UNIVERSITY OF AMSTERDAM

MSc ARTIFICIAL INTELLIGENCE
MASTER THESIS

VALIDATING THE INFORMATION BOTTLENECK PRINCIPLE

---

by
NIKITA TOKOVENKO
12185892

August 27, 2020

36 EC
February 2020 - August 2020

Assessor:
DR PATRICK FORRÉ
AMLAB
UNIVERSITY OF AMSTERDAM

Supervisor:
MSC MARCO FEDERICI
AMLAB
UNIVERSITY OF AMSTERDAM

INFORMATICS INSTITUTE
AMLab

Nikita Tokovenko
Validating the Information Bottleneck principle

August 27, 2020
36 EC
February 2020 - August 2020
Examiner: Dr Patrick Forré
Supervisor: MSc Marco Federici

University of Amsterdam
AMLab
Informatics Institute
Science Park 904
1098 XH and Amsterdam

Abstract

Information-based models for deep learning has become a sought-after subject of ongoing research. Many recent methods for self-supervised and unsupervised representation learning train feature extractors using an estimate of the mutual information. Such an approach proved its ability to reach state-of-the-art performance in many applications. We believe that investigating the Information Bottleneck principle will give useful insights to find a key to understanding how to build more efficient and accurate algorithms to solve the tasks of deep learning.

The target goal of this thesis is to define a procedure to make one able to reason about the generalization capability of different encoding procedures. In order to do this we proposed a framework to make a comparison of various supervised encoding techniques.

Acknowledgement

> *The real challenge of growth mentally, emotionally, spiritually comes when you get knocked down. It takes courage to act.*
>
> — **Author unknown**

I want to thank everyone who believed in me and helped to keep the fire of curiosity about life burning.

Contents

1 Introduction

1.1 Motivation and Problem Statement

UPD: The deadline is 23:59 Monday, August 24th - Independence Day. Day of my independence.

Initially Information Bottleneck was formulated by. In this work authors presented a general formulation of the information theoretic approach for finding representations of the signal in a manner to capture its relevant structure. Back in the days there was no way to expressively use it in application to real-world problems in machine learning - the issue of applying Information Bottleneck (IB) directly to training deep neural networks is that estimation of mutual information, which is crucial for definition of the target function, is a tough and opened task stand-alone. Later in **deep learning ib**, after the deep learning revolution, the adaptation of the information bottleneck to deep neural networks has been proposed. Authors provided a theoretical overview of IB as a special type of Rate Distortion problem to be solved by neural networks. However, proposed procedure relied on optimizing objective using iterative Blahut Arimoto algorithm, which is unfeasible in application to deep neural networks.

Recent works in **variational mi** showed some promising results on bounding the estimate of the true mutual information. It is not entirely clear how tight these bounds should be in order to provide a solution for building high-quality representations. Also investing a lot of computational power might be unfeasible because building high-confidence bound on entropy requires an exponentially big sample size. An explanation on why maximizing MI does not necessarily lead to useful representations **max of mi**.

Recently proposed Variational Information Bottleneck (VIB) Alemi et al., 2016 addresses the problem of fitting stochastic encoders using the variational inference. Such approach showed to be more robust to overfitting while still achieving the state-of-the-art. In the supervised learning literature, their work is related to confidence penalty methd **conf penalty**, while in the unsupervised learning is closely related to the work of Kingma and Welling, 2014 on variational autoencoders. VIB, same

as VAE, regulates compression and fitting with only difference that target labels are available during training time.

Variational inference is a natural way to approximate the problem. Variational bounds on mutual information have previously been explored in Agakov (2004), though not in conjunction with the information bottleneck objective. Mohamed Rezende (2015) also explore variational bounds on mutual information, and apply them to deep neural networks, but in the context of reinforcement learning. We recently discovered Chalk et al. (2016), who independently developed the same varia- tional lower bound on the IB objective as us. However, they apply it to sparse coding problems, and use the kernel trick to achieve nonlinear mappings, whereas we apply it to deep neural networks, which are computationally more efficient. In addition, we are able to handle large datasets by using stochastic gradient descent, whereas they use batch variational EM. Finally, concurrent with our work and most closely related Achille Soatto (2016) propose and study a variational bound on the information bottleneck objective, from the perspective of variational dropout and demonstrate its utility in learning disentangled representations for variational autoencoders.

Supervised representation learning is fundamental in machine learning

Definition 1: Natural Numbers

Natural numbers are numbers used for counting and ordering.

Theorem 1: John's Theorem

$1 + 1 = 2$

Note 1: A note on John's Theorem

It follows that $1 + 2 = 3$

Example 1: An example

This is useful to show that $2 + 2 = 4$.

Recall Def. 1. As in Thm. 1, $1 + 1 = 2$. See also note 1 and the example 1.

1.2 Hypothesis and research questions

Our hypothesis: deep learning models that incorporate information bottleneck principle can learn high-level representations from labeled data of not much lower (or even higher) quality while working at higher compression rates when compared to

other ways to increase the robustness of features learnt (e.g. different regularization techniques).

In order to validate this hypothesis we aim to answer following research questions:

1. How true is that usage of information bottleneck method helps to compress the data without the loss of its expressiveness?

2. How can we quantitatively evaluate the quality of representations with respect to the amount of mutual information preserved?

