A RATE-DISTORTION THEORY OF ADVERSARIAL EXAMPLES

Anonymous authors
Paper under double-blind review

ABSTRACT

The generalization ability of deep neural networks (DNNs) is interwined with model complexity, robustness and capacity. We employ information theory to establish an equivalence between a DNN and a noisy communication channel, and obtain a notion of capacity that allows us to characterize generalization behavior of DNNs for adversarial inputs.

1 INTRODUCTION

The mathematical concept of information, as introduced by Shannon (1948) in the context of communication over noisy channels, has revolutionized fields like psychology and neuroscience. Despite the fact that deep learning has been heavily influenced by these fields, information theory is not the established framework for its development. Built on mainly empirically derived design principles, modern neural networks achieve impressive results. However, a theoretical understanding as to when good performance can be expected is lacking (Zhang et al., 2017; Arpit et al., 2017).

In particular, the inability to characterize generalization imposes hard constraints on the responsible deployment of deep learning in performance-critical settings where fault tolerance is required. The challenge of obtaining such a characterization arises from the difficulty of interpreting (Bau et al., 2017) and quantifying the uncertainty of the predictions (Gal, 2016; Guo et al., 2017). Furthermore, the adversarial examples phenomenon (Biggio et al., 2013; Szegedy et al., 2014; Nguyen et al., 2015; Athalye et al., 2018) shows that solely maximizing test accuracy can result in potentially unsafe model behavior.

Many studies have attempted to understand generalization via model complexity and introduced various measures to quantify it. However, Neyshabur et al. (2017) concludes that a combination of several measures is required to explain some of the empirically observed phenomena. A more compact description of generalization was proposed by Tishby & Zaslavsky (2015), who analyzed the learning problem in terms of mutual information. This measure incorporates many of the previously proposed characteristics, effectively summarizing the interplay between the data, network architecture, and the optimization procedure.

We adopt this view for the image classification problem which leads to an insightful understanding of empirical findings, including adversarial examples and regularization techniques. In particular, this work makes the following contributions:

1. We formalize the trade-off between a model's prediction accuracy and sensitivity to adversarial examples through rate-distortion theory. Practically speaking, this is the information bottleneck (IB) trade-off in the context of supervised deep learning for image classification. We attribute sensitivity to adversarial examples to excess rate, i.e., complexity of input representations.
2. We explain the mechanisms by which explicit regularization strategies determine the model's capacity, drawing on an intuitive sphere packing argument for the Gaussian communication channel. Weight decay has a natural interpretation as a power constraint on the channel. Batch normalization acts as an information-theoretic "short-circuit" between adjacent representations, easing the flow of relevant information between the layers and accelerating convergence, but making it strictly more difficult to successively refine information, which is essential for minimizing excess rate.

3. We demonstrate through experiments a reasonable trade-off between the test accuracy and robustness towards adversarial attacks for the Street View House Numbers (SVHN) dataset. We propose evaluating robustness in terms of *fault tolerance* by plotting the information transmission rate versus the signal-to-noise ratio for unbounded and unseen "worst case" noise, and by demonstrating successful communication across the channel in the reverse direction, i.e, by generating plausible messages from only the label.

2 THEORY

We begin by formulating the machine learning problem in the framework of information theory, focusing in particular on image classification, with the model being a feedforward deep neural network (DNN) trained in the supervised setting. We find that this problem has a natural interpretation as that of communication over noisy channels and use it as a concrete example for the following analysis (which is, however, not limited to image classification). We assume basic familiarity with information-theoretic quantities, such as entropy and mutual information.

The input (images) and output (labels) to the communication channel are represented by the random variables $X \in \mathbb{R}^{h \times w \times c}$ and $Y \in \{0, 1, \ldots, n_c\}$, respectively. The goal of the classification task is to adjust the parameters of the model such that the prediction $\hat{Y}$ given X is as close as possible to the true label Y for unseen inputs. In the communication setting, the intended message is the label Y. We model each of the l layers in a DNN as a continuous communication channel, and describe the corresponding output as a random variable T_j, $j = 1, \ldots, n_l$. The DNN is the series connection of such channels. The image X can be interpreted as a highly redundant and noisy encoding of Y. The decoded message $\hat{Y}$ is obtained from the final layer T_{n_l}. This mapping is formally characterized through the conditional probability distribution $P(\hat{Y}|X)$.

A central quantity characterizing a physical channel is its capacity, defined as the maximum transmission rate at which communication is possible with arbitrarily low probability of error. In practice, channel capacity is determined by two factors: noise, e.g., from the environment or physical properties of materials, and the available power. Consider that in order to transmit information over a channel, the message has first to be encoded in terms of signals that the channel can actually represent – for example, voltage pulses of varying amplitude. These signals are composed to sequences of length n, or "codewords", used to represent the message. An ideal continuous channel allows for infinitely many codewords that are all uniquely decodable, and thus error-free information transmission. In the presence of noise and the power constraint, decodeability cannot be achieved with certainty.

