On the emergence of invariance and disentangling in deep representations

Alessandro Achille & Stefano Soatto
Department of Computer Science
University of California, Los Angeles
405 Hilgard Ave., Los Angeles, 90095, CA, USA
{achille,soatto}@cs.ucla.edu

Technical Report UCLA UCSD 170010

May 30, 2017

Abstract

Using classical notions of statistical decision and information theory, we show that invariance in a deep neural network is equivalent to minimality of the representation it computes, and can be achieved by stacking layers and injecting noise in the computation, under realistic and empirically validated assumptions. We use an Information Decomposition of the empirical loss to show that overfitting can be reduced by limiting the information content stored in the weights. We then present a sharp inequality that relates the information content in the weights – which are a representation of the training set and inferred by generic optimization agnostic of invariance and disentanglement – and the minimality and total correlation of the activation functions, which are a representation of the test datum. This allows us to tackle recent puzzles concerning the generalization properties of deep networks and their relation to the geometry of the optimization residual.

Keywords: Deep learning; neural network; representation; flat minima; information bottleneck; overfitting; generalization; sufficiency; minimality; sensitivity; information complexity; stochastic gradient descent; regularization; total correlation.

1 Introduction

Efforts to understand the reasons for the empirical success of deep learning have followed two main lines: Representation learning and optimization.

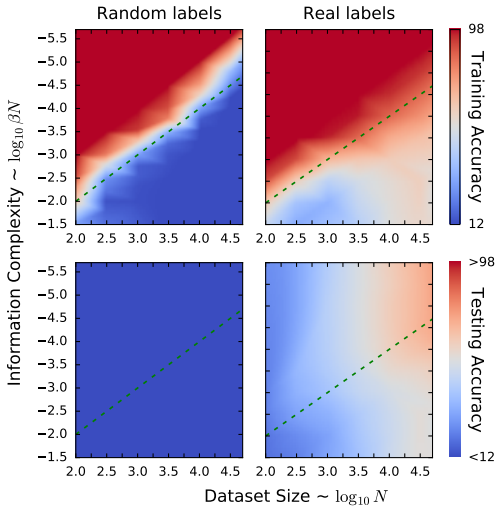

Figure 1: **(Left)** The AlexNet model of [36] achieves high accuracy (red) even when trained with random labels on CIFAR-10. Using the IB Lagrangian to limit information in the weights leads to a sharp transition to underfitting (blue) predicted by the theory (dashed line). To overfit, the network needs to memorize the dataset, which grows linearly. **(Right)** For real labels, however, the information sufficient to fit the data without overfitting saturates to a value that depends on the dataset, but somewhat independent of the number of samples. Each point in the plots is a deep network trained with a different regularizer β and dataset size N (details in Appendix A.1). What is displayed is the loss (training error). Test accuracy produces a uniform blue plot for random labels, while for real labels it increases with the number of training samples, and is higher near the critical regularizer value $\beta = 1$.

In the latter, a deep network is treated as a black-box family of functions for which we want to find parameters (*weights*) that yield good generalization. Aside from the difficulties due to high-dimensionality and non-convexity of the loss function, the fact that deep networks are heavily over-parametrized presents theoretical challenges: One would expect, from the bias-variance tradeoff, that they would easily overfit; yet, simple optimization algorithms like stochastic gradient descent (SGD), even without explicit regularization, perform surprisingly well in practice. Recent work suggests that properties of the loss landscape and implicit regularization performed by SGD play an important role, but the overall theoretical picture is still hazy [36].

Representation learning, on the other hand, focuses on the properties of the representation of the data learned by the layers of the network, while remaining agnostic to the particular optimization process used. The effectiveness of deep learning is often ascribed to the ability of deep networks to learn representations that are insensitive (invariant) to nuisance variability in the data, such as affine transformations or occlusions in images, and also "disentangled," separating factors in the high-dimensional space of data. Clearly, careful engineering of the architecture, *e.g.,* using convolutional and pooling layers, plays an important role in achieving insensitivity to simple nuisance transformations. However, more complex and dataset-specific nuisances still need to be learned. This poses a riddle: If neither the architecture nor the loss function explicitly enforce invariance and disentangling, how can these properties emerge consistently in deep networks trained by simple generic optimization?

In this work, we aim to address these questions by establishing some strong information-theoretic connections between the optimization process, the loss landscape, and the properties of the learned representation. In particular, we show that: (a) a sufficient representation of the data is invariant if and only if it is minimal, meaning that it contain the lowest possible amount of information; (b) the quantity of information in the representation, along with its total correlation (a measure of disentanglement) are tightly bounded by the quantity of information that the weights of the network retain about the dataset; (c) the information in the weights, which is related to generalization [18] and flat minima [19], can be controlled by implicit or explicit regularization. Moreover, we show that adding noise during the training of the network is a simple and natural way of biasing the network towards invariant representations.

At the core of these results is an information-theoretic duality that we establish between the weights, which can be considered a *representation of the training dataset* agnostic to invariance and disentanglement, and the "activations" of the network,[1] which are a *representation of the test datum (input)* ideally sufficient for the task, invariant to nuisances, and disentangled. We derive bounds that show that simply learning weights that minimize information from the training set yields activations that are maximally sufficient, invariant and disentangled representations of the test datum.

Information-theoretic interpretations underlie several successful techniques, such as variational auto-encoders (VAE) [21] and generative adversarial networks (GAN) [15]. An information-theoretic framework for representations has been advocated by [33] leading to information-theoretic regularization [1, 2]. Here, we extend the study to the weights of the network, similarly to what [18, 19] already suggested nearly a quarter of century ago, and show how information in the weights and information in the representation are surprisingly strongly connected.

Finally, we perform several experiments on realistic architectures and datasets to validate the assumptions underlying our claims. In particular, we show that using the information in the weights to measure the complexity of a deep neural network (DNN), rather than the number of its parameters, leads to a sharp and theoretically predicted transition between overfitting and underfitting regimes for random labels, shedding some light on the questions of [36]. Moreover, we can recover a form of bias-variance tradeoff for deep networks. We also adapt techniques from the GAN literature to measure and display various information-theoretic identities.

1.1 Related work

In this work we establish connections between several topics in optimization and representation learning.

Regularization. Regularization is practiced both explicitly and implicitly when training deep networks. For instance, the L_2 norm of the weights is sometimes added to the empirical cross-entropy loss, a practice knowns as *weight decay*, while stochastic gradient descent (SGD) as well as Dropout [30] are known to have

[1] The nomenclature for deep networks is made precise in Sect. 2.1.

regularizing effects. However, their role on generalization is still poorly understood, as networks can overfit random labels even if they generalize on real data [36]. In [18], limiting the quantity of information that the weights retain about the dataset was suggested as a criterion to prevent overfitting. However, such a criterion has not been adopted widely, in part due to the difficulties in optimizing it, although recent advances in stochastic gradient variational Bayes have now made it possible [20]. There are multiple reasons why the information in the weights, which we call *information regularizer*, should be considered a good regularizer, not last that it leads to a clean theoretical analysis and yields a complete characterization of the behavior of a network learning random labels.

Information Bottleneck. The Information Bottleneck was introduced as a generalization of the notion of minimal sufficient statistic [32] that allows trading off fidelity (sufficiency) and complexity. In particular, the introduction of the Information Bottleneck (IB) Lagrangian reduces finding a minimal sufficient representation of the data to a variational optimization problem. Interestingly, the IB Lagrangian can also be seen as the empirical cross-entropy loss commonly used in deep learning, plus an information regularizer. A network minimizing this regularized cross-entropy loss thus recovers weights that are a minimal representation of the training data for the inference task, giving an alternative and unexplored derivation for the information regularizer of [18].

An orthogonal direction applies the IB Lagrangian to representation learning [33, 25] and suggests that in order to solve a given task, each layer of the network needs to be a representation of the original data that is increasingly minimal, meaning that it discards as much useless variability in the data as possible, while retaining and exposing all information relative to the task. This properties can be forced in real networks [1, 2] through the use of suitable information regularizers.

While applying an Information Bottleneck to the weights and the activations are two conceptually different problems, the particular structure of deep network unexpectedly ties them together. A consequence of this connection is that the quantity of information in the weights tightly controls the quantity of information in the activations. This implies that simply minimizing the information in the weights makes the representation learned in the final layer *minimal*.

Minimality and invariance. In turn, minimality of the representation is related (in fact, equivalent) to invariance. Representations learned by deep networks are observed to be insensitive to complex nuisance transformations of the data. To a certain extent, this can be attributed to the architecture. For instance, the use of convolutional layers and max-pooling can be shown to yield insensitivity to local group transformations [7, 3]. But for more complex, dataset-specific, and in particular non-local, non-group transformations, such insensitivity is acquired as part of the learning process, rather than being coded in the architecture. We show that a sufficient representation is maximally insensitive to nuisances if and only if it is minimal, allowing us to prove that a regularized network is naturally biased toward learning invariant representations of the data.