3. How does the choice of the model architecture can affect this metric?

4. How does using information bottleneck objective relate to applying regularization to the model trained?

5. Can we design a more effective method to discard only irrelevant information with respect to VIB?

1. Set-up and framing of research - quality of outline and research questions + relevance to development of scientific field - legitimation and explanation of theoretical/historical framework and method - evidence of general knowledge and critical evaluation of national/international developments in the field and its history

2 Mathematical preliminaries

> *The best thesis defense is a good thesis offense.*
>
> — **XKCD**

2.1 Motivation and Problem Statement

Introduce Mutual information, entropy, kl divergence in the context of machine and deep learning mutual information is a measre pf true dpenedency between random varibale Write-down all definitions needed, theorems, properties add derivations of bounds.

3 Information-theoretic perspective of Deep learning

In this chapter, we investigate the application of Information Bottleneck (IB) to deep neural networks used to learn representations of the data. We formulate the general overview of the information-theoretic approach in the context of supervised representation learning. Also, we discuss how can we improve upon the existing methods that were supposed to be state-of-the-art (Alemi et al., 2016).

3.1 Extracting relevance using Information theory

In the information theory formulation the goal of representation learning is to define an encoding procedure to get the relevant quantization of the input signal X[1]. To build a space of compressed representations (or quantized codebook) $\widetilde{\mathcal{X}}$, we aim to find a possibly stochastic mapping characterized by p.d.f. $p(X|\widetilde{X})$. As in this section we assume both $\mathcal{X}$ and $\widetilde{\mathcal{X}}$ to be finite, by marginalizing all possible values of X we can define the prior probability distribution for our codewords, so we can access the probability measure $p(\widetilde{X})$ directly.

Traditionally, the average number of elements from $\mathcal{X}$ that correspond to the same element in the codebook is $2^{H(X|\widetilde{X})}$, where $H(X|\widetilde{X})$ is the conditional entropy of X given $\widetilde{X}$. Recall that mutual information $I(X;\widetilde{X})$ shows the reduction in the uncertainty about the value of the input X after observing its representation $\widetilde{X}$. Namely, it can be calculated as the difference:

$$I(X;\widetilde{X}) = H(X) - H(X|\widetilde{X}). \tag{3.1}$$

The question that arises is how to determine the quality of quantization built.

[1] Here $X \in \mathcal{X}$ denotes random variable (message) coming from the space of all possible signals $\mathcal{X}$ with provided fixed probability measure $p(X)$

3.1.1 Approaching relevant quantization via Rate Distortion theory

As we intend to obtain the compressed version of the original data, the quality of quantization can be assessed by calculating the rate[2]. For each possible message from $\mathcal{X}$ this value is bounded from below by $I(X;\widetilde{X})$. However, decreasing the information rate causes discarding attributes of the original signal, potentially worsening the predictive capability of representations.

In rate distortion theory this issue is addressed by introducing the distortion function, $d : \mathcal{X} \times \widetilde{\mathcal{X}} \to \mathbb{R}^+$, that is presumed to be small for good solutions. Naturally, having more attributes preserved by the encoding procedure leads to smaller expected distortion $\mathbb{E}_{p(X,\widetilde{X})} d(X,\widetilde{X})$, since representations are becoming more informative about the input. According to the rate distortion theorem of Shannon and Kolmogorov (Cover and Thomas, 1991) the trade-off can be characterized by the rate distortion function $R(D)$, where argument D corresponds to the maximum tolerable level of distortion. Now the problem can be formalized as a constraint optimization problem, since we aim to minimize the achievable rate while not exceeding some level of expected distortion:

$$R(D) = \min_{\{p(\widetilde{X}|X):\mathbb{E}_{p(X,\widetilde{X})} d(X,\widetilde{X}) \leq D\}} I(X;\widetilde{X}) \tag{3.2}$$

This is a variational problem of finding proper conditional p.d.f. $p(\widetilde{X}|X)$. Solutions to it can be found as the one to minimize the functional with introduced Lagrange multiplier β for the expected distortion we want to constrain:

$$\mathcal{F}[p(\widetilde{X}|X), \beta] = I(X;\widetilde{X}) + \beta \mathbb{E}_{p(X,\widetilde{X})} d(X,\widetilde{X}) \tag{3.3}$$

In such formulation this problem has the following closed-form solution:

$$p(\widetilde{X}|X) = \frac{p(\widetilde{X})}{Z(X,\beta)} e^{-\beta d(X,\widetilde{X})}, \tag{3.4}$$

where $Z(X,\beta)$ is a normalization function. For the given level of distortion D, the corresponding value of the Lagrange multiplier is positive and satisfies $\frac{\partial R}{\partial D} = -\beta$.

[2]Rate, in this case, defines the average number of bits per any message $x \in \mathcal{X}$ that is sufficient to specify the corresponding codeword with no probability of confusion

However, there is no easy way to define the right form of the distortion function in the general case making this solution inapplicable to real-world problems.

3.1.2 Relevance through Information Bottleneck

In the original work on the Information Bottleneck (IB) method (Tishby et al., 1999) authors address this issue by proposing an alternative way to define the relevance of quantization - by measuring the amount of relevant information preserved about another variable. Namely, if we are solving the task of predicting Y from X, Y must not be independent of X, causing the mutual information $I(X;Y)$ to be positive. Since $I(X;Y)$ shows exactly the amount of information available to solve the task, the reasonable intention would be to build the relevant quantization $\widetilde{\mathcal{X}}$ in a way to share the as much amount of information about Y ($I(\widetilde{X},Y)$) as possible. As compression is lossy, it is obvious that compressed representations cannot convey more information about the task than the original data, meaning $I(\widetilde{X},Y) \leq I(X;Y)$. However, the same as before the goal is to maximize the level of compression. We again face a trade-off between preserving meaningful information and compressing the representations.

