On the Information Bottleneck

The Information Bottleneck, introduced by Naftali Tishby et al. in 1999, formalizes the notion of an information-theoretic "optimal" representation in terms of the fundamental tradeoff between having a concise representation (compression) and one with good predictive power (accuracy). We draw connections to (1) minimal sufficient statistics, (2) the dynamics of stochastic gradient descent, (3) topologies of deep neural networks, (4) generalization bounds in learning theory, and (5) the formulation of variational auto-encoders.

1 Information Theory

Let X, Y, Z denote discrete random variables. The Shannon **entropy** captures the notion of the "amount of information" held in a random variable X:

$$H(X) = \mathbb{E}_p[-\log p(x)] = -\sum_x p(x) \log p(x). \tag{1}$$

The **conditional entropy** of Y given X quantifies the "amount of information" needed to describe the outcome of Y given that the value of X is known

$$H(Y|X) = -\sum_{x,y} p(x,y) \log p(y|x) = H(X,Y) - H(Y). \tag{2}$$

The **cross entropy** between X and Y of identical underlying set of events measures the average number of bits needed to identify an event drawn from the set

$$H(X,Y) = \mathbb{E}_p[-\log q(x)] = -\sum_x p(x) \log q(x) = H(P) + D_{KL}(P \,\|\, Q). \tag{3}$$

The **relative entropy** (Kullback-Leibler divergence) of X with respect to Y is a measure of how one distribution P diverges from a expected distribution Q

$$D_{KL}(P \,\|\, Q) = \mathbb{E}_p[log \frac{p(x)}{q(x)}] = \sum_x p(x) \log \frac{p(x)}{q(x)} = H(P,Q) - H(P). \tag{4}$$

The **mutual information** quantifies the "amount of information", the average number of relevant bits, obtained about one random variable X, through the other random variable Y:

$$I(X;Y) = D_{KL}(p(x,y) \| p(x)p(y)) = \mathbb{E}_p[D_{KL}(p(x|y) \| p(x))] = H(X) - H(X|Y). \tag{5}$$

It measures the inherent dependence expressed in the joint distribution of X and Y relative to the joint distribution of X and Y under the assumption of independence:

$$X \perp Y \;\Leftrightarrow\; I(X;Y) = 0. \tag{6}$$

We say X, Y, Z form a ***Markov chain*** denoted by

$$X \to Z \to Y, \tag{7}$$

if the Markov property $P(Y|X,Z) = P(Y|Z)$ holds. A simple random walk is a Markov chain. The **Data Processing Inequality** (DPI) for a Markov chain $X \to Z \to Y$ ensures

$$I(X;Z) \geq I(X;Y). \tag{8}$$

The **Reparametrization Invariance** for invertible transformation ψ and ϕ ensures

$$I(X;Y) = I(\psi(X);\phi(Y)). \tag{9}$$

2 Optimal Representation Learning

Optimal Representation

Let random variable X denote an input, Z a representation of X, and Y observed output. We want to find an **optimal representation** Z satisfying the following conditions:

1. Z is a **representation** of X, that is, Z only depends on X, or $Y \to X \to Z$.
2. Z is **sufficient** to infer Y, that is $I(X;Y) = I(Z;Y)$, or $Y \to Z \to X$.
3. Z has **minimal** mutual information $I(X;Z)$ among all Z satisfying (1-2).
4. Z is **invariant** to the effect of noise ϵ on X, that is $I(Z;\epsilon) = 0$ for any ϵ.
5. Z is **disentangled**, that is the components $\{Z_i\}$ are maximal independent.

Then we cast representation learning as the problem of the finding an **optimal encoder** as mapping $X \to Z$ such that Z satisfies the above conditions.

