A Universal Law of Robustness via Isoperimetry

Mark Sellke (Stanford)

Joint with Sébastien Bubeck (MSR)

https://arxiv.org/abs/2105.12806

Adversarial Examples

- “Axioms” for today: Modern neural networks…
  1. Memorize their training data near-perfectly.
  2. Are vulnerable to small perturbations.
- What is going on?

- Some hypotheses:
  - 1. Robust memorization is computationally hard
  - 2. Neural networks cannot memorize robustly
  - 3. Robust memorization needs more data
  - 4. Robust memorization requires large models

Adversarial Examples

- “Axioms”: Neural networks…
  1. Memorize training data.
  2. Are vulnerable to small perturbations.
- What is going on?
- Some hypotheses:
  - 1. Robust memorization is computationally hard
  - 2. Neural networks cannot memorize robustly
  - 3. Robust memorization needs more data
  - **4. Robust memorization requires large models**

Law of Robustness

The Model of Memorization

- Input: $n = d^{O(1)}$ random points $x_1, \dots x_n$ on d-dimensional unit sphere.
- Labels $y_i = g(x_i) + Z_i$: signal + noise.
  - Noise variance = σ^2.

- Perfect memorization: $f: \mathbb{R}^d \to \mathbb{R}$ fits data perfectly:

$$f(x_i) = y_i, \quad \mathrm{i} \in \{1,2, \dots, n\}.$$

- Partial memorization: fit data *much better than the signal:*

$$\sum_i (f(x_i) - y_i)^2 \leq \frac{1}{2} \sum_i Z_i^2.$$

Robustness and Memorization

- Definition: a function $f: \mathbb{R}^d \to \mathbb{R}$ is L-robust if $Lip(f) \leq L$, i.e.

$$|f(x) - f(x')| \leq L||x - x'||.$$

  - Reason: Lipschitz implies robustness to adversarial perturbations!
  - This is a strong notion of robustness.
- Fact: w.h.p, perfect memorization is possible with an $O(1)$-robust function.
  - Proof: w.h.p, $|x_i - x_j| \geq 0.1$ for all $i, j.$ Follows from Kirszbraun extension theorem.
  - This is abstract and non-constructive…
- How **complicated** does a good memorizer need to be?
- More precisely: how large a function class $\mathcal{F}$ must be fixed **beforehand** to contain a (robust) memorizer w.h.p?

Size vs Robustness

- Q: if some f $\in \mathcal{F}$ (robustly) memorizes, how large is the function class $\mathcal{F}$?
- Measure size by # parameters P. Formally: $w \to f_w \in \mathcal{F}$ for $w \in \mathbb{R}^P$ with:

$$|w| \le poly(d), \qquad |f_w(x) - f_{w'}(x)| \le poly(d) \cdot |w - w'| \ \forall w, w', x.$$

  - Captures "true" parameter count for convolutional networks, weight sharing, …
  - P is # parameters in the **model class**
    - Count all possible weights even under post-training sparsification.
- Fact: $P = n$ parameters suffice to memorize
  - [Baum 1988]: use a 2-layer neural network with n/d neurons. Not robust.
- Fact: $P = nd$ parameters suffice to *robustly* memorize.
  - Put 1 radial basis function on each input. Each RBF specified by d parameters.

A Universal Law of Robustness

- Conjecture [Bubeck-Li-Nagaraj 20]: $Lip(f) \geq \sqrt{\frac{nd}{P}}$ for 2-layer neural networks.
- Theorem [Bubeck-S. 21]: for P-parameter function classes $\mathcal{F}$, partial memorization of noisy data by some f ∈ $\mathcal{F}$ implies:

$$Lip(f) \gg \sigma \sqrt{\frac{nd}{P}}.$$

  - Input distribution can be a mixture of $n^{0.99}$ *isoperimetric* components.
  - Heteroscedastic noise is also fine. Just need $\sigma^2 = \mathbb{E}\left[Var[y_i|x_i]\right]$.

- Tight for any $P \gg n$: project down to dimension $\tilde{d} = P/n$, use n RBFs in $\mathbb{R}^{\tilde{d}}$.

Isoperimetry

- Key property of high-dimensional space: **isoperimetry**. Many related definitions.

- Relevant Definition: μ is c-isoperimetric if for any L-Lipschitz $f: \mathbb{R}^d \to \mathbb{R}$,

$$\mathbb{P}^{\mu}[|f(x) - \mathbb{E}^{\mu}[f]| \geq t] \leq 2e^{-\frac{dt^2}{2cL^2}}$$

- Applies to many “genuinely high-dimensional” distributions
  - Sphere/Gaussian
  - Cube with Hamming distance
  - Negatively curved manifolds, Gaussian plus small independent noise,…
  - Holds when μ has a nice **log-Sobolev constant**.

Interpretation

- Real datasets are mixtures
  - Cat component vs dog component.
  - 1 cat, 2 cat, red cat, blue cat?
  - Components could have small diameter or live on a lower-dimensional manifold.
    - Optimistically, law of robustness holds with appropriate *effective dimension*.
    - Determine naïve vs effective dimension scaling empirically to extrapolate?

- What is noise?
  - In theory: no noise → nothing to learn
  - Real life: noise is “complicated” part of the function?
    - Learning algorithms may have “inductive bias” that helps to learn the simple part.

MNIST and ImageNet

- Back-of-the-envelope on robust ImageNet leads to realistic modern parameter scale.
  - Lots more work needed to make a real prediction. Goal is to illustrate potential for scaling laws.
- MNIST results from [MMSTV 18]:
  - $n \approx 10^5, d = 28^2 \approx 10^3$.
  - Good robust accuracy achieved at $\mathrm{P} \approx 10^6$ parameters.
  - Effective dimension $\hat{d} \approx \frac{P}{n} = 10^1 = d/100$?
- ImageNet
  - $n_I \approx 10^7, d_I \approx 10^5$.
  - Prediction: $P_I \approx n_I \widehat{d_I} = \frac{n_I d_I}{100} \approx \mathbf{10^{10}}$.
  - ImageNet pictures "seem" more complicated than MNIST, so maybe 10^{11}?
  - Current models: typically $P \approx 10^9$.

Proportionality constant for image effective dimension?

Generalization Perspective

- Recall: small Rademacher complexity $\mathcal{R}_{\mathcal{F}}$ implies uniform generalization for all f ∈ $\mathcal{F}$.

- Classically, function class $\mathcal{F}$ has Rademacher complexity

$$\mathcal{R}_{\mathcal{F}} \leq \sqrt{\frac{\log|\mathcal{F}|}{n}} \approx \sqrt{\frac{P}{n}}.$$

- Theorem: for Lipschitz function classes $\mathcal{F}$ and mixtures of isoperimetric distributions,

$$\mathcal{R}_{\mathcal{F}} \leq \sqrt{\frac{P}{n{\color{red}d}}}.$$

- Consequence: law of robustness holds for *any Lipschitz loss function* (not just square-loss).

Open Directions

- Other norms
  - Just need Lipschitz functions to concentrate. When does this hold in e.g. infinity or Wasserstein norm?
- More refined notions of robustness
  - Sobolev norms like $\mathbb{E}^{\mu}|\nabla f(x)|^2$ don't work. Need small gradient **everywhere**.
  - Connect more precisely to robust test error?
  - Algorithmic law of robustness for gradient-based training?
    - Might not require noisy labels.
- Empirical study and Architecture-Specific Scaling Laws
  - Could there be different slightly different laws of robustness for CNNs, transformers, …?