TOWARD DISENTANGLING DOUBLE DESCENT AND INFORMATION FLOW IN DEEP NEURAL NETWORKS

Chris Liu[*]
yliu298@ucsc.edu

Brendan King[*]
bking2@ucsc.edu

Jing Gu[*]
jgu110@ucsc.edu

ABSTRACT

Despite the impressive practical successes observed in applying increasingly large neural networks to challenging problems, theoretical work has not caught up in explaining their behavior. Our work aims to explore a connection between two distinct phenomena observed in neural networks. The first is double descent, in which increasingly complex neural networks eventually exhibit decreased expected risk, despite what conventional understanding of over-fitting would imply. The second is the information bottleneck principle, in which examining neural networks as a Markov chain suggests that the optimal neural representation T must trade-off mutual information with its input $I(X;T)$ and with its output $I(T;Y)$. Following empirical studies which observe dynamics of this trade-off beginning with Shwartz-Ziv and Tishby (2017), we reproduce this phenomena and the double-descent in order to present a combined understanding of the two. We find that while over-parameterized networks can consistently generalize across choices of architecture and loss, only some of these settings include the compression (reduction of $I(X;T)$) expected under the information bottleneck hypothesis. Finally, we observe similar phase transitions in the levels of gradient noise discussed inShwartz-Ziv and Tishby (2017), and find intriguing relationships between such training phases and model size.

1 Introduction

Overparameterized neural networks have become the go-to choice for tackling difficult real-world problems in both research and industry due to its unprecedented ability to learn rich representations and generalize well. However, our understanding of large neural networks also has become stagnant, and more and more mysterious phenomena (Belkin et al., 2018; Jacot et al., 2018; Frankle and Carbin, 2018; Papyan et al., 2020; Sorscher et al., 2022; Soudry et al., 2018; Nacson et al., 2019) start to emerge and still remain unsolved or uninterpretable. While many work try to demystify the generalization power of neural networks (especially large ones) from the perspective of machine learning and deep learning theory, answers that utilize tools from other domains are lacking.

We propose to study one of the most intriguing phenomena in deep neural networks, double descent, a counterintuitive phenomenon where the performance of a neural network is first deteriorated and then improves as the number of parameters increases. This phenomenon is interesting as it captures both the traditional understanding of machine learning models, the bias-variance trade-off (Geman et al., 1992) , and also modern model-error scaling laws (Kaplan et al., 2020). Specifically, we employ the information plane analysis (Shwartz-Ziv and Tishby, 2017; Geiger, 2021) to analyze the mutual information of the internal representation T_i from each layer i with both input X and the label Y in a deep neural network when the double descent phenomenon is presented. Therefore, we ask the following three research questions:

1. Is the amount of compression an indicator of good generalization? If not, what is?
2. Do neural networks in different sizes behave the same during training in terms of compression?
3. What, in terms of mutual information compression (phase transition), distinguishes the two stages on a double descent curve?

[1]The authors contributed equally to this work.

Our findings suggests that the amount of compression in $I(X;T)$ might not be a good proxy for generalization. This is because the compression cannot reflect the non-linear function of generalization error versus model size and exhibits different compression behaviors in some settings. Our results show that the point when phase transition occurs and the noisiness of the gradient distinguishes underparameterized, $|\theta| \approx N$, and overparameterized networks: 1) in underparamterized networks, all layer compress; 2) If $|\theta| \approx N$, only the last layer compresses; 3) if a network is overparameterized, the last layer expands, and all other layers become stationary. This is also related to the previous controversy of whether compression is causally linked to generalization.

The paper is structured as follows. In Section 2, we provide necessary background of double descent (Belkin et al., 2018; Nakkiran et al., 2019), the information bottleneck principle, (Tishby and Zaslavsky, 2015; Shwartz-Ziv and Tishby, 2017), and the estimation of mutual information. In Section 3, we describe the method we select to estimate mutual information in our setup and the mathematical formulation. In Section 4 and Section 5, we discuss our experimental setup and analyze the results. We conclude in Section 6.

2 Background

2.1 The double descent phenomenon

According to Loog et al. (2020), the double descent phenomenon was already observed in the history of machine learning (Vallet et al., 1989; Penrose, 1956; Opper et al., 1990; Watkin et al., 1993; Duin, 2000; Skurichina and Duin, 1998; Raudys and Duin, 1998). However, Belkin et al. (2018) is the first to systematically study the phenomenon and demonstrate it in a wide range of settings, albeit simple[1]. Such a phenomenon contradicted the traditional wisdom in bias-variance trade-off (Geman et al., 1992): When the number of parameters of a network is smaller than a certain threshold, the expected error follows the traditional U-shaped curve; if the size of a model continues growing, the expected error experiences a second descent. The threshold is referred to as the interpolation threshold, which is usually located at a point where the model parameter size equals to sample size, but it can vary slightly (Nakkiran et al., 2019), depending on the specific setup. More empirical results on this phenomenon and factors that affect the risk curve are shown in Nakkiran et al. (2019) for a variety of model architectures and tasks, showing that the phenomenon indeed applies to real-world problems and popular architecture choices. Nakkiran et al. (2020) further suggests optimal regularization techniques can mitigate the double descent peak. A recent work Lee and Cherkassky (2022) argues that the VC theory is still capable of explaining the double descent phenomenon. They show the norm of the output layer of a random feature network can be used to approximate its VC-dimension, and the norm decreases as the network enters the large width regime. A similar phenomenon was also identified in Yang et al. (2020), which demonstrates empirically estimated bias and variance decrease monotonically in the large width regime, matching the second descent in generalization error. The double descent phenomenon also occurs in the infinite neural networks (e.g., neural tangent kernel) (Jacot et al., 2018), which shows that as a network proceeds to the infinite width limit, its first-order Taylor approximation around the initialization point becomes more accurate, making the network linear in its weights and thus simpler. Liu et al. (2020) provides more insights into when such linearization occurs and points out limitations; Adlam and Pennington (2020) demonstrates an extension of the double descent, the triple descent, where the third descent happens when the number of parameters scales quadratically with the training sample size, as a network enters the infinite width regime. The universality of triple descent needs further investigation, so we do not consider it in this work.

We now give a formal definition of double descent phenomenon. Let $f(\cdot;\theta)$ denote a function (we sometimes also write f_θ) learned based on i.i.d. training samples $\mathcal{D} = \{(x_k, y_k)\}_{k=1}^N \sim p(X,Y)$ with a fixed training procedure $\mathcal{P}$. Given an error function ϵ and a loss function $\mathcal{L}$, the empirical error and the loss L of the function $f(\cdot;\theta)$ with respect to the true labels on $\mathcal{D}$ can be expressed as

$$\epsilon(f_\theta, \mathcal{D}) = \frac{1}{N}\sum_i^N \mathbb{1}\left[f(x_i;\theta) \neq y_i\right], \quad L(f_\theta, \mathcal{D}) = \frac{1}{N}\sum_i^N \mathcal{L}\left(f(x_i;\theta), y_i\right).$$

The true error and loss are in the expectation form $\mathbb{E}_{(x,y)\sim p(X,Y)}\mathbb{1}\left[f(x;\theta) \neq y\right]$ and $\mathbb{E}_{(x,y)\sim p(X,Y)}L\left(f(x;\theta), y\right)$, respectively. Suppose we have a set of learned functions with the function parameters $\theta^M = \{\theta_0, \theta_1, ..., \theta_M\}$ obtained from the same training procedure $\mathcal{P}$ as above. Let i be an index associated with each θ such that for any two indices $i, j \in \mathbb{Z}^+$ and $0 \leq i < j \leq M$, the elements in the set have the property $|\theta_i| < |\theta_j|$. In other words, the number of parameters of model i has to be strictly smaller than that of model j, if $i < j$ (i.e., the set is ordered in both index and parameter count), and the converse must also hold true. The sizes are also required to be unique in the set

[1]To the best of our knowledge, the term "double descent" was not coined until Nakkiran et al. (2019).

