entropy

MDPI

Article

Markov Information Bottleneck to Improve Information Flow in Stochastic Neural Networks

Thanh Tang Nguyen [1,*,†] and Jaesik Choi [2,*,‡]

[1] Applied Artificial Intelligence Institute, Deakin University, Geelong VIC 3220, Australia

[2] Graduate School of Artificial Intelligence, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea

[*] Correspondence: thanhnt@deakin.edu.au (T.T.N.); jaesik.choi@kaist.ac.kr (J.C.)

[†] Part of this work was done at Ulsan National Institute of Science and Technology, Ulsan 44919, Korea.

[‡] Part of this work was done at Ulsan National Institute of Science and Technology, Ulsan 44919, Korea; part of the work was done at KAIST.

Received: 8 September 2019; Accepted: 30 September 2019; Published: 6 October 2019

Abstract: While rate distortion theory compresses data under a distortion constraint, information bottleneck (IB) generalizes rate distortion theory to learning problems by replacing a distortion constraint with a constraint of relevant information. In this work, we further extend IB to multiple Markov bottlenecks (i.e., latent variables that form a Markov chain), namely Markov information bottleneck (MIB), which particularly fits better in the context of stochastic neural networks (SNNs) than the original IB. We show that Markov bottlenecks cannot simultaneously achieve their information optimality in a non-collapse MIB, and thus devise an optimality compromise. With MIB, we take the novel perspective that each layer of an SNN is a bottleneck whose learning goal is to encode relevant information in a compressed form from the data. The inference from a hidden layer to the output layer is then interpreted as a variational approximation to the layer's decoding of relevant information in the MIB. As a consequence of this perspective, the maximum likelihood estimate (MLE) principle in the context of SNNs becomes a special case of the variational MIB. We show that, compared to MLE, the variational MIB can encourage better information flow in SNNs in both principle and practice, and empirically improve performance in classification, adversarial robustness, and multi-modal learning in MNIST.

Keywords: information bottleneck; stochastic neural networks; variational inference; machine learning

1. Introduction

The information bottleneck (IB) principle [1] extracts relevant information about a target variable Y from an input variable X via a *single* bottleneck variable Z. In particular, it constructs a *bottleneck* variable $Z = Z(X)$ that is a *compressed* version of X but preserves as much *relevant* information in X about Y as possible. This principle of introducing relevant information under compression finds vast applications in clustering problems [2], neural network compression [3], disentanglement learning [4–7], and reinforcement learning [8,9]. In addition, there have been many variants of the original IB principle, such as multivariate IB [10], Gaussian IB [11], meta-Gaussian IB [12], deterministic IB [13], and variational IB [14]. Despite these vast applications and variants of IB, alongside the theoretical analysis of the IB principle in neural networks [15,16], the context of stochastic neural networks in which mutual information can be most naturally well-defined [17] has not been sufficiently studied from the IB insight. In this work, we are particularly interested in this context in which multiple stochastic variables are constructed for representation in the form of a Markov chain.

Stochastic neural networks (SNNs) are a general class of neural networks with stochastic neurons in the computation graph. There has been an active line of research in SNNs, including restricted Boltzmann machines (RBMs) [18], deep belief networks (DBNs) [19], sigmoid belief networks (SBNs) [20], and stochastic feed-forward neural networks (SFFNs) [21]. One of the advantages of SNNs is that they can induce rich multi-modal distributions in the output space [20] and enable exploration in reinforcement learning [22]. For learning SNNs (and deep neural networks in general), the maximum likelihood estimate (MLE) principle (in its various forms, such as maximum log-likelihood or Kullback–Leibler divergence) has generally been a de-facto standard. The MLE principle maximizes the likelihood of the model for observing the entire training data. However, this principle is generic and not specially tailored to the hierarchical structure of neural networks. Particularly, MLE treats the entire neural network as a whole without considering the explicit contribution of its hidden layers to model learning. As a result, the information contained within the hidden structure may not be adequately modified to capture the data regularities reflecting a target variable. Thus, it is reasonable to ask if the MLE principle effectively and sufficiently exploits a neural network's representative power, and whether there is a better alternative.

Contributions. In this paper, (i) we propose Markov information bottleneck (MIB), a variant of the IB principle for multiple Markov bottlenecks that directly offers an alternative learning principle for SNNs. In MIB, there are multiple bottleneck variables (as opposed to one single bottleneck variable in the original IB) that form a Markov chain. These multiple Markov bottlenecks sequentially extract relevant information for a learning task. From the perspective of MIB, each layer of an SNN is a bottleneck whose information is encoded from the data via the network parameters connecting the layer to the data layer. (ii) We show that in a non-collapse MIB, the information optimality is not simultaneously achievable for all bottlenecks; thus, an optimality compromise is devised. (iii) When applied to SNNs for a learning task, we interpret the inference from a hidden layer to the output layer in SNNs as a variational approximation to that layer's intractable decoding of relevant information. Consequently, the variational MIB in SNNs generalizes the MLE principle. We demonstrate via a simple analytical argument and synthetic experiment that MLE is unable to learn a good information representation, while the variational MIB can. (iv) We then empirically show that MIB improves the performance in classification, adversarial robust learning, and multi-modal learning in the standard hand-digit recognition data MNIST [23]. This work is an extended version of our preprint [24] and the first author's Master thesis [25].

2. Related Work

There have been many extensions of the original IB framework [1]. One natural consideration is to extend it to continuous variables, yet under special settings where the optimal information representation is analytic [11,12]. Another direction uses alternative measures for compression and/or relevance in IB [13]. Since the optimal information representation in IB is tractable only in limited settings such as discrete variables [1], Gaussian variables [11], and meta-Gaussian variables [12], scaling the IB solution using neural networks and variational inference is a very successful extension [14]. The closest extension to our MIB is multivariate IB [10], in which they define multi-information to capture the dependence among the elements of a multivariate variable. However, in MIB, we do not focus on capturing such multi-information but rather the optimal information sequentially processed by a Markov chain of (possibly multivariate) bottleneck variables.

The line of work applying the IB principle to learn information representation in neural networks is also relevant to our approach. For example, Reference [15] proposes the use of the mutual information of a hidden layer with the input layer and the output layer to quantify the performance of neural networks. However, it is not clear as to how the IB information optimality changes in multiple bottlenecks in a neural network and how we can approximate the IB solutions in this high-dimensional context. In addition, MLE is a standard learning principle for neural networks. It has been shown that the IB principle is mathematically equivalent to the MLE principle in the multinomial mixture model for the clustering problem when the input distribution X is uniform or has a large sample size [26]. However, it is also not clear how these two principles are related to each other in the context of neural networks. Moreover, regarding the feasibility of the IB principle for representation learning in neural networks, Reference [17] analyzes two critical issues of mutual information that representation learning might suffer from: indefinite in deterministic encoding, and invariant under bijective transformations. These are inherent properties of mutual information which are also studied in, for example, [7,27,28]. In MIB, we share with [17] the same insight in these caveats by considering only the scenario where mutual information is well defined. This also explains our rationale in applying MIB to stochastic neural networks.

Deep learning compression schemes [3,29] loosely bear some similarity with our work. Both of the directions aim for a more compressed and useful neural networks for given tasks. The critical distinction is that deep learning compression schemes attempt to produce a smaller-sized neural network with similar performance of a larger one so that the network can be efficiently deployed in small devices such as mobile phones. This task therefore involves size-reduction techniques such as neural network pruning, low-rank factorization, transferred convolution filters and knowledge distillation [29] . On the other hand, our work asks an important representation learning question that given a neural network, what learning principles are the best we can do to improve the information content learned from the data for a given task? In this work, we attempt to address this question via the perspective that a neural network is a set of stochastic variables that sequentially encode information into its layers. We then explicitly improve the information flow (in the sense of more compressed but relevant information) for each layer via our introduced Markov Information Bottleneck framework.

3. Preliminaries

3.1. Notations

We denote random variables (RVs) by capital letters (e.g., X), and their specific realization value by the corresponding lowercase letter (e.g., x). We write $X \perp Y$ (respectively, $X \not\perp Y$) to indicate that X and Y are independent (respectively, not independent). We denote a Markov chain by $Y \rightarrow X \rightarrow Z$, that is, Y and Z are conditionally independent given X, or $Y \perp Z|X$. We use the integral notation when taking expectation (e.g., $\int p(x)f(x)dx$) over the distribution of a random variable regardless of whether the variable is discrete or continuous. We also adopt the following conventions from [27] for defining entropy (denoted by H), mutual information (denoted by I), and Kullback–Leibler (KL) divergence (denoted by D_{KL}): $0 \log \frac{0}{0} = 0, 0 \log \frac{0}{q} = 0, p \log \frac{p}{0} = \infty$.

3.2. Information Bottleneck

Given a (possibly unknown) data joint distribution $p(X,Y)$, the IB framework constructs a *bottleneck* variable $Z = Z(X)$ that is a *compressed* version of X but preserves as much *relevant* information in X about Y as possible. The compression of the representation Z is quantized by $I(Z;X)$, the mutual information of Z and X. The relevance in Z, the amount of information Z contains about Y, is specified by $I(Z;Y)$. The optimal representation Z satisfying a certain compression–relevance trade-off constraint is then determined via minimization of the following Lagrangian $\mathcal{L}_{IB}[p(z|x)] = I(Z;X) - \beta I(Z;Y)$, where β is a positive Lagrangian multiplier that controls the trade-off. Due to the convexity of Lagrangian and constrained conditions with respect to the encoders $\{p(z|x)\}$, the Karush–Kuhn–Tucker (KKT) conditions for this constrained minimization problem become the sufficient and necessary conditions for finding the optimal encoders $\{p(z|x)\}$. By solving the KKT conditions, we can obtain the optimal encoders which can be expressed in an energy-based form as the following:

$$\arg\min_{p(z|x)} \mathcal{L}_{IB}[p(z|x)] \propto p(z) \exp\left(-\beta D_{KL}\left[p(Y|x) \| p(Y|z)\right]\right), \tag{1}$$

where $p(z) = \int p(z|x)p(x)dx$.

