On the Unlearnability of Deep Neural Networks

Authors: D. Moshkovitz & M. Moshkovitz & N. Tishby [MT17, MM17]
Presenter: TT Nguyen [1]

SAIL@UNIST

September 1, 2018

[1] nguyent2792@gmail.com

Contents

Motivation

UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION

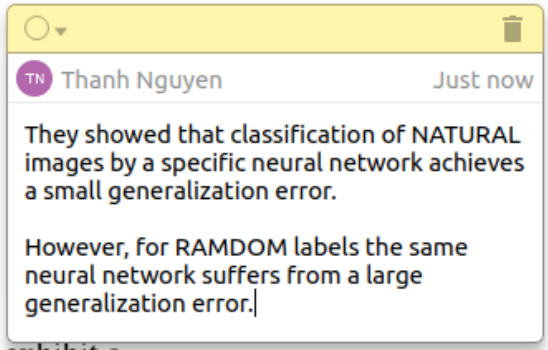
Chiyuan Zhang*
Massachusetts Institute of Technology
chiyuan@mit.edu

Samy Bengio
Google Brain
bengio@google.com

Moritz Hardt
Google Brain
mrtz@google.com

Benjamin Recht†
University of California, Berkeley
brecht@berkeley.edu

Oriol Vinyals
Google DeepMind
vinyals@google.com



ABSTRACT

Despite their massive size, successful deep artificial neural networks can exhibit a remarkably small difference between training and test performance. Conventional wisdom attributes small generalization error either to properties of the model family, or to the regularization techniques used during training.

Through extensive systematic experiments, we show how these traditional approaches fail to explain why large neural networks generalize well in practice. Specifically, our experiments establish that state-of-the-art convolutional networks for image classification trained with stochastic gradient methods easily fit a random labeling of the training data. This phenomenon is qualitatively unaffected by explicit regularization, and occurs even if we replace the true images by com-

Motivation

- **Model family** and **regularization techniques** fail explain why large (Deep Neural Networks) DNNs generalize well in practice
- Provoke a call for a **new measure for generalization in DNNs**
- **Clue**: Not everything can be learned by DNNs.

What is this work about?

- Introduce new concepts and theorems to **quantify what cannot be learned** by DNNs in particular and bounded-memory algorithm in general.
- The main achievement of this work:

Theorem

Close-to-random hypothesis classes cannot be learned by bounded-memory algorithms of which DNNs (with SGD) is a particular bounded-memory algorithm

- In today's talk[2], we will
  - walk through new relevant concepts
  - formally state the main theorem of the unlearnability of DNNs
  - and provide some intuitions behind theorem

[2] For formal proof and implication of the main theorem, we will present it in another dedicated talk

Notations

- Data set: $\mathcal{X}$ is a finite set of binary-labeled data (x, b) where $b \in (0, 1)$
- Hypothesis class: $\mathcal{H}$ is a finite set of binary hypotheses $h : \mathcal{X} \to \{0, 1\}$, e.g., DNNs define a hypothesis class
- A **learning algorithm**: receives training examples $S = (x_i, b_i)_{i=1}^N$ where $(x_i, b_i) \in \mathcal{X} \times \{0, 1\}$, and outputs a **hypothesis** $h : \mathcal{X} \to \{0, 1\}$
- The goal of a learning algorithm is to minimize the **test error**:

$$e_{(\mathcal{D},f)}(h) := Pr_{x\sim\mathcal{D}}[h(x) \neq f(x)] \tag{1}$$

where $\mathcal{D}$ is the (unknown) data distribution, and f is the true hypothesis.

Notations

- **Realizability assumption**: There is a hypothesis in the class with a test error equal to 0
- **Training error**:

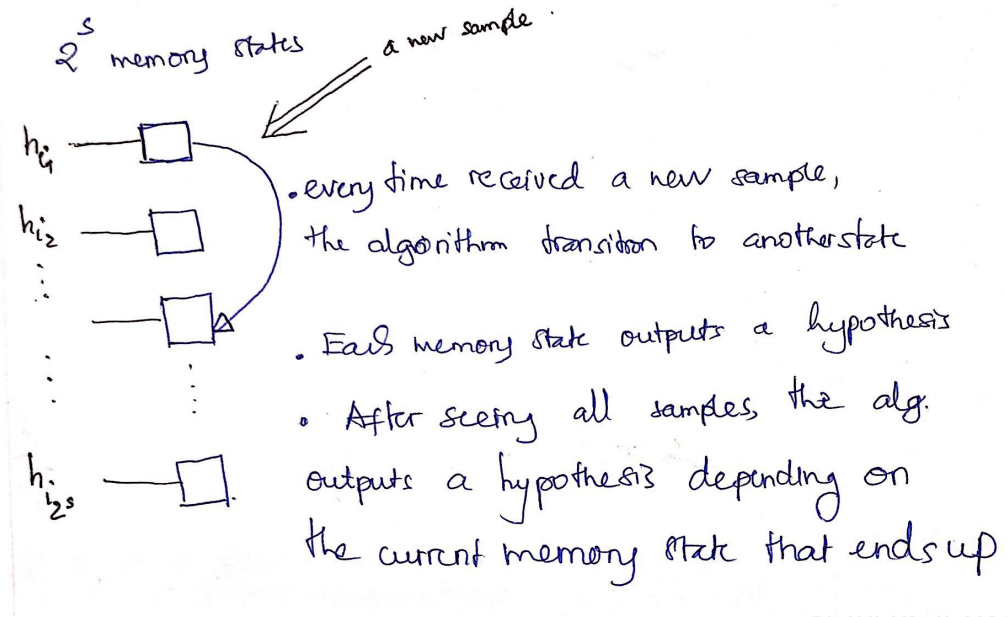
$$e_{(S,f)}(h) := Pr_{x\sim\mathcal{S}}[h(x) \neq f(x)] \tag{2}$$

- **Generalization error**:

$$e_{gen}(h) := e_{(S,f)}(h) - e_{(\mathcal{D},f)}(h) \tag{3}$$

Bounded-memory algorithms

A bounded-memory algorithm is a Turing machine with a bounded size tape s in which each cell is binary:



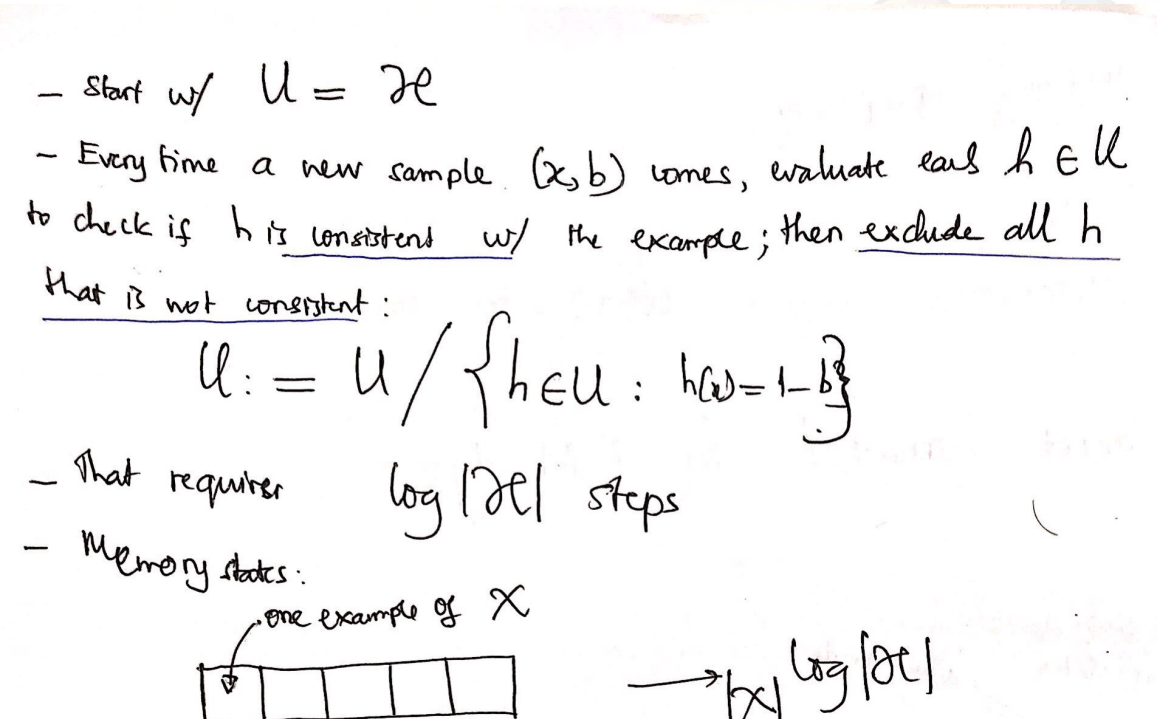
e.g., SGD is a bounded-memory algorithm: When it gets a new example it changes the current weights of the neural network, based on the appropriate gradient.

