Direct Validation of the Information Bottleneck Principle for Deep Nets – Supplementary Material

Adar Elad*, Yochai Blau, Tomer Michaeli
Technion - Israel Institute Of Technology

Doron Haviv*
Weill Cornell Medical College

A. Training algorithm for the AA-MINE

Algorithm 1: Estimating noise-regularized MI $I(X, L_i + \varepsilon)$

b - batch size ; σ - noise std ; μ - learning rate

Input: X - input distribution
F_i - network up to layer i
Output: $I(X, L_i + \varepsilon)$

$\theta_0 \leftarrow$ Initialize the discriminator;
for $k = 1$ *to* N **do**
1. Draw b samples of X: $X^J = \{ x_J^{(1)}, x_J^{(2)}, ..., x_J^{(b)} \}$;
2. Draw b samples of X: $X^M = \{ x_M^{(1)}, x_M^{(2)}, ..., x_M^{(b)} \}$;
3. Feed X^J and X^M through the network (F_i), getting: $L_i^J = \{ l_J^{(1)}, l_J^{(2)}, ..., l_J^{(b)} \}$, $L_i^J = \{ l_M^{(1)}, l_M^{(2)}, ..., l_M^{(b)} \}$;
4. Generate noise samples $\varepsilon_J, \varepsilon_M \sim \mathcal{N}(0, \sigma^2 I)$;
5. Evaluate: $\mathrm{I}(\theta_k) = \frac{1}{b} \sum_{i=1}^{b} D_{\theta_{k-1}}(x_J^{(i)}, l_J^{(i)} + \varepsilon_J) - \log(\frac{1}{b} \sum_{i=1}^{b} e^{D_{\theta_{k-1}}(x_J^{(i)}, l_M^{(i)} + \varepsilon_M)})$;
6. $\theta_k \leftarrow \theta_{k-1} + \mu \cdot \nabla \mathrm{I}(\theta_k)$;

end

B. Related work

B.1. Deep neural nets and the information bottleneck

The Information Bottleneck (IB) technique for summarizing a random variable X while maintaining maximal mutual information with a desirable output Y, was first introduced by Tisbhy et al. [18]. It is designed to find the optimal trade-off between prediction accuracy and representation complexity, by compressing X while retaining the essential information for predicting Y. The IB optimization problem can be directly solved in discrete settings, as well as in certain simple families of continuous distributions, like jointly Gaussian random variables [8]. However, in general high-dimensional continuous scenarios, exact optimization becomes impractical due to the intractable continuous MI.

The IB principle was initially linked to DNNs by Tishby and Zaslavsky [19], which hypothesized that DNN layers optimally converge to the IB curve. The subsequent work by Shwartz-Ziv and Tishby [17] experimentally analyzed the effect of training with *a cross-entropy loss* on the IB functional of DNN layers. In this work, the MI estimation was made tractable by binning (discretization) of the latent representations L_i, which works for "toy" examples but does not scale to real-world scenarios. Among several claims, the authors report that two training phases emerge: a "fitting" phase followed by a "compression" phase, which to their understanding is linked to increased generalization. Yet Saxe et al. [16] contradict these claims, by linking the compression phase to the activation type and discretization strategy, and question the connection between this compression phase and generalization. Moreover, these authors as well as Amjad and Geiger [4], also recognize that the term $I(X; L)$ in the IB functional is theoretically infinite for deterministic DNNs with a continuous input X, and thus the attempt to measure it is meaningless. Both works propose to remedy this by adding noise, which ensures this term is finite.

Despite these obstacles, DNNs were in fact trained on real high-dimensional data for classification tasks with the IB functional. The difficulties arising from the intractable and infinite term $I(X; L)$ were overcome by (i) training *stochastic* DNNs which ensure it is finite, and (ii) using a variational approximation of the MI which makes it tractable [3, 13, 7, 2, 5, 6]. These schemes all rely on some form of injected stochasticity, and in fact, most enforce the IB objective only on a *single* "bottleneck layer". An attempt to optimize the IB functional for deterministic DNNs in a layer-wise fashion, as in the original theory, has yet to appear due to these practical difficulties. Here, we report on

*Equal contribution.

such a training scheme and present its results in the following sections.

