Emergence of invariance and disentangling in deep representations

Alessandro Achille \(^1\) Stefano Soatto \(^1\) \(^2\)

Abstract

We show that invariance in a deep neural network is equivalent to information minimality of the representation it computes, and that stacking layers and injecting noise during training naturally bias the network towards learning invariant representations. Then, we show that overfitting is related to the quantity of information stored in the weights, and derive a sharp bound between this information and the minimality and Total Correlation of the layers. This allows us to conclude that implicit and explicit regularization of the loss function not only help limit overfitting, but also foster invariance and disentangling of the learned representation. We also shed light on the properties of deep networks in relation to the geometry of the loss function.

1. Introduction

Efforts to understand the empirical success of deep learning have followed two main lines: Representation learning and optimization. In optimization, 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 the non-convexity of the loss function, the fact that deep networks are heavily over-parametrized presents a theoretical challenge: The bias-variance tradeoff suggests they should overfit; yet, even without explicit regularization, SGD performs well in practice. Recent work suggests that this is related to properties of the loss landscape and to the implicit regularization performed by SGD, but the overall picture is still hazy (Zhang et al., 2017).

Representation learning, on the other hand, focuses on the properties of the representation learned by the layers of the network, while remaining largely agnostic to the particular optimization process used. In fact, the effectiveness of deep learning is often ascribed to the ability of deep networks to learn representations that are insensitive (invariant) to nuisances such as translations, rotations, occlusions, and also “disentangled,” separating factors in the high-dimensional space of data (Bengio et al., 2009). Careful engineering of the architecture plays an important role in achieving insensitivity to simple geometric 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 address these questions by establishing information theoretic connections between these concepts. In particular, we show that: (a) a sufficient representation is invariant if and only if it is minimal, i.e., it contain the

\(^{1}\) University of California, Los Angeles, USA \(^{2}\) Amazon Web Services, Los Angeles, USA. Correspondence to: Alessandro Achille <achille@cs.ucla.edu>, Stefano Soatto <soatto@ucla.edu>.

Presented at the ICML 2017 Workshop on Principled Approaches to Deep Learning, Sydney, Australia, 2017. Copyright 2017 by the author(s).
Emergence of invariance and disentangling in deep representations

smallest amount of information; (b) the information in the representation, along with its total correlation (a measure of disentanglement) are tightly bounded by the information that the weights retain about the dataset; (c) the information in the weights, which is related to overfitting (Hinton & Van Camp, 1993) and flat minima (Hochreiter & Schmidhuber, 1997), can be controlled by implicit or explicit regularization. Moreover, we show that adding noise during the training is a simple and natural way of biasing the network towards invariant representations.

Finally, we perform several experiments with 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 light on the questions of (Zhang et al., 2017).

1.1. Related work

The Information Bottleneck (IB) was introduced by Tishby et al. (1999) as a generalization of minimal sufficient statistic that allows trading off fidelity (sufficiency) and complexity of a representation. In particular, the introduction of the IB Lagrangian reduces finding a minimal sufficient representation of the data to a variational optimization problem. Later, Tishby & ZaSlavsky (2015); Shwartz-Ziv & Tishby (2017) advocated using this theory to study deep neural networks, since they build increasingly more minimal representations of the data. The IB Lagrangian can also be seen as a regularized loss function for training, leading to new information theoretic regularizers (Achille & Soatto, 2016; Alemi et al., 2017). Here, we extend this study and for the first time apply the IB to the weights of the network, related to Hinton & Van Camp (1993), and show how minimality of the weights and of the learned representation are connected. Practical implementation of this theory is made possible by advances in Stochastic Gradient Variational Bayes (Kingma & Welling, 2014; Kingma et al., 2015).

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 (Bruna & Mallat, 2011; Anselmi et al., 2016; Soatto & Chiuso, 2016). But for more complex, dataset-specific, and in particular non-local, non-group transformations, such insensitivity must be 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.

Efforts to develop a theoretical framework for representation learning include Tishby & ZaSlavsky (2015) and Shwartz-Ziv & Tishby (2017), who consider representations as stochastic functions that approximate minimal sufficient statistics, different from Bruna & Mallat (2011) 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. Anselmi et al. (2016) develop a theory of invariance to locally compact groups, and aim to construct maximal (“distinctive”) invariants, like Sundaramoorthi et al. (2009) that, however, assume nuisances to be infinite-dimensional groups (Grenander, 1993). These efforts are limited by the assumption that nuisances have a group structure. Other theoretical efforts focus on complexity considerations, and explain the success of deep networks by ways of statistical or computational efficiency (Lee et al., 2017; Bengio et al., 2009; LeCun, 2012). “Disentanglement” is an often-cited property of deep networks (Bengio et al., 2009), but seldom formalized and studied analytically, although Ver Steeg & Galstyan (2015) has suggested studying it using the Total Correlation of the representation, which we do.

We also connect invariance properties of the representation to the geometry of the optimization residual, and to the phenomenon of flat minima (Dinh et al., 2017).