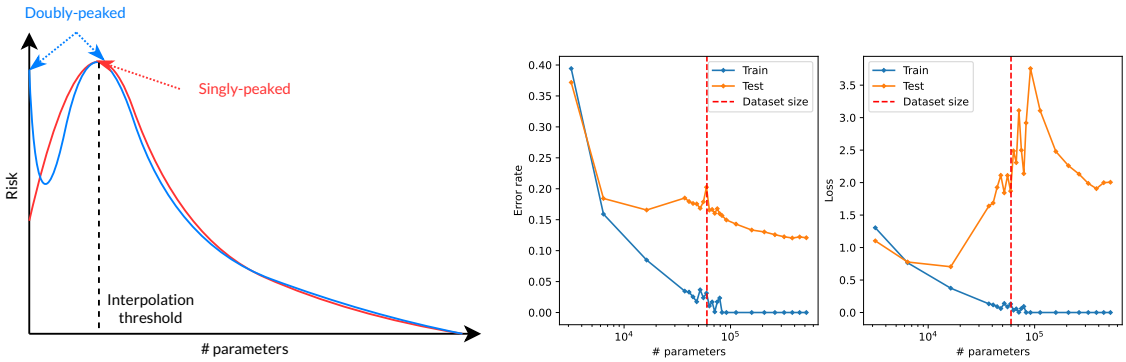


Figure 1: Left: The singly- and doubly-peaked double descent curves were later identified in Yang et al. (2020) due to different dominating stages of bias and variance. Right: When the number of parameters of a network is smaller than a certain threshold, the expected risk follows the traditional U-shaped curve; if the size of a model continues growing, the expected risk experiences a second descent. The plot is generated by a fully-connected network with ELU on Fashion-MNIST.

θ^M, and therefore no equality between $|\theta_i| < |\theta_j|$ can be held. For notation convenience, we also define a function $\text{Index} : \mathbb{Z} \to \mathbb{Z}$ that takes in the size of a particular $\theta \in \theta^M$ and returns its index i.

Definition 2.1 (Interpolation threshold (IT)). The interpolation threshold is located at $|\theta| = N$. Typically, the threshold allows certain degree of shift (α, β), resulting in $|\theta| - \alpha \leq \text{IT} \leq |\theta| + \beta$. The range $[|\theta| - \alpha, |\theta| + \beta]$ is also known as the cirtical rigime (Nakkiran et al., 2019).

Definition 2.2 (Optimal index). Optimal index is the index of the parameter of a function $f(\cdot; \theta)$ such that the objective $\min_{0 \leq i \leq \text{Index(IT)}} \epsilon(f_{\theta_i}, \mathcal{D})$ is minimized.

Definition 2.3 (Double descent). Under the double descent phenomenon, the expected error rate does not always decrease monotonically as the parameter size increases. Instead, the error rate follows the traditional U-shape risk curve at first until reaching the interpolation threshold and only decreases monotonically afterward. We say a double descent curve is *single-peaked* (variance dominates) if the optimal index is exactly 0, and *double-peaked* (bias and variance dominate asynchronously) if the optimal index is $0 < i < \text{Index(IT)}$.

Note that Index(IT) is the index of a particular θ with $|\theta| = \text{IT}$ that achieves the highest expected error in a single-peaked curve. This clain cannot be made directly for a double-peaked descent curve, because the relationship between the error of θ_0 and $\theta_{\text{Index(IT)}}$ still needs to be determined. The single- and double-peaked double descent was observed in early work (Nakkiran et al., 2019) but was not officially identified until Yang et al. (2020). The authors found qualitative difference in the double descent curve when the bias and variance of the model dominate in different regimes. Specifically, a double-peaked descent is due to bias dominating when the model complexity is small and variance dominating after when the model complexity reaches the interpolation threshold. For a single-peaked descent curve, variance almost dominates regardless of the model complexity. Double descent also occurs on the axis of epochs (Nakkiran et al., 2019; Stephenson and Lee, 2021; Heckel and Yilmaz, 2020) and the training data size (Belkin et al., 2018; Nakkiran et al., 2019). In this work, we only consider double descent in model size versus the generalization error for both single- and double-pearked curves.

2.2 Information bottleneck principle

The information bottleneck (IB) principle (Tishby and Zaslavsky, 2015; Shwartz-Ziv and Tishby, 2017) suggests that the best representation learned by the neural network is the one that minimizes the mutual information between the learned representation (i.e., the internal representation) and the input data while at the same time maximizing the mutual information between the internal representation and the output. The principle treats the outputs T_i from a cascade of network layers as variables in a Markov chain consisting of the representations T, the inputs X, and outputs Y as $Y \leftrightarrow X \leftrightarrow T$. In other words, the network extracts information about a target label Y through an observation X, and the extraction is represented by $T = f(X)$. This can also be viewed from a topological perspective of an actual neural network: a multi-layer neural network is sliced into two parts, the encoder and the decoder. The encoder consists of a stack of parallel neurons and accepts the raw input X and produces T. The decoder (with a similar structure) takes in T and produces a prediction $\hat{Y}$. Both parts are essentially linear combined with non-linear transformations. Note that the slicing point is not necessarily located at the middle of the network. Instead, for a set of hidden layers with outputs

$\{T_1, T_2, ..., T_k\}$, we can take any T_i as the slicing point. In fact, a mutual information figure in Shwartz-Ziv and Tishby (2017) typically shows mutiple curves, each corresponding to a single slicing point.

