PATHOLOGIES IN INFORMATION BOTTLENECK FOR DETERMINISTIC SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Information bottleneck (IB) is a method for extracting information from one random variable X that is relevant for predicting another random variable Y. To do so, IB identifies an intermediate "bottleneck" variable T that has low mutual information $I(X;T)$ and high mutual information $I(Y;T)$. The ***IB curve*** characterizes the set of bottleneck variables that achieve maximal $I(Y;T)$ for a given $I(X;T)$, and is typically explored by optimizing the ***IB Lagrangian***, $I(Y;T) - \beta I(X;T)$. Recently, there has been interest in applying IB to supervised learning, particularly for classification problems that use neural networks. In most classification problems, the output class Y is a deterministic function of the input X, which we refer to as "deterministic supervised learning". We demonstrate three pathologies that arise when IB is used in any scenario where Y is a deterministic function of X: (1) the IB curve cannot be recovered by optimizing the IB Lagrangian for different values of β; (2) there are "uninteresting" solutions at all points of the IB curve; and (3) for classifiers that achieve low error rates, the activity of different hidden layers will not exhibit a strict trade-off between compression and prediction, contrary to a recent proposal. To address problem (1), we propose a functional that, unlike the IB Lagrangian, can recover the IB curve in all cases. We finish by demonstrating these issues on the MNIST dataset.

1 INTRODUCTION

The *information bottleneck* (IB) method (Tishby et al., 1999) provides a principled way to extract information that is present in one variable that is relevant for predicting another variable. Given two random variables X and Y, IB posits a "bottleneck" variable T that obeys the Markov condition $Y - X - T$. By the data processing inequality (DPI) (Cover & Thomas, 2012), this Markov condition implies that $I(X;T) \geq I(Y;T)$, meaning the bottleneck variable cannot contain more information about Y than it does about X. In fact, any particular choice of the bottleneck variable T can be quantified by two terms: the mutual information $I(X;T)$, which reflects how much T compresses X, and the mutual information $I(Y;T)$, which reflects how well T can be used to predict Y.

In IB, bottleneck variables are chosen to maximize prediction $I(Y;T)$ given a constraint on the compression (Witsenhausen & Wyner, 1975; Ahlswede & Körner, 1975; Gilad-Bachrach et al., 2003),

$$F(r) := \max_{T \in \Delta} I(Y;T) \quad \text{s.t.} \quad I(X;T) \leq r\,, \tag{1}$$

where Δ is the set of all random variables T that obey the Markov condition $Y - X - T$. In practice, the IB curve is almost always explored not via the constrained optimization problem of Eq. (1), but rather by maximizing the so-called *IB Lagrangian*,

$$\mathcal{L}^{\beta}_{\text{IB}}(T) := I(Y;T) - \beta I(X;T)\,, \tag{2}$$

where $\beta \in [0,1]$ is a parameter that controls the trade-off between compression and prediction. The advantage of optimizing $\mathcal{L}^{\beta}_{\text{IB}}$ is that it avoids the non-linear constraint in Eq. (1)[1].

The values of $F(r)$ for different values of r specify the ***IB curve***. It is known that F is concave in r, though it may not be strictly concave (Witsenhausen & Wyner, 1975; Ahlswede & Körner, 1975;

[1] Note that optimizing $\mathcal{L}^{\beta}_{\text{IB}}$ is still a constrained problem in that $p(t|x)$ must be a valid conditional probability. However, this constraint is usually easier to handle, e.g., by using an appropriate parameterization.

Gilad-Bachrach et al., 2003). This seemingly minor issue of strict vs. non-strict concavity of the IB curve will play a central role in our analysis.

Several recent papers have drawn connections between IB and *supervised learning* (Shamir et al., 2010; Tishby & Zaslavsky, 2015), in particular classification using neural networks. In this context, X represents neural network inputs, Y represents the output classes, and T represents intermediate representations used by the learning architecture, such as the activity of hidden layer(s). The application of IB to supervised learning has opened a promising set of novel research directions. Some of this research investigates neural network training algorithms that optimize the IB Lagrangian (Kolchinsky et al., 2017; Alemi et al., 2016; Chalk et al., 2016), in this way allowing for the application of IB to high-dimensional, continuous random variables. Other research (Shwartz-Ziv & Tishby, 2017) has suggested, somewhat controversially (Saxe et al., 2018), that stochastic gradient descent (SGD) training dynamics may implicitly favor hidden layer activity that optimally balances compression and prediction. In particular, this research has suggested that earlier hidden layers may favor prediction rather than compression, while latter hidden layers may favor compression rather than prediction (Shwartz-Ziv & Tishby, 2017). More generally, there has been an implicit idea that intermediate representations that are optimal in the IB sense correspond to "interesting" or "useful" compressions of the input (Amjad & Geiger, 2018).

Note that in most supervised classification problems, the output class Y is assumed to be a deterministic function of the input X, i.e., $Y = f(X)$ for some single-valued function f, which we refer to as *deterministic supervised learning*. In this paper, we demonstrate that IB suffers from three fundamental pathologies when applied to any situation where Y is a deterministic function of X:

1. There is no one-to-one mapping between different points on the IB curve and maximizers of the IB Lagrangian $\mathcal{L}_{\text{IB}}^{\beta}$ for different values of β, meaning that the IB curve cannot be explored by maximizing $\mathcal{L}_{\text{IB}}^{\beta}$ while varying β. This occurs because when Y is a deterministic function of X, the IB curve has a piecewise-linear shape and is therefore not strictly concave. The dependence of the IB Lagrangian on the strict concavity of $F(r)$ has been previously noted (Gilad-Bachrach et al., 2003; Shwartz-Ziv & Tishby, 2017), but the pervasiveness of this pathology in classification scenarios has not been recognized. We analyze this issue and propose a solution in the form of an alternative objective function, which can be used to explore the IB curve even when Y is a deterministic function of X.
2. When Y is a deterministic function of X, all points on the IB curve contain "uninteresting" solutions. In particular, the entire IB curve can be generated as a mixture of trivial solutions. This suggests that IB-optimality is not sufficient for an intermediate representation to be an interesting or useful compression of input data.
3. For a neural network with several hidden layers that achieves a low probability of error, the hidden layers cannot display a strict trade-off between compression and prediction (in particular, different layers can only differ in the amount of compression, not prediction).

In this paper we focus on supervised classification problems, although our analysis applies to any use of IB when Y is a deterministic function of X, including some regression scenarios. In addition, in Appendix B, we show that the above pathologies also apply to the recently proposed *deterministic IB* variant of IB (Strouse & Schwab, 2017), in which the compression term is quantified using the entropy $H(T)$ rather than the mutual information $I(X;T)$. In that Appendix, we propose an alternative objective function that can be used to resolve the first pathology for dIB.

Note that a recent paper (Amjad & Geiger, 2018) also discusses pitfalls in using IB to analyze intermediate representations in supervised learning. That paper does not consider the particular pathologies that arise from the assumption that Y is a deterministic function of X, thus its arguments are largely complementary to ours.

In the next section, we review some of the proposed connections between supervised learning and IB. In Section 3, we show that when Y is a deterministic function of X, as occurs in most classification problems, the IB curve has a piecewise-linear, rather than strictly concave, shape. In Sections 4, 5 and 6, we discuss the three issues mentioned in the list above. In Section 7, we demonstrate these issues in a real-world example, using a neural-network implementation of IB on the MNIST dataset.

2 Supervised Classification and IB

In supervised learning, one is provided with a training dataset $\{x_i, y_i\}_{i=1..N}$ of inputs and outputs, as well as a parameterized family of conditional distributions, $\{q_\theta(y|x)\}$, where θ are parameters. We use the random variables X and Y to refer to the inputs and outputs respectively, as well as the sets of outcomes of those random variables. We use $\hat{p}(x, y)$ to indicate the empirical distribution of inputs and outputs in the training data.

It is usually assumed that the x's and y's are sampled i.i.d. from some "true" distribution $w(y|x)w(x)$. The high-level goal of supervised learning is to use the dataset to select parameters θ such that the distribution $q_\theta(y|x)$ is a good approximation of $w(y|x)$. Supervised learning is called *classification* when Y takes a finite set of values, and *regression* when Y is continuous-valued. In this manuscript, we focus on classification and, without loss of generality, assume that the set of possible output values can be written as the integers $Y = \{0, 1, \ldots, m\}$.