2. Preliminaries

A training dataset \( \mathcal{D} = \{x, y\} \) is comprised of independent and identically distributed (IID) samples from an unknown distribution \( p_\theta(x, y) \) parametrized by \( \theta \); unless specified otherwise, we denote by \( x = \{x^{(i)}\}_{i=1}^{N} \) the measured data, while \( y = \{y^{(i)}\}_{i=1}^{N} \) are usually discrete labels associated with the data; a test datum is a random variable (or random vector) \( x \). Given a sample of \( x \), our goal is to infer the random variable \( y \), which is therefore referred to as our task.

We will make frequent use of the following standard information theoretic quantities (Cover & Thomas, 2012): the Shannon entropy \( H(x) = \mathbb{E}_p[- \log p(x)] \), the conditional entropy \( H(x|y) = H(x,y) - H(y) \), the (conditional) mutual information \( I(x; y|z) = \frac{1}{2} \{H(x|z) - H(x|y, z) \} \), the Kullbach-Liebler (KL) divergence \( K L(p(x)||q(x)) = \mathbb{E}_p[\log p/q] \), the cross-entropy \( H_{p,q}(x) = \mathbb{E}_p[- \log q(x)] \),
and the total correlation $TC(z)$, which is defined as

$$TC(z) = KL(p(z) \| \prod_{i} p(z_{i}) ),$$

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

$$I(z; x) = E_{z \sim p(z)} KL( p(z|x) \| p(z) ).$$

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

### 2.1. 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 $p(z|x)$. In particular we have the Markov chain $y \to x \to z$. We say that a representation $z$ of $x$ is sufficient for $y$ if $y \perp \!\!\!\!\!\!\perp z \mid x$, or equivalently if $I(y; z) = I(y; x)$; it is minimal when $I(x; z)$ is smallest among sufficient representations. To study the trade-off between sufficiency and minimality, Tishby et al. (1999) introduces the Information Bottleneck Lagrangian

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

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

### 2.2. 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 $n$.

One typical example of nuisance is a group $G$, such as translation or rotation, acting on the data. In this case, a deterministic representation $f$ is invariant to the nuisances if and only if for all $g \in G$ we have $f(y \cdot x) = f(x)$. Our definition however is more general in that it is not restricted to deterministic functions, nor to group nuisances. An important consequence of this generality 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 proposition.

**Proposition 2.1** (Task-nuisance decomposition). Given a joint distribution $p(x, y)$, where $y$ is 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 $f$.

### 3. Properties of representations

To simplify the inference process, instead of working directly with the observed high dimensional data $x$, we want to use a representation $z$ that captures and exposes only the relevant information about the task $y$. 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 the final classifier will not overfit to spurious correlations present in the training dataset between nuisances $n$ and labels $y$. Such a representation, if it exists, would not be unique, since any bijective mapping preserves all these properties. We can use this to our advantage and further aim to make the representation (d) maximally disentangled, i.e., such that $TC(z)$ is minimal. This simplifies the classifier rule, since no information will be present in the higher-order correlations between the components of $z$.

Infering 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 thanks to the stacking of noisy layers of computation in deep networks. 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 3.1** (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 $\epsilon$

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

where $\epsilon := I(z; y|n) - I(z; y)$. In particular $0 \leq \epsilon \leq H(y|x)$, and $\epsilon = 0$ whenever $y$ is a deterministic function of $x$. Under these conditions, a sufficient statistic $z$ is invariant (maximally insensitive) to nuisances if and only if it is minimal.
Remark 3.2. Since \( \epsilon \leq H(y|x) \), and usually \( H(y|x) \ll I(x;z) \), we can generally ignore the extra term.

The relevance of this proposition is that we can construct invariants by simply 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 provides the network a way to automatically learn invariance to complex nuisances, which is complementary to the invariance imposed by the architecture. One way of enforcing minimality explicitly, and hence invariance, is through the IB Lagrangian.

Corollary 3.3 (Invariants from the Information Bottleneck). Minimizing the ID Lagrangian

\[
\mathcal{L}(p(z|x)) = H(y|z) + \beta I(z;x),
\]

in the limit \( \beta \to 0 \), yields a sufficient invariant representation \( z \) of the test datum \( x \) for the task \( y \).

Remarkably, the ID Lagrangian can be seen as a standard cross-entropy loss, plus a regularizer \( I(z;x) \) that promotes invariance. This fact, without proof, is implicitly used in (Achille & Soatto, 2016), that also provides an efficient algorithm to perform the optimization. (Alemi et al., 2017) also propose a related algorithm and shows improved resistance to adversarial nuisances. In addition to modifying the cost function, invariance can also be fostered by choice of architecture.

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

\[
x \rightarrow z_1 \rightarrow z_2,
\]

and suppose that there is a communication or computation bottleneck between \( z_1 \) and \( z_2 \) such that \( I(z_1;z_2) < I(z_1;x) \). Then, if \( z_2 \) is still sufficient, it is more invariant to nuisances than \( z_1 \). More precisely, for all nuisances \( n \) we have \( I(z_2;n) \leq I(z_1;z_2) - I(x;y) \).

Such a bottleneck can happen for example because \( \dim(z_2) < \dim(z_1) \), e.g., after a pooling layer, or because the channel between \( z_1 \) and \( z_2 \) is noisy, e.g., because of dropout.

