The Current Thinking at NIPS On Why Neural Networks Generalize

Tim van Erven

Universiteit Leiden

NIPS 2017 Debriefing

Leiden, February 28, 2018

This Year's Juicy Controversy

Ali Rahimi (test of time award)

Rahimi:

- Machine learning has become alchemy
- Alchemists discovered metallurgy, glass-making, and various medications; while machine learning researchers have managed to make machines that can beat human Go players, identify objects from pictures, and recognize human voices.
- However, alchemists believed they could cure diseases or transmute basic metals into golds, which was impossible.
- The Scientific Revolution had to dismantle 2000 years worth of alchemical theories.

Two Papers That Go Beyond Alchemy

- Wilson, Roelofs, Stern, Srebro, Recht. **The Marginal Value of Adaptive Gradient Methods in Machine Learning.** NIPS 2017.
- Bartlett, Foster, Telgarsky. **Spectrally-normalized margin bounds for neural networks.** NIPS 2017.

Generalization Questions

- High-dimensional setting: typically number of parameters is $d \geq 25n$
- So uniform convergence impossible. Need to do some kind of regularization/restrict the parameters.
- But even if you disable all standard regularization, it still works! [Zhang,Bengio,Hardt,Recht,Vinyals,ICLR 2017]
- So how are the parameters restricted?

Generalization Questions

- High-dimensional setting: typically number of parameters is $d \geq 25n$
- So uniform convergence impossible. Need to do some kind of regularization/restrict the parameters.
- But even if you disable all standard regularization, it still works! [Zhang,Bengio,Hardt,Recht,Vinyals,ICLR 2017]
- So how are the parameters restricted?

By the behavior of the optimization algorithm!

Paper 1

Wilson, Roelofs, Stern, Srebro, Recht. **The Marginal Value of Adaptive Gradient Methods in Machine Learning.** NIPS 2017.

- Prior work: early stopping of optimization algorithms acts as implicit regularization by restricting the complexity of the parameters that can be reached.
- This work: adaptive optimization methods often give better fit on train set, but worse generalization to test set, because they find different types of solutions.

Example: The Potential Perils of Adaptivity

Least Squares with $d \gg n$:

$$\text{minimize in } \boldsymbol{w} \qquad \frac{1}{2}\|X\boldsymbol{w} - y\|_2^2$$

for $X = \begin{pmatrix} \boldsymbol{x}_1^\intercal \\ \vdots \\ \boldsymbol{x}_n^\intercal \end{pmatrix}$ an $n \times d$ matrix, $\boldsymbol{w} \in \mathbb{R}^d$, $y \in \mathbb{R}^n$.

- For $d > n$, solution is not unique.
- Which solution does an optimization algorithm find?

Example: The Potential Perils of Adaptivity

Least Squares with $d \gg n$:

$$\text{minimize in } \boldsymbol{w} \qquad \frac{1}{2}\|X\boldsymbol{w} - y\|_2^2 \tag{1}$$

Non-adaptive methods:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t(\boldsymbol{x}_i^\intercal \boldsymbol{w} - y_i)\boldsymbol{x}_i = \boldsymbol{w}_t - c_t \boldsymbol{x}_i \qquad \text{(Stochastic GD)}$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \sum_{i=1}^{n} (\boldsymbol{x}_i^\intercal \boldsymbol{w} - y_i)\boldsymbol{x}_i = \boldsymbol{w}_t - \sum_{i=1}^{n} c_{t,i} \boldsymbol{x}_i \qquad \text{(GD)}$$

- If $\boldsymbol{w}_1$ is a linear combination of the feature vectors, then so is $\boldsymbol{w}_t$.
- Among such linear combinations, (1) has a unique minimum: the minimizer of (1) with smallest $\|\boldsymbol{w}\|_2$!