Flat minima. Our framework suggests that invariance properties arise in networks regularized by minimizing information in the weights. This may appear in contrast with common practice, which does not explicitly regularize the information. However, there is ample empirical evidence that SGD tends to converge towards "flat minima" in the loss function, and such minima are associated with good generalization [19]. We show that flat minima contain low information under our modeling assumptions, so even if explicit regularization is not performed, SGD implicitly limit the information in the weights when it converges to flat minima, fitting with our theory. The reverse (sharp minima having high information), is not necessarily the case, so there is no contradiction with [11], who notice that sharp minima can generalize too. Our model suggests that what determines overfitting is thus not flatness alone, but information in the weights which, unlike the eigenspectrum of the Hessian, is invariant to rescaling. In conjunction with the other properties we prove, this also suggests that flat minima have better invariance properties.

Alternate theoretical frameworks Efforts to develop a theoretical framework for representation learning include [32, 33], which are the closest in spirit to our approach and consider representations as stochastic functions that approximate minimal sufficient statistics, different from [7] who construct representations as (deterministic) operators that are invertible in the limit, while exhibiting reduced sensitivity ("stability") to small perturbations of the data. Some of the deterministic constructions are based on the assumption

that the underlying data is spatially stationary, and therefore work best on textures and other visual data that are not subject to occlusions and scaling nuisances. [3] develop a theory of invariance to locally compact groups, and aim to construct maximal ("distinctive") invariants, like [31] that, however, assume nuisances to be infinite-dimensional groups [16]. These efforts are limited by the assumption that nuisances have a group structure. Again, scaling/quantization and occlusion are not invertible, and therefore theories that ignore them do not capture critical aspects of the phenomenology of image formation. Other theoretical efforts focus on complexity considerations, and explain the success of deep networks by ways of statistical or computational efficiency [23, 5, 22].

"Disentanglement" is an often-cited property of deep networks [5], but seldom formalized and studied analytically, although [34] has suggested a powerful way of studying and optimizing for it using the Total Correlation of the representation.

Visual representations The study of invariant representations was central to Computer Vision for decades until it waned in the nineties following wide misinterpretation of results claiming the non-existence of non-trivial viewpoint invariants. Instead, viewpoint and contrast invariants that are not only non-trivial, but *maximal*, were shown to exist in [31]. The fact that such invariants are supported on a thin set means that the Actionable Information (the complexity of the invariant [26]) is far smaller than that of the data, enabling lossless symbolization. While maximal invariance is relevant for reconstruction, it is overkill for most other tasks, and the restriction to group nuisances, even if infinite-dimensional, overly limiting because of occlusion and scaling/quantization. In [28], following a position document [27], the notion was put forth that *sufficient invariance* was instead the key defining characteristic of a representation, leading to a connection with the Information Bottleneck. At about the same time, interest in invariance was returning [24, 7], but was still focused on maximal invariance (also called *distinctiveness* or *selectivity* [14]). The focus on sufficient invariance allowed to transform the tradeoff between invariance and selectivity (both invariance and sufficiency can be achieved in theory [4]) to a tradeoff is between complexity and approximation of a minimal sufficient invariant. In the meantime, cross-entropy was emerging as the interpretation of choice for the loss function used to train deep networks for visual recognition. Cross entropy is minimized when the network approximates the marginal log-likelihood, which is a sufficient invariant [28]. While the theory was finally beginning to align with the practice without the restriction to nuisances exhibiting a group structure, available frameworks could not quantify these tradeoffs or precisely predict observed phenomena.

Sufficient reduction An apparently related literature stream is concerned with sufficient dimensionality reduction [9, 12] where, however, the representation maps a high-dimensional data space to a lower-dimensional space, typically via linear projections. A key to our approach is to measure the complexity of a (stochastic) representation by its information content and not its dimensionality, thus allowing to build high dimensional representations that are still minimal. This concept was already underlying the Information Bottleneck principle, but only recently developments in variational Bayesian inference have made the computation of the underlying information quantities possible in cases other than linear/Gaussian, or discrete.

This paper naturally follows [1], where we showed that assuming independent marginals yields disentangled representations, and that the IB Lagrangian leads to a slightly modified loss function that also penalizes Total Correlation.

2 Preliminaries

2.1 Deep Neural Networks

A layer of a neural network is a nonlinear function that maps an input datum x to an "activation" via a map $f_W(x) := \phi(Wx)$ that consists of linear multiplication by a matrix W (*weight matrix*), followed by a non-linear activation function ϕ. Common activation functions include the rectified linear unit (ReLU), $\phi(z) = \max\{z, 0\}$. A deep neural network (DNN) is obtained by composing multiple layers $f_w(x) = \phi(W^L\phi(W^{L-1}\dots\phi(W^1x)))$. Often, layers have some special structure. For example, when working with spatial or temporal data, it is common to use a convolution as the linear operator.

2.2 Notation

A training dataset $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}$ is comprised of independent and identically distributed (IID) samples from an unknown distribution $p_\theta(x, y)$ parametrized by θ;

unless specified otherwise, we denote by $\mathbf{x} = \{x^{(i)}\}_{i=1}^N$ the measured data, while $\mathbf{y} = \{y^{(i)}\}_{i=1}^N$ are usually discrete labels associated with the data; a test datum is a random variable[2] x. Given a sample of x, our goal is to infer the random variable y, which is therefore referred to as our *task*.

For random variables x, y, z we denote by $\mathbb{E}_p(x)$ the expectation of x with respect to the measure $p(x)$. We will make frequent use of the following information-theoretic quantities [10]: Shannon entropy $H(x) = \mathbb{E}_p[-\log p(x)]$, conditional entropy $H(x|y) = H(x,y) - H(y)$, (conditional) mutual information $I(x;y|z) = H(x|z) - H(x|y,z)$, Kullbach-Liebler (KL) divergence $KL(p(x)||q(x)) = \mathbb{E}_p[\log p/q]$, cross-entropy $H_{p,q}(x) = \mathbb{E}_p[-\log q(x)]$, and total correlation $TC(z)$, defined as

$$\mathrm{TC}(z) = KL(p(z)||p(z_1)p(z_2),\ldots,p(z_L)),$$

where $p(z_i)$ are the marginal distribution of the components of z. Recall that the KL divergence between two distribution is non-negative and zero if and only if they are equal. In particular $\mathrm{TC}(z)$ is zero if and only if the components of z are independent, in which case we say that z is *disentangled*. We will make frequent use of the following identity:

$$I(z;x) = \mathbb{E}_{x\sim p(x)} \mathrm{KL}(\, p(z|x) \,\|\, p(z)\,).$$

We say that x, z, y form a Markov chain, indicated with $y \to x \to z$, if $p(y|x,z) = p(y|z)$. The Data Processing Inequality (DPI) for a Markov chain $x \to z \to y$ ensures that $I(x;z) \geq I(x;y)$.

2.3 Information Bottleneck

We say that z is a *representation* of x if z is a stochastic function of x, or equivalently if the distribution of z is fully described by the conditional $p(z|x)$. We say that a representation z of x is sufficient for y if $y \perp\!\!\!\perp x \mid z$, or equivalently if $I(z;y) = I(x;y)$; it is minimal when $I(x;z)$ is smallest among sufficient representations. The Information Bottleneck (IB) Lagrangian [32] is the functional

$$\mathcal{L}(p(z|x)) = H(y|z) + \beta\, I(z;x) \tag{1}$$

where β trades off sufficiency (first term) and complexity (second term); in the limit $\beta \to 0$, the IB Lagrangian is minimized when z is minimal and sufficient.

[2]We use the term random variable or random vector interchangeably.

2.4 Nuisances for a task

A **nuisance** is any random variable that affects the observed data x, but is not related to the task we are trying to solve. More formally, a random variable n is a nuisance for the task y if $y \perp\!\!\!\perp n$, or equivalently $I(y;n) = 0$. Similarly, we say that the representation z is **invariant** to the nuisance n if $z \perp\!\!\!\perp n$, or $I(z;n) = 0$. When z is not strictly invariant but minimizes $I(z;n)$ among all sufficient representations, we say it is **maximally insensitive** to n.

Remark 2.1 (Group nuisances)**.** A special case of nuisances are those that form a group G acting on the data, such as translations, or affine transforms of the image plane. In this case, it is easy to see that a deterministic function $f(x)$ is an invariant representation of the data if and only if for each $n \in G$, we have $f(n \cdot x) = f(x)$, in agreement with the more classical notion of invariant function. Our definition is more general in that it is not restricted to deterministic functions, nor to group nuisances. This is especially important for images, since occlusion and scaling/quantization do not form a group. Notice also that any deterministic invariant must assume the same value along the orbits of the group; an invariant function that assumes distinct values for each orbit is called *maximal*, or sometime *distinctive*. These are the only invariant representations of the data which are sufficient for *all tasks* for which G is a nuisance.