Given the above topological view of Y, X, T, The IB principle imposes an upper bound constraint on the the mutual information between the input and the representation, $I(X;T) \leq I_c$, and maximizes the mutual information between the representation and the output, $I(T;Y)$. The optimization of the network can then be formulated as a constraint optimization problem:

$$\max_{\theta} I(T;Y)$$
$$\text{s.t. } I(X;T) \leq I_c.$$

It is also worth noting that, because of the data processing inequality (Thomas and Joy, 2006) and the Markovian nature of the chain $Y \leftrightarrow X \leftrightarrow T$, for multiple layers,

$$I(Y;X) \geq I\left(Y;T_j\right) \geq I\left(Y;T_i\right) \geq I(Y;\hat{Y}),$$

where $i \leq j$. The equality of the above only holds when T_i is a sufficient statistic of X or its previous representation T_{i-1}. In other words, to ensure a strong information flow, $I(Y;T_i)$ needs to be maximized and $I(T_{i-1};Ti)$ needs to be minimized. An intermediate result from the IB principle is the information plane theorem. Informally, the theorem states that the sample complexity of a deep neural network solely depends on the minimization of $I(X;T)$, whereas the generalization error is governed by the maximization of $I(T;Y)$. We will not discuss it detail here as it is not directly related to our work, because we only focus on the relationship between the generalization error and the model complexity. However, one extremely valueable result from the information bottleneck principle is the relationship between $I(X;T)$ and $I(T;Y)$ over the course of training. Specifically, $I(X;T)$ shows a two-phase transition: in the first phase, $I(X;T)$ in all layers increases, followed by a second phase where $I(X;T)$ decreases, which accounts for most of the epochs during the optimization lifecycle.

It is worthing noting that a follow-up work (Saxe et al., 2018) argues against Shwartz-Ziv and Tishby (2017) by demonstrating that the phenomenon by IB principle largely depends on the type of non-linearities used. For example, single-sided satuarating functions (e.g., ReLU) do not exhibit the compression bahavior. They also emphasize the lack of causal evidence in compression and generalization because ReLU networks still generalize well. A recent work (Lorenzen et al., 2021) addressed the controversy and took a perspective from quantization. The authors argue that the infinite precision in a continuous network introduces problem in mutual information estimation, becase the binning opeartion already discards information. By employ fully discretized networks with no estimation difficulties, they confirmed that the compression phase is not an artifact of binning and is universal to all layers. However, following Saxe et al. (2018), the authors did point out tha absence of the compression phase in single-sided activation functions. For reference of other related work, a comprehensive analysis of the IB methods is in Geiger (2021).

2.3 Mutual information estimation

A fundamental challenge in studying the information bottleneck principle empirically is estimating the mutual information between layers and inputs or outputs. Specifically, we are tasked with estimating a non-linear function of multiple continuous probability densities using only a finite number of samples:

$$I(X;T_i) = \int f(x,t_i)\, log\left(\frac{f(x,t_i)}{f(x)f(t_i)}\right) dxdt_i \tag{1}$$

The continuous and high dimensional nature of the space of all activations T_i makes this particularly challenging Paninski (2003). Further, a continuous random variable T_i may have infinite mutual information with another, particularly in this case as it is a function of X. As such, a meaningful value for $\hat{I}(X;T_i)$ depends strongly on the estimator, for which there is no general consensus approach Geiger (2021). It is typical to use various approximation methods that are (often discrete) functions of these activations, such that we get a lower bound for $I(\cdot;T_i)$ by the data processing inequality.

Initial efforts try to approximate mutual information in deep neural networks by discretizing layerwise outputs (Shwartz-Ziv and Tishby, 2017; Saxe et al., 2018). Specifically, the output of each layer in a neural network is binned, and the mutual information $I(X;\mathrm{Bin}(T))$ is used as a proxy of the true $I(X;T)$. However, the drawback of the binning approach is that the estimation only approaches the true value when the bin size is small enough or even close to zero (Thomas and Joy, 2006), which in turn makes it computationally infeasible. Goldfeld et al. (2019) further suggested that the fluctuation of $I(X;\mathrm{Bin}(T))$ in the compression stage presented in Shwartz-Ziv and Tishby (2017) and Saxe et al. (2018) is due to error in estimation instead of the real change in information, because the mutual information $I(X;T)$ in deterministic networks with strictly monotone non-linearities is constant or infinite (Amjad and Geiger, 2019). The

error in measuring mutual information in continuous networks also led to the controversy between Shwartz-Ziv and Tishby (2017) and Saxe et al. (2018), as discussed above in Section 2.2. Other more advanced estimation methods have also been proposed to address the issue mentioned above (Belghazi et al., 2018; Cheng et al., 2018; Hjelm et al., 2019; Poole et al., 2019; Molavipour et al., 2021; Mukherjee et al., 2019; Wen et al., 2020; Guo et al., 2021; Gabrié et al., 2018; Song and Ermon, 2020; Goldfeld and Greenewald, 2021; Brekelmans et al., 2021).

3 Method

3.1 Estimating mutual information

A variety of mutual information estimation methods have been proposed, as discussed in Section 2.3. After considering a number of these, we decided to follow the estimation method used in Shwartz-Ziv and Tishby (2017). Below we describe it in detail as well as its strengths and limitations.

The binning method used in Shwartz-Ziv and Tishby (2017) works by computing discrete estimates of probability mass functions corresponding to the densities in Eq. (1) using a binning function $\mathrm{Bin}(\cdot)$, described in detail below. The overall estimate is computed according to the following equation:

$$\hat{I}(X;T_i) = \sum_{\mathcal{X}\times\mathrm{Bin}(\mathcal{T}_i)} \hat{P}(x,\mathrm{Bin}(t_i))\; log\left(\frac{\hat{P}(x,\mathrm{Bin}(t_i))}{\hat{P}(x)\times\hat{P}(\mathrm{Bin}(t_i))}\right) \tag{2}$$

Let T_i be a random variable for the activated output of the ith layer in a neural network. As such, T_i is vector of continuous component random variables in $\mathbb{R}^d$, where d is the hidden dimension or width of the network layer. For notation, we'll use T_i^j to indicate the jth component of T_i, and $\mathcal{T}$ to indicate the space or range of real numbers such that $T_i^j \in \mathcal{T}$. The range $\mathcal{T}$ depends on our choice of activation, but we'll describe the procedure using the $tanh$ activation, such that $\forall T_i^j \in \mathcal{T} : -1 \leq T_i^j \leq 1$.

Let t_i be an instance of T_i. Given the range defined in $\mathcal{T}$ and a pre-defined number of bins K, the binning function transforms each component into t_i^j into a number $1...K$ corresponding to the interval it lies in when $\mathcal{T}$ is evenly divided into K intervals. If $T_i' = Bin_K(T_i)$, then where T_i is a vector of continuous random variables in $\mathcal{T}$, then T_i' is a vector of discrete random variables in $\mathcal{T}'$, where $|\mathcal{T}_i'| = K$.

Given this function, the method for estimating $P(\hat{\mathrm{Bin}}(t_i))$ (the joint over this vector of categorical r.v.s) is straightforward:

$$\hat{P}(\mathrm{Bin}(t_i)) = \hat{P}(T_i' = t_i') = \frac{count(\mathrm{Bin}(t_i) = t_i')}{N} \tag{3}$$

For an individual activated component j in the output of the ith layer, the binning function is essentially the transformation needed to compute a histogram over all t_i^j given by our N data points. Algorithm 1 describes the algorithm in detail.

The key strengths of this method are its simplicity and efficiency to compute, which are important in our setting, since we must run many sizes of neural models for thousands of epochs each. In order to facilitate comparison to experiments in Shwartz-Ziv and Tishby (2017), it is also important that we use the same estimator (and thus same choice of $K = 30$) Geiger (2021).