Now optimization objective 3.3 can be reformulated as:

$$\mathcal{L}[p(\widetilde{X}|X),\beta] = I(X;\widetilde{X}) - \beta I(\widetilde{X},Y) \tag{3.5}$$

We can control the trade-off by adjusting only parameter β. As $\beta \to \infty$, we push the encoder to build an arbitrarily detailed quantization, while setting $\beta = 0$ will result in assigning a single codeword as a representative for all possible messages.

According to authors of (Tishby et al., 1999), the optimal assignment for $p(\widetilde{X}|X)$ that minimizes 3.5 should satisfy the equation:

$$p(\widetilde{X}|X) = \frac{p(\widetilde{X})}{Z(X,\beta)} e^{-\beta \sum_{y\in\mathcal{Y}} p(y|X) \log \frac{p(y|X)}{p(y|\widetilde{X})}}, \tag{3.6}$$

where all random variables follow a Markov chain under the condition $Y \leftrightarrow X \leftrightarrow \widetilde{X}$ and the distribution $p(Y|\widetilde{X})$ is obtained by the Bayes rule. Notice that the exponent argument corresponds to $D_{KL}(p(Y|X)||p(Y|\widetilde{X}))$ scaled by the factor β. Thus, solution becomes similar to 3.4 with the Kullback-Leibler divergence as the distortion measure.

In (Tishby et al., 1999) authors provide detailed derivation for 3.4 and 3.6, as well as, an iterative method for finding these unknown distributions based on the Blahut-Arimoto (BA) algorithm (Yeung, 2008).

3.2 Supervised learning prospect

In real-life supervised machine learning tasks, probabilistic spaces are rarely accessible. Instead we are provided with a dataset $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ of pairs of instances of random variables X and Y drawn from the joint distribution $p(X, Y) = P(Y|X)P(X)$. In such cases, X and Y are often referred to as input features and task outputs, respectively. Supervised learning aims to learn the approximation for the true conditional density $p(Y|X)$. This task has proved to be successfully solvable by deep neural networks (Hinton et al., 2012). Namely, we seek for a reconstruction $\hat{Y}$ characterized by the conditional density $q(\hat{Y}|X)$ parametrized by the set of trainable parameters $\boldsymbol{\theta}$, or simply $q(\hat{Y}|X;\boldsymbol{\theta})$, to be as close to the original task outputs as possible.

In deep representation learning we want to learn intermediate representations Z of the input features (e.g. hidden layers activations), such that all random variables follow the Markov condition $Y \leftrightarrow X \leftrightarrow Z \leftrightarrow \hat{Y}$. Given that, we can achieve the end-goal of learning $q(\hat{Y}|X;\boldsymbol{\theta})$ by learning encoding and decoding distributions $q(Z|X)$ and $q(\hat{Y}|Z)$ approximated by the neural network of our choice parametrized by $\boldsymbol{\theta}$[3].

3.2.1 Relevance through cross-entropy minimization

In the tasks of classification, when the variable Y is discrete and represents belonging to one of the target classes, the common choice for the cost function falls on the cross-entropy, which in our case can be formulated as the representation cross-entropy cost function:

$$J_{CE}(p(X,Y);\boldsymbol{\theta}) = \mathbb{E}_{(z,x)\sim q(Z|X;\boldsymbol{\theta})p(X)}\big[-\mathbb{E}_{y\sim p(Y|Z=z)}\log(q(y|Z=z;\boldsymbol{\theta}))\big], \quad (3.7)$$

where $p(Y|Z) = \mathbb{E}_{x\sim p(X)}\left[\frac{p(Y|X=x)q(Z|X=x;\boldsymbol{\theta})}{\mathbb{E}_{x\sim p(X)}\left[q(Z|X=x;\boldsymbol{\theta})\right]}\right]$.

Such choice is reasonable since by measuring the relative entropy we know how much information will be lost if to treat $p(Y|Z)$ and $q(\hat{Y}|Z)$ being same. Thus, by training neural networks concerning minimizing 3.7 we aim to learn the structure of the data that mimics its true structure characterized by $p(Y|X)$.

[3] In order to avoid confusion with $p(\cdot)$ we refer to the true distribution and with $q(\cdot)$ to the variational approximation. For the improved readability from now on we will assume all approximating distributions q to be parametrized by the set of trainable parameters $\boldsymbol{\theta}$ present in the model, making notation $q(\cdot;\boldsymbol{\theta})$ and $q(\cdot)$ being equal. We still use the expanded notation to show the dependence on the parameters explicitly where it is needed.

It is also known that minimizing cross-entropy implies maximizing the amount of relevant information $I(Z;Y)$ preserved by the intermediate variable (Proposition 1 of Rodríguez Gálvez, 2020). We will show this fact explicitly in the following section (see 3.9).

In turn, direct minimization of $I(X;Z)$ while either minimizing 3.7 (Theorem 1 of Vera et al., 2018) or maximizing $I(Z;Y)$ (Theorem 4 of Sharim et al., 2010) leads to the tightening of the generalization gap[4].

Thus, we built a bridge between the information-theoretic and supervised learning points of view on building representations of the data.

3.3 IB in the context of training deep neural networks

View of the Information Bottleneck in the context of training deep neural networks was first pointed in (Tishby and Zaslavsky, 2015). In this work, the authors address questions of theoretical limits on the efficiency of such training, as well as complexity bounds on learning empirical estimate of the mutual information based on the finite sample distributions. The worth-noting conclusion drawn by authors is that we should constrain the complexity of representations to sustain a tolerable level of generalization drop when given unseen data.