Information Bottleneck

Let us focus on the first three conditions. The classical notion of **minimal sufficient statistics** provides good candidates for optimal representations. Sufficient statistics $S(X)$ are a partition on X, that captures all the information that X has on Y:

$$I(S(X);Y) = I(X;Y). \tag{10}$$

Minimal sufficient statistics, $Z(X)$, are the simplest sufficient statistics and induce the coarsest sufficient partition on X. Formally, they are functions of any other sufficient statistic. We can formulate this by a Markov chain:

$$Y \to X \to S(X) \to Z(X), \tag{11}$$

which holds for any minimal sufficient statistic $Z(X)$ with any other sufficient statistic $S(X)$. Using the DPI (8), we cast this into an optimization problem:

$$Z(X) = \underset{\{S(X):I(S(X);Y)=I(X;Y)\}}{\arg\min} I(X;S(X)). \tag{12}$$

Since exact minimal sufficient statistics only exist for distributions of exponential families, Tishby relaxed this optimization problem by first, allowing the map to be stochastic, defined as an encoder $P(Z|X)$, and second, by allowing the map to capture *as much as possible* of $I(X;Y)$, not necessarily all of it.

This is known as the **information bottleneck** tradeoff, which provides a computational framework for finding approximate minimal sufficient statistics, or, the optimal tradeoff between compression of X and prediction of Y. In this sense, efficient representations are approximate minimal sufficient statistics.

Define $z \in Z$ as a compressed representation of $x \in X$, then the mapping $p(z|x)$ defines the representation of x. This information bottleneck tradeoff is formulated by the following optimization problem with Markov chain $Y \to X \to Z$,

$$Z(X) = \underset{p(z|x)}{\arg\min}\, I(X;Z) - \beta I(Z;Y), \quad \beta > 0. \tag{13}$$

The Lagrange multipler β determines the level of relevant information $I(Z;Y)$ captured by the representation Z, where large β corresponds to high $I(Z;Y)$, and hence low compression.

To obtain the information bottleneck optimality equations, we solve the following optimization problem, carried independently for the distributions $p(z|x), p(z), p(y|z)$,

$$\min_{p(z|x),p(z),p(y|z)} I(X;Z) - \beta I(Z;Y), \quad \beta > 0. \tag{14}$$

The implicit solution to this problem is given by three self-consistent equations:

$$\begin{cases} p(z|x) &= \frac{p(z)}{Z_\beta(x)} \exp\left(-\beta D_{KL}\left[p(y|x) \| p(y|z)\right]\right) \\ p(z) &= \int p(z|x)p(x)dx \\ p(y|z) &= \int p(y|x)p(x|z)dx, \end{cases} \tag{15}$$

where $Z_\beta(x)$ denotes the normalization function. These equations are satisfied along the **information curve**, which is a monotonic concave line of optimal representations that separates achievable and unachievable regions in the information-plane. For smooth $P(X,Y)$, i.e. when Y is not a completely deterministic function of X, the information curve is strictly concave with unique slope β^{-1}, at every point. In these cases, β determines a single point on the information curve with specified encoder $P_\beta(Z|X)$ and decoder $P_\beta(Y|Z)$.

Information Bottleneck Bound

(write)

3 Deep Neural Networks

Network As Markov Chain

Any representation Z defined as a (possibly stochastic) map of input X is characterized by its joint distribution $P(X, Y)$, or, by it's **encoder** and **decoder** distributions, $P(Z|X)$ and $P(Y|Z)$, respectively.

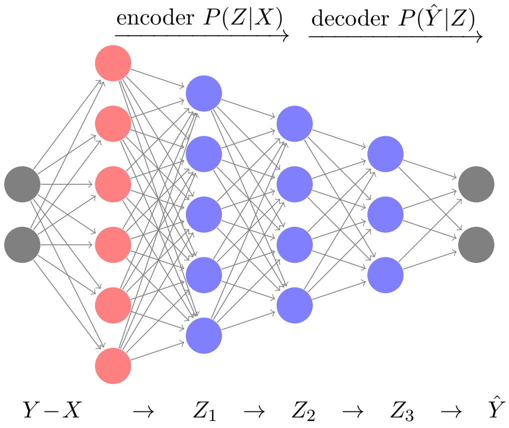

Figure 1: The layers form a Markov chain of successive internal representations.