An important consequence of our definition of nuisance is that the observed data x can always be written as a deterministic function of the task y and of all nuisances n affecting the data, as explained by the following lemma.

Lemma 2.2 (Task-nuisance decomposition)**.** *Given a joint distribution $p(x,y)$, where y a discrete random variable, we can always find a random variable n independent of y such that $x = f(y,n)$, for some deterministic function f.*

Proof. Fix $n \sim \mathrm{Uniform}(0,1)$ to be the uniform distribution on $[0,1]$. We claim that, for a fixed value of y, there is a function $\Phi_y(n)$ such that $x|y = \Phi_{y*}(n)$, where $(\cdot)_*$ denotes the push-forward map of measures. Given the claim, let $\Phi(y,n) = (y, \Phi_y(n))$. Since y is a discrete random variable, $\Phi(y,n)$ is easily seen to be a measurable function and by construction $(x,y) \sim \Phi_*(y,n)$. To see the claim, notice that, since there exists a measurable isomorphism between $\mathbb{R}^n$ and $\mathbb{R}$ [6, Theorem 3.1.1], we can assume without loss of generality that $x \in \mathbb{R}$. In this case, by definition, we

can take $\Phi_y(n) = F_y^{-1}(n)$ where $F_y(t) = \mathbb{P}[x < t \,|\, y]$ is the cumulative distribution function of $p(x|y)$. □

3 Properties of representations

In many applications, the observed data x is high-dimensional (*e.g.,* images or video), while the task y is low-dimensional, *e.g.,* a label or a coarsely quantized location. For this reason, instead of working directly with x, we want to use a representation z that captures all the information the data x contains about the task y, while also being simpler than the data itself.

Ideally, such a representation should be (a) **sufficient** for the task y, *i.e.* $I(y;z) = I(y;x)$, so that information about y is not lost; among all sufficient representations, it should be (b) **minimal**, *i.e.* $I(z;x)$ is minimized, so that it retains as little about x as possible, simplifying the role of the classifier; finally, it should be (c) **invariant** to the effect of nuisances $I(z;n) = 0$, so that decisions based on the representation z will not overfit to spurious correlations between nuisances n and labels y present in the training dataset.

Assuming such a representation exists, it would not be unique, since any bijective function preserves all these properties. We can use this fact to our advantage and further aim to make the representation (d) maximally **disentangled**, *i.e.,* $\mathrm{TC}(z)$ is minimal. This simplifies the classifier rule, since no information is present in the complicated higher-order correlations between the components of z, a.k.a. "features." In short, an *ideal representation* of the data is a minimal sufficient invariant representation that is disentangled.

Inferring a representation that satisfies all these properties may seem daunting. However, in this section we show that we only need to enforce (a) sufficiency and (b) minimality, from which invariance and disentanglement follow naturally. Between this and the next section, we will then show that sufficiency and minimality of the learned representation can be promoted easily through implicit or explicit regularization during the training process.

Proposition 1 (Invariance and minimality)**.** *Let n be a nuisance for the task y and let z be a sufficient representation of the input x. Suppose that z depends on n only through x (*i.e., $n \to x \to z$*). Then,*

$$I(z;n) \leq I(z;x) - I(x;y).$$

Moreover, there exists a nuisance n such that equality holds up to a (generally small) residual ϵ

$$I(z;n) = I(z;x) - I(x;y) - \epsilon,$$

where $\epsilon := I(z;y|n) - I(x;y)$. In particular $0 \leq \epsilon \leq H(y|x)$, and $\epsilon = 0$ whenever y is a deterministic function of y. Under these conditions, a sufficient statistic z is invariant (maximally insensitive) to nuisances if and only if it is minimal.

Remark 3.1. The relevance of this proposition is that we can construct invariants by reducing the amount of information z contains about x, while retaining the minimum amount $I(z;x)$ that we need for the task y. This can be enforced using an IB Lagrangian (Corollary 3.3), but also happens implicitly in presence of a noisy or constrained optimization process. This provides the network a way to automatically learn invariance to complex nuisances, which is complementary to the invariance imposed by the architecture.

Proof. By hypothesis, we have the Markov chain $(y, n) \to x \to z$; therefore, by the DPI, we have $I(z;y,n) \leq I(z;x)$. The first term can be rewritten using the chain rule as $I(z;y,n) = I(z;n) + I(z;y|n)$, giving us

$$I(z;n) \leq I(z;x) - I(z;y|n).$$

Now, since y and n are independent, $I(z;y|n) \geq I(z;y)$. In fact,

$$\begin{aligned} I(z;y|n) &= H(y|n) - H(y|z,n) \\ &= H(y) - H(y|z,n) \\ &\geq H(y) - H(y|z) = I(y;z). \end{aligned}$$

Substituting in the inequality above, and using the fact that z is sufficient, we finally obtain

$$I(z;n) \leq I(z;x) - I(z;y) = I(z;x) - I(x;y).$$

Moreover, let n be as in Lemma 2.2. Then, since x is a deterministic function of y and n, we have

$$I(z;x) = I(z;n,y) = I(z;n) + I(z;y|n),$$

and therefore

$$I(z;n) = I(z;x) - I(z;y|n) = I(z;x) - I(x;y) - \epsilon.$$

with ϵ defined as above. Using the sufficiency of z, the previous inequality for $I(z;y|n)$, the DPI, we get the chain of inequalities

$$I(x;y) = I(z;y) \leq I(z;y|n) \leq I(x;y|n),$$

from which we obtain the desired bounds for ϵ. □

Remark 3.2 (Information in the nuisance)**.** Since $\epsilon \leq H(y|x)$, and usually $H(y|x) \ll I(x;z)$, we can generally ignore the extra term. In general, ϵ will be different from zero in pathological cases where knowing the exact nuisance n affecting the data x allows us to recover additional information about the task y.[3] Such instances tend to not occur in practice when inferring representations from real data.

We can now assess the consequences of Proposition 1 pertaining to invariance properties of real architectures. The most immediate regards the connection between the IB Lagrangian and invariance of the representations.

Corollary 3.3 (Invariants from the Information Bottleneck)**.** *In the limit $\beta \to 0$, minimizing the Information Bottleneck (IB) Lagrangian* (1) *yields a sufficient invariant representation z of the test datum x for the task y.*

Proof. A representation minimizing the IB Lagrangian in the limit $\beta \to 0$ is minimal and sufficient [32], and therefore invariant by Proposition 1. □

Remarkably, the IB Lagrangian can be seen as a the standard cross-entropy loss, plus a regularizer $I(z;x)$ that promotes invariance. This fact, without proof, is implicitly used in [1], that also provides an efficient algorithm to perform the optimization. [2] also propose a related algorithm and shows improved resistance to adversarial nuisances.

While the IB Lagrangian allows us to construct an optimal bottleneck, that promotes invariance, a well designed architecture can also create a similar bottleneck. This is the content of the next few propositions.

Corollary 3.4 (Bottlenecks promote invariance)**.** *Suppose we have the Markov chain of representations (layers)*

$$x \to z_1 \to z_2,$$

and suppose that there is a communication or computation bottleneck between z_1 and z_2 such that $I(z_1;z_2) < I(z_1;x)$. Such a bottleneck can happen for example because $\dim(z_2) < \dim(z_1)$ (e.g. pooling), or because the channel between z_1 and z_2 is noisy (e.g. because of Dropout). If z_2 is still sufficient, then it is more invariant to nuisances than z_1. More precisely, for all nuisances n we have

$$I(z_2;n) \leq I(z_1;z_2) - I(x;y).$$

[3] For instance, let n and y be a sequence of uniformly distributed bits, and let $x = y \oplus n$, where $\oplus$ denotes the exclusive "or" (XOR). Then $I(x;y) = 0$, but $I(x;y|n) = H(y)$ since, given n, we have $y = x \oplus n$.

Suppose we have a multi-layer architecture with activations $z_1, \ldots, z_L$, with the last corresponding to the class variable y, successfully trained to minimize the cross-entropy loss, which makes z_L a sufficient statistic of x for y. The following proposition shows the benefits of stacking layers.

Proposition 2 (Stacking increases invariance)**.** *Assume that z_L is sufficient of x for y, and we have the Markov chain*

$$x \to z_1 \to z_2 \to \ldots \to z_L.$$

Then all preceding representations z_i, $i < L$ are also sufficient; by the DPI, we have $I(z_L;x) \leq I(z_i;x)$ for each $i \leq L$, and by Proposition 1, z_L is more insensitive (invariant) to nuisances than all the preceding representations (layers).