However, no experimental results were provided, since the learning method proposed relied on the BA algorithm. The latter makes training neural networks interminable due to the high-dimension nature of the data, such as images. Furthermore, the goal of the BA algorithm is to find an optimal partitioning of the input space $\mathcal{X}$, given the chosen space of representatives $\mathcal{Z}$ (or $\widetilde{\mathcal{X}}$, as formulated in Section 3.1), but not to learn the mapping directly from data.

Alternatively to the objective 3.5 one can aim to maximize the inverse functional. The motivation under it is that we narrow our focus on the problem of maintaining the performance level, while ensuring that a certain level of compression is achieved. This implies maximizing the following Lagrangian:

$$\mathcal{L}[q(Z|X), \beta] = I(Z;Y) - \beta I(X;Z) \tag{3.8}$$

[4]By this we refer to the error incurred by computing the empirical loss function as an average of point-wise losses for the dataset rather than the expectation over the entire joint distribution $P(X,Y)$, i.e. the real cost.

Note that such reformulation implies corresponding reparametrization of the trade-off parameter β.

But still, the major issue that arises when applying IB directly is that computing true mutual information is notoriously hard, as this requires knowledge of the marginal distributions of variables involved. Of course, in cases when the cardinality of sample space is finite, meaning that we can access the discrete distribution, IB can be applied directly, but often it is not the case.

3.4 Bounding the IB objective with variational inference

Instead of fighting hard trying to access the values of MI directly, we can derive a tractable lower bound on 3.8 using variational inference. Such variational approximation to the Information Bottleneck was first proposed in (Alemi et al., 2016) and the corresponding training method was given the name of Variational Information Bottleneck (VIB).

The core idea is to derive lower and upper bounds on $I(Z;Y)$ and $I(X;Z)$, respectively.

$$
\begin{aligned}
I(Z;Y) = \mathbb{E}_{(y,z)\sim p(Y|Z)p(Z)}\big[\log \frac{p(y|Z=z)}{p(y)}\big] = H(Y) + \mathbb{E}_{(y,z)\sim p(Y|Z)p(Z)}\big[\log p(y|Z=z)\big] = \\
H(Y) + \mathbb{E}_{z\sim p(Z)}\big[D_{KL}(p(Y|Z=z)||q(\hat{Y}|Z=z;\boldsymbol{\theta})\big] + \mathbb{E}_{(y,z)\sim p(Y|Z)p(Z)}\big[\log q(y|Z=z;\boldsymbol{\theta})\big] \geq \\
H(Y) + \mathbb{E}_{(y,z)\sim p(Y|Z)p(Z)}\big[\log q(y|Z=z;\boldsymbol{\theta})\big] = H(Y) - J_{CE}(p(X,Y);\boldsymbol{\theta}).
\end{aligned}
\tag{3.9}
$$

Since the entropy $H(Y)$ is constant, we can achieve maximization of $I(Z;Y)$ directly by minimizing $J_{CE}(p(X,Y);\theta)$. This approach is good since it only requires samples both from our stochastic encoder characterized by $q(Z|X)$ and samples from our joint data distribution $p(X,Y)$, as well as, a tractable variational approximation $q(\hat{Y}|Z)$, which in such case we have.

In turn, the variational upper bound on $I(X;Z)$ can be derived as follows:

$$
\begin{aligned}
I(X;Z) = \mathbb{E}_{(x,z)\sim q(Z|X)p(X)}\big[\log \frac{q(z|X=x;\boldsymbol{\theta})}{p(z)}\big] = \\
\mathbb{E}_{(x,z)\sim q(Z|X)p(X)}\big[\log q(z|X=x;\boldsymbol{\theta})\big] + \mathbb{E}_{z\sim p(Z)}\big[\log p(z)\big].
\end{aligned}
\tag{3.10}
$$

As computing the marginal $p(Z)$ might be hard, its variational approximation $q(Z)$ can be introduced. Thus, following the same idea based on the non-negativity of the Kullback-Leibler divergence as in 3.9:

$$I(X;Z) = \mathbb{E}_{(x,z)\sim q(Z|X)p(X)}\big[\log q(z|X=x;\boldsymbol{\theta})\big] - \mathbb{E}_{z\sim p(Z)}\big[D_{KL}(p(z)||q(z))\big] + \mathbb{E}_{z\sim p(Z)}\big[\log q(z)\big] \leq \mathbb{E}_{(x,z)\sim q(Z|X)p(X)}\big[\log q(z|X=x;\boldsymbol{\theta})\big] + \mathbb{E}_{z\sim p(Z)}\big[\log q(z)\big] = \mathbb{E}_{(x,z)\sim q(Z|X)p(X)}\big[D_{KL}(q(z|X=x;\boldsymbol{\theta})||q(z))\big] \tag{3.11}$$

Traditionally, the choice of $q(z)$ falls to a standard Gaussian $\mathcal{N}(Z|0,I)$. By putting 3.9 and 3.11 together we obtain the following training objective for VIB:

$$L(p(X,Y),\boldsymbol{\theta},\beta) = J_{CE}(p(X,Y);\boldsymbol{\theta}) - \beta\mathbb{E}_{(x,z)\sim q(Z|X)p(X)}\big[D_{KL}(q(z|X=x;\boldsymbol{\theta})||q(z)\big] \tag{3.12}$$

We can model the encoder $q(Z|X)$ to correspond to output the minimal sufficient statistics for the Gaussian distribution $\mathcal{N}(Z|f_e^{\mu}(X), f_e^{\Sigma}(X))$. To avoid backpropagation through the sampling operation, we can use the reparametrization trick (Kingma and Welling, 2014) to introduce the noise term that is independent of the parameters of the model $\boldsymbol{\theta}$. Using the experimentation results provided in (Alemi et al., 2016) authors conclude that VIB works as a decent regularizer in and of itself.

