# Minimal Achievable Sufficient Statistic Learning

**Milan Cvitkovic** [1] **Günther Koliander** [2]

## Abstract

We introduce Minimal Achievable Sufficient Statistic (MASS) Learning, a machine learning training objective for which the minima are minimal sufficient statistics with respect to a class of functions being optimized over (e.g., deep networks). In deriving MASS Learning, we also introduce Conserved Differential Information (CDI), an information-theoretic quantity that — unlike standard mutual information — can be usefully applied to deterministically-dependent continuous random variables like the input and output of a deep network. In a series of experiments, we show that deep networks trained with MASS Learning achieve competitive performance on supervised learning, regularization, and uncertainty quantification benchmarks.

## 1. Introduction

The *representation learning* approach to machine learning focuses on finding a representation $Z$ of an input random variable $X$ that is useful for predicting a random variable $Y$ (Goodfellow et al., 2016).

What makes a representation $Z$ "useful" is much debated, but a common assertion is that $Z$ should be a *minimal sufficient statistic* of $X$ for $Y$ (Adragni, Kofi P. & Cook, R. Dennis, 2009; Shamir et al., 2010; James et al., 2017; Achille & Soatto, 2018b). That is:

1. $Z$ should be a *statistic* of $X$. This means $Z = f(X)$ for some function $f$.
2. $Z$ should be *sufficient* for $Y$. This means $p(X|Z, Y) = p(X|Z)$.
3. Given that $Z$ is a sufficient statistic, it should be *minimal* with respect to $X$. This means for any measurable, non-invertible function $g$, $g(Z)$ is no longer sufficient for $Y$.[1]

In other words: a minimal sufficient statistic is a random variable $Z$ that tells you everything about $Y$ you could ever care about, but if you do any irreversible processing to $Z$, you are guaranteed to lose some information about $Y$.

Minimal sufficient statistics have a long history in the field of statistics (Lehmann & Scheffe, 1950; Dynkin, 1951). But the minimality condition (3, above) is perhaps too strong to be useful in machine learning, since it is a statement about *any* measurable function $g$, rather than about functions in a practical hypothesis class like the class of deep neural networks.

Instead, in this work we consider *minimal achievable sufficient statistics*: sufficient statistics that are minimal within some particular set of functions.

**Definition 1** (Minimal Achievable Sufficient Statistic)**.** Let $f(X)$ be a sufficient statistic of $X$ for $Y$. $f(X)$ is *minimal achievable* with respect to a set of functions $\mathcal{F}$ if $f \in \mathcal{F}$ and for any Lipschitz continuous, non-invertible function $g$, $g(f(X))$ is no longer sufficient for $Y$.

In this work, we give a characterization of minimal achievable sufficient statistics that is applicable to deep neural networks and show that it can be used to train models with competitive performance on classification accuracy, uncertainty quantification, and out-of-distribution input detection.

**Contributions:**

- We introduce Conserved Differential Information (CDI), an information-theoretic quantity that, unlike mutual information, is meaningful for deterministically-dependent continuous random variables, such as the input and output of a deep network.
- We introduce Minimal Achievable Sufficient Statistic Learning (MASS Learning), a training objective

[1] Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, California, USA [2] Acoustics Research Institute, Austrian Academy of Sciences, Vienna, Austria. Correspondence to: Milan Cvitkovic <mcvitkov@caltech.edu>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

[1] Although this is not the most common phrasing of statistical minimality, we feel it is more understandable. For the equivalence of this phrasing and the standard definition see Supplementary Material 7.1.

based on CDI for finding minimal achievable sufficient statistics.

- We provide empirical evidence that models trained by MASS Learning achieve competitive performance on supervised learning, regularization, and uncertainty quantification benchmarks.

## 2. Conserved Differential Information

Before we present MASS Learning, we need to introduce Conserved Differential Information (CDI), on which MASS Learning is based.

CDI is an information-theoretic quantity that addresses an oft-cited issue in machine learning (Bell & Sejnowski, 1995; Amjad & Geiger, 2018; Saxe et al., 2018; Nash et al., 2018; Goldfeld et al., 2018), which is that for a continuous random variable $X$ and a continuous, non-constant function $f$, the mutual information $I(X, f(X))$ is infinite. (See Supplementary Material 7.2 for details.) This makes $I(X, f(X))$ unsuitable for use in a learning objective when $f$ is, for example, a standard deep network.

The infinitude of $I(X, f(X))$ has been circumvented in prior works by two strategies. One is discretize $X$ and $f(X)$ (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017), though this is controversial (Saxe et al., 2018). Another is to use a random variable $Z$ with distribution $p(Z|X)$ as the representation of $X$ rather than using $f(X)$ itself as the representation (Alemi et al., 2016; Kolchinsky et al., 2017; Achille & Soatto, 2018b). In this latter approach, $p(Z|X)$ is usually implemented by adding noise to a deep network that takes $X$ as input.

These are both reasonable strategies for avoiding the infinitude of $I(X, f(X))$. But another approach would be to derive a new information-theoretic quantity that is better suited to this situation. To that end we present Conserved Differential Information:

**Definition 2.** For a continuous random variable $X$ taking values in $\mathbb{R}^d$ and a Lipschitz continuous function $f$, the **Conserved Differential Information** (CDI) is

$$\boxed{C(X, f(X)) := H(f(X)) - \mathbb{E}_X\left[\log\left(J_f(X)\right)\right]} \quad (1)$$

where $H$ denotes the differential entropy

$$H(Z) = -\int p(z) \log p(z) \, \mathrm{d}z$$

and $J_f$ is the Jacobian determinant of $f$

$$J_f(x) = \sqrt{\det\left(\frac{\partial f(x)}{\partial x^{\mathrm{T}}}\left(\frac{\partial f(x)}{\partial x^{\mathrm{T}}}\right)^{\mathrm{T}}\right)}$$

with $\frac{\partial f(x)}{\partial x^{\mathrm{T}}}$ the Jacobian matrix of $f$ at $x$.

Readers familiar with normalizing flows (Rezende & Mohamed, 2015) or Real NVP (Dinh et al., 2016) will note that the Jacobian determinant used in those methods is a special case of the Jacobian determinant in the definition of CDI. This is because normalizing flows and Real NVP are based on the change of variables formula for invertible mappings, while CDI is based in part on the more general change of variables formula for non-invertible mappings. More details on this connection are given in Supplementary Material 7.3. The mathematical motivation for CDI based on the recent work of (Koliander et al., 2016) is provided in Supplementary Material 7.4. Figure 1 gives a visual example of what CDI measures about a function.