Notice, however, that the above corollary does not simply imply that the more layers the merrier, as it assumes that one has successfully trained the network (z_L is sufficient), which becomes increasingly difficult as the size grows. Also note that some architectures, such as Residual Networks [17], do not form a Markov chain because of skip connections (however, their "blocks" still do).

Proposition 3 (Actionable Information)**.** *When $z = f(x)$ is deterministic, a representation that minimizes the IB Lagrangian also maximizes Actionable Information (AI), which is the entropy of a sufficient invariant $AI(x) = H(f(x))$.*

Proof. The encoding cost of $z = f(x)$ is $I(x;z) = H(z) - H(z|x)$ with the second term zero because z is a deterministic function of x. □

Up to this point we have studied representations *of the test datum x*, and showed how desirable properties such as minimality, invariance and disentanglement, can be enforced through minimizing regularized loss functions (Corollary 3.3) and architecture design choices (use of pooling, dropout noise). Still, networks trained as a black-box seem to yield invariant representation even when no explicit regularization is performed. To tackle this issue, in the next section we study the *weights of the network* as a representations *of the training set $\mathcal{D}$*, and in the following one the relation between the two, which is key to the emergence of desirable representational properties from agnostic black-box optimization.

4 Learning minimal weights

In this section we consider a deep network that implements a map $x \mapsto f_w(x) := q(\cdot\,|x, w)$ from an input x to a class y, trained to minimize the dataset cross-entropy loss

$$H_{p,q}(\mathbf{y}|\mathbf{x}, w) = \mathbb{E}_{\mathcal{D}} \sum_{i=1}^{N} -\log q(y^{(i)}|x^{(i)}, w)$$

with respect to w, in order for $q(y|x, w)$ to approximate $p_\theta(y|x)$.[4] One of the main problems in optimizing a DNN is that the cross-entropy loss in notoriously prone to overfitting. To gain some insights about the possible causes, we can use the following decomposition.

Proposition 4 (Information Decomposition)**.** *Let $\mathcal{D} = (\mathbf{x}, \mathbf{y})$ denote the training dataset, then for any training procedure, we have*

$$\begin{aligned} H_{p,q}(\mathbf{y}|\mathbf{x}, w) &= H(\mathbf{y}|\mathbf{x}, \theta) + I(\theta; \mathbf{y}|\mathbf{x}, w) + \\ &\quad \mathrm{KL}(\, q(\mathbf{y}|\mathbf{x}, w) \,\|\, p(\mathbf{y}|\mathbf{x}, w)\,) - I(\mathbf{y}; w|\mathbf{x}, \theta) \end{aligned} \quad (2)$$

Proof. Recall that cross-entropy can be written as

$$H_{p,q}(\mathbf{y}|\mathbf{x}, w) = H(\mathbf{y}|\mathbf{x}, w) + \mathrm{KL}(\, q(\mathbf{y}|\mathbf{x}, w) \,\|\, p(\mathbf{y}|\mathbf{x}, w)\,),$$

so we only have to prove that

$$H(\mathbf{y}|\mathbf{x}, w) = H(\mathbf{y}|\mathbf{x}, \theta) + I(\mathbf{y}; \theta|\mathbf{x}, w) - I(\mathbf{y}; w|\mathbf{x}, \theta),$$

which is easily done using the following identities:

$$\begin{aligned} I(\mathbf{y}; \theta|\mathbf{x}, w) &= H(\theta, \mathbf{y}|w) - H(\mathbf{y}|\theta, \mathbf{x}, w), \\ I(\mathbf{y}; w|\mathbf{x}, \theta) &= H(\mathbf{y}|\mathbf{x}, \theta) - H(\mathbf{y}|\mathbf{x}, \theta, \mathbf{w}). \end{aligned}$$

□

The first term of the right-hand side of (2) relates to the intrinsic error and depends on p_θ; the second term relates to the efficiency of the model and the class of functions f_w, the third depends on the universality of the architecture, and the last term relates to how much information about the labels is memorized in the weights. Without implicit or explicit regularization, the left-hand side (LHS) of (2) can be minimized by just maximizing the last term, *i.e.,* by memorizing the dataset, which yields poor generalization. This can be avoided by adding the last term back to the loss function, leading to a regularized loss $H_{p,q}(\mathbf{y}|\mathbf{x}, w) + I(\mathbf{y}; w|\mathbf{x}, \theta)$, where the negative term on the RHS is canceled. However, computing, or even approximating, the value of $I(\mathbf{y}, w|\mathbf{x}, \theta)$ is at least as difficult as fitting the model itself.

Notice however, that for w to be sufficient we only need to memorize in w the information that $\mathcal{D}$ has about θ, that is we need $I(w; \mathcal{D}) = I(\mathcal{D}; \theta) \leq H(\theta)$, which is bounded above by a constant. On the other hand, to overfit, the term $I(\mathbf{y}; w|\mathbf{x}, \mathcal{D}) \leq I(\mathcal{D}; w|\theta)$ needs to grow linearly with the number of training samples N. We can exploit this fact to prevent overfitting by adding a Lagrange multiplier β to make the amount of information a constant with respect to N, leading to the loss function

$$\mathcal{L}(p(w|\mathcal{D})) = H_{p,q}(\mathbf{y}|\mathbf{x}, w) + \beta I(w; \mathcal{D}), \quad (3)$$

which is, remarkably, the same IB Lagrangian in (1), but now interpreted as a function of w rather than z.

An alternative derivation for this regularized loss function is to seek ideal weights w that are a minimal sufficient representation of the dataset $\mathcal{D}$. In this case, again the IB Lagrangian (3) emerges as the natural training criterion. Moreover, it is precisely the standard cross-entropy loss with an added regularizer, which aims to limit the information content stored at the weights. This may at first appear inconsistent with common practice, which is to minimize the cross-entropy loss without an explicit regularizer.

Remark 4.1 (Information Bottleneck, Variational Learning, and Dropout)**.** Minimizing the information stored at the weights $I(w; \mathcal{D})$ was proposed as far back as [18] as a way of simplifying neural networks, but no efficient algorithm to perform the optimization was known. For the particular choice $\beta = 1$, the IB Lagrangian reduces to the variational lower-bound (VLBO) of the marginal log-likelihood $p(\mathbf{y}|\mathbf{x})$. Therefore, minimizing eq. (3) can be seen as a generalization of variational learning. A particular case of this was studied by [20], who showed that a generalization of Dropout, in conjunction with the *reparametrization trick* [21], could be used to minimize the loss efficiently, an idea later used in a number of related works [1, 2].

Remark 4.2 (Information in the weights as a measure of complexity)**.** Just as [18] suggested, we also advocate using the information regularizer $I(w; \mathcal{D})$ as a measure the effective complexity of a network, rather than the number of parameters $\dim(w)$. As we show in experiments (Section 6), this allows to recover a version of the bias-variance tradeoff where networks

[4] Note that we always treat the dataset $\mathcal{D}$ as a random variable. In practice, when a single dataset is given, the expectation w.r.t. the dataset can be ignored.

with lower information complexity underfit the data, and networks with higher complexity overfit. In contrast, there is no clear relationship between number of parameters and overfitting [36]. Moreover, in the case of random labels, the information complexity allows us to predict a sharp phase transition between overfitting and underfitting regimes, that can be observed in practice.

As we have seen, the IB Lagrangian emerges as a natural criterion *both* for inferring a representation of the test datum x that is sufficient and invariant (with no explicit notion of overfitting), and for inferring a representation w of the training dataset $\mathcal{D}$ that avoids overfitting (with no explicit notion of invariance). One of our main contribution is to show that this two aspects are not independent, but rather dual to each other. More precisely, that the minimality of the weights representation implies the minimality (and hence invariance) of the representation learned by the layers.

To derive precise and empirically verifiable statements about $I(w;\mathcal{D})$, we need an analytical expression for it. To this end, following [20], we make the following modeling assumptions.

Modeling assumptions. Let w denote the vector containing all the parameters (weights) in the network, and let W^k denote the weight matrix at layer k. We assume an improper log-uniform prior on w, that is $p(w_i) = c/|w_i|$. Notice that this is the only scale-invariant prior, and closely matches the real distributions of the weights in a trained network [20]. Then, we assume that the posterior distribution $p(w_i|\mathcal{D})$ is defined by

$$w_i|\mathcal{D} \sim \epsilon_i \hat{w}_i,$$

where $\hat{w}_i$ is a learned mean, and $\epsilon_i \sim \log\mathcal{N}(-\alpha_i/2, \alpha_i)$ is IID multiplicative log-normal noise with mean 1 and variance $\exp(\alpha_i) - 1$.[5] Note that we may think of the specified $p_{\hat{w},\alpha}(w|\mathcal{D})$ as being a local approximation of the real posterior $p(w|\mathcal{D})$, that we are making closer by optimizing the parameters α. However, in this work we prefer to follow a variational approach, and rather *define* the random variable w through the specified posterior $p_{\hat{w},\alpha}(w|\mathcal{D})$, so that no further assumption is required.