We note that layer-wise training with a related information-theoretic objective has been studied in [20] and [10], which have shown an effective and practical method to compose and analyze hierarchical representations. These works however consider an *unsupervised* setting, whereas we analyze supervised DNN training (specifically) with the IB functional.

B.2. Estimating mutual information

Quantifying the MI between distributions is inherently difficult [15], and is tractable only in discrete settings or for a limited family of problems. In other more general settings the exact computation is impossible, and known approximations do not scale well with dimension and sample size [11]. Recently, Belghazi et al. [5] proposed the MI Neural Estimator (MINE) for approximating the MI between continuous high-dimensional random variables via back-prop over a DNN. The core idea is to estimate the Kullback-Leibler (KL) divergence that is used to define MI, through the maximization of the dual representation of Donsker and Varadhan [9].

Minimizing the MI between the input X and hidden layer L_i of a DNN using MINE can be efficiently accomplished, by formulating a min-max objective between the examined DNN and the neural MI estimator (an auxiliary net), similarly to adversarial training (see details in [5]). We adopt this strategy and improve it in order to enforce the IB objective on DNN layers during training. Note that the original authors also demonstrate the IB principle with this estimator, however they only do so on a single "bottleneck layer", by using the cross-entropy loss as an approximation for the MI with the desired output Y, and in an end-to-end manner (as described above).

C. Information plane dynamics

It is of interest to examine the dynamics of the layers in the information plane defined by axes $I(X;L_i), I(Y;L_i)$, similarly to [17, 16]. Figure 1 depicts the information plane dynamics when training layer-by-layer with the IB objective in the MNIST experiment in Sec. 4. Interestingly, we see a clear two-phase process. Namely, each layer starts by moving upwards in the information plane to increase its MI with the desired output Y, and then turns to compress its latent representation by decreasing its MI with X, i.e. by moving leftwards. This two-phase training dynamics was previously observed in [17], yet we observe dynamics which are slightly different. Specifically, the MI with the input does not increase at any stage during training, contrary to their observations in the first "fitting" phase.

To compare between layer-wise training with IB functional and the common training scheme, we additionally

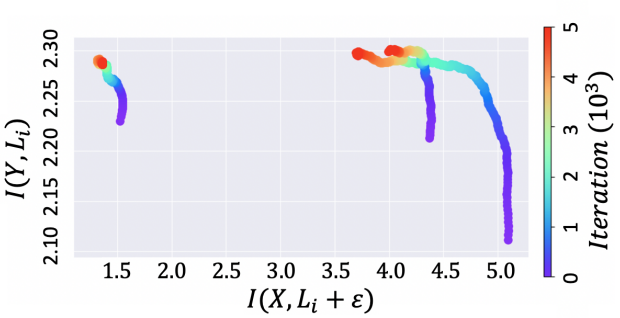

Figure 1. Information plane dynamics when training layer-by-layer explicitly with the IB functional. In this experiment, a 3-layer MLP with ReLU activations was trained to classify MNIST digits. The curves plot the change in the mutual information with the input/target for each of the layers, as training progresses. The MI was measured with an AA/standard MINE (see Section 3). Two-phase training dynamics are apparent, and we observe a succeeding "compression" phase. Note that the data-processing inequality does not apply when estimating the *noise-regularized* MI, so that $I(X, L_i + \varepsilon)$ need not necessarily be larger than $I(X, L_j + \varepsilon)$ for $j > i$ [16, App. C].