The most common model used in theory is the Gaussian channel with additive white noise $Z \sim \mathcal{N}(0, N)$ with variance N, depicted in Figure 1(a), with a constraint on the average power of the signal, proportional to the signal amplitude x_i^2:

$$\frac{1}{n} \sum_{i=1}^{n} x_i^2 \leq P. \tag{1}$$

Shannon (1948) proved that the information capacity of the Gaussian channel is a function of the signal-to-noise ratio (SNR) P/N:

$$C = \frac{1}{2} \log_2 \left(1 + \frac{P}{N} \right) \qquad \text{bits per transmission.} \tag{2}$$

In image classification setting, the noise is already built into X, as shown in Figure 1(b). By assumption, a learnable classification task must have a causal relationship between input and output, and the ideal way to learn the rule is by uncovering this causal relationship. The fact that we wish to transmit only the message, modeled as a latent cause variable Φ, observable only *with noise* through X – but not X itself – is an essential detail that lets us reason about deterministic neural network layers as noisy channels, even though there is no "new" noise added in the channel itself after the input layer. That is, the hidden layers of the DNN are not stochastic[1].

[1] We acknowledge that stochastic feed-forward DNNs are considered in the literature (c.f. Tang & Salakhutdinov, 2013) but only consider the more typical deterministic layers in our work.

The values of DNN input data are usually bounded, e.g., $x \in [0, 1]$ for image pixels. The weighted input of layer j corresponds to the input for channel j. A power constraint (1) on the input to the channel is therefore equivalent to a squared L_2 norm constraint on the weights, or the traditional weight decay.

The maximal number of codewords that can be decoded with an arbitrarily small probability of error under these constraints can be estimated via a simple geometrical argument: An input sequence $x = \{x_i\}_{i=1}^n$ and the corresponding received sequence $y = \{y_i\}_{i=1}^n = \{(x_i + z_i)\}_{i=1}^n$ can be represented as two points in an n-dimensional space of encodings. The average power constraint forces x to lie inside a sphere of radius $\sqrt{nP}$ centered around the origin. Since y differs from x only due to noise with variance N, it will be found with high probability in a sphere of radius $\sqrt{nN}$ centered around x. In this way, the sphere with radius $\sqrt{n(P + N)}$ is partitioned into decoding spheres with radius $\sqrt{nN}$, one for each codeword x. The maximum number of non-confusable codewords M is given by the number of non-intersecting decoding spheres that can be "packed" into the large sphere, obtained from the ratio of sphere volumes:

$$M = \left(\frac{P + N}{N}\right)^{n/2} = 2^{\frac{n}{2}\log_2\left(1+\frac{P}{N}\right)}. \tag{3}$$

The exponent in the final expression is precisely nC, since capacity C is the maximal achievable rate.

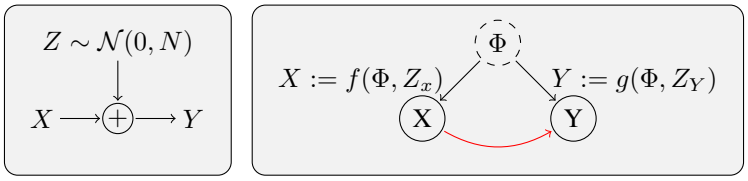
This geometric view can be extended to understand the role that capacity plays in the deep learning context: We say that the model has learned the rule if the inputs are mapped to k non-overlapping clusters, each corresponding to the associated label. Achieving this already means a perfect solution to the classification task in the sense of arbitrarily small prediction error. However, this is not sufficient for robustness or generalization in the stricter sense: Regions of the encoding space that are not assigned to any label correspond to excess capacity, meaning that the channel encodings are more complex than required for the given communication problem. To achieve robustness, we have to minimize the excess capacity, or, in terms of the geometric interpretation, we need to maximally fill out the available encoding space. The ideal result with densest packing of the spherical volume requires the n_c non-overlapping clusters to be spherical. Spherical cluster shape can be achieved either by whitening (i.e., sphering) the input, or by learning projections that explain the natural variability in the data to the extent that it is relevant.

Once the power budget is corrected, the representation learning problem is that of finding projections that rotate and reflect the weighted input into compact volumes that can be partitioned homogeneously with respect to the label. The trade-off is that making the class-specific volumes more compact requires compression of information, but we need to preserve enough information to distribute the decoding spheres among the available representation space such that they are all non-overlapping. This illustrates how the problem of reliable communication over a noisy channel is essentially that of optimal information compression. The rate R is a function of the number of codewords M and their length n: $R = \log(M)/n$. Increasing M for fixed channel properties (P and N) will increase the overlap between decoding spheres, while decreasing n corresponds to compressing the input representation. Therefore, increasing R is only possible at the cost of transmission accuracy. This fundamental trade-off is characterized analytically by rate-distortion (RD) theory in the form of a constrained optimization problem. The RD theory introduces a distortion measure $d(x, t)$ that quantifies the distance between a random variable X and its representation T, and describes the rate R as a function of the average distortion D.