Our estimator choice is not perfect though. First, it can be challenging to adapt to different choices of activation functions. For doubly-saturated activations like *tanh* (bounded on both sides by a finite value), the binning procedure is straight forward. For singly-saturated activations such as *RELU*, which are only bounded on one side, it can be hard to determine how to lay out the bins. Following prior work, for singly saturated activations we set the upper bound of the range of $\mathcal{T}$ to be the current highest activation in t_i given all N samples Saxe et al. (2018). This can fluctuate over the course of training, and Saxe et al. (2018) specifically were not able to reproduce the compression phase for singly-saturated activations. In our reproduction however, we were able to reproduce the compression phase of the information bottleneck phenomena in networks with singly-saturated activations.

Algorithm 1 Estimating $I(X;T_i)$. Our implementation makes use of tensor operations vs. loops described below.

procedure BIN($z, bins$) ▷ Given $z \in \mathbb{R}$, return the bin in $[1...K]$ it would be assigned to
  $bins$ ▷ vector of $K-1$ boundaries, breaking down $\mathbb{R}$ into K intervals
  for $i : 1 \to len(bins)$ **do**
    if $z \leq bins[i]$ **then**
      return i
    end if
  end for
end procedure
procedure ESTIMATEMI($x, t, y, \mathcal{T}, K$) ▷ Given N instances of x, t_i, y estimate $I(X,T_i)$ and $I(Y;T_i)$
  $x_{id} = distinct(x)$ ▷ Assign each distinct x an id $1...N$
  $bins = linspace(\mathcal{T}, K)$ ▷ vector of boundaries which break up $\mathcal{T}$ into K intervals
  $p_x \leftarrow dict(* \to 0)$ ▷ empty map w/ default value 0
  $p_{x,t} \leftarrow dict(* \to 0)$
  $p_{y,t} \leftarrow dict(* \to 0)$
  $p_y \leftarrow dict(* \to 0)$
  $p_t \leftarrow dict(* \to 0)$
  for $i : 1 \to N$ **do** ▷ iterate data
    $t' \leftarrow vector(d)$ ▷ create an empty d length vector
    for $j : 1 \to d$ **do** ▷ iterate dimensions
      $t'[j] \leftarrow$ BIN$(t_i^j, bins)$
    end for
    $p_t[t'] \leftarrow p_t[t'] + \frac{1}{N}$ ▷ Update our mass estimates
    $p_x[x] \leftarrow \frac{1}{N}$
    $p_{x,t}[(x,t')] \leftarrow \frac{1}{N}$
    $p_y[y] \leftarrow p_y[y] + \frac{1}{N}$
    $p_{y,t}[(y,t')] \leftarrow p_{y,t}[(y,t')] + \frac{1}{N}$
  end for
  $I_{X;T}, I_{Y;T} \leftarrow 0, 0$
  for $(x,t') \in p_{x,t}$ **do** ▷ Iterate over unique values in joint estimate
    $I_{X;T} \leftarrow I_{X;T} + p_{x,t'}[(x,t')] * log_2(\frac{p_{x,t}[(x,t')]}{(p_x[x] * p_{t'}[t'])})$
  end for
  for $(y,t') \in p_{y,t}$ **do**
    $I_{Y;T} \leftarrow I_{Y;T} + p_{y,t'}[(y,t')] * log_2(\frac{p_{y,t}[(y,t')]}{(p_y[y] * p_{t'}[t'])})$
  end for
  return $I_{X;T}, I_{Y;T}$
end procedure

3.2 Producing double descent

Producing double descent is a relatively easy task compared to the estimation of mutual information. We apply the same network architecture, training procedure, and datasets described in Section 3.1. Specifically, because the dataset size $|\mathcal{D}|$ needs to satisfy $\min_i\{|\theta_i|\} \leq |\mathcal{D}| \leq \max_i\{|\theta_i|\}$, the range of the network size needs to be carefully selected. To facilitate the visualization of information, a network with multiple layers is preferred to shallow two-layer networks widely used in Belkin et al. (2018); Nakkiran et al. (2020), leaving only a single T to be studied. Generally, the procedure can be described as follows:

1. Given a fixed dataset $\mathcal{D}$, a fixed training procedure $\mathcal{P}$, a model architecture function f, generate a set of parameter sizes $\mathcal{S} = \{s_1, s_2, ..., s_M\}$ such that $|\theta_i| = s_i$ for all i and IT $= s_i$ for some i.
2. For each θ_i, train $f(\cdot;\theta_i)$ with $\mathcal{P}$ on $\mathcal{D}$.
3. Record the training and testing error rate $\epsilon(f_{\theta_i}, \mathcal{D})$.
4. Repeat the above steps for N times and calculate the empirical averaged error rate $\epsilon(f_{\theta_i}, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\epsilon(f_{\theta_i}, \mathcal{D})$.

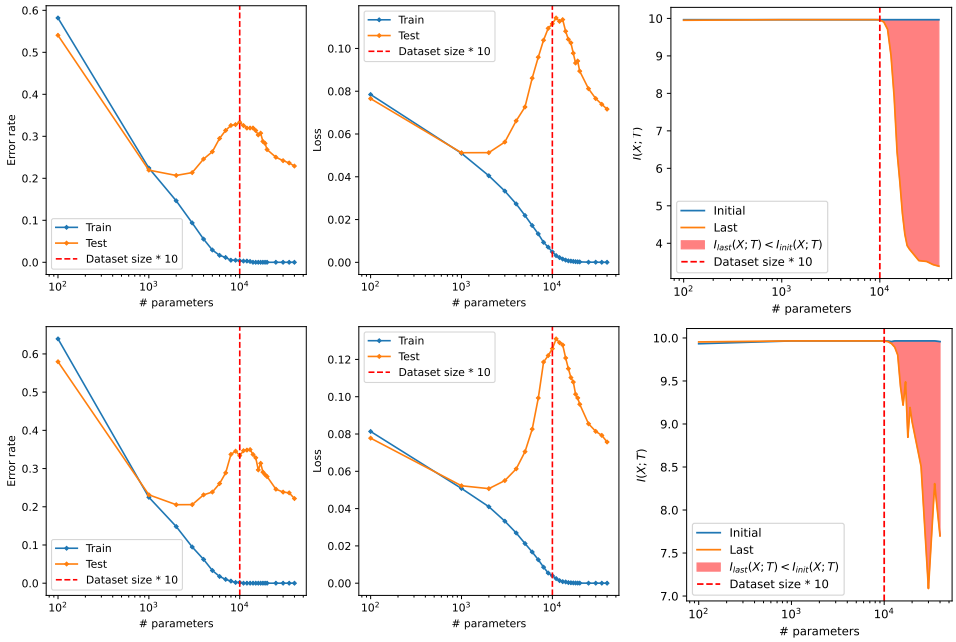


Figure 2: Only overparameterized linear networks compress, and the amount of compression is positively related to the number of parameters in the network. The setup uses a random feature network with Tanh (row 1) and ReLU (row 2) features trained on MNIST using MSE loss.