4. Markov Information Bottleneck

Given a data joint distribution $p(X, Y)$ which is possibly only observed via a set of i.i.d. samples $S = \{(x_i, y_i)_{i=1}^{N}\}$, an information representation Z for $p(X, Y)$ is said to be good if it encodes sufficient relevant information in X about Y in a compressed manner. Ideally, Z summarizes only the relevant information in X about Y and discards all the irrelevant information; more formally, Z is a minimal sufficient statistic for Y. Such information representation is desirable because it can capture the regularities in the data and is helpful for generalization in learning problems [30,31]. Our main interest is in solving the optimal information representation for a latent variable Z that has Markov structure, that is, $Z = (Z_1, Z_2, \ldots, Z_L)$, where $Z_1 \to Z_2 \to \cdots \to Z_L$. The Markov structure is common in deep neural networks whose advantage is the powerful modeling capacity coming from multiple layers. In MIB, each encoder $p(z_{l+1}|x)$ relates the encoders of the previous bottlenecks in the Markov chain via Bayes' rule:

$$p(z_{l+1}|x) = \int p(z_{l+1}, z_{1:l}|x) dz_{1:l} = \int \prod_{i=1}^{l+1} p(z_i|z_{i-1}) dz_{1:l}, \forall 1 \leq l \leq L-1, \tag{2}$$

where $z_{1:l} := (z_1, \ldots, z_l)$ and $z_0 := x$. In addition, each *encoder* $p(z_l|x)$ corresponds to a unique decoder, namely *relevance decoder*, that decodes the relevant information in x about y from representation z_l:

$$p(y|z_l) = \int p(x, y) \frac{p(z_l|x)}{p(z_l)} dx. \tag{3}$$

In MIB, we further introduce a surrogate target variable $\hat{Y}$ (for the target variable Y) into the Markov chain: $Y \to X \to Z_l \to Z_{l+1} \to \hat{Y}$ (Figure 1). The purpose of the surrogate target variable becomes clear in the section on variational MIB.

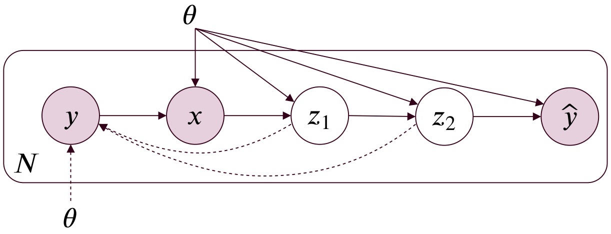


Figure 1. A directed graphical model for the Markov information bottleneck of two Markov bottlenecks. In a non-collapse Markov chain $Y \to X \to Z_1 \to Z_2$ with $0 < \beta_1, \beta_2 < \infty$, the information optimality in Z_1 prevents the information optimality in Z_2. Solid lines denote the encoders $p_\theta(z_i|x)$ (for $i \in \{1, 2\}$), dashed lines denote the variational approximations $p_\theta(\hat{y}|z_i)$ to the intractable *relevance* decoder $p_\theta(y|z_i)$. The variational relevance decoder $p_\theta(\hat{y}|z_i)$ encodes the information from z_i into a surrogate target variable $\hat{y}$. In the case of stochastic neural networks, z_i, θ, and the surrogate target variable represents the hidden layers, the network weights, and the output layer, respectively.

A trivial solution to the optimal information representation problem for Z is to apply the original IB principle for Z as a whole by computing the optimal IB solution in Equation (1). However, this solution ignores the Markov structure of Z. As a principled approach, leveraging the intrinsic structure of a problem can generally provide a new insight that goes beyond the limitation of the perspective that ignores such structure. Thus, in Markov information bottleneck (MIB), we explicitly leverage the Markov structure of Z to derive a principled and tractable approximate solution to the optimal information representation. We then empirically show that leveraging the intrinsic structure in the case of MIB is indeed beneficial for learning.

In MIB, we reframe the optimal information representation problem as multiple IB problems for each of the bottlenecks Z_l:

$$\min_{p(z_l|x)} \mathcal{L}_l[p(z_l|x)] := \min_{p(z_l|x)} \{I(Z_l;X) - \beta_l I(Z_l;Y)\}, \tag{4}$$

for all $1 \leq l \leq L$.

This extension is a natural approach for multiple bottlenecks because it aims for each bottleneck to achieve its own optimal information, and thus allows more relevant but compressed information to be encoded into Z. Another advantage is that we can leverage our understanding of the IB solution for each individual IB problem in Equation (4). Though this approach is promising and has good interpretation, there are two main challenges:

1. The Markov structure among Z_l prevents them from achieving their own information optimality simultaneously in non-trivial cases.
2. The intractability of $p(y|z_l)$ in Equation (3) and $p(z_l|x)$ in Equation (2) results in intractable mutual information in MIB.

In what follows, we formally establish and present the first challenge, the conflicting property of information optimality in Markov structure, in Theorem 1 followed by a simple compromise to overcome the information conflict. After that, we present variational MIB to address the second challenge.

Without loss of generality, we consider the case when $L = 2$ (the result trivially generalizes to $L > 2$). We first define the collapse mode of the representation Z to be the two extreme cases in which Z_2 either contains all the information in Z_1 about X or simply random noise:

Definition 1 (The *collapse* mode of MIB)**.** *$Z = (Z_1, Z_2)$ is said to be in the collapse mode if it satisfies either of the following two conditions:*

1. *Z_2 is a sufficient statistic of Z_1 for X and Y (i.e., $Y \to X \to Z_2 \to Z_1$);*
2. *Z_2 is independent of Z_1.*

For example, if $Z_2 = f(Z_1)$ where f is a deterministic bijection, Z_2 is a sufficient statistic for X. We then establish the conflicting property of information optimality in the Markov representation Z via the following theorem:

Theorem 1 (*Conflicting Markov Information Optimality*)**.** *Given X, Y, Z_1, and Z_2 such that $Y \to X \to Z_1 \to Z_2$ and $H(Y|X) > 0$, consider two constrained minimization problems:*

$$\arg\min_{p(z_l|x)} \mathcal{L}_l[p(z_l|x)] := \arg\min_{p(z_l|x)} \{I(Z_l;X) - \beta_l I(Z_l;Y)\}, l \in \{1,2\}, \tag{5}$$

where $0 < \beta_1 < \infty$, $0 < \beta_2 < \infty$, and $p(z_2|x) = \int p(z_2|z_1)p(z_1|x)dz_1$. Then, the following two statements are equivalent:

1. *Z is not in the collapse mode;*
2. *The two optimal solutions to $\mathcal{L}_1$ and $\mathcal{L}_2$ in (5) are* conflicting, *that is, there is no single solution that minimizes $\mathcal{L}_1$ and $\mathcal{L}_2$ simultaneously.*

Theorem 1 suggests that the Markov information optimality conflicts for most cases of interest (e.g., stochastic neural networks, which we will present in detail in the next section). The values of β_1 and β_2 are important to control the ratio of the relevant information versus the irrelevant one presented in the bottlenecks. These values also determine the conflictability of multiple bottlenecks on the edge cases. Recall by the data processing inequality (DPI) [27] that for $Y \to X \to Z$, we have $0 \leq I(Z;X) \leq H(X)$ and $0 \leq I(Z;Y) \leq I(X;Y)$. If β_1 and β_2 go to infinity, the optimal bottlenecks

Z_1 and Z_2 are both deterministic functions of X and they do not conflict. When $\beta_1 = \beta_2 = 0$, the information about Y in X is maximally compressed in Z_1 and Z_2 (i.e., $Z_1 \perp X, Z_2 \perp X$), and they do not conflict. The optimal solutions conflict when $\beta_1 = 0$ and $\beta_2 > 0$, as the former leads to a maximally compressed Z_1 while the latter prefers an informative Z_2 (this contradicts the Markov structure $X \to Z_1 \to Z_2$, which indicates that maximal compression in Z_1 leads to maximal compression in Z_2).

We can also easily construct non-conflicting MIBs for $0 < \beta_1, \beta_2 < \infty$ that violate the condition. For example, if X and Y are jointly Gaussian, the optimal bottlenecks Z_1 and Z_2 are linear transforms of X and jointly Gaussian with X and Y [11]. In this case, Z_2 is a sufficient statistic of Z_1 for X. In the case of neural networks, we can also construct a simple but non-trivial neural network that can obtain a non-conflicting Markov information optimality. For example, consider a neural network of two hidden layers Z_1 and Z_2, where Z_1 is arbitrarily mapped from the input layer X but Z_2 is a sample mean of n samples i.i.d. drawn from the normal distribution $\mathcal{N}(Z_1; \Sigma)$. This construction guarantees that Z_2 is a sufficient statistic of Z_1 for X, and thus there is non-conflicting Markov information optimality.

Theorem 1 is a direct result of DPI if $\beta_i \in \{0, \infty\}$. In the case that $0 < \beta_i < \infty$, we trace down to the Lagrangian multiplier as in the original IB [1] to complete the proof. Formally, before proving Theorem 1, we first establish the two following lemmas. The first lemma expresses the uncertainty reduction in a Markov chain.