A major drawback of RD theory is that it does not specify how to choose d. The IB method, which can be seen as a generalization of RD theory, answers this question by introducing the concept of the relevant variable Y. In this setting, the representation T is required to maximally preserve information that X carries about Y (Tishby et al., 1999). This way, the variational problem is expressed solely in terms of mutual information:

$$\mathcal{L} = I(X; T) - \beta I(Y; T), \tag{4}$$

where $I(X; T)$ measures the compression of X and β is a positive parameter associated with the constraint on the preserved information $I(Y; T)$. Thus, the solution to (4) characterizes an optimal trade-off between compressing X and preserving the relevant information it carries.



(a) (b)

Figure 1: (a) The additive white Gaussian noise (AWGN) communication channel (Cover & Thomas, 1991). The additive noise Z, originating from various sources in the environment, is assumed to be independent of X. (b) Structural causal model (SCM) of a pattern recognition task, with the intended message as a latent random variable Φ, causal mechanisms f, g, and noise variables Z_x, Z_y; adapted from Peters et al. (2017). The red arrow from X to Y represents the DNN channel.

It turns out that it is more efficient to split up the task of encoding the message into multiple stages via a *successive refinement* of information (Equitz & Cover, 1991). In DNNs, each of the "stages" are the hidden layers' representations T of the input X, which introduce a partition of the input space w.r.t. the relevant variable Y and coarsen it gradually in the course of learning. This implies that we should be able to decode the relevant variable Y from *any* of the layers T_j, inducing a distortion D_j, and the description will be optimal in the RD sense at each layer. This corresponds to the case of no excess rate.

Remarkably, it has been suggested that optimizing all layers in a DNN simultaneously with plain stochastic gradient descent (SGD) is sufficient to achieve successive refinement. The process is characterized by a brief fitting or memorization phase where $I(X;T)$ and $I(Y;T)$ increase monotonically – consistent with empirical observations that DNNs begin memorizing after the first epoch (Carlini et al., 2018) – and followed by a compression phase where $I(X;T)$ decreases. The aim of compression phase is that $I(Y;T)$ should be non-decreasing, but this is not guaranteed – it is possible to overfit by compressing too much (Schwartz-Ziv & Tishby, 2017). To check the degree of successive refinement, and ultimately verify the utility of each of the layers, we can inspect the information coordinates $I(X;T_j), I(T_j;Y)$ of each layer (Tishby & Zaslavsky, 2015).

It is argued that the primary mechanism facilitating compression is the inherent noise in the dataset, which increases the entropy of the weights subject to a constraint on the empirical risk. This entropy increase is unfortunately logarithmic in the number of epochs; however, this is consistent with the observations that we may reap substantial benefits by continuing to train well after the empirical loss is minimized, e.g., for obtaining a maximum margin solution (Soudry et al., 2018). Given that we do not typically wish to wait so long, it seems prudent to apply preprocessing in practice, e.g., by reducing $H(X|Y)$, and especially when concerned with maximizing out-of-sample performance, given that SGD will not forget confounding spurious correlations in the data.

If there are no constraints on the complexity of the system, there is no need for successive refinement: relevant information can always be encoded with sub-optimal representations at a higher cost, e.g., through some degree of memorization. Such a solution achieves small distortion (error) with significant excess rate (complexity), making it easier to transmit messages along the channel which are decoded arbitrarily at test time – i.e., adversarial examples.

3 Implications

The interplay between the three fundamental quantities n, N, and P discussed in the sphere packing argument for the Gaussian channel is the key for a clear understanding of works that study the generalization performance of DNNs.

3.1 Effective Capacity

Bartlett (1998) showed that if an l-layer network can be found with small error on the training set, then the generalization performance depends on the size of the weights rather than the number. This

is evident from the sphere packing illustration: The norm of the weights affects P only, and by extension, the volume of the outer sphere. The number of weights n scales both the small and large sphere proportionally. Thus, assuming the case that the model can learn the rule at all, changing P significantly impacts the excess rate, wheras n does not.

Neyshabur et al. (2015) show that capacity cannot be controlled in general through P alone for unbounded n. In a linear model, however, n is set by the data and is therefore finite, so we can always bound capacity through P.

The experiments of Zhang et al. (2017) raise the question: Should we be surprised if a particular architecture – with n larger than the number of examples – fits random labels? If we bound P – yes, otherwise – no. We cannot establish to which extent Zhang et al. limit P, and thus the extent to which we would expect their models to generalize. Generalization gap in terms of accuracy is only a proxy for that in terms of the loss, which is a signal more sensitive to P. Furthermore, many of their architectures use batch normalization layers that introduce a learnable scaling parameter that is not penalized, and therefore the capacity is infinite.

Implicit regularization by early stopping is an obvious form of capacity control in that P will be finite, but this addresses only symptoms of a poorly posed learning task – not the cause.