In practice, many supervised learning architectures use some kind of intermediate representation to make predictions about the output, such as hidden layers in neural networks. We use the random variable T to represent the activity of some particular hidden layer in a neural network (or, more generally, some intermediate representation in a supervised learning architecture). We assume that T is a (possibly stochastic) function of the inputs, as determined by some parameterized conditional distribution $q_\theta(t|x)$, thus T obeys the Markov condition $Y - X - T$. The mapping from inputs to hidden layer activity can be either deterministic (as traditionally done in neural networks) or stochastic (as used in some architectures (Alemi et al., 2016; Kolchinsky et al., 2017))[2]. The mapping from hidden layer activity to outputs is represented with the parameterized conditional distribution $q_\theta(y|t)$. The overall mapping from inputs to outputs implemented by a neural network with a hidden layer can be written as $q_\theta(y|x) := \int q_\theta(y|t) q_\theta(t|x) dt$.

For classification problems, training often involves selecting θ to minimize *cross-entropy loss*, $\mathcal{L}_{\text{CE}}(\theta) := -\frac{1}{N}\sum_{i=1}^{N} \log q_\theta(y_i|x_i)$. In the presence of a hidden layer, $\mathcal{L}_{\text{CE}}$ can be written as

$$\mathcal{L}_{\text{CE}}(\theta) = -\frac{1}{N}\sum_i \int q_\theta(t|x_i) \log q_\theta(y_i|t)\, dt = \mathbb{E}_{\hat{p}_\theta(Y,T)}\left[-\log q_\theta(Y|T)\right] , \tag{3}$$

where $\mathbb{E}_p$ indicates expectation with respect to p, $\hat{p}_\theta(y, t) := \frac{1}{N}\sum_i \delta(y, y_i) q_\theta(t|x_i)$ is the empirical distribution of outputs and hidden layer activity given the dataset, and δ is the Kronecker delta.

The cross-entropy loss in Eq. (3) can also be written as

$$\mathcal{L}_{\text{CE}}(\theta) = H(\hat{p}_\theta(Y|T)) + D_{\text{KL}}(\hat{p}_\theta(Y|T) \| q_\theta(Y|T)) \tag{4}$$

where H is (conditional) Shannon entropy, D_{KL} is Kullback-Leibler (KL) divergence, and $\hat{p}_\theta(y|t)$ is defined via the definition of $\hat{p}_\theta(y, t)$ above. Since KL is non-negative, cross-entropy loss is an upper bound on the conditional entropy $H(\hat{p}_\theta(Y|T))$. During training, one can decrease $\mathcal{L}_{\text{CE}}(\theta)$ either by decreasing the conditional entropy $H(\hat{p}_\theta(Y|T))$, or by decreasing the KL term by changing $q_\theta(Y|T)$ to better approximate $\hat{p}_\theta(Y|T)$. Minimizing $\mathcal{L}_{\text{CE}}(\theta)$ thus minimizes an upper bound on $H(\hat{p}_\theta(Y|T))$. This is equivalent to maximizing a lower bound on $I(\hat{p}_\theta(Y;T))$, since $I(\hat{p}_\theta(Y;T)) = H(\hat{p}(Y)) - H(\hat{p}_\theta(Y|T))$, where $H(\hat{p}(Y))$ is a constant that doesn't depend on θ.

We can now make explicit the relationship between supervised learning and IB. In IB, one is provided with a joint distribution $p(x, y)$, and then seeks a bottleneck variable T that obeys the Markov condition $Y - X - T$ and that minimizes $I(X;T)$ while maximizing $I(Y;T)$. In supervised learning, one is provided with an empirical distribution $\hat{p}(x, y)$, defines an intermediate representation T that obeys the Markov condition $Y - X - T$, and then (during training) minimizes cross-entropy loss, thus maximizing a lower bound on $I(Y;T)$. To make the analogy complete, one might choose θ so as to simultaneously minimize $I(X;T)$, i.e., seek hidden layers that provide compressed representations of input data (Alemi et al., 2016; Chalk et al., 2016; Kolchinsky et al., 2017). In fact, we use such an approach in our experiments on the MNIST dataset, as reported in Section 7.

Before proceeding, we note a fact that will play a central role throughout the rest of this paper: in most supervised classification problems, there is only one correct label associated with each

[2] If T is continuous-valued and a deterministic function of a continuous-valued X, $I(X;T)$ will generally be infinite (Saxe et al., 2018; Amjad & Geiger, 2018). One should either consider noisy or quantized mappings from inputs to hidden layers, which will generally have finite $I(X;T)$.

possible input. Formally, this means that given the empirical distribution $\hat{p}(y|x)$, one can write $Y = f(X)$ for some single-valued function f. This relationship between X and Y may arise because it holds under the "true" distribution $w(y|x)$ from which the training dataset is sampled, or simply because each input vector x occurs at most once in the training data (as happens in most real-world classification training datasets). Importantly, we make no assumptions about the relationship between X and Y under $q_\theta(y|x)$, the approximate parameterized distribution implemented by the supervised learning architecture. In particular, our analysis still holds if $q_\theta(y|x)$ is non-deterministic (e.g., as implemented by the softmax function, often used in neural networks). In addition, our results are based on analytically-provable properties of the IB curve (i.e., global optima of Eq. (1)), and do not concern issues of local optima that may be important in practical scenarios. They also do not concern issues related to generalization error and finite data sampling (Shamir et al., 2010; Tishby & Zaslavsky, 2015), which are relevant for analyzing the relationship between IB and out-of-sample performance.

It is important to note that not all classification problems are deterministic. The map from X to Y may be intrinsically noisy or non-deterministic (e.g., as considered by a subfield of machine learning called *multilabel classification* (Tsoumakas & Katakis, 2007)), or noise may be intentionally added as a regularization technique (e.g., as done in the *label smoothing* method (Szegedy et al., 2016)). Moreover, one of the pioneering papers on the relationship between IB and supervised learning (Shwartz-Ziv & Tishby, 2017) analyzed an artificially-constructed classification problem in which $p(y|x)$ was explicitly defined to be noisy (see their Eq. 10). However, the stochastic relationship between Y and X in that paper was arbitrary, implicitly chosen to avoid the kinds of pathologies analyzed here. Given these caveats, we believe that many if not most real-world classification problems do in fact exhibit a deterministic relation between X and Y.

3 The IB curve when Y is a deterministic function of X

Consider any X, Y where Y is discrete-valued and a deterministic function of X, so that $Y = f(X)$. There are two bounds on the maximal achievable $I(Y;T)$, $I(Y;T) \leq I(X;T)$ by the DPI, and $I(Y;T) \leq H(Y)$ (Cover & Thomas, 2012). To visualize these two bounds, as well as other results of our analysis, we use so-called "information plane" diagrams (Tishby et al., 1999). The information plane represents various possible bottleneck variables in terms of their compression ($I(X;T)$, horizontal axis) and prediction ($I(Y;T)$, vertical axis) values. The two bounds mentioned above, $I(Y;T) \leq I(X;T)$ and $I(Y;T) \leq H(Y)$, are plotted on an information plane diagram in Fig. 1.

In this section, we show that when Y is a deterministic function of X, the IB curve saturates both bounds mentioned above, and is therefore not strictly concave but rather piece-wise linear (thick gray line in Fig. 1).

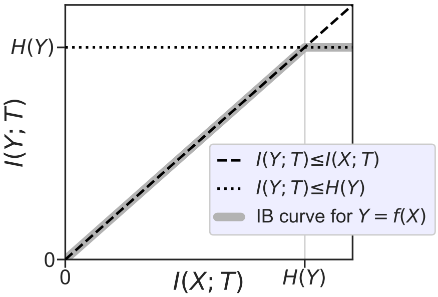

Figure 1: A schematic IB curve. Dashed line is DPI bound $I(Y;T) \leq I(X;T)$, dotted line is $I(Y;T) \leq H(Y)$. When Y is a deterministic function of X, the IB curve saturates both of these bounds, and looks like the thick gray line.