The k layers of a DNN form a Markov chain $Y \rightarrow X \rightarrow Z_1 \rightarrow \ldots \rightarrow Z_k \rightarrow \hat{Y}$ of successive representations $\{Z_i : i = 1, \ldots, k\}$. Each representation Z_i of the i-th hidden layer is a single multivariate random variable. The Markov chain satisfies the DPI (8):

$$I(X;Y) \geq I(Z_1;Y) \geq I(Z_2;Y) \geq \ldots \geq I(Z_k;Y) \geq I(\hat{Y};Y), \tag{16}$$

$$H(X) \geq I(X;Z_1) \geq I(X;Z_2) \geq \ldots \geq I(X;Z_k) \geq I(X;\hat{Y}). \tag{17}$$

A DNN is designed to learn how to describe X to predict Y and eventually, to compress X to only hold the information related to Y. Tishby describes this processing as “successive refinement of relevant information”.

Information Plane

Given $P(X;Y)$, Z is uniquely mapped to a point in the **information plane** with coordinates $(I(X;Z), I(Z,Y))$. Since layers related by invertible re-parametrization (9) appear in the same point, each information path in the plane corresponds to many different networks, with possibly very different architectures. The layers are mapped to k monotonic connected points in the plane. This unique **information path** satisfies the DPI chains.

The decoder mutual information $I(Z;Y)$ corresponds to the **generalization error** or out of sample error. The encoder mutual information $I(X;Z)$ corresponds to the **sample complexity** or number of samples required achieve some accuracy.

Network and Data

We train a deep neural network with fully-connected topology of layer sizes 12-10-8-6-4-2-1 neurons and with hyperbolic tangent function, shifted to a sigmoidal function in the final layer. The networks were trained using SGD and the cross-entropy loss function, with no other explicit regularization. The data is **12 binary inputs** that represent 12 uniformly distributed points on a 2D sphere. With such rules, the 4,096 different patterns of the input variable X are divided into 64 disjoint orbits of the rotation group. These orbits form a minimal sufficient partition/statistics for spherically symmetric rules.

Computing Mutual Information

Estimating the mutual information of the layers in a Neural Network is done by estimating the probability density from a finite number of samples. Assume that we have a number N of samples in the training set and the probability density $P(X)$, $P(Z|X)$ and $P(X,Y)$ are approximated by counting the number of cases with the values of the variables belonging to a **discretized interval**.

Findings

We define a deep neural network (DNN) as a Markov chain and study the information paths of its hidden layers in the information plane spanned by the two order parameters $I(Z;X)$ and $I(Z;Y)$. This is feasible if $P(X,Y)$ is known and $P(Z|X)$ and $P(Y|Z)$ are tractable (or can be estimated). The main findings are:

1. **Dynamics of SGD:** We observe a phase transition from fast drift (ERM) to random diffusion or stochastic relaxation (compression) constrained by the training error when the training errors becomes small. Most of the epochs are spent on compression, not fitting the labels.

2. **Optimal representations:** The converged layers lie very close to the Information Bottleneck theoretical bound. This generalization through noise mechanism is unique to deep networks and absent in one layer networks.

3. **Benefits of the hidden layers:** The training time is dramatically reduced when adding more hidden layers. Thus the main advantage of the hidden layers is computational. This can be explained by the reduced relaxation time, as this it scales super-linearly (exponentially for simple diffusion) with the information compression from the previous layer.

4. **Benefits of sample size:** With increased sample size, the decoder mutual information is pushed up and gets closer to the theoretical information bottleneck bound. It is the mutual information, not the layer size or VC dimension, that determines generalization, different from standard theories.