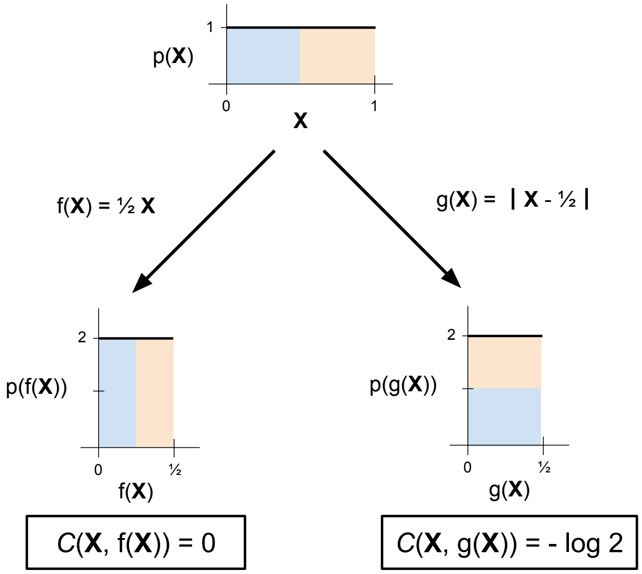

*Figure 1.* CDI of two functions $f$ and $g$ of the random variable $X$. Even though the random variables $f(X)$ and $g(X)$ have the same distribution, $C(X, f(X))$ is different from $C(X, g(X))$. This is because $f$ is an invertible function, while $g$ is not.

The conserved differential information $C(X, f(X))$ between deterministically-dependent random variables behaves a lot like mutual information does on discrete random variables. For example, when $f$ is invertible, $C(X, f(X)) = H(X)$, just like with the mutual information between discrete random variables. Most importantly for our purposes, though, $C(X, f(X))$ obeys the following data processing inequality:

**Theorem 1** (CDI Data Processing Inequality)**.** *For Lipschitz continuous functions $f$ and $g$ with the same output space,*

$$C(X, f(X)) \geq C(X, g(f(X)))$$

*with equality if and only if $g$ is invertible almost everywhere.*

The proof is in Supplementary Material 7.5.

## 3. MASS Learning

With CDI and its data processing inequality in hand, we can give the following optimization-based characterization of minimal achievable sufficient statistics:

**Theorem 2.** *Let $X$ be a continuous random variable, $Y$ be a discrete random variable, and $\mathcal{F}$ be any set of Lipschitz continuous functions with a common output space (e.g., different parameter settings of a deep network). If*

$$\begin{aligned} f &\in \arg\min_{S\in\mathcal{F}} C(X, S(X)) \\ &s.t.\ I(S(X), Y) = \max_{S'} I(S'(X), Y) \end{aligned}$$

*then $f(X)$ is a minimal achievable sufficient statistic of $X$ for $Y$ with respect to $\mathcal{F}$.*

*Proof.* First note the following lemma (Cover & Thomas, 2006):

**Lemma 1.** *$Z = f(X)$ is a sufficient statistic for a discrete random variable $Y$ if and only if $I(Z, Y) = \max_{S'} I(S'(X), Y)$.*

Lemma 1 guarantees that any $f$ satisfying the conditions in Theorem 2 is sufficient. If such an $f$ was not minimal achievable there would exist a non-invertible, Lipschitz continuous $g$ such that $g(f(X))$ was sufficient and by Theorem 1 $C(X, g(f(X))) < C(X, f(X))$ contradicting $f$ minimizing $C(X, S(X))$. □

We can turn Theorem 2 into a learning objective over functions $f$ by relaxing the strict constraint into a Lagrangian formulation with Lagrange multiplier $1/\beta$ with $\beta > 0$:

$$C(X, f(X)) - \frac{1}{\beta} I(f(X), Y)$$

The larger the value of $\beta$, the more our objective will encourage minimality over sufficiency. We can then simplify this formulation using the identity $I(f(X), Y) = H(Y) - H(Y|f(X))$, which gives us the following optimization objective:

$$\boxed{\begin{aligned} \mathcal{L}_{MASS}(f) := H(Y|f(X)) + \beta H(f(X)) \\ - \beta \mathbb{E}_X[\log J_f(X)]. \end{aligned}} \quad (2)$$

We refer to minimizing this objective as **MASS Learning**.

In practice, we are interested in using MASS Learning to train a deep network $f_\theta$ with parameters $\theta$ using a finite dataset $\{(x_i, y_i)\}_{i=1}^N$ of $N$ datapoints sampled from the joint distribution $p(x, y)$ of $X$ and $Y$. To do this, we introduce a parameterized variational approximation $q_\phi(f_\theta(x)|y) \approx p(f_\theta(x)|y)$. Using $q_\phi$, we minimize the following empirical upper bound to $\mathcal{L}_{MASS}$:

$$\begin{aligned} \widehat{\mathcal{L}}_{MASS}(\theta, \phi) := \frac{1}{N}\sum_{i=1}^{N} & -\log q_\phi(y_i|f_\theta(x_i)) \\ & - \beta \log q_\phi(f_\theta(x_i)) \\ & - \beta \log J_{f_\theta}(x_i) \geq \mathcal{L}_{MASS}, \end{aligned}$$

where the quantity $q_\phi(f_\theta(x_i))$ is computed as $\sum_y q_\phi(f_\theta(x_i)|y)p(y)$ and the quantity $q_\phi(y_i|f_\theta(x_i))$ is computed with Bayes rule as $\frac{q_\phi(f_\theta(x_i)|y_i)p(y_i)}{\sum_y q_\phi(f_\theta(x_i)|y)p(y)}$. When $Y$ is discrete and takes on finitely many values, as in classification problems, and when we choose a variational distribution $q_\phi$ that is differentiable with respect to $\phi$ (e.g., a multivariate Gaussian), then we can minimize $\widehat{\mathcal{L}}_{MASS}(\theta, \phi)$ using stochastic gradient descent.

To perform classification using our trained network, we use the learned variational distribution $q_\phi$ and Bayes rule

$$p(Y|X) \approx p(Y|f_\theta(X)) \approx \frac{q_\phi(f_\theta(X)|Y)p(Y)}{\sum_y q_\phi(f_\theta(X)|y)p(y)}.$$

## 4. Related Work

### 4.1. Connection to the Information Bottleneck

The well-studied Information Bottleneck learning method (Tishby et al., 2000; Tishby & Zaslavsky, 2015; Strouse & Schwab, 2016; Alemi et al., 2016; Saxe et al., 2018; Amjad & Geiger, 2018; Goldfeld et al., 2018; Kolchinsky et al., 2018; Achille & Soatto, 2018b;a) is based on minimizing the Information Bottleneck Lagrangian

$$\mathcal{L}_{IB}(Z) := \beta I(X, Z) - I(Y, Z)$$

for $\beta > 0$, where $Z$ is the representation whose conditional distribution $p(Z|X)$ we are trying to learn.