Proposition 3.5 (Stacking increases invariance). Assume that we have the Markov chain of layers

\[
x \rightarrow z_1 \rightarrow z_2 \rightarrow \ldots \rightarrow z_L,
\]

and that the last layer \( z_L \) is sufficient for \( x \) for \( y \). Then \( z_L \) is more insensitive to nuisances than all the preceding 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 \( (z_L \text{ is sufficient}) \), which becomes increasingly difficult as the size grows. Also note that in some architectures, such as ResNets (He et al., 2016), the layers do not necessarily form a Markov chain because of skip connections; however, their “blocks” still do.

Proposition 3.6 (Actionable Information). When \( z = f(x) \) is a deterministic invariant, if it minimizes the ID Lagrangian it also maximizes Actionable Information (Soatto, 2013), which is \( H(x) := H(f(x)) \).

Although Soatto (2013) addressed maximal invariants, we only need to consider sufficient invariants.

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\(^1\)

\[
H_{p,q}(y|x,w) = E_D \sum_{i=1}^{N} - \log q(y^{(i)}|x^{(i)},w),
\]

with respect to \( w \), in order for \( q(y|x,w) \) to approximate \( p_0(y|x) \). One of the main problems in optimizing a DNN is that the cross-entropy loss is notoriously prone to overfitting. To gain some insights about the possible causes, we can use the following decomposition:

\[
H_{p,q}(y|x,w) = H(y|x,\theta) + I(\theta;y|x,w) + \text{KL}(q(y|x,w) \| p(y|x,w)) - I(y,w|x,\theta). \tag{2}
\]

The first term of the right-hand side of (5) relates to the intrinsic error and depends on \( p_0 \); the second term measures how much information that the dataset has about the parameter \( \theta \) is captured by the weights, the third term relates to the efficiency of the model and the class of functions \( f_w \) with respect to which the loss is optimized, and the last, and only negative, term relates to how much information about the labels, but uninformative of the underlying data distribution, is memorized in the weights. Unfortunately, without implicit or explicit regularization, the left-hand side (LHS) of (5) 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 \( H_{p,q}(y|x,w) + I(\theta;y|x,w) \), where the negative term on the RHS is canceled. However, computing, or even approximating, the value of \( I(y,w|x,\theta) \) is at least as difficult as fitting the model itself.

However, notice that to successfully learn the distribution \( p_0 \), we only need to memorize in \( w \) the information

\(^1\) Note that for generality we always treat the dataset \( D \) as a random variable. In practice, when a single dataset is given, the expectation w.r.t. the dataset can be ignored.
about the latent parameters \( \theta \), that is we need \( I(D; w) = I(D; \theta) \leq H(\theta) \), which is bounded above by a constant. On the other hand, to overfit, the term \( I(y; w|x) \leq I(D; w|\theta) \) needs to grow linearly with the number of training samples \( N \). We can exploit this fact to prevent overfitting by adding a Lagrange multiplier \( \beta \) to make the amount of information a constant with respect to \( N \), leading to the regularized loss function

\[
\mathcal{L}(p(w|D)) = H_{p,q}(y|x, w) + \beta I(w; D),
\]

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

**Remark 4.1** (Information Bottleneck, Variational Learning, and Dropout). Minimizing the information stored at the weights \( I(w; D) \) was proposed as far back as Hinton & Van Camp (1993) as a way of simplifying neural networks, but no efficient algorithm to perform the optimization was known at the time. For the particular choice \( \beta = 1 \), the IB Lagrangian reduces to the variational lower-bound (VLBO) of the marginal log-likelihood \( p(y|x) \). Therefore, minimizing eq. (3) can also be seen as a generalization of variational learning. A particular case of this was studied by Kingma et al. (2015), who first showed that a generalization of Dropout, called Variational Dropout, could be used in conjunction with the reparametrization trick (Kingma & Welling, 2014) to minimize the loss efficiently.

**Remark 4.2** (Information in the weights as a measure of complexity). Just as Hinton & Van Camp (1993) suggested, we also advocate using the information regularizer \( I(w; D) \) as a measure the effective complexity of a network, rather than the number of parameters \( \text{dim}(w) \), which is merely an upper bound on the complexity. As we show in experiments, this allows us to recover a version of the bias-variance tradeoff where networks 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 (Zhang et al., 2017). Moreover, for random labels the information complexity allows us to precisely predict the overfitting and underfitting behavior of the network (Section 6).

To derive precise and empirically verifiable statements about \( I(w; D) \), we need an analytical expression for it. To this end, following (Kingma et al., 2015), 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 (Kingma et al., 2015), and closely matches the real distributions of the weights in a trained network. Then, we assume that the posterior distribution \( p(w_i|D) \) is defined by

\[
w_i|D \sim \epsilon_i \hat{w}_i,
\]

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

**Proposition 4.3** (Information in the weights). Under the previous modeling assumptions, the information the weights contain about the dataset is

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

where the constant \( C \) is arbitrary due to the improper prior.

**Remark 4.4** (On the constant \( C \)). To simplify the exposition, since the optimization is unaffected by any additive constant, in the following we abuse the notation and, under the modeling assumptions stated above, we rather define \( I(w; D) := -\frac{1}{2} \sum_{i=1}^{\text{dim}(w)} \log \alpha_i \). Neklyudov et al. (2017) also suggest a principled way of dealing with the arbitrary constant by using a proper log-uniform prior.