Saxe et al. (2018) fail to observe compression for models using the the ReLU activation function. This is because the capacity is indeed infinite as they do not bound P, e.g., with weight decay. Remarkably, it is still possible to observe compression with ReLUs without weight decay, but we should not count on this in general as such behaviour depends on many factors including the SNR in the dataset (where "signal" is w.r.t. the relevant variable), the parameter initialization scheme, and the batch size.

3.2 Adversarial Examples

The existence of a trade-off between a model's prediction accuracy and sensitivity to adversarial examples – where arbitrarily low error implies greater sensitivity – is not yet universally accepted.

Tanay & Griffin (2016) proposed a taxonomy of adversarial examples affecting linear models, which suggested a basic trade-off. They define "Type 1" examples that affect *optimal classifiers*, such that "the inconvenience of their existence is balanced by the performance gains allowed". This view has been maintained for DNNs (Galloway et al., 2018) and advanced more formally by Tsipras et al. (2018).

On the other hand, Gilmer et al. (2018) examine a situation for which *non-zero* error *implies* that a model is sensitive to small perturbations. There is a simple explanation for the apparent contradiction: the synthetic dataset considered by Gilmer et al. has no noise, so there is no rate-distortion trade-off, and the optimal strategy is to drive the error to zero. However, this is not representative of computer vision for natural images, where the intended message is always observed with noise.

Dube (2018) draws on high-dimensional geometry and attributes adversarial examples to "negative space" that is unoccupied by legitimate image manifolds; this can be interpreted as excess rate in the context of RD theory and our channel analogy. Alemi et al. (2017) make a direct connection between the IB principle and adversarial examples, and tune β in the constrained optimization given by (4), but they do not formalize the problem in terms of excess rate. That is, tuning β chooses the RD optimal trade-off, but is not a regularizer on its own Chechik et al. (2005).

Chalupka et al. (2015) suggest partitioning the information that X contains about a relevance variable Y into visual causes Φ, and spurious correlates S. We attribute the most impressive adversarial examples – in the sense that they are indistinguishable to the human eye – to the model fitting spurious correlates, which are not typically removed when optimizing for high accuracy. We suggest that Φ is equivalent to robust primary features, and S the weak secondary predictors, recently identified by Tsipras et al. (2018) and Tanay et al. (2018). Therefore, we would like to discard as much of S, while retaining as much of Φ, as possible. This implies a drop in prediction accuracy for the training set, and possibly even the held-out test set. The payoff is conferred via strong generalization, and some degree of fault tolerance, i.e., graceful failure, for worst-case inputs.

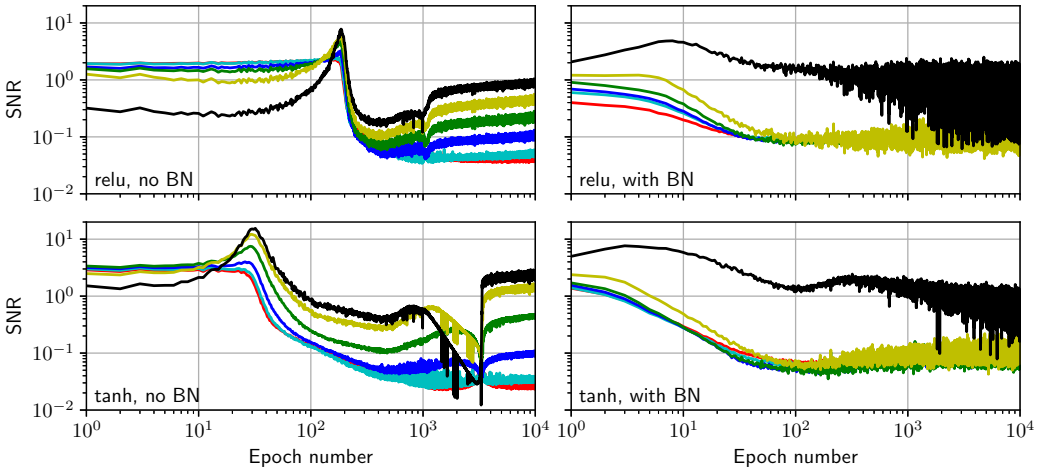


Figure 2: The SNR for the stochastic gradient of the cross-entropy loss $\mathcal{L}$ w.r.t. the weights versus epochs, with $\|\operatorname{mean}(\nabla_w \mathcal{L})\|_2 / \|w\|_2$ as signal and $\|\operatorname{std}(\nabla_w \mathcal{L})\|_2$ as noise. We use constant learning rate SGD with a mini-batch size of 100, ReLU and tanh activation functions, with and without batch normalization layers on all but the last layer. Best viewed in colour.

