵ is log-normal and cannot be 0) and due to invarinace of KL-divergence under parameter transformation with a strictly monotone function

$$= KL(p_\theta(log(z)|x)||p_\theta(log(z)) =$$

by definition of KL divergence

$$= \int p_\theta(log(z)|x)log(\frac{p_\theta(log(z)|x)}{p_\theta(log(z))}dz =$$

since $log(z) = log(\epsilon) + log(f(x))$ and $p_\theta(log(z)) = c$ for improper log-uniform prior

$$= \int p_\theta((log(\epsilon) + log(f(x)))|x)log(\frac{p_\theta((log(\epsilon) + log(f(x)))|x)}{c})dz =$$

taking into account that $p(x + const) = p(x)$ and $log(f(x))|x$ is constant; also that $p_\theta(log(\epsilon)|x) = p_\theta(log(\epsilon))$ because ϵ is independent of x [epsilon is dependant on x - its variance is a function of x]

$$= \int p_{\alpha(x)}(log(\epsilon))log(p_{\alpha(x)}(log(\epsilon)))d\epsilon - \int p_{\alpha(x)}(log(\epsilon))log(c)d\epsilon =$$

since $\int p_{\alpha(x)}(log(\epsilon))d\epsilon = 1$

$$= \int p_{\alpha(x)}(log(\epsilon))log(p_{\alpha(x)}(log(\epsilon)))d\epsilon - log(c) = -H(log(\epsilon)) - log(c) =$$

since $log(\epsilon)$ is normally distributed, its entropy can be computed in closed form

$$= -(log(\alpha(x)) + \frac{1}{2}log(2\pi e)) - log(c) \tag{8}$$

In the case when we put $f(x) = 0$, we also get $z = 0$. In this case $p(z|x) = \delta_0$ and the mutual information will become:

$$= \int p_\theta(z|x)log(\frac{p_\theta(z|x)}{p_\theta(z)}dz = \int \delta_0 log(\frac{\delta_0}{q\delta_0})dz = -log(q)$$

Analogously, if instead of ReLU the network has SoftPlus activations, i.e., the prior on the representations distribution is log-normal instead of log-uniform, the formula changes to

$$KL(p_\theta(z|x)||p_\theta(z)) = \frac{1}{2}(\alpha^2(x)) - log(\alpha(x)) - \frac{1}{2}$$

Should be noted, that finiteness of $I(x, z)$ in the case of Information Dropout can be proven in the same way as in the previous section—here the critical term is $\mathbb{E}_x[log(\alpha(x))]$ that is equivalent to the expectation of the logarithm of representation in the continuous noise dropout.

A.3 Estimation of MI under Gaussian Dropout

In the Algorithm 1 we describe how the estimation of $I(X;Z)$ with Z being a representation under Gaussian dropout can be done. This is the way we estimated MI for our experiments, but any other estimator can be used in this setup.

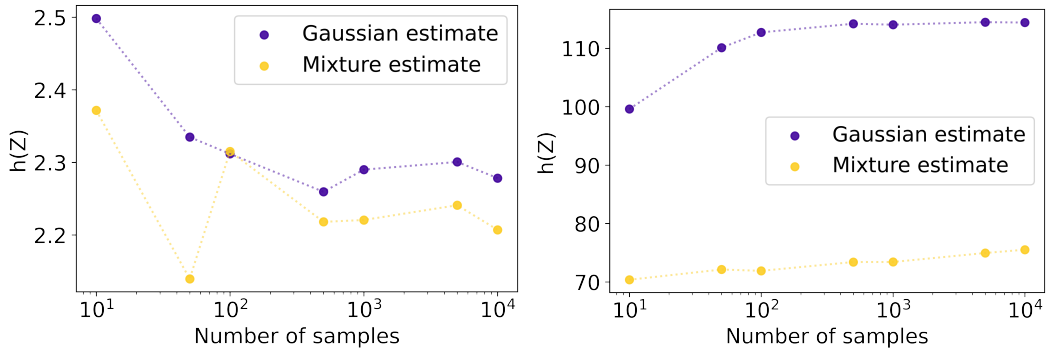
Fig. 8 shows upper bounds and estimation of $h(Z)$ with a higher noise than in the Fig. 3. Larger noise increases the gap between upper Gaussian estimation and mixture estimation as expected.