Thus far we have suggested that adding the explicit information regularizer \( I(w; D) \) 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 undermining the theory. However, we claim that, even when not explicitly controlled, \( I(w; D) \) is implicitly regularized by the use of SGD. In particular, empirical evidence (Chaudhari et al., 2017) suggests that 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 \( I(w; D) \), as suggested early on by Hochreiter & Schmidhuber (1997). 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 nuclear norm of the loss Hessian), and the information content.

**Proposition 4.5** (Flat minima have low information). Let \( \hat{w} \) be a local minimum of the cross-entropy loss \( H_{p,q}(y|x, w) \), and let \( \mathcal{H} \) be the Hessian at that point. Then,

\[
\text{For a log-normal } \log \mathcal{N}(\mu, \sigma^2) \text{ mean and variance are respectively } \exp(\mu + \sigma^2/2) \text{ and } [\exp(\sigma^2) - 1] \exp(2\mu + \sigma^2).}
\]
for the optimal choice of the posterior \( w \| D = \varepsilon \odot \tilde{w} \) centered at \( \tilde{w} \) that optimizes the IB Lagrangian, we have

\[
I(w; D) \leq \frac{1}{2} K \| \| \tilde{w} \|_2^2 + \log \| H \|_1 - K \log (K^2/2)]
\]

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

Notice that a converse inequality, that is, low information implies flatness, needs not hold, so there is no contradiction with the results of Dinh et al. (2017). Also note that for \( I(w; D) \) to be invariant to reparametrization one has to consider the constant \( C \), which we have ignored (Remark 4.4).

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, and are related to flat minima.

5. Duality of the Bottleneck

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

**Proposition 5.1.** Let \( z = W x \), and assume as before \( W = \varepsilon \odot \tilde{W} \), with \( \epsilon_{ij} \sim \log N(-\alpha_i/2, \alpha_i) \). 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,

\[
I(z; x) + TC(z) = -\frac{1}{2} \sum_{i=1}^{\dim(z)} \mathbb{E}_x \log \frac{\tilde{\alpha}_i \tilde{W}_i^2 \cdot x^2}{\tilde{W}_i \cdot \text{Cov}(x) W_i + \tilde{\alpha}_i \tilde{W}_i^2 \cdot \mathbb{E}(x^2)},
\]

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, \( I(z; x) + TC(z) \) is a monotone decreasing function of the weight variances \( \alpha_i \).

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

**Proposition 5.2** (Uniform bound for one layer). Let \( z = W x \), where \( W = \varepsilon \odot \tilde{W} \), where \( \epsilon_{ij} \sim \log N(-\alpha_i/2, \alpha_i) \); assume that the components of \( x \) are uncorrelated, and that their kurtosis is uniformly bounded. 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 \( \alpha \) is related to \( I(w; D) \) by \( \alpha = \exp(-I(w; D)/\dim(W)) \). In particular, \( I(x; z) + TC(z) \) is tightly bounded by \( I(w; D) \) and increases strictly with it.

Using the Markov property of the layers, we can now easily extend this bound to multiple layers.

**Corollary 5.3** (Multi-layer case). Let \( W^k \) for \( k = 1, ..., L \) be weight matrices, with \( W^k = e^k \odot \tilde{W}^k \) and \( e^k_{ij} = \log N(-\alpha^k_i/2, \alpha^k_i) \), and let \( z_{i+1} = \phi(W^k z_k) \), where \( z_0 = x \) and \( \phi \) is any nonlinearity. Then,

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

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

**Remark 5.4** (Tightness). While the bound in Proposition 5.2 is tight, the bound in the multilayer case needs not be. This is to be expected: Reducing the information in the weights creates a bottleneck, but we do not know how much information about \( x \) will actually go through this bottleneck. Often, the first layers will act as a bottleneck through, while initial layers will drop the most.

6. Empirical validation

As pointed out by Zhang et al. (2017), when a standard convolutional neural network (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 (Figure 2). 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 \( I(y; w|x, \theta) = I(y; w) \leq I(w; D) \), since \( y \) is by construction random. Therefore, eq. (3) is an optimal regularizer: Regardless of the dataset size \( N \), for \( \beta > 1 \) it should completely prevent memorization and hence overfitting, while for \( \beta < 1 \) overfitting is possible. The empirical behavior of the network, shown in Figure 1, closely follows 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 \( \beta N \propto I(w; D) \) is larger than \( H(\theta) \), the model trained on real data fits real labels without excessive overfitting (Figure 1).

In Figure 2, we measure the quantity information in the weights for different levels of corruption of the labels. To do this, we fix \( \beta < 1 \) so that the network is able to overfit, and for various level of corruption we train until convergence, and then compute \( I(w; D) \) for the trained
As expected, increasing the randomness of the labels increases the quantity of information we need to fit the dataset. For completely random labels, $I(w; D)$ increases by $\sim 3$ nats/sample, which is the same order of magnitude as the quantity required to memorize a 10-class labels (2.30 nats/sample), as shown in Figure 2.