3.5 Learning tighter bounds via Superfluous information minimization

But can we do better? What if we learn the approximation of the true posterior distribution $p(Z|X)$ modeled by the encoder $q(Z|X;\boldsymbol{\theta})$ while being regularized to fit more complex variational prior $q(Z)$ rather than just Unit Gaussian?

The idea of modeling complex multi-modal distributions has found a wide range of applications in deep learning using Normalizing Flows (Rezende and Mohamed, 2015, Dinh et al., 2016). We won't go far in this topic, as the focus of this thesis is mainly in supervised representation learning. Anyways, it's worth saying that such approach gained some success in finding complex structure in the real-world data, although it constrains the variety of architecture choices, due to the requirement of all transformations to be invertible, meaning that we cannot directly represent data in a lower-dimensional, compressed space.

As the task labels are available during the training time we could infer the information about the partition of the compressed space under certain regions being occupied by only associating points of the correspondent target class. We can see this as modeling the embedding space to be characterized by a mixture of distributions. Namely, if we model these distributions to be Gaussian we obtain the mixture of Gaussians (Chapter 9 from Bishop, 2006).

How can we achieve this in the VIB set-up? By introducing additional variational distribution modeled by the "inverse" decoder $q(Z|Y)$ we can infer the knowledge of labels directly. Thus, we model another distribution $\mathcal{N}(Z|f_d^{\mu}(X), f_d^{\Sigma}(X))$. We believe that learning more flexible regularizing distributions might lead to improved generalization capability of the representations produced.

This switches our focus from minimizing the KL-divergence in 3.12 to a modified lower bound $D_{KL}(q(Z|X)||q(Z|Y)$. We again said lower bound not to no purpose. In turn, it can be reformulated as follows:

$$D_{KL}(q(Z|X)||q(Z|Y)) = I(X;Z|Y) + D_{KL}(p(Z|Y)||q(Z|Y)), \tag{3.13}$$

where conditional mutual information $I(X;Z|Y)$ is often referred to as the superfluous information. It measures the amount of information conveyed in representations that cannot be used to learn predicting the task. Hence, preserving this information only increases the complexity of representations making the intention to get rid of it reasonable.

More precise inspection of this can be done by subdividing $I(X;Z)$ using the chain rule of mutual information:

$$I(X;Z) = I(X;Z|Y) + I(X;Y) - I(X;Y|Z). \tag{3.14}$$

Note that the predictive information, $I(X;Y)$, is constant and defined by the dataset, while the amount of predictive information not in Z, $I(X;Y|Z)$, has to be 0 for sufficient representations. Thus, the minimization of $I(X;Z)$ is equal to minimizing $I(X;Z|Y)$ for sufficient representations. According to Proposition 2.1 in (Federici et al., 2020), sufficient representation is minimal whenever $I(X;Z|Y)$ is minimal. This means that by directly optimizing superfluous information we can obtain representations presumed to be more optimal in terms of its complexity.

Furthermore, following the chain rule of mutual information, superfluous information in the general case can be defined as:

$$I(X;Z|Y) = I(X;Z) + H(Y|X) + H(Y|Z) - H(Y|X,Z) - H(Y). \tag{3.15}$$

Note that given the Markov condition $Y \leftrightarrow X \leftrightarrow Z$, Y and Z are conditional independent when X is observed. This means that once we know values of X, having additional information on the values of Z will not decrease the uncertainty about the values of Y. Hence, $H(Y|X,Z) = H(Y|X)$. Substituting this into 3.15 gives us:

$$\begin{aligned} I(X;Z|Y) = I(X;Z) + H(Y|X) + H(Y|Z) - H(Y|X) - H(Y) = \\ I(X;Z) + H(Y|Z) - H(Y) = I(X;Z) - I(Z;Y). \end{aligned} \tag{3.16}$$

This results gives us an intuition that having 0 bits of superfluous information implies having representations Z to convey the same amount of relevant information about the task Y as is present in the original data X ($I(X;Z) = I(Y;Z)$). In turn, this lets us to assume that for sufficient representations superfluous information should not just be minimal, as stated before, but to be equal to 0.

Given all above, the final modified optimization objective can be given as follows:

$$L(p(X,Y),\boldsymbol{\theta},\beta) = J_{CE}(p(X,Y);\boldsymbol{\theta}) - \beta \mathbb{E}_{(x,y,z)\sim p(Y|X)q(Z|X)p(X)}\big[D_{KL}(q(z|X=x)||q(z|Y=y)\big] \tag{3.17}$$

This approach can be stated to be novel as opposed to previously presented in the literature. Worth noting that we came up with such an idea during the intermediate phase of work on this thesis. We name such a method for supervised representation learning as the Conditional Variational Information Bottleneck (CVIB). However, while working on its development we found out the concurrent work on the Conditional Entropy Bottleneck (Fischer, 2019). Discouraging to us, authors also aimed to focus on minimizing superfluous information, although their motivation relies on the defined Minimum Necessary Information (MNI) criterion rather than on the improvement of regularizing priors. On top of that, they only aim to learn the mean modeled by $f_d^{\mu}(X)$ while keeping variance $f_d^{\Sigma}(X)$ to be unit, i.e. fixed and not learned at all. We study the comparison of these two training methods and report correspondent results and discussion in Chapter 6. Taking into account us being unaware of the existence of this work when choosing the direction of research of

ours and the fact of this work is unpublished, we believe our contribution to the research community may not be underestimated.

