The Power of Stagewise Learning

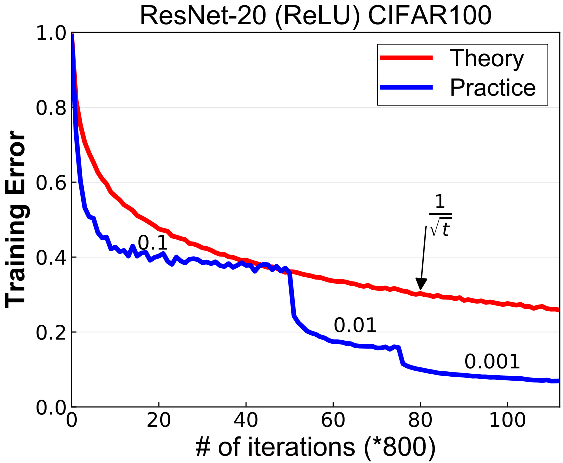
From Support Vector Machine to Generative Adversarial Nets

Tianbao Yang

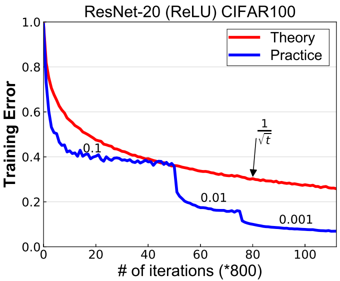
Department of Computer Science
The University of Iowa

Outline

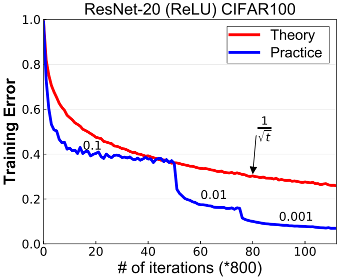
Gap between Practice vs Theory



Gap between Practice vs Theory

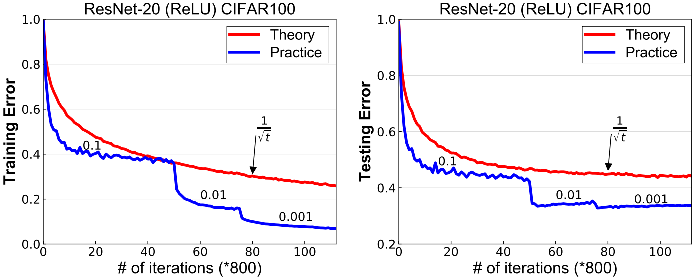


Gap between Practice vs Theory



Q1: Why does Stagewise Learning (SL) Converge Faster?

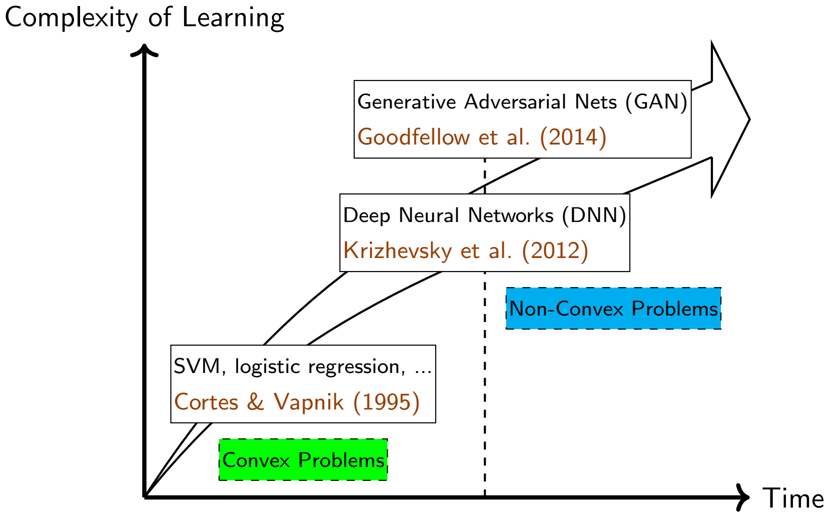
Gap between Practice vs Theory



Q1: Why does Stagewise Learning (SL) Converge Faster?

Q2: How to design better SL algorithms for DNN and Other problems?

The Evolution of Learning Methods



Big Data: Challenges and Opportunities

1.2 million of images

AlexNet for Image Classification (Krizhevsky et al., 2012)

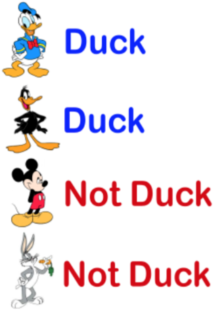
Google's JFT 300 millions of images

BigGAN for Image Generation (Brock et al., 2019)

Training on Huge datasets becomes a bottleneck!