Lemma 1. *Given $Y \to X \to Z_1 \to Z_2$, we have*

$$I(Z_2; X) = I(Z_1; X) - I(Z_1; X|Z_2) \tag{6}$$

$$I(Z_2; Y) = I(Z_1; Y) - I(Z_1; Y|Z_2). \tag{7}$$

Proof. It follows from [27] that $I(X; Z_1; Z_2) = I(X; Z_2) + I(X; Z_1|Z_2) = I(X; Z_1) + I(X; Z_2|Z_1)$, but $I(X; Z_2|Z_1) = 0$ since $X \not\perp Z_2|Z_1$, hence Equation (6). The proof for Equation (7) is similar by replacing variable X with variable Y. (Q.E.D.) □

Lemma 2. *Given $Y \to X \to Z_1 \to Z_2, 0 < \beta_2 < \infty$ and $H(X|Y) > 0$, let us define the conditional information bottleneck objective:*

$$\mathcal{L}^c := \mathcal{L}^c[p(z_2|z_1), p(z_1|x)] := I(Z_1; X|Z_2) - \beta_2 I(Z_1; Y|Z_2). \tag{8}$$

If Z is not in the collapse mode, $\partial \mathcal{L}^c / \partial p(z_1|x)$ depends on $\{p(z_2|z_1)\}$.

Proof. Informally, if Z_2 in the conditional information bottleneck objective $\mathcal{L}^c$ is not a trivial transform of the bottleneck variable Z_1, Z_2 induces a non-trivial topology into the conditional information bottleneck objective. Formally, by the definition of the conditional mutual information

$$I(Z_1; X|Z_2) = \int\int\int p(x, z_1, z_2) \log \frac{p(z_1, x|z_2)}{p(z_1|z_2) p(x|z_2)} dz_2 dz_1 dx,$$

$I(Z_1; X|Z_2)$ depends on $p(x, z_1, z_2)$ as long as the presence of Z_2 in the conditional information bottleneck objective does not vanish (we will discuss the conditions for Z_2 to vanish in the final part of this proof). Note that due to the Markov chain $X \to Z_1 \to Z_2$, we have $p(x, z_1, z_2) = p(x) p(z_1|x) p(z_2|z_1)$.

Thus, $\partial I(Z_1; X|Z_2) / \partial p(z_1|x)$ depends on $p(z_2|z_1)$ as long as Z_2 does not vanish in the objective. Similarly, the same result also applies to $\partial I(Z_1; Y|Z_2) / \partial p(z_1|x)$. Hence, $\partial \mathcal{L}^c / \partial p(z_1|x)$ depends on $\{p(z_2|z_1)\}$ (note that $H(X|Y) > 0$ prevents the collapse of $\mathcal{L}^c$ when summing two mutual informations) if Z_2 does not vanish in the objective.

Now we discuss the vanishing condition for Z_2 in the objective. It follows from Lemma 1 that:

$$0 \leq I(Z_1;X|Z_2) \leq I(Z_1;X), \tag{9}$$
$$0 \leq I(Z_1;Y|Z_2) \leq I(Z_1;Y). \tag{10}$$

Note that Z_2 vanishes in $\mathcal{L}^c$ iff each of the mutual informations in $\mathcal{L}^c$ does not depend on Z_2 iff the equality in both (9) and (10) occur. If $I(Z_1;X|Z_2) = 0$, we have $Y \to X \to Z_2 \to Z_1$ (i.e., Z_2 is a sufficient statistic for X and Y), which also implies that $I(Z_1;Y|Z_2) = 0$. Similarly, $I(Z_1;X|Z_2) = I(Z_1;X)$ implies that Z_2 is independent of Z_1, which in turn implies that $I(Z_1;Y|Z_2) = I(Z_1;Y)$. (Q.E.D.) □

We now prove Theorem (1) by using Lemma (6) and Lemma (7).

Proof of Theorem 1. (⇐) This direction is obvious. When $I(Z_2;X) = I(Z_1;X)$ and $I(Z_2;Y) = I(Z_1;Y)$, or $I(Z_2;X) = 0$ and $I(Z_2;Y) = 0$, there is effectively only one optimization problem for $\mathcal{L}_1$, and this reduces into the original information bottleneck (with single bottleneck) [1].
(⇒) First we prove that if Z is not in the collapse mode, the constrained minimization problems are conflicting. Assume, by contradiction, that there exists a solution that minimizes both $\mathcal{L}_1$ and $\mathcal{L}_2$ simultaneously, that is, $\exists p(z_1|x), p(z_2|z_1)$ s.t. $\mathcal{L}_1$ has a minimum at $\{p(z_1|x)\}$ and $\mathcal{L}_2$ has a minimum at $\{p(z_1|x), p(z_2|z_1)\}$. Note that $\{p(z_1|x)\}$ and $\{p(z_2|z_1)\}$ are independent variables for the optimization. By introducing Lagrangian multipliers $\lambda_1(x)$ and $\lambda_2(x)$ for the constraint $\int p(z_1|x)dz_1 = 1$ of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively, the stationarity in the Karush–Kuhn–Tucker (KKT) conditions becomes:

$$\frac{\partial L_1}{\partial p(z_1|x)} = 0, \tag{11}$$
$$\frac{\partial L_2}{\partial p(z_1|x)} = 0, \tag{12}$$

where L_1 and L_2 are the Lagrangians:

$$L_1[p(z_1|x), \lambda_1] := I(Z_1;X) - \beta_1 I(Z_1;Y) - \int\int \lambda_1(x) p(z_1|x) dz_1 dx \tag{13}$$
$$L_2[p(z_1|x), \lambda_2] := I(Z_2;X) - \beta_2 I(Z_2;Y) - \int\int \lambda_2(x) p(z_1|x) dz_1 dx. \tag{14}$$

It follows from Lemma 1 that:

$$L_2 - L_1 = (\beta_1 - \beta_2) I(Z_1;Y) - \mathcal{L}^c - \int\int (\lambda_2(x) - \lambda_1(x)) p(z_1|x) dz_1 dx, \tag{15}$$

where $\mathcal{L}^c = I(Z_1;X|Z_2) - \beta_2 I(Z_1;Y|Z_2)$ (defined in Lemma 2). We take the derivative w.r.t. $p(z_1|x)$ both sides of Equation (15) and use Equations (11)–(12):

$$\frac{\partial \mathcal{L}^c}{\partial p(z_1|x)} = (\beta_1 - \beta_2) \frac{\partial I(Z_1;Y)}{\partial p(z_1|x)} + \lambda_1(x) - \lambda_2(x). \tag{16}$$

Notice that the left hand side of Equation (16) strictly depends on $p(z_2|z_1)$ (Lemma 2) while the right hand side is independent of $\{p(z_2|z_1)\}$. This contradiction implies that the initial existence assumption is invalid, and thus implies the conclusion in Theorem 1. (Q.E.D.) □

4.1. Markov Information Optimality Compromise

Due to Theorem 1, we cannot simultaneously achieve the information optimality for all bottlenecks. Thus, we need some compromised approach to instead obtain a compromised optimality. We propose two simple compromise strategies, namely, `JointMIB` and `GreedyMIB`. `JointMIB` is a weighted sum of the IB objectives $\mathcal{L}^{joint} := \sum_{l=0}^{L} \gamma_l \mathcal{L}_l$ where $\gamma_l \geq 0$. The main idea of `JointMIB` is to simultaneously optimize all encoders. Even though each bottleneck might not achieve its individual optimality, their joint optimality encourages a joint compromise. On the other hand, `GreedyMIB` progressively solves the information optimality for each bottleneck given that the encoders for the previous bottlenecks are fixed. In other words, `GreedyMIB` tries to obtain the conditional optimality of a current bottleneck which is conditioned on the fixed greedy-optimal information of the previous bottlenecks.

4.2. Variational Markov Information Bottleneck

Due to the intractability of encoders in Equation (2) and relevance decoders in Equation (3), the resulting mutual information in Equation (4) is also intractable. In this section, we present variational methods to derive a lower bound on mutual information in MIB.

4.2.1. Approximate Relevance

Note that $I(Z_l;Y) = H(Y) - H(Y|Z_l)$, where $H(Y) = \textit{constant}$, which can be ignored in the minimization of $\mathcal{L}_l$. It follows from the non-negativity of KL divergence that:

$$\begin{aligned} H(Y|Z_l) &= -\int p(y|z_l)p(z_l)\log p(y|z_l)dydz_l \leq -\int p(y|z_l)p(z_l)\log p_v(y|z_l)dydz_l \\ &= -\mathbb{E}_{(X,Y)}\mathbb{E}_{Z_l|X}\log p_v(Y|Z_l) = -\mathbb{E}_{(X,Y)}\mathbb{E}_{Z_l|X}\log p(\hat{Y}|Z_l) =: \tilde{H}(Y|Z_l), \end{aligned} \tag{17}$$

where we specifically use the relevance decoder for surrogate target variable $p_v(y|z_l) = p(\hat{y}|z_l)$ as a variational distribution to the intractable distribution $p(y|z_l)$:

$$p_v(y|z_l) := \mathbb{E}_{Z_L|z_l}\left[p(\hat{y}|Z_L)\right]. \tag{18}$$

The variational relevance $\tilde{I}(Z_l;Y) := H(Y) - \tilde{H}(Y|Z_l)$ is a lower bound on $I(Z_l;Y)$. This bound is tightest (i.e., zero gap) when the variational relevance decoder $p(\hat{y}|z_l)$ equals the relevance decoder $p(y|z_l)$. In what follows, we establish the relationship between the variational relevance and the log likelihood function, thus connecting MIB with the MLE principle:

Proposition 1 (Variational Relevance Inequalities)**.** *Given the definition of variational relevance $\tilde{I}(Z_l;Y) = H(Y) - \tilde{H}(Y|Z_l)$ where $\tilde{H}(Y|Z_l)$ is defined in Equation (17), and $Z = (Z_1, ..., Z_L)$, we have:*

$$H(Y) + \mathbb{E}_{(X,Y)}\left[\log p(\hat{Y}|X)\right] = \tilde{I}(Z_0;Y) \geq \tilde{I}(Z_l;Y) \geq \tilde{I}(Z_{l+1};Y) \geq \tilde{I}(Z_L;Y) = \tilde{I}(Z;Y), \tag{19}$$

for all $0 \leq l \leq L-1$. where $Z = (Z_1, ..., Z_L)$.