4 Experimental Setup

For comparison purpose and the simplicity in analysis, we choose both linear random feature networks and standard multi-layer fully-connected networks as our main focus. Following Belkin et al. (2018), the random feature network is essentially a two-layer fully-connected network with a frozen first layer. For an input $\mathbf{X}$ and random matrix (weights) $\mathbf{W} \sim \mathcal{N}(0, \sigma^2)$ ($\sigma = \frac{1}{\sqrt{\text{Dim}(X)}}$), and a non-linear function g, the features $\mathbf{Z}$ is calculated by

$$\mathbf{Z} = g(\langle \mathbf{W}, \mathbf{X} \rangle), \quad i = 1, \ldots, M.$$

We consider various g's, including both singly-saturating activation functions (i.e., ReLU and ELU) and doubly-saturating activation functions (i.e., Tanh and soft sign) because Saxe et al. (2018)'s finding shows inconsistent behavior in compression for ReLU and Tanh. We also consider five-layer fully-connected networks with base width W and network configuration $(\text{Dim}(X), 8W, 4W, 2W, W, C)$, where C is the number of classes. All weights are sample uniformly from $\mathcal{U}\left(-\frac{1}{\sqrt{\text{Dim}(X)}}, \frac{1}{\sqrt{\text{Dim}(X)}}\right)$. For datasets, we use MNIST (Cireşan et al., 2011)and Fashion-MNIST (Xiao et al., 2017). In all settings, the networks are trained using standard SGD (with momentum for linear models).

During mutual information estimation, we use 30 bins following Shwartz-Ziv and Tishby (2017); Saxe et al. (2018) and determine the range of the bins based on a particular non-linear activation function. Specifically, we use $[-1, 1]$ for Tanh and soft sign, $[0, \max)$ for ReLU, and $[-1, \max)$ for ELU.

5 Results

5.1 Compression and generalization

Tishby and Zaslavsky (2015) and Shwartz-Ziv and Tishby (2017) claim that compression in $I(X;T)$ is necessary for generalization, but other work find the link between compression and generalization to be weak (Chelombiev et al., 2019; Goldfeld et al., 2019; Saxe et al., 2018; Schiemer and Ye, 2019). We study the relationship between compression and generalization on the domain of model size via double descent. When double descent is present, the generalization error and the model size exhibit a non-linear relationship. Is generalization error versus compression also a non-linear function?

We attempt to answer the question by directly studying how the compression varies when the model size changes, when double descent happens. For random feature networks with MSE loss, the compression in $I(X;T)$ is close to 0 for underparameterized models, as shown by the area of the red region in Fig. 2. More specifically, the two curves represent the initial and the final mutual information, respectively, and the red shaded region shows the difference between the two. If the number of parameters goes beyond the dataset size, the amount of compression shows an monotone non-decreasing curve. The same result holds for ReLU, ELU, and soft sign. For doubly-saturating activation functions (i.e., Tanh and soft sign), the curve of the amount of compression for overparameterized models is smooth, but it becomes a little bit noisy for singly-saturating activations (i.e., ReLU and ELU). In that case, some larger models may have slightly smaller compression, but the overall trend does not change. It is also worth noticing that large models with doubly-saturating activations tend to compress more than the singly-sided counterparts. This is reflected by the y-axis of the last column in Fig. 2. Therefore, **for linear networks with MSE loss, the positive relationship between compression and model size only holds for overparameterized models.**

While a clear pattern can be observed for MSE loss, the same result of the compression does not hold for random feature networks trained with cross-entropy loss. The quantity $I(X;T)$ at the last epoch barely changes across all model sizes, but the mutual information at the first epoch generally decreases as the number of parameters in the model increases. **For linear networks with cross-entropy loss, the compression in $I(X;T)$ does not correlate well with the model size.**

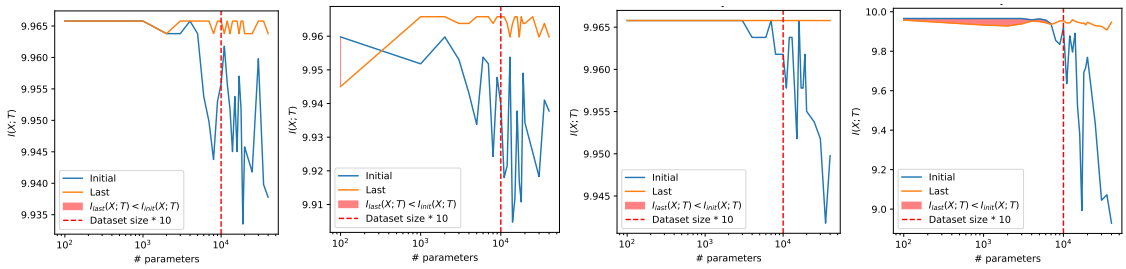


Figure 3: Random feature networks do not have the same compression behavior as observed earlier in Fig. 2 for MSE loss. The setup uses a random feature network with ELU (1), ReLU (2), soft sign (3), and Tanh (4) activations to produce non-linear features.

We further explore the pattern in fully-connected networks with different activations and observe the same results. Similar to random feature networks with cross-entropy loss, for all layers, an overparameterized fully-connected network obtains high generalization error with less compression. However, fully-connected networks still exhibit compression if they are underparamterized, and a slight amount of compression in the last layer is visible. It is now natural to ask if some property of the MSE loss has an effect on the intrinsic mechanism mutual information compression between the input and the internal representation. However, due to the difficulty [2] of training large neural networks with MSE loss, we leave that to future work. To this end, **for fully-connected networks, as the model size grows, the amount compression in all layers gets smaller.**

[2]In our experiments, we find it hard for large fully-connect networks with MSE to converge on classification tasks.

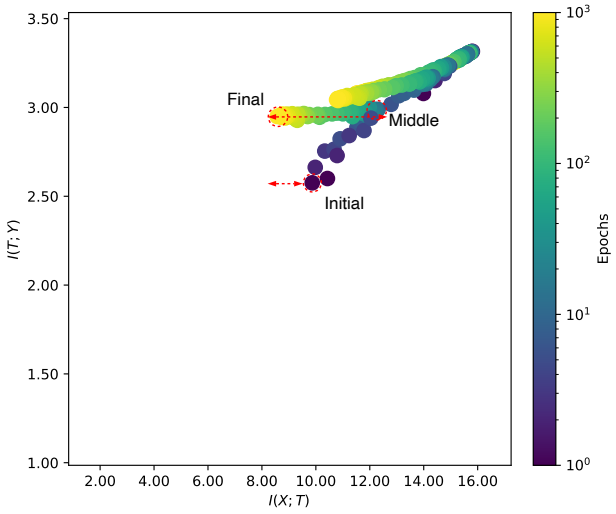


Figure 5: Initial - final vs. initial - middle.