To show that the IB curve saturates the DPI bound $I(Y;T) \leq I(X;T)$ for $I(X;T) \in [0, H(Y)]$, we define a manifold of bottleneck variables T_α parameterized by $\alpha \in [0,1]$, where the set of outcomes of each T_α is the same as that of Y. Let B_α be a Bernoulli-distributed random variable that is equal to 1 with probability α and equal to 0 with probability $1-\alpha$. We define each T_α as

$$T_\alpha := B_\alpha \cdot Y = B_\alpha \cdot f(X)\,. \tag{5}$$

Thus, T_α is equal to Y with probability α, and equal to 0 with probability $1-\alpha$. From Eq. (5), it is clear that T_α is a (stochastic) function of both X and Y, thus both Markov conditions $Y - X - T_\alpha$ and $X - Y - T_\alpha$ hold. Applying the DPI to both Markov conditions gives $I(X;T_\alpha) \leq I(Y;T_\alpha)$ and $I(Y;T_\alpha) \leq I(X;T_\alpha)$, meaning that $I(T_\alpha;X) = I(T_\alpha;Y)$ for any T_α. Note that for $\alpha = 1$, $T_\alpha = Y$ and thus $I(T_\alpha;Y) = H(Y)$. At the same time, for $\alpha = 0$, $T_\alpha = 0$, thus $I(T_\alpha;Y) = 0$. Since mutual information is continuous in probabilities, the manifold of bottleneck variables T_α must sweep the full range of values $\langle I(X;T_\alpha), I(Y;T_\alpha)\rangle = \langle 0,0\rangle$ to $\langle I(X;T_\alpha), I(Y;T_\alpha)\rangle = \langle H(Y), H(Y)\rangle$ as α

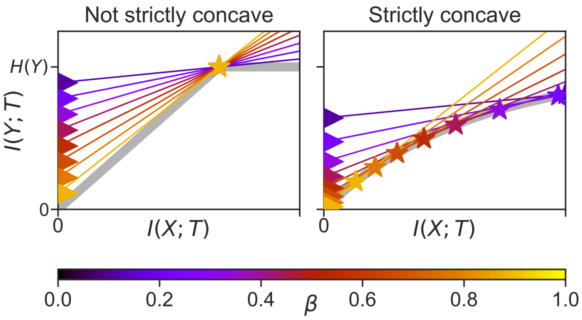

Figure 2: **Failure of IB Lagrangian.** Thick gray lines shows two possible IB curves, thin lines are isolines of $\mathcal{L}_{\text{IB}}^{\beta}$ for different β (colors indicated different β), stars indicate achievable $\langle I(X;T), I(Y;T)\rangle$ that maximize $\mathcal{L}_{\text{IB}}^{\beta}$. For not strictly concave IB curve, different $\beta \in (0,1)$ only recover a single point. For strictly concave IB curve, different β recover different points.

ranges from 0 to 1, all while obeying $I(T_\alpha;X) = I(T_\alpha;Y)$. Thus, the manifold of T_α bottleneck variables achieves the DPI bound over $I(T;X) \in [0, H(Y)]$.

Given its definition in Eq. (1), the IB curve must be monotonically increasing. Since $\langle H(Y), H(Y)\rangle$ is on the IB curve (e.g., T_α for $\alpha = 1$), it must be that $I(Y;T) \geq H(Y)$ whenever $I(X;T) \geq H(Y)$. At the same time, $I(Y;T) \leq H(Y)$ for all T. Thus, for all optimal bottleneck variables T with $I(X;T) \geq H(Y)$, the IB curve is a flat line with $I(Y;T) = H(Y)$.

We have shown that when Y is a deterministic function of X, the IB curve is piecewise-linear, having the shape of the thick gray line in Fig. 1, and thus not strictly concave. Note that the IB curve may also be not strictly concave in other situations. A sufficient condition for the IB curve to be strictly concave is for $p(y|x) > 0$ everywhere (Gilad-Bachrach et al., 2003, Lemma 6).

4 Issue 1: IB Curve cannot be explored using the IB Lagrangian

In this section, we show that when the IB curve is piecewise-linear, there is no one-to-one mapping between points on the IB curve and optimizers of the IB Lagrangian for different values of β. This means that one cannot recover bottleneck variables at different points on the IB curve by solving $T^\star \in \arg\max_{T\in\Delta} I(Y;T) - \beta I(X;T)$ while varying β. This issue is diagrammed in Fig. 2, and explained formally below.

Assume $Y = f(X)$. For any bottleneck variable T and all $\beta \in [0,1]$, $\mathcal{L}_{\text{IB}}^{\beta}$ is bounded by[3]

$$\mathcal{L}_{\text{IB}}^{\beta}(T) = I(Y;T) - \beta I(X;T) \leq (1-\beta) I(Y;T) \leq (1-\beta) H(Y) \tag{6}$$

where the first inequality uses the DPI, and the second $I(Y;T) \leq H(Y)$.

Now consider the bottleneck variable $T_{\text{copy}} := Y$ (or, equivalently, any one-to-one transformation of T_{copy}). Note that T_{copy} corresponds to T_α for $\alpha = 1$ as defined in Eq. (5), and is the "corner point" of the IB curve in Fig. 1, achieving $\langle I(X;T_{\text{copy}}), I(Y;T_{\text{copy}})\rangle = \langle H(Y), H(Y)\rangle$. Plugging these values into the definition of $\mathcal{L}_{\text{IB}}^{\beta}$ gives $\mathcal{L}_{\text{IB}}^{\beta}(T_{\text{copy}}) = (1-\beta)H(Y)$, which means that T_{copy} achieves the bound of Eq. (6) for all $\beta \in [0,1]$. Therefore, T_{copy} is an optimizer of $\mathcal{L}_{\text{IB}}^{\beta}$ for all $\beta \in [0,1]$, even though it corresponds to only a single point on the IB curve.

Now consider the IB Lagrangian for $\beta = 1$. For this β, $\mathcal{L}_{\text{IB}}^{\beta}(T) = I(Y;T) - I(X;T) \leq 0$, where the inequality follows from the DPI. Any T that has $I(X;T) = I(Y;T)$, such as the manifold of T_α defined in Eq. (5), achieves the maximum $\mathcal{L}_{\text{IB}}^{\beta}(T) = 0$. Therefore, all solutions on the non-flat part of IB curve in Fig. 1 simultaneously achieve the same maximum value of $\mathcal{L}_{\text{IB}}^{\beta}$ for $\beta = 1$.

Finally, consider the IB Lagrangian for $\beta = 0$. In this case, there is no penalty on $I(X;T)$ and $\mathcal{L}_{\text{IB}}^{\beta}(T) = I(Y;T)$, which is simultaneously maximized by all solutions that achieve the bound $I(Y;T) = H(Y)$. Therefore, all solutions on the flat part of the IB curve in Fig. 1 simultaneously achieve the same optimum value of $\mathcal{L}_{\text{IB}}^{\beta}$ for $\beta = 0$. (Note that supervised learning algorithms with cross-entropy loss can be considered to have $\beta = 0$, and their corresponding intermediate representations will generally fall into this regime.)

The pathology is illustrated visually in Fig. 2, which shows two hypothetical IB curves (in thick gray lines): a not strictly concave (piecewise-linear) curve (left), as occurs when Y is a deterministic

[3] It is sufficient to consider $\beta \in [0,1]$ because for $\beta < 0$, uncompressed solutions like $T = X$ are maximizers of $\mathcal{L}_{\text{IB}}^{\beta}$, while for $\beta > 1$, $\mathcal{L}_{\text{IB}}^{\beta}$ is non-positive and trivial solutions such as $T = \text{const}$ are maximizers.

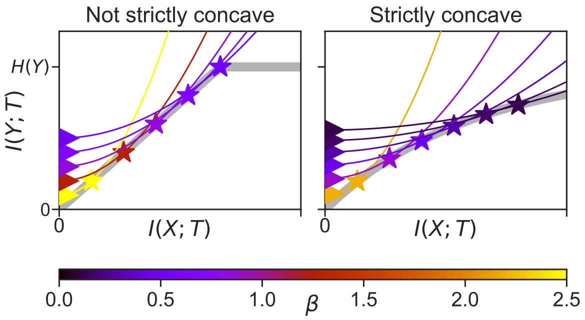

Figure 3: **Success of squared-IB functional**. Thick gray lines shows two possible IB curves, thin lines indicate isolines which have the same value of $\mathcal{L}^{\beta}_{\text{sq-IB}}$ for different values of β (indicated by color), the stars indicate achievable $\langle I(X;T), I(Y;T)\rangle$ that maximize $\mathcal{L}^{\beta}_{\text{sq-IB}}$. For both the not strictly concave IB curve and the strictly concave curve, different β values recover different solutions.

function of X, and a strictly concave curve (right). In both subplots, the colored lines indicate isolines of the IB Lagrangian. The highest point on the IB curve crossed by a given colored line is marked by a star, and represents the $\langle I(X;T), I(Y;T)\rangle$ values for a T which optimizes $\mathcal{L}^{\beta}_{\text{IB}}$ for the corresponding β. For the piecewise-linear IB curve, all $\beta \in (0,1)$ only recover a single solution (T_{copy}). For the strictly concave IB curve, different β values recover different solutions.

To summarize, when Y is a deterministic function of X, the IB curve is piecewise-linear and cannot be explored by optimizing $\mathcal{L}^{\beta}_{\text{IB}}$ while varying β . Of course, in principle any IB curve can be explored by solving the constrained optimization problem of Eq. (1) for different values of r (Miettinen, 1999, Thm. 3.2.2). In practice, however, solving this constrained optimization problem — even approximately — is far from a simple task, in part due to the non-linear constraint on $I(X;T)$. In particular, many off-the-shelf optimization tools cannot handle this kind of problem.