Proposition 1 suggests that: (i) the log likelihood of $p(\hat{y}|x)$ (plus the constant output entropy $H(Y)$) is a special case of the variational relevance at bottleneck $Z_0 = X$; (ii) the log likelihood bound $H(Y) + \mathbb{E}_{(X,Y)}\left[\log p(\hat{Y}|X)\right]$ is an upper bound on the variational relevance for all the intermediate bottlenecks Z_l and for the composite bottleneck $Z = (Z_1, ..., Z_L)$. Therefore, maximizing the log likelihood, as in MLE, does not guarantee to increase the variational relevance for all the the intermediate bottlenecks and the composite bottleneck.

Proof. It follows from Jensen's inequality and the Markov chain that:

$$\begin{aligned} \int p(z_l|x)\log p(\hat{y}|z_l)dz_l &= \int p(z_l|x)\log\left(\int p(\hat{y}|z_{l+1})p(z_{l+1}|z_l)dz_{l+1}\right)dz_l \\ &\geq \int p(z_l|x)\int p(z_{l+1}|z_l)\log p(\hat{y}|z_{l+1})dz_{l+1}dz_l \\ &= \int\int p(z_l|x)p(z_{l+1}|z_l)\log p(\hat{y}|z_{l+1})dz_l dz_{l+1} \\ &= \int p(z_{l+1}|x)\log p(\hat{y}|z_{l+1})dz_{l+1}, \end{aligned}$$

for all $0 \leq l \leq L-1$. Thus, we have:

$$\begin{aligned}\tilde{I}(Z_l;Y) &= H(Y) - \tilde{H}(Y|Z_l) \\ &= H(Y) + \mathbb{E}_{(X,Y)}\mathbb{E}_{Z_l|X} \log p(\hat{Y}|Z_l) \\ &\geq H(Y) + \mathbb{E}_{(X,Y)}\mathbb{E}_{Z_{l+1}|X} \log p(\hat{Y}|Z_{l+1}) \\ &= \tilde{I}(Z_{l+1};Y).\end{aligned}$$

It also follows from the Markov chain that:

$$p(\hat{y}|z) = p(\hat{y}|z_L, z_{L-1}, ..., z_1) = p(\hat{y}|z_L).$$

Therefore, we have:

$$\begin{aligned}\tilde{I}(Z;Y) &= H(Y) + \mathbb{E}_{(X,Y)}\mathbb{E}_{Z_L|Z_{L-1},...,Z_1|Z_0} \log p(\hat{Y}|Z_L) \\ &= H(Y) + \mathbb{E}_{(X,Y)}\mathbb{E}_{Z_L|X} \log p(\hat{Y}|Z_L) \\ &= \tilde{I}(Z_L;Y).\end{aligned}$$

Finally, by the definition in Equation (17), we have:

$$\begin{aligned}\tilde{I}(Z_0;Y) &= H(Y) + \mathbb{E}_{(X,Y)}\mathbb{E}_{Z_0|X} \log p(\hat{Y}|Z_0) \\ &= H(Y) + \mathbb{E}_{(X,Y)} \log p(\hat{Y}|X).\end{aligned}$$

(Q.E.D.) □

4.2.2. Approximate Compression

In practice (e.g., in SNN presented in the next section), we can model the encoding between consecutive layers $p(z_l|z_{l-1})$ with an analytical form. However, the encoding of non-consecutive layers $p(z_l|x)$ for $l > 1$ is generally not analytic as it is a mixture of $p(z_l|z_{l-1})$. We thus propose to avoid directly estimating $I(Z_l;X)$ by instead resorting to its upper bound $I(Z_l;Z_{l-1})$ as its surrogate in the optimization. However, $I(Z_l;Z_{l-1})$ is still intractable as it involves the intractable marginal distribution $p(z_l) = \int p(z_l|x)p(x)dx$. We then approximate $I(Z_l;Z_{l-1})$ using a mean-field (factorized) variational distribution $q(z_l) = \prod_{i=1}^{n_l} q(z_{l,i})$ where $z_l = (z_{l,1}, \ldots, z_{l,n_l})$:

$$\begin{aligned}I(Z_l;X) \leq I(Z_l;Z_{l-1}) &= \int p(z_l|z_{l-1})p(z_{l-1}) \log \frac{p(z_l|z_{l-1})}{p(z_l)} dz_l dz_{l-1} \\ &\leq \int p(z_l|z_{l-1})p(z_{l-1}) \log \frac{p(z_l|z_{l-1})}{q(z_l)} dz_l dz_{l-1} = \mathbb{E}_{Z_{l-1}} D_{KL}\left[p(Z_l|Z_{l-1})||q(Z_l)\right] \\ &= \mathbb{E}_{Z_{l-1}} \sum_{i=1}^{n_l} D_{KL}\left[p(Z_{l,i}|Z_{l-1})||q(Z_{l,i})\right] =: \tilde{I}(Z_l;Z_{l-1}).\end{aligned} \tag{20}$$

The mean-field variational inference not only helps derive a tractable approximation but also encourages distributed representation by constraining each neuron to capture an independent factor of variation for the data [32]; thus, it can potentially represent an exponential number of concepts using independent factors.

5. Case Study: Learning Binary Stochastic Neural Networks

In this section, we officially connect the variational MIB in Section 4 to stochastic neural networks (SNNs). We consider an SNN with L hidden layers (without any feedback or skip connection) where the input layer X, the hidden layers Z_l for $1 \leq l \leq L$, and the output layer $\hat{Y}$ are considered as random variables. We use the convention that $Z_0 := X, Z_{L+1} := \hat{Y}$, and $Z_l = \emptyset$ for all $l \notin \{0, 1, \ldots, L, L+1\}$. Without any feedback or skip connection, Y, X, Z_l, Z_{l+1}, and $\hat{Y}$ form a Markov chain in that order. The output layer $\hat{Y}$ is the surrogate target variable presented in Section 4. The role of SNNs is therefore reduced to transforming from one random variable to another via the Markov chain $X \to Z_l \to Z_{l+1} \to \hat{Y}$ such that it achieves the good information representation (i.e., the compression–relevance tradeoff) for each layer. With the MLE principle, the learning in SNNs is performed by maximizing the log likelihood $\mathbb{E}_{(X,Y)}\left[\log p(\hat{Y}|X)\right]$. However, maximizing the log likelihood does not guarantee to improve the variational relevance for all the intermediate bottlenecks and the composite bottleneck (Proposition 1).

Algorithm 1: `JointMIB`

```
Input: data $S_0 \leftarrow (x_i, y_i)_{i=1}^N \sim p_D(x, y)$, layer IB weights $\gamma_l$, information tradeoff $\beta_l$, number
       of particles for Monte Carlo simulation $M$.
Output: $\theta$
Initialization: $\theta$
while not converged do
    for $i = 1$ to $L$ do
        $S_i \leftarrow \emptyset$
        for $z_{i-1} \in S_{i-1}$ do
            /* Monte Carlo simulates $M$ particles $z_i^{(k)}$ given each $z_{i-1}$ */
            $S_i \leftarrow S_i \cup \{z_i^{(k)} : 1 \leq k \leq M\}$ where $z_i^{(k)} \sim p(z_i|z_{i-1})$
        end
    end
    /* Estimate the variational relevance and compression using Monte Carlo samples */
    $\tilde{I}(Z_l; Y) \leftarrow$ Equation (17) and $\{S_i\}_{i=0}^L$
    $\tilde{I}(Z_l; Z_{l-1}) \leftarrow$ Equation (20) and $\{S_i\}_{i=0}^L$
    $\tilde{\mathcal{L}}^{joint}(\theta) \leftarrow \sum_{l=0}^L \gamma_l \left(-\tilde{I}(Z_l; Y) + \beta_l \tilde{I}(Z_l; Z_{l-1})\right)$
    $g \leftarrow \frac{\partial}{\partial \theta} \tilde{\mathcal{L}}^{joint}(\theta)$ /* Using the Raiko estimator in Binary SNNs */

    $\theta \leftarrow \theta - \nu g$ /* Update using SGD */
end
```