Claim 1 (Information in the weights)**.** *Under the previous modeling assumptions, the information the weights contain about the dataset is*

$$I(w;\mathcal{D}) = -\frac{1}{2}\sum_{i=1}^{\dim(w)} \log\alpha_i + C,$$

where the constant C is arbitrary due to the improper prior.

Proof. This is an easy consequence of the fact that the KL divergence is reparametrization invariant:

$$\begin{aligned}\mathrm{KL}(\, p(w|\mathcal{D}) \,\|\, p(w)\,) &= \mathrm{KL}(\, \log\mathcal{N}(\mu,\alpha) \,\|\, \log\mathrm{Uniform}\,) \\ &= \mathrm{KL}(\, \mathcal{N}(\mu,\alpha) \,\|\, \mathrm{Uniform}\,) \\ &= H(\mathcal{N}(\mu,\alpha)) + \mathrm{const} \\ &= -\sum_{i=1}^{\dim(w)} \frac{1}{2}\log(\alpha_i) + \mathrm{const},\end{aligned}$$

where we have used the formula for the entropy of a Gaussian and the fact that the KL divergence of a distribution from the uniform prior is the entropy of the distribution modulo an arbitrary constant. □

Remark 4.3 (On the constant C)**.** In every proposition that follows, we should assume a proper prior $\log\mathrm{Uniform}(a,b)$, and prove that some version of the proposition holds in the limit of $a \to -\infty$ and $b \to +\infty$. To simplify the exposition, since the optimization is unaffected by any additive constant, we abuse the notation and *define* $I(w;\mathcal{D}) := -\frac{1}{2}\sum_{i=1}^{\dim(w)} \log\alpha_i$ under the modeling assumptions stated above. Notice, however, that this expression *is not* reparametrization invariant without the constant C. This is relevant only in the discussion on flat minima in Remark 4.4.

Thus far we have suggested that adding the explicit information regularizer $I(w;\mathcal{D})$ prevents the network from memorizing the dataset and thus avoid overfitting, which we also confirm empirically in Section 6. However, real networks are not commonly trained with this regularizer, thus seemingly reducing the practical applicability of this theory. However, we claim that, even when not explicitly controlled, $I(w;\mathcal{D})$ is implicitly regularized by the use of SGD. In particular, empirical evidence suggests that [8] SGD biases the optimization toward "flat minima", that are local minima whose Hessian has mostly small eigenvalues. These minima can be interpreted exactly as having low information $I(w;\mathcal{D})$, as suggested early on by [19]. As a consequence of previous claims, flat minima can be seen as having better generalization properties. For completeness, we derive a more precise relationship between flatness (measured by the

[5] For a log-normal distribution $\log\mathcal{N}(\mu,\sigma^2)$ mean and variance are respectively $\exp(\mu+\sigma^2/2)$ and $[\exp(\sigma^2)-1]\exp(2\mu+\sigma^2)$.

nuclear norm of the Hessian of the loss), and the information regularizer, which is more closely related to overfitting.

Proposition 5 (Flat minima have low information)**.** *Let $\hat{w}$ be a local minimum of the cross-entropy loss $H_{p,q}(\mathbf{y}|\mathbf{x}, w)$, and let $\mathcal{H}$ be the Hessian at that point. Then, for the optimal choice of the posterior $w|\mathcal{D} = \epsilon \odot \hat{w}$ centered at $\hat{w}$ that optimizes the IB Lagrangian, we have*

$$I(w;\mathcal{D}) \leq \frac{1}{2} K[\log \|w\|_2^2 + \log \|H\|_* - K \log(K^2\beta/2)]$$

where $K = \dim(w)$ and $\|\cdot\|_*$ denotes the nuclear norm.

Remark 4.4. The bound above is not tight, but illustrates the qualitative dependency between flatness, that has been empirically associated with generalization but is not parametrization-invariant, and information in the weights, that has been suggested as a regularization criterion, and we have shown to be related to the Information Bottleneck. Increasing the flatness of a minimum decreases the information in the weights, but not vice-versa: It is possible for a sharp minimum to have low information and yield good generalization [11]. Note that the information in the weights is invariant to reparametrization only after accounting for the constant C, which we ignore as previously discussed.

Proof. First, we switch to a logarithmic parametrization of the weights, and let $h := \log |w|$ (we can ignore the sign of the weights since it is locally constant). In this parametrization, we can approximate the IB Lagrangian to second order as

$$\begin{aligned}\mathcal{L} =& \mathbb{E}_{h\sim p(h|\mathcal{D})}[H_0 + [(h-h_0)\odot w]^T \mathcal{H}[(h-h_0)\odot w] \\ & - \frac{\beta}{2}\sum_i \log \alpha_i\end{aligned}$$

where $H_0 = H(\mathbf{y}|\mathbf{x}, \hat{w})$. Now, notice that since $p(w|\mathcal{D})$ is a log-normal distribution, we have $p(h|\mathcal{D}) \sim \mathcal{N}(h_0, \alpha)$.[6] Therefore, can compute the expectation exactly as

$$\mathcal{L} = H_0 + \sum_{i=1}^{\dim(w)} \alpha_i w_i^2 \mathcal{H}_{ii} - \frac{\beta}{2}\sum_i \log \alpha_i.$$

Optimizing w.r.t. α_i we get

$$\alpha_i = \frac{\beta}{2w_i^2\mathcal{H}_{ii}},$$

[6]Note that we have ignored the offset $\alpha/2$ in the mean of the log-normal distribution to simplify the exposition.

and plugging it back in the expression for $I(w;\mathcal{D})$

$$I(w;\mathcal{D}) = \frac{1}{2}\sum_i \log(w_i^2) + \log(\mathcal{H}_{ii}) - \log(\beta/2).$$

Now, by Jensen's inequality, we have

$$\begin{aligned}I(w;\mathcal{D}) \leq & \frac{1}{2} K[\log(\sum_i w_i^2) + \log(\sum_i \mathcal{H}_{ii}) \\ & - \log(K^2\beta/2)] \\ = & \frac{1}{2} K[\log(\|w\|_2^2) + \log(\|H\|_*) \\ & - \log(K^2\beta/2)].\end{aligned}$$

□

In summary, we have shown that: (i) generalization can be improved by controlling the quantity of information in the weights; (ii) this can be done by explicit regularization using the IB Lagrangian; (iii) experimental evidence suggests that SGD implicitly minimizes the quantity of information in the weights even without regularization; (iv) this is related to properties of the loss landscape (flatness).

In the next section, we prove one of our main results, that networks with low information in the weights realize invariant and disentangled representations. Therefore, invariance and disentanglement emerge naturally when training a network with implicit (SGD) or explicit (IB Lagrangian) regularization.

5 Duality of the Bottleneck

In this section we establish a tight bound between the information in the weights w, which is minimized as part of the training process in a way that is agnostic of any properties of the resulting representation z, and the invariance and entanglement properties of the latter. Specifically, the information in the weights bounds total correlation and minimality, therefore low information in the weights implies invariance of the representation to nuisance variability.

The following proposition gives the fundamental link in our model between information in the weights, minimality of the representation, and disentanglement.

Proposition 6. *Let $z = Wx$, and assume as before $W = \epsilon \odot \hat{W}$, with $\epsilon_{i,j} \sim \log\mathcal{N}(-\alpha_i/2, \alpha_i)$. Further assume that the marginals of $p(z)$ and $p(z|x)$ are both*

approximately Gaussian (which is reasonable for large $\dim(x)$ *by the Central Limit Theorem). Then,*

$$I(z;x) + \mathrm{TC}(z) = \\ = -\frac{1}{2}\sum_{i=1}^{\dim(z)} \mathbb{E}_x \log \frac{\tilde{\alpha}_i \hat{W}_i^2 \cdot x^2}{\hat{W}_i \cdot \mathrm{Cov}(x)\hat{W}_i + \tilde{\alpha}_i \hat{W}_i^2 \cdot \mathbb{E}(x^2)}, \tag{4}$$

where W_i *denotes the* i*-th row of the matrix* W*, and* $\tilde{\alpha}_i$ *is the noise variance* $\tilde{\alpha}_i = \exp(\alpha_i) - 1$*. In particular,* $I(z;x) + \mathrm{TC}(z)$ *is a monotone decreasing function of the weight variances* α_i*.*