It is desirable to have an unconstrained objective function that can be used to explore any IB curve, whether strictly or non strictly concave. For this purpose, we propose an alternative objective function, which we call the *squared-IB functional*,

$$\mathcal{L}^{\beta}_{\text{sq-IB}}(T) := I(Y;T) - \beta I(X;T)^2 \,. \tag{7}$$

We show that maximizing $\mathcal{L}^{\beta}_{\text{sq-IB}}$ while varying β recovers all points on the increasing part of the IB curve. To do so, we first provide a bit of analysis of why the IB Lagrangian fails. Over the increasing region of the IB curve, the inequality constraint in Eq. (1) can be replaced by an equality constraint (Witsenhausen & Wyner, 1975, Thm 2.5), and maximizing $\mathcal{L}^{\beta}_{\text{IB}}$ can be written as $\max_T I(Y;T) - \beta I(X;T) = \max_r F(r) - \beta r$ (i.e., the Legendre transform of $-F(r)$). The derivative of $F(r) - \beta r$ is zero when $F'(r) = \beta$, and any point on the IB curve that has $F'(r) = \beta$ will maximize $\mathcal{L}^{\beta}_{\text{IB}}$ for that β. For a piecewise linear IB curve, all points on the increasing part of the curve have $F'(r) = 1$, and will all simultaneously maximize $\mathcal{L}^{\beta}_{\text{IB}}$ when $\beta = 1$.

Using a similar analysis, we write $\max_T \mathcal{L}^{\beta}_{\text{sq-IB}} = \max_r F(r) - \beta r^2$. The derivative of $F(r) - \beta r^2$ is 0 when $F'(r)/(2r) = \beta$. Since F is (not strictly) concave, $F'(r)$ is (not strictly) decreasing in r, and $F'(r)/(2r)$ is *strictly* decreasing in r. Thus, there can be at most one r such that $F'(r)/(2r) = \beta$ for a given β, meaning that there can only be one point on the IB curve that optimizes $\mathcal{L}^{\beta}_{\text{sq-IB}}$ for a given β. Therefore, the IB curve can be explored by maximizing $\mathcal{L}^{\beta}_{\text{sq-IB}}$ while varying β.[4]

Note that any point that satisfies $F'(r) = \beta$ also satisfies $F'(r)/(2r) = \beta/(2r)$. Thus, any point that maximizes $\mathcal{L}^{\beta}_{\text{IB}}$ for a given β also maximizes $\mathcal{L}^{\beta}_{\text{sq-IB}}$ under the transformation $\beta \mapsto \beta/(2 \cdot I(X;T))$, and vice versa. Importantly, unlike for $\mathcal{L}^{\beta}_{\text{IB}}$, there can be non-trivial maximizers of $\mathcal{L}^{\beta}_{\text{sq-IB}}$ for $\beta > 1$.

The effect of optimizing $\mathcal{L}^{\beta}_{\text{sq-IB}}$ is illustrated in Fig. 3. This figure shows two different IB curves (not strictly concave and strictly concave), isolines of $\mathcal{L}^{\beta}_{\text{sq-IB}}$ for different β, and points on the curves that maximize $\mathcal{L}^{\beta}_{\text{sq-IB}}$. For both IB curves, the maximizers correspond to different points for different β.

Finally, note that the IB curve is the Pareto front of the multi-objective optimization problem, $\{\text{maximize } I(Y;T), \text{minimize } I(X;T)\}$. $\mathcal{L}^{\beta}_{\text{sq-IB}}$ is one modification of $\mathcal{L}^{\beta}_{\text{IB}}$ that allows us to explore a non-strictly-concave Pareto front. However, it is not the only possible modification, and other alternatives may be considered (this may be particularly relevant when T lives in a restricted model class). For a full treatment of multi-objective optimization techniques, see Miettinen (1999).

[4] For simplicity, we've restricted our analysis to differentiable points on the IB curve (a more thorough analysis would concern the subderivatives of F). Note, however, that because F is concave, it must be differentiable almost everywhere (Rockafellar, 2015, Thm 25.5).

5 Issue 2: All points on IB curve have "uninteresting" solutions

It is often implicitly or explicitly assumed that optimal bottleneck variables on the IB curve will provide "useful" or "interesting" representations of the relevant information in X about Y (see also discussion and proposed criteria in (Amjad & Geiger, 2018)). While the concepts like usefulness and interestingness are subjective, the general intuition can be illustrated using the following simple example. Consider the ImageNet task (Deng et al., 2009), which involves labeling images according to 1000 classes, such as border collie, golden retriever, coffeepot, teapot, etc. It is natural to expect that as one explores the IB curve for the ImageNet dataset, one will identify useful compressed representations of the space of images. Such useful representations might, for instance, specify hard-clusterings that merge together inputs belonging to perceptually similar classes, such as border collie and golden retriever (but not border collie and teapot). In this section, we show that such intuitions do not necessarily hold whenever Y is a deterministic function of X.

Recall the analysis from Section 3, where Eq. (5) defines the manifold of bottleneck variables T_α parameterized by $\alpha \in [0, 1]$. The manifold of T_α, which spans the entire increasing portion of the IB curve, represents a mixture of two "trivial" solutions: a constant mapping to 0, and an exact copy of Y. (Note that the flat part of the IB curve can also be explored by a mixture of two trivial solutions, though we do not focus on this part of the curve here.) While the manifold of T_α bottleneck variables is optimal in the IB sense, these bottleneck variables do not offer interesting or useful representations of the input data. The compression offered by T_α arises by "forgetting" the input with some probability $1 - \alpha$, rather than performing any kind of useful coarse-graining, while the prediction comes from full knowledge of the mapping from inputs to outputs, $f : X \to Y$.

To summarize, when Y is a deterministic function of X, the fact that a bottleneck variable T is on the IB curve does not necessarily imply that the same bottleneck variable is useful compressed representation of X. At the same time, we do not claim that all bottleneck variables on the IB curve will have the trivial structure of Eq. (5). In fact, there may also be IB-optimal bottleneck variables that compress X in interesting and useful ways (however such notions may be formalized). Nonetheless, when Y is a deterministic function of X, for any "interesting" T, there will also be a "trivial" T_α that achieves the same values of compression $I(X; T)$ and prediction $I(Y; T)$, and IB does not distinguish between the two. Therefore, identifying useful compressed representations will generally require the use of quality functions other than just IB (Amjad & Geiger, 2018).

In Appendix B, we also analyze this issue for deterministic IB (dIB). In that case, we show that intermediate representations T that are hard-clusterings of the outputs Y will be optimal in terms of dIB. However, the resulting clusterings will not generally obey any intuitions about semantic or perceptual similarity between grouped-together elements of Y, and will thus also not necessarily provide any useful representations of the inputs.

6 Issue 3: No trade-off among different neural network layers

So far we've considered the relationship between supervised learning and IB in terms of a single intermediate representation T (e.g., a single hidden layer in a neural network). Recent research in machine learning, however, has focused on so-called "deep" neural networks, in which there are multiple successive hidden layers. What is the relationship between the compression and prediction achieved by these different layers? A recent theory (Shwartz-Ziv & Tishby, 2017) suggests that due to SGD training dynamics, different layers of a deep neural network will explore a strict trade-off between compression and prediction: early layers will sacrifice compression (high $I(T; X)$) for good prediction (high $I(T; Y)$), while latter layers will sacrifice prediction (low $I(T; Y)$) for good compression (low $I(T; X)$). Shwartz-Ziv & Tishby (2017) demonstrated a strict trade-off using an artificial classification dataset, in which Y was defined to be a noisy function of X (their Fig. 6). As we show here, however, such results cannot generally hold when Y is a deterministic function of X.

Recall that we consider IB in terms of the empirical distribution of inputs, hidden layers, and outputs given the training data (Section 2). Imagine that a classifier achieves 0, or near 0, probability of error (i.e., classifies every input correctly) on training data. This event, which can only occur when Y is a deterministic function of X, is in fact often observed in real-world neural networks. In such cases, we show that while latter layers may have better compression (lower $I(T; X)$) than earlier layers, they can't have worse prediction (lower $I(T; Y)$) than earlier layers. Therefore, different layers can only demonstrate a *weak* trade-off between compression and prediction. (The same arguments also hold if the neural network achieves near-0 probability of error on held-out testing data, as long as the information-theoretic measures are evaluated on the same held-out data distribution.)