We here instead combine the variational MIB and the MIB compromise to derive a practical learning principle that encourages compression and relevance for each layer, improving the information flow in SNNs. To make it concrete and simple, we consider a simple network architecture: binary stochastic feed-forward (fully-connected) neural networks (SFNNs). In binary SFNNs, we use a sigmoid as the activation function: $p(z_l = 1|z_{l-1}) = \sigma(W_{l-1} z_{l-1} + b_{l-1})$, where $\sigma(.)$ is the (element-wise) sigmoid function, W_{l-1} is the network weights connecting layer $l-1$ to layer l, b_{l-1} is a bias vector, and $Z_l \in \{0, 1\}^{n_l}$. Let us define $\tilde{\mathcal{L}}_l := -\tilde{I}(Z_l; Y) + \beta_l \tilde{I}(Z_l; Z_{l-1})$, where $\tilde{I}(Z_l; Y)$ and $\tilde{I}(Z_l; Z_{l-1})$ are the approximate relevance and compression defined in Equation (17) and (20), respectively. Note that the position of β_l here is slightly different from its position in Equation (4). In Equation (4), β_l is associated with the relevance term to respect the convention of the original IB, while here it is associated with the compression term for practical reasons. In practice, the contribution of $\tilde{I}(Z_l; Y)$ is higher than $\tilde{I}(Z_l; Z_{l-1})$. In computing $\tilde{I}(Z_l; Y)$ and $\tilde{I}(Z_l; Z_{l-1})$, any expectation with respect to $p(z_l|z_{l-1})$ is approximated by Monte Carlo simulation in which we sample M particles

$z_l \sim p(z_l|z_{l-1})$. Regarding the information optimality compromise, we combine the variational MIB objectives into a weighted sum in `JointMIB`:

$$\tilde{\mathcal{L}}^{joint} := \sum_{l=0}^{L} \gamma_l \tilde{\mathcal{L}}_l, \tag{21}$$

where $\gamma_l \geq 0$. In `GreedyMIB`, we greedily minimize $\tilde{\mathcal{L}}_l$ for each $0 \leq l \leq L$. We also make each $q(Z_{l,i})$ a learnable Bernoulli distribution. The `JointMIB` is presented in Algorithm 1. The Monte Carlo sampling operation of Algorithm 1 in stochastic neural networks precludes the backpropagation in a computation graph. It becomes even more challenging with binary stochastic neural networks, as it is not well-defined to compute gradients w.r.t. discrete-valued variables. Fortunately, we can find approximate gradients, which have been proved to be efficient in practice: the REINFORCE estimator [33,34], the straight-through estimator [35], the generalized EM algorithm [20], and the Raiko (biased) estimator [21]. Especially, we found that the Raiko gradient estimator works best in our specific setting and thus deployed it in this application. In the Raiko estimator, the gradient of a bottleneck particle $z_{l,i} \sim p(z_{l,i} = 1|z_{l-1}) = \sigma(a_i^{(l)})$ is propagated only through the deterministic term $\sigma(a_i^{(l)})$: $\frac{\partial z_{l,i}}{\partial \theta} \approx \frac{\partial \sigma(a_i^{(l)})}{\partial \theta}$.

6. Experimental Evaluation

We evaluated the effectiveness of the MIB framework on binary SNNs in synthetic data and MNIST hand-digit recognition data [23]. Each data sample in MNIST is a 28 × 28 gray-scale image representing a handwritten digit from 0 to 9. The dataset is split into 60000 training samples and 1000 test samples. In the synthetic data, we visualized the learning dynamics of the SNNs trained with the variational MIB variants (i.e., `JointMIB` and `GreedyMIB`), and those trained with MLE. In MNIST, we evaluate the effectiveness of the variational MIB variants by comparing them against the baselines MLE and VIB [14] in classification, adversarial robustness and multi-modal learning problems. We make the code for our framework publicly available at https://github.com/thanhnguyentang/pib.

6.1. Synthetic Data: Learning Dynamics of Variational MIB

To better understand how MIB modified the information within the layers during the learning process, we visualized the compression and relevance of each layer over the course of training of stochastic feed-forward neural networks (SFNNs) [21], `JointMIB`, and `GreedyMIB` in synthetic data. $SFN1N$ is different from MIB only in the objective functions: $SFNN$ is trained with the negative log likelihood while MIB is trained with the variational MIB objective. To simplify our analysis, we considered a binary decision problem where X is 12 binary inputs making up $2^{12} = 4096$ equally likely input patterns and Y is a binary variable equally distributed among 4096 input patterns [16]. The base neural network architecture had 4 hidden layers with widths: 10–8–6–4 neurons. Since the network architecture was small, we could precisely compute the true compression $I_x := I(Z_i; X)$ and true relevance $I_y := I(Z_i; Y)$ over training epochs. We fixed $\beta_l = \beta = 10^{-4}$ for both `JointMIB`, trained five different randomly initialized neural networks for each comparative model with stochastic gradient descent (SGD) up to 20,000 epochs on 80% of the data, and averaged the mutual information. In `JointMIB`, we set $\gamma_l = \gamma = 1, \forall l$.

Figure 2 provides a visualization of the learning dynamics of SFNN versus `JointMIB` on the information plane (I_x, I_y). Firstly, we observed a common trend in the learning dynamics of MLE (in the SFNN model) and `JointMIB` frameworks. Both principles allow the network to gradually encode more information about X and the relevant information about Y into the hidden layers at the beginning as $I(Z_i; X)$ and $I(Z_i; Y)$ both increase. Intuitively, in order for the representations Z_l to make sense of the task, the representations should encode enough information about X; thus, $I(Z_l; X)$ should increase. This is especially true for shallow layers because, due to the Markov chain property, the shallower a layer, the greater its burden of carrying enough information to make sense of a task. Especially, we can observe that the increase of $I(Z_l; X)$ slowed down at some point for the deeper layers for both $SFNN$ and MIB. This slowing effect was especially stronger in MIB where the compression is explicitly encouraged during the learning. Secondly, MIB was different from MLE in the maximum level of relevance at each layer and the number of epochs to encode the same level of relevance. `JointMIB` at $l = 1$ needed only about 4.68% of the training epochs to achieve at least the same level of relevance in all layers of SFNN at the final epoch. In addition, MLE was unable to encode the network layers to reach the maximum level of relevance enabled by MIB (we also trained SFNN up to 100,000 epochs and observed that the level of relevance of each layer never reached the value of 0.8 bits).

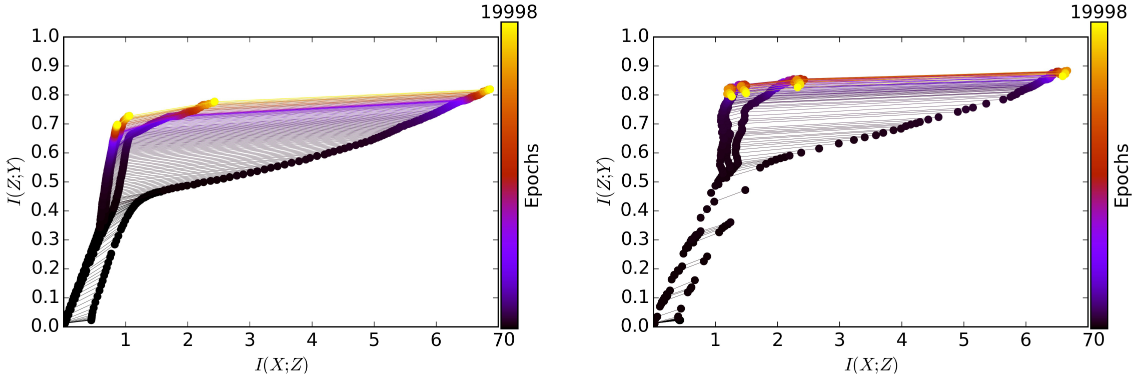


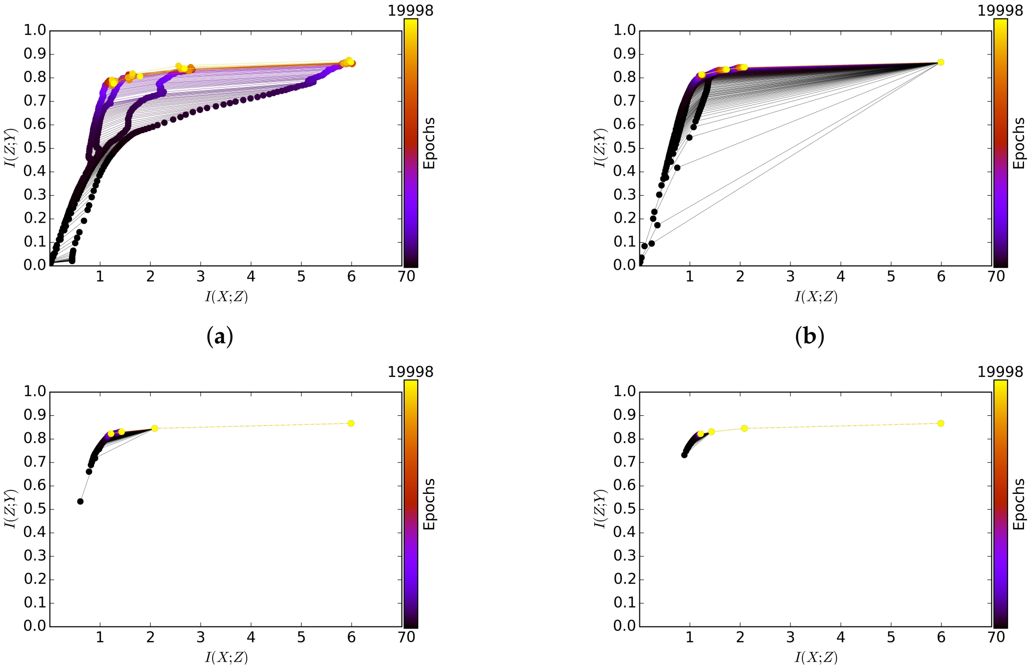
Figure 2. The learning dynamics of the stochastic feed-forward (fully-connected) neural network (`SFNN`) (**left**) and `JointMIB` (**right**). The color indicates the training epochs while each node in a color in the graph represents $(I(Z_l; X), I(Z_l; Y))$ at the corresponding epoch. Note that at each epoch, $I(Z_l; X) \geq I(Z_{l+1}; X), \forall l$ (data processing inequality—DPI). `JointMIB` jointly encodes relevant information into every layer of stochastic neural networks (SNNs), while keeping each layer informatively concise. Compared to maximum likelihood estimation (MLE), the level of relevant information encoded by `JointMIB` increased more quickly over training epochs and reached a higher value. MIB: Markov information bottleneck.