The $\mathcal{L}_{IB}$ learning objective can be motivated based on pure information-theoretic elegance. But some works like (Shamir et al., 2010) also point out the connection between the $\mathcal{L}_{IB}$ objective and minimal sufficient statistics, which is based on the following theorem:

**Theorem 3.** *Let $X$ be a discrete random variable drawn according to a distribution $p(X|Y)$ determined by the discrete random variable $Y$. Let $\mathcal{F}$ be the set of deterministic functions of $X$ to any target space. Then $f(X)$ is a minimal sufficient statistic of $X$ for $Y$ if and only if*

$$\begin{aligned} f &\in \arg\min_{S\in\mathcal{F}} I(X, S(X)) \\ &s.t.\ I(S(X), Y) = \max_{S'\in\mathcal{F}} I(S'(X), Y). \end{aligned}$$

The $\mathcal{L}_{IB}$ objective can then be thought of as a Lagrangian relaxation of the optimization problem in this theorem.

Theorem 3 only holds for discrete random variables. For continuous $X$ it holds only in the reverse direction, so minimizing $\mathcal{L}_{IB}$ for continuous $X$ has no formal connection to finding minimal sufficient statistics, not to mention minimal achievable sufficient statistics. See Supplementary Material 7.6 for details.

Nevertheless, the optimization problems in Theorem 2 and Theorem 3 are extremely similar, relying as they both do on Lemma 1 for their proofs. And the idea of relaxing the optimization problem in Theorem 2 into a Lagrangian formulation to get $\mathcal{L}_{MASS}$ is directly inspired by the Information Bottleneck. So while MASS Learning and Information Bottleneck learning entail different network architectures and loss functions, there is an Information Bottleneck flavor to MASS Learning.

### 4.2. Jacobian Regularization

The presence of the $J_{f_\theta}$ term in $\widehat{\mathcal{L}}_{MASS}$ is reminiscent of the contrastive autoencoder (Rifai et al., 2011) and Jacobian Regularization literature (Sokolic et al., 2017; Ross & Doshi-Velez, 2017; Varga et al., 2017; Novak et al., 2018; Jakubovitz & Giryes, 2018). Both these literatures suggest that minimizing $\mathbb{E}_X[\|D_f(X)\|_F]$ where $D_f(x) = \frac{\partial f(x)}{\partial x^\mathrm{T}}$ is the Jacobian matrix seems to improve generalization and/or adversarial robustness.

This may seem paradoxical at first, since by applying the AM-GM inequality to the eigenvalues of $D_f(x)D_f(x)^\mathrm{T}$ where $D_f = \frac{\partial f(x)}{\partial x^\mathrm{T}} \in \mathbb{R}^{r\times d}$, we have

$$\begin{aligned}
\mathbb{E}_X[\|D_f(X)\|_F^{2r}] &= \mathbb{E}_X[\mathrm{Tr}(D_f(X)D_f(X)^\mathrm{T})^r] \\
&\geq \mathbb{E}_X[r^r \det(D_f(X)D_f(X)^\mathrm{T})] \\
&= \mathbb{E}_X[r^r J_f(X)^2] \\
&\geq \log \mathbb{E}_X[r^r J_f(X)^2] \\
&\geq 2\mathbb{E}_X[\log J_f(X)] + r\log r
\end{aligned}$$

and $\mathbb{E}_X[\log J_f(X)]$ is being *maximized* by $\widehat{\mathcal{L}}_{MASS}$. So $\widehat{\mathcal{L}}_{MASS}$ would seem to be optimizing for worse generalization according to the Jacobian regularization literature. However, the conditional entropy term in $\widehat{\mathcal{L}}_{MASS}$ strongly encourages minimizing $\mathbb{E}_X[\|D_f(X)\|_F]$. So overall $\widehat{\mathcal{L}}_{MASS}$ seems to be seeking the right balance of sensitivity (dependent on the value of $\beta$) in the network to its inputs, which is precisely in alignment with what the Jacobian regularization literature suggests.

## 5. Experiments

Code to reproduce all experiments is available online.[2] Full details on all experiments is in Supplementary Material 7.7.

In this section we compare MASS Learning to other approaches for training deep networks. We use the abbreviation "SoftmaxCE" to refer to the standard approach of training deep networks for classification problems by minimizing the softmax cross entropy loss

$$\widehat{\mathcal{L}}_{SoftmaxCE}(\theta) := -\frac{1}{N}\sum_{i=1}^{N}\Big(\log \mathtt{softmax}(f_\theta(x_i))_{y_i}\Big)$$

where $\mathtt{softmax}(f_\theta(x_i))_{y_i}$ is the $y_i$th element of the softmax function applied to the outputs $f_\theta(x_i)$ of the network's last linear layer. As usual, $\mathtt{softmax}(f_\theta(x_i))_{y_i}$ is taken to be the network's estimate of $p(y_i|x_i)$.

We also compare against the Variational Information Bottleneck (Alemi et al., 2016) method for representation learning, which we abbreviate as "VIB".

We use two networks in our experiments. "SmallMLP" is a feedforward network with two fully-connected layers of 400 and 200 hidden units, respectively, both with `elu` nonlinearities (Clevert et al., 2015). "ResNet20" is the 20-layer residual net of He et al. (2015).

In all our experiments, the variational distribution $q_\phi(x|y)$ for each possible output class $y$ is a mixture of multivatiate Gaussian distributions for which we learn the mixture weights, means, and covariance matrices.

Computing the $J_{f_\theta}$ term in $\widehat{\mathcal{L}}_{MASS}(\theta,\phi)$ for every sample in a minibatch is too expensive to be practical. Doing so would require on the order of $|Y|$ times more operations than computing $\widehat{\mathcal{L}}_{SoftmaxCE}(\theta)$, since computing the $J_{f_\theta}$ term in $\widehat{\mathcal{L}}_{MASS}(\theta,\phi)$ requires (in our implementation) computing the full Jacobian matrix of the network. Thus to make training tractable, we use a subsampling strategy: we estimate the $J_{f_\theta}$ term using only a $1/|Y|$ fraction of the datapoints in a minibatch. In practice, we do not notice any performance detriment when using the subsampling strategy, and the numerical value of the $J_{f_\theta}$ during training with subsampling is indistinguishable from training with no subsampling.