4 Validation through estimation and evaluation

Way to validate the principle: Discussing the way to evaluate models: mutual information estimation and its limitations Mutual information estimation, biases, how to reason using this values and what is the goal Linear evaluation protocol describe evaluation metric, including all problems faced

In this chapter we address the problem of evaluating the quality of representation learning algorithms in unified information-based framework. We describe methods and approaches to infer the model's generalization capability from the amount of mutual information preserved. Such problem was already addressed by Cheng et al., 2018

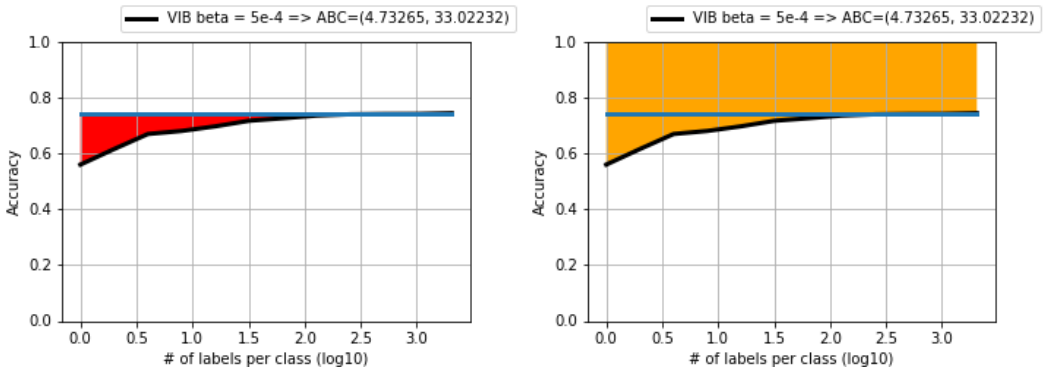
4.1 Mutual Information Estimation

Even though most of the methods investigated in this thesis do not require estimating mutual information directly during training, getting estimates of the real values gives us an ability to evaluate the compressing capability of deep neural networks.

Mutual information (MI) measures the amount of information that can be obtained about some random variable Y by observing another random variable X. Quantifying the degree of relationship between values from so variables gives us a metric to better assess the true dependency between

4.2 Modifying the linear evaluation protocol

To access the generalization capability of the models we use a custom modification of the linear evaluation protocol. The core idea is to build a linear classifier (e.g. logistic regression) on top of (frozen) representations and then evaluate the accuracy of solutions for the downstream task. In turn, we extend this metric to measure the residual performance given a different amount of additional training examples. We believe it to be extremely useful for the evaluation of solutions to problems when obtaining extra data might be expensive, so we want to know the estimated



(a) Max achievable performance gap **(b)** Max possible performance gap

Fig. 4.1: Visualization of the areas defining ABC loss

performance gap between maximum achievable performance (accuracy of classifier fine-tuned using the entire training set) across cases with different amount of labels per class modeling the case when the amount of additional data provided is cut-off. We can put all these gaps together by evaluating the area between two curves: first - the accuracy curve for a different amount of labels available, and second - the level of maximum achievable performance. We give a name for this metric of Area Between Curves or, simply, ABC loss. Thus, models that can encode data without the loss of its expressiveness while needing less additional examples to achieve requested performance correspond to small values of ABC loss, while those not able to generalize well correspond to higher values of ABC.

Worth mentioning is that logistic regression has a closed-form solution allowing to access the optimal solution for the weights fast. This property of logistic regression appears to be extremely useful, as it gives us an opportunity to increase the significance of results by averaging across an arbitrary amount of random seeds without a high loss of time-efficiency.

The drawback of such an evaluation approach is that we might obtain the same values of the ABC loss for encoders of different strength and a level of quality. Hence, we need to incorporate these loss values with something else to distinguish models of bad initial performance. While having accuracy values for every label partition available, we can measure the gap between the accuracy curve and the maximum performance ever possible - 100% accuracy. Having both of these values modifies the proposed metric to output not a single value, but a tuple consisting of two values. Visual representation of the gaps mentioned can be found in 4.1. Such an approach allows us to state that model A is superior to model B, if for model A the first value in the ABC loss tuple is significantly smaller than the one for model B while having second values to be more or less equal.

In such a way, we provide a unified tool to quantitatively reason about the generalization capability of the models studied in this work.

4.2.1 Notes on implementation

gencap: choice of number for number of samples, evaluation on the most likeliest outcome - mean, averaging over 40 seeds.

5 Experimental setup

5.1 Datasets

To validate the hypothesis outlined in Section 1.2 we investigate training deep neural models using one artificially created toy dataset and two image datasets.

We start our investigation with the toy dataset introduced in (Shwartz-Ziv and Tishby, 2017) to show the validity of the metrics chosen for further comparison. We synthesize inputs as vectors of 10 binaries, and the outputs are just single binaries. The inputs could be represented by integers from 0 to 1023 ($= 2^{10} - 1$). The 1024 possible inputs are divided into 16 groups (each group has 64 numbers), and each integer input $n \in [0, 1023]$ belongs to group i if $x \equiv i \pmod{16}$, where $i \in [0, 15]$. Each group i is then associated with a random binary number - we build kind of distribution over space of possible discrete states.