Example: The Potential Perils of Adaptivity

$$\text{minimize in } \boldsymbol{w} \qquad \frac{1}{2}\|X\boldsymbol{w} - y\|_2^2 \tag{2}$$

Adaptive methods (AdaGrad,RmsProp,Adam):

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - c_t H_t^{-1}\boldsymbol{x}_i + \beta_t H_t^{-1} H_t (w_t - w_{t-1})$$

$$H_t = \operatorname{diag}\Big(\sum_{s=1}^{t} \eta_s \boldsymbol{g}_s \boldsymbol{g}_s^\intercal\Big)^{1/2}$$

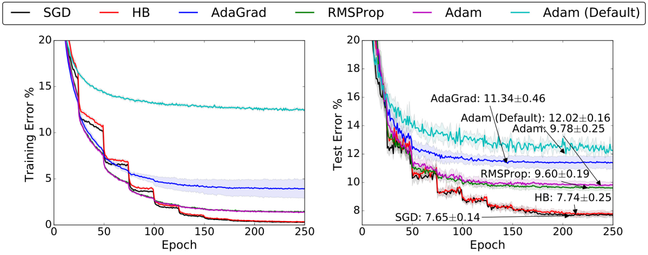
"Can construct a variety of instances where these methods converge to solutions with small $\|\boldsymbol{w}\|_\infty$ instead of $\|\boldsymbol{w}\|_2$, and this can overfit in high d."

Lemma

If there exists a c such that $X \operatorname{sign}(X^\intercal y) = cy$, then these methods converge to a unique $\boldsymbol{w} \propto \operatorname{sign}(X^\intercal y)$.

E.g. $\operatorname{sign}(X^\intercal y)$ looks like $(+1, -1, \ldots, +1, +1)^\intercal$.

Deep Learning Experiments



(a) CIFAR-10 (Train) **(b)** CIFAR-10 (Test)

- The adaptive methods generalize worse than non-adaptive methods, even when they achieve the same or smaller training error
- Adaptive methods often display faster initial progress on the training set, but their performance quickly plateaus on a separate 'development' data set
- Tuning is often said not to be necessary for Adam, but it makes a big difference

Remarks

- Paper 1 is really about AdaGrad, RmsProp, Adam, which are designed to favor small $\|\boldsymbol{w}\|_\infty$, so conclusions are about this behavior, not necessarily about adaptivity.

Remarks

- Paper 1 is really about AdaGrad, RmsProp, Adam, which are designed to favor small $\|\boldsymbol{w}\|_\infty$, so conclusions are about this behavior, not necessarily about adaptivity.
- If Adam usually generalizes significantly worse than SGD, then why is it becoming the standard choice?
  - Surely people would notice this...
  - Relatedly: Adam does not even always converge on simple linear one-dimensional tasks [Reddi,Kale,Kumar,ICLR 2018]

Paper 2

Bartlett, Foster, Telgarsky. **Spectrally-normalized margin bounds for neural networks.** NIPS 2017.

- Prior work: deep neural nets even fit **random labels** with 0 training error [Zhang,Bengio,Hardt,Recht,Vinyals,ICLR 2017]
- This work:
  - Generalization performance is not well explained (solely) by $\|\boldsymbol{w}\|_2$ of solution
  - This work: explain generalization by margin-normalized spectral complexity (theory matches empirical results)

- Consider neural networks for k classes:

$$F_A(\boldsymbol{x}) = \sigma_L(A_L\sigma_{L-1}(A_{L-1}\cdots\sigma_1(A_1\boldsymbol{x})\cdots)) \in \mathbb{R}^k.$$

- Classify by $\max_j F_A(\boldsymbol{x})_j$
- Margin measures gap with correct label $y \in \{1, \ldots, k\}$:

$$m_a(\boldsymbol{x}, y) := F_A(\boldsymbol{x})_y - \max_{j\neq y} F_A(\boldsymbol{x})_j$$

- Consider neural networks for k classes:

$$F_A(\boldsymbol{x}) = \sigma_L(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 \boldsymbol{x}) \cdots)) \in \mathbb{R}^k.$$