4 Experiments

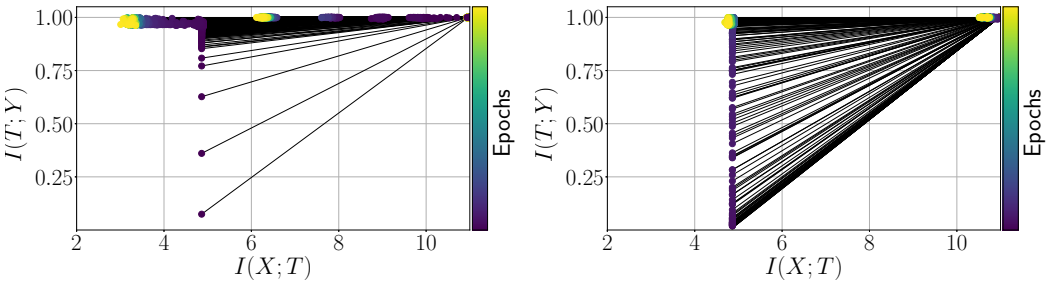
The purpose of the experiments is to show how we can achieve stronger fault tolerance through careful parameterization of SGD and training to convergence. We first establish that convergence can be observed in terms of SNR in the weight updates. We then show how explicit regularization changes the optimization dynamics.

4.1 The Convergence of SGD via the Gradients' Signal-to-Noise Ratio

In this section we examine the convergence properties of an over-parameterized six-layer fully-connected MLP trained on MNIST 3 versus 7 with SGD. In Figure 2 we show the SNR for the stochastic gradients with respect to the parameters of this model. As suggested by Schwartz-Ziv & Tishby (2017), the SNR does eventually become roughly constant in all cases, regardless of the choice of nonlinear activation function or explicit regularization via batch normalization (BN). The number of epochs required to achieve this, however, differs dramatically: it is 1,000 for ReLU, 3,000 for tanh, but only 100 when BN is applied. The layers' SNR levels in the model without BN are distributed more uniformly along the y-axis and converge to a different values, wheras in case of BN, the curves for all layers collapse.

For both nonlinearities without BN, the difference between the initial and final SNR level is smallest for the last layer (black), and largest for the earlier layers (e.g., red, cyan). Recall that the "signal" is a function of how much information the layer preserves about the relevant variable Y. At random initialization, the layers close to the input preserve all information about Y, while the last layer has the least. On convergence, all layers have roughly the same information about Y if we neglect minor losses due to the data processing inequality, but the last layer maintains the least information about the input, and therefore has the least "noise". Thus, the trajectories in the information plane shown by Schwartz-Ziv & Tishby and in Section 4.2 of the present work, are closely related to the gradients' SNR, which predicts the sorting of the layers in descending order as observed on the left-side panels in Figure 2.

The fundamental difference between normalizing the input vs. the hidden layers, is that we only *linearly* decorrelate the input, which preserves nonlinear structure, e.g., edges, such that it is possible to spread out the decoding spheres that initially cluster around the origin and overlap. With BN, we do this normalization post nonlinearity, which is problematic for successive refinement. Futhermore, it is irrelevant if the individual neurons maintain some nonlinear effect; if the cumulative effect of the neurons in the layer is to make T more normal, then this description becomes less refineable,



(a) (b)

Figure 3: Information plane visualization for the model from Section 4.1 and the relu activation function trained on the original labels (a), and random labels (b). Intuitively, we only see compression for the original labels, wheras the model memorizes in the random label case. The degree of memorization is roughly the difference in $I(X;T)$ values between the two cases. Eventually, we would expect to see some compression in (b) if we had continued to train for many more epochs than are required for fitting the original labels.

and cannot be successively refined at all if it is exactly normal and the block length is one (Equitz & Cover, 1991). We do have a block length of one in a feedforward DNN, because we consider transmission w.r.t. one use of the channel.

4.2 Memorization is the Lack of Compression

Arpit et al. (2017) use a working definition for memorization as "the behavior exhibited by DNNs trained on noise". Through information theory, we can provide a more quantitative definition: Memorization is the difference between the realized $I(X;T)$ – i.e., the mutual information between the output of layer T and input X – from the IB-optimal value. In other words, the degree of memorization is given by how much compression did not occur, relative to what was theoretically possible. Importantly, this quantity has well-defined upper and lower bounds. We can estimate the lower bound $\hat{I}(X;T)$ through the data processing inequality (DPI) and plug-in a maximum likelihood estimator on samples from the training set. However, we can also upper-bound $I(X;T)$ by $H(X)$, for which a conservative estimate can always be obtained by treating the pixels as spatially independent. For example, an image with 8-bit pixels has a maximum absolute entropy of 8-bits per pixel, achieved if and only if the pixels are distributed uniformly.

Experimentally, memorization is characterized by the absence of a phase transition in the information plane, as shown in Figure 3 for original versus random labels. Since the activation function is not bounded, we estimate the information with a maximum entropy adaptive binning scheme that bins the CDF for each layer equally by recomputing the bin ranges every epoch. We use 30 bins and 2,000 samples from the training set. See Darbellay & Vajda (1999) and Paninski (2003) for an overview of such methods.