Finding 1: Dynamics of Stochastic Gradient-Descent

When depicting the dynamics of Stochastic Gradient-Descent (SGD) on the information plane over epochs, we observe a phase transition between a short drift (i.e. ERM) and a long random diffusion (i.e. compression) phase:

1. **Drift Phase:** The layers increase the information on the labels, $I(Z;Y)$, while preserving the DPI order (lower layers have higher information), i.e., ERM.
2. **Diffusion Phase:** The layers information on the input, $I(X;Z)$, decreases and the layers lose irrelevant information until convergence, i.e., compression.

First, in the drift phase, the decoder mutual information $I(Z;Y)$ increases which results in a decreased generalization error (accuracy increases). Second, in the diffusion phase, the encoder mutual information $I(X;Z)$ decreases as the hidden layers compress the information and "forget" some information. Tishby believes that **"the most important part of learning is actually forgetting"**.

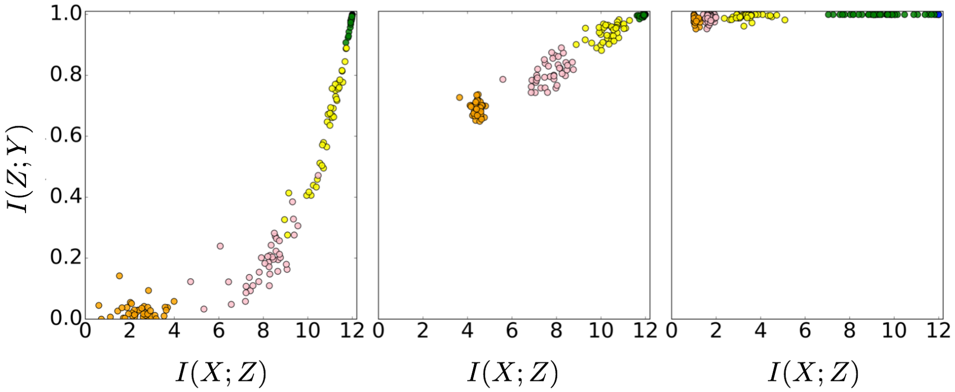

Figure 2: Three snapshots the encoder $I(X;Z)$ and decoder $I(Z;Y)$ mutual information of the layers of 50 randomized networks during the SGD optimization process in the information plane (in bits): **left:** 1 epoch, **center:** 400 epochs, **right:** 9,000 epochs. Layers in green being the closest to input X, layers in orange being the furthest away.

We can observe the phenomena in the behavior of the weight gradients ∇W_i. In the drift phase $I(Z;Y)$ increases quickly as the empirical error decreases. The gradient means are much smaller than their standard deviations. In the diffusion phase, the gradients means are very small compared to their batch to batch fluctuations.

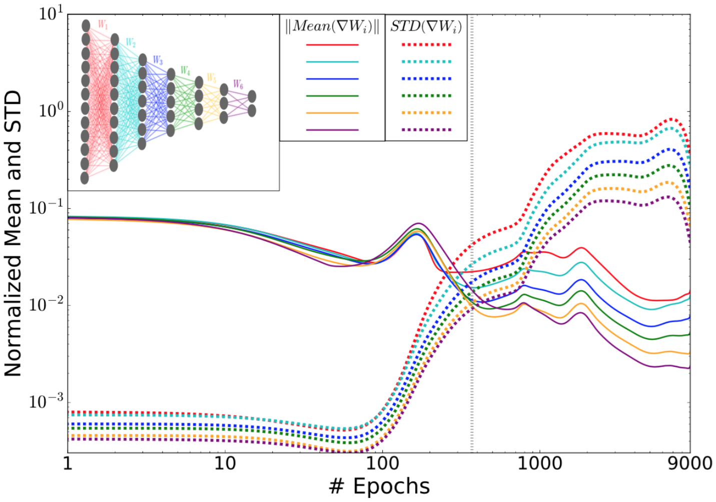

Figure 3: Mean $\|Mean(\nabla W_i)\|_2$ (filled line) and standard deviation $STD(\nabla W_i)$ (dotted line) of stochastic gradient ∇W_i. The grey line ($\sim$ 350 epochs) marks the transition between the first phase, with large gradient means and small variance (drift, high gradient SNR), and the second phase, with large fluctuations and small means (diffusion, low SNR).

