The Information Theoretic Approach Taken to Investigate Neural Network Generalization

Thomas Walker

Summer 2023

Contents

1 Introduction

Training a neural network is essentially transferring information from the training data into the weights of the network. The goal is to effectively represent the training data such that it can be utilized to make inferences in a different setting. However, there exist multiple representations of information, some of which are richer than others. Analysing networks from an information theoretic perspective provides insight into the power of the learned representations beyond their performance on a singular metric, the loss. The framework allows the tracking of information transfer from the input data to the outputs of the network and can indicate different stages of the learning process. Information-theoretic approaches are inherently related to the data inputted into the network and hence can provide more robust metrics for bounding the generalization error of the network.

1.1 Notation

We will first introduce some basic notation that will remain constant throughout the report. Along the way, we will need to introduce some more specialized notation for the different sections. We are going to consider problems over feature space $\mathcal{X}$ and a label space $\mathcal{Y}$ which combine to form the data space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ for which some unknown $\mathcal{D}$ is defined on. The challenge is to train a network $h : \mathcal{X} \to \mathcal{Y}$ that correctly labels samples from $\mathcal{X}$ according to $\mathcal{D}$. The training data $S = \{(x_i, y_i)\}_{i=1}^m$ consists of m $\mathrm{i.i.d}$ samples from $\mathcal{D}$. As we are considering neural networks, a model will be parameterized by a weight vector $\mathbf{w}$ which we will denote $h_{\mathbf{w}}$. Let $\mathcal{W}$ denote the set of possible weights for a model and the set of all possible models $\mathcal{H}$ will sometimes be referred to as the hypothesis set. To assess the quality of a model we define a measurable function $l : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ called the loss function and we will assume that $0 \leq l \leq C$. As our training data is just a sample from the underlying (unknown) distribution $\mathcal{D}$ there is the possibility that our model performs well on the training data, but performs poorly on the true distribution. Let the risk of our model be defined as

$$R(h_{\mathbf{w}}) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left(l(h(x), y) \right).$$

As our model is parameterized $\mathbf{w}$ we will instead write $R(\mathbf{w})$ for the risk of our classifier. Similarly, we define the empirical risk of our model to be

$$\hat{R}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} l(h_{\mathbf{w}}(x_i), y_i).$$

Note that $\mathbb{E}_{S\sim\mathcal{D}^m} \left(\hat{R}(\mathbf{w}) \right) = R(\mathbf{w})$.

2 VC Dimension

It is generally accepted that having a more straightforward function that correctly classifies a dataset is more likely to generalize well to unseen data. The generalization bounds we will discuss will often reward simpler models, where the definition of simple may vary in different contexts. A lot of work in this field tends to be empirical as heuristics are derived for complexity that can be monitored during training. Sometimes theoretically motivated complexity measures are proposed that can be used to derive explicit generalization bounds. An important complexity measure is Rademacher complexity, which forms the basis of one of the main results in generalization theory. Suppose we have a hypothesis class $\mathcal{H}$ (i.e. the possible neural networks defined by a particular set of hyper-parameters), and a training set S. For a loss function l, let $l \circ \mathcal{H} := \{l \circ h : h \in \mathcal{H}\}$.

Definition 2.1 ([6]). *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{Z}$ to $[a, b]$ and $S = (z_1, \ldots, z_m)$ a fixed sample of size m with elements in $\mathcal{Z}$. Then, the empirical Rademacher complexity of $\mathcal{G}$ with respect to the sample S is defined as:*

$$\mathfrak{R}_S(\mathcal{G}) = \frac{1}{m} \mathbb{E}_{\xi\in\{\pm1\}^m} \left(\sup_{f\in\mathcal{G}} \sum_{i=1}^{m} \xi_i f(x_i) \right).$$

Definition 2.2 ([6]). *Let $\mathcal{D}$ be a distribution from which samples are drawn. For any integer $m \geq 1$, the Rademacher complexity of a family of functions $\mathcal{G}$ is the expectation of the empirical Rademacher complexity over all samples of size m drawn from $\mathcal{D}$. That is,*

$$\mathfrak{R}_m(\mathcal{G}) = \mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{\mathfrak{R}}_S(\mathcal{G})\right).$$

Theorem 2.3 ([6]). *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{Z}$ to $[0,1]$. Then for any $\delta > 0$ with probability $1-\delta$ over the draw of an* i.i.d *sample S of size m then,*

$$\mathbb{E}(g(z)) \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\mathfrak{R}_m(\mathcal{G}) + \sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}, \text{ and}$$

$$\mathbb{E}(g(z)) \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\hat{\mathfrak{R}}_m(\mathcal{G}) + 3\sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2m}}.$$

holds for all $g \in \mathcal{G}$.

Rademacher complexity is a complexity measure that yields theoretical bounds on the expected error of the function of our network. By understanding our hypothesis class can separate the data points of our training set. Subsequent work aims to develop a tractable heuristic for this measure. Refer to [9] to see how Rademacher complexity can be bounded by VC dimension. The setup is as follows, we are performing binary classification on input data $\mathcal{X} \subseteq \mathbb{R}^d$ and labels $\mathcal{Y} = \{\pm 1\}$. With $\mathcal{A} \subseteq \mathcal{B} = \{a : \mathcal{X} \to \{\pm 1\}\}$ let

$$\mathcal{A} \circ \{x_1, \ldots, x_n\} = \{(a(x_1), \ldots, a(x_n)) \in \{\pm 1\}^n : a \in \mathcal{A}\}.$$

Note that $\mathcal{A} \circ x$ is finite even if $\mathcal{A}$ is infinite.

Definition 2.4. *The growth function of $\mathcal{A}$ is defined as*

$$\tau_{\mathcal{A}}(n) := \sup_{x\in\mathcal{X}} |\mathcal{A} \circ x|$$

for any integer $n \geq 1$.

That is, $\tau_{\mathcal{A}}$ is the maximal cardinality of the set of distinct labelling of n points in $\mathcal{X}$ obtained using classifiers from $\mathcal{A}$.

Proposition 2.5. *For any $x = \{x_1, \ldots, x_n\} \in \mathcal{X}^n$ we have*

$$\mathfrak{R}(\mathcal{A} \circ x) \leq \sqrt{\frac{2\log(\tau_{\mathcal{A}}(n))}{n}}.$$

Note that $\tau_{\mathcal{A}}(n) \leq 2^n$.