Proof. First, we consider the case in which $\dim(z) = 1$, and so $w := W$ is a single row vector. By hypothesis, $p(z)$ is approximately Gaussian, with mean and variance

$$\begin{aligned}
\mu_1 &:= \mathbb{E}[z] = \mathbb{E}[\sum_i \epsilon_i \hat{w}_i x_i] = \sum_i \hat{w}_i \mathbb{E}[x_i] = \hat{w} \cdot \mathbb{E}[x] \\
\sigma_1^2 &:= \mathrm{var}[z] = \mathbb{E}[(\sum_i \epsilon_i \hat{w}_i x_i)^2] - (\mathbb{E}[\sum_i \epsilon_i \hat{w}_i x_i])^2, \\
&= \mathbb{E}[\sum_{i,j} \epsilon_i \epsilon_j \hat{w}_i \hat{w}_j x_i x_j] - \sum_{i,j} \hat{w}_i \hat{w}_j \mathbb{E}[x_i]\mathbb{E}[x_j] \\
&= \tilde{\alpha} \sum_i \hat{w}_i^2 \mathbb{E}[x_i]^2 + \sum_{i,j} \hat{w}_i \hat{w}_j \left(\mathbb{E}[x_i x_j] - \mathbb{E}[x_i]\mathbb{E}[x_j]\right) \\
&= \tilde{\alpha}\hat{w}^2 \cdot \mathbb{E}[x^2] + \hat{w} \cdot \mathrm{Cov}(x)\hat{w}.
\end{aligned}$$

A similar computation gives us mean and variance of $p(z|x)$:

$$\begin{aligned}
\mu_0 &:= \mathbb{E}[z|x] = \hat{w} \cdot x, \\
\sigma_0^2 &:= \mathrm{var}[z|x] = \tilde{\alpha}\hat{w}^2 \cdot x^2.
\end{aligned}$$

Since we are assuming $\dim(z) = 1$, we trivially have $\mathrm{TC}(z) = 0$, so we are only left with $I(z;x)$ which is given by

$$\begin{aligned}
I(z;x) &= \mathbb{E}_x \,\mathrm{KL}(\, p(z|x) \,\|\, p(z) \,) \\
&= \mathbb{E}_x \,\mathrm{KL}(\, \mathcal{N}(\mu_0, \sigma_0^2) \,\|\, \mathcal{N}(\mu_1, \sigma_1^2) \,) \\
&= \frac{1}{2}\mathbb{E}_x \frac{\tilde{\alpha}\hat{w}^2 \cdot x^2 + (\hat{w} \cdot x - \hat{w} \cdot \mathbb{E}[x])^2}{\sigma_1^2} \\
&\quad - 1 - \log \frac{\sigma_0^2}{\sigma_1^2} \\
&= -\frac{1}{2}\mathbb{E}_x \log \frac{\tilde{\alpha}\hat{w}^2 \cdot x^2}{\hat{w} \cdot \mathrm{Cov}(x)\hat{w} + \tilde{\alpha}\hat{w}^2 \cdot \mathbb{E}[x^2]}.
\end{aligned}$$

Now, for the general case of $\dim(z) \geq 1$, notice that

$$\begin{aligned}
I(\mathbf{z};\mathbf{x}) + \mathrm{TC}(\mathbf{z}) &= \mathbb{E}_x \,\mathrm{KL}(\, \prod_k p(z_i|\mathbf{x}) \,\|\, \prod_k p(z_i) \,) \\
&= \sum_{i=1}^{\dim(z)} \mathbb{E}_x \,\mathrm{KL}(\, p(z_i|\mathbf{x}) \,\|\, p(z_i) \,),
\end{aligned}$$

where $p(z_i)$ is the marginal of the k-th component of z. We can then use the previous result for each component separately, and sum everything to get the desired identity. □

The above identity is difficult to apply in practice, but with some additional hypotheses, we can derive a cleaner uniform tight bound on $I(z;x) + \mathrm{TC}(z)$.

Proposition 7 (Uniform bound for one layer)**.** *Let* $z = Wx$*, where* $W = \epsilon \odot \hat{W}$*, where* $\epsilon_{i,j} \sim \log \mathcal{N}(-\alpha/2, \alpha)$*; assume that the components of* x *are uncorrelated, and that their kurtosis is uniformly bounded.*[7] *Then, there is a strictly increasing function* $g(\alpha)$ *s.t. we have the uniform bound*

$$g(\alpha) \leq \frac{I(x;z) + TC(z)}{\dim(z)} \leq g(\alpha) + c,$$

where $c = O(1/\dim(x)) \leq 1$*,* $g(\alpha) = \log(1 - e^{-\alpha})/2$ *and* α *is related to* $I(w;\mathcal{D})$ *by* $\alpha = \exp\{-I(W;\mathcal{D})/\dim(W)\}$*. In particular,* $I(x;z) + TC(z)$ *is tightly bounded by* $I(W;\mathcal{D})$ *and increases strictly with it.*

Proof. To simplify the notation we do the case $\dim z = 1$, the general case being identical. Let $w := W$ be the only row of W. First notice that, since x is uncorrelated, we have

$$\hat{w} \cdot \mathrm{Cov}(x)\hat{w} = \sum_i w_i^2 (\mathbb{E}[x_i^2] - \mathbb{E}[x_i]^2) \leq w^2 \cdot \mathbb{E}[x^2]$$

Therefore,

$$\begin{aligned}
I(x;z) &= -\frac{1}{2}\mathbb{E}_x \log \frac{\tilde{\alpha}\hat{w}^2 \cdot x^2}{\hat{w} \cdot \mathrm{Cov}(x)\hat{w} + \tilde{\alpha}\hat{w}^2 \cdot \mathbb{E}[x^2]} \\
&\leq -\frac{1}{2}\mathbb{E}_x \log \frac{\tilde{\alpha}\hat{w}^2 \cdot x^2}{(1+\tilde{\alpha})\hat{w}^2 \cdot \mathbb{E}[x^2]} \\
&= \frac{1}{2}\log(1 + \tilde{\alpha}^{-1}) \\
&\quad - \frac{1}{2}\mathbb{E}_x \log\left[1 + \frac{\hat{w}^2 \cdot (x^2 - \mathbb{E}[x^2])}{\hat{w}^2 \cdot \mathbb{E}[x^2]}\right].
\end{aligned}$$

[7] This is a technical hypothesis to avoid heavy tailed distributions, and is always satisfied if the components x_i are IID, (sub-)Gaussian, or with uniformly bounded support.

To conclude, we want to approximate the expectation of the logarithm using a Taylor expansion, but we first need to check that the variance of the term inside the logarithm is low, which is where we need the bound on the kurtosis. In fact, since the kurtosis is bounded, there is some constant C such that for all i

$$\frac{\mathbb{E}(x_i^2 - \mathbb{E}[x_i^2])^2}{\mathbb{E}[x_i^2]^2} \leq C.$$

Now,

$$\begin{aligned}\operatorname{var} \frac{\hat{w}^2 \cdot (x^2 - \mathbb{E}[x^2])}{\hat{w}^2 \cdot \mathbb{E}[x^2]} &= \frac{\sum_i \hat{w}_i^4 \mathbb{E}(x^2 - \mathbb{E}[x^2])^2}{\sum_{i,j} \hat{w}_i^2 \hat{w}_j^2 \mathbb{E}[x_i^2]\mathbb{E}[x_j^2]} \\ &\leq C \frac{\sum_i \hat{w}_i^4 \mathbb{E}[x_i^2]^2}{\sum_{i,j} \hat{w}_i^2 \hat{w}_j^2 \mathbb{E}[x_i^2]\mathbb{E}[x_j^2]} \\ &= O(1/\dim(x)).\end{aligned}$$

Therefore, we can conclude

$$I(x; z) \leq \frac{1}{2} \log(1 + \tilde{\alpha}^{-1}) + O(1/\dim(x)).$$

□

Corollary 5.1 (Multi-layer case)**.** *Let W^k for $k = 1, ..., L$ be weight matrices, with $W^k = \epsilon^k \odot \hat{W}^k$ and $\epsilon_{i,j}^k = \log \mathcal{N}(-\alpha^k/2, \alpha^k)$, and let $z_{i+1} = \phi(W^k z_k)$, where $z_0 = x$ and ϕ is any nonlinearity. Then,*

$$I(z_L; x) \leq \min_{k<L} \left\{ \dim(z_k) \left[g(\alpha^k) + 1 \right] \right\}$$

where $\alpha^k = \exp\left\{-I(W^k; \mathcal{D})/\dim(W^k)\right\}$.