Next, to perform a comprehensive benchmarking to compare the impact of the different design choices covered in this work we use the MNIST12k dataset. Each model is trained and evaluated on the version of the original MNIST dataset, which consists of 12000 training images and 50000 test images of hand-written digits. The motivation for such a choice is that by doing so we decrease the number of training labels available during the training time, thus, weakening the models to make spotting the desired behavior easier.

Further, we expand our analysis to a more complex task of predicting classes for real-world images using the CIFAR10 dataset. Here we use the original version consisting of 50000 training and 10000 testing examples each corresponding to one of the following categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.

5.2 Models overview

To provide a comprehensive comparison we consider 4 major types of models: deterministic and stochastic neural networks trained with the application of common regularization techniques (weight decay (Krogh and Hertz, 1992) and dropout

(Srivastava et al., 2014)), stochastic neural networks trained using VIB and CVIB. Investigation of our method, CVIB, involves the study of different extents of flexibility for the learned conditional prior (e.g. to learn the variance or to keep it fixed).

To operate with MNIST12k we adopt the architecture used in (Alemi et al., 2016). Namely, we model variational distributions using MLPs of the form of encoder 784 - 1024 - 1024 - $2K$ for stochastic encoders and 784 - 1024 - 1024 - K for deterministic, where K is the size of the bottleneck, i.e dimensionality of the compressed space. Note that in stochastic variant last encoding layer has twice more parameters (first K outputs encode μ, while the remaining K outputs after softplus transformation, to ensure non-negativity, encode Σ) than its deterministic analog (intermediate representations are then characterized by a single embedding). The decoder is modeled by two successive fully-connected hidden layers both of size 64, resulting in the variational decoder of the form K - 64 - 64 - 10. We use a more complicated form of the decoder as opposed to the one used in (Alemi et al., 2016), on purpose to learn high-level representations in not necessarily linearly separable space, since the usage of simple logistic regression is involved in the adaptation of linear evaluation protocol described in 4.2. So for the sanity of evaluation, we do not use the same linear model on the down-stream task twice.

For learning on CIFAR10 we adopt the encoder to replicate the architecture of VGG11: 8 convolutional layers, each of which is applied in combination with the successive max-pooling operation followed by the activation, with one last dense layer to enter the embedding space. The architecture of the variational decoder was chosen to be consistent with the MNIST12k case described above: mapping of the form K - 64 - 64 - 10.

Across all experiments, we keep K=256. On top of that, there is something that is in common for all models investigated - in every case we use ReLU activations. Single-sided saturating nonlinearities, such as ReLUs, unlike $\tanh$, do not yield any compression as neural activations do not enter the saturating regime (Saxe et al., 2018). Hence, we do so, so not to induce any additional compression caused by the choice of architecture (unless specified explicitly).

Note that in neither case batch normalization was used and was turned off on purpose in cases where using it is presumed by default (e.g. when using VGG11-motivated encoder).

As an architecture for the neural mutual information estimator, we use an MLP of the form $(S_1 + S_2)$ - 1024 - 1024 - 1, where S_1 and S_2 are sizes of flattened feature vectors of variables we aim to measure the statistical dependency between concatenated across the feature dimension. For the toy dataset the validation of the

MI estimator is done for the case of having target model to be an MLP of the form 10 - 16 - 12 - 8 - 6 - 4 - 1.

5.3 Experiments

In this section we discuss all distinct experiments made throughout the whole process of work on this project.

5.3.1 Validating the Mutual Information Estimator

Since our end-goal is to make one being able to accurately reason about the properties of the trained models, before proceeding to the main part of our research we aim to validate the tool to measure the complexity of representations. Namely, mutual information estimator. To do so we put ourselves under conditions where we have access if not to a true value of mutual information, but at least to its more or less accurate empirical estimation for the toy dataset described in 5.1. This lets us ensure the quality of the JS-divergence based mutual information estimator used in this work.

5.3.2 Measuring the complexity change caused by introducing stochasticity in the encoders

Traditionally, the common choice for benchmarking object of comparison to the IB-motivated encoders is made in a favour of simple deterministic encoders. Note that using IB-based training procedure implies the stochastic nature of the encoder, since we aim to build a distribution over the compressed space and then provide representations as samples from it. In order to validate the idea of incorporating information-theoretic concepts to training deep neural networks, we first want to establish the relationship between the stochastic and deterministic natures of encoding procedures in terms of an ability of sustaining the generalization with respect to the compression rate achieved. We train stochastic and deterministic encoders trained under various levels of regularization applied. Apparently, the phenomenon that we aim to observe can be described as that increasing the strength of regularization should make the model to compress the data more when compared to the completely unregularized versions of encoding procedures studied. Moreover, we expect the compression ratio to increase monotonically when using higher values of weight decay or higher dropout rate.

5.3.3 Regularization via compression

Further, we discover the effect of applying the information bottleneck principle to capture meaningful representations using stochastic neural networks. We study the trade-off of preserving task relevant information versus discarding attributes of data at different level of The information bottleneck trade-off By discarding attributes of the data features . We train the VIB model given a set of values of interest for the choice of hyperparameter β

5.3.4 Additional experimental support

- Effect of switching distributions - Visualization on the 2-dimensional manifold for cvib vs ceb to discuss the effect of the learned variance - Fix weight initialization to pass every time to the model for different training method choice to see the effect of the change of the training objective (try to maximize the intersection between properties of trainable parameters for each model) - Measure the variance of the generalization capability curves: run encoders with same seed but multiple times to see the effect of weight initialization on the behavior of the model - to motivate the discussion of fixing

5.4 Notes on implementation

We train all our models using the Adam optimizer (Kingma and Ba, 2014).