Trivial bounds

Lemma (Learnability with Unbounded memory)

One can learn any hypothesis class $\mathcal{H}$ with $O(\log|\mathcal{H}|)$ examples with $|\mathcal{X}|^{O(\log|\mathcal{H}|)}$ memory states

Proof:

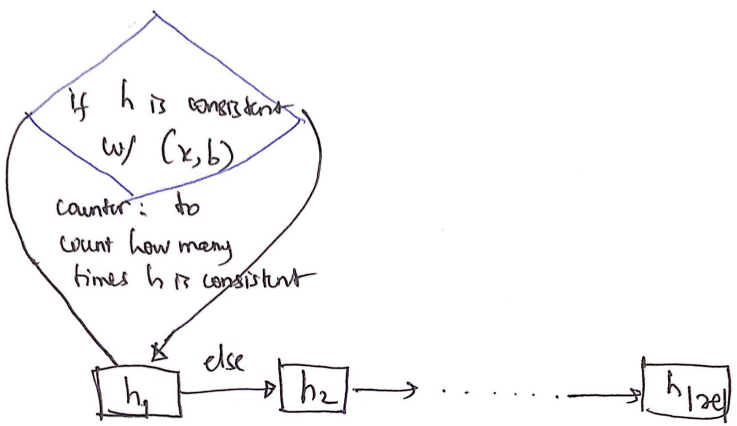


Trivial bounds

Lemma (Upper Bounds on # examples for Learnability)

It requires at least $|\mathcal{H}|$ memory states for learning a hypothesis class $\mathcal{H}$. In addition, at the minimum number of memory states, $|\mathcal{H}|$, $O(|\mathcal{H}| \log |\mathcal{H}|)$ examples always suffice for learning $\mathcal{H}$.

Proof: We iterate every hypothesis h from the hypothesis class $\mathcal{H}$, each of which needs $\log |\mathcal{H}|$ examples to rule out h.



Definition of Unlearnability

Definition (Unlearnability)

A learning algorithm is said to be unable to learn $\mathcal{H}$ if it requires at least $|\mathcal{H}|^c$ for some $c > 0$ to learn the class. Note that $|\mathcal{H}|^c$ is exponentially larger than the number of examples, $\log |\mathcal{H}|$, needed for learning with unbounded memory.

Motivating question for the main theorem

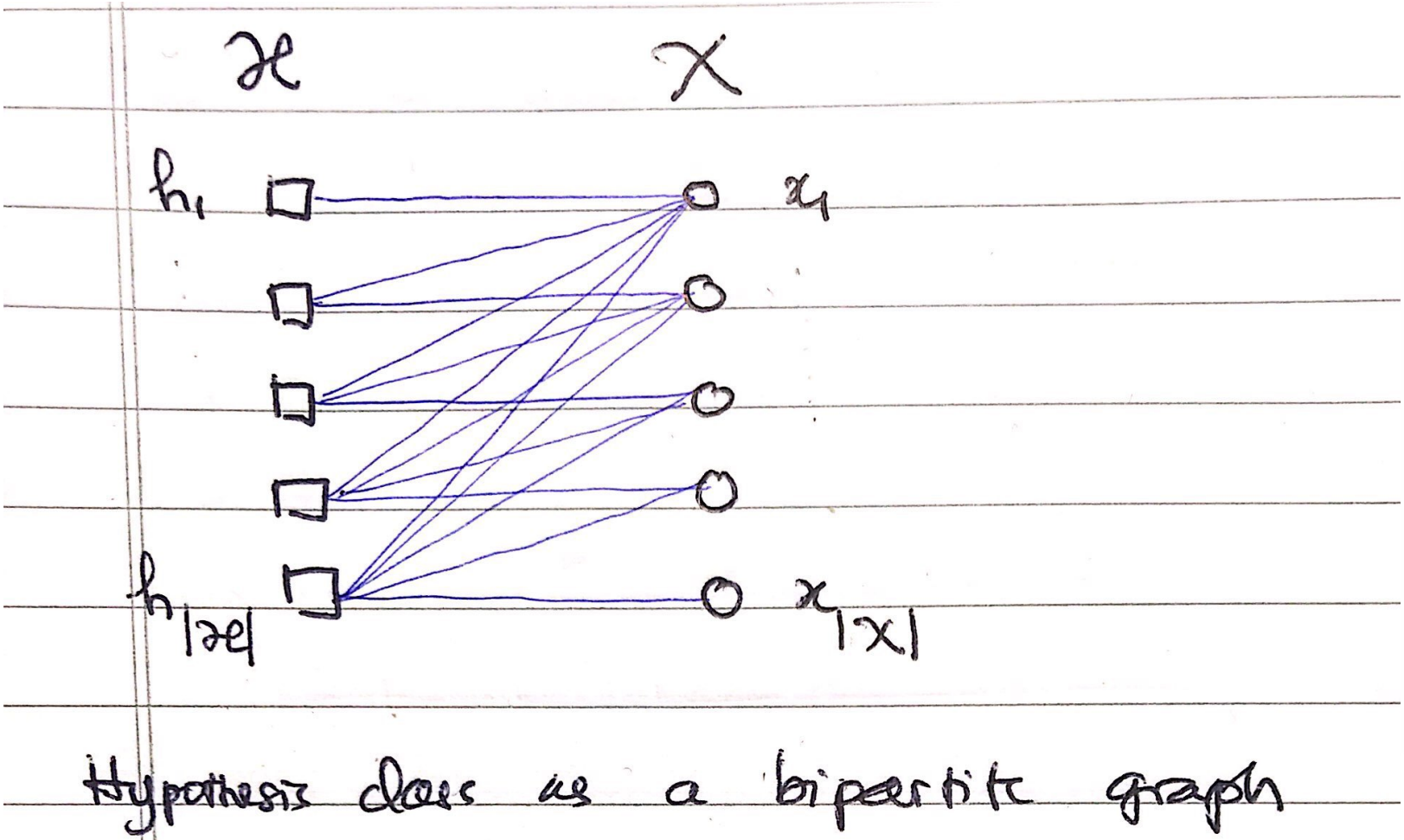
Given a memory constraint $O(|\mathcal{H}|^{1+\epsilon})$ for some $\epsilon > 0$, to suffice for learning $\mathcal{H}$, the upper bound on the number of examples is $O(|\mathcal{H}| \log |\mathcal{H}|)$, what is a **lower bound**?