Definition 2.6. *The Vapnik-Chervonenkis (VC) dimension of $\mathcal{A}$ is the largest integer n such that $\tau_{\mathcal{A}}(n) = 2^n$,*

$$\mathrm{VC}(\mathcal{A}) := \max\{n \in \mathbb{N} : \tau_{\mathcal{A}}(n) = 2^n\}$$

The quantities $\tau_{\mathcal{A}}(1), \tau_{\mathcal{A}}(2), \ldots$ are known as shatter coefficients, and we say that $\mathcal{A}$ shatters $\{x_1, \ldots, x_n\}$ if $|\mathcal{A} \circ \{x_1, \ldots, x_n\}| = 2^n$. Hence, VC dimension is the maximum number of different elements that can be shattered by $\mathcal{A}$.

Proposition 2.7. *For any $x = \{x_1, \ldots, x_n\} \in \mathcal{X}^n$ we have*

$$\mathfrak{R}(\mathcal{A} \circ x) \leq \sqrt{\frac{2\mathrm{VC}(\mathcal{A})\log\left(\frac{en}{\mathrm{VC}(\mathcal{A})}\right)}{n}}.$$

This is a data-independent bound that holds for any $x \in \mathcal{X}^n$. It can be improved through a technique known as chaining.

3 Controlling Bias in Data Analysis

One of the first efforts to contextualize learning bias in information theory was done in [2]. The main result of [2] is an information-theoretic bound on the bias of adaptively choosing a function from data. A dataset S is drawn from a probability distribution $\mathcal{D}$ defined over $\mathcal{Z}$. There is a set of analyses

$$\phi_1, \ldots, \phi_m : \mathcal{Z} \to \mathbb{R}$$

that can be run on the data. Each ϕ_i is a random variable dependent on the realization $S \sim \mathcal{D}$. When the realization is made a particular $\phi_T(S)$ is reported for $T \in [m]$. The selection rule $T : \mathcal{Z} \to [m]$ determines how the realization relates to the reported result. Bias may be present in the selection rule as it is a function of the realization which is itself a proxy of the underlying distribution $\mathcal{D}$. Let

$$\Phi = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_m \end{pmatrix} : \Omega \to \mathbb{R}^m, \; T : \Omega \to [m]$$

and

$$\mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_m \end{pmatrix} := \mathbb{E}(\Phi).$$

Then the bias of the learning process will be captured by the quantity $\mathbb{E}(\phi_T - \mu_T)$.

Definition 3.1. *A real-valued random variable X is σ-sub-Gaussian if for all $\lambda \in \mathbb{R}$*

$$\mathbb{E}\left(e^{\lambda X}\right) \leq e^{\frac{\lambda^2 \sigma^2}{2}}.$$

Definition 3.2 ([3])**.** *For two random variables X and Y, with joint distribution $p(x, y)$, their Mutual Information is defined as,*

$$\begin{aligned} I(X;Y) = \mathrm{KL}(p(x,y), p(x)p(y)) &= \sum_{x \in X, y \in Y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) \\ &= H(X) - H(X|Y), \end{aligned}$$

where $H(X)$ and $H(X|Y)$ are the entropy and conditional entropy of X and Y.

Mutual information quantifies the number of relevant bits that the input variable X contains about the label Y on average.

Theorem 3.3. *Suppose that for each $i \in [m]$, $\phi_i - \mu_i$ is σ-sub-Gaussian. Then,*

$$|\mathbb{E}(\phi_T) - \mathbb{E}(\mu_T)| \leq \sigma\sqrt{2I(T;\Phi)},$$

Remark 3.4.

- $\mathbb{E}(\phi_T)$ *is taken jointly over the realized values of the ϕ_i and T.*
- $\mathbb{E}(\mu_T)$ *is taken over the selection procedure T.*
- $|\mathbb{E}(\phi_T) - \mathbb{E}(\mu_T)|$ *quantifies the bias due to T.*
- $I(T;\phi)$ *quantifies the dependence of the selection process on the noise in the test statistics.*

Proposition 3.5. *Let $\Phi = (\phi_1 \; \phi_2 \; \ldots)^\top$ be a collection of independent normally distributed random variables with mean 0 and variance σ^2. For $B > 1$, let $T_B = \mathrm{argmax}_{1 \leq i \leq \lfloor e^B \rfloor} \phi_i$. Then, $I(T_B; \Phi) \leq B$ and*

$$\mathbb{E}(\Phi_{T_B}) - \sigma\sqrt{2B} \to 0$$

as $B \to \infty$. Furthermore, there exists a $c > 0$ such that

$$\mathbb{E}(\phi_{T_B}) \geq c\sigma\sqrt{2B}, \quad \forall B \geq 2.$$

The above are bounds on the bias involving specific assumptions on the distributions and the selection procedure. The next section considers [1] which shows how some of these assumptions can be relaxed.

4 Generalizing the Framework

5 Generalizing the Framework

In [1] Theorem 3.3 is extended to distributions with non-trivial moment generating functions. Furthermore, a new measure is introduced which generalizes mutual information.

Definition 5.1. *The cumulant generating function of a random variable X is*

$$\psi(\lambda) = \log\left(\mathbb{E}\left(e^{\lambda x}\right)\right), \quad \lambda \geq 0.$$

In what follows assume that for all considered random variables there exists a $\lambda > 0$ such that $\mathbb{E}\left(e^{\lambda X}\right) < \infty$.

Definition 5.2. *A random variable X is sub-Exponential with parameters (σ, b) if*

$$\mathbb{E}\left(e^{\lambda X}\right) \leq e^{\frac{\lambda^2\sigma^2}{2}}, \quad 0 \leq \lambda < \frac{1}{b}.$$

Definition 5.3. *A random variable X is sub-Gamma on the right tail with variance factor σ^2 and scale parameter c if*

$$\psi(\lambda) \leq \frac{\lambda^2\sigma^2}{2(1-c\lambda)}, \quad 0 < \lambda < \frac{1}{c}.$$

Definition 5.4. *The β-norm of a random variable X for $\beta \geq 1$ is*

$$\|X\|_\beta = \begin{cases} \left(\mathbb{E}\left(|X|^\beta\right)\right)^{\frac{1}{\beta}} & 1 \leq \beta < \infty \\ \operatorname{ess\,sup}|X| & \beta = \infty, \end{cases}$$

where $\operatorname{ess\,sup} = \inf\{M : \mathbb{P}(X > M) = 0\}$.