4.3 Explicit Regularization and the Generalization Gap

In the next two sections we focus on the Street View House Numbers (SVHN) dataset (Netzer et al., 2011), which is an appropriate dataset for characterizing adversarial robustness since compelling results have already been shown for MNIST (Tanay & Griffin, 2016; Schott et al., 2018). Learning the SVHN dataset can be considered more "difficult" than MNIST due to having significantly more noise, e.g., from the "distracting" digits in addition to the relevant one, and occupying a larger RGB canvas in $R^{32\times32\times3}$. Our primary objective is not to construct an "adversarial defense", but simply to maximize out-of-sample performance, i.e, generalize to the global population of house numbers using Arabic numerals. We characterize the fault tolerance of our models for adversarial examples in Section 4.4.

Intuitively, colour and texture are not legitimate causes of a digit's class, so we first convert from RGB to grayscale (NTSC). We also apply PCA, retaining the top 400 principal components, and

Table 1: Test accuracy (Test (%)) and generalization gap (Gap (%)) for A: the baseline, B: A + weight decay (WD), and C: B + batch normalization (BN). There is no large-batch generalization gap (LBGG) for B, however a small LBGG appears when we introduce BN in C. Not shown: the loss gap is one order of magnitude less for B vs. A. Training accuracy is omitted for brevity. Model A achieves 100±0% training accuracy. We report the mean accuracy and standard error of the mean, assuming the error is normally distributed and independent over 5 seeds.

	A		B		C	
Batch Size	Test (%)	Gap (%)	Test (%)	Gap (%)	Test (%)	Gap (%)
128	85.9 ± 0.2	14.1 ± 0.2	88.4 ± 0.2	4.8 ± 0.1	87.1 ± 0.4	12.42 ± 0.04
64	86.3 ± 0.1	13.7 ± 0.1	87.6 ± 0.1	5.5 ± 0.2	85.1 ± 0.5	11.3 ± 0.2
32	86.6 ± 0.1	13.4 ± 0.1	87.2 ± 0.4	5.4 ± 0.2	85.2 ± 0.4	9.0 ± 0.3

Table 2: Models A, B, and C from Table 1, but trained instead with random labels sampled uniformly. Clearly, model B has the least information capacity, yet also happens to achieve the highest test accuracy for the original labels in Table 1. Notice the 25±1% LBGG for C, and the 54±2% GG between C and B for batch size 128. There is a $3 \pm 4\%$ LBGG in A.

	A		B		C	
Batch Size	Train (%)	Gap (%)	Train (%)	Gap (%)	Train (%)	Gap (%)
128	85 ± 3	76 ± 3	10.20 ± 0.03	3 ± 2	67 ± 1	57 ± 1
32	90 ± 2	79 ± 2	–	–	42 ± 1	32 ± 1

then a zero-phase whitening (ZCA) to linearly decorrelate neighboring pixels (Bell & Sejnowski, 1997). This increases the SNR by emphasizing the information in the edges.

We characterize two different explicit regularization strategies applied to a simple four-layer CNN: batch normalization (Ioffe & Szegedy, 2015) without momentum, applied to the two middle hidden layers, and L^2 weight decay. The training regime was inspired by the IB method applied to deep learning (Schwartz-Ziv & Tishby, 2017). We set up the information dynamics, i.e., the parameterization of SGD's stationary distribution, and let the system run to steady state. All hyperparameters were chosen intentionally; no automated search was invoked.

We trained all models for 500 epochs, as this was at least one order of magnitude longer than the "fitting" phase required to minimize the training loss. This would allow sufficient time to observe compression. We chose a relatively large and constant learning rate of 1e-2 to maximize the power of the noise during the compression phase, and by extension the energy used to maximize the entropy of the weights under the ERM constraint in the finite number of epochs. Lastly, we used the smallest weight decay regularization constant λ, i.e., the average power constraint, that prevented the model from fitting random labels significantly better than chance ($0.20 \pm 0.03\%$ absolute percent), as shown for Model B in Table 2. This λ turned out to be approximately 1e-2. Interestingly, the model with this setting of weight decay *happened* to also have the highest test accuracy.

We vary the batch size, optionally adding the explicit regularizers, and report the resulting generalization gap in Table 1.[2] Model C includes weight decay and batch normalization because the baseline model A already perfectly fits the training set, adding batch norm alone yields a similarly large generalization gap as the baseline. Results for the same experiments repeated with random labels are provided in Table 2, where we demonstrate that Model B (baseline + weight decay) has a generalization gap of $3 \pm 2\%$. This particular generalization gap is not to be trusted, since the SVHN test set is unbalanced. In fact, one particular seed achieved a generalization gap of 9.4%, or a test accuracy of 19.6%, which is exactly the accuracy obtained by always predicting class 1. The information is 0 bits in this case, but we report accuracy here in keeping with standard practice.

Using the strategy that minimized generalization error (Model B), we retrain with additional training data to boost the test accuracy (Model B+), gaining $2 \pm 0.3\%$ for a $0.3 \pm 0.1\%$ smaller generalization gap.

[2] The sigificant figures are consistent with the precision of the standard error in each experiment.