plot the information plane dynamics for the same experiment when training in an end-to-end fashion with the cross-entropy loss in the top row of Fig. 2 (see details in Appendix E). As can be seen, both terms tend to increase throughout the optimization process, and there is no apparent compression phase in which the MI with the input suddenly begins to decrease. This phenomenon has also been observed in the work of Saxe et al. [16], and is in contradiction with the observations of Shwartz-Ziv & Tishby [17]. Saxe et al. associate the compression phase apparent in [17] to the combination of the binning (discretization) strategy used in [17] to quantify the MI and the saturating tanh activations. This leads to their conclusion that the two-phase dynamics with a compression phase is not a general property of end-to-end DNN training with the CE loss. However, interestingly, when we *add weight decay* as a regularizer, the two-phase dynamics emerge (see bottom row of Fig. 2). As discussed in Sec. 2, weight decay induces a penalty on the MI with the input, and our experiments suggest that this penalty induces a compression phase which begins only after the MI with the output has reached high values. As weight decay is known to increase generalization [14], it seems that the compression phase is indeed linked to enhanced generalization.

D. Training details for the MNIST and CIFAR-10 experiments in Section 4

In the MNIST experiments, we trained a three-layer multi-layer perceptron (MLP) with dimensions $784-512-512-10$ and ReLU activations. In the CIFAR-10 exper-

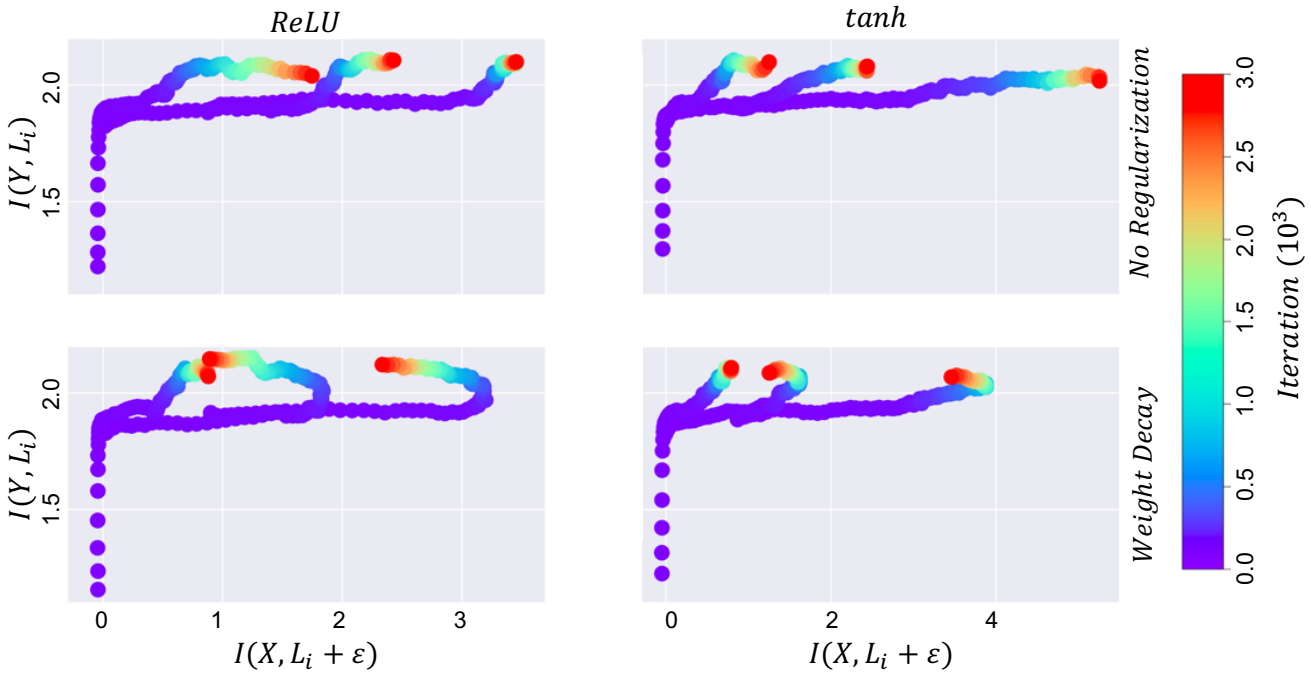

Figure 2. Information plane dynamics when training end-to-end (all layers jointly) with the common cross-entropy loss and using ReLU/tanh activations within the DNN. The curves plot the change in the mutual information with the input/target for each of the layers, as training progresses. Two-phase training dynamics *does not* emerge when training with the conventional cross-entropy loss, for both ReLU and tanh activations. Yet, when weight decay regularization is added a compression phase emerges, i.e. the MI with the input begins to decrease after the MI with the target reaches high values (similar to layer-by-layer training with the IB functional, see Fig. 1). This indicates that two-phase dynamics, and specifically the emergence of a compression phase, is induced by regularization.