- Classify by $\max_j F_A(\boldsymbol{x})_j$
- Margin measures gap with correct label $y \in \{1, \ldots, k\}$:

$$m_a(\boldsymbol{x}, y) := F_A(\boldsymbol{x})_y - \max_{j \neq y} F_A(\boldsymbol{x})_j$$

- Spectral complexity (relative to $M_1, \ldots, M_L$):

$$R_A := \left(\prod_{i=1}^{L} \rho_i \|A_i\|_\sigma \right) \left(\sum_{i=1}^{L} \frac{\|A_i^\mathsf{T} - M_i^\mathsf{T}\|_{2,1}^{2/3}}{\|A_i\|_\sigma^{2/3}} \right)^{3/2}$$

- Consider neural networks for k classes:

$$F_A(\boldsymbol{x}) = \sigma_L(A_L \sigma_{L-1}(A_{L-1} \cdots \sigma_1(A_1 \boldsymbol{x}) \cdots)) \in \mathbb{R}^k.$$

- Classify by $\max_j F_A(\boldsymbol{x})_j$
- Margin measures gap with correct label $y \in \{1, \ldots, k\}$:

$$m_a(\boldsymbol{x}, y) := F_A(\boldsymbol{x})_y - \max_{j \neq y} F_A(\boldsymbol{x})_j$$

- Spectral complexity (relative to $M_1, \ldots, M_L$):

$$R_A := \left(\prod_{i=1}^{L} \rho_i \|A_i\|_\sigma \right) \left(\sum_{i=1}^{L} \frac{\|A_i^\mathsf{T} - M_i^\mathsf{T}\|_{2,1}^{2/3}}{\|A_i\|_\sigma^{2/3}} \right)^{3/2}$$

- Margin normalized spectral complexity of $(\boldsymbol{x}, y)$:

$$\frac{m_A(\boldsymbol{x}, y)}{R_A}$$

(assuming normalized inputs $\frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{x}_i\|_2^2 = 1$)

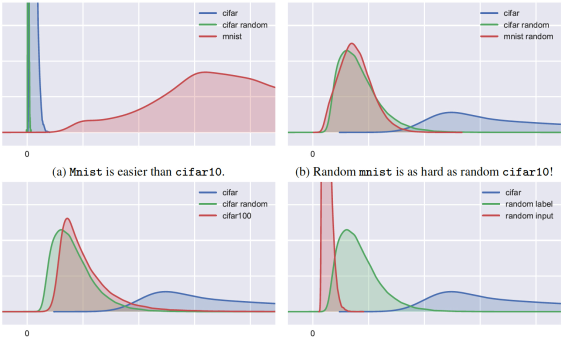
Results

Theory

Good margin + small spectral complexity implies small generalization error

Empirical Results:

Density of $\frac{m_A(\boldsymbol{x},y)}{R_A}$ seems to match with "hardness" of data sets (very hand-wavy):



(a) `Mnist` is easier than `cifar10`.

(b) Random `mnist` is as hard as random `cifar10`!

(c) `cifar100` is (almost) as hard as `cifar10` with random labels!

(d) Random inputs are harder than random labels.

Details of Theoretical Result

Theorem

For i.i.d. data, with probability at least $1-\delta$, for every margin $\gamma > 0$ and any network F_A:

$$\Pr(\arg\max_j F_A(X)_j \neq y) \leq \tilde{R}_\gamma(F_A) + \tilde{O}\left(\frac{\|X\|_2 R_A}{\gamma n}\ln(W) + \sqrt{\frac{\ln(1/\delta)}{n}}\right)$$

where $\tilde{R}_\gamma(f) \leq \frac{1}{n}\sum_i \mathbf{1}[f(\boldsymbol{x}_i)_{y_i} \leq \gamma + \max_{j \neq y_i} f(\boldsymbol{x}_i)_j]$ *and* $\|X\|_2 = \sqrt{\sum_i \|\boldsymbol{x}_i\|_2^2}$.