Lecture 17: Midterm 1 and problem-solving

Nisha Chandramoorthy

October 24, 2023

Last time

- VCDim of FCNN, VC generalization bounds, then implementation

Last time

- VCDim of FCNN, VC generalization bounds, then implementation
- Today: Midterm 1 and problem-solving

Last time

- VCDim of FCNN, VC generalization bounds, then implementation
- Today: Midterm 1 and problem-solving
- After this: CNNs, VAEs, feature extraction/Dimension reduction

Asking questions

Understanding Deep Learning (Still) Requires Rethinking Generalization

By **Chiyuan Zhang**, **Samy Bengio**, **Moritz Hardt**, **Benjamin Recht**, and **Oriol Vinyals**

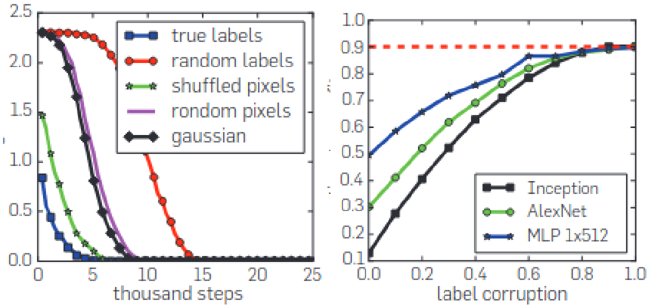
- Training data consists of random labels.

Asking questions

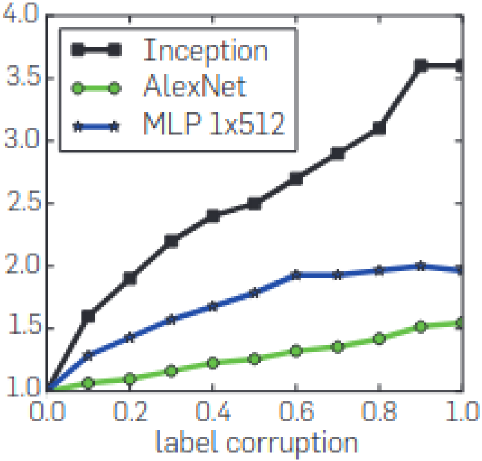
Understanding Deep Learning (Still) Requires Rethinking Generalization

By Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals

- Training data consists of random labels.

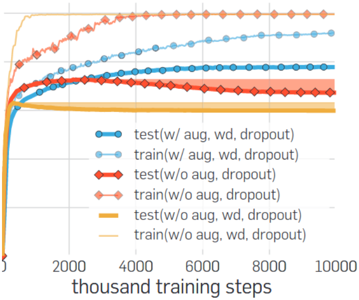


Does our understanding of generalization hold?

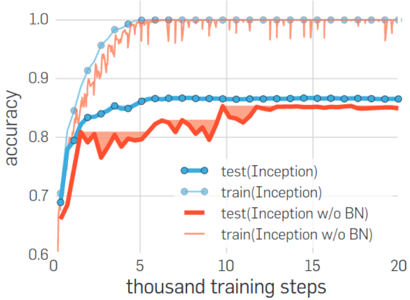


- Hand-wavy: test error is higher when we expect complexity to be higher.

How to improve generalization?



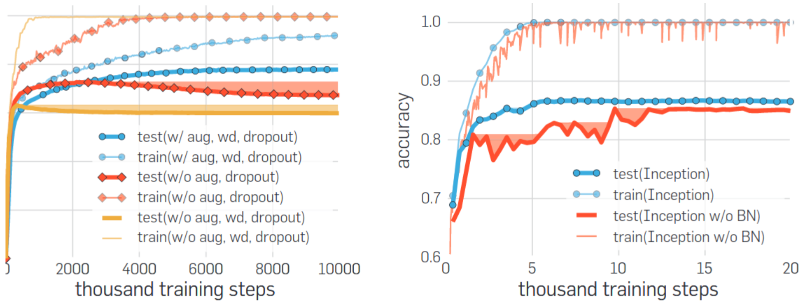
(a) Inception on ImageNet



(b) Inception on CIFAR10

- Early stopping (implicit regularization)

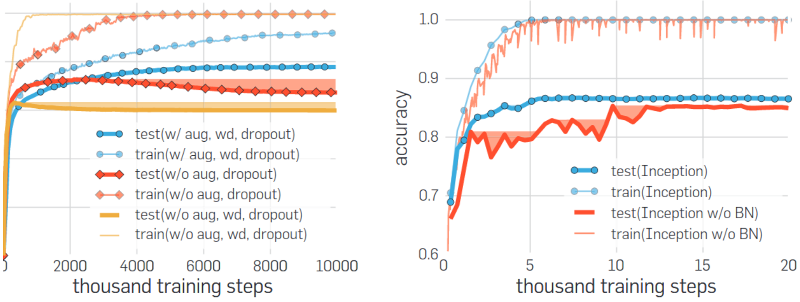
How to improve generalization?



(a) Inception on ImageNet (b) Inception on CIFAR10

- Early stopping (implicit regularization)
- Batch normalization: normalize the inputs to each layer for each mini-batch, i.e., make the inputs have zero mean and unit variance.

How to improve generalization?



(a) Inception on ImageNet

(b) Inception on CIFAR10

- Early stopping (implicit regularization)
- Batch normalization: normalize the inputs to each layer for each mini-batch, i.e., make the inputs have zero mean and unit variance.
- Dropout: randomly set some activations to zero.

What is the conclusion?

- Bias-complexity tradeoff

What is the conclusion?

- Bias-complexity tradeoff
- Need to get new ideas/insights into generalization

What is the conclusion?

- Bias-complexity tradeoff
- Need to get new ideas/insights into generalization
- The role of the data distribution...

What is the conclusion?

- Bias-complexity tradeoff
- Need to get new ideas/insights into generalization
- The role of the data distribution...
- Bubeck Sellke 2021:

2. *The distribution μ of the covariates x_i satisfies isoperimetry (or is a mixture theoreof).*
3. *The expected conditional variance of the output (i.e., the "noise level") is strictly positive, denoted $\sigma^2 := \mathbb{E}^{\mu}[Var[y|x]] > 0$.*

Then, with high probability over the sampling of the data, one has simultaneously for all $f \in \mathcal{F}$:

$$\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 \leq \sigma^2 - \epsilon \;\Rightarrow\; \mathrm{Lip}(f) \geq \widetilde{\Omega}\left(\epsilon\sqrt{\frac{nd}{p}}\right).$$

What are the assumptions?

- Isoperimetry: if for an l-Lipschitz function $h : \mathbb{R}^d \to \mathbb{R}$, $\mathbb{P}[|h(X) - Eh| \geqslant t] \leqslant 2e^{(-dt^2)/(2cl^2)}$, then, the distribution of X is c-isoperimetric.
- for learning smooth functions ($\mathrm{Lip}(f) \leqslant l$), the number of parameters is $\Omega(nd\epsilon^2/l)$

What are the assumptions?

- Isoperimetry: if for an l-Lipschitz function $h : \mathbb{R}^d \to \mathbb{R}$, $\mathbb{P}[|h(X) - Eh| \geqslant t] \leqslant 2e^{(-dt^2)/(2cl^2)}$, then, the distribution of X is c-isoperimetric.
- for learning smooth functions ($\mathrm{Lip}(f) \leqslant l$), the number of parameters is $\Omega(nd\epsilon^2/l)$
- For imagenet, Bubeck and Sellke estimate needing $\mathcal{O}(10^{10} - 10^{11})$ parameters.

Thinking from first principles

- Kernel ridge regression: “Simply applying a Gaussian kernel on pixels and using no regularization achieves 46% test error.” [Zhang et al 2021]

Thinking from first principles

- Kernel ridge regression: “Simply applying a Gaussian kernel on pixels and using no regularization achieves 46% test error.” [Zhang et al 2021]
- “By preprocessing with a random convolutional neural net with 32,000 random filters, this test error drops to 17% error” [Zhang et al 2021]

Thinking from first principles

- Kernel ridge regression: “Simply applying a Gaussian kernel on pixels and using no regularization achieves 46% test error.” [Zhang et al 2021]
- “By preprocessing with a random convolutional neural net with 32,000 random filters, this test error drops to 17% error” [Zhang et al 2021]
- ℓ^2 regularization leads to better generalization. Why?

Regularization and generalization

- “the ℓ^2-norm of the minimum norm solution with no preprocessing is approximately 220. With wavelet preprocessing, the norm jumps to 390. Yet the test error drops by a factor of 2” [Zhang et al 2021]

Regularization and generalization

- "the ℓ^2-norm of the minimum norm solution with no preprocessing is approximately 220. With wavelet preprocessing, the norm jumps to 390. Yet the test error drops by a factor of 2" [Zhang et al 2021]
- "So while this minimum-norm intuition may provide some guidance to new algorithm design, it is only a very small piece of the generalization story"

Regularization and generalization

- “the ℓ^2-norm of the minimum norm solution with no preprocessing is approximately 220. With wavelet preprocessing, the norm jumps to 390. Yet the test error drops by a factor of 2” [Zhang et al 2021]
- “So while this minimum-norm intuition may provide some guidance to new algorithm design, it is only a very small piece of the generalization story”
- Rademacher complexity of linear class (on features) lower.

Developing computational thinking

- Theoretical and computational problems are always intertwined

Developing computational thinking

- Theoretical and computational problems are always intertwined
- Spending time (struggling) with problems productively

Developing computational thinking

- Theoretical and computational problems are always intertwined
- Spending time (struggling) with problems productively
- Crisis, reflection on concepts that lead to resolution

Developing computational thinking

- Theoretical and computational problems are always intertwined
- Spending time (struggling) with problems productively
- Crisis, reflection on concepts that lead to resolution
- Implementing algorithms improves understanding

Developing computational thinking

- Theoretical and computational problems are always intertwined
- Spending time (struggling) with problems productively
- Crisis, reflection on concepts that lead to resolution
- Implementing algorithms improves understanding
- Reading papers, books, but working on your own

Developing computational thinking

- Theoretical and computational problems are always intertwined
- Spending time (struggling) with problems productively
- Crisis, reflection on concepts that lead to resolution
- Implementing algorithms improves understanding
- Reading papers, books, but working on your own
- Question-first approach may be more efficient

Developing computational thinking

- Theoretical and computational problems are always intertwined
- Spending time (struggling) with problems productively
- Crisis, reflection on concepts that lead to resolution
- Implementing algorithms improves understanding
- Reading papers, books, but working on your own
- Question-first approach may be more efficient
- Pattern-matching is non-trivial

Developing computational thinking

- Theoretical and computational problems are always intertwined
- Spending time (struggling) with problems productively
- Crisis, reflection on concepts that lead to resolution
- Implementing algorithms improves understanding
- Reading papers, books, but working on your own
- Question-first approach may be more efficient
- Pattern-matching is non-trivial
- Repetition helps!

Having the right attitude toward grad school courses and research

- Computational science: intersection of math, data science, computer science, and domain science

Having the right attitude toward grad school courses and research

- Computational science: intersection of math, data science, computer science, and domain science
- Developing computational thinking takes time

Having the right attitude toward grad school courses and research

- Computational science: intersection of math, data science, computer science, and domain science
- Developing computational thinking takes time
- Difficulties compounded by various intersections!

Having the right attitude toward grad school courses and research

- Computational science: intersection of math, data science, computer science, and domain science
- Developing computational thinking takes time
- Difficulties compounded by various intersections!
- No need to panic or despair or feel inadequate!

Having the right attitude toward grad school courses and research

- Computational science: intersection of math, data science, computer science, and domain science
- Developing computational thinking takes time
- Difficulties compounded by various intersections!
- No need to panic or despair or feel inadequate!
- Flourishing? “Mathematics for human flourishing” - Francis Su