In Fig. 9 we see convergence of MC estimate for $h(Z|X)$ under larger noise.

As expected larger noise variance results in smaller MI values (Fig. 10), with it the trend observed when changing dimensionality stays the same.

Algorithm 1 Estimation of MI under Gaussian dropout

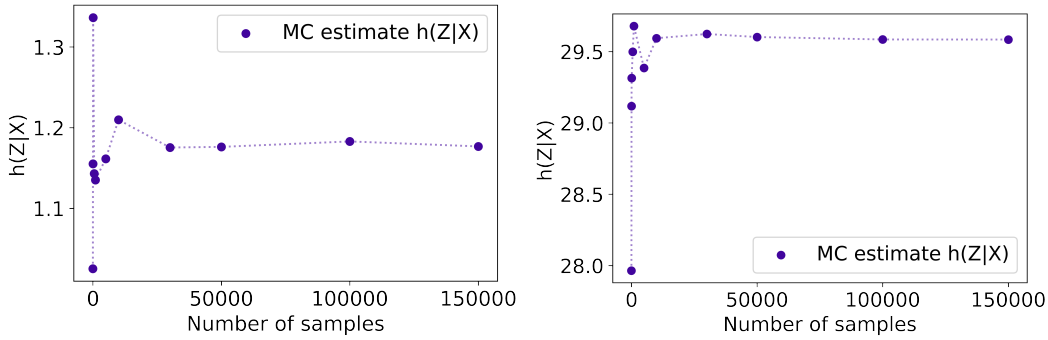
Require: GMM-MEANS, σ, nonoise-reprs ▷ Amount of Gaussians in GM for approximation; noise variance; no noise representations
  $reprs \leftarrow []$ ▷ Generate noisy samples with corresponding variance
  for all nr in nonoise-reprs **do**
    for $i \leftarrow 1, n$ **do**
      $\epsilon \leftarrow noise_p$
      $reprs \leftarrow reprs + nr * \epsilon$
    end for
  end for
  points $\leftarrow$ nonoise-reprs[: GMM-MEANS] ▷ Create a GMM on restricted amount of points for faster computation
  $d \leftarrow []$
  for all p in $points$ **do**
    $d \leftarrow d +$ Gaussian$(p, \sigma * |p|)$
  end for
  gmm $\leftarrow$ MixtureModel(d)
  $lp \leftarrow []$ ▷ Get estimates of log-probabilities from GMM for noisy samples
  for all r in reprs **do**
    $lp \leftarrow lp +$ gmm.log_probability(r)
  end for
  $h(z) \leftarrow$ mean(lp)
  $h(z|x) \leftarrow 0$ ▷ Compute conditional entropy using closed form formula
  for $i \leftarrow 1,$ dim(reprs[0]) **do** ▷ For each dimension of the representation
    $h(z|x) \leftarrow h(z|x) +$ mean$(ln(\sqrt{2\pi e}\sigma|$nonoise-reprs$[:, i]|))$ ▷ Use no noise representations here, each dimension separately
  end for
  $I(x, z) \leftarrow h(z) - h(z|x)$ ▷ Obtain final estimate for the MI