Subsampling for the $J_{f_\theta}$ term results in a significant performance improvement, but it must nevertheless be emphasized that even with the subsampling strategy, our implementation of MASS Learning is roughly twice as computationally costly as SoftmaxCE training. (Unless $\beta = 0$, in which case the cost is the same as SoftmaxCE.) This is by far the most significant drawback of (our implementation of) MASS Learning. There are many easier-to-compute upper bounds or estimates of $J_{f_\theta}$ that one could use to make MASS Learning faster, but we do not explore these in this work.

We performed all experiments on the CIFAR-10 dataset (Krizhevsky, 2009), and coded all our models in PyTorch

[2] https://github.com/mwcvitkovic/MASS-Learning

(Paszke et al., 2017).

### 5.1. Classification Accuracy and Regularization

We first confirm that networks trained by MASS Learning can make accurate predictions in supervised learning tasks. We also compare the classification accuracy of networks trained on varying amounts of data to see whether MASS Learning successfully regularizes networks and improves their generalization performance.

Classification accuracies for the SmallMLP network are shown in Table 1, and for the ResNet20 network in Table 2. For the SmallMLP network, MASS Learning does not appear to offer any performance benefits. For the larger ResNet20 network, the results show that while MASS Learning maintains or improves accuracy compared to SoftmaxCE training, often fairly significantly, these improvements do not seem to be due to the MASS loss $\widehat{\mathcal{L}}_{MASS}(\theta, \phi)$ itself, since the same performance improvements are obtained even when the $H(f(X))$ and $\mathbb{E}_X[\log J_f(X)]$ terms in the MASS loss are set to 0 (i.e. the case when $\beta = 0$).

This suggests that it is the use of the variational distribution $q_\phi(x|y)$ to produce the output of the network, rather than the MASS Learning approach, that is providing the benefit. This is an interesting finding, but does not suggest an advantage to using the full MASS Learning method if one is concerned with accuracy or regularization.

### 5.2. Uncertainty Quantification

We also evaluate the ability of networks trained by MASS Learning to properly quantify their uncertainty about their predictions. We assess uncertainty quantification in two ways: using proper scoring rules (Lakshminarayanan et al., 2016), which are scalar measures of how well a network's predictive distribution is calibrated, and by observing performance on an out-of-distribution (OOD) detection task.

Tables 3 and 4 show the uncertainty quantification performance of networks according to three proper scoring rules: the Negative Log Likelihood (NLL), the Brier Score, and entropy of the predictive distribution $p(y|f_\theta(x))$. With the SmallMLP network SoftmaxCE and VIB training perform best, while with the ResNet20 network the results are more varied. In general, though, any benefits produced by MASS Learning seem to derive not from the MASS objective but from the network architecture, since MASS Learning with $\beta = 0$ gives performance comparable to MASS Learning with $\beta \neq 0$.

Table 5 shows scalar metrics for performance on an OOD detection task where the network is asked to identify whether an image is from its training distribution (CIFAR-10 images) or from another distribution (SVHN images (Netzer et al., 2011)). Following Hendrycks & Gimpel (2016) and Alemi

*Table 1.* Test-set classification accuracy (percent) on CIFAR-10 dataset using the SmallMLP network trained by various methods. Full experiment details are in Supplementary Material 7.7. Values are the mean classification accuracy over 4 training runs with different random seeds plus or minus the standard deviation. Emboldened accuracies are those for which the maximum observed mean accuracy in the column was within one standard deviation. WD is weight decay; D is dropout.

| METHOD | TRAINING SET SIZE | | |
|---|---|---|---|
| | 2500 | 10,000 | 40,000 |
| SoftmaxCE | $33.9 \pm 0.5$ | $\mathbf{44.5 \pm 0.3}$ | $52.4 \pm 1.1$ |
| SoftmaxCE, WD | $26.2 \pm 0.9$ | $36.5 \pm 0.8$ | $47.8 \pm 0.6$ |
| SoftmaxCE, D | $33.0 \pm 1.1$ | $\mathbf{43.9 \pm 0.6}$ | $\mathbf{54.2 \pm 0.5}$ |
| VIB, $\beta$=1e$-$1 | $32.3 \pm 0.4$ | $40.6 \pm 0.6$ | $46.4 \pm 0.6$ |
| VIB, $\beta$=1e$-$2 | $34.2 \pm 0.4$ | $\mathbf{44.1 \pm 0.5}$ | $51.6 \pm 0.4$ |
| VIB, $\beta$=1e$-$3 | $\mathbf{35.1 \pm 0.7}$ | $\mathbf{44.2 \pm 0.6}$ | $51.7 \pm 0.7$ |
| VIB, $\beta$=1e$-$1, D | $28.9 \pm 0.9$ | $39.9 \pm 0.5$ | $49.8 \pm 0.1$ |
| VIB, $\beta$=1e$-$2, D | $32.9 \pm 1.2$ | $\mathbf{43.7 \pm 0.8}$ | $53.9 \pm 0.4$ |
| VIB, $\beta$=1e$-$3, D | $\mathbf{34.1 \pm 1.0}$ | $\mathbf{44.3 \pm 0.5}$ | $\mathbf{54.5 \pm 0.3}$ |
| MASS, $\beta$=1e$-$2 | $30.3 \pm 0.4$ | $39.9 \pm 1.1$ | $45.4 \pm 1.4$ |
| MASS, $\beta$=1e$-$3 | $32.6 \pm 0.6$ | $40.9 \pm 0.6$ | $47.0 \pm 0.8$ |
| MASS, $\beta$=1e$-$4 | $33.4 \pm 0.6$ | $40.7 \pm 0.4$ | $47.1 \pm 1.1$ |
| MASS, $\beta$=0 | $34.0 \pm 0.5$ | $40.8 \pm 1.0$ | $47.0 \pm 0.6$ |
| MASS, $\beta$=1e$-$2, D | $29.6 \pm 1.2$ | $42.2 \pm 0.5$ | $51.9 \pm 0.5$ |
| MASS, $\beta$=1e$-$3, D | $31.8 \pm 1.3$ | $43.4 \pm 0.4$ | $53.0 \pm 0.5$ |
| MASS, $\beta$=1e$-$4, D | $31.9 \pm 0.8$ | $43.2 \pm 0.2$ | $52.9 \pm 0.6$ |
| MASS, $\beta$=0, D | $32.1 \pm 1.3$ | $43.4 \pm 0.4$ | $52.7 \pm 0.4$ |