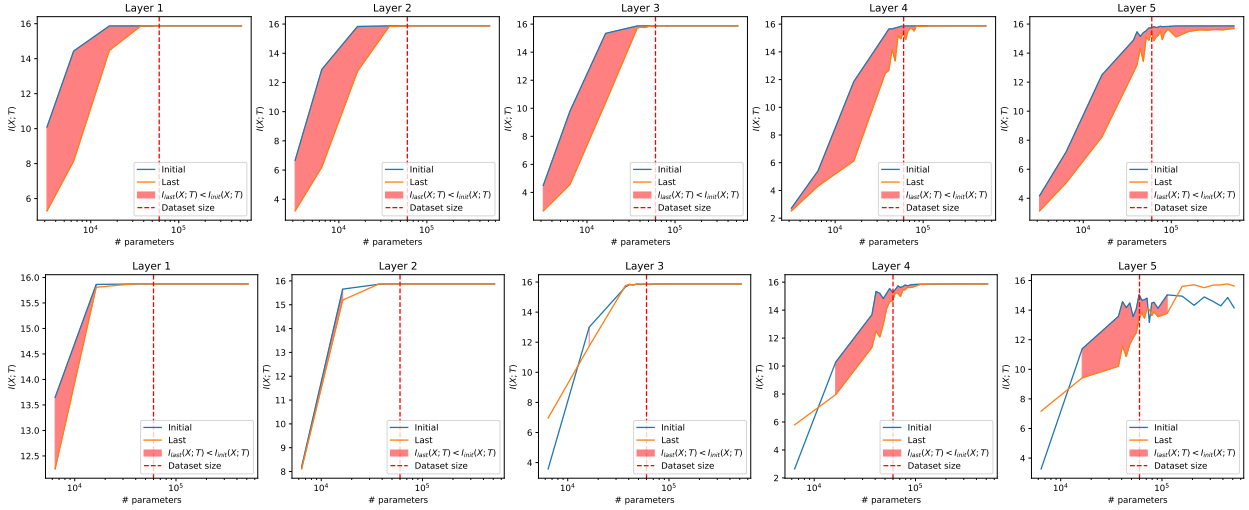


Figure 4: For fully-connected neural networks with Tanh (upper) and ReLU (lower) activations, the compression mostly happens to the underparameterized ones.

We also emphasize that the method we use to quantify the amount of compression might not be the only one to use. This is because subtracting the final $I(X;T)$ from the initial one does not always reflect the real amount of compression happened. For example, in Fig. 5 the quantity of initial - final might not be as expressive as middle - final, as we include both stages.

5.2 Phase transition and signal-to-noise ratio

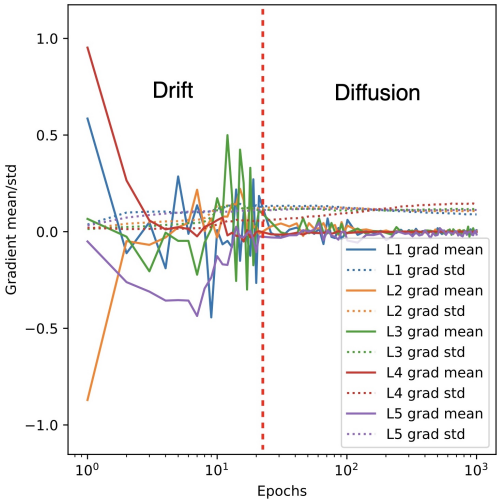
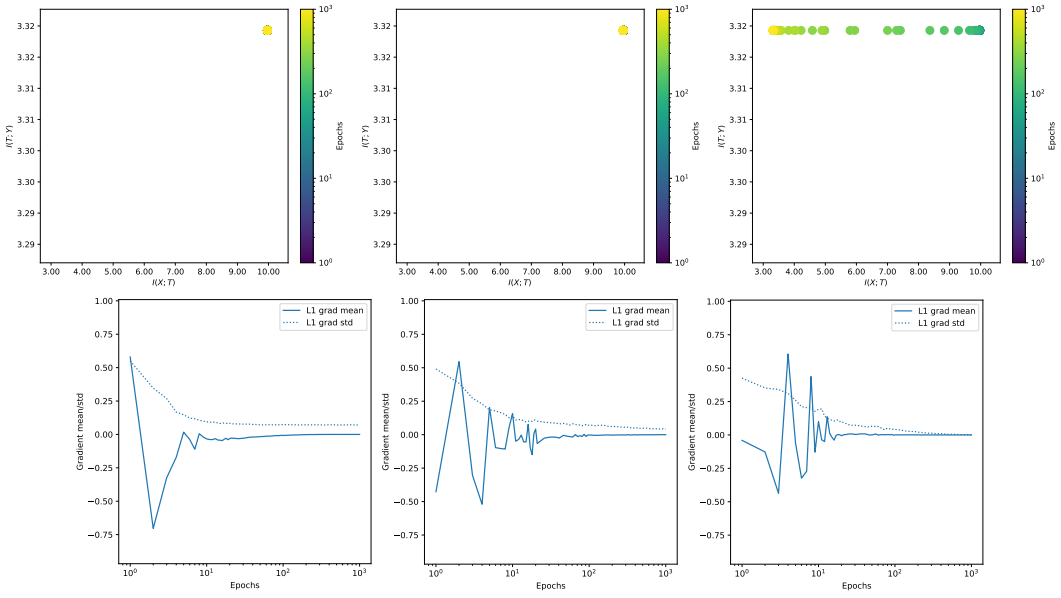


Figure 6: Drift phase vs. diffusion phase.

If the amount of compression in both linear and non-linear networks does not correlate with double descent or generalization, what does? Shwartz-Ziv and Tishby (2017) argues that a different perspective to analyze the problem is to view the signal-to-noise ratio (SNR) of the gradient on the course of training. More specifically, they find that at the phase transition on a information plane, there exists a change in the gradient SNR. In the drift phase, the gradient means are larger than their standard deviations, indicating a stable gradient flow and low stochasticity. In the diffusion phase, the gradient standard deviations are larger than its mean, corresponding to high noise. These two phases also correspond to high and low SNR, as illustrated in Fig. 6. The authors also prove that the diffusion phase is governed by a diffusion process, where the stationary distribution of such a process is characterized by a Focker-Plank equation, which maximizes the conditional entropy $H(X \mid T_i)$. Thus, the diffusion process is also minimizing the mutual information $I(X;T)$ because $I(X;T_i) = H(X) - H(X \mid T_i)$. Because $H(X)$ is a constant during training, maximizing $H(X \mid T_i)$ is equivalent to minimizing $I(X;T_i)$. Here, the entropy maximization of $H(X \mid T_i)$ is also known as a stochastic relaxation with low training error constraint.