Consider a neural network with k hidden layers, where each successive layer is a (possibly stochastic) function of the preceding layer. Let $T_1, T_2, \ldots, T_k$ indicate the activity of hidden layer $1, 2, \ldots, k$ respectively. Typically, the last hidden layer T_k is used to make a prediction about the output Y. We write this predicted Y as the random variable $\hat{Y} := a_\theta(T_k)$, where a_θ is taken as a deterministic function for simplicity. For instance, T_k might be mapped through a softmax function to give a vector of output class probabilities $q_\theta(Y|T_k)$, from which one selects the class with the highest probability, $\hat{Y} := \arg\max_y q_\theta(y|T_k)$. The probability of error — that is, the probability of predicting the incorrect output class — is then simply $P_e = \mathbb{E}[Y \neq \hat{Y}]$, where the expectation is over the empirical distribution of the training dataset as well as any stochasticity generated by the network itself (for instance if the layer-to-layer mappings are stochastic).

The above architecture obeys the Markov condition $Y - X - T_1 - T_2 - \cdots - T_k - \hat{Y}$, where we use the fact that the true output Y is a function of X. By the DPI, for any $i < j$ we have the inequalities

$$I(Y;T_i) \geq I(Y;T_j)\,, \quad (8) \qquad\qquad I(X;T_i) \geq I(X;T_j)\,. \quad (9)$$

Applying Fano's inequality (Cover & Thomas, 2012, Thm. 2.10.1) to the chain $Y - T_k - \hat{Y}$ gives

$$\mathcal{H}(P_e) + P_e \log|Y| \geq H(Y|T_k)\,, \tag{10}$$

where $|Y|$ is the number of output classes, and $\mathcal{H}$ is the binary entropy function, $\mathcal{H}(x) := -x\log x - (1-x)\log(1-x)$. Given our assumption that the probability of error obeys $P_e = 0$, Eq. (10) implies that $H(Y|T_k) = 0$ and the mutual information between the last layer and Y must obey $I(T_k;Y) = H(Y) - H(Y|T_k) = H(Y)$. Since $H(Y)$ is the largest mutual information that any random variable can have about Y, by Eq. (8) it must be that $I(T_i;Y) = H(Y)$ for all $i = 1...k$.

Our arguments do not undermine, or confirm, the high-level idea of Shwartz-Ziv & Tishby (2017), which states that SGD training dynamics may favor compression of hidden layer activity. Instead, our point is that if a classifier achieves 0 probability of error — which is only possible when Y is a deterministic function of X — then different layers must achieve the same amount of prediction $I(T_i;Y) = H(Y)$ about the output class, and a strict compression/prediction trade-off between different layers is impossible. At the same time, by Eq. (9), it may still be that latter layers compress the input more than earlier layers. Thus, there could be a *weak* trade-off between prediction and compression, i.e., latter layers will not give up any prediction but possibly gain some compression. In terms of Fig. 1, our arguments suggest that different layers will be on the flat part of the IB curve.

7 A REAL-WORLD EXAMPLE: MNIST

We demonstrate the three issues discussed above using the MNIST dataset of hand-written digits. We identify IB solutions for this dataset using the "nonlinear IB" method (Kolchinsky et al., 2017), which uses gradient-descent-based training to minimize cross-entropy loss plus a differentiable non-parametric estimate of $I(X;T)$ (Kolchinsky & Tracey, 2017). We use this technique to optimize an upper bound on the IB Lagrangian, as explained in (Kolchinsky et al., 2017), as well as (via a small modification) an upper bound on the squared-IB functional.

In our architecture, the bottleneck variable, $T \in \mathbb{R}^2$, corresponds to the activity of a hidden layer with two hidden units. Using a two-dimensional bottleneck variable facilitates easy visual analysis of its activity. The map from inputs X to bottleneck T has the form $T = a_\theta(X) + Z$, where $Z \sim \mathcal{N}(0, \mathbf{I})$ is the noise, and a_θ is a deterministic function implemented using three fully-connected layers: two layers with 800 ReLU units each, and a third layer with two linear units. Note that the stochasticity in the mapping from X to T makes our mutual information term, $I(X;T)$, well-defined and finite. The decoding map, $q_\theta(y|t)$, is implemented using a fully-connected layer with 800 ReLU units, followed by an output layer of 10 softmax units. Results are reported for training data. See Appendix A for more technical details and figures, including results on testing data. TensorFlow code is available at `anonymized`.

Like other practical general-purpose IB methods, "nonlinear IB" is not guaranteed to find globally-optimal IB bottleneck variables, due to factors like: (a) difficulty of optimizing the non-convex IB

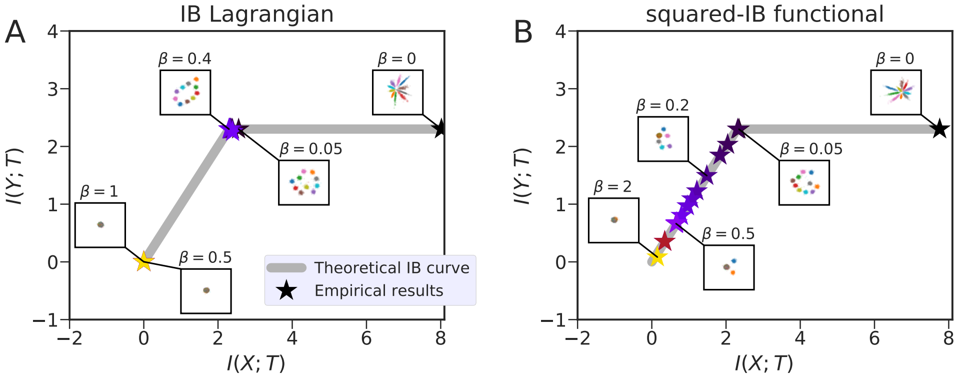

Figure 4: Theoretical and empirical IB curve found by maximizing the IB Lagrangian (A) and the squared-IB functional (B). Stars represent solutions recovered for different values of β. Insets show the bottleneck variable states (i.e., activity of the hidden layer) for different solutions, where colors represent inputs corresponding to different classes (i.e., different digits). MI values in nats.

objective; (b) error in estimating $I(X;T)$; (c) limited model class of T (i.e., T must be expressible in the form of $T = a_\theta(X) + Z$); (d) mismatch between actual decoder $q_\theta(y|x)$ and optimal decoder $\hat{p}_\theta(y|x)$ (see Section 2); and (e) stochasticity of training due to SGD. Nonetheless, in practice, the solutions discovered by nonlinear IB were very close to IB-optimal, and are sufficient to demonstrate all of the issues discussed in the previous sections.

Issue 1: IB Curve cannot be explored using the IB Lagrangian

We first demonstrate that the IB curve cannot be explored by maximizing the IB Lagrangian, but can be explored by maximizing the squared-IB functional, supporting the arguments in Section 4. Fig. 4 shows the theoretical IB curve for the MNIST dataset, as well as the IB curve empirically recovered by maximizing these two functionals (see also Fig. A5 for more details).

By optimizing the IB Lagrangian (Fig. 4A), we are only able to find three IB-optimal points:
(1) Maximum prediction accuracy and no compression, $I(Y;T) = H(Y)$, $I(X;T) \approx 8$ nats;
(2) Maximum compression possible at maximum prediction, $I(Y;T) = I(X;T) = H(Y)$;
(3) Maximum compression and zero prediction, $I(X;T) = I(Y;T) = 0$.

Note that the identified solutions are all very close to the theoretically-predicted IB-curve. However, the switch between the 2nd regime (maximal compression possible at maximum prediction) and 3rd regime (maximum compression and zero prediction) in practice happens at $\beta \approx 0.45$. This is different from the theoretical prediction, which states that this switch should occur at $\beta = 1.0$. The deviation from the theoretical prediction arises due to various practical details of the optimization done by the nonlinear IB method, as mentioned above. The switch from the 1st regime (no compression) to the 2nd regime happened as soon as $\beta > 0$, as predicted theoretically.

In contrast to the IB Lagrangian, by optimizing the squared-IB functional (Fig. 4B), we discover solutions located along different points on the IB curve for different values of β.

Additional insight is provided by visualizing the bottleneck variable $T \in \mathbb{R}^2$ (i.e., hidden layer activity) for both types of experiments. This is shown for different β in the scatter plot insets in Fig. 4. As expected, the IB Lagrangian experiments displayed three types of bottleneck variables: non-compressed variables (regime 1), compressed variables where each of the 10 classes is represented by its own compact cluster (regime 2), and a trivial solution where all activity is collapsed to a single cluster (regime 3). For the squared-IB functional, a different behavior was observed: As β increases, multiple classes become clustered together and the total number of clusters decreased. Thus, nonlinear IB with the squared-IB functional learned to group X into a varying number of clusters, in this way exploring the full trade-off between compression and prediction.

Issue 2: All points on IB curve have "uninteresting" solutions

By maximizing the squared-IB functional, we could find (nearly) IB-optimal solutions along different points of the IB curve. Here we show that such solutions do not provide particularly useful representations of the input data, supporting the arguments made in Section 5.