6.1. Nuisance invariance

Corollary 5.3 shows that by decreasing the information in the weights $I(w; D)$, which can be done for example using eq. (3), the learned representation will be increasingly minimal, and therefore insensitive to nuisance factors $n$, as measured by $I(z; n)$. Here, we adapt a technique from the GAN literature (Sønderby et al., 2016) 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 nuisance. Recall that by definition we have

$$I(z; n) = \mathbb{E}_{n \sim p(n)} \text{KL}(p(z|n) \parallel p(z))$$

$$= \mathbb{E}_{n \sim p(n)} \mathbb{E}_{z \sim p(z|n)} \log[p(z|n)/p(z)].$$

To approximate the expectations via sampling we need a way to approximate the likelihood ratio $\log p(z|n)/p(z)$. 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 $n$. Precisely, since the optimal discriminator is

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

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

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

due to the logarithm, we can use $D$ to estimate the log-likelihood ratio, and so also the mutual information $I(z; n)$. 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 $\beta$, we train a classifier on this data in order to learn a representation $z$, and, for each representation obtained this way, we train a discriminator as described above and we compute the resulting approximation of $I(z; n)$. The results in Figure 3 show that decreasing the information in the weights makes the representation increasingly more insensitive to $n$.

6.2. Visualizing the representation

Even when we cannot generate data affected by nuisances like in the previous section, we can still visualize the information content of $z$ to learn what nuisances are discarded in the representation. To this end, given a representation $z$, we want to learn to sample from 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 $z$:

$$\mathbb{E}_z \mathbb{E}_{\hat{x} \sim q(\hat{x}|z)} [\log p(\hat{x}|z)] = \mathbb{E}_z \mathbb{E}_{\hat{x} \sim q(\hat{x}|z)} [\log p(z|\hat{x})]$$

Reconstruction error

$$+ \mathbb{E}_{\hat{x} \sim q(\hat{x})} [\log p(\hat{x})] + C.$$

Distance from prior
Emergence of invariance and disentangling in deep representations

Unfortunately, the term \( p(\hat{x}) \) in the expression is difficult to estimate. However, (Sønderby et al., 2016) notice that the modified gain function

\[
E_{\hat{z}} E_{z \sim q(\hat{z}|z)} [\log p(\hat{x}|z)] + H(p(\hat{x})) = E_{\hat{z}} E_{z \sim q(\hat{z}|z)} [\log p(z|\hat{x})] - KL(q(\hat{x}) || p(\hat{x})) + C,
\]

differs from the amortizes MAP only by a term \( H(p(\hat{x})) \), which has the positive effect of improving the exploration of the reconstruction, and contains the term \( KL(q(\hat{x}) || p(\hat{x})) \), which can be estimated easily using the discriminator network of a GAN (Sønderby et al., 2016). To maximize this gain, we can simply train a GAN with an additional reconstruction loss \( -\log p(z|\hat{x}) \).

To test this algorithm, we train a representation \( z \) to classify the 40 binary attributes in the CelebA face dataset (Yang et al., 2015), and then use the above loss function to train a GAN network to reconstruct an input image \( \hat{x} \) from the representation \( z \). The results in Figure 4 show that, as expected, increasing the value of \( \beta \), and therefore reducing \( I(w;D) \), generates samples that have increasingly more random backgrounds and hair style (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 and conclusion

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 surprising result that reducing information stored in the weights about the past (dataset) results in desirable properties of the learned internal representation of the test datum.

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 4.5 gives us a way of converting it to a stochastic representation whose quality depends on the flatness of the minimum. 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 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 N00014-17-1-2072, ARO W911NF-17-1-0304, AFOSR FA9550-15-1-0229.

References


Emergence of invariance and disentangling in deep representations


A. Details of the experiments

A.1. Random labels

We use a similar experimental setup as (Zhang et al., 2017). In particular, we train a small version of AlexNet on a 28×28 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 the number of training samples \( N \) 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, and by interpolating their value on the resulting simplex. Outside of the convex envelope (where the accuracy is mostly constant), the value was obtained by inpainting.

To measure the information content of the weights as the percentage of corrupted labels varies, we fix \( \beta = 0.1 \), \( N = 30000 \) and \( \eta = 0.005 \) and train the network on different corruption levels with the same settings as before.

A.2. Nuisance invariance

The cluttered MNIST dataset is generated by adding 10 \( 4 \times 4 \) squares uniformly at random on the digits of the MNIST dataset (LeCun et al., 1998). For each level of \( \beta \), we train the classifier in Table 1 on this dataset. The weights of all layers, excluding the first and last one, are treated as a random variable with multiplicative gaussian noise (Appendix B) and optimized using the local reparameterization trick (Kingma et al., 2015). We use the last convolutional layer before classification as representation \( z \).