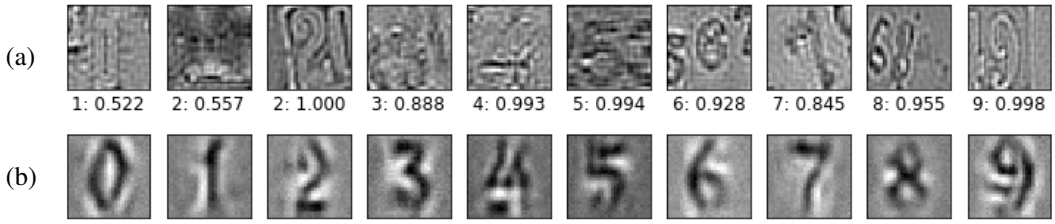


Figure 4: Demonstration of regularized generative modeling with an adversarial attack. (a) Samples from our preprocessed version of the SVHN dataset for each class $Y \in \{0, \ldots, 9\}$ arranged from left to right, with the predicted class (argmax softmax probability). (b) Transmitting a message across the channel in the reverse direction (i.e., from Y to X) by minimizing the loss for each $y \in Y$ w.r.t. $X \sim \mathcal{N}(\mu, \sigma^2)$ with step size 1e-2. For visualization purposes, we set μ equal to the population mean, and σ^2 as two orders of magnitude less than the population standard deviation. We show the pattern X obtained after iterating until full confidence to within 3 decimal places, which took roughly 100 steps. Unlike in Nguyen et al. (2015), we recover prototypical examples because we have reduced the excess rate of the channel. Note that this would not be a suitable model of the original data-generating distribution as it is regularized by Y in a way that the irrelevant digits in (a) are "forgotten".

4.3.1 Zero and one-shot transfer

We were curious to see if the best SVHN model would generalize to handwritten digits. To this end, we bilinearly upsample MNIST to a 32x32 grid, and apply the same preprocessing as was used for SVHN. The model obtained 63% accuracy zero-shot. Selecting one example from each class of the MNIST validation set, and fine-tuning with SGD for ten steps while rotatating the ten instances randomly $\leq 10^\circ$ at each step, boosted overall test accuracy to 70%. This is similar to figures reported elsewhere for one-shot learning, e.g., Vinyals et al. (2016), although they transfer from the Omniglot dataset, which has a flat background similar to MNIST.

4.4 Fault Tolerance

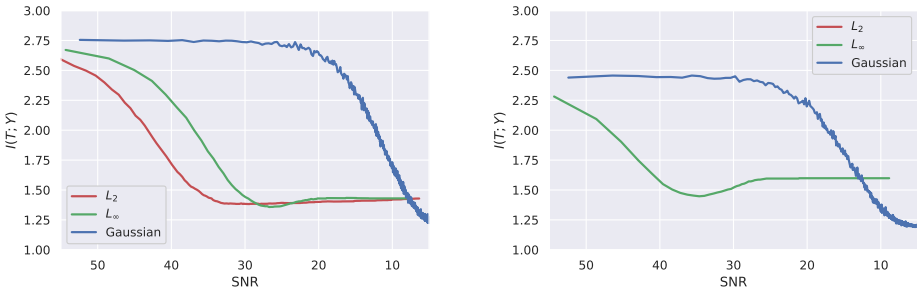
Figure 4 shows inputs and outputs to model B+. The human legible outputs shown in Figure 4(b) were obtained via a similar gradient-based procedure used by Nguyen et al. (2015) to craft "fooling images" initialized from noise. There was little qualitative difference to these samples compared to those obtained from Model B trained on only the smaller training set consisting of images with distracting digits, which we show in Figure 4(a) along with their prediction confidences.

In Figure 5 we characterize fault tolerance in terms of a gradual degradation in informatation conveyed about the label, while the SNR decreases. We compute SNR in dB as in Cisse et al. (2017) for original input x and noise δ:

$$20 \log \left(1 + \frac{\|x\|}{\|\delta x\|} \right)$$

We consider three variants of "noise", the first two are obtained by interatively minimizing the loss w.r.t. the input with a step size of 1e-2, then either taking the real gradient or the signed gradient denoted as L_2, and L_∞ respectively in Figure 5. As baseline we compare with white Gaussian noise.

The model trained with batch normalization conveys less information for all sources of noise at the same SNR than without. Note that the information plateau in Figure 5(a) after SNR of 30dB for the adversarial noise is not due to the gradient masking effect – the accuracy indeed goes to 0% for this unbounded attack. At this point, there is so much salient structure in the noise that we are interpolating between legitimate images, thus information slowly rises due to label leaking into pixel space, that is, the model reliably predicts `not` Y, which logically conveys some information about Y. The plateau for the less robust model in Figure 5(b) can be explained given that L_∞ perturbations induce significantly more noise in pixel space for the same effect, therefore we are less able to take the most directed Euclidean path to the nearest misclassified manifold corresponding to another class.



(a) (b)

Figure 5: Characterizing fault tolerance in terms of $I(T;Y)$ and SNR for two kinds of adversarial noise, and benign Gaussian noise for context. Model (a) is with weight decay, and (b) includes BN, which makes the model more sensitive to all sources of noise. The L_2 curve in (b) is missing because the SNR was infinite due to lack of structure in the noise.