Proposition 5.5. *For any function f, lets its convex conjugate, f^*, be defined as*

$$f^*(y) = \sup_{x \in X}(\langle x, y\rangle - f(x)).$$

Definition 5.6. *For $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}$ a convex lower semi-continuous function satisfying $\phi(1) = 0$, the ϕ-divergence of two probability distributions is*

$$D_\phi(Q, P) = \int \phi\left(\frac{dQ}{dP}\right) dP.$$

Remark 5.7. *For $\phi(x) = x\log(x) - x + 1$, it follows that $D_\phi(Q, P) = \mathrm{KL}(Q, P)$.*

For non-negative sequences $\{a_n\}$ and $\{b_n\}$ let $a_n \lesssim b_n$ if there is a constant $C > 0$ such that $\limsup_n \frac{a_n}{b_n} \leq C$.

Theorem 5.8. *Suppose that $\phi_i - \mu_i$ has cumulant generating function upper bounded by $\psi_i(\lambda)$ over domain $[0, b_i)$ where $b_i \in (0, \infty]$. Suppose $\psi_i(\lambda)$ is convex with $\psi_i(0) = \psi_i'(0) = 0$. Define the expected cumulant generating function $\bar{\psi}(\lambda)$ as*

$$\bar{\psi}(\lambda) = \mathbb{E}_T\left(\psi_T(\lambda)\right), \quad \lambda \in \left[0, \min_i b_i\right).$$

Then,

$$\mathbb{E}(\phi_T - \mu_T) \leq \left(\bar{\psi}^*\right)^{-1} I(T; \Phi).$$

Definition 5.9. *For $1 \leq \alpha < \infty$ let*

$$I_\alpha(X; Y) = D_{\phi_\alpha}(P_{XY}, P_X P_Y)$$

where $\phi_\alpha(x) = |x-1|^\alpha$.

Remark 5.10.
- $I_\alpha(X;Y) \geq 0$, *and*
- $I_\alpha(X;Y) = 0$ *if and only if* X *ad* Y *are independent.*

Theorem 5.11. *Suppose* $\phi_i - \mu_i$ *has its* β*-norm upper bounded by* σ_i*, for* $1 < \beta \leq \infty$*. Let* α *be such that* $\frac{1}{\alpha} + \frac{1}{\beta} = 1$*. Then,*

$$|\mathbb{E}(\phi_T - \mu_T)| \leq \|\sigma_T\|_\beta I_\alpha(T;\Phi)^{\frac{1}{\alpha}}.$$

For $\beta = 2$ *we have*

$$|\mathbb{E}(\phi_T - \mu_T)| \leq \|\sigma_T\|_2 \sqrt{n-1},$$

and for $2 < \beta \leq \infty$ *and* $1 \leq \alpha < 2$ *we have*

$$|\mathbb{E}(\phi_T - \mu_T)| \leq \|\sigma_T\|_\beta \left(1 + n^{\alpha-1}\right)^{\frac{1}{\alpha}} \leq 2^{\frac{1}{\alpha}} \|\sigma_T\|_\beta n^{\frac{1}{\beta}}.$$

Corollary 5.12. *Suppose* $\phi_i - \mu_i$ *is* σ_i*-sub-Gaussian. Then,*

$$\mathbb{E}(\phi_T - \mu_T) \leq \|\sigma_T\|_2 \sqrt{2I(T;\Phi)}.$$

Corollary 5.13. *Suppose* $\phi_i - \mu_i$ *are sub-Gamma random variables with parameters* $\left(\sigma^2, c\right)$*. Then,*

$$\mathbb{E}(\phi_T - \mu_T) \leq \sigma\sqrt{2I(T;\Phi)} + cI(T;\Phi).$$

Corollary 5.14. *Suppose* $\phi_i - \mu_i$ *are sub-Exponential random variables with parameters* (σ, b)*. Then,*

$$\mathbb{E}_T(\phi_T - \mu_T) \leq \begin{cases} \sigma\sqrt{2I(T;\Phi)} & I(T;\Phi) \leq \frac{\sigma^2}{2b} \\ bI(T;\Phi) + \frac{\sigma^2}{2b^2} & \textit{otherwise.} \end{cases}$$

6 Evolution of Mutual Information

7 Evolution of Mutual Information

It has been shown that layered neural networks form a Markov chain, which suggests studying them from the perspective of mutual information. That is, consider each layer to be a single random variable and look at its mutual information with the input X and the output Y. The properties of neural network training are explored using this approach in [3]. In this work, the learning process is formulated as finding a good representation $T(X)$ of the input patterns $x \in X$ that generates good predictions for label $y \in Y$. Training is conducted through stochastic gradient descent (SGD), where at each step we aim to minimize the empirical error over the weights of the network. As this minimization occurs layer-by-layer the whole layer is treated as a single representation T which is characterized by the encoder $P(T|X)$ and the decoder $P(Y|T)$ distributions.

Theorem 7.1. *For any invertible functions ϕ and ψ,*

$$I(X;Y) = I(\psi(X), \phi(Y)).$$

Theorem 7.2. *For any three variables that form a Markov chain $X \to Y \to Z$,*

$$I(X;Y) \geq I(X,Z).$$

7.1 Information Plane

Given $P(X,Y)$, any representation T corresponds to a unique point in the information plane with coordinates $(I(X;T), I(T;Y))$. Consider a K-layered deep neural network, with T_i denoting the representation of the i^{th} layer then there is a unique information path which satisfies Theorem 7.2,

$$I(X;Y) \geq I(T_1;Y) \geq \cdots \geq I(T_k;Y) \geq I\left(\hat{Y};Y\right),$$
$$H(X) \geq I(X;T_1) \geq \cdots \geq I(X;T_k) \geq I\left(X;\hat{Y}\right).$$

Due to Theorem 7.1 it is possible that many different deep neural networks correspond to an information path.