*Table 2.* Test-set classification accuracy (percent) on CIFAR-10 dataset using the ResNet20 network trained by various methods. No data augmentation or learning rate scheduling was used — full details in Supplementary Material 7.7. Values are the mean classification accuracy over 4 training runs with different random seeds plus or minus the standard deviation. Emboldened accuracies are those for which the maximum observed mean accuracy in the column was within one standard deviation.

| METHOD | TRAINING SET SIZE | | |
|---|---|---|---|
| | 2500 | 10,000 | 40,000 |
| SoftmaxCE | $37.4 \pm 0.7$ | $\mathbf{52.0 \pm 1.1}$ | $\mathbf{67.8 \pm 2.7}$ |
| VIB, $\beta$=1e$-$3 | $33.5 \pm 0.9$ | $49.1 \pm 1.5$ | $66.0 \pm 0.6$ |
| VIB, $\beta$=1e$-$4 | $34.0 \pm 1.0$ | $50.3 \pm 1.6$ | $67.1 \pm 0.6$ |
| VIB, $\beta$=1e$-$5 | $34.7 \pm 0.6$ | $50.2 \pm 1.6$ | $67.8 \pm 0.6$ |
| VIB, $\beta$=0 | $35.3 \pm 0.7$ | $50.0 \pm 1.7$ | $68.0 \pm 0.1$ |
| MASS, $\beta$=1e$-$3 | $38.5 \pm 0.9$ | $\mathbf{52.0 \pm 1.0}$ | $67.1 \pm 0.5$ |
| MASS, $\beta$=1e$-$4 | $39.1 \pm 0.3$ | $\mathbf{52.7 \pm 0.7}$ | $\mathbf{68.9 \pm 1.1}$ |
| MASS, $\beta$=1e$-$5 | $\mathbf{39.0 \pm 1.0}$ | $\mathbf{52.5 \pm 1.1}$ | $\mathbf{69.5 \pm 0.6}$ |
| MASS, $\beta$=0 | $\mathbf{39.7 \pm 0.5}$ | $\mathbf{52.9 \pm 0.4}$ | $\mathbf{69.0 \pm 0.8}$ |

et al. (2018), the metrics we report are the Area under the ROC curve (AUROC) and Average Precision score (APR). APR depends on whether the network is tasked with predicting whether an image is in-distribution or out of distribution; we report both metrics as APR In and APR Out, respectively. The Entropy detection method uses the entropy of the network's learned predictive distribution $p(y|f_\theta(x))$ as the OOD detection value. The $\max_i q_\phi(f_\theta(x)|y_i)$ detection method uses the maximum pdf value for any of the potential output classes $y_i$ as the OOD detection value. (For the SoftmaxCE trained networks, $q_\phi(f_\theta(x)|y_i)$ was estimated by MLE of a mixture of 10 full-covariance, 10-dimensional multivariate Gaussians on the training set.) And for the VIB networks, the Rate detection method uses the KL divergence between the VIB's marginal distribution and the representation as the OOD detection value.

Here we see MASS Learning outperforming SoftmaxCE and VIB, but again with the caveat that the benefits appear to be due to the variational distribution in the network architecture, rather than the MASS loss function.

### 5.3. Does MASS Learning finally solve the mystery of why stochastic gradient descent with the cross entropy loss works so well in deep learning?

We do not believe so. MASS Learning and SoftmaxCE training seem to be producing fairly different representations during training. Figure 2 shows how the values of the three terms in $\widehat{\mathcal{L}}_{MASS}(\theta, \phi)$ change as the MLP network trains on the CIFAR-10 dataset using either the usual SoftmaxCE training or MASS training. Despite achieving similar accuracy, the SoftmaxCE training method does not seem to be implicitly performing MASS Learning, based on the differing values of the entropy (orange) and Jacobian (green) between the two methods as training progresses.

## 6. Conclusion

MASS Learning is a new approach to representation learning based on the goal of finding minimal achievable sufficient statistics. We have shown that networks trained by MASS Learning perform well on classification tasks and on regularization and uncertainty quantification benchmarks, despite not being directly formulated for any of these tasks.

There remain many open questions about MASS Learning. Of primary interest is more investigation into the properties of the representations learned by MASS Learning and how they differ from those learned in standard deep learning. There is also much to learn about how to best minimize the MASS loss. In this paper we used optimizer settings tuned for standard softmax cross entropy learning, but $\widehat{\mathcal{L}}_{MASS}(\theta, \phi)$ is such a different optimization objective that there are likely many potential improvements to be made in how we train the networks. We also plan to explore more expressive variational distributions $q_\phi$. Finally, in terms of efficiency, although MASS Learning is applicable in principle to any deep learning architecture, there is currently a significant computational cost in computing the $J_{f_\theta}$ term in the MASS Loss function. Finding non-invertible network architectures which admit more efficiently computable Jacobians, as is done in methods like normalizing flows (Rezende & Mohamed, 2015) or RealNVP (Dinh et al., 2016), would greatly increase the utility of MASS Learning.

## Acknowledgements

We would like to thank Georg Pichler, Thomas Vidick, Alex Alemi, Alessandro Achille, and Joseph Marino for useful discussions.

*Table 3.* Uncertainty quantification metrics (proper scoring rules) on CIFAR-10 using the SmallMLP network trained on 40,000 datapoints. Values are the mean over 4 training runs with different random seeds plus or minus the standard deviation. Emboldened values are those for which the minimum observed mean value in the column was within one standard deviation. Lower values are better.