5 Conclusion

The established framework for characterizing information transmission in the presence of noise is Shannon's rate-distortion theory. The insight that essentially the same trade-off is central to deep learning problems allows us to establish a notion of capacity for deep neural networks that explains their generalization behavior to a sigificant extent. Guided by the IB principle, which augments rate-distortion theory with the relevant variable, and minimizing the difference between the empirical and expected loss – as suggested by statistical learning theory – we derive guidelines for efficient use of this capacity and obtain a recipe that yields compelling fault tolerance for "worst-case" inputs, such as adversarial examples. We confirm that it is indeed possible to generalize in a narrow sense to the "clean" test set when the model has excess capacity, but that minimizing this excess capacity is essential for fault-tolerant generalization behavior.

The implications for practical deep learning are simple: In order to obtain a model with good generalization behavior and fault tolerance, i) irrelevant information in the dataset, such as, e.g., colour and texture in the task of digit recognition, can be safely removed; ii) training with constant learning rate SGD should be prolonged well beyond the number of epochs required to minimize the empirical loss; iii) sufficient constraints, e.g., weight decay, have to be applied. Particular care has to be taken when applying batch normalization, since it impedes successive refinement of information in the learning procedure and might reduce model robustness. As a general rule, we suggest that preventing the model from fitting random labels better than chance as a good first step for calibrating such constraints.

REFERENCES

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep Variational Information Bottleneck. In *International Conference on Learning Representations*, 2017.

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A Closer Look at Memorization in Deep Networks. In *International Conference on Machine Learning*, pp. 233–242, 2017.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *International Conference on Machine Learning*, pp. 274–283, 2018.

P. L. Bartlett. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. In *Computer Vision and Pattern Recognition*, 2017.

A. J. Bell and T. J. Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim ˇSrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion Attacks against Machine Learning at Test Time. In *ECML-KDD*. Springer, 2013.

Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets. *arXiv:1802.08232*, 2018.

Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual Causal Feature Learning. In *Uncertainty in Artificial Intelligence*, 2015.

Gal Chechik, Amir Globerson, Naftali Tishby, and Yair Weiss. Information Bottleneck for Gaussian Variables. *Journal of Machine Learning Research*, 6:165–188, 2005.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval Networks: Improving Robustness to Adversarial Examples. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 854–863, International Convention Centre, Sydney, Australia, 2017. PMLR.

Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, 1991.

G. A. Darbellay and I. Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 45(4):1315–1321, 1999.

Simant Dube. High Dimensional Spaces, Deep Learning and Adversarial Examples. *arXiv:1801.00634*, 2018.

W. H. R. Equitz and T. M. Cover. Successive refinement of information. *IEEE Transactions on Information Theory*, 37(2):269–275, 1991.

Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Angus Galloway, Thomas Tanay, and Graham W. Taylor. Adversarial Training Versus Weight Decay. *arXiv:1804.03308*, 2018.

Justin Gilmer, Luke Metz, Fartash Faghri, Sam Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial Spheres. 2018.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-Based Capacity Control in Neural Networks. In *Conference on Learning Theory*, pp. 1376–1401, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro. Exploring Generalization in Deep Learning. In *Advances in Neural Information Processing Systems*, pp. 5947–5956. 2017.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Computer Vision and Pattern Recognition*, pp. 427–436. IEEE Computer Society, 2015.

Liam Paninski. Estimation of Entropy and Mutual Information. In *Neural Computation*, volume 15, pp. 1191–1253. MIT Press Journals, 2003.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.

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.

Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on MNIST. *arXiv:1805.09190*, 2018.

Ravid Schwartz-Ziv and Naftali Tishby. Opening the Black Box of Deep Neural Networks via Information. *arXiv:1703.00810*, 2017.

Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, and Nathan Srebro. The Implicit Bias of Gradient Descent on Separable Data. In *International Conference on Learning Representations*, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Thomas Tanay and Lewis D. Griffin. A Boundary Tilting Persepective on the Phenomenon of Adversarial Examples. *arXiv:1608.07690*, 2016.

Thomas Tanay, Jerone T. A. Andrews, and Lewis D. Griffin. Built-in Vulnerabilities to Imperceptible Adversarial Perturbations. *arXiv:1806.07409*, 2018.

Yichuan Tang and Ruslan R Salakhutdinov. Learning Stochastic Feedforward Neural Networks. In *Advances in Neural Information Processing Systems 26*, pp. 530–538. Curran Associates, Inc., 2013.

Naftali Tishby and Noga Zaslavsky. Deep Learning and the Information Bottleneck Principle. *arXiv:1503.02406*, 2015.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Allerton Conference on Communication, Control and Computing*, 1999.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. There Is No Free Lunch In Adversarial Robustness (But There Are Unexpected Benefits). *arXiv:1805.12152*, 2018.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29*, pp. 3630–3638. Curran Associates, Inc., 2016.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.