Answer: For close-to-random hypothesis class [explained next], we cannot learn the class with bounded memory.

Q: What does it mean by "**randomness of a hypothesis class**"? $\rightarrow$ We need the concept of **hypothesis graph** and **d-mixing**

Hypothesis Graph

There is an edge btw $h \in \mathcal{H}$ and $x \in \mathcal{X}$ iff $h(x) = 1$.



d-mixing: Quantifying the randomness of a hypothesis class

Definition (d-mixing)

A bipartie graph $G = (A, B, E)$ is d-mixing if $\forall T \subset A, \forall S \subset B$ with $|S| = s, |T| = t$, it holds that:

$$\left| e(S, T) - \frac{st}{2} \right| \leq d\sqrt{st} \tag{4}$$

Intuition:

- The number of edges between any two subset S and T will be close to their average, up to some constant d of the standard deviation *sqrtst*
- Smaller d means any two subset S and T has $s(S, T)$ is closer to its average $\rightarrow$ the graph is closer to random

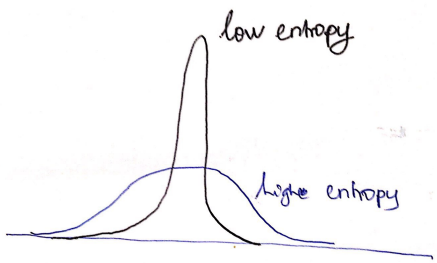
Mixing class

Definition (Mixing class)

A class $\mathcal{H}$ is mixing if it is $O(\sqrt{|X|})$-mixing

Intuition:

- A mixing class is a class that is **close enough** to random class.
- For a mixing class: Assume that the underlying hypothesis h_{true} was taken from set S, knowing that h_{true} is consistent with at least one sample of T **reveal little information about** h_{true}
- Analogy: More random $\rightarrow$ high entropy $\rightarrow$ more uncertainty



Main theorem

Theorem

Suppose $\mathcal{H}$ *is* $\sqrt{|\mathcal{X}||\mathcal{H}|^a}$*-mixing for* $a \in [0,1]$*, then* $\forall s \in (0,1), \exists s' > 0$ *s.t. any algorithm for* $\mathcal{H}$ *that has at most* $O(|\mathcal{H}|^{1.25-s-3a})$ *memory states require at least* $|\mathcal{H}|^{s'}$ *examples to return the underlying hypothesis (or an approximation of it) with probability of* $1/3$*.*

Intuition:

- Any bounded-memory algorithm cannot learn a mixing class.
- For a learning algorithm that has at most $|\mathcal{H}|^{1.25}$ memory states, the number of examples needed to learn a mixing class is exponentially larger than in case of unbounded memory.

References I

- Michal Moshkovitz and Dana Moshkovitz, *Mixing implies lower bounds for space bounded learning*, Electronic Colloquium on Computational Complexity (ECCC) **24** (2017), 17.
- Michal Moshkovitz and Naftali Tishby, *Mixing complexity and its applications to neural networks*, CoRR **abs/1703.00729** (2017).