(a) Noise variance 0.4, dimensionality 1

(b) Noise variance 0.4, dimensionality 50

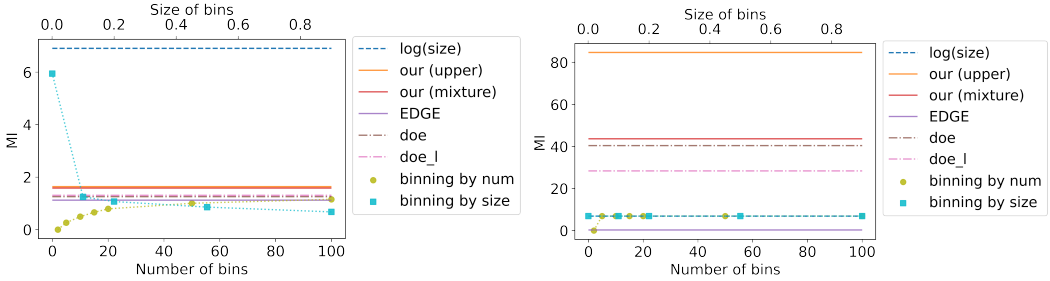
Figure 8: It can be seen that with growing dimensionality the Gaussian estimation becomes very loose, compared to the Gaussian mixture estimation.



(a) Noise variance 0.4, dimensionality 2

(b) Noise variance 0.4, dimensionality 50

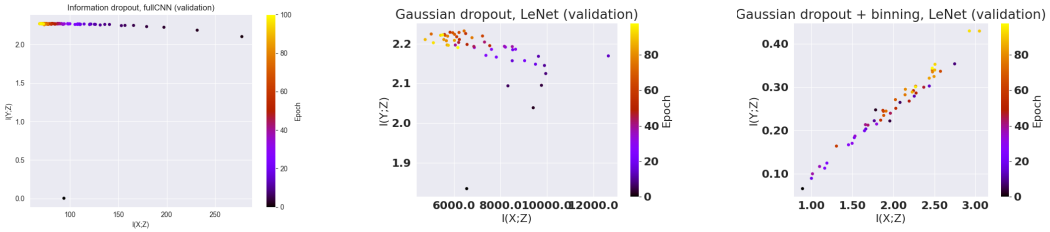
Figure 9: Independent of the dimensionality MC estimation of $h(Z|X)$ stabilizes with the amount of samples.



(a) Noise variance 0.4, dimensionality 1

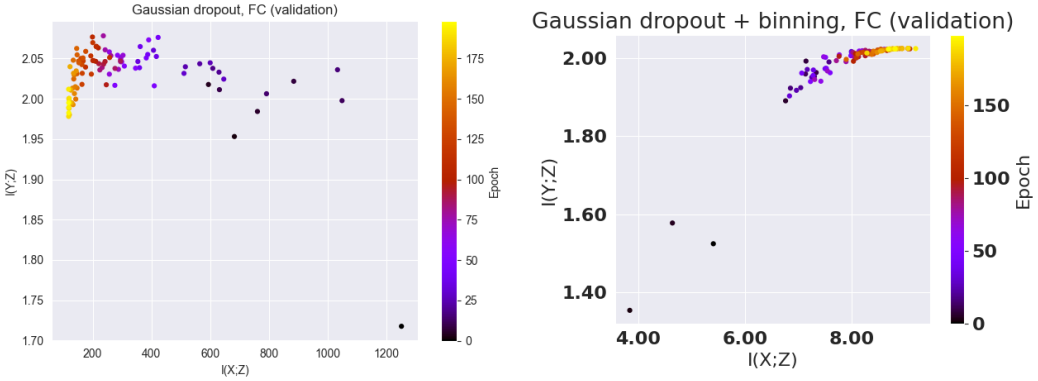
(b) Noise variance 0.4, dimensionality 50

Figure 10: Comparison of various approaches to MI estimation for the setup of the multiplicative Gaussian noise.



(a) Information dropout (b) Gaussian dropout (our) (c) Gaussian dropout (binning)

Figure 11: Compared to the binning (discrete) estimation of MI for an IP analysis, our approach shows compression as well as the compression when information dropout is applied.



(a) Our estimator (b) Binning estimator

Figure 12: Compared to the binning (discrete) estimation of MI for an IP analysis our approach shows compression.