Note that "stochastic mixture"-type solutions, in particular the family of T_α discussed in Section 3 and Section 5, are not in our model class, since they cannot be expressed in the form $T = a_\theta(X)+Z$. Instead, our implementation favors "hard-clustering" solutions in which all inputs belonging to a given output class are mapped to a compact, well-separated cluster in the space of hidden layer activity (note that inputs belonging to multiple output classes may be mapped to a single cluster). For instance, from Fig. 4B, we can see that there are solutions with 10 clusters ($\beta = 0.05$), 6 clusters ($\beta = 0.2$), 3 clusters ($\beta = 0.5$), and 1 cluster ($\beta = 2.0$). Interstingly, such hard-clusterings are characteristic of optimal solutions for deterministic IB (dIB), as discussed in Appendix B. At the same time, in our results, the classes are not clustered in any particularly meaningful or useful way, and clusters contain different combinations of classes for different solutions. For instance, the solution shown for squared-IB functional with $\beta = 0.5$ has 3 clusters, one of which contains the classes $\{0, 2, 3, 4, 5, 6, 8, 9\}$, another contains the class 1, and the last contains the class 7. However, in other runs for the same β value, different clusterings of the classes arose. Moreover, because the different classes appear with close-to-uniform frequency in the MNIST dataset, any solution that groups the 10 classes into 3 clusters of size $\{8, 1, 1\}$ will achieve similar values of $I(X;T), I(Y;T)$ as the solution shown for $\beta = 0.5$.

Issue 3: No trade-off among different neural network layers

For both types of experiments, runs with $\beta = 0$ minimize cross-entropy loss only, without any regularization term that favors compression. Such runs are examples of "vanilla" supervised learning, though with stochasticity in the mapping between the input and the 2-node hidden layer.

For $\beta = 0$ runs, the 2-node hidden layer achieved nearly-perfect prediction, $I(T;Y) \approx H(Y)$. However, as shown in the scatter plots in Fig. 4, these hidden layer activations fell into spread-out clusters, rather than point-like clusters seen for $\beta > 0$. In general, hidden layer activity did not exhibit compression, meaning that the hidden layer retained information about X that was irrelevant for predicting the class Y, and fell onto the flat part of the IB curve.

Recall that our neural network architecture has three earlier hidden layers that precede T, as well as one hidden layer that succeeds it. Due to the DPI inequalities Eqs. (8) and (9), the earlier hidden layers must have worse compression than T, while the latter hidden layer must have better compression than T. At the same time, $\beta = 0$ runs achieve nearly 0 probability of error on the training dataset (results not shown), meaning that all layers must achieve $I(Y;T) \approx H(Y)$, the maximum possible. Thus, for $\beta = 0$ runs, the activity of the all hidden layers is located on the flat part of the IB curve, demonstrating a lack of a strict trade-off between prediction and compression.

8 Conclusion

There has been increasing interest in the connection between IB and supervised learning, especially in the context of classification using neural networks. In most classification problems, the output class Y is a deterministic function of X. In this work, we showed that in such scenarios, IB suffers from pathologies that give it a qualitatively different behavior from when the mapping from X to Y is stochastic. First, the IB curve cannot be recovered by maximizing the IB Lagrangian $I(Y;T) - \beta I(X;T)$ while varying β. Second, all points on the IB curve will contain "trivial" representations of inputs, and in fact, when Y is a deterministic function of X, one can recover the entire IB curve in a closed-form way, without performing any optimization. Finally, classifiers that achieve vanishing probability of error cannot have a strict trade-off between prediction and compression among successive layers, contrary to recent proposals. Our findings do not apply exclusively to supervised learning, but rather to any scenario where Y is a deterministic function of X.

Our results should not be taken to mean that the application of IB to supervised learning is without merit. First, they do not apply to various non-deterministic classification problems where the output is stochastic. Second, even for deterministic supervised learning, one may still wish to favor compression during training, possibly to improve factors like interpretability, generalization performance, or robustness to adversarial inputs. In this case, however, our work shows that to achieve varying rates of compression, one should use a different objective function than the IB Lagrangian.

REFERENCES

Rudolf Ahlswede and János Körner. Source coding with Side Information and a Converse for Degraded Broadcast Channels. *IEEE Transaction on Information Theory*, pp. 9, 1975.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep Variational Information Bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Rana Ali Amjad and Bernhard C Geiger. Learning representations for neural network-based classification using the information bottleneck principle. *arXiv preprint arXiv:1802.09766*, 2018.

Matthew Chalk, Olivier Marre, and Gasper Tkacik. Relevant sparse codes with variational information bottleneck. In *Advances in Neural Information Processing Systems*, pp. 1957–1965, 2016.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. Ieee, 2009.

Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. An Information Theoretic Tradeoff between Complexity and Accuracy. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Bernhard Schölkopf, and Manfred K. Warmuth (eds.), *Learning Theory and Kernel Machines*, volume 2777, pp. 595–609. Springer Berlin Heidelberg, 2003.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Artemy Kolchinsky and Brendan D Tracey. Estimating mixture entropy with pairwise distances. *Entropy*, 19(7):361, 2017. Corrected version available at `https://arxiv.org/abs/1706.02419`.

Artemy Kolchinsky, Brendan D. Tracey, and David H. Wolpert. Nonlinear Information Bottleneck. *arXiv preprint arXiv:1705.02436*, May 2017.

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.

Kaisa Miettinen. Nonlinear multiobjective optimization, volume 12 of international series in operations research and management science, 1999.

Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.

AM Saxe, Y Bansal, J Dapello, M Advani, A Kolchinsky, BD Tracey, and DD Cox. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations*, 2018.

Ohad Shamir, Sivan Sabato, and Naftali Tishby. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29-30):2696–2711, 2010.

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

Dj Strouse and David J. Schwab. The Deterministic Information Bottleneck. *Neural Computation*, pp. 1–20, April 2017.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *37th Allerton Conf on Communication*, 1999.

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

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

H. Witsenhausen and A. Wyner. A conditional entropy bound for a pair of discrete random variables. *IEEE Transactions on Information Theory*, 21(5):493–501, 1975.

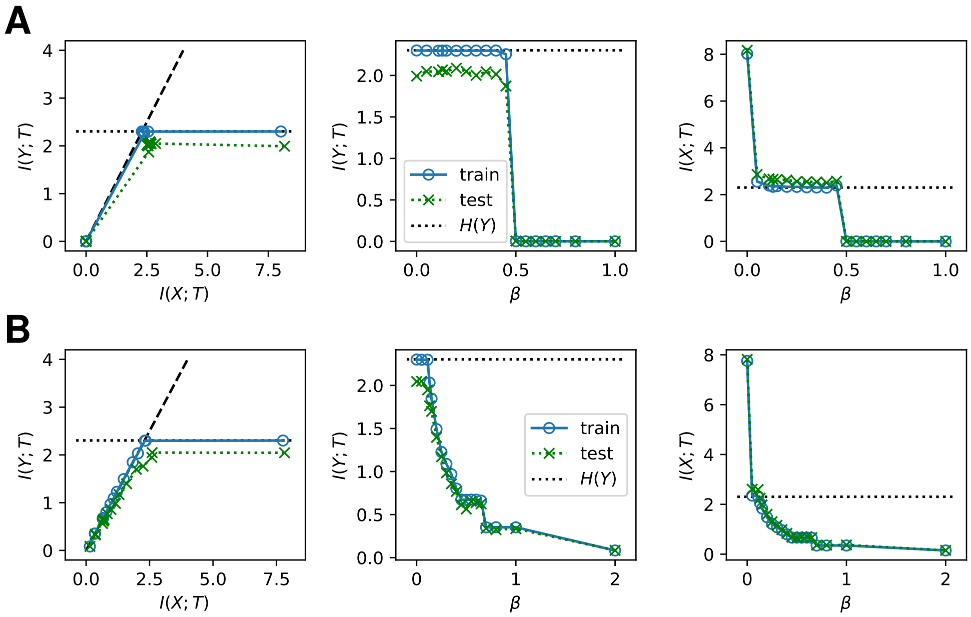

Figure A5: The top row (A) shows results for the IB Lagrangian, $I(Y;T) - \beta I(X;T)$, and the bottom row (B) shows results for the squared-IB functional, $I(Y;T) - \beta I(X;T)^2$. In each row, the left column shows the information plane (compression $I(X;T)$ versus prediction $I(Y;T)$), with the black dashed line showing the DPI bound $I(Y;T) \le I(X;T)$; the middle column shows prediction $I(Y;T)$ as a function of β; the right column shows compression $I(X;T)$ as a function of β. In all plots, the solid blue line indicates values calculated for the training data set, the dashed green line indicates values calculated for the held-out testing data set, and the black dotted line indicates $H(Y) = \ln 10$. All information-theoretic quantities are plotted in nats.