While the increase of $I(Z;Y)$ in the ERM phase is expected from the cross-entropy loss minimization, the surprising compression phase requires an explanation. There was no explicit regularization that could simplify the representations. The diffusion phase mostly adds random noise to the weights, and they evolve like Wiener processes, under the constraint of the training error. Such diffusion processes can be described by the Fokker-Planck equation, whose stationary distribution maximizes the entropy of the weights distribution, under the training error constraint. This maximizes $H(X|Z_i)$, or minimizes $I(X;Z_i) = H(X) - H(X|Z_i)$, because the input entropy $H(X)$ remains constant. This entropy maximization by additive noise, also known as **stochastic relaxation**, is constrained by the empirical error. By minimizing $I(X;Z_i)$ for each layer i, the diffusion phase leads to more compressed representations.

From Stochastic Gradient-Descent to Ornstein-Uhlenbeck Process

Consider an objective function of the form $\mathcal{L}(\theta) = \sum_{n=1}^{N} l_n(\theta)$. Let $S(t)$ be a set of indices drawn uniformly at random from set $\{1, 2, \ldots, N\}$. We can form a stochastic estimate of the objective and a stochastic gradient,

$$\hat{L}(\theta) = \frac{N}{S} \sum_{n \in S} l_n(\theta), \tag{18}$$

$$\hat{g}_s(\theta) = \nabla_\theta \hat{L}(\theta). \tag{19}$$

In expectation, the stochastic gradient is the full gradient, $g(\theta) = \mathbb{E}(\hat{g}_s(\theta))$. The stochastic gradient is used for the update:

$$\theta(t+1) = \theta(t) - \hat{\epsilon}\hat{g}_S(\theta(t)). \tag{20}$$

To approximate SGD with a continuous time process, some assumptions are made:

1. Assume that the gradient noise $\hat{g}_S(\theta) - g(\theta)$ is Gaussian distributed.

2. Assume that the iterates $\theta(t)$ are constrained to a small enough region in parameter space that the sampling noise covariance of the stochastic gradients is constant.

3. Assume that the step size is small enough so that we can approximate the discrete-time Markov chain defined by SGD with a continuous-time Markov process.

Based on assumption 1, if S is big enough, then the central limit theorem should apply and we can write the stochastic gradient as

$$\hat{g}_S(\theta) \approx g(\theta) + \hat{\xi}_S(\theta), \quad \hat{\xi}_S(\theta) \sim \mathcal{N}(0, C(\theta)/S). \tag{21}$$

Decompose the covariance matrix C into $C = BB^T$, we introduce a rescaled noise covariance matrix $B_{\epsilon/S} = \sqrt{\epsilon/S}B$. And according to assumption 2 we can relax B to a constant irrelevant to θ. Then

$$\theta(t+1) - \theta(t) = -\epsilon g(\theta(t) + \sqrt{\epsilon} B_{\epsilon/S} W(t), \quad W(t) \sim \mathcal{N}(0, I). \tag{22}$$

In the continuous version is

$$d\theta(t) = -\nabla_\theta \mathcal{L}(\theta) dt + B_{\epsilon/S} dW(t). \tag{23}$$

As we want $\theta(t)$ to be close to the stationary point when $d\theta(t)$ is small, where the noise dominates the gradient. So we need a further assumption on the gradient, that is

4. Assume that the stationary distribution of the iterates is constrained to a region within which the objective is well approximated by a quadratic function.