The discriminator network used to estimate the log-likelihood ratio is constructed as follows: the inputs are the nuisance pattern \( n \), which is a 28×28×1 image containing 10 random occluding squares, and the 7×7×192 representation \( z \) obtained from the classifier. First we preprocess \( n \) using the following network:

<table>
<thead>
<tr>
<th>Input 32x32</th>
<th>192 \rightarrow \text{conv } 192 \rightarrow \text{conv } 1\times1\times192 \rightarrow \text{conv } 1\times1\times1 \rightarrow \text{AvgPooling } 7\times7 \rightarrow \text{sigmoid}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{conv } 64</td>
<td>\text{ReLU}</td>
</tr>
<tr>
<td>\text{MaxPool } 2\times2</td>
<td></td>
</tr>
<tr>
<td>\text{conv } 64 + \text{BN}</td>
<td>\text{ReLU}</td>
</tr>
<tr>
<td>\text{MaxPool } 2\times2</td>
<td></td>
</tr>
<tr>
<td>\text{FC } 3136x384 + \text{BN}</td>
<td>\text{ReLU}</td>
</tr>
<tr>
<td>\text{FC } 384x192 + \text{BN}</td>
<td>\text{ReLU}</td>
</tr>
<tr>
<td>\text{FC } 192x10</td>
<td>\text{softmax}</td>
</tr>
</tbody>
</table>

Table 1. (Left) The Small AlexNet model used in the random label experiment, adapted from (Zhang et al., 2017). All convolutions have a 5×5 kernel. The use of batch normalization makes the training procedure more stable, but did not significantly change the results of the experiments. (Right) All Convolutional Network (Springenberg et al., 2014) used as a classifier in the experiments. All convolutions but the last one use a 3×3 kernel, “s2” denotes a convolution with stride 2. The final representation we use are the activations of the last “\text{conv } 192” layer.

A.3. Visualizing the representation

We train a classifier on the images from the CelebA datasets resized to 32×32. The task is to recover the 40 binary attributes associated to each image. The classifier network is the same as the one in Table 1 with the following modifications: we use Exponential Linear Units (ELU) (Clevert et al., 2015) for the activations, instead of ReLU, since invertible activations generally perform better when training a GAN, and we divide by two the number of filters in all layers to reduce the training time. A sigmoid nonlinearity is applied to the final 40-way output of the network.

To generate the image \( \hat{x} \) given the 8×8×96 representation \( z \) computed by the classifier, we use a similar structure to DCGAN (Radford et al., 2016), namely \( z \rightarrow \text{conv } 256 \rightarrow \text{ConvT } 256\text{s2} \rightarrow \text{ConvT } 128\text{s2} \rightarrow \text{conv } 3 \rightarrow \text{tanh} \), where \text{ConvT } 256\text{s2} denotes a transpose convolution with 256 feature maps and stride 2. All convolutions have a batch normalization layer before the activations.

Finally, the discriminator network is \( \hat{x} \rightarrow \text{conv } 64\text{s2} \rightarrow \text{conv } 128\text{s2} \rightarrow \text{ConvT } 256\text{s2} \rightarrow \text{conv } 1 \rightarrow \text{sigmoid} \). Here, all convolutions use batch normalization followed by Leacky ReLU activations.

In this experiment, we use Gaussian multiplicative noise which is slightly more stable during training (Appendix B). To stabilize the training of the GAN, we found useful to (1) scale down the “reconstruction error” term in the loss func-

Emergence of invariance and disentangling in deep representations
tion and (2) slowly increase the weight of the reconstruction error up to the desired value during training.

**B. Gaussian multiplicative noise**

In developing the theory, we chose to use log-normal multiplicative noise for the weights: The main benefit is that with this choice the information in the weights \( I(w; D) \) can be expressed in closed form, up to an arbitrary constant \( C \) which does not matter during the optimization process (but see also Neklyudov et al., 2017 for a principled approach to this problem that uses a proper log-uniform prior). Another possibility, suggested by Kingma et al. (2015) is to use Gaussian multiplicative noise with mean 1. Unfortunately, there is no analytical expression for \( I(w; D) \) when using Gaussian noise, but \( I(w; D) \) can still be approximated numerically (Molchanov et al., 2017) with high precision, and it makes the training process slightly more stable. All our theory holds with minimal changes also in this case, and we use this choice in some experiments.

**C. Proofs of theorems**

**Lemma** (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 \( n \sim \text{Uniform}(0, 1) \) 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 \( \cdot | \cdot \) 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 \( (x, y) \sim \Phi_y(y, n) \). To see the claim, notice that, since there exists a measurable isomorphism between \( \mathbb{R}^n \) and \( \mathbb{R} \) (Theorem 3.1.1 of Berberian (1988)), we can assume without loss of generality that \( x \in \mathbb{R} \). In this case, by definition, we can take \( \Phi_y(n) = F^{-1}_y(n) \) where \( F_y(t) = \mathbb{P}[x < t | y] \) is the cumulative distribution function of \( p(x|y) \). \( \square \)

**Proposition** (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., \( n \rightarrow x \rightarrow z \)). Then,

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

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

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

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

*Proof.* By hypothesis, we have the Markov chain \((y, n) \rightarrow x \rightarrow z\); 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,

\[
I(z; y | n) = H(y | n) - H(y | z, n) \\
= H(y) - H(y | z, n) \\
\geq H(y) - H(y | z) = I(z; y).
\]

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

\[
I(z; n) \leq I(z; x) - I(z; y) = I(z; x) - I(z; y) - \epsilon.
\]