There is also a subtle observation in Figure 2 that the relevance for MIB increased until some point before decreasing, while the relevance for SFNN increased until some point where the value almost stayed the same without a noticeable decrease. This could be explained by the fact that that the MIB objective can eventually allow the encoding of relevant information into each layer to its optimal information trade-off at some point. After this point, if training is continued, due to the mismatch between the exact MIB objective and its variational bound, the further minimization of the variational bound would decrease $I(Z_l; Y)$. Consequently, in order for $\beta_l I(Z_l; X) - I(Z_l; Y)$ to be small, $I(Z_l; X)$ also needs to decrease after this point to compensate for the decrease in $I(Z_l; Y)$. In the case of SFNN (trained with MLE), the MLE objective reaches its local minimum before the information of each layer can even reach its optimal information trade-off (if ever). This also suggests that MIB is better than MLE in terms of exploiting information for each layer during the learning.

`GreedyMIB` also obtained the representation of higher relevance as compared to MLE (Figure 3). `GreedyMIB` at $l = 1$ needed only about 17.95% of the training epochs to achieve at least the same level of relevance in all layers of the SFNN at the final epoch. Recall that in `GreedyMIB` at $l = 1$ the MIB principle is applied only to the first hidden layer. The layer representation at the final epoch gradually shifts to the left (i.e., more compressed) while not degrading the relevance over the greedy training from layer 1 to layer 4 in Figure 3.

We also see the compression effect that the compression constraints within the MIB framework prevented the layer representation from shifting to the right (in the information plane) during the encoding of relevant information (e.g., it slowed down the increase of $I(Z_l; X)$ during the information encoding, keeping the representation more concise). As compared with `JointMIB`, `GreedyMIB` also obtained a comparable information representation.

To conclude, two main advantages of MIB as compared to MLE are: (i) MIB can improve the information representation in SNNs in terms of higher relevance while keeping the information in each layer concise during encoding; (ii) MIB uses much fewer training epochs to obtain such information representation.



(**a**) (**b**)

(**c**) (**d**)

Figure 3. Subfigures (**a**), (**b**), (**c**), and (**d**) represent `GreedyMIB`'s encoding of relevant information into layers $1 \leq l \leq 4$, respectively. `GreedyMIB` greedily encodes relevant information into each layer given the encoded information of the previous layers. `GreedyMIB` also achieved a significantly higher level of relevant information at each layer compared to MLE.

6.2. Image Classification

In this experiment, we compared `JointMIB` and `GreedyMIB` with three other comparative models which used the same network architecture without any explicit regularizer: (1) a standard deterministic neural network (DET) which simply treated each hidden layer as deterministic; (2) a stochastic feed-forward neural network (SFNN) [21] which is a binary stochastic neural network as in MIB but is trained with the MLE principle; and (3) variational information bottleneck (VIB) [14], which uses the entire deterministic network as an encoder, adds an extra stochastic layer as a out-of-network bottleneck variable, and is then trained with the IB principle on that single bottleneck layer. The base network architecture in this experiment had two hidden layers with 512 sigmoid-activated neurons per layer. These models were trained in MNIST [23].

Adopted from the common practice, we used the last 10,000 images of the training set as a validation (holdout) set for tuning hyperparameters. We then retrained the models from scratch in the full training set with the best validated configuration. We trained each of the five models with the same set of five different initializations and reported the average results over the set. For the stochastic models (all except DET), we drew $M = 32$ samples per stochastic layer during both training and inference, and performed inference 10 times at test time to report the mean classification errors for MNIST. The value of $M = 32$ is empirically reasonable in this experiment, as illustrated in Figure 4.

For JointMIB and GreedyMIB, we set $\gamma_l = 1$ (in JointMIB only) and $\beta_l = \beta, \forall 1 \leq l \leq L$, tuned β on a linear log scale $\beta \in \{10^{-i} : 1 \leq i \leq 10\}$. We found $\beta = 10^{-4}$ worked best for both models (Figure 5). For VIB, we found that $\beta = 10^{-3}$ worked best on MNIST . We trained all the models on MNIST with Adadelta optimization [36], except for VIB for which we used Adam optimization [37], as we found that they worked best in the validation set.

The results are shown in Table 1. It shows that JointMIB substantially outperformed DET, MLE, and VIB on MNIST while GreedyMIB outperformed only DET and underperformed SFNN. Though JointMIB and GreedyMIB could have comparable information representation, as illustrated in the synthetic experiment in Section 6.1, in practice, it can be harder to obtain a comparable information representation for GreedyMIB. In GreedyMIB, it is necessary to train each layer greedily in order to obtain its information representation. The greedy nature makes it difficult to determine when would be a good time to stop the training and conclude the information representation for each layer. In addition, training greedily is expensive. JointMIB makes it more efficient by jointly obtaining a compromised information representation in each layer. Thus, it allows the compromised information representations of all the layers to jointly interact with each other during the learning. In principle, it is also harder to obtain good information representation in GreedyMIB. Due to the conflicting information optimality in MIB (Theorem 1), the good encoder for the first layer does not guarantee a good information trade-off in the the deeper layers. Though JointMIB also suffers from the conflicting information optimality, jointly and explicitly inducing relevant but compressed information into each layer of a neural network via MIBs as in JointMIB can make it easier for the training.

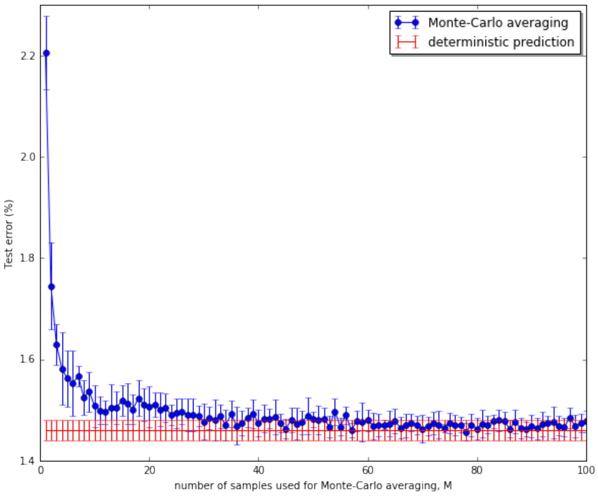


Figure 4. The value of M versus validation error. $M = 32$ gave a reasonably good performance as compared to other larger values.

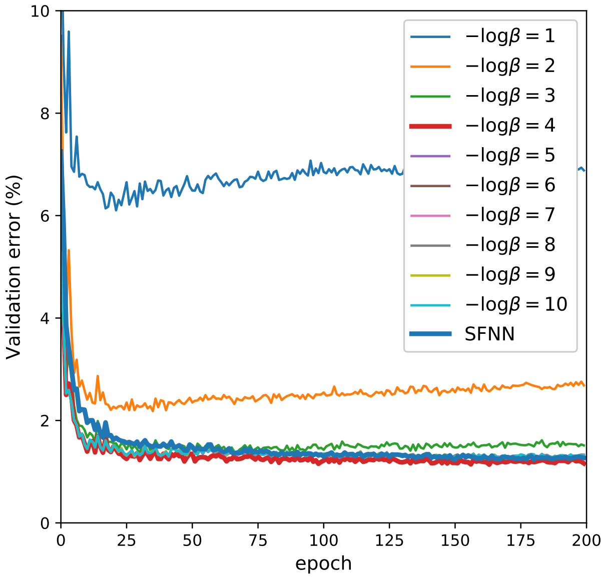


Figure 5. The learning curve of `JointMIB` and SFNN in the MNIST validation set. Either a too-large or too-small value of β could hurt the generalization of learning. While a large value of β introduces aggressive compression, a smaller value allows more irrelevant information into the representation. In this experiment, we found that $\beta = 10^{-4}$ was the best trade-off hyperparameter in `JointMIB` for this experiment.

Table 1. The performance of the variational MIB variants (i.e., `JointMIB` and `GreedyMIB`) for classification and adversarial robustness on MNIST in comparison with MLE and variational information bottleneck (VIB). MIB explicitly induces compression–relevance trade-offs in each layer during the training, which outperforms and is more adversarially robust than the other models of the same architecture. DET: deterministic neural network.

Model	**Classification MNIST (Error %)**	**Adv. Robustness (%)**	
		Targeted	**Untargeted**
DET	1.73	00.00	00.00
VIB [14]	1.45	83.70	93.10
SFNN [21]	1.44	83.00	95.20
`GreedyMIB`	1.54	83.21	94.30
`JointMIB`	**1.36**	**84.16**	**96.00**

6.3. Robustness against Adversarial Attacks

We consider here the adversarial robustness of neural networks trained with MIBs. Neural networks are prone to adversarial attacks which disturb the input pixels by small amounts that are imperceptible to humans [38,39]. Adversarial attacks generally fall into two categories: untargeted and targeted attacks. An untargeted adversarial attack $\mathcal{A}$ maps the target model M and an input image x into an adversarially perturbed image x': $\mathcal{A} : (M, x) \to x'$, and is considered successful if it can fool the model $M(x) \neq M(x')$. A targeted attack, on the other hand, has an additional target label l: $\mathcal{A} : (M, x, l) \to x'$, and is considered successful if $M(x') = l \neq M(x)$.