Now we can write the $\nabla_\theta \mathcal{L}(\theta)$ explicitly as

$$d\theta(t) = -A(\theta(t) - \hat{\theta}) dt + B_{\epsilon/S} dW(t). \tag{24}$$

The solution of SDE (24) is identified as a multidimensional Ornstein-Uhlenbeck process. The probability density function of the Ornstein-Uhlenbeck process satisfies the Fokker-Planck equation.

Finding 2: Optimal Representation

(write)

Finding 3: Benefits of the Hidden Layers

The hidden layers can be understood of a computational means reducing the stochastic relaxation phase.

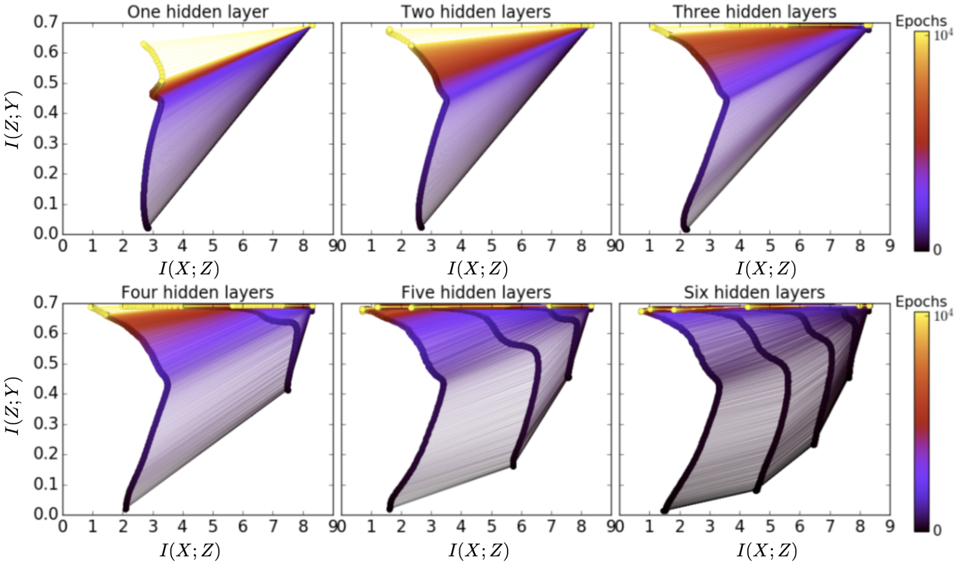

Figure 4: Hidden layers shorten the training time in the diffusion phase.

In one spatial dimension x, for an Ito process driven by the standard Wiener process W_t and described by the SDE

$$dX_t = \mu(X_t, t)\, dt + \sigma(X_t, t)\, dW_t \tag{25}$$

with drift velocity $\mu(X_t, t)$ and diffusion coefficient $D(X_t, t) = \sigma^2(X_t, t)/2$, the Fokker-Planck equation for the probability density $p(x, t)$ of the random variable X_t is

$$\frac{\partial}{\partial t} p(x,t) = -\frac{\partial}{\partial x}\left[\mu(x,t)p(x,t)\right] + \frac{\partial}{\partial x} D(x,t) \frac{\partial}{\partial x} p(x,t). \tag{26}$$

Then, according to the diffusion equation (26), the relaxation time of layer k is proportional to the exponential of this layer's compression amount $\Delta S_k : \Delta t_k \sim \exp(\Delta S_k)$. We can compute the layer compression as $\Delta S_k = I(X; Z_k) - I(X; Z_{k-1})$. Because $\exp(\sum_k \Delta S_k) \geq \sum_k \exp(\Delta S_k)$, we would expect an **exponential decrease in the training epochs** with increasing number of hidden layers k.

Finding 4: Benefit of Sample Size

Fitting more training data requires more information captured by the hidden layers. With increased training data size, the decoder mutual information (recall that this is directly related to the generalization error), $I(Z;Y)$, is pushed up and gets closer to the theoretical information bottleneck bound.