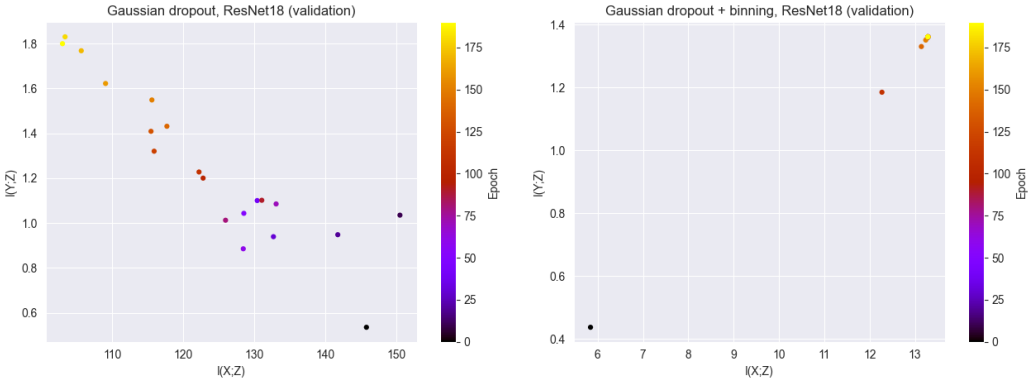
A.4 Information Plane Analysis

Note, that in the experiments we analyze IP on the train sample and test sample separately. In order to obtain a valid sample of variables for MI estimation during the inference, we apply MC-Dropout, as opposed to the usual way of inference with dropout being turned off. According to Srivastava et al. (2014) this is the theoretically sound way to obtain predictions, while turned off dropout is an approximation to it that allows for faster computation.

In Fig. 11, Fig.12 and Fig. 13 we provide IP-s built on the test set of the corresponding datasets.

In the Fig. 14 we provide additional IP-s for binning estimator with varying amount of bins used for MI estimation. We report the results for the fully-connected network trained on MNIST with Gaussian dropout variance 0.2.

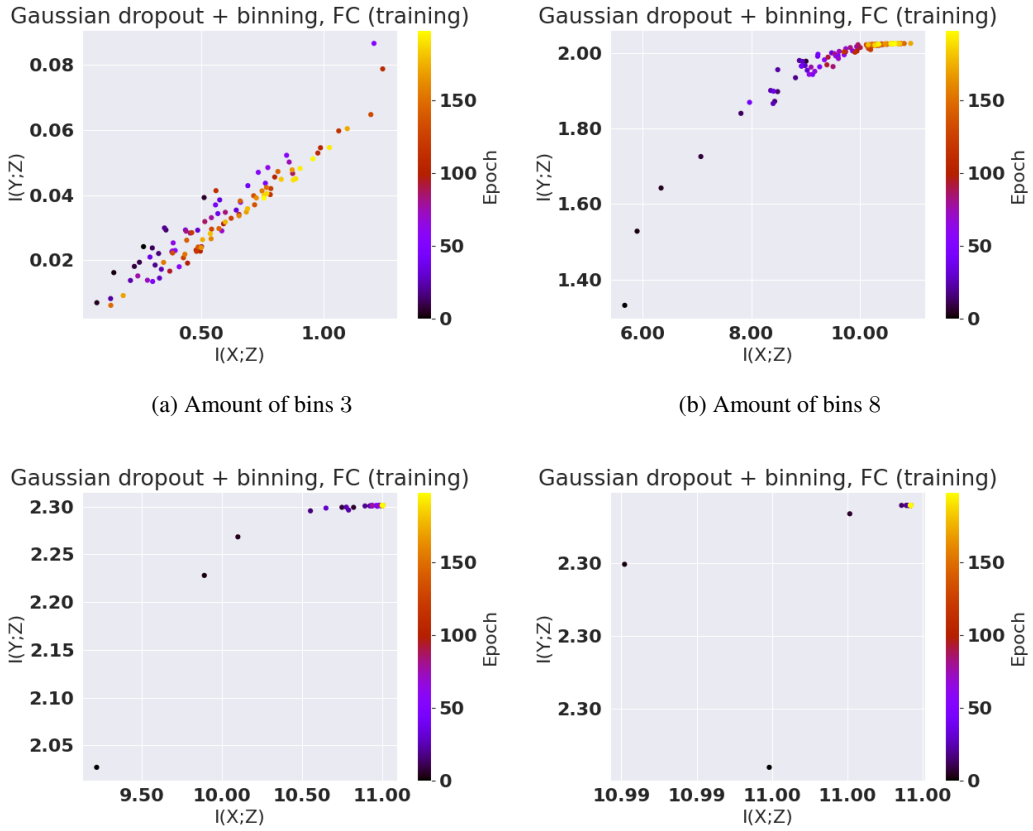
In the Fig. 15 we show the IP-s obtained for the same fully-connected network trained on MNIST with the variance of the Gaussian dropout 0.4.