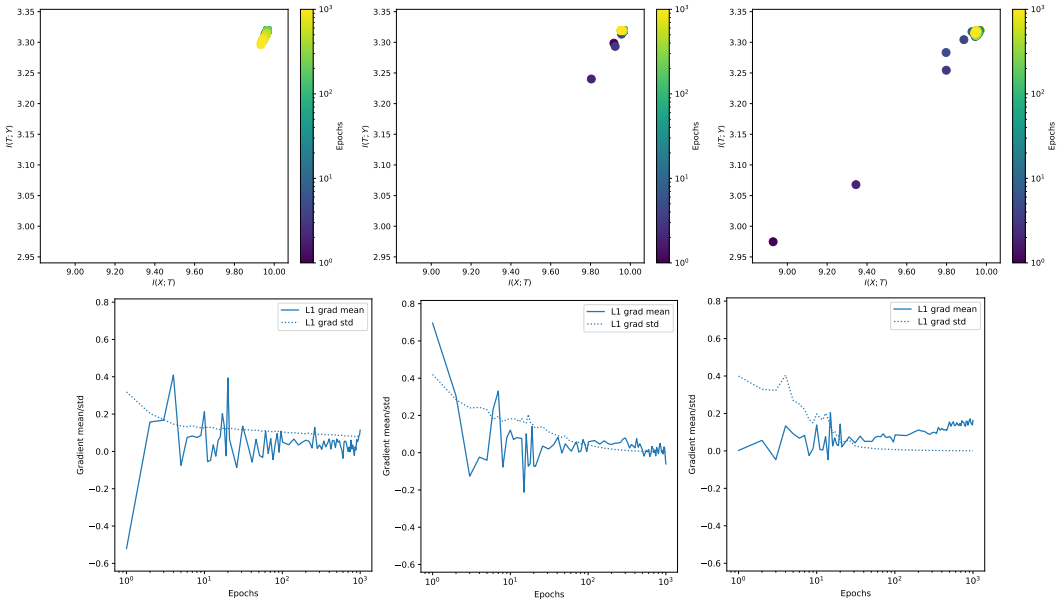
Below we view the SNR as a phase transition and the corresponding compression on an information plane. For linear random feature networks with MSE loss, as the model size grows, the gradient mean experiences more fluctuations, and the gradient deviation converges to 0 with a faster rate. In other words, while for models in all sizes networks trained with MSE loss almost always stay in the diffusion phase, the effect from the noise becomes weaker as the number of parameters increases. Therefore, for linear models with MSE loss, 1) the drift phase lasts longer for overparameterized networks and barely occurs for underparameterized models; 2) the gradient mean always converges to 0, but the gradient standard deviation only converges to 0 for large models.



(a) Underparameterized (b) $|\theta| \approx N$ (c) Overparameterized

Figure 7: Information plane and SNR of linear networks with MSE loss.

For linear networks trained with cross-entropy loss, their final mutual information $I(X;T)$ does not move much, but the initial quantities are smaller for larger networks. Similar to those with MSE loss, the gradient standard deviation also converges to 0 at a faster rate as the number of a parameter in a network increases, but the gradient mean diverges for large model sizes. From the phase transition perspective, small linear models with CE loss, similar to the observation in Shwartz-Ziv and Tishby (2017), transitions from drift phase to diffusion phase, but then reverts back into the drift phase. This might be the reason why large CE networks expands the mutual information instead of compressing it, but it requires further study to confirm this hypothesis.

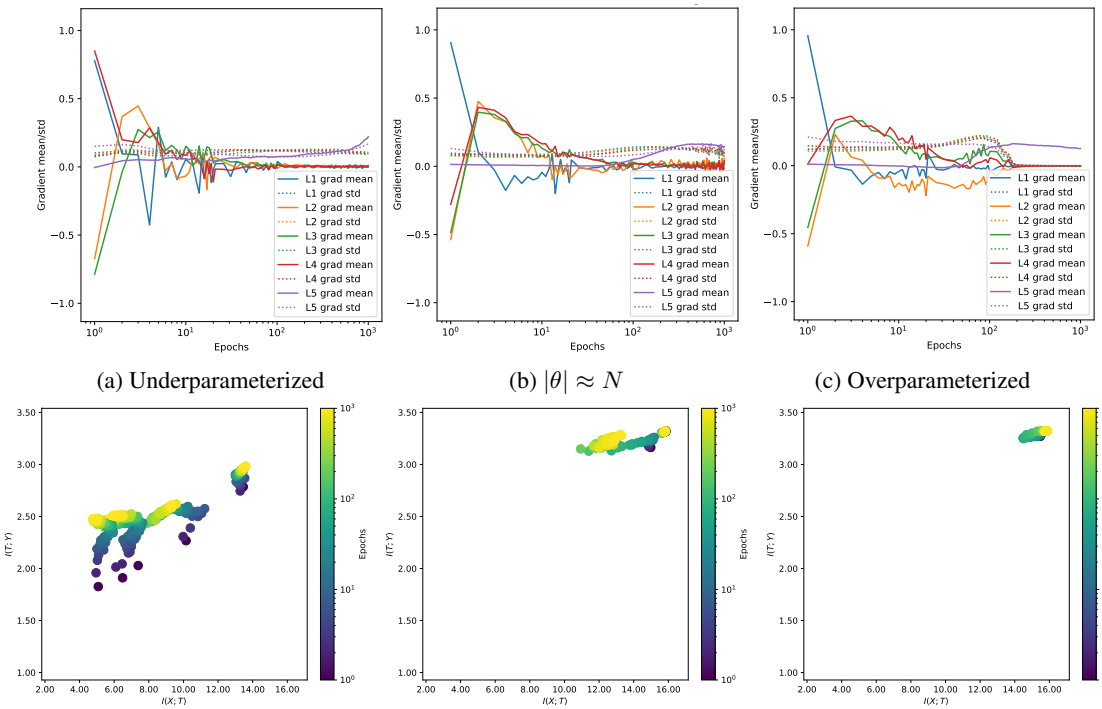


(a) Underparameterized (b) $|\theta| \approx N$ (c) Overparameterized

Figure 8: Information plane and SNR of linear networks with CE loss.

For fully-connected neural networks, their SNR also follow a typical phase transition from the drift phase to the diffusion phase. In Fig. 9, a convergence in gradient standard deviation in all layers can be observed, and the gradient means for all layers converge, except for the last layer. The small network undergoes drift and diffusion phases; the medium network stays in the diffusion phase but has noisy gradient mean and standard deviation; the large network escapes the diffusion phase. It is also interesting that the gradient mean of the last layer is initially 0 but increasing during training. By viewing the corresponding information plane, we can summarize Fig. 9 by the findings below:

- Underparametrized networks enter the diffusion phase and stays, which possibly leads to more compression;
- $|\theta| \approx N$ networks behaves similar to underparameterized ones, but they generally compress less and have noisy gradients;
- Overparameterized networks experience the diffusion phase for a short amount of time and immediately becomes stationary except for the last layer, which possibly yields high SNR.
- The "stationary" phase is unique to overparameterized networks.



(a) Underparameterized (b) $|\theta| \approx N$ (c) Overparameterized

(d) Underparameterized (e) $|\theta| \approx N$ (f) Overparameterized

Figure 9: Upper: The left, middle, and right figures correspond to neural networks with size smaller, equal, and larger than the dataset size, respectively. Networks in (b) and (c) have the same training error (i.e., 0). Lower: Left: All layers still compress. Middle: $I(X;T)$ in the last layer compresses, but other layers do not move. Right: $I(X;T)$ in the last layer expands, all other layers becomes almost stationary.

