Designing Explicit Regularizers

Tengyu Ma

Stanford University

Occam's Razor:

"The simplest solution is mostly likely the right one"

Low-complexity models likely generalize well

Q: what are correct definitions of model complexity in deep learning?

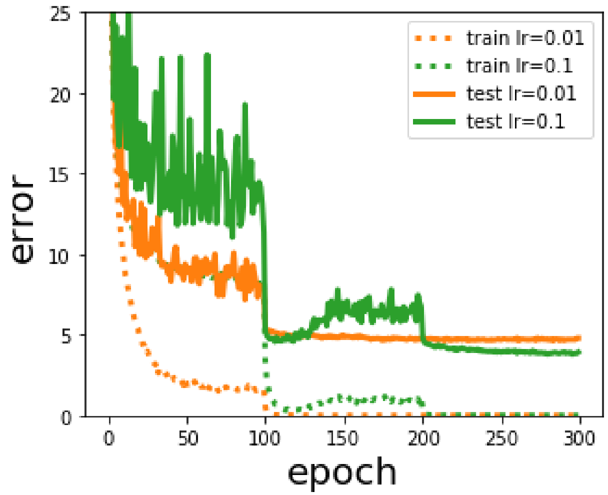
Implicit/Algorithmic Regularization

- Algorithms prefer low-complexity solutions
- Low-complexity solutions generalize well

⬇

1. Understand existing algorithms
2. Discover complexity measure

Faster training may lead to worse generalization

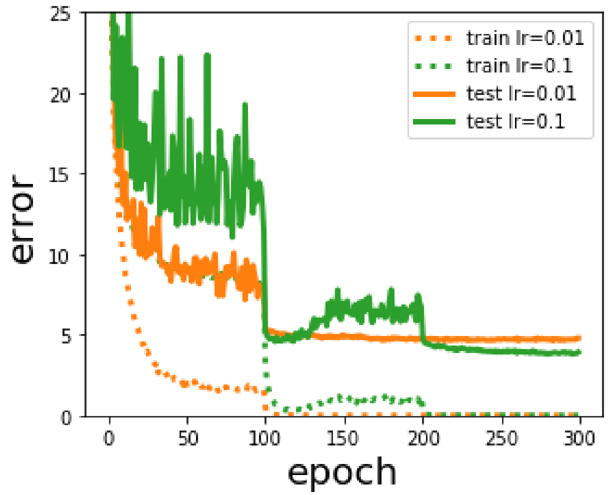


New phenomenon: algorithms can regularize!

The lack of understanding of the generalization hampers the study of optimization!

[Keskar et al’17, Hoffer et al’18]

Faster training may lead to worse generalization



Models Preferred by Large Learning Rate?

- Explainable for some toy cases with complicated analysis [Li-Wei-M.'19]
    1. Noisy gradients effectively restrict the complexity of a two-layer neural network to be a linear model
    2. (Some reason for why the initial learning rate matters)

Models Preferred by the Initializations

- Large initialization prefers staying minimum NTK norm solution [Chizat-Bach'19]
- Small initialization prefers the "rich" regime ([Woodworth et al.'19, Li-M.-Zhang'18], c.f. Nati's talk in the afternoon)

Is Understanding the Implicit/Algorithmic Regularization the Only Approach?

This talk: revisiting a classic approach --- explicit regularization, which I think also deserve some attention

Implicit/Algorithmic Regularization

- Algorithms prefer low-complexity solutions

- Here is most of the technical challenges

- Low-complexity solutions generalize well

1. Understand existing algorithms
2. Discover complexity measures

- All the analyzable DL algorithms can be replaced by a simpler and more explicit one
  - Explicit regularization
  - (Iterative) kernels

Implicit/Algorithmic Regularization

- Algorithms prefer low-complexity solutions

- Low-complexity solutions generalize well

1. Understand existing algorithms
2. Discover complexity measures
3. Simplify existing algorithms

- Here is most of the technical challenges
- All the analyzable DL algorithms can be replaced by a simpler and more explicit one
  - Explicit regularization
  - (Iterative) kernels

Implicit/Algorithmic Regularization

➢ Algorithms prefer low-complexity solutions

➢ Low-complexity solutions generalize well

1. Understand existing algorithms
2. Discover complexity measures
3. Simplify existing algorithms

➢ (This may also lead to new complexity and algorithms)

Explicit Regularization

➢ Regularize the complexity; hope success of optimization

➢ Find complexity measure that leads to a better generalization

~~1. Understand existing algorithms~~
2. Design complexity/regularizers
3. Design new algorithms
4. Separate opt. & statistics

➢ No double descent phenomenon

➢ Other counterexamples [Nagarajan et al'19] are gone

➢ Replace the implicit regularization?