| Method | Test Accuracy | NLL | Brier Score | Entropy |
|---|---|---|---|---|
| SoftmaxCE | $52.4 \pm 1.1$ | $4.19 \pm 0.15$ | $0.0835 \pm 0.0018$ | $0.230 \pm 0.003$ |
| SoftmaxCE, WD | $47.8 \pm 0.6$ | $\mathbf{1.47 \pm 0.02}$ | $0.0662 \pm 0.0006$ | $1.511 \pm 0.019$ |
| SoftmaxCE, D | $54.2 \pm 0.5$ | $1.56 \pm 0.01$ | $\mathbf{0.0642 \pm 0.0006}$ | $0.739 \pm 0.007$ |
| VIB, $\beta$=1e$-$1 | $46.4 \pm 0.6$ | $4.78 \pm 0.13$ | $0.0919 \pm 0.0009$ | $0.296 \pm 0.008$ |
| VIB, $\beta$=1e$-$2 | $51.6 \pm 0.4$ | $4.81 \pm 0.10$ | $0.0861 \pm 0.0006$ | $0.207 \pm 0.002$ |
| VIB, $\beta$=1e$-$3 | $51.7 \pm 0.7$ | $5.09 \pm 0.27$ | $0.0863 \pm 0.0013$ | $\mathbf{0.194 \pm 0.008}$ |
| VIB, $\beta$=1e$-$1, D | $49.8 \pm 0.1$ | $1.49 \pm 0.01$ | $0.0642 \pm 0.0001$ | $1.101 \pm 0.008$ |
| VIB, $\beta$=1e$-$2, D | $53.9 \pm 0.4$ | $1.52 \pm 0.00$ | $\mathbf{0.0636 \pm 0.0002}$ | $0.803 \pm 0.010$ |
| VIB, $\beta$=1e$-$3, D | $54.5 \pm 0.3$ | $1.53 \pm 0.01$ | $0.0641 \pm 0.0002$ | $0.754 \pm 0.009$ |
| MASS, $\beta$=1e$-$2 | $45.4 \pm 1.4$ | $6.85 \pm 0.26$ | $0.0979 \pm 0.0027$ | $0.207 \pm 0.007$ |
| MASS, $\beta$=1e$-$3 | $47.0 \pm 0.8$ | $5.85 \pm 0.24$ | $0.0943 \pm 0.0019$ | $0.218 \pm 0.007$ |
| MASS, $\beta$=1e$-$4 | $47.1 \pm 1.1$ | $5.71 \pm 0.25$ | $0.0942 \pm 0.0025$ | $0.219 \pm 0.006$ |
| MASS, $\beta$=0 | $47.0 \pm 0.6$ | $5.67 \pm 0.28$ | $0.0945 \pm 0.0019$ | $0.221 \pm 0.004$ |
| MASS, $\beta$=1e$-$2, D | $51.9 \pm 0.5$ | $1.60 \pm 0.03$ | $0.0662 \pm 0.0004$ | $0.846 \pm 0.025$ |
| MASS, $\beta$=1e$-$3, D | $53.0 \pm 0.5$ | $1.56 \pm 0.02$ | $0.0648 \pm 0.0008$ | $0.812 \pm 0.017$ |
| MASS, $\beta$=1e$-$4, D | $52.9 \pm 0.6$ | $1.55 \pm 0.02$ | $0.0646 \pm 0.0005$ | $0.831 \pm 0.020$ |
| MASS, $\beta$=0, D | $52.7 \pm 0.4$ | $1.55 \pm 0.02$ | $0.0648 \pm 0.0004$ | $0.832 \pm 0.012$ |

*Table 4.* Uncertainty quantification metrics (proper scoring rules) on CIFAR-10 using the ResNet20 network trained on 40,000 datapoints. Values are the mean over 4 training runs with different random seeds plus or minus the standard deviation. Emboldened values are those for which the minimum observed mean value in the column was within one standard deviation. Lower values are better.

| Method | Test Accuracy | NLL | Brier Score | Entropy |
|---|---|---|---|---|
| SoftmaxCE | $67.8 \pm 2.7$ | $1.98 \pm 0.15$ | $\mathbf{0.0546 \pm 0.0043}$ | $0.209 \pm 0.021$ |
| VIB, $\beta$=1e$-$3 | $66.0 \pm 0.6$ | $2.28 \pm 0.12$ | $0.0577 \pm 0.0011$ | $0.210 \pm 0.004$ |
| VIB, $\beta$=1e$-$4 | $67.1 \pm 0.6$ | $2.23 \pm 0.07$ | $0.0563 \pm 0.0010$ | $0.196 \pm 0.003$ |
| VIB, $\beta$=1e$-$5 | $67.8 \pm 0.6$ | $2.35 \pm 0.11$ | $0.0559 \pm 0.0012$ | $0.175 \pm 0.003$ |
| VIB, $\beta$=0 | $68.0 \pm 0.1$ | $2.45 \pm 0.05$ | $0.0558 \pm 0.0003$ | $\mathbf{0.167 \pm 0.003}$ |
| MASS, $\beta$=1e$-$3 | $67.1 \pm 0.5$ | $\mathbf{1.77 \pm 0.03}$ | $0.0555 \pm 0.0010$ | $0.227 \pm 0.006$ |
| MASS, $\beta$=1e$-$4 | $68.9 \pm 1.1$ | $1.91 \pm 0.07$ | $\mathbf{0.0533 \pm 0.0018}$ | $0.193 \pm 0.011$ |
| MASS, $\beta$=1e$-$5 | $69.5 \pm 0.6$ | $1.96 \pm 0.05$ | $\mathbf{0.0522 \pm 0.0011}$ | $0.188 \pm 0.007$ |
| MASS, $\beta$=0 | $69.0 \pm 0.8$ | $2.00 \pm 0.08$ | $\mathbf{0.0528 \pm 0.0015}$ | $0.190 \pm 0.003$ |

*Table 5.* Out-of-distribution detection metrics on CIFAR-10 with SVHN digits as the out-of-distribution examples using ResNet20 network trained on 40,000 datapoints. Values are the mean over 4 training runs with different random seeds plus or minus the standard deviation. Emboldened values are those for which the maximum observed mean value in the column was within one standard deviation. Higher values are better.