We performed adversarial attacks on the neural networks trained with MLE and MIB, and used the accuracy on adversarially perturbed versions of the test set to rank a model's robustness. In addition, we used the L_2 attack method for both targeted and untargeted attacks [40], which has shown to be the most effective attack algorithm with smaller perturbations. Specifically, we attacked the same four comparative models described from the previous experiment on the first 1000 samples of the MNIST test set. For the targeted attacks, we targeted each image into the other 9 labels other than the true label of the image. We used the same hyperparameters as in the classification experiment. The value of $\beta = \beta_l = 10^{-4}$ was also reasonable for this adversarial robustness task (Figure 6).

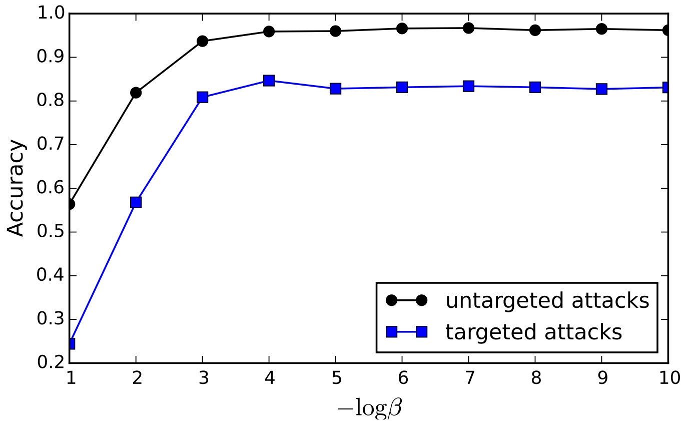


Figure 6. Adversarial robustness of `JointMIB` for various values of β. Introducing aggressive compression (i.e., large values of β) reduced adversarial robustness while smaller values of β introduced comparable robustness. The best information trade-off for targeted attacks was at $\beta = 10^{-4}$ in this experiment.

The results are shown in Table 1. Firstly, it was expected that the adversarial robustness accuracy in the targeted attacks would be smaller than that in the untargeted attacks because the targeted attacks are more challenging for the neural networks to overcome than untargeted attacks. This result is consistent in our experiment. Secondly, the deterministic model DET was totally fooled by all the attacks. It is known that stochasticity in neural networks improves adversarial robustness, which is consistent with our experiment as SFNN was significantly more adversarially robust than DET. Thirdly, VIB had comparable adversarial robustness to SFNN even if VIB had "less stochasticity" than SFNN (VIB had one stochastic layer while all hidden layers of the SFNN were stochastic). We hypothesize that this is because VIB performance was compensated with the IB principle for its stochastic layer. Finally, `JointMIB` was more adversarially robust than the other models. Again, `GreedyMIB` was not very effective in adversarial robustness (it was worse than VIB in the targeted attack and SFNN in the untargeted attack). We hypothesize that this relates to the difficulty for `GreedyMIB` to have a good information representation for all layers. In conclusion, this experiment suggests that explicitly and jointly inducing compression and relevance into each layer has a good potential of being more adversarially robust for neural networks.

6.4. Multi-Modal Learning

One of the main advantages of stochastic neural networks is their ability to model structured output space in which a one-to-many mapping is required. A binary stochastic variable z_l of dimensionality n_l can take on 2^{n_l} different states, each of which would give a different $\hat{y}$. Thus, the conditional distribution $p(\hat{y}|x)$ in stochastic neural networks is multi-modal. Hence in this experiment, we evaluated how MIB affected the multi-modal learning capability of SNNs.

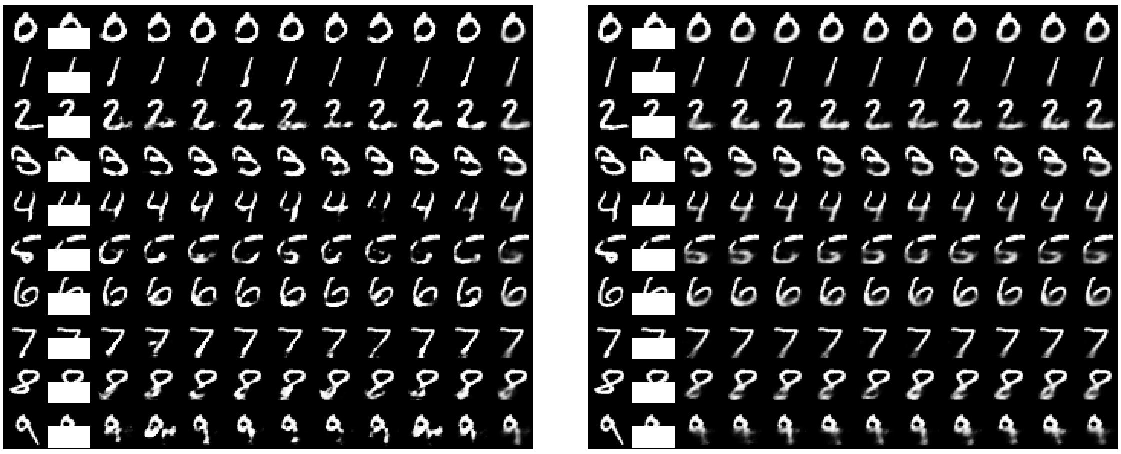


Figure 7. Samples drawn from the prediction of the lower half of the MNIST test data digits based on the upper half for `JointMIB` (**right**, after 60 epochs) and SFNN (**left**, after 200 epochs). The leftmost column is the original MNIST test digit followed by the masked out digits and nine samples. The rightmost column was obtained by averaging over all generated samples of bottlenecks drawn from the prediction. The figures illustrate the capability of modeling structured output space using `JointMIB` and SFNN. `JointMIB` generated more recognizable digits within much fewer training epochs.

In this experiment, we followed [21] and predicted the lower half of the MNIST digits using the upper half as inputs. We used the same neural network architecture of 392–512–512–392 for `JointMIB` and SFNN and trained them with SGD with a constant learning rate of 0.01 (due to the under-performance of `GreedyMIB` from the previous experiments and its expensive training, we compared only `JointMIB` with SFNN in this experiment). We trained the models on the full training set of 60,000 images and tested with the test set. For `JointMIB`, we also used $\beta_l = \beta = 10^{-4}$. The results of `JointMIB` at epoch 60 and MLE at epoch 200 are shown in Figure 7. Firstly, `JointMIB` could generate digit variations which were more recognizable than those generated by MLE. In particular, some samples of digits 2, 4, 5, and 7 generated by MLE were distorted, while all digit samples generated by `JointMIB` were recognizable. Secondly, `JointMIB` used much fewer epochs to achieve good samples. In `JointMIB`, we trained only up to 60 epochs while in MLE, we trained up to 200 epochs but did not observe as good samples in between. This further highlights the advantage of MIB in obtaining good information representation in much fewer training epochs. Furthermore, we expect that the advantage of inducing compression and relevance into each layer by `JointMIB` is particularly helpful for multi-modal learning because in multi-modal learning, the modes generated in each hidden layer are critical for representing multiple modes. While MLE ignores the explicit contribution of each layer to the information representation of the neural network, `JointMIB` explicitly takes into account the compression and relevance of each layer.

7. Discussion and Future Work

In this work, we introduce Markov Information Bottleneck, an extension of the original Information Bottleneck to the context where a representation is multiple stochastic variables that form a Markov chain. In this context, we show that one cannot simply directly apply the original IB principle to each variable as their information optimality is conflicting for most of the interesting cases. We suggest a simple but efficient fix via a joint compromise. In this scheme, we jointly combine the information trade-offs of each variable into a weighted sum, encouraging the information trade-offs for all the variables better off during the learning. In particular in the context of Stochastic Neural Networks, we present the variational inference to estimate the compression and relevance for each bottleneck. As a result, the variational MIB turns the intractable decoding of each bottleneck approximately into an efficient inference for that bottleneck. This variational approximation turns out to generalize the MLE principle in the context of Stochastic Neural Networks. We empirically demonstrate the effectiveness of MIB by comparing it with the baselines using MLE principle and Variational Information Bottleneck in classification, adversarial robustness and multi-modal learning. The empirical performance supports the potential benefit of explicitly inducing compression and relevance into each layer (e.g., in a jointly manner), presenting a special link between information representation and the performance in classification, adversarial robustness and multi-modal learning.