Complexity via Data-dependent Generalization Bounds

$$\forall f, \text{test error}(f) - \text{training error}(f) \leq \sqrt{\frac{\text{complexity}(f, \text{training data})}{n}}$$

- Misha's talk: there is a fundamental limitation if the complexity only depends on the hypothesis class and the data distribution

Related works:

- [Golowich et al, Bartlett et al'17, Neyshabur et al.'17]: complexity depends on the product of norms of the weights and the output margin
- [Arora et al.'18]: compression-based bounds
- [Dziugaite-Roy'18a,b, Nagarajan-Kolter'19]: PAC-Bayes based data-dependent bounds

A Simple Bound Based on "All-layer Margin"

Theorem (informal): W.h.p over the randomness of the n data $(x_1, y_1), \ldots (x_n, y_n)$,

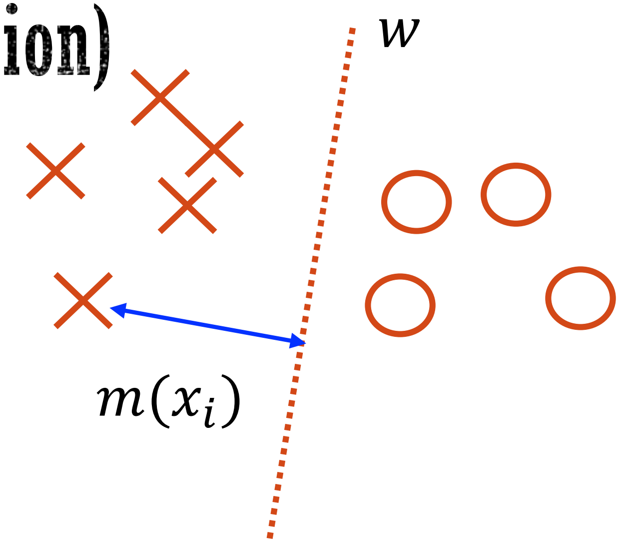
$$\text{Generalization} \lesssim \frac{1}{n}\sqrt{\sum_{i=1}^{n} \frac{1}{m\,(x_i)^2}} \cdot \text{norm of weights}$$

where $m(\cdot)$ is the "all-layer margin" (defined in next slide.)

- For linear models, $m(\cdot)$ is the standard output margin; the theorem was essentially proved in [Srebro-Sridharan-Tewari'2010]

Improved Sample Complexities for Deep Networks and Robust Classification via an All-Layer Margin [Wei-M.'19]

All-Layer Margin (Binary Classification)



- Recall for linear models

$$\text{margin} = \frac{y w^\top x}{||w||}$$
$$= \min \delta$$
$$\text{s.t. } y w^\top (x + \delta) \le 0$$

- For non-linear models

$$\text{normalized output margin} = \frac{f(x; W)}{?}$$

- All-layer margin:

$$m(x) = \min \text{ perturbation } \delta_1, \dots, \delta_r \text{ of the layers}$$
s.t. output is flipped to be incorrect after the perturbation

(or $y f(x; W, \delta) \le 0$)

A Simple Proof

$$\text{Generalization} \lesssim \frac{1}{n}\sqrt{\sum_{i=1}^{n}\frac{1}{m\,(x_i)^2}} \cdot \text{norm of weights}$$

- $m(\cdot)$ "correlates" with 0-1 loss in the sense that $m(x) = 0$ if x is misclassified
  - $\Rightarrow$ With standard tools, it suffices to bound the complexity of $m(\cdot)$ [Srebro-Sridharan-Tewari'2010]
- $m(x)$ is 1-Lipschitz in the parameters (w.r.t the spectral norm)
  - $m(\cdot)$ reshapes the heterogenous geometry into a homogenous one
  - => $m(x)$ has low complexity (by standard tools)

All-Layer Margin <--> Lipschitzness and Output Margin

- $m(x)$ measures the robustness of the output w.r.t to intermediate layer perturbation
- Small Lipschitzness + big output margin => big all-layer margin

Corollary: generalization $\lesssim \frac{1}{n}\sqrt{\sum_{i=1}^{n} \frac{\text{Lipschitzness}^2}{\text{output margin}^2}} \cdot$ norm of weights

- Lipschitzness is a "non-parametric" notion, and it's evaluated at the training data points
- Reminiscent of the noise stability in [Arora et al'2018]