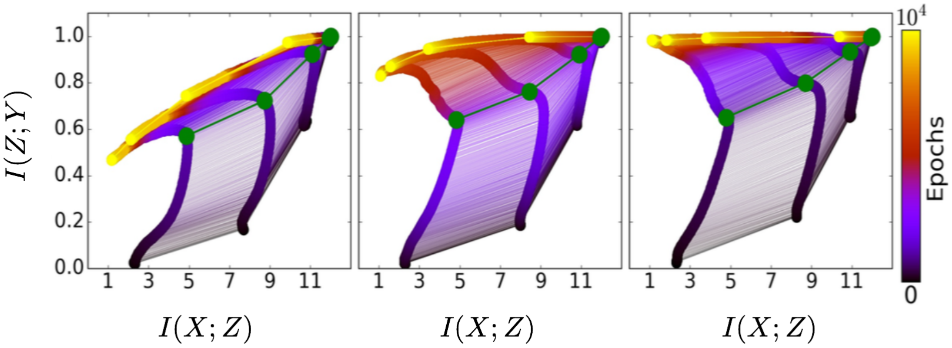

Figure 5: The green line indicates SGD drift-diffusion phase transition with respect to sample size: **left:** 5% data, **center:** 45% data epochs, **right:** 85% data.

Tishby emphasized that it is the **mutual information, not the layer size or the VC dimension, that determines generalization**, different from standard theories.

4 Learning Theory

"Old" Generalization Bounds

The generalization bounds defined by classic learning theory is:

$$\epsilon^2 < \frac{\log |H_\epsilon| + \log 1/\delta}{2m} \tag{27}$$

where

- ϵ: The difference between training and generalization error.
- H_ϵ: ϵ-cover of the hypothesis class, typically, we assume $|H_\epsilon| \sim (1/\epsilon)^d$.
- δ: Confidence.
- m: The number of training samples.
- d: The VC dimension of the hypothesis.

The bound (27) states that the difference between the training error and the generalization error is bounded by a function of the hypothesis space size and the dataset size. The larger the hypothesis space gets, the bigger the generalization error becomes.

However, it does not work for deep learning. The larger a network is, the more parameters it needs to learn. With this generalization bounds, **larger networks (larger d) would have worse bounds**. This is contrary to the intuition that larger networks are able to achieve better performance with higher expressivity.

"New" Input Compression Bounds

To solve this counterintuitive observation, Tishby et al. proposed a new input compression bound for DNN.

First, let us have Z_ϵ as an ϵ-partition of the input variable X. This partition compresses the input with respect to the homogeneity of the labels into small cells. The cells in total can cover the whole input space. If the prediction outputs binary values, we can replace the cardinality of the hypothesis, $|H_\epsilon|$, with $2^{|Z_\epsilon|}$.

$$|H_\epsilon| \sim 2^{|X|} \to 2^{|Z_\epsilon|}. \tag{28}$$

When X is large, the size of X is approximately $2^{H(X)}$. Each cell in the ϵ-partition is of size $2^{H(X|Z_\epsilon)}$. Therefore, we have $|Z_\epsilon| \sim \frac{2^{H(X)}}{2^{H(X|Z_\epsilon)}} = 2^{I(Z_\epsilon;X)}$. Then, the input compression bound becomes:

$$\epsilon^2 < \frac{2^{I(Z_\epsilon;X)} + \log 1/\delta}{2m}. \tag{29}$$

5 Variational Bottleneck

Variational Formulation

(write)

From Information Bottleneck to Variational Autoencoder

The information bottleneck objective is finding the statistic $Z(X)$ that is sufficient $I(X;Y) = I(Z;Y)$ with minimal $I(X;Z)$:

$$\begin{aligned} \min \quad & I(X;Z) && (30)\\ s.t. \quad & I(Z;Y) = I(X;Y). && (31) \end{aligned}$$