Proof. Since we have the Markov chain $x \to z_1 \to \ldots \to z_L$, by the Data Processing Inequality we have $I(z_L; x) \leq \min\{I(z_L; z_{L-1}), I(z_{L-1}; x)\}$. Iterating this inequality, we have

$$I(z_L; x) \leq \min_{k<L} I(z_{k+1}, z_k).$$

Now, notice that $I(z_{k+1}; z_k) \leq I(\phi(W^k z_k); z_k) \leq I(W^k z_k; z_k)$, since applying a deterministic function can only decrease the information. But $I(W^k z_k; z_k)$ is exactly the quantity we bounded in Proposition 7, leading us to the desired inequality. □

Remark 5.2 (Tightness)**.** While the bound in Proposition 7 was tight, the bound in the multilayer case is not. This is to be expected: Reducing the information in the weights will create a bottleneck, but we do not know how much information about x will actually go through that bottleneck. In general, the last few layers will keep most information, while the first layer will drop a larger amount.

6 Empirical validataion

In this section we describe experiments on real networks to validate the assumptions underlying the claims above.

6.1 Random labels

In Proposition 4 and the following discussion, we listed among the reasons for overfitting the ability of the network to memorize the training set with all its nuisances (by maximizing the last term in (2)), rather than learning the informative part (minimizing the positive terms in (2)). This can be prevented by limiting the amount of information in the weights as done in eq. (3). As pointed out by [36], when a standard CNN is trained on CIFAR-10 to fit random labels, the network is able to (over)fit them perfectly. This is easily explained in our framework: It simply means that the network is complex enough to overfit but, as we show here, it has to pay a steep price in terms of information complexity of the weights. On the other hand, information regularization prevents overfitting in exactly the way predicted by the theory.

In particular, in the case of completely random labels, we have $I(w; \mathcal{D}|\theta) = I(w; \mathcal{D})$, since $\mathcal{D}$ is by construction random, and therefore independent from any θ. Therefore, the optimal regularizer is exactly eq. (3), and, in particular, empirical behavior of the network, shown in Figure 1, follows closely this prediction.

For real labels, the model is still able to overfit when $\beta < 1$, but importantly there is a large interval of $\beta > 1$ where the model fits the data *without* overfitting. Indeed, as soon as $\beta N \propto I(w; \mathcal{D})$ is larger than $I(\mathcal{D}; \theta)$, the model trained on real data fits real labels without overfitting independently of the particular value of β and N, as also shown in Figure 1.

In Figure 2, we measure the quantity information in the weights for different levels of corruption of the labels in the same experiment. To do this, we fix $\beta < 1$ so that the network is able to overfit, train until convergence, and then compute $I(w; \mathcal{D})$ for the trained model. As expected, increasing the randomness of the labels increases the quantity of information we need to fit the dataset. For completely random labels, $I(w; \mathcal{D})$ increases by ~ 2 nats/sample, which is close to the quantity required to memorize a 10-class labels (2.30 nats/sample), as shown in Figure 2.

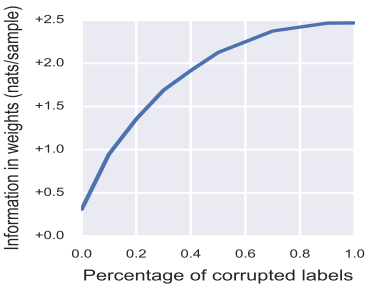

Figure 2: We measure the value of the information regularizer where transition to overfitting occurs as we vary the percentage of corrupted labels under the same settings of Figure 1. To fit increasingly random labels, the network needs to memorize more information in the weights; the increase needed to fit entirely random labels is about 2 nats per sample as expected.

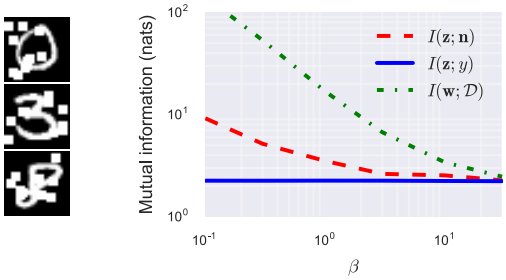

Figure 3: **(Right)**Reducing the information in the weights makes the representation z learned by the network increasingly invariant to nuisances ($I(n;z)$ decreases), while sufficiency is retained ($I(z;y) = I(x;y)$ is constant). Here the task is to classify digits in MNIST, and n is artificial clutter. The plot of $I(w,\mathcal{D})$ is modulo an arbitrary additive constant. **(Left)** A few training samples generated from the dataset.

6.2 Nuisance invariance

Corollary 5.1 shows that by decreasing $I(w;\mathcal{D})$, which can be done using Equation (3), the learned representation will be increasingly minimal, and therefore insensitive to nuisance factors n, as measured by $I(z;n)$. Here, we borrow a technique from the GAN literature [29] that allows us to explicitly measure $I(z;n)$ and validate this effect, provided we can sample from the nuisance distribution $p(n)$ and from $p(x|n)$; that is, if given a nuisance n we can generate data x affected by that particular nuisance. Recall that by definition we have

$$\begin{aligned} I(z;n) &= \mathbb{E}_{n\sim p(n)}\, \mathrm{KL}(\, p(z|n) \,\|\, p(z)\,) \\ &= \mathbb{E}_{n\sim p(n)}\mathbb{E}_{z\sim p(z|n)} \log[p(z|n)/p(z)]. \end{aligned}$$

Thus, to approximate the expectations we first need a way to approximate the likelihood ratio $\log p(\mathbf{z}|\mathbf{n})/p(\mathbf{z})$. This can be done as follows: Let $D(z;n)$ be a binary discriminator that given the representation z and the nuisance n tries to decide whether z is sampled from the posterior distribution $p(z|n)$ or from the prior $p(z)$. Since by hypothesis we can generate samples from both distributions, we can generate data to train this discriminator. Intuitively, if the discriminator is not able to classify, it means that z is insensitive to changes of n. More formally, since the optimal discriminator is

$$D^*(z|x,n) = \frac{p(z|x)}{p(z|x) + p(z|x,n)},$$

if we assume that D is close to the optimal discriminator D^*, we have

$$\log \frac{p(z|n,x)}{p(z|x)} = \log \frac{1 - D^*(z|x,n)}{D^*(z|x,n)} \simeq \log \frac{1 - D(z|x,n)}{D(z|x,n)}.$$

therefore we can use D to estimate the log-likelihood ratio, and so also the mutual information $I(z;n)$.[8] Notice however that this comes with no guarantees on the quality of the approximation.

To test this algorithm, we add random occlusion nuisances to MNIST digits (Figure 3). In this case, the nuisance n is the occlusion pattern, while the observed data x is the occluded digit. For various values of β, we train a classifier on this data in order to learn a representation z. Now, for each representation z obtained this way, a discriminator is trained as described above in order to estimate how insensitive z is to the occlusions n. The results are in (Figure 3).

6.3 Visualizing the representation

In the previous section, we showed how to measure nuisance invariance under the restrictive hypothesis that we can synthetically generate new nuisances. In this section, we show how, without any further assumption, we can visualize the information content of the representation z learned by the network. This way, we can visualize what kind of information is contained in the representation, and what nuisances are forgotten.

[8] Notice that if the output of D is given by a sigmoid, then the log-ratio is exactly the input of the sigmoid.

To this end, given a representation z, we want to learn a distribution $q(\hat{x}|z)$ of images that are maximally likely to have z as their representation. Formally, this means that we want a distribution $q(\hat{x}|z)$ that maximizes the amortized maximum a posteriori estimate of z:

$$\mathbb{E}_z \mathbb{E}_{\hat{x}\sim q(\hat{x}|z)}[\log p(\hat{x}|z)] = \mathbb{E}_z \underbrace{\mathbb{E}_{\hat{x}\sim q(\hat{x}|z)}[\log p(z|\hat{x})]}_{\text{Reconstruction error}} + \underbrace{\mathbb{E}_{\hat{x}\sim q(\hat{x})}[\log p(\hat{x})]}_{\text{Distance from prior}} + C.$$

Unfortunately, the term $p(\hat{x})$ in the expression is difficult to estimate. However, [29] notice that the modified gain function

$$\mathbb{E}_z \mathbb{E}_{\hat{x}\sim q(\hat{x}|z)}[\log p(\hat{x}|z)] + H(p(\hat{x})) = \mathbb{E}_z \mathbb{E}_{\hat{x}\sim q(\hat{x}|z)}[\log p(z|\hat{x})] + \mathrm{KL}(\, q(\hat{x}) \,\|\, p(\hat{x})\,) + C,$$

differs from the amortizes MAP only by a term $H(p(\hat{\mathbf{x}}))$, which has the positive effect of improving the exploration of the reconstruction, and contains the term $\mathrm{KL}(\, q(\hat{x}) \,\|\, p(\hat{x})\,)$, which can be estimated easily using the discriminator network of a GAN (see [29] for details).