Moreover, let \( n \) be as in Proposition 2.1. 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

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

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

\[
\epsilon = I(z; y | n) - I(z; x | z) \leq I(z; y | n) - I(z; x) \\
\leq H(y | n) - H(y | z, n) - H(y) + H(y | x) \\
\leq H(y) - H(y | n, z) - H(y) + H(y | x) \\
= H(y | x) - H(y | n, z) \\
\leq H(y | x)
\]

from which we obtain the desired bounds for \( \epsilon \). \( \square \)

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

\[
H_{p,q}(y|x, w) = H(y|x, \theta) + I(\theta; y|x, w) + KL(p(y|x, w) || q(y|x, w)) - I(y; w|x, \theta)
\]

(5)

*Proof.* Recall that cross-entropy can be written as

\[
H_{p,q}(y|x, w) = H(y|x, w) + KL(q(y|x, w) || p(y|x, w)),
\]
so we only have to prove that
\[ H(y|x, w) = H(y|x, \theta) + I(y; \theta|x, w) - I(y; w|x, \theta), \]
which is easily done using the following identities:
\[
\begin{align*}
I(y; \theta|x, w) &= H(\theta, y|w) - H(y|\theta, x, w), \\
I(y; w|x, \theta) &= H(y|x, \theta) - H(y|x, \theta, w).
\end{align*}
\]

\[ \square \]

**Proposition (Information in the weights).** Under the previous modeling assumptions, the information the weights contain about the dataset is
\[ I(w; D) = -\frac{1}{2} \sum_{i=1}^{\dim(w)} \log \alpha_i + C, \]
where \( C \) is arbitrary due to the improper prior.

**Proof.** This is an easy consequence of the fact that the KL divergence is reparametrization invariant:
\[
\begin{align*}
\text{KL}(p(w|D) \parallel p(w)) &= \text{KL}(\log \mathcal{N}(\mu, \alpha) \parallel \log \text{Uniform}) \\
&= \text{KL}(\mathcal{N}(\mu, \alpha) \parallel \text{Uniform}) \\
&= H(\mathcal{N}(\mu, \alpha)) + \text{const} \\
&= -\sum_{i=1}^{\dim(w)} \frac{1}{2} \log(\alpha_i) + \text{const},
\end{align*}
\]
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.

\[ \square \]

**Proposition (Flat minima have low information).** Let \( \hat{w} \) be a local minimum of the cross-entropy loss \( H_{p,q}(y|x, \hat{w}) \), and let \( \mathcal{H} \) be the Hessian at that point. Then, for the optimal choice of the posterior \( w|D = \epsilon \odot \hat{w} \) centered at \( \hat{w} \) that optimizes the IB Lagrangian, we have
\[ I(w; D) \leq \frac{1}{2} K [\log \| w \|^2 + \log \| H \|_\star - K \log(K^2 \beta/2)] \]
where \( K = \dim(w) \) and \( \| \cdot \|_\star \) denotes the nuclear norm.

**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
\[
\mathcal{L} = \mathbb{E}_{h \sim p(h|D)} [H_0 + ((h - h_0) \odot w)^T \mathcal{H} ((h - h_0) \odot w) - \frac{1}{2} \sum_i \log \alpha_i \\
- \frac{1}{2} \sum \log \alpha_i]
\]
where \( H_0 = H(y|x, \hat{w}) \). Now, notice that since \( p(w|D) \) is a log-normal distribution, we have \( p(h|D) \sim \mathcal{N}(h_0, \alpha) \).

Thus, 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. \( \alpha_i \) we get
\[ \alpha_i = \frac{\beta}{2w_i^2 \mathcal{H}_{ii}}, \]
and plugging it back in the expression for \( I(w; D) \)
\[ I(w; D) = \frac{1}{2} \sum_i \log(w_i^2) + \log(\mathcal{H}_{ii}) - \log(\beta/2). \]

Now, by Jensen’s inequality, we have
\[ I(w; D) \leq \frac{1}{2} K [\log(\sum_i w_i^2) + \log(\sum_i \mathcal{H}_{ii}) - \log(\beta/2)] \]
\[ = \frac{1}{2} K [\log(\| w \|^2) + \log(\| H \|_\star) - \log(\beta/2)]. \]

\[ \square \]

**Proposition.** Let \( z = Wx \), and assume as before \( W = \epsilon \odot \hat{W} \), with \( \epsilon_{i,j} \sim \log \mathcal{N}(\alpha_{i,j}, \alpha_i) \). 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,
\[ I(z; x) + TC(z) = \]
\[ = -\frac{1}{2} \sum_{i=1}^{\dim(z)} \mathbb{E}_x \log \frac{\alpha_i W_i^2 \cdot x^2}{W_i \cdot \text{Cov}(x) W_i + \alpha_i W_i^2 \cdot \mathbb{E}(x^2)}, \]
where \( W_i \) denotes the \( i \)-th row of the matrix \( W \), and \( \alpha_i \) is the noise variance \( \alpha_i = \exp(\alpha_i) - 1 \). In particular, \( I(z; x) + TC(z) \) is a monotone decreasing function of the weight variances \( \alpha_i \).