7.2 Information Bottleneck

Definition 7.3. *Let X and Y be two random variables. A sufficient statistic $S(X)$ is map or partition of X that captures all the information that X has on Y,*

$$I(S(X);Y) = I(X,Y).$$

A minimal sufficient statistic, $T(X)$, induces the coarsest such partition on X. Consequently, one can form the Markov Chain

$$Y \to X \to S(X) \to T(X).$$

Using Theorem 7.2 finding T can be formulated as the optimization problem,

$$T(X) = \text{argmin}_{S(X);I(S(X);Y)=I(X;Y)} I(S(X);X).$$

The Information Bottleneck trade-off enables a framework for finding approximate minimal sufficient statistics. Let $t \in T$ be the compressed representation of $x \in X$, so that x is represented as $p(t|x)$. The Information Bottleneck trade-off is captured in the optimization problem

$$\min_{p(t|x),p(y|t),p(t)} (I(X;T) - \beta I(T;Y)).$$

Where β determines the level of relevant information captured by T. The solution to this problem is given by

$$\begin{cases} p(t|x) = \frac{p(t)}{Z(x;\beta)} \exp\left(-\beta \mathrm{KL}(p(y|x), p(y|t)\right) \\ p(t) = \sum_x p(t|x)p(x), \\ p(y|t) = \sum_x p(y|x)p(x|t), \end{cases} \tag{1}$$

where $Z(x;\beta)$ is the normalized function.

7.3 Student-Teacher Analysis

Work done by [7] implements this theory in different settings. In this first setting, there is a linear teacher network that generates training examples for a deep linear student network to learn. The teacher network has an input size of N_i and an output size of 1. The input is a multi-variate normal, $X \sim \mathcal{N}(0, \frac{1}{N_i} I_{N_i})$, and the weights of the network, W_o, are sampled independently from $\mathcal{N}(0, \sigma_2^2)$. The output for a given input is given by $Y = W_o X + \epsilon_o$ where $\epsilon_o \sim \mathcal{N}\left(0, \sigma_o^2\right)$. Take P samples from this teacher network to train a student network using gradient descent to minimize the mean squared error. The student network consists of an input layer, hidden layers and a single output neuron. The activation function on the hidden layer neurons is just the identity function. This setup can be represented as $\hat{Y} = W_{D+1} \dots W_1 X$ when the network has depth D. The activity of the i^{th} hidden layer is given by $T = \bar{W} X = W_i \dots W_1 X$. The true generalization error at training step t is given by

$$E_g(t) = \|W_o - W_{\text{tot}}(t)\|_F^2 + \sigma_o^2.$$

To calculate the mutual information some noise needs to be added otherwise, it would be infinite. Hence, let $T = \bar{W} X + \epsilon_{MI}$ for $\epsilon_{MI} \sim \mathcal{N}(0, \sigma_{MI}^2)$. With these assumptions, it follows that

$$I(T;X) = \log\left|\bar{W}\bar{W}^\top + \sigma_{MI}^2 I_{N_h}\right| - \log\left|\sigma_{MI}^2 I_{N_h}\right|,$$

where $|\cdot|$ denotes the determinant of a matrix and N_h is the number of hidden units in the layer. Similarly, the mutual information with the output Y can be calculated as

$$\begin{aligned} H(Y) &= \frac{N_o}{2}\log(2\pi e) + \frac{1}{2}\log\left|W_o W_o^\top + \sigma_o^2 I_{N_o}\right|, \\ H(T) &= \frac{N_h}{2}\log(2\pi e) + \frac{1}{2}\log\left|\bar{W}\bar{W}^\top + \sigma_{MI}^2 I_{N_h}\right|, \\ H(Y;T) &= \frac{N_o + N_h}{2}\log(2\pi e) + \frac{1}{2}\log\begin{vmatrix} \bar{W}\bar{W}^\top + \sigma_{MI}^2 I_{N_h} & \bar{W} W_o^\top \\ W_o \bar{W}^\top & W_o W_o^\top + \sigma_o^2 I_{N_o} \end{vmatrix}, \\ I(Y;T) &= H(Y) + H(T) - H(Y;T), \end{aligned}$$

where N_o is the size of the input. For the implementation, there will only be a single hidden layer so that $\bar{W} = W_1$, refer to Figure 1 for the results.

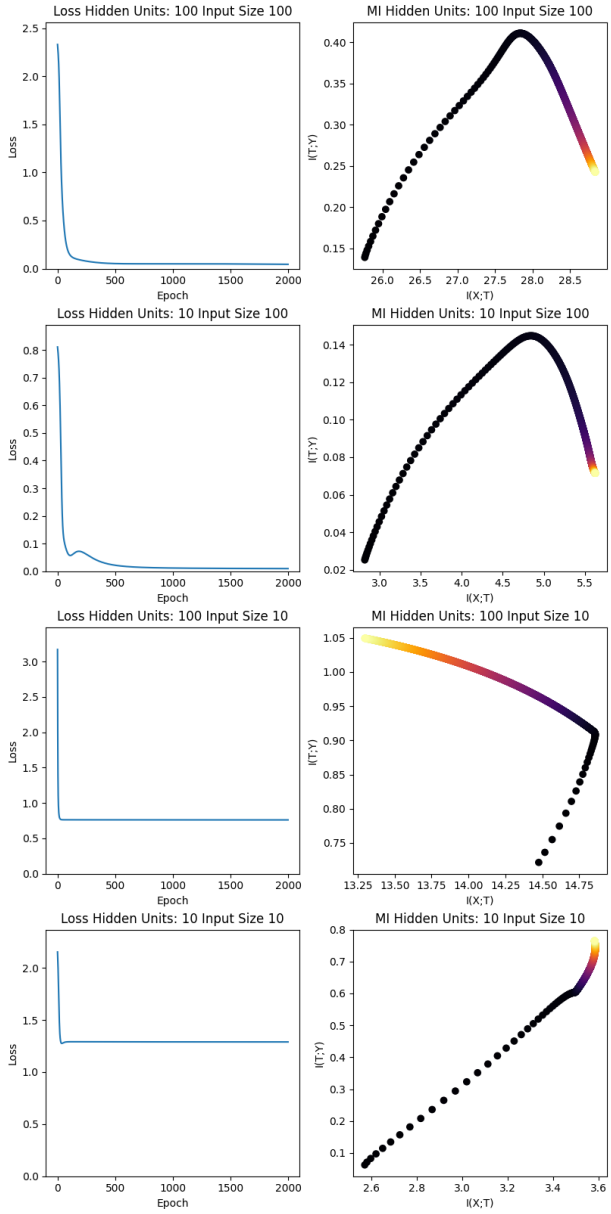

Figure 1: Explores the Mutual Information in the teacher-student scenario with different architectures and training data.