Learning a Predictive Model

$\mathbf{x} \in \mathbb{R}^d$ y



- $(\mathbf{x}, y)$ is generated i.i.d.
- predictive model: $f(\mathbf{x}) \rightarrow y$

Risk Minimization

$$f^* = \arg\min_{f \in \mathcal{F}} R(f) := \mathrm{E}_{\mathbf{x},y}[\ell(f(\mathbf{x}), y)]$$

- $\mathcal{F}$ is a hypothesis class
- loss function $\ell(z, y)$: measures the prediction error

Risk Minimization

Risk of model f

$$f^* = \arg\min_{f\in\mathcal{F}} R(f) := \mathrm{E}_{\mathbf{x},y}[\ell(f(\mathbf{x}), y)]$$

- $\mathcal{F}$ is a hypothesis class
- loss function $\ell(z, y)$: measures the prediction error

Empirical Risk Minimization

Empirical Risk Minimization (Offline Learning)

- Collect $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$
- Find an approximate solution $\hat{f}$ to solve

$$f_n^* = \arg\min_{f \in \mathcal{F}} R_n(f) := \frac{1}{n}\sum_{i=1}^{n} \ell(f(\mathbf{x}_i), y_i)$$

Empirical Risk

Research Questions in Machine Learning

Iterative Algorithms:

$$f_{t+1} \leftarrow f_t + \mathcal{A}(\text{available information at iteration } t)$$

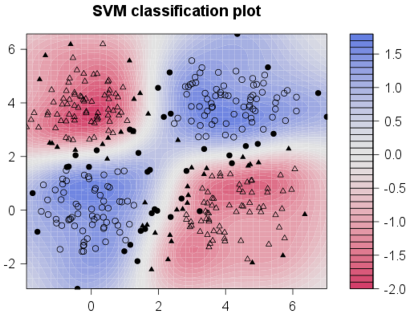
T: total time for training (e.g., # of iterations)

1. How fast is learning?
   Faster Training: Training Error

2. How accurate is the learned model?

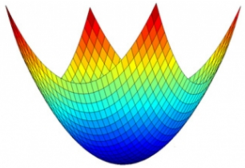
   Better Generalization: Testing Error

Shallow Model: Convex Methods

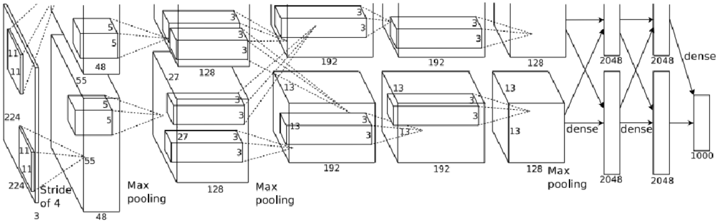


$$\mathbf{x} \rightarrow f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x})$$

$$\min_{\mathbf{w}\in\mathbb{R}^D} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}; \mathbf{x}_i, y_i)$$

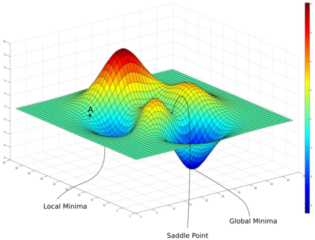


Deep Neural Networks: Non-Convex Methods



$$\mathbf{x} \rightarrow f_{\mathbf{w}}(\mathbf{x}) = w_L \circ \sigma(\cdots \sigma(w_3 \circ \sigma(w_2 \circ \sigma(w_1 \circ \mathbf{x}))))$$

$$\min_{\mathbf{w} \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}; \mathbf{x}_i, y_i)$$



Outline

Convex Methods

Consider

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) \triangleq \mathrm{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{P}}[\ell(\mathbf{w};\mathbf{x},y)]$$

- $F(\mathbf{w})$ is a convex function
- SVM, Logistic regression, Least-squares, LASSO, etc.

Goal: For a sufficiently small $\epsilon > 0$, find a solution $\widehat{\mathbf{w}}$ such that

$$F(\widehat{\mathbf{w}}) - \min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) \leq \epsilon$$

Stochastic Gradient Descent (Robbins & Monro, 1951)

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) \triangleq \mathrm{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{P}}[\ell(\mathbf{w};\mathbf{x},y)]$$

Stochastic Gradient Descent (SGD) Method

$$\text{Sample} \quad (\mathbf{x}_t, y_t) \sim \mathcal{P}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \partial \ell(\mathbf{w}_t, \mathbf{x}_t, y_t)$$

Stochastic Gradient Descent (Robbins & Monro, 1951)

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) \triangleq \mathrm{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{P}}[\ell(\mathbf{w};\mathbf{x},y)]$$

Stochastic Gradient Descent (SGD) Method

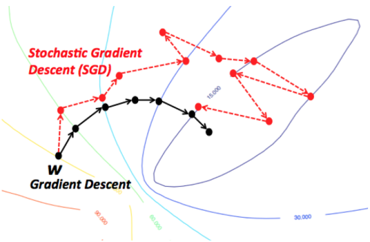
step size

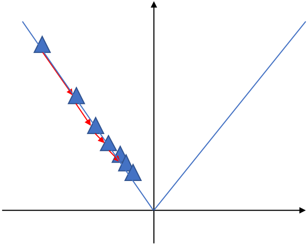
$$\text{Sample} \quad (\mathbf{x}_t, y_t) \sim \mathcal{P}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \partial\ell(\mathbf{w}_t, \mathbf{x}_t, y_t)$$

Slow Convergence of SGD

1. variance of stochastic gradient
2. decreasing step size: standard theory $\eta_t \propto 1/\sqrt{t}$

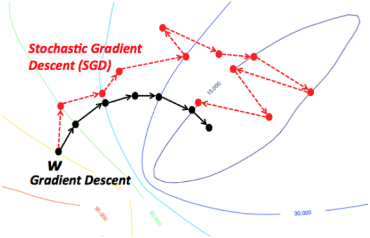


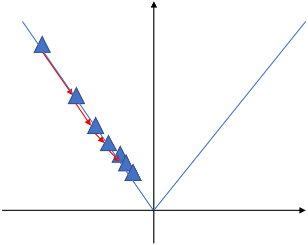


3. $O\left(\frac{1}{\epsilon^2}\right)$ iteration complexity (Nemirovski et al., 2009)

Slow Convergence of SGD

1. variance of stochastic gradient
2. decreasing step size: standard theory $\eta_t \propto 1/\sqrt{t}$





3. $O\left(\frac{1}{\epsilon^2}\right)$ iteration complexity (Nemirovski et al., 2009)

How to improve the Convergence speed?

Previous approaches

- Mini-batch SGD: sampling multiple samples each iteration
  - Pros: can have parallel speed-up
  - Cons: cannot not reduce total time complexity
- Making stronger assumptions
  - e.g. strong convexity, smoothness, using full gradients
  - Pros: speed-up for some family of problems
  - Cons: may not hold

Can we do better without imposing these strong assumptions? ICML 2017

Stagewise Stochastic Gradient (ICML 2017)

One-Stage SGD($\mathbf{w}_1, \eta, D, T$)

for $\tau = 1, \ldots, T$

$$\mathbf{w}_{\tau+1} = \text{Proj}_{\|\mathbf{w}-\mathbf{w}_1\|_2 \leq D}[\mathbf{w}_\tau - \eta\partial\ell(\mathbf{w}_\tau, \mathbf{x}_{i_\tau}, y_{i_\tau})]$$

Output: $\widehat{\mathbf{w}} = \sum_{\tau=1}^{T} \mathbf{w}_\tau / T$

Stagewise Stochastic Gradient (ICML 2017)

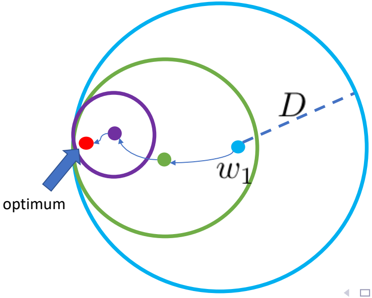
One-Stage SGD($\mathbf{w}_1, \eta, D, T$)

projection onto a ball

for $\tau = 1, \ldots, T$

$$\mathbf{w}_{\tau+1} = \mathrm{Proj}_{\|\mathbf{w}-\mathbf{w}_1\|_2 \le D}[\mathbf{w}_\tau - \eta\partial\ell(\mathbf{w}_\tau, \mathbf{x}_{i_\tau}, y_{i_\tau})]$$

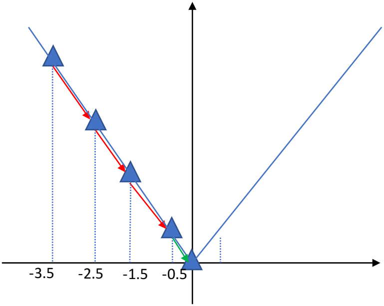
Output: $\widehat{\mathbf{w}} = \sum_{\tau=1}^{T} \mathbf{w}_\tau / T$



Stagewise Stochastic Gradient (ICML 2017)

SSGD

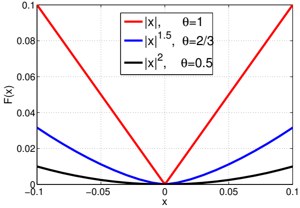
Set η_1, D_1 and T
for $k = 0, \ldots, K-1$ **do**
  $\mathbf{w}_{k+1} = \text{One-Stage SGD}(\mathbf{w}_k, \eta_k, D_k, T)$
  Set $\eta_{k+1} = \eta_k/2$, $D_{k+1} = D_k/2$
end for
Output: $\mathbf{w}_K$



Theoretical Result: Faster Convergence

Growth/Sharpness Condition: $\exists c < \infty$ and $\theta \in (0, 1]$ such that:

$$\|\mathbf{w} - \mathbf{w}_*\|_2 \leq c(F(\mathbf{w}) - F(