iments, we train a net with three conv-layers (16 filters, ReLU activations, max-pooling after each), which are followed by three fully-connected layers (512-512-10, ReLU activations). The architecture of the MINE discriminator was an MLP with two hidden layers of 1500 neurons and Leaky-ReLU activations, and a final linear layer to a single neuron. The input was always taken to be a concatenation of the two variables between which MI is measured. For AA-MINE, we used the same architecture, only after passing the first input through a copy of layers $1 \dots i$ of the primary DNN. Therefore, this architecture was always applied to $F_i(X)$ and $F_i(X) + \epsilon$ in the case of AA-MINE.

In these experiments, we started by training both MINE discriminators D_x, D_Y separately until convergence. Then, layer-by-layer training was performed with a total of 5000 iterations for each layer. In each of these iterations, we alternate between 1 step for updating the trained layer, and 10 steps for updating each of the MINE discriminators. For both the MNIST and CIFAR-10 experiments, the variance of the Gaussian noise ε was $\sigma_\varepsilon^2 = 2$, and the learning rate was 10^{-4} for the MINE discriminators and 10^{-3} for the the DNN, respectively. In the MNIST experiment, the bottleneck parameter was $\beta = 10^{-3}$ for all the layers. In the CIFAR-10 experiment, the bottleneck parameter was $\beta = 0$ for the convolutional layers and $\beta = 10^{-3}$ for the fully-connected layers.

In all experiments, training was performed with the Adam optimizer [12] with the TensorFlow package [1].

E. Training details for the information-plane dynamics experiments in Appendix C.

In this experiment, each network was trained for a total of $3{\cdot}10^3$ training steps using the ADAM optimizer [12] with a learning rate of 10^{-3} and a batch size of 128. The weight-decay regularization factor was 10^{-3}. Mutual information with the input/target was quantified with the MI neural estimator (see Section 3). The discriminators' D_x, D_y architecture was as in Appendix D, and were trained for $2 \cdot 10^2$ training steps at each of the sampled iterations.

References

[1] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[2] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through

noisy computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[3] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2018.

[4] 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.

[5] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Devon Hjelm, and Aaron Courville. Mutual information neural estimation. In *International Conference on Machine Learning (ICML)*, pages 530–539, 2018.

[6] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning (ICML)*, pages 675–685, 2019.

[7] Matthew Chalk, Olivier Marre, and Gasper Tkacik. Relevant sparse codes with variational information bottleneck. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1957–1965, 2016.

[8] Gal Chechik, Amir Globerson, Naftali Tishby, and Yair Weiss. Information bottleneck for gaussian variables. *Journal of machine learning research (JMLR)*, 6(Jan):165–188, 2005.

[9] Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

[10] Shuyang Gao, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. Auto-encoding total correlation explanation. *arXiv preprint arXiv:1802.05822*, 2018.

[11] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *Artificial Intelligence and Statistics*, pages 277–286, 2015.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, 2015.

[13] Artemy Kolchinsky, Brendan D Tracey, and David H Wolpert. Nonlinear information bottleneck. *arXiv preprint arXiv:1705.02436*, 2017.

[14] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems (NIPS)*, pages 950–957, 1992.

[15] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003.

[16] Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations (ICLR)*, 2018.

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

[18] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. In *The 37th annual Allerton Conference on Communication, Control, and Computing*, pages 268–377, 1999.

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

[20] Greg Ver Steeg and Aram Galstyan. Maximally informative hierarchical representations of high-dimensional data. In *Artificial Intelligence and Statistics*, pages 1004–1012, 2015.