7.4 Non-Linear Setting via KDE

In another setting developed by [7] the mutual information for non-linear neural networks is investigated. This is done by appealing to Kernel Density Estimation. To apply KDE the hidden activity is assumed to be distributed as a mixture of Gaussians. As the layer activity is a deterministic function of the input noise needs to be added to get finite mutual information. Hence, we let $T = h + \epsilon$ where h is the activity of the

hidden layer and $\epsilon \sim \mathcal{N}\left(0, \sigma^2 I\right)$. Under these assumptions, it follows that

$$I(T;X) \leq -\frac{1}{P}\sum_i \log\left(\frac{1}{P}\sum_j \exp\left(-\frac{1}{2}\frac{\|h_i - h_j\|_2^2}{\sigma^2}\right)\right)$$

$$I(T;Y) \leq -\frac{1}{P}\sum_i \log\left(\frac{1}{P}\sum_j \exp\left(-\frac{1}{2}\frac{\|h_i - h_j\|_2^2}{\sigma^2}\right)\right) - \sum_{l=1}^{L} p_l \left(-\frac{1}{P_l}\sum_{Y_i=l} \log\left(\frac{1}{P_l}\sum_{Y_j=l} \exp\left(-\frac{1}{2}\frac{\|h_i - h_j\|_2^2}{\sigma^2}\right)\right)\right),$$

where

- P is the number of training samples,
- h_i is the hidden activity in response to sample i,
- L is the number of output labels,
- P_l is the number of samples with label l,
- $p_l = \frac{P_l}{P}$, and
- $Y_i = l$ is the sum over examples with output l.

See the results of this implementation in Figure 2. Where a network with layers $784 \to 1024 \to 20 \to 20 \to 20 \to 10$ is trained on 1000 images from the MNIST dataset. The activations of the third hidden layer are observed and used to estimate the mutual information between the inputs and outputs. The orange line corresponds to the entropy of a uniform distribution across the 1000 samples ($\log_2(1000)$).

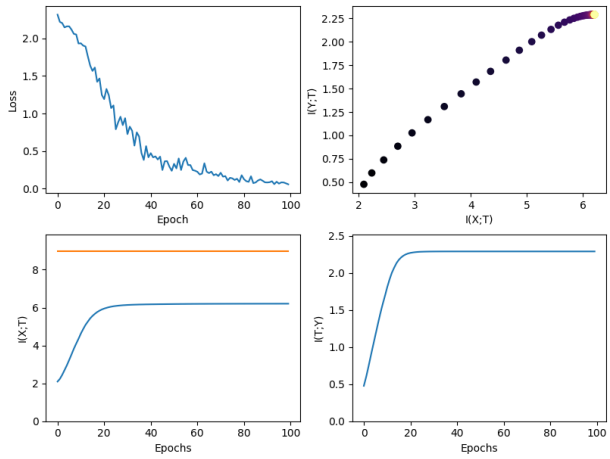

Figure 2: The Mutual Information of a ReLU neural network estimated using Kernel Density Estimation.

8 Generalization Bounds for Learning Algorithms

Returning to the framework set out by [2] and [1] we can translate it into the domain of learning algorithms to get generalization bounds using the work of [4]. Recall, that generalization error is the difference between the true risk of a model and its empirical risk on the training data. In learning problems, generalization error is equivalent to the bias in data analysis discussed previously. This is a promising approach to generating generalization bounds as mutual information is strongly dependent on the input dataset, and generalization error is also impacted by the input dataset. We can characterize the learning algorithm by a Markov kernel $P_{W|S}$. The learning algorithm is a random variable W on $\mathcal{W}$ with distribution $P_{W|S}$. For $\mathbf{w} \in \mathcal{W}$, recall that the true risk is

$$R(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}(l(h_{\mathbf{w}}(x), y)).$$

Ideally, the learning algorithm will be such that the excess risk

$$R(W) - \inf_{\mathbf{w}\in\mathcal{W}} R(\mathbf{w})$$

and its expectation

$$R_{\text{excess}}(P_{W|S})$$

are small. In practice we have to work with the empirical risk

$$\hat{R}(\mathbf{w}) := \frac{1}{m}\sum_{i=1}^{m} l(h_{\mathbf{w}}(x_i), y_i),$$

which is the implicitly dependent on the dataset S. Using the empirical is a potential error which is known as the generalization error. Let

$$\text{gen}(P_{W|S}) := \mathbb{E}_{S\sim\mathcal{D}^m, W\sim P_{W|S}}\left(R(W) - \hat{R}(W)\right),$$

then we can decompose the true risk as

$$\mathbb{E}_{W\sim P_{W|S}}(R(W)) = \mathbb{E}_{S\sim\mathcal{D}^m}\left(\hat{R}(W)\right) + \text{gen}(P_{W|S}).$$

Definition 8.1. *A learning algorithm is (ϵ, μ)-stable in input-output mutual information if, under the data-generating distribution $\mathcal{D}$ we have that,*

$$I(S; W) \leq \epsilon.$$

Definition 8.2. *A learning algorithm is ϵ-stable in input-output mutual information if*

$$\sup_{\mathcal{D}} I(S; W) \leq \epsilon.$$

The learning algorithm $P_{W|S}$ can be viewed as a channel from $\mathcal{Z}^n$ to $\mathcal{W}$ with $\sup_{\mathcal{D}} I(S; W)$ being the information capacity of the channel. Therefore, a learning algorithm is more stable if its information capacity is smaller. Now we proceed to try and bound generalization using $I(S; W)$. Consider a pair of random variables X and Y with joint distribution $P_{X,Y}$. Let $\bar{X}$ and $\bar{Y}$ be independent copies of X and Y respectively such that $P_{\bar{X},\bar{Y}} = P_X \otimes P_Y$.