We relax the formulation into the information bottleneck lagrangian:

$$\mathcal{L} = I(X;Z) - \beta I(Z;Y), \quad \beta > 0. \tag{32}$$

Let $P_\theta(Z|X)$ and $P_\theta(Y|Z)$ denote variational approximations parameterized by θ. We can rewrite the two terms of (32) as:

$$I(Z;Y) \simeq \mathbb{E}_{x,y\sim P(X,Y)}\left[\mathbb{E}_{z\sim P_\theta(Z|X)}[-\log p_\theta(y|z)]\right], \tag{33}$$

$$I(X;Z) = \mathbb{E}_{x\sim P(X)}\left[D_{KL}(p_\theta(z|x)\|p_\theta(z))\right]. \tag{34}$$

Given training set $\{(x_i, y_i) : i = 1\ldots, N\} \sim P(X,Y)$, we approximate (32) empirically as

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{z\sim p_\theta(Z|X=x_i)}[-\log p_\theta(y_i|z)] + \beta D_{KL}(p_\theta(z|x_i)\|p_\theta(z)). \tag{35}$$

In an unsupervised regime, the task is reconstruction $y = x$ and (35) reduces to:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{z\sim p_\theta(Z|X=x_i)}[-\log p_\theta(x_i|z)] + \beta D_{KL}(p_\theta(z|x_i)\|p_\theta(z)). \tag{36}$$

In fact, for $\beta = 1$ we obtain the objective of the variational auto-encoder (VAE). For arbitrary β, this is precisely the objective formulation the β-VAE .

Proof. (rewrite) Consider the information bottleneck objective in an unsupervised setting:

$$\max I(Z;X) - \beta I(Z;i). \tag{37}$$

Then, bound the first term

$$\begin{aligned}
I(Z;X) &= \iint dx\,dz\,p(x,z)\log p(x|z)/p(x) && (38)\\
&= H(x) + \iint dx\,dz\,p(x,z)\log p(x|z) && (39)\\
&= H(x) + \int dz\,p(z)\int dx\,p(x|z)\log p(x|z) && (40)\\
&\geq \int dz\,p(z)\int dx\,p(x|z)\log q(x|z) && (41)\\
&= \iint dx\,dz\,p(x,z)\log q(x|z) && (42)\\
&= \int dx\,p(x)\int dz\,p(z|x)\log q(x|z). && (43)
\end{aligned}$$

where in (41) we dropped $H(X)$ and used the non-negativity of the Kullback-Leibler divergence $D_{KL}(p(x|z)\|q(x|z)) \geq 0$ to substitute intractable $p(x|z)$ by variational decoder $q(x|z)$. For the second term, note:

$$p(z|i) = \int dx\,p(z|x)p(x|i) = \int dx\,p(z|x)\delta(x - x_i) = p(z|x_i), \tag{44}$$

and take $p(i) = 1/N$. Then, bound the second term from above

$$\begin{aligned} I(Z, i) &= \sum_i \int dz\, p(z|i)p(i) \log \frac{p(z|i)}{p(z)} && (45) \\ &= \frac{1}{N} \sum_i \int dz\, p(z|x_i) \log \frac{p(z|x_i)}{p(z)} && (46) \\ &\leq \frac{1}{N} \sum_i \int dz\, p(z|x_i) \log \frac{p(z|x_i)}{r(z)}, && (47) \end{aligned}$$

where in (47) we replace the intractable $p(z)$ by variational marginal $r(z)$. When applying both bounds to (37) the objective takes the form

$$I(Z;X) - \beta I(Z;i) \leq \int dx\, p(x) \int dz\, p(z|x) \log q(x|z) - \beta \frac{1}{N} \sum_i D_{KL}(p(Z|x_i) \| r(Z)), \tag{48}$$

□