A Details of MNIST experiments

In Section 7, we demonstrate our results on the MNIST dataset (LeCun et al., 1998). This dataset contains a training set of 60,000 images and a test set of 10,000 images, each labeled according to digit. $X \in \mathbb{R}^{784}$ is defined to be a vector of pixels for a single 28×28 image, and $Y \in \{0, 1, ..., 9\}$ is defined to be the class label. Our experiments were carried out using the “nonlinear IB” method (Kolchinsky et al., 2017). $I(X;T)$ was computed using the kernel-based mutual information upper bound (Kolchinsky & Tracey, 2017; Kolchinsky et al., 2017) and $I(Y;T)$ was computed using the lower bound $I(Y;T) \ge H(Y) - \mathbb{E}_{\hat{p}_\theta(Y,T)}\left[-\log q_\theta(Y|T)\right]$ (see Eq. (4)).

The neural network was trained using the Adam algorithm (Kingma & Ba, 2014) with a mini-batch size of 128 and a learning rate of 10^{-4}. Unlike the implementation in (Kolchinsky et al., 2017), the same mini-batch was used to estimate the gradients of both $I_\theta(X;T)$ and the cross-entropy term. Training was run for 200 epochs. At the beginning of each epoch, the order of training examples was randomized. To eliminate the effect of the local minima, for each possible value of β, we carried out 20 runs and then selected the run that achieved the best value of the objective function. TensorFlow code is available at `anonymized`.

Results for the MNIST dataset are shown in Fig. 4 and Fig. A5, computed for a range of β values. Fig. A5 shows results for both training and testing datasets, though the main text focuses exclusively on training data. It can be seen that while the solutions found by IB Lagrangian jump discontinuously from the “fully clustered” solution ($I(T;X) = I(T;Y) = H(Y)$) to the trivial solution ($I(T;X) = I(T;Y) = 0$), solutions found by the squared-IB functional explore the trade-off in a continuous manner. See figure captions for details.

B DETERMINISTIC INFORMATION BOTTLENECK

Here we show that our analysis also applies to a recently-proposed (Strouse & Schwab, 2017) variant of IB called *deterministic IB (dIB)*. dIB replaces the standard IB compression cost, $I(X;T)$, with the entropy of the bottleneck variable, $H(T)$. This can be interpreted as operationalizing compression costs via a source-coding, rather than a channel-coding, scenario.

Formally, in dIB one is given random variables X and Y. One then identifies bottleneck variables T that obey the Markov condition $Y - X - T$ and maximize the *dIB Lagrangian*,

$$\mathcal{L}_{\text{dIB}}^{\beta}(T) := I(Y;T) - \beta H(T) \tag{11}$$

for $\beta \in [0,1]$, which can be considered as a relaxation of the constrained optimization problem

$$F_{\text{dIB}}(r) := \max_{T \in \Delta} I(Y;T) \quad \text{s.t.} \quad H(T) \le r\,. \tag{12}$$

To guarantee that the compression cost is well-defined, T is typically assumed to be discrete-valued (Strouse & Schwab, 2017). We call F_{dIB} the *dIB curve*.

Before proceeding, we note that the inequality constraint in the definition of $F_{\text{dIB}}(r)$ can be replaced by an equality constraint,

$$F_{\text{dIB}}(r) := \max_{T \in \Delta} I(Y;T) \quad \text{s.t.} \quad H(T) = r \tag{13}$$

We do so by showing that $F_{\text{dIB}}(r)$ is monotonically increasing in r. Consider any T which maximizes $I(Y;T)$ subject to the constraint $H(T) = r$. By definition, the Markov condition $Y - X - T$ must hold. Now imagine some random variable D which obeys the Markov condition $Y - X - T - D$, and define a new bottleneck variable $T' := (T, D)$ (i.e., the joint outcome of T and D). We have

$$I(Y;T') = I(Y;T,D) = I(Y;T) + I(Y;D|T) = I(Y;T)\,,$$

where we've used the chain rule for mutual information. At the same time, D can always be chosen so that $H(T') = H(T,D) = r'$ for any $r' \ge r$. Thus, we have shown that there are always random variables T' that achieve at least $I(Y;T') = \max_{T:H(T)=r} I(I;T)$ and have $H(T') > r$, meaning that $F_{\text{dIB}}(r)$ is monotonically increasing in r. This means the inequality constraint in Eq. (12) can be replaced with an equality constraint.

As we will see, unlike the standard IB curve, F_{dIB} is not necessarily concave. Since F_{dIB} can be defined using equality constraints, one can rewrite maximization of $\mathcal{L}_{\text{dIB}}^{\beta}$ as $\max_T I(Y;T) - \beta H(T) = \max_r F_{\text{dIB}}(r) - \beta r$, the Legendre-Fenchel transform of $-F_{\text{dIB}}(r)$. By properties of the Legendre-Fenchel transform, this means that optimizers of $\mathcal{L}_{\text{dIB}}^{\beta}$ must lie on the concave envelope of F_{dIB}, which we indicate as F_{dIB}^{*}.

B.1 THE dIB CURVE WHEN Y IS A DETERMINISTIC FUNCTION OF X

As in standard IB, for a discrete-valued Y we have the inequality

$$I(Y;T) \le H(Y)\,. \tag{14}$$

However, instead of the standard IB inequality $I(Y;T) \le I(X;T)$, we now employ

$$I(Y;T) \le H(T)\,, \tag{15}$$

which makes use of the assumption that T is discrete-valued. The dIB curve will have the same bounds as those shown for the standard IB curve (Fig. 1), except that $H(T)$ replaces $I(X;T)$ on the horizontal axis.

Now consider the case where Y is a deterministic function of X, i.e., $Y = f(X)$. It is easy to check that $T_{\text{copy}} := f(X) = Y$ achieves equality for both Eqs. (14) and (15), and thus lies on the dIB curve. Since F_{dIB} is monotonically increasing, the dIB curve is flat and achieves $I(Y;T) = H(Y)$ for all $H(T) \ge H(Y)$.

We now consider the increasing part of the curve, $H(T) \in [0, H(Y)]$. We call any T which is a deterministic function of Y (that is, any $T = g(Y) = g(f(X))$, where g is any deterministic function) a "hard-clustering" of Y. Any hard-clustering of Y has $H(T) \le H(Y)$. At the same time, any hard-clustering of Y has $H(T|Y) = 0$, thus $I(Y;T) = H(T) - H(T|Y) = H(T)$, achieving the bound of Eq. (15). Thus, any hard-clustering of Y lies on the increasing part of the dIB curve. (Note that any hard-clustering of Y will also be a deterministic function of X, thus have $H(T|X) = 0$ and $I(X;T) = H(T) - H(T|X) = H(T) = I(Y;T)$, and will therefore also fall on the increasing part of the standard IB curve.)

At the same time, the dIB curve cannot be composed entirely of hard-clustering, since — under the assumption that Y is discrete-valued — there can only be a countable number of T's that are hard-clusterings of Y. Thus, the dIB curve must also contain bottleneck variables that are not deterministic functions of Y, thus have $H(T|Y) > 0$ and $I(Y;T) < H(T)$, and do not achieve the bound of Eq. (15). Geometrically-speaking, when Y is a deterministic function of X, the dIB curve must have a "step-like" structure over $H(T) \in [0, H(Y)]$, rather than increasing smoothly like the standard IB curve. These results are shown schematically in Fig. A6, where blue dots indicate hard clusters of Y.

As mentioned, optimizers of $\mathcal{L}^{\beta}_{\text{dIB}}$ must lie on the concave envelope of F_{dIB}, indicated by F^{*}_{dIB}. Clearly, the step-like dIB curve that occurs when Y is a deterministic function of X is not concave, and only hard-clusterings of Y lie on its concave envelope for $H(T) \in [0, H(Y)]$. In the analysis below, we will generally concern ourselves with optimizers which are hard-clusterings.

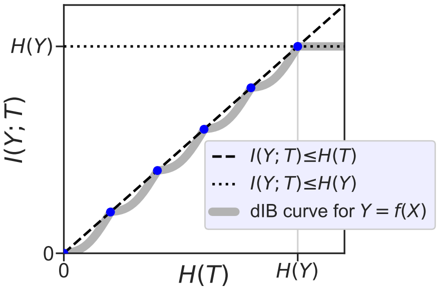

Figure A6: A schematic of the dIB curve. Dashed line is the bound $I(Y;T) \leq H(T)$, dotted line is $I(Y;T) \leq H(Y)$. When Y is a deterministic function of X, the dIB curve saturate the second bound always. Furthermore, it also saturates the first bound for any T that is a deterministic function of Y. The qualitative shape of the resulting dIB curve is shown as thick gray line.

B.2 The three pathologies

We now briefly consider the three pathologies discussed in the main text, in the context of deterministic IB. As usual, we assume that Y is a deterministic function of X.