Lemma 8.3. *If $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is such that $f\left(\bar{X}, \bar{Y}\right)$ is σ-sub-Gaussian under $P_{\bar{X},\bar{Y}}$, then*

$$\left|\mathbb{E}(f(X, Y)) - \mathbb{E}\left(f\left(\bar{X}, \bar{Y}\right)\right)\right| \leq \sqrt{2\sigma^2 I(X; Y)}.$$

Let $X = S, Y = W$ and $f(s, w) = \frac{1}{m}\sum_{i=1}^{m} l(h_{\mathbf{w}}(x_i), y_i)$. For $\mathbf{w} \in W$ the empirical risk can be written as $\hat{R}(\mathbf{w}) = f(S, \mathbf{w})$ and the population risk can be written as $R(\mathbf{w}) = \mathbb{E}(f(S, \mathbf{w}))$. Therefore, generalization error is

$$\text{gen}(P_{W|S}) = \mathbb{E}\left(f\left(\bar{S}, \bar{W}\right)\right) - \mathbb{E}(f(S, W)),$$

where the joint distribution of S and W is $P_{S,W} = \mathcal{D}^m \otimes P_{W|S}$. Using the fact that if $l(h_{\mathbf{w}}(x), y)$ is σ-sub-Gaussian then $f(S, \mathbf{w})$ is $\frac{\sigma}{\sqrt{m}}$-sub-Gaussian and Lemma 8.3 yields the following.

Theorem 8.4. *Suppose $l(h_{\mathbf{w}}, y)$ is σ-sub-Gaussian under $\mathcal{D}$ for all $\mathbf{w} \in \mathcal{W}$, then*

$$\left|\mathrm{gen}(P_{W|S})\right| \leq \sqrt{\frac{2\sigma^2 I(S;W)}{m}}.$$

Theorem 8.5. *Let the hypothesis space $\mathcal{W}$ be finite. Suppose that $l(h_{\mathbf{w}}, y)$ is σ-sub-Gaussian under $\mathcal{D}$ for all $\mathbf{w} \in \mathcal{W}$, then*

$$\left|\mathrm{gen}(P_{W|S})\right| \leq \sqrt{\frac{2\sigma^2 I(\Lambda_{\mathcal{W}}(S);W)}{m}},$$

where $\Lambda_{\mathcal{W}}(S) := \left(\hat{R}(\mathbf{w})\right)_{\mathbf{w}\in\mathcal{W}}$.

Theorem 8.6. *Suppose $l(h_{\mathbf{w}}, y)$ is σ-sub-Gaussian under $\mathcal{D}$ for all $\mathbf{w} \in \mathcal{W}$. If a learning algorithm satisfies $I(\Lambda_{\mathcal{W}}(S);W) \leq \epsilon$, then for any $\alpha > 0$ and $0 < \beta \leq 1$ it follows that*

$$\mathbb{P}_{P_{S,W}}\left(\left|R(W) - \hat{R}(W)\right| > \alpha\right) \leq \beta$$

can be guaranteed by a sample complexity of

$$m = \frac{8\sigma^2}{\alpha^2}\left(\frac{\epsilon}{\beta} + \log\left(\frac{2}{\beta}\right)\right).$$

Theorem 8.7. *Suppose $l(h_{\mathbf{w}}, y)$ is σ-sub-Gaussian under $\mathcal{D}$ for all $\mathbf{w} \in \mathcal{W}$. If a learning algorithm satisfies $I(\Lambda_{\mathcal{W}}(S);W) \leq \epsilon$, then*

$$\mathbb{E}_{P_{S,W}}\left(\left|R(W) - \hat{R}(W)\right|\right) \leq \sqrt{\frac{2\sigma^2(\epsilon + \log(2))}{m}}.$$

8.1 Application to Binary Classification

For binary classification $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ where $\mathcal{Y} = \{0, 1\}$. With $\mathcal{W}$ being a collection of classifiers and $l(h_{\mathbf{w}}, y) = \mathbb{I}\{h_{\mathbf{w}}(x) \neq y\}$. Given a data set S split it into S_1 and S_2 with sizes m_1 and m_2 respectively. Choose a subset of hypothesis $\mathcal{W}_1 \subset \mathcal{W}$ such that $(h_{\mathbf{w}}(x_1), \ldots, h_{\mathbf{w}}(x_{m_1}))$ for $\mathbf{w} \in \mathcal{W}_1$ are all distinct and

$$\{h_{\mathbf{w}}(x_1), \ldots, h_{\mathbf{w}}(x_{m_1}) : \mathbf{w} \in \mathcal{W}_1\} = \{h_{\mathbf{w}}(x_1), \ldots, h_{\mathbf{w}}(x_{m_1}) : \mathbf{w} \in \mathcal{W}\}.$$

Next, choose a hypothesis from $\mathcal{W}_1$ such that

$$W = \mathrm{argmin}_{\mathbf{w}\in\mathcal{W}_1} \hat{R}_{S_2}(\mathbf{w}).$$

Denote the n^{th} shatter coefficient and the VC dimension of $\mathcal{W}$ by $\mathbb{S}_n$ and V respectively. Then,

$$\mathbb{E}(R(W)) - \mathbb{E}\left(\hat{R}_{S_2}(W)\right) \leq \sqrt{\frac{V\log(m_1+1)}{2m_2}}. \tag{2}$$

Furthermore,

$$\mathbb{E}\left(\hat{R}_{S_2}(W)\right) \leq \inf_{\mathbf{w}\in\mathcal{W}} R(\mathbf{w}) + c\sqrt{\frac{V}{m_1}}, \tag{3}$$

for some constant c. Combining (2) and (3) for $m_1 = m_2 = \frac{m}{2}$ gives the bound

$$\mathbb{E}(R(W)) \leq \inf_{\mathbf{w}\in\mathcal{W}} R(\mathbf{w}) + c\sqrt{\frac{V\log(m)}{m}}.$$

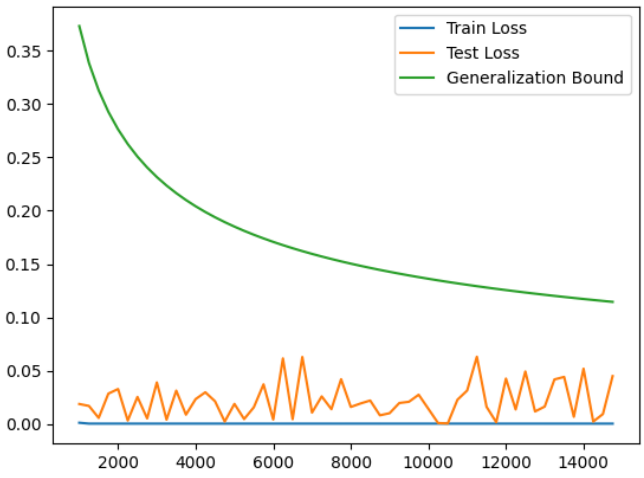

Figure 3: Bounding the test loss of a linear classifier.