6 Conclusion

In this work, we study double descent and generalization form an information-theoretic perspective. Specifically, we estimate the information flow in linear models and fully-connected neural networks trained with a variety of settings and visualize the the mutual information paths of $I(X;T)$ and $I(Y;T)$ information plane. Our results suggests that the amount of compression in $I(X;T)$ might not be a good proxy for generalization. This is because the compression cannot reflect the non-linear function of generalization error versus model size and exhibits different compression behaviors in some settings. However, we do emphasize the importance of the phase transition perspective with the signal-to-noise ratio of the gradient. Our analysis shows that the point when phase transition occurs and the noisiness of the gradient distinguishes underparameterized, $|\theta| \approx N$, and overparameterized networks: 1) in underparamterized networks, all layer compress; 2) If $|\theta| \approx N$, only the last layer compresses; 3) if a network is overparameterized, the

last layer expands, and all other layers become stationary. This is also related to the previous controversy of whether compression is causally linked to generalization. Therefore, we hope this work can be considered as a partial answer to the controversy and shed some light in the relationship between compression and generalization, and even the black-box nature of large neural networks in general.

References

Adlam, B. and Pennington, J. (2020). The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 74–84. PMLR.

Amjad, R. A. and Geiger, B. C. (2019). Learning representations for neural network-based classification using the information bottleneck principle. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2225–2239.

Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Hjelm, R. D., and Courville, A. C. (2018). Mutual information neural estimation. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 530–539. PMLR.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2018). Reconciling modern machine learning practice and the bias-variance trade-off. *ArXiv preprint*, abs/1812.11118.

Brekelmans, R., Huang, S., Ghassemi, M., Ver Steeg, G., Grosse, R. B., and Makhzani, A. (2021). Improving mutual information estimation with annealed and energy-based bounds. In *International Conference on Learning Representations*.

Chelombiev, I., Houghton, C., and O'Donnell, C. (2019). Adaptive estimators show information compression in deep neural networks. *arXiv preprint arXiv:1902.09037*.

Cheng, H., Lian, D., Gao, S., and Geng, Y. (2018). Evaluating capability of deep neural networks for image classification via information plane. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 168–182.

Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. (2011). High-performance neural networks for visual object classification. *ArXiv preprint*, abs/1102.0183.

Duin, R. P. (2000). Classifiers in almost empty spaces. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 2, pages 1–7. IEEE.

Frankle, J. and Carbin, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Gabrié, M., Manoel, A., Luneau, C., Barbier, J., Macris, N., Krzakala, F., and Zdeborová, L. (2018). Entropy and mutual information in models of deep neural networks. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1826–1836.

Geiger, B. C. (2021). On information plane analyses of neural network classifiers–a review. *IEEE Transactions on Neural Networks and Learning Systems*.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58.

Goldfeld, Z. and Greenewald, K. (2021). Sliced mutual information: A scalable measure of statistical dependence. *Advances in Neural Information Processing Systems*, 34:17567–17578.

Goldfeld, Z., van den Berg, E., Greenewald, K. H., Melnyk, I., Nguyen, N., Kingsbury, B., and Polyanskiy, Y. (2019). Estimating information flow in deep neural networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2299–2308. PMLR.

Guo, Q., Chen, J., Wang, D., Yang, Y., Deng, X., Carin, L., Li, F., and Tao, C. (2021). Tight mutual information estimation with contrastive fenchel-legendre optimization. *ArXiv preprint*, abs/2107.01131.

Heckel, R. and Yilmaz, F. F. (2020). Early stopping in deep networks: Double descent and how to eliminate it. *ArXiv preprint*, abs/2007.10099.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2019). Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jacot, A., Hongler, C., and Gabriel, F. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8580–8589.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Lee, E. H. and Cherkassky, V. (2022). Vc theoretical explanation of double descent. *ArXiv preprint*, abs/2205.15549.

Liu, C., Zhu, L., and Belkin, M. (2020). On the linearity of large non-linear models: when and why the tangent kernel is constant. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Loog, M., Viering, T., Mey, A., Krijthe, J. H., and Tax, D. M. (2020). A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20):10625–10626.

Lorenzen, S. S., Igel, C., and Nielsen, M. (2021). Information bottleneck: Exact analysis of (quantized) neural networks. *ArXiv preprint*, abs/2106.12912.

Molavipour, S., Bassi, G., and Skoglund, M. (2021). Neural estimators for conditional mutual information using nearest neighbors sampling. *IEEE Transactions on Signal Processing*, 69:766–780.

Mukherjee, S., Asnani, H., and Kannan, S. (2019). CCMI : Classifier based conditional mutual information estimation. In Globerson, A. and Silva, R., editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 1083–1093. AUAI Press.

Nacson, M. S., Srebro, N., and Soudry, D. (2019). Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3051–3059. PMLR.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. (2019). Deep double descent: where bigger models and more data hurt. iclr 2020. *ArXiv preprint*, abs/1912.02292.

Nakkiran, P., Venkat, P., Kakade, S., and Ma, T. (2020). Optimal regularization can mitigate double descent. *ArXiv preprint*, abs/2003.01897.

Opper, M., Kinzel, W., Kleinz, J., and Nehl, R. (1990). On the ability of the optimal perceptron to generalise. *Journal of Physics A: Mathematical and General*, 23(11):L581.

Paninski, L. (2003). Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253.

Papyan, V., Han, X., and Donoho, D. L. (2020). Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117:24652–24663.

Penrose, R. (1956). On best approximate solutions of linear matrix equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 17–19. Cambridge University Press.

Poole, B., Ozair, S., van den Oord, A., Alemi, A., and Tucker, G. (2019). On variational bounds of mutual information. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5171–5180. PMLR.

Raudys, S. and Duin, R. P. (1998). Expected classification error of the fisher linear classifier with pseudo-inverse covariance matrix. *Pattern recognition letters*, 19(5-6):385–392.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. (2018). On the information bottleneck theory of deep learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Schiemer, M. and Ye, J. (2019). Revisiting the information plane.

Shwartz-Ziv, R. and Tishby, N. (2017). Opening the black box of deep neural networks via information. *ArXiv preprint*, abs/1703.00810.

Skurichina, M. and Duin, R. P. (1998). Regularization by adding redundant features. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 564–572. Springer.

Song, J. and Ermon, S. (2020). Understanding the limitations of variational mutual information estimators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., and Morcos, A. S. (2022). Beyond neural scaling laws: beating power law scaling via data pruning. *arXiv preprint arXiv:2206.14486*.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. (2018). The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878.

Stephenson, C. and Lee, T. (2021). When and how epochwise double descent happens. *ArXiv preprint*, abs/2108.12006.

Thomas, M. and Joy, A. T. (2006). *Elements of information theory*. Wiley-Interscience.

Tishby, N. and Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE.

Vallet, F., Cailton, J.-G., and Refregier, P. (1989). Linear and nonlinear extension of the pseudo-inverse solution for learning boolean functions. *EPL (Europhysics Letters)*, 9(4):315.

Watkin, T. L., Rau, A., and Biehl, M. (1993). The statistical mechanics of learning a rule. *Reviews of Modern Physics*, 65(2):499.

Wen, L., Zhou, Y., He, L., Zhou, M., and Xu, Z. (2020). Mutual information gradient estimation for representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Yang, Z., Yu, Y., You, C., Steinhardt, J., and Ma, Y. (2020). Rethinking bias-variance trade-off for generalization of neural networks. In *International Conference on Machine Learning*, pages 10767–10777. PMLR.