One limitation of our current approach is the number of samples generated via $z_l \sim p(z_l|z_{l-1})$ used to estimate the variational compression and relevance scales exponentially with the number of layers. This is however a common drawback for performing inference in fully stochastic neural networks. This difficulty can be overcome by using partially stochastic neural networks. In addition, the Monte Carlo sampling to estimate the variational mutual information, though unbiased, is of high variance and sample inefficiency. This sample inefficiency limitation can be overcome by resorting to more advanced methods of estimating mutual information such as [41,42]. The MIB framework also admits several possible future extensions including scaling the framework to bigger networks and real-valued stochastic neural networks. The extension to real-valued stochastic neural networks are straightforward by, e.g., constructing a Gaussian layer for modeling $p(z_l|z_{l-1})$ and using reparameterization tricks [43] to perform back-propagation via sampling. Another dimension of improvement is to study hyperparameter effect of MIB. This current work only considers equal $\gamma_l = \gamma$ for `JointMIB` and equal $\beta_l = \beta$, and tuned β via grid search. We can use, e.g., Bayesian optimization [44] to efficiently tune γ_l and β_l with expandable bounds. In addition, we believe that the challenge of applying our methods to more advanced datasets such as Imagenet [45] is partly associated with that of scaling the stochastic neural network as we tend to need more expressive models for more challenging datasets. Given this perspective, the challenge to scale to large datasets can be partially addressed with the solutions from scaling stochastic neural networks some of which we suggest above. Furthermore, we believe that, as one of the main messages from our work, explicitly inducing compressed and relevant information (e.g., via mutual information as in MIB) into many intermediate layers can be more beneficial to large-scale tasks than simply resorting to the MLE principle. An intuition is to think of this as a way for *information-theoretic regularization for intermediate layers*. Finally, a followup important question to ask is whether there is any theoretical and stronger empirical link between an improved information representation (e.g., in the MIB sense) and the generalization of neural networks. This connection might be intuitively correct but a systematically empirical study or a theoretical suggestion are an important future research direction.

Author Contributions: Conceptualization by J.C. and T.T.N.; writing and conducting experiments supervised by J.C.; methodology, software, validation, formal analysis, investigation, visualization, and writing of the original draft by T.T.N.

Funding: This work was supported by the Institute for Information and Communications Technology Planning and Evaluation (IITP) grant (No.2017-0-01779, A machine learning and statistical inference framework for explainable artificial intelligence).

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

Abbreviations

The following abbreviations are used in this manuscript:

IB	Information Bottleneck
MIB	Markov Information Bottleneck
SNN	Stochastic Neural Network
DPI	Data Processing Inequality
MLE	Maximum Likelihood Estimation
SGD	Stochastic Gradient Descent
SFNN	Stochastic Feed-forward Neural Network
VIB	Variational Information Bottleneck

References

1. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. In Proceedings of the Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, USA, 22–24 September 1999.
2. Strouse, D.; Schwab, D.J. The Information Bottleneck and Geometric Clustering. *Neural Comput.* **2019**, *31*, doi:10.1162/neco_a_01136. [CrossRef] [PubMed]
3. Dai, B.; Zhu, C.; Guo, B.; Wipf, D.P. Compressing Neural Networks using the Variational Information Bottleneck. In Proceedings of the 35th International Conference on Machine Learning (ICML 2018) Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018; pp. 1143–1152.
4. Achille, A.; Soatto, S. Emergence of Invariance and Disentanglement in Deep Representations. *J. Mach. Learn. Res.* **2018**, *19*, 50:1–50:34.
5. Yamada, M.; Heecheol, K.; Miyoshi, K.; Yamakawa, H. FAVAE: Sequence Disentanglement using Information Bottleneck Principle. *arXiv* **2019**, arXiv:1902.08341 .
6. Jeon, I.; Lee, W.; Kim, G. IB-GAN: Disentangled Representation Learning with Information Bottleneck GAN. 2019. Available online: https://openreview.net/forum?id=ryljV2A5KX (accessed on 25 September 2019)
7. Tschannen, M.; Djolonga, J.; Rubenstein, P.K.; Gelly, S.; Lucic, M. On Mutual Information Maximization for Representation Learning. *arXiv* **2019**, arXiv:1907.13625.
8. Tishby, N.; Polani, D. Information Theory of Decisions and Actions. In *Perception-Action Cycle*; Springer: New York, NY, USA, 2011; pp. 601–636.
9. Goyal, A.; Islam, R.; Strouse, D.; Ahmed, Z.; Larochelle, H.; Botvinick, M.; Levine, S.; Bengio, Y. Transfer and Exploration via the Information Bottleneck. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
10. Friedman, N.; Mosenzon, O.; Slonim, N.; Tishby, N. Multivariate Information Bottleneck. In Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence, Seattle, WA, USA, 2–5 August, 2001.
11. Chechik, G.; Globerson, A.; Tishby, N.; Weiss, Y. Information Bottleneck for Gaussian Variables. *J. Mach. Learn. Res.* **2005**, *6*, 165–188.
12. Rey, M.; Roth, V. Meta-Gaussian Information Bottleneck. In Proceedings of the Annual Conference on Neural Information Processing Systems, NIPS, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1925–1933.
13. Strouse, D.; Schwab, D.J. The Deterministic Information Bottleneck. In Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence (UAI 2016), New York, NY, USA, 25–29 June 2016; Ihler, A.T., Janzing, D., Eds.; AUAI Press: Corvallis, OR, USA, 2016.
14. Alemi, A.A.; Fischer, I.; Dillon, J.V.; Murphy, K. Deep Variational Information Bottleneck. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
15. Tishby, N.; Zaslavsky, N. Deep learning and the information bottleneck principle. In Proceedings of the IEEE Information Theory Workshop (ITW), Jerusalem, Israel , 26 April–1 May 2015; pp. 1–5.
16. Shwartz-Ziv, R.; Tishby, N. Opening the Black Box of Deep Neural Networks via Information. *arXiv* **2017**, arXiv:1703.00810.
17. Amjad, R.A.; Geiger, B.C. Learning Representations for Neural Network-Based Classification Using the Information Bottleneck Principle. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [CrossRef] [PubMed]
18. Hinton, G.E. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Comput.* **2002**, *14*, 1771–1800, doi:10.1162/089976602760128018. [CrossRef] [PubMed]
19. Hinton, G.E.; Osindero, S.; Teh, Y.W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527–1554, doi:10.1162/neco.2006.18.7.1527. [CrossRef] [PubMed]
20. Tang, Y.; Salakhutdinov, R. Learning Stochastic Feedforward Neural Networks. In *Advances in Neural Information Processing Systems 26: Proceedings of the 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–10 December 2013*; Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran: Norwich, UK, 2013; pp. 530–538.
21. Raiko, T.; Berglund, M.; Alain, G.; Dinh, L. Techniques for Learning Binary Stochastic Feedforward Neural Networks. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
22. Florensa, C.; Duan, Y.; Abbeel, P. Stochastic Neural Networks for Hierarchical Reinforcement Learning. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017.

23. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
24. Nguyen, T.T.; Choi, J. Layer-wise Learning of Stochastic Neural Networks with Information Bottleneck. *arXiv* **2017**, arXiv:1712.01272.
25. Nguyen, T.T. Parametric Information Bottleneck to Optimize Stochastic Neural Networks. Master's Thesis, Ulsan National Institute of Science and Technology, Ulsan, Korea, 2018.
26. Slonim, N. Information Bottleneck Theory and Applications. Ph.D. Thesis, Hebrew University of Jerusalem, Jerusalem, Israel, 2003.
27. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; Wiley Series in Telecommunications and Signal Processing; Wiley: New York, NY, USA, 2006.
28. Saxe, A.M.; Bansal, Y.; Dapello, J.; Advani, M.; Kolchinsky, A.; Tracey, B.D.; Cox, D.D. On the Information Bottleneck Theory of Deep Learning. In Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 April–3 May 2018.
29. Cheng, Y.; Wang, D.; Zhou, P.; Zhang, T. A Survey of Model Compression and Acceleration for Deep Neural Networks. *arXiv* **2017**, arXiv:1710.09282.
30. Rasmussen, C.E.; Ghahramani, Z. Occam's Razor. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2001, pp. 294–300.
31. Arora, S.; Ge, R.; Neyshabur, B.; Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018; pp. 254–263.
32. Bengio, Y. Learning Deep Architectures for AI. *Found. Trends Mach. Learn.* **2009**, *2*, 1–127. [CrossRef]
33. Williams, R.J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.* **1992**, *8*, 229–256, doi:10.1007/BF00992696. [CrossRef]
34. Bengio, Y.; Léonard, N.; Courville, A.C. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *arXiv* **2013**, arXiv:1308.3432.
35. Hinton, G. Lecture 9.3—Using Noise as a Regularizer. In *Neural Networks for Machine Learning*; University of Toronto: Toronto, ON, USA, 2016.
36. Zeiler, M.D. ADADELTA: An Adaptive Learning Rate Method. *arXiv* **2012**, arXiv:1212.5701,
37. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR, 2015, San Diego, CA, USA, 7–9 May 2015.
38. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.J.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014.
39. Nguyen, A.M.; Yosinski, J.; Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 427–436, doi:10.1109/CVPR.2015.7298640. [CrossRef]
40. Carlini, N.; Wagner, D.A. Towards Evaluating the Robustness of Neural Networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP 2017), San Jose, CA, USA, 22–26 May 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 39–57, doi:10.1109/SP.2017.49. [CrossRef]
41. Lin, X.; Sur, I.; Nastase, S.A.; Divakaran, A.; Hasson, U.; Amer, M.R. Data-Efficient Mutual Information Neural Estimator. *arXiv* **2019**, arXiv:1905.03319.
42. Belghazi, M. I.; Baratin, A.; Rajeshwar, S.; Baratin, A.; Ozair, S.; Bengio, Y.; Hjelm, R.D.; Courville, A.C. Mutual Information Neural Estimation. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsässan, Stockholm, Sweden, 10–15 July 2018; pp. 530–539.
43. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014.

44. Ha, H.; Rana, S.; Gupta, S.; Nguyen, T.; Tran-The, H.; Venkatesh, S. Bayesian Optimization with Unknown Search Space. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2019, Vancouver, BC, Canada, 8–14 December 2019.

45. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, Florida, 20–25 June 2009.