9 Chaining Mutual Information

The bounds illustrated above have some key limitations. Firstly, they ignore the dependencies between the hypotheses within the sample space, and secondly, they ignore the dependencies between the input data and the output. To resolve these [5] applies a method known as chaining that was developed to tighten uniform bounds on random processes. The method works by first capturing the dependencies between hypotheses through a metric d on a set T, and then to discretize T to determine the maximum values of bounds on the random process. To develop this formally consider the setting of supervised learning, where there is an input domain $\mathcal{X}$ and a label domain $\mathcal{Y}$ with $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. The learning algorithm picks $h_W \in \mathcal{H}$ according to a random transformation $P_{W|S}$. Let

$$\text{gen}^+(P_{W|S}) := \mathbb{E}\left(\left|R(\mathbf{w}) - \hat{R}(\mathbf{w})\right|\right).$$

We will use the notation $X_{\mathcal{N}} := \{X_i : i \in \mathcal{N}\}$, $\mathbf{0}$ for the zero function, $H(x)$ to denote the Shannon entropy of discrete random variable X, and $h(Y)$ the differential entropy of an absolutely continuous random variable Y.

Definition 9.1. *Let d be a metric on the set T*

1. *A finite set $\mathcal{N}$ is an ϵ-net for (T, d) if there exists a function $\pi_{\mathcal{N}}$ which maps every point $t \in T$ to $\pi_{\mathcal{N}}(t) \in \mathcal{N}$ such that $d(t, \pi_{\mathcal{N}}(t)) \leq \epsilon$.*
2. *The covering number for a metric space (T, d) is the smallest cardinality of an ϵ-net for that space denoted $N(T, d, \epsilon)$. That is,*

$$N(T, d, \epsilon) := \inf\{|\mathcal{N}| : \mathcal{N} \text{ is an } \epsilon\text{-net for } (T, d)\}.$$

3. *An ϵ-net $\mathcal{N}$ for the metric space (T, d) is called minimal if $|\mathcal{N}| = N(T, d, \epsilon)$.*

Definition 9.2. *The random process $\{X_t\}_{t \in T}$ on the metric space (T, d) is called sub-Gaussian if $\mathbb{E}(X_t) = 0$ for all $t \in T$ and*

$$\mathbb{E}\left(e^{\lambda(X_t - X_s)}\right) \leq e^{\frac{1}{2}\lambda^2 d^2(t,s)}$$

for all $t, s \in T, \lambda \geq 0$.

Definition 9.3. *The random process $\{X_t\}_{t\in T}$ is called separable if there is a countable set $T_0 \subseteq T$ such that $X_t \in \lim_{s\to t, s\in T_0} X_s$ for all $t \in T$ a.s, where $x \in \lim_{s\to t, s\in T_0} x_s$ means that there is a sequence (s_n) in T_0 such that $s_n \to t$ and $x_{s_n} \to x$.*

Definition 9.4. *Call a partition $\mathcal{P} = \{A_1, \ldots, A_m\}$ of the set T an ϵ-partition of the metric space (T, d) if for all $i = 1, \ldots, m$ A_i can be contained within a ball of radius ϵ. A sequence of partitions $\{\mathcal{P}_k\}_{k=m}^{\infty}$ of a set T is called an increasing sequence if for all $k \geq m$ and each $A \in \mathcal{P}_{k+1}$ there exists $B \in \mathcal{P}_k$ such that $A \subseteq B$. For any such sequence and any $t \in T$ let $[t]_k$ denote the unique set $A \in \mathcal{P}_k$ such that $t \in A$.*

From now on assume that (T, d) is a bounded metric space, with $k_1(T)$ an integer such that $2^{-(k_1(T)-1)} \geq \mathrm{diam}(T)$.

Theorem 9.5. *Assume that $\{X_t\}_{t\in T}$ is a separable sub-Gaussian process on the bounded metric space $(\mathcal{W}, d)$. Let $\{\mathcal{P}_k\}_{k=k_1(\mathcal{W})}$ be an increasing sequence of partitions of $\mathcal{W}$, where for each $k \geq k_1(T)$, $\mathcal{P}_k$ is a 2^{-k}-partition of (T, d).*

1.

$$\mathbb{E}(X_W) \leq 3\sqrt{2} \sum_{k=k_1(T)}^{\infty} 2^{-k} \sqrt{I([W]_k; X_T)},$$

2. For arbitrary $t_0 \in T$,

$$\mathbb{E}(|X_W - X_{t_0}|) \leq 3\sqrt{2} \sum_{k=k_1(T)}^{\infty} 2^{-k} \sqrt{I([W]_k; X_T) + \log(2)}.$$

Corollary 9.6. *Assume that $\{\mathrm{gen}(\mathbf{w})\}_{\mathbf{w}\in\mathcal{W}}$ is a separable sub-Gaussian process on the bounded metric space $(\mathcal{W}, d)$. Let $\{\mathcal{P}_k\}_{k=k_1(\mathcal{W})}$ be an increasing sequence of partitions of $\mathcal{W}$, where for each $k \geq k_1(\mathcal{W})$, $\mathcal{P}_k$ is a 2^{-k}-partition of $(\mathcal{W}, d)$.*

1.

$$\mathrm{gen}(P_{W|S}) \leq 3\sqrt{2} \sum_{k=k_1(\mathcal{W})}^{\infty} 2^{-k} \sqrt{I([W]_k; S)},$$

2. If $\mathbf{0} \in \{l(h_{\mathbf{w}}, \cdot) : \mathbf{w} \in \mathcal{W}\}$, then

$$\mathrm{gen}^{+}(P_{W|S}) \leq 3\sqrt{2} \sum_{k=k_1(\mathcal{W})}^{\infty} 2^{-k} \sqrt{I([W]_k; S) + \log(2)}.$$

10 Conditional Mutual Information

When investigating learning algorithms using mutual information noise had to be introduced into our observations to get finite values of mutual information. The work of [8] resolves this by considering conditional mutual information instead (CMI). CMI measures how well the learning algorithm can recognize the input given the output. This is calculated by using a "supersample", which consists of regular data points and "ghost" data points. CMI measures the ability to distinguish the regular inputs from their ghosts.