To test this algorithm, we train a representation z to classify the 40 binary attributes in the CelebA face dataset [35], and then use the above loss function to train a GAN network to reconstruct the input image x from its representation z. The results in Figure 4 show that, as expected, increasing the value of β (and therefore reducing $I(w;\mathcal{D})$, generates samples that have increasingly more random backgrounds and hairdo (nuisances), while retaining facial features. In other words, the representation z is increasingly insensitive to nuisances affecting the data, while information pertaining the task is retained in the reconstruction $\hat{x}$.

7 Discussion

Deep networks are known to yield representations that discard irrelevant aspects of the data and isolate explanatory hidden factors within. How such behavior manages to emerge from simple black-box optimization however is deeply intriguing.

It has been conjectured that such desirable properties emerge from noisy computation in the network, or other operations that limit information flow such as pooling and dropout, but precisely how information flow connects to invariance and disentanglement has not been formalized, and to the best of our knowledge there are no previously known bounds that describes their relation.

In this work, we have presented bounds, some of which tight, that connect the amount of information in the weights, the amount of information in the activations, the invariance property of the network, and the geometry of the residual loss. These results leverage the structure of deep networks, in particular the multiplicative action of the weights, and the Markov property of the layers. This leads to the somewhat surprising result that reducing information stored in the weights about the past (dataset) results in desirable properties of the representation of future data (test datum).

We conducted experiments to validate the assumptions underlying the these bounds, and found that the results match the qualitative behavior observed on real data and architectures. In particular, the theory predicts a verifiable phase transition between an underfitting and overfitting regime for random labels, and the amount of information in nats needed to cross the transition. To gain more insight on the information contained in the representation learned by modern networks, we have leveraged on recent sampling technique, that shows that task-informative aspects of the data are retained, whereas uninformative variability is discarded.

As we have seen, what is informative and what is a nuisance is ultimately defined by the task, and therefore a representation is naturally task-dependent. Absent any knowledge about the task, the only sensible representation is a (compressed) copy of the data. However, it is possible for many tasks to share the same representation. It is also possible to determine tasks that are "universal" in the sense that the representation learned through them transfers to other tasks [13].

Our notion of representation is intrinsically stochastic. This simplifies the computation as well as the derivation of information-based relations. However, note that even if we start with a deterministic representation w, Proposition 5 gives us a way of converting it to a stochastic representation whose quality depends on the flatness of the minima. Our theory leverages heavily on the Information Bottleneck Principle, which dates back to over two decades ago, but that until recently was under-utilized because of the lack of tools to efficiently approximate and optimize the Information Bottleneck Lagrangian.

This work focuses on the inference and learning

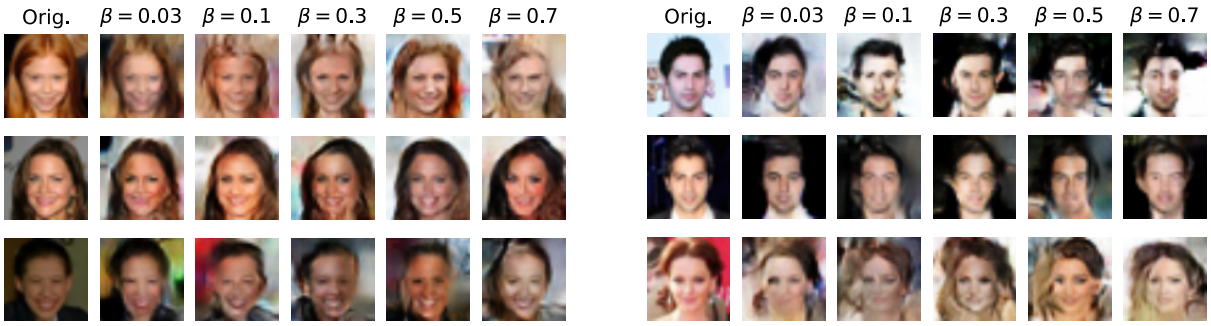

Figure 4: For different values of β, we show the image $\hat{x}$ reconstructed from a representation $z \sim p(z|x)$ of the original image x in the first column. For low values of β, the representation z contains most of the information regarding x, thus the reconstructed image $\hat{x}$ is close to x, background included. Increasing β to decrease the information in the weighs, the representation z becomes more invariant to nuisances, consequently the reconstructed image still matches important information in x that where retained in z (i.e. hair color, sex, expression), but background, lights, and other nuisances not contained in z change significantly across samples.

of optimal representations, that seek to get the most out of the data we have for a specific task. This does not guarantee a good outcome since, due to the Data Processing Inequality, the representation can be easier to use but ultimately no more informative than the data themselves. An orthogonal but equally interesting issue is how to get the most informative data possible, which is the subject of active learning, experiment design, and perceptual exploration.

Acknowledgments

Supported by ONR, ARO, AFOSR.

References

[1] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *ArXiv preprints arXiv:1611.01353*, 2016.

[2] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[3] F. Anselmi, L. Rosasco, and T. Poggio. On invariance and selectivity in representation learning. *arXiv preprint arXiv:1503.05938*, 2015.

[4] R. R. Bahadur. Sufficiency and statistical decision functions. *Annals of Mathematical Statistics*, 25(3):423–462, 1954.

[5] Y. Bengio. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

[6] Sterling K. Berberian. Borel spaces, April 1988.

[7] J. Bruna and S. Mallat. Classification with scattering operators. *arXiv preprint arXiv:1011.3023*, 2010.

[8] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, and Yann LeCun. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838*, 2016.

[9] F. Chiaromonte, B. Li, and R. Cook. Sufficient dimension reduction in regressions with categorical predictors. *Annals of Statistics*, 30:475–497, 2002.

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

[11] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*, 2017.

[12] K. Fukumizu, F. Bach, and M. I. Jordan. Kernel dimension reduction in regression. *Annals of Statistics*, 37:1871–1905, 2009.

[13] Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):3618–3623, 2015.

[14] S. Geman. Invariance and selectivity in the ventral visual pathway. *Journal of Physiology-Paris*, 100(4):212–224, 2006.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[16] U. Grenander. *General Pattern Theory*. Oxford University Press, 1993.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[18] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the 6th annual conference on Computational learning theory*, pages 5–13. ACM, 1993.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[20] Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 2575–2583, 2015.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, number 2014, 2013.

[22] Y. LeCun. Learning invariant feature hierarchies. In *ECCV*, pages 496–505, 2012.

[23] Holden Lee, Rong Ge, Andrej Risteski, Tengyu Ma, and Sanjeev Arora. On the ability of neural nets to express distributions. *arXiv preprint arXiv:1702.07028*, 2017.

[24] T. Poggio. The computational magic of the ventral stream. Technical report, Nature Precedings, 2011.

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

[26] S. Soatto. Actionable information in vision. In *Proc. of the Intl. Conf. on Comp. Vision*, October 2009.

[27] S. Soatto. *Steps Toward a Theory of Visual Information*. Technical Report UCLA-CSD100028, September 13, 2010.

[28] Stefano Soatto and Alessandro Chiuso. Visual representations: Defining properties and deep approximations. *Proceedings of the International Conference on Learning Representations (ICLR); ArXiv: 1411.7676*, May 2016.

[29] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

[30] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[31] G. Sundaramoorthi, P. Petersen, V. S. Varadarajan, and S. Soatto. On the set of images modulo viewpoint and contrast changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2009.

[32] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. In *The 37th annual Allerton Conference on Communication, Control, and Computing*, pages 368–377, 1999.

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

[34] G. Ver Steeg and A. Galstyan. Maximally informative hierarchical representations of high-dimensional data. *in depth*, 13:14, 2015.

[35] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3676–3684, 2015.

[36] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Input 32x32
Conv5 64 ReLU
MaxPool 2x2
Conv5 64 + BN ReLU
MaxPool 2x2
FC 3136x384 + BN ReLU
FC 384x192 + BN ReLU
FC 192x10
softmax

Table 1: The Small AlexNet model used in the experiments.

A Details of the experiments

A.1 Phase transition

We use a similar experimental setup as [36]. In particular, we train a small version of AlexNet on a 28x82 central crop of CIFAR-10 with completely random labels. The dataset is normalized using the global channel-wise mean and variance, but no additional data augmentation is performed. The exact structure of the network is in Table 1. As common in practice we use batch normalization before all the ReLU nonlinearities, except for the first layer. We train with learning rates $\eta = 0.02, 0.005$ and pick the best performing network of the two. Generally, we found that a higher learning rate is needed to overfit when N is small, while a lower learning rate is needed for larger N. We train with SGD with momentum 0.9 for 360 epochs reducing the learning rate by a factor of 10 every 140 epochs. We use a large batch-size of 500 to minimize the noise coming from SGD. No weight decay or other regularization methods are used.

The final plot is obtained by triangulating the convex envelope of the data points obtained this way, and by interpolating their value on the resulting simplexes.