Issue 1: dIB Curve cannot be explored using the dIB Lagrangian

In analogy to the analysis done in Section 4, we use the inequalities Eqs. (14) and (15) to bound the dIB Lagrangian as

$$\mathcal{L}^{\beta}_{\text{dIB}}(T) = I(Y;T) - \beta H(T) \leq (1-\beta) I(Y;T) \leq (1-\beta) H(Y)\,.$$

Now consider the bottleneck variable T_{copy}, for which $\mathcal{L}^{\beta}_{\text{dIB}}(T_{\text{copy}}) = (1-\beta)H(Y)$. Therefore, T_{copy} (or any one-to-one transformation of T_{copy}) will maximize $\mathcal{L}^{\beta}_{\text{dIB}}$ for all $\beta \in [0,1]$. It is also straightforward to show that when $\beta = 0$, all bottleneck variables residing on the flat part of the dIB curve will simultaneously optimize the dIB Lagrangian. Similarly, one can show that all hard-clusterings of Y, which achieve the bound of Eq. (15), will simultaneously optimize the dIB Lagrangian for $\beta = 1$. As before, this means that different points on the dIB curve do not generally correspond to optima of $\mathcal{L}^{\beta}_{\text{dIB}}$ for different values of β. Thus, there is no one-to-one map between points on the dIB curve and optimizers of the dIB Lagrangian for different β.

As in Section 4, we propose to resolve this pathology by maximizing an alternative objective function, which we call the *squared-dIB functional*,

$$\mathcal{L}^{\beta}_{\text{sq-dIB}}(T) := I(Y;T) - \beta H(T)^2\,. \tag{16}$$

We first demonstrate that any optimizer of the dIB Lagrangian must also be an optimizer of the squared-dIB functional. Consider that maximization of $\mathcal{L}^{\beta}_{\text{dIB}}$ can be written as $\max_T I(Y;T) - \beta H(T) = \max_r F^{*}_{\text{dIB}}(r) - r$. Then, for the point $\langle r, F^{*}_{\text{dIB}}(r)\rangle$ on the dIB curve to maximize $\mathcal{L}^{\beta}_{\text{dIB}}$, it must have $\beta \in \partial_r F^{*}_{\text{dIB}}(r)$, where ∂_r indicates the superderivative with regard to r. At the same time, for the point $\langle r, F^{*}_{\text{dIB}}(r)\rangle$ to maximize $\mathcal{L}^{\beta}_{\text{sq-dIB}}$, it must satisfy $0 \in \partial_r \left[F^{*}_{\text{dIB}}(r) - \beta r^2\right]$, or after rearranging,

$$2r\beta \in \partial_r F^{*}_{\text{dIB}}(r)\,. \tag{17}$$

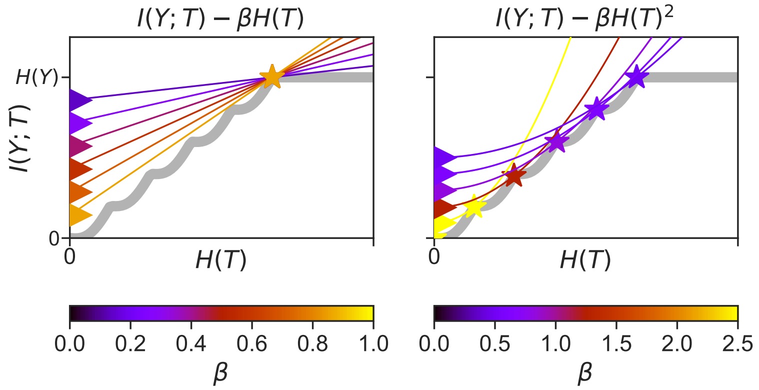

Figure A7: **Success of squared-dIB functional**. Colored lines indicate manifolds which have the same value of the dIB Lagrangian (left) and the squared-dIB functional (right) for different values of β, the stars indicate achievable $(H(T), I(Y;T))$ that maximize each function. For the dIB curve when $Y = f(X)$, different β values recover different solutions only for the squared-dIB functional.

It is easy to see that if $\beta \in \partial_r F^*_{\text{dIB}}(r)$ is satisfied, then Eq. (17) is also satisfied under the transformation $\beta \mapsto \frac{\beta}{2r}$. Thus, any optimizer of $\mathcal{L}^\beta_{\text{dIB}}$ will also optimize $\mathcal{L}^\beta_{\text{sq-dIB}}$, given $\beta \mapsto \frac{\beta}{2r}$.

We now show that different hard-clusterings of Y will optimize the squared-dIB functional for different values of β, meaning that we can explore the envelope of the dIB curve by optimizing $\mathcal{L}^\beta_{\text{sq-dIB}}$ while varying β. Formally, we show that for any given $\beta > 0$, the point

$$\langle H(T), I(Y;T)\rangle = \left\langle \frac{1}{2\beta}, \frac{1}{2\beta} \right\rangle \tag{18}$$

on the dIB curve will be a unique maximizer of $\mathcal{L}^\beta_{\text{sq-dIB}}$ for the corresponding β. Consider the value of $\mathcal{L}^\beta_{\text{sq-dIB}}$ for any T satisfying Eq. (18),

$$\mathcal{L}^\beta_{\text{sq-dIB}}(T) = \frac{1}{2\beta} - \beta\left(\frac{1}{2\beta}\right)^2 = \frac{1}{2\beta} - \frac{1}{4\beta} = \frac{1}{4\beta}$$

Now consider the value of of $\mathcal{L}^\beta_{\text{sq-dIB}}$ for any other T' on the dIB curve, which has $H(T') \neq \frac{1}{2\beta}$,

$$\begin{aligned}
\mathcal{L}^\beta_{\text{sq-dIB}}(T') &= I(Y;T') - \beta H(T')^2 \\
&= I(Y;T') - \beta\left(\left(H(T') - \frac{1}{2\beta}\right)^2 + \frac{H(T')}{\beta} - \frac{1}{4\beta^2}\right) \\
&\overset{(a)}{<} (I(Y;T') - H(T')) + \frac{1}{4\beta} \\
&\overset{(b)}{\leq} \frac{1}{4\beta} = \mathcal{L}^\beta_{\text{sq-dIB}}(T)\,.
\end{aligned}$$

Inequality (a) comes from the assumption that $H(T') \neq \frac{1}{2\beta}$, thus $(H(T') - 1/(2\beta))^2 > 0$, while inequality (b) comes from the bound $I(Y;T') \leq H(T')$. We have shown that $\mathcal{L}^\beta_{\text{sq-dIB}}(T') < \mathcal{L}^\beta_{\text{sq-dIB}}(T)$, meaning that any point on the dIB curve satisfying Eq. (18) for a given β, assuming it exists, will be the unique maximizer of $\mathcal{L}^\beta_{\text{sq-dIB}}$. The situation is diagrammed visually in Fig. A7.

Issue 2: All points on dIB curve have "uninteresting" solutions

The family of bottleneck variables T_α defined in Eq. (5), i.e., the mixture of two "trivial" solutions, are no longer optimal from the point of view of dIB. However, as mentioned above, any T that is a hard-clustering of Y achieves the bound of Eq. (15), and is thus on the dIB curve.

However, given a hard-clustering of Y, there is no reason for the clusters to obey any intuitions about semantic or perceptual similarity between grouped-together classes. To use the ImageNet example from Section 5, there is no reason for dIB to prefer a coarse-graining with "natural" groups like $\{\text{border collie}, \text{golden retriever}\}$ and $\{\text{teapot}, \text{coffeepot}\}$, rather than a coarse-graining with groups like $\{\text{border collie}, \text{teapot}\}$ and $\{\text{golden retriever}, \text{coffeepot}\}$, assuming those classes are of the same size. Distinguishing between such different clusterings requires some similarity or distortion measure between classes, which is not provided by the standard information theoretic measures.

Issue 3: No trade-off among different neural network layers

In Section 6, we showed that for a neural network with many hidden layers and vanishing probability of error, the activity of the different layers will lie along the flat part of the standard IB curve, where there is no strict trade-off between compression $I(X;T)$ and prediction $I(Y;T)$. Note that any bottleneck variable that lies along the flat part of a standard IB curve will have $I(X;T) \geq H(Y)$ and $I(Y;T) = H(Y)$. Using the standard information-theoretic inequality $H(T) \geq I(X;T)$, the same bottleneck variable must therefore have $H(T) \geq H(Y)$ and $I(Y;T) = H(Y)$, thus also lying on the flat part of the dIB curve. This shows that for a neural network with many hidden layers and vanishing probability of error, the activity of the different layers will also lie along the flat part of the dIB curve, and not demonstrate any strict trade-off between compression $H(T)$ and prediction $I(Y;T)$.