Definition 10.1. *Let $A : \mathcal{Z}^m \to \mathcal{W}$ be a randomized or deterministic algorithm. Let $\mathcal{D}$ be a probability distribution on $\mathcal{Z}$ and let $S = \{(x_i, y_i)\}_{i=1}^{2m}$ consist of $2m$ samples drawn independently from $\mathcal{D}$. Let $\zeta \in \{0,1\}^n$ be uniformly random and independent from S and the randomness of A. Define $S_\zeta \in \mathcal{Z}^m$ by $(S_\zeta)_i = S_{i,\zeta_i+1}$ for all $i \in [n]$.*

- *The conditional mutual information (CMI) of A with respect to $\mathcal{D}$ is,*

$$\mathrm{CMI}_{\mathcal{D}}(A) := I\left(A\left(S_\zeta\right); \zeta | S\right).$$

- *The (distribution-free) conditional mutual information (CMI) of A is*

$$\mathrm{CMI}(A) := \sup_{S \in \mathcal{Z}^{2m}} I\left(A\left(S_\zeta\right); \zeta\right).$$

Remark 10.2.

- *For any A and $\mathcal{D}$ it follows that $0 \le \mathrm{CMI}_{\mathcal{D}}(A) \le \mathrm{CMI}(A) \le n\log(2)$.*
- *If $\mathrm{CMI}_{\mathcal{D}}(A) = 0$ then the output of A is independent of its input.*
- *If $\mathrm{CMI}_{\mathcal{D}}(A) = n\log(2)$ then the output of A reveals all of the input.*
- *CMI is always finite.*

10.1 CMI and VC Dimension

Recall that VC dimension is a property of a hypothesis class, whereas, CMI depends also on the algorithm. However, there is a connection between the two. Which enables one to utilize the theory of VC dimension and learnability to the generalization results provided by CMI. Let $\mathcal{W}$ be a class of function $h : \mathcal{X} \to \{0,1\}$. Consider the 0-1 loss $l : \mathcal{Y} \times \mathcal{Y} \to \{0,1\}$ defined by

$$l(h(x), y) = \begin{cases} 0 & h(x) = y \\ 1 & \text{otherwise.} \end{cases}$$

We call $A : \mathcal{Z}^m \to \mathcal{W}$ an empirical risk minimizer for $\mathcal{W}$ if

$$\hat{R}(A(S)) = \inf_{\mathbf{w} \in \mathcal{W}} \hat{R}(\mathbf{w})$$

for all $S \in \mathcal{Z}^m$.

Theorem 10.3. *Let $\mathcal{Z} = \mathcal{X} \times \{0,1\}$ and $\mathcal{H} = \{h : \mathcal{X} \to \{0,1\}\}$ a hypothesis class with VC dimension d. Then, there exists an empirical risk minimizer $A : \mathcal{Z}^n \to \mathcal{H}$ such that $\mathrm{CMI}(A) \le d\log(n) + 2$.*

10.2 Generalization via CMI

CMI can be used to generate generalization bounds.

Theorem 10.4. *Let $\mathcal{D}$ be a distribution on $\mathcal{Z}$. Let $A : \mathcal{Z}^n \to \mathcal{W}$ be a randomized algorithm. Let $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be an arbitrary (deterministic and measurable) function. Suppose there exists $\Delta : \mathcal{Z}^2 \to \mathbb{R}$ such that $|l(h_{\mathbf{w}}(x_1), y_1) - l(h_{\mathbf{w}}(x_2), y_2)| \leq \Delta((x_1, y_1), (x_2, y_2))$ for all $(x_1, y_1), (x_2, y_2) \in \mathcal{Z}$ and $\mathbf{w} \in \mathcal{W}$. Then,*

$$\left|\mathbb{E}_{S\sim\mathcal{D}^m, A}\left(\hat{R}(A(S)) - R(A(S)\right)\right| \leq \sqrt{\frac{2}{m}\mathrm{CMI}_{\mathcal{D}}(A)\mathbb{E}_{(z_1,z_2)\sim\mathcal{D}^2}\left(\Delta\left((x_1, y_1), (x_2, y_2)\right)^2\right)}.$$

A tighter stronger statement can be obtained by losing a factor in the bound.

Theorem 10.5. *Let $\mathcal{D}$ be a distribution on $\mathcal{Z}$. Let $A : \mathcal{Z}^n \to \mathcal{W}$ be a randomized algorithm. Let $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be an arbitrary function. Suppose there exists $\Delta : \mathcal{Z}^2 \to \mathbb{R}$ such that $|l(h_{\mathbf{w}}(x_1), y_1) - l(h_{\mathbf{w}}(x_2), y_2)| \leq \Delta((x_1, y_1), (x_2, y_2))$ for all $(x_1, y_1), (x_2, y_2) \in \mathcal{Z}$ and $\mathbf{w} \in \mathcal{W}$. Then,*

$$\left|\mathbb{E}_{S\sim\mathcal{D}^m, A}\left(\hat{R}(A(S)) - R(A(S)\right)\right| \leq \sqrt{\frac{2}{m}(\mathrm{CMI}_{\mathcal{D}}(A) + \log(2))\mathbb{E}_{(z_1,z_2)\sim\mathcal{D}^2}\left(\Delta\left((x_1, y_1), (x_2, y_2)\right)^2\right)}$$

References

[1] Jiantao Jiao, Yanjun Han, and Tsachy Weissman. “Dependence Measures Bounding the Exploration Bias for General Measurements”. In: *CoRR* (2016).

[2] Daniel Russo and James Zou. “Controlling Bias in Adaptive Data Analysis Using Information Theory”. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Ed. by Arthur Gretton and Christian C. Robert. Vol. 51. Proceedings of Machine Learning Research. PMLR, 2016, pp. 1232–1240.

[3] Ravid Shwartz-Ziv and Naftali Tishby. “Opening the Black Box of Deep Neural Networks via Information”. In: *CoRR* (2017).

[4] Aolin Xu and Maxim Raginsky. “Information-theoretic analysis of generalization capability of learning algorithms”. In: *CoRR* (2017).

[5] Amir R. Asadi, Emmanuel Abbe, and Sergio Verdú. “Chaining Mutual Information and Tightening Generalization Bounds”. In: *CoRR* (2018).

[6] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, 2018.

[7] Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. “On the Information Bottleneck Theory of Deep Learning”. In: *International Conference on Learning Representations*. 2018.

[8] Thomas Steinke and Lydia Zakynthinou. “Reasoning About Generalization via Conditional Mutual Information”. In: *CoRR* (2020).

[9] Patrick Rebeschini. *Algorithmic Foundations of Learning*. Nov. 2022.