5.4.1 Ensuring stability of MIE results

We consider points used in this work to increase the validity of mutual information estimations. Fitting

MIE: lr, pruning, convergence criteria - training for same amount epochs

gencap: choice of number for number of samples, evaluation on the most likeliest outcome - mean, averaging over 40 seeds.

5.4.2 Hypeparameters regions of interest management

Since the achievement of the desired compression rate cannot be guaranteed in advance, we perform several optimizations for different values of parameters that stand for the amount of regularization applied. Namely, we build experiments over

the set of parameters values of interest that we pre-define for every case considered. Note that the choice for the range of values for weight decay, dropout rate and trade-off parameter β might differ depending on the dataset, since different architectures were used to solve different tasks. plots, discussion on results

Interesting result: learnt sigma gives much higher level of compression with smaller choice of beta (mnist12k)

6 Results and Discussion

get literature review chapter

framing where the work is place in the bigger picture: compare to other papers who were studying same topic (before chapter 3)

the problem of scale of representation to the scale of initialization of weights of estimator

unless the information collapses we don't see any related compression for weight decay

7 Conclusion

> *Pass on what you have learned.*
>
> — **Master Yoda**

7.1 Future Work

> *Impossible to see, the future is.*
>
> — **Master Yoda**

Bibliography

Alemi, Alexander A., Ian Fischer, Joshua V. Dillon, and Kevin Murphy (2016). *Deep Variational Information Bottleneck* (cit. on pp. 1, 7, 12, 13, 22).

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag (cit. on p. 14).

Cheng, Hao, Dongze Lian, Shenghua Gao, and Yanlin Geng (2018). "Evaluating Capability of Deep Neural Networks for Image Classification via Information Plane". In: *European Conference on Computer Vision (ECCV)* (cit. on p. 17).

Cover, Thomas M. and Jay A. Thomas (1991). *Elements of Information Theory*. Wiley (cit. on p. 8).

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2016). "Density estimation using Real NVP". In: cite arxiv:1605.08803Comment: 10 pages of main content, 3 pages of bibliography, 18 pages of appendix. Accepted at ICLR 2017 (cit. on p. 13).

Federici, Marco, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata (2020). "Learning and generalization with the information bottleneck". In: *ICLR* (cit. on p. 14).

Fischer, Ian (2019). "The Conditional Entropy Bottleneck". In: (cit. on p. 15).

Hinton, G., L. Deng, D. Yu, et al. (2012). "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". In: *IEEE Signal Processing Magazine* 29.6, pp. 82–97 (cit. on p. 10).

Kingma, Diederik P. and Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015 (cit. on p. 24).

Kingma, Diederik P. and Max Welling (2014). "Auto-Encoding Variational Bayes." In: *ICLR*. Ed. by Yoshua Bengio and Yann LeCun (cit. on pp. 1, 13).

Krogh, Anders and John A. Hertz (1992). "A Simple Weight Decay Can Improve Generalization". In: *Advances in Neural Information Processing Systems 4*. Ed. by John E. Moody, Steve J. Hanson, and Richard P. Lippmann. San Francisco, CA: Morgan Kaufmann, pp. 950–957 (cit. on p. 21).

Rezende, Danilo and Shakir Mohamed (2015). "Variational Inference with Normalizing Flows". In: ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1530–1538 (cit. on p. 13).

Rodríguez Gálvez B.; Thobaben, R.; Skoglund M. (2020). "The Convex Information Bottleneck Lagrangian". In: *Entropy* 22.98 (cit. on p. 11).

Saxe, Andrew M., Yamini Bansal, Joel Dapello, et al. (2018). “On the Information Bottleneck Theory of Deep Learning.” In: *ICLR (Poster)*. OpenReview.net (cit. on p. 22).

Sharim, Ohad, Sivan Sabato, and Naftali Tishby (2010). “Learning and generalization with the information bottleneck”. In: *Theoretical Computer Science* 411.2696-2711 (cit. on p. 11).

Shwartz-Ziv, Ravid and Naftali Tishby (Mar. 2017). “Opening the Black Box of Deep Neural Networks via Information”. In: arXiv: 1703.00810 [cs.LG] (cit. on p. 21).

Srivastava, Nitish, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). “Dropout: a simple way to prevent neural networks from overfitting.” In: *Journal of Machine Learning Research* 15.1, pp. 1929–1958 (cit. on p. 22).

Tishby, Naftali, Fernando C. Pereira, and William Bialek (1999). “The Information Bottleneck method”. In: *The 37th annual Allerton Conference on Communication, Control, and Computing*, pp. 368–377 (cit. on p. 9).

Tishby, Naftali and Noga Zaslavsky (2015). “Deep learning and the information bottleneck principle”. In: *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5 (cit. on p. 11).

Vera, M., P. Piantanida, and L. R. Vega (2018). “The Role of the Information Bottleneck in Representation Learning”. In: *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 1580–1584 (cit. on p. 11).

Yeung, R.W. (2008). *Elements of Information Theory*. Springer, pp. 211–228 (cit. on p. 9).

Nomenclature

Notation

$\mathcal{X}$	sample space
X	random variable
x	an instance of random variable (e.g. value)

List of Definitions

List of Theorems

List of Figures

List of Tables