**Proof.** First, we consider the case in which \( \dim(z) = 1 \), and so \( w = W \) is a single row vector. By hypothesis, \( p(z) \)
\[ \footnote{Note that for simplicity we have ignored the offset \( \alpha/2 \) in the mean of the log-normal distribution.} \]
Emergence of invariance and disentangling in deep representations

is approximately Gaussian, with mean and variance

$$
\mu_1 := \mathbb{E}[z] = \mathbb{E}[\sum_i \epsilon_i \hat{w}_i x_i] = \sum_i \hat{w}_i \mathbb{E}[x_i] = \hat{w} \cdot \mathbb{E}[x]
$$

$$
\sigma_1^2 := \text{var}[z] = \mathbb{E}[\left(\sum_i \epsilon_i \hat{w}_i x_i\right)^2] - (\mathbb{E}[\sum_i \epsilon_i \hat{w}_i x_i])^2,
$$

$$
= \mathbb{E}[\sum_{i,j} \epsilon_i \epsilon_j \hat{w}_i \hat{w}_j x_i x_j] - \sum_i \hat{w}_i \mathbb{E}[x_i] \mathbb{E}[x_j]
$$

$$
= \hat{\alpha} \sum_{i,j} \hat{w}_i^2 \mathbb{E}[x_i^2] + \sum_{i,j} \hat{w}_i \hat{w}_j (\mathbb{E}[x_i x_j] - \mathbb{E}[x_i] \mathbb{E}[x_j])
$$

$$
= \hat{\alpha} \hat{w}^2 \cdot \mathbb{E}[x^2] + \hat{w} \cdot \text{Cov}(x) \hat{w}.
$$

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

$$
\mu_0 := \mathbb{E}[z|x] = \hat{w} \cdot x,
$$

$$
\sigma_0^2 := \text{var}[z|x] = \hat{\alpha} \hat{w}^2 \cdot x^2.
$$

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

$$
I(z; x) = \mathbb{E}_x \mathbb{K}L(p(z|x) \parallel p(z))
$$

$$
= \mathbb{E}_x \mathbb{K}L(\mathcal{N}(\mu_0, \sigma_0^2) \parallel \mathcal{N}(\mu_1, \sigma_1^2))
$$

$$
= \frac{1}{2} \mathbb{E}_x \hat{\alpha} \hat{w}^2 \cdot x^2 + (\hat{w} \cdot x - \hat{w} \cdot \mathbb{E}[x])^2
$$

$$
= \frac{1}{2} \mathbb{E}_x \hat{\alpha} \hat{w}^2 \cdot x^2 + 1 - \log \frac{\sigma_0^2}{\sigma_1^2}
$$

$$
= -\frac{1}{2} \mathbb{E}_x \log \frac{\hat{\alpha} \hat{w}^2 \cdot x^2}{\hat{w} \cdot \text{Cov}(x) \hat{w} + \hat{\alpha} \hat{w}^2 \cdot \mathbb{E}[x^2]}.
$$

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

$$
I(z; x) + TC(z) = \mathbb{E}_x \mathbb{K}L(\prod_k p(z_k|x) \parallel \prod_k p(z_k))
$$

$$
= \sum_{i=1}^{\dim(z)} \mathbb{E}_x \mathbb{K}L(p(z_i|x) \parallel p(z_i)),
$$

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.

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

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

Therefore,

$$
I(x; z) = -\frac{1}{2} \mathbb{E}_x \log \frac{\hat{\alpha} \hat{w}^2 \cdot x^2}{\hat{w} \cdot \text{Cov}(x) \hat{w} + \hat{\alpha} \hat{w}^2 \cdot \mathbb{E}[x^2]}
$$

$$
\leq -\frac{1}{2} \mathbb{E}_x \log \frac{\hat{\alpha} \hat{w}^2 \cdot x^2}{\hat{w}^2 \cdot \mathbb{E}[x^2]}
$$

$$
= \frac{1}{2} \log(1 + \hat{\alpha}^{-1})
$$

$$
-\frac{1}{2} \mathbb{E}_x \log \left[1 + \frac{\hat{w}^2 \cdot (x^2 - \mathbb{E}[x^2])}{\hat{w}^2 \cdot \mathbb{E}[x^2]}\right].
$$

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,

$$
\text{var} \frac{\hat{w}^2 \cdot (x^2 - \mathbb{E}[x^2])}{\hat{w}^2 \cdot \mathbb{E}[x^2]} = \sum_{i,j} \hat{w}_i^2 \hat{w}_j^2 \mathbb{E}[x_i^2 - \mathbb{E}[x_i]^2]^2
$$

$$
\leq C \sum_{i,j} \hat{w}_i^2 \hat{w}_j^2 \mathbb{E}[x_i^2]^2 \mathbb{E}[x_j^2]
$$

$$
= O(1/\text{dim}(x)).
$$

Therefore, we can conclude

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

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

$$
I(z_L; x) \leq \min_{k < L} \{\dim(z_k) \lfloor g(\alpha^k) + 1 \rfloor\}
$$

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

Proof. Since we have the Markov chain $x \rightarrow z_1 \rightarrow \ldots \rightarrow z_L$, 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 5.2, leading us to the desired inequality. $\square$