(a) Our estimator

(b) Binning estimator

Figure 13: Comparing to the discrete estimation of MI for an IP analysis our approach clearly shows compression.



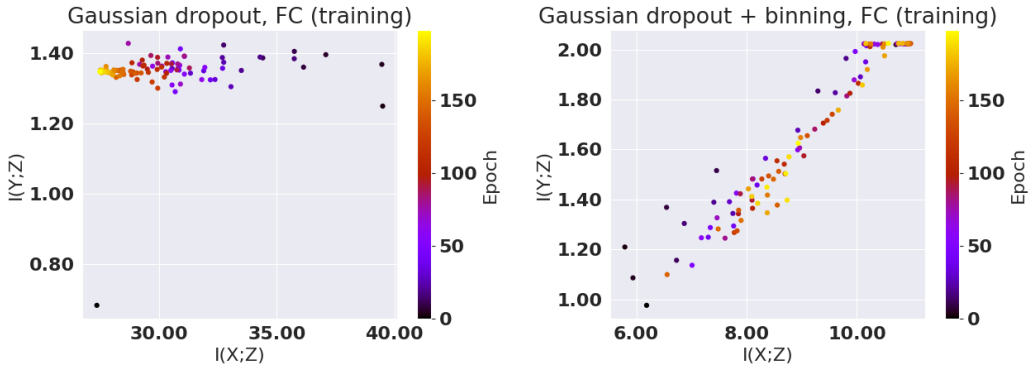
(a) Amount of bins 3

(b) Amount of bins 8

(c) Amount of bins 15

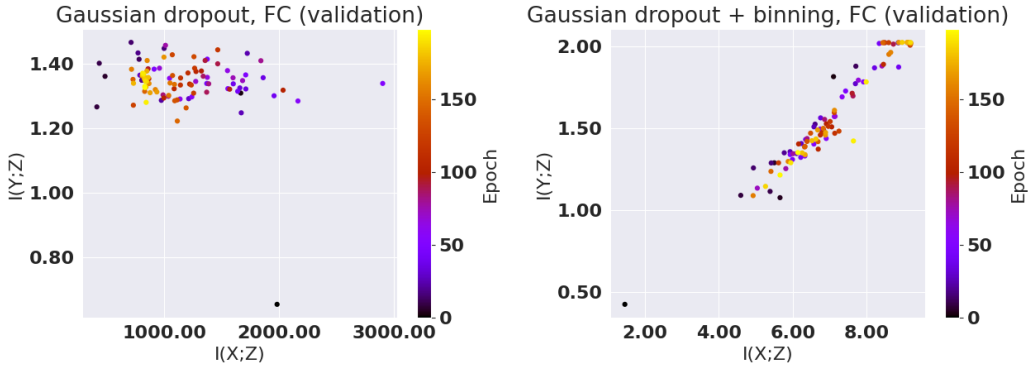
(d) Amount of bins 30

Figure 14: Larger amount of bins used for estimation of MI leads to collapse of the IP into one point corresponding to the amount of samples available. All the smaller amount of bins demonstrates no compression in terms of $I(X;Z)$.



(a) Our estimator (training)

(b) Binning estimator (training)



(c) Our estimator (testing)

(d) Binning estimator (testing)

Figure 15: Same as for the variance of the dropout 0.2 we observe compression when measured with our estimator compared to no compression with binning.