| Training Method | Test Accuracy | Detection Method | AUROC | APR In | APR Out |
|---|---|---|---|---|---|
| SoftmaxCE | $67.8 \pm 2.7$ | Entropy | $0.62 \pm 0.01$ | $0.66 \pm 0.02$ | $0.57 \pm 0.01$ |
| | | $\max_i q_\phi(f_\theta(x)\vert y_i)$ | $0.72 \pm 0.02$ | $\mathbf{0.73 \pm 0.03}$ | $0.70 \pm 0.03$ |
| VIB, $\beta$=1e$-$3 | $66.0 \pm 0.6$ | Entropy | $0.57 \pm 0.01$ | $0.60 \pm 0.01$ | $0.53 \pm 0.01$ |
| | | Rate | $0.71 \pm 0.03$ | $0.71 \pm 0.03$ | $0.69 \pm 0.02$ |
| VIB, $\beta$=1e$-$4 | $67.1 \pm 0.6$ | Entropy | $0.57 \pm 0.02$ | $0.59 \pm 0.03$ | $0.53 \pm 0.01$ |
| | | Rate | $\mathbf{0.72 \pm 0.04}$ | $\mathbf{0.71 \pm 0.05}$ | $0.70 \pm 0.04$ |
| VIB, $\beta$=1e$-$5 | $67.8 \pm 0.6$ | Entropy | $0.56 \pm 0.04$ | $0.58 \pm 0.05$ | $0.53 \pm 0.02$ |
| | | Rate | $0.68 \pm 0.01$ | $0.68 \pm 0.01$ | $0.64 \pm 0.02$ |
| VIB, $\beta$=0 | $68.0 \pm 0.1$ | Entropy | $0.60 \pm 0.03$ | $0.63 \pm 0.04$ | $0.55 \pm 0.02$ |
| | | Rate | $0.61 \pm 0.03$ | $0.60 \pm 0.02$ | $0.57 \pm 0.04$ |
| MASS, $\beta$=1e$-$3 | $67.1 \pm 0.5$ | Entropy | $0.63 \pm 0.02$ | $0.68 \pm 0.02$ | $0.57 \pm 0.02$ |
| | | $\max_i q_\phi(f_\theta(x)\vert y_i)$ | $0.69 \pm 0.02$ | $0.68 \pm 0.02$ | $0.68 \pm 0.02$ |
| MASS, $\beta$=1e$-$4 | $68.9 \pm 1.1$ | Entropy | $0.64 \pm 0.01$ | $0.69 \pm 0.01$ | $0.58 \pm 0.01$ |
| | | $\max_i q_\phi(f_\theta(x)\vert y_i)$ | $0.74 \pm 0.01$ | $0.73 \pm 0.01$ | $0.72 \pm 0.02$ |
| MASS, $\beta$=1e$-$5 | $69.5 \pm 0.6$ | Entropy | $0.64 \pm 0.01$ | $0.68 \pm 0.01$ | $0.58 \pm 0.01$ |
| | | $\max_i q_\phi(f_\theta(x)\vert y_i)$ | $\mathbf{0.76 \pm 0.04}$ | $\mathbf{0.75 \pm 0.04}$ | $\mathbf{0.75 \pm 0.04}$ |
| MASS, $\beta$=0 | $69.0 \pm 0.8$ | Entropy | $0.65 \pm 0.01$ | $0.69 \pm 0.02$ | $0.59 \pm 0.01$ |
| | | $\max_i q_\phi(f_\theta(x)\vert y_i)$ | $\mathbf{0.76 \pm 0.03}$ | $\mathbf{0.76 \pm 0.03}$ | $\mathbf{0.75 \pm 0.03}$ |

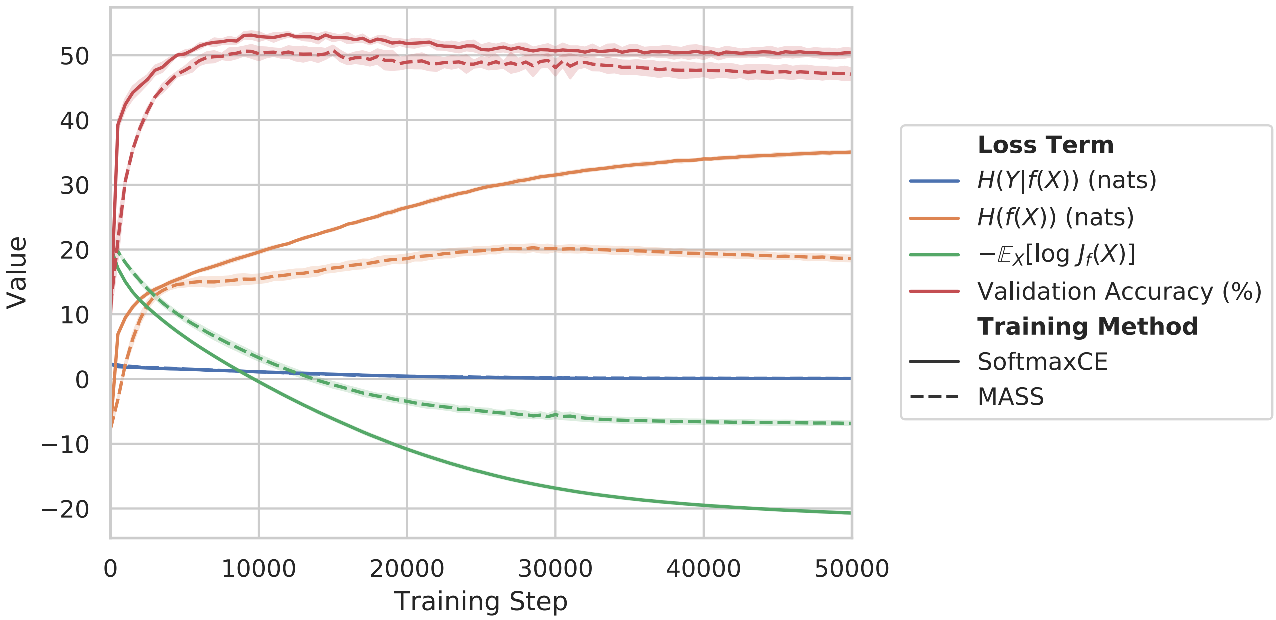

*Figure 2.* Value of each term in the MASS Learning loss function, $\mathcal{L}_{MASS}(f) = H(Y|f(X)) + \beta H(f(X)) - \beta \mathbb{E}_X[\log J_f(X)]$, during training of the SmallMLP network on the CIFAR-10 dataset. The MASS training was performed with $\beta = 0.001$, though the plotted values are for the terms without being multiplied by the $\beta$ coefficients. The values of these terms for SoftmaxCE training are estimated using a variational distribution $q_\phi(x|y)$, the parameters of which were estimated at each timestep by MLE over the training data.

## References

Achille, A. and Soatto, S. Emergence of Invariance and Disentanglement in Deep Representations. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9, February 2018a. doi: 10.1109/ITA.2018.8503149.

Achille, A. and Soatto, S. Information Dropout: Learning Optimal Representations Through Noisy Computation, 2018b.

Adragni, Kofi P. and Cook, R. Dennis. Sufficient dimension reduction and prediction in regression. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4385–4405, November 2009. doi: 10.1098/rsta.2009.0110. URL https://royalsocietypublishing.org/doi/full/10.1098/rsta.2009.0110.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep Variational Information Bottleneck. *arXiv:1612.00410 [cs, math]*, December 2016. URL http://arxiv.org/abs/1612.00410. arXiv: 1612.00410.

Alemi, A. A., Fischer, I., and Dillon, J. V. Uncertainty in the Variational Information Bottleneck. *arXiv:1807.00906 [cs, stat]*, July 2018. URL http://arxiv.org/abs/1807.00906. arXiv: 1807.00906.

Amjad, R. A. and Geiger, B. C. Learning Representations for Neural Network-Based Classification Using the Information Bottleneck Principle. *arXiv:1802.09766 [cs, math]*, February 2018. URL http://arxiv.org/abs/1802.09766. arXiv: 1802.09766.

Bell, A. J. and Sejnowski, T. J. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, November 1995. ISSN 0899-7667.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv:1511.07289 [cs]*, November 2015. URL http://arxiv.org/abs/1511.07289. arXiv: 1511.07289.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory 2nd Edition*. Wiley-Interscience, Hoboken, NJ, 2 edition edition, July 2006. ISBN 978-0-471-24195-9.

Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear Independent Components Estimation. *arXiv:1410.8516 [cs]*, October 2014. URL http://arxiv.org/abs/1410.8516. arXiv: 1410.8516.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. May 2016. URL https://arxiv.org/abs/1605.08803.

Dynkin, E. B. Necessary and sufficient statistics for afamily of probability distributions. *Uspekhi Mat. Nauk*, 6(1): 68–90, 1951. URL http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=rm&paperid=6820&option_lang=eng.

Federer, H. *Geometric Measure Theory*. Springer, New York, NY, 1969.

Goldfeld, Z., Berg, E. v. d., Greenewald, K., Melnyk, I., Nguyen, N., Kingsbury, B., and Polyanskiy, Y. Estimating Information Flow in Neural Networks. *arXiv:1810.05728 [cs, stat]*, October 2018. URL http://arxiv.org/abs/1810.05728. arXiv: 1810.05728.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. URL http://arxiv.org/abs/1512.03385. arXiv: 1512.03385.

Hendrycks, D. and Gimpel, K. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. October 2016. URL https://arxiv.org/abs/1610.02136v3.

Jakubovitz, D. and Giryes, R. Improving DNN Robustness to Adversarial Attacks using Jacobian Regularization. *arXiv:1803.08680 [cs, stat]*, March 2018. URL http://arxiv.org/abs/1803.08680. arXiv: 1803.08680.

James, R. G., Mahoney, J. R., and Crutchfield, J. P. Trimming the Independent Fat: Sufficient Statistics, Mutual Information, and Predictability from Effective Channel States. *Physical Review E*, 95(6), June 2017. ISSN 2470-0045, 2470-0053. doi: 10.1103/PhysRevE.95.060102. URL http://arxiv.org/abs/1702.01831. arXiv: 1702.01831.

Kingma, D. and Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, December 2014. URL http://arxiv.org/abs/1412.6980. arXiv: 1412.6980.

Kolchinsky, A., Tracey, B. D., and Wolpert, D. H. Nonlinear Information Bottleneck. *arXiv:1705.02436 [cs, math, stat]*, May 2017. URL http://arxiv.org/abs/1705.02436. arXiv: 1705.02436.

Kolchinsky, A., Tracey, B. D., and Van Kuyk, S. Caveats for information bottleneck in deterministic scenarios. *arXiv:1808.07593 [cs, stat]*, August 2018. URL http://arxiv.org/abs/1808.07593. arXiv: 1808.07593.

Koliander, G., Pichler, G., Riegler, E., and Hlawatsch, F. Entropy and Source Coding for Integer-Dimensional Singular Random Variables. *IEEE Transactions on Information Theory*, 62(11):6124–6154, November 2016. ISSN 0018-9448, 1557-9654. doi: 10.1109/TIT.2016.2604248. URL http://arxiv.org/abs/1505.03337. arXiv: 1505.03337.

Krantz, S. G. and Parks, H. R. *Geometric Integration Theory*. Birkhuser, Basel, Switzerland, 2009.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *arXiv:1612.01474 [cs, stat]*, December 2016. URL http://arxiv.org/abs/1612.01474. arXiv: 1612.01474.

Lehmann, E. L. and Scheffe, H. Completeness, Similar Regions, and Unbiased Estimation: Part I. *Sankhy: The Indian Journal of Statistics (1933-1960)*, 10(4):305–340, 1950. ISSN 0036-4452. URL https://www.jstor.org/stable/25048038.

Nash, C., Kushman, N., and Williams, C. K. I. Inverting Supervised Representations with Autoregressive Neural Density Models. June 2018. URL https://arxiv.org/abs/1806.00400.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Sensitivity and Generalization in Neural Networks: an Empirical Study. *arXiv:1802.08760 [cs, stat]*, February 2018. URL http://arxiv.org/abs/1802.08760. arXiv: 1802.08760.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in PyTorch. In *NIPS-W*, 2017.

Rezende, D. J. and Mohamed, S. Variational Inference with Normalizing Flows. *arXiv:1505.05770 [cs, stat]*, May 2015. URL http://arxiv.org/abs/1505.05770. arXiv: 1505.05770.

Rifai, S., Mesnil, G., Vincent, P., Muller, X., Bengio, Y., Dauphin, Y., and Glorot, X. Higher Order Contractive Auto-Encoder. In Gunopulos, D., Hofmann, T., Malerba, D., and Vazirgiannis, M. (eds.), *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pp. 645–660. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23783-6.

Ross, A. S. and Doshi-Velez, F. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients. *arXiv:1711.09404 [cs]*, November 2017. URL http://arxiv.org/abs/1711.09404. arXiv: 1711.09404.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. On the Information Bottleneck Theory of Deep Learning. February 2018. URL https://openreview.net/forum?id=ry_WPG-A-.

Shamir, O., Sabato, S., and Tishby, N. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29):2696–2711, June 2010. ISSN 0304-3975. doi: 10.1016/j.tcs.2010.04.006. URL http://www.sciencedirect.com/science/article/pii/S030439751000201X.

Shwartz-Ziv, R. and Tishby, N. Opening the Black Box of Deep Neural Networks via Information. *arXiv:1703.00810 [cs]*, March 2017. URL http://arxiv.org/abs/1703.00810. arXiv: 1703.00810.

Sokolic, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. D. Robust Large Margin Deep Neural Networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, August 2017. ISSN 1053-587X, 1941-0476. doi: 10.1109/TSP.2017.2708039. URL http://arxiv.org/abs/1605.08254. arXiv: 1605.08254.

Strouse, D. J. and Schwab, D. J. The deterministic information bottleneck. April 2016. URL https://arxiv.org/abs/1604.00268.

Tishby, N. and Zaslavsky, N. Deep Learning and the Information Bottleneck Principle. March 2015. URL https://arxiv.org/abs/1503.02406.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv:physics/0004057*, April 2000. URL http://arxiv.org/abs/physics/0004057. arXiv: physics/0004057.

Varga, D., Csiszrik, A., and Zombori, Z. Gradient Regularization Improves Accuracy of Discriminative Models. *arXiv:1712.09936 [cs]*, December 2017. URL http://arxiv.org/abs/1712.09936. arXiv: 1712.09936.