Generalization of Adversarially Robust Loss

- Background: robust loss has severe generalization issues for CIFAR [Mardy et al.'18]

$$\text{Robust test} - \text{robust train} \lesssim \frac{1}{n}\sqrt{\sum_{i=1}^{n} \frac{1}{\min_{||\delta_i|| \le \epsilon} m(x_i+\delta_i)}} \cdot \text{norm of weights}$$

- Prior works bounds the generalization of the relaxation of the robust loss [Khim and Loh, 2018, Yin et al., 2018]

- Robust VC dimension can be infinity in the worst case [Montasser-Hanneke-Srebro'19]

Maximizing the All-Layer Margin

- Max-min problem, because margin definition involves minimum
- Alternating minimization (similar to robust optimization)

Dataset	Arch.	Setting	Standard SGD	AMO
CIFAR-10	WRN16-10	Baseline	4.15%	**3.42%**
		No data augmentation	9.59%	**6.74%**
		20% random labels	9.43%	**6.72%**
	WRN28-10	Baseline	3.82%	**3.00%**
		No data augmentation	8.28%	**6.47%**
		20% random labels	8.17%	**6.01%**
CIFAR-100	WRN16-10	Baseline	20.12%	**19.14%**
		No data augmentation	31.94%	**26.09%**
	WRN28-10	Baseline	18.85%	**17.78%**
		No data augmentation	30.04%	**24.67%**

Adeversarially Robust Errors (Preliminary)

- ℓ_∞ attack on CIFAR

Arch.	Standard	Robust AMO
WideResNet16-10	50.12%	**44.68%**
WideResNet28-10	49.16%	**42.24%**

Applications of Data-dependent Regularizers: Learning Imbalanced Datasets

- Real-world datasets have imbalanced class distribution
- Minority classes have worse generalization
- Our approach:
  - Derive a generalization bound
  - Regularize the RHS of the bound
- => Regularize the complexity on the minority classes more strongly

$$\sum_{x \in \text{minority}} \text{complexity}(f, x)$$

Loss	Schedule	Top-1	Top-5
ERM	SGD	42.86	21.31
CB Focal [8]	SGD	38.88	18.97
ERM	DRW	36.27	16.55
LDAM	SGD	35.42	16.48
LDAM	DRW	**32.00**	**14.82**

ours

Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss
[Cao-Wei-Gaidon-Arechiga-M., Neurips 2019]

Conclusions

Designing explicit regularizers:

generalization bounds -> complexity measure -> regularization

- Tighter bounds lead to better empirical results

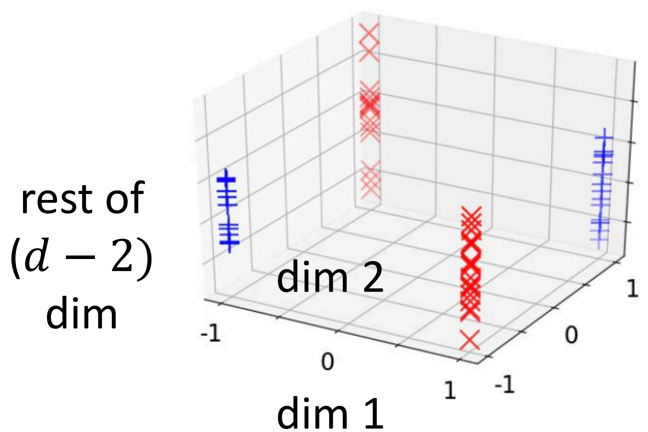
Open questions:

- Bounds for other data-dependent regularizations?
  - dropout
  - data augmentation (mixup, cutout, …)
  - new ones?
- Optimization for regularized loss?
- Heterogenous datasets?

Thank you!

ℓ_2-Regularization vs No Regularization

- Recall that no regularization + certain initialization ⟺ minimum norm solution of the neural tangent kernel (NTK)



Theorem [Wei-Lee-Liu-M.'18] :

- Using 2-layer neural net with cross entropy loss and ℓ_2 regularization, assuming optimization succeeds; sample complexity = $\tilde{O}(d)$
- With the NTK that corresponds to 2-layer neural nets, sample complexity $\gtrsim \Omega(d^2)$

- Gap empirically observed on both synthetic and real data
- Optimization can be provably solved in poly iteration if width $\rightarrow \infty$ [Wei-Lee-Liu-M.'18]

Setting	Normalization	Jacobian Reg	Test Error
Baseline	BatchNorm	×	4.43%
	BatchNorm	✓	**3.99**%
Low learning rate (0.01)	BatchNorm	×	5.98%
		✓	**5.46**%
No data augmentation	BatchNorm	×	10.44%
		✓	**8.25**%
No BatchNorm	None	×	6.65%
	LayerNorm	×	6.20%
		✓	**5.57**%