Learning CIFAR-10 with a Simple Entropy Estimator Using Information Bottleneck Objectives

Andreas Kirsch [1] **Clare Lyle** [1] **Yarin Gal** [1]

Abstract

The Information Bottleneck (IB) principle characterizes learning and generalization in deep neural networks in terms of the change in two information theoretic quantities and leads to a regularized objective function for training neural networks. These quantities are difficult to compute directly for deep neural networks. We show that it is possible to backpropagate through a simple entropy estimator to obtain an IB training method that works for modern neural network architectures. We evaluate our approach empirically on the CIFAR-10 dataset, showing that IB objectives can yield competitive performance on this dataset with a conceptually simple approach while also performing well against adversarial attacks out-of-the-box.

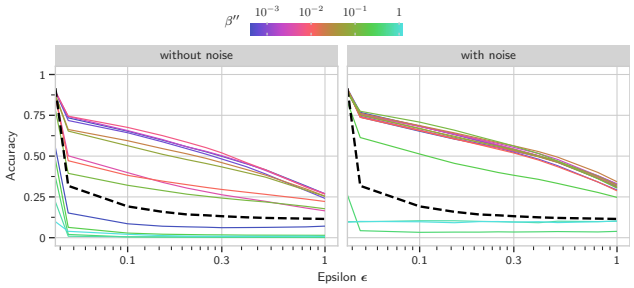


Figure 1. Adversarial robustness of ResNet18 models trained with our objective $\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta'' \mathrm{H}[\mathrm{Z}]$ *for different* β''. Models are trained on CIFAR-10, then evaluated on their robustness to FGSM attacks of varying ϵ values. We see that models trained with surrogate IB objectives (shown in coloured lines) largely see improved robustness over a model trained to minimize the regular cross-entropy training objective (shown in black) compared to our objective that does not use cross-entropies. We show the performance of models where we inject noise and models without (as a matter of ablation), and see that injecting noise is beneficial.

1. Introduction

The information bottleneck (IB) principle, introduced by Tishby et al. (2000), states that training a deep neural network that generalizes well can be expressed as a problem of finding a minimal representation of the input from which to predict its label. This notion of minimality is captured by the *mutual information* between the input X and its latent representation Z, denoted $\mathrm{I}[\mathrm{X};\mathrm{Z}]$, and the predictive accuracy is captured by the mutual information between the latent Z and the target Y, $\mathrm{I}[\mathrm{Y};\mathrm{Z}]$. The IB objective recasts training as the following constrained optimization problem

$$\min \mathrm{I}[\mathrm{X};\mathrm{Z}] \text{ s.t. } \mathrm{I}[\mathrm{Y};\mathrm{Z}] \geq C, \tag{1}$$

for some C that specifies the minimum amount of information that must be preserved in the latent. It can be relaxed to the following penalized optimization problem

$$\min \mathrm{I}[\mathrm{X};\mathrm{Z}] - \beta \mathrm{I}[\mathrm{Y};\mathrm{Z}]. \tag{2}$$

While the mutual information provides an appealing theoretical grounding for neural network training, it is notoriously difficult to compute. A number of alternate objectives have been proposed (Alemi et al., 2016; Strouse & Schwab, 2017), but these alternatives trade off to varying degrees the quality of the approximation to the information bottleneck, and the breadth of the class of models under consideration. There are few scalable IB training objectives in the literature which can be applied in a straightforward manner to current state-of-the-art architectures (Kirsch et al., 2020).

This work presents another such objective. Following Kirsch et al. (2020), we rewrite the IB objective as an explicit linear combination of entropies, for which we use a simple non-parametric entropy estimator that we can backpropagate through.

An analogous perspective would be to approximate the mutual information quantities in equation 2 by using differences of entropies of the form $\mathrm{I}[Y;Z] = \mathrm{H}[Z] - \mathrm{H}[Z \mid Y]$, following (McAllester & Stratos, 2018).

Training with this estimator, we recover competitive performance on the CIFAR-10 dataset, as well as good performance out-of-the-box against adversarial examples created

[1]OATML, Department of Computer Science University of Oxford, Oxford, UK. Correspondence to: Andreas Kirsch <andreas.kirsch@cs.ox.ac.uk>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning.

by FGSM (Goodfellow et al., 2014) compared to models trained with regular cross-entropy and dropout regularization (Srivastava et al., 2014). We obtain information plane diagrams that show our objective, indeed, creates an information bottleneck.

Our method is straightforward to apply to arbitrary network architectures, laying the groundwork for future investigation into the information-theoretic principles underlying state-of-the-art neural networks.

Altogether, it provides an intriguing perspective on using an old approach on modern neural network architectures with surprisingly good results. Our results show that the simple estimator suffers from continual compression as we keep on training, which will require future research.

2. Overview of IB Objectives

In their seminal work, Tishby et al. (2000) provide an optimal algorithm for the tabular case, when X, Y and Z are all categorical. This has spawned additional research to optimize the objective for other cases and specifically for DNNs (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017; Achille & Soatto, 2018). We will use the notation p_θ to refer to a (possibly probabilistic) encoder defining the conditional distribution $p_\theta(z|x)$, and q_θ to refer to a decoder. Typically, p_θ and q_θ will be neural networks, possibly with added stochasticity such as that due to dropout (Srivastava et al., 2014). Whether the benefits of training with IB *objectives* are due to the IB *principle*, or some other unrelated mechanism, remains unclear (Saxe et al., 2019; Amjad & Geiger, 2019; Tschannen et al., 2019), but independent of the validity of the IB principle in explaining deep learning, recent work has also tied the IB objective to successful results in both unsupervised and self-supervised learning (Oord et al., 2018; Belghazi et al., 2018; Zhang et al., 2018, among others)

Deterministic Information Bottleneck (DIB) presents a variation on the standard IB objective. Strouse & Schwab (2017) introduce the *deterministic* information bottleneck objective (DIB)

$$\min \mathrm{H}[Z] - \beta \mathrm{I}[Y;Z] \tag{3}$$

Like Tishby et al. (2000), they provide an algorithm for the tabular case. To do so, they examine an analytical solution for their objective as it is unbounded: $\mathrm{H}[Z] \to -\infty$ for the optimal solution. The DIB objective induces subtly different behavior in the latent representation, but its practical implementation faces similar hurdles to IB.

Deep Variational Information Bottleneck (DVIB) addresses the challenge of estimating the mutual information. Alemi et al. (2016) rewrite the terms in the bottleneck as maximization problem "$\max \mathrm{I}[Y;Z] - \beta \mathrm{I}[X;Z]$", and compute a variational lower bound on this objective, using a prior $r(z)$ on the distribution of latent representations, which is fixed to be a unit Gaussian distribution.

$$\min \mathbb{E}_{p_\theta(z|x_n)}[-\log q_\theta(y|z)] - \beta \mathrm{KL}(p_\theta(z|x_n) \| r(z)).$$

In principle, the distributions q_θ and p_θ could be given by arbitrary parameterizations and function approximators. In practice, the implementation of DVIB presented by Alemi et al. (2016) constructs p_θ as a multivariate Gaussian with parameterized mean and parameterized diagonal covariance using a neural network followed by a linear decoder feeding into a softmax to yield q_θ. The requirement for p_θ to have a closed-form Kullback-Leibler divergence with respect to the prior $r(z)$ limits the applicability of the DVIB objective.

"Conditional Entropy Bottleneck" (Fisher, 2019) introduces their Conditional Entropy Bottleneck as follows.

$$\min \mathrm{I}[X;Z \mid Y] - \mathrm{I}[Y;Z] \tag{4}$$

Fisher (2019) provides experimental results that favorably compare to Alemi et al. (2016), possibly due to additional flexibility as Fisher (2019) do not constrain $p_\theta(z)$ to be a unit Gaussian and employ variational approximations for all terms.

3. IB Objectives without Mutual Information Terms

Following Kirsch et al. (2020), we can rewrite the objectives to make use of entropies and conditional entropies instead of mutual information terms.

Observation 1. *For IB, we obtain*

$$\begin{aligned} &\arg\min \mathrm{I}[X;Z] - \beta \mathrm{I}[Y;Z] && (5)\\ &= \arg\min \mathrm{H}[Y \mid Z] + \beta' \underbrace{\mathrm{I}[X;Z \mid Y]}_{=\mathrm{H}[Z|Y]-\mathrm{H}[Z|X]}, && (6) \end{aligned}$$

and, for DIB,

$$\begin{aligned} &\arg\min \mathrm{H}[Z] - \beta \mathrm{I}[Y;Z] && (7)\\ &= \arg\min \mathrm{H}[Y \mid Z] + \beta' \mathrm{H}[Z \mid Y] && (8)\\ &= \arg\min \mathrm{H}[Y \mid Z] + \beta'' \mathrm{H}[Z] && (9) \end{aligned}$$

with $\beta' := \frac{1}{\beta - 1} \in [0, \infty)$ *and* $\beta'' := \frac{1}{\beta} \in [0, 1)$.

3.1. Entropy Estimation for Continuous Variables

One of the principal challenges in training with IB objectives is the computation of the mutual information quantities required. We have rewritten the IB objectives using differential (conditional) entropies. However, differential entropies have their own number of undesirable properties.

Most importantly, they are unbounded from below. This means that in principle a neural network could minimize

$\mathrm{H}[\mathrm{Z} \mid \mathrm{X}]$ in the DIB objective by scaling the latent representation Z to be arbitrarily close to zero, thus obtaining monotonically "improving" and unbounded objective values despite not meaningfully changing the representation.

One way to solve this problem is by adding noise to Z. Again, following Kirsch et al. (2020), we add noise to the latent representation in order to lower-bound entropies, which allows us to enforce non-negativity across all terms in our objective (as in the discrete case): for a continuous $\hat{\mathrm{Z}} \in \mathbb{R}^k$ and independent noise ϵ, we set $\mathrm{Z} := \hat{\mathrm{Z}} + \epsilon$; the differential entropy then satisfies $\mathrm{H}[\mathrm{Z}] = \mathrm{H}[\hat{\mathrm{Z}} + \epsilon] \geq \mathrm{H}[\epsilon]$; and by using *zero-entropy noise* $\epsilon \sim \mathcal{N}(0, \frac{1}{2\pi e} I_k)$ specifically, we obtain $\mathrm{H}[\mathrm{Z}] \geq \mathrm{H}[\epsilon] = 0$.

Observation 2. *After adding zero-entropy noise, the inequality* $\mathrm{I}[\mathrm{X}; \mathrm{Z} \mid \mathrm{Y}] \leq \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] \leq \mathrm{H}[\mathrm{Z}]$ *also holds in the continuous case, and we can minimize* $\mathrm{I}[\mathrm{X}; \mathrm{Z} \mid \mathrm{Y}]$ *in the IB objective by minimizing* $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$ *or* $\mathrm{H}[\mathrm{Z}]$*, similarly to the DIB objective. See section A.1.*

Strictly speaking, zero-entropy noise is not necessary for optimizing bounds: any Gaussian noise is sufficient, but zero-entropy noise is necessary to ensure non-negativity and for the inequalities.

4. Method

We are going to focus on deterministic models while injecting zero-entropy noise into the latent Z. We then have $\mathrm{H}[\mathrm{Z} \mid \mathrm{X}] = 0$ and $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] \leq \mathrm{H}[\mathrm{Z}]$, which means that the DIB objective and IB objective match. To see this, compare equation (6) and (8) after setting $\mathrm{H}[\mathrm{Z}] = 0$. We can thus focus on $\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta'' \mathrm{H}[\mathrm{Z}]$ as objective, which we can rewrite as

$$\min \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] + (\beta'' - 1)\mathrm{H}[\mathrm{Z}] \tag{10}$$

after dropping a constant $\mathrm{H}[\mathrm{Y}]$ term. This objective is more amenable to entropy estimation because $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] = \frac{1}{C} \sum_y \mathrm{H}[\mathrm{Z} \mid y]$ for categorical labels Y for which we can easily estimate $\mathrm{H}[\mathrm{Z} \mid y]$ individually.

Our method consists of training a deterministic neural network encoder (from input X to latent Z) by minimizing the objective without having to specify a decoder explicitly. We can then train a decoder separately after training the encoder.

Instead of fitting a DNN as decoder (which works of course), we can also fit a Gaussian mixture model to the latent Z: each Gaussian represents one output class, as optimizing IB objectives for deterministic models under noise leads to clustering, which we show in section A.2. We will show that we obtain competitive performance on CIFAR-10 with either kind of decoder.

4.1. A simple non-parametric entropy estimator

We devote this section to the entropy estimation component of our method. We use the entropy estimator as presented in Kraskov et al. (2004), which uses a k-nearest-neighbors estimate of the Shannon entropy (Kozachenko & Leonenko, 1987):

$$\hat{\mathrm{H}}[Z] = -\psi(k) - \psi(N) + \log c_d + \frac{d}{N} \sum_{i=1}^{N} \log \epsilon(i), \tag{11}$$

where ψ is the digamma function, c_d is the volume of the d-dimensional unit ball with respect to the norm used in the nearest-neighbors computation, and $\epsilon(i) = 2 \|z_i - n_k(z_i)\|$ is twice the distance from sample z_i to its k^{th} nearest neighbor. z_i are latent samples that can be obtained for the whole training set or using minibatches.

Differentiation is then straightforward:

$$\nabla \hat{\mathrm{H}}[\mathrm{Z}] = \frac{d}{N} \sum_{i=1}^{n} \nabla \left[\log \|x_i - n_k(x_i)\| \right]. \tag{12}$$

We can thus estimate our objective in (10) using the estimator in equation (11) and back-propagate through it.

4.2. Improving Stability of the Gradient

To improve stability of the gradients (under ℓ^2-norm), we use the squared distance:

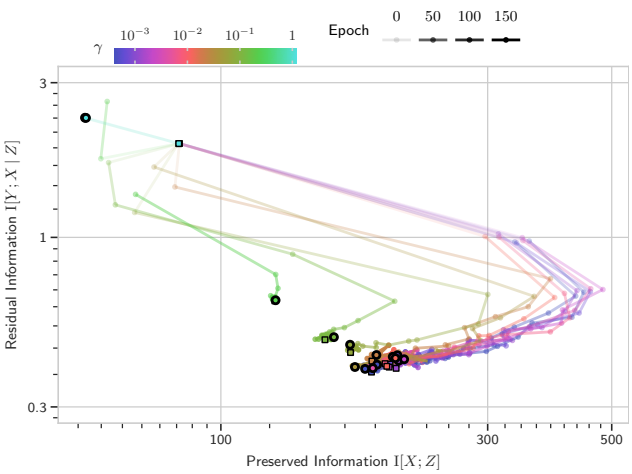
$$\hat{\mathrm{H}}[\mathrm{Z}] = -\psi(k) - \psi(N) + \log c_d + \frac{d}{2N} \sum_{i=1}^{N} \log \epsilon^2(i), \tag{13}$$

which leads to

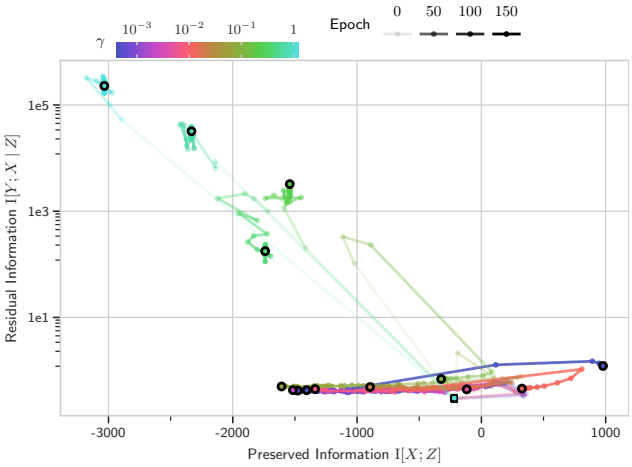
$$\nabla \hat{H}(X) = \frac{d}{2N} \sum_{i=1}^{n} \nabla \left[\log \|x_i - n_k(x_i)\|^2 \right]. \tag{14}$$

5. Empirical Evaluation

We use a regular deterministic ResNet18 model (He et al., 2016) as encoder with $K = 256$ continuous latent dimensions. As decoder, we either fit a Gaussian mixture model, for which we compute a covariance matrix based on sampled $Z|Y$ from the training set, or we use a neural network. We have both used a simple logistic regression model, but also a deep decoder that is inspired by ResNets but uses fully-connected layers instead of convolutions. Specifically, we use a layer that maps from the latent to 1024 units, then 5 residual blocks of BatchNorm $\times$ ReLU $\times$ 1024 FC linear $\times$ BatchNorm $\times$ ReLU $\times$ 1024 FC layer, and a final BatchNorm and fully-connected linear layer to the output classes (using softmax). Both perform similarly, yet we saw slightly better performance with the deeper model. The



(a) with injected zero-entropy noise



(b) without injected zero-entropy noise

Figure 2. Trajectories for the surrogate objective $\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta'' \mathrm{H}[\mathrm{Z}]$ *on the test set with a ResNet18 model on CIFAR-10.* The trajectories are colored by their respective γ; their transparency changes by epoch. We estimate $\mathrm{I}[\mathrm{X}; \mathrm{Z}]$ using equation (11) and estimate $\mathrm{I}[\mathrm{X}; \mathrm{Y} \mid \mathrm{Z}]$ by fitting on the test set to obtain a tight upper bound on $\mathrm{I}[\mathrm{X}; \mathrm{Y} \mid \mathrm{Z}] = \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$. Compression (Preserved Information ↓) trades-off with performance (Residual Information ↓). The circle marks the final epoch of a trajectory. The square marks the best epoch (Residual Information ↓↓).

deeper model provides better entropy estimates for visualizations, however, again following the approach in Kirsch et al. (2020). *We obtain an accuracy of 92% when training using our objective with either a GMM or neural network decoder*. See figure 3.

For our experiments, we use PyTorch (Paszke et al., 2019) and the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.5×10^{-3} and multiply the learning rate by $\sqrt{0.1}$ whenever the loss plateaus for more than 10 epochs. We train for 150 epochs.

Robustness to Adversarial Attacks The DVIB objective

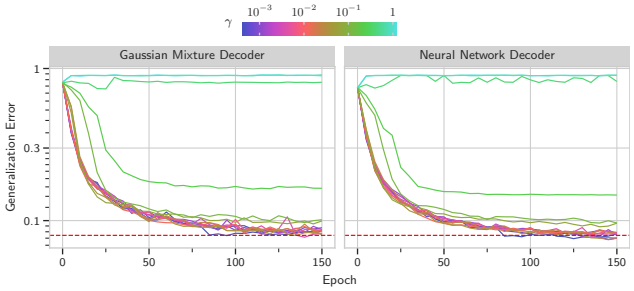


Figure 3. Generalization error for ResNet18 models trained with our objective $\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta'' \mathrm{H}[\mathrm{Z}]$ *for different* β''. Models are trained on CIFAR-10 without training through a decoder using our objective. Either a neural network decoder or a Gaussian mixture decoder are then trained using samples from the training set and then evaluated on the test set. A similar ResNet18 trained with regular cross-entropy using the same hyperparameters and training schedule also achieves an accuracy of 92% (generalization error of 8%, red dashed line).

from Alemi et al. (2016) leads to improved adversarial robustness. We perform a similar evaluation. In figure 1 we see that it performs favorably against adversarially perturbed images using the Fast Gradient Sign Method (Szegedy et al., 2013) for varying levels of the perturbation magnitude parameter ϵ. We use a ResNet18 model trained with regular cross-entropy with added DropConnect regularization (Wan et al., 2013) as comparison (black dashed line).

Information Plane Plots We also create information plane plots, see figure 2a. It shows the trade-off between compression and accuracy. However, the models for lower γ do not seem to converge within 150 epochs and all of them seem to keep compressing (minimizing $\mathrm{H}[\mathrm{Z}]$). For a certain value of $\mathrm{I}[\mathrm{X}; \mathrm{Y} \mid \mathrm{Z}]$, one can see that different γ regularize compression differently, especially at the beginning of training. For the first 50 epochs, the trajectories remain separated. We hypothesize that given the form of our objective in (10), the gradients for $\mathrm{H}[\mathrm{Z}]$ dominate over $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$, such that small changes in β'' have little effect.

Injecting noise is necessary. We train models without injecting noise and obtain plots as in figure 2b, which are quite chaotic.

We measure $\mathrm{I}[\mathrm{X}; \mathrm{Y} \mid \mathrm{Z}] = \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ (as $\mathrm{H}[\mathrm{Y} \mid \mathrm{X}] = 0$ for CIFAR-10) on the test set by fitting a ResFC model on the test set, while keeping the encoder frozen. The cross-entropy loss of the decoder is an upper bound on $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$. This is similar to approaches in (Alemi et al., 2016; Xu et al., 2020; McAllester & Stratos, 2018). To estimate $\mathrm{I}[\mathrm{X}; \mathrm{Z}] = \mathrm{H}[\mathrm{Z}]$ (as $\mathrm{H}[\mathrm{Z} \mid \mathrm{X}] = 0$ for our deterministic model with injected zero-entropy noise), we use the estimator from equation (11) that we also use to train the decoder.

References

Achille, A. and Soatto, S. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Amjad, R. A. and Geiger, B. C. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International Conference on Machine Learning*, pp. 531–540, 2018.

Bercher, J.-F. and Vignat, C. A renyi entropy convolution inequality with application. In *2002 11th European Signal Processing Conference*, pp. 1–4. IEEE, 2002.

Fisher, I. The Conditional Entropy Bottleneck. *Submission to ICLR 2019, International Conference on Learning Representations*, 2019.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

Kirsch, A., Lyle, C., and Gal, Y. Unpacking information bottlenecks: Unifying information-theoretic objectives in deep learning, 2020.

Kozachenko, L. F. and Leonenko, N. N. A statistical estimate for the entropy of a random vector. 1987.

Kraskov, A., Stögbauer, H., and Grassberger, P. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

McAllester, D. and Stratos, K. Formal limitations on the measurement of mutual information. *arXiv preprint arXiv:1811.04251*, 2018.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

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

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Strouse, D. and Schwab, D. J. The deterministic information bottleneck. *Neural computation*, 29(6):1611–1630, 2017.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

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

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S., and Lucic, M. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pp. 1058–1066, 2013.

Xu, Y., Zhao, S., Song, J., Stewart, R., and Ermon, S. A theory of usable information under computational constraints, 2020.

Zhang, Y., Xiang, T., Hospedales, T. M., and Lu, H. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4320–4328, 2018.

A. Appendix

The proofs here are taken from Kirsch et al. (2020), which is a prepublication and thus not well-known.

A.1. Differential entropies

Observation. *After adding zero-entropy noise, the inequality* $\mathrm{I}[X;Z \mid Y] \leq \mathrm{H}[Z \mid Y] \leq \mathrm{H}[Z]$ *also holds in the continuous case, and we can minimize* $\mathrm{I}[X;Z \mid Y]$ *in the IB objective by minimizing* $\mathrm{H}[Z \mid Y]$ *or* $\mathrm{H}[Z]$*, similarly to the DIB objective.*

Theorem 1. *For random variables A, B, we have*

$$\mathrm{H}[A + B] \geq \mathrm{H}[B].$$

Proof. See Bercher & Vignat (2002, section 2.2). □

Proposition 1. *Let Y, Z and X be random variables satisfying the independence property* $Z \perp Y|X$*, and F a possibly stochastic function such that* $Z = F(X) + \epsilon$*, with independent noise* ϵ *satisfying* $\epsilon \perp F(X), \epsilon \perp Y$ *and* $H(\epsilon) = 0$*. Then the following holds whenever* $\mathrm{I}[Y;Z]$ *is well-defined.*

$$\mathrm{I}[X;Z \mid Y] \leq \mathrm{H}[Z \mid Y] \leq \mathrm{H}[Z].$$

Proof. First, we note that $\mathrm{H}[Z \mid X] = \mathrm{H}[F(X) + \epsilon \mid X] \geq \mathrm{H}[\epsilon \mid X] = \mathrm{H}[\epsilon]$ with theorem 1, as ϵ is independent of X, and thus $\mathrm{H}[Z \mid X] \geq 0$. We have $\mathrm{H}[Z \mid X] = \mathrm{H}[Z \mid X, Y]$ by the conditional independence assumption, and by the non-negativity of mutual information, $\mathrm{I}[Y;Z] \geq 0$. Then:

$$\mathrm{I}[X;Z \mid Y] + \underbrace{\mathrm{H}[Z \mid X]}_{\geq 0} = \mathrm{H}[Z \mid Y]$$
$$\mathrm{H}[Z \mid Y] + \underbrace{\mathrm{I}[Y;Z]}_{\geq 0} = \mathrm{H}[Z]$$

□

The probabilistic model from section **??** fulfills the conditions exactly, and the two statements motivate our observation.

It is important to note that while zero-entropy noise is necessary for preserving inequalities like $\mathrm{I}[X;Z \mid Y] \leq \mathrm{H}[Z \mid Y] \leq \mathrm{H}[Z]$ in the continuous case, any Gaussian noise will suffice for optimization purposes: we optimize via pushing down an upper bound, and constant offsets will not affect this.

Thus, if we had $\mathrm{H}[\epsilon] \neq 0$, even though $\mathrm{I}[X;Z \mid Y] + \mathrm{H}[Z \mid X] \not\leq \mathrm{H}[Z \mid Y]$, we could instead use

$$\mathrm{I}[X;Z \mid Y] + \mathrm{H}[Z \mid X] - \mathrm{H}[\epsilon] \leq \mathrm{H}[Z \mid Y] - \mathrm{H}[\epsilon]$$

as upper bound to minimize. The gradients remain the same.

This also points to the nature of differential entropies as lacking a proper point of origin by themselves. We choose one by fixing $\mathrm{H}[\epsilon]$. Just like other literature usually only considers mutual information as meaningful, we consider $\mathrm{H}[Z \mid X] - \mathrm{H}[\epsilon]$ as more meaningful than $\mathrm{H}[Z \mid X]$. However, we can side-step this discussion conveniently by picking a canonical noise as point of origin in the form of zero-entropy noise $\mathrm{H}[\epsilon] = 0$.

A.2. Soft clustering by entropy Minimization with Gaussian noise

This section motivates why we can fit a GMM, with one Gaussian per class, to our latent space and obtain high accuracy: minimizing with our objective (10) leads to clustering of the latent encodings.

Consider the problem of minimizing $\mathrm{H}[Z \mid Y]$ and $\mathrm{H}[Y \mid Z]$, in the setting where $Z = f_\theta(X) + \epsilon \sim \mathcal{N}(0, \sigma^2)$—i.e. the embedding Z is obtained by adding Gaussian noise of fixed σ to a deterministic function of the input. Let the training set be enumerated $x_1, \ldots, x_n$, with $\mu_i = f_\theta(x_i)$. Then the distribution of Z is given by a mixture of Gaussians with the following density, where $d(x, \mu_i) := \|x - \mu_i\|^2 / \sigma^2$.

$$\mathrm{p}(z) \propto \frac{1}{n} \sum_{i=1}^{n} \exp(-d(z, \mu_i))$$

Assuming that each x_i has a deterministic label y_i, we then find that the conditional distributions $\mathrm{p}(y \mid z)$ and $\mathrm{p}(z \mid y)$ are given as follows:

$$\mathrm{p}(z \mid y) \propto \frac{1}{n_y} \sum_{i:y_i=y} \exp(-d(z, \mu_i))$$
$$\mathrm{p}(y \mid z) = \sum_{i:y_i=y} \mathrm{p}(\mu_i \mid z) = \sum_{i:y_i=y} \frac{\mathrm{p}(z \mid \mu_i)\,\mathrm{p}(\mu_i)}{\mathrm{p}(z)}$$
$$= \frac{\sum_{i:y_i=y} \mathrm{p}(z \mid \mu_i)}{\sum_{k=1}^{n} \mathrm{p}(z \mid \mu_k)} = \frac{\sum_{i:y_i=y} \exp(-d(z, \mu_i))}{\sum_{k=1}^{n} \exp(-d(z, \mu_k))},$$

where n_y is the number of x_i with class $y_i = y$. Thus, the conditional $Z|Y$ can be interpreted as a mixture of Gaussians and $Y|Z$ as a Softmax marginal with respect to the distances between Z and the mean embeddings. We observe that $\mathrm{H}[Z \mid Y]$ is lower-bounded by the entropy of the random noise added to the embeddings:

$$\mathrm{H}[Z \mid Y] = \mathrm{H}[f_\theta(X) + \epsilon \mid Y] \geq \mathrm{H}[\epsilon]$$

with equality when the distribution of $f_\theta(X)|Y$ is deterministic – that is when f_θ is constant for each class y.

Further, the entropy $\mathrm{H}[Y \mid Z]$ is minimized when $\mathrm{H}[Z]$ is large compared to $\mathrm{H}[Z \mid Y]$ as we have the decomposition

$$\mathrm{H}[Y \mid Z] = \mathrm{H}[Z \mid Y] - \mathrm{H}[Z] + \mathrm{H}[Y].$$

In particular, when f_θ is constant over equivalence classes of the input, then $\mathrm{H}[Y \mid Z]$ is minimized when the entropy

$H[f_\theta(X) + \epsilon]$ is large – i.e. the values of $f_\theta(x_i)$ for each equivalence class are distant from each other and there is minimal overlap between the clusters. Therefore, the optima of the information bottleneck objective under Gaussian noise share similar properties to the optima of geometric clustering of the inputs according to their output class.

To gain a better understanding of local optimization behavior, we decompose the objective terms as follows:

$$\begin{aligned}
\mathrm{H[Z \mid Y]} &= \mathbb{E}_{\hat{\mathrm{p}}(y)}\, \mathrm{H}(\mathrm{p}(z \mid y) \,\|\, \mathrm{p}(z \mid y)) \\
&= \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, \mathrm{H}(\mathrm{p}(z \mid x) \,\|\, \mathrm{p}(z \mid y)) \\
&= \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, D_{\mathrm{KL}}(\mathrm{p}(z \mid x) \,\|\, \mathrm{p}(z \mid y)) + \mathrm{H}[Z \mid x] \\
&= \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, D_{\mathrm{KL}}(\mathrm{p}(z \mid x) \,\|\, \mathrm{p}(z \mid y)) \\
&\qquad + \underbrace{\mathrm{H[Z \mid X]}}_{=const}.
\end{aligned}$$

To examine how the mean embedding μ_k of a single datapoint x_k affects this entropy term, we look at the derivative of this expression with respect to $\mu_k = f_\theta(x_k)$. We obtain:

$$\begin{aligned}
\frac{d}{d\mu_k}\mathrm{H[Z \mid Y]} &= \frac{d}{d\mu_k}\, \mathrm{H}[Z \mid y_k] \\
&= \frac{d}{d\mu_k}\, \mathbb{E}_{\mathrm{p}(x|y_k)}\, D_{\mathrm{KL}}(\mathrm{p}(z \mid x) \,\|\, \mathrm{p}(z \mid y)) \\
&= \sum_{i \neq i : y_i = y_k} \frac{1}{n_{y_k}} \frac{d}{d\mu_k} D_{\mathrm{KL}}(\mathrm{p}(z \mid x_i) \,\|\, \mathrm{p}(z \mid y_k)) \\
&\qquad + \frac{1}{n_{y_k}} \frac{d}{d\mu_k} D_{\mathrm{KL}}(\mathrm{p}(z \mid x_k) \,\|\, \mathrm{p}(z \mid y_k)).
\end{aligned}$$

While these derivatives do not have a simple analytic form, we can use known properties of the KL divergence to develop an intuition on how the gradient will behave. We observe that in the left-hand sum μ_k only affects the distribution of $Z|Y$ (that is we are differentiating a sum of terms that look like a reverse KL), whereas it has greater influence on $\mathrm{p}(z \mid x_k)$ in the right-hand term, and so its gradient will more closely resemble that of the forward KL. The left-hand-side term will therefore push μ_k towards the centroid of the means of inputs mapping to y, whereas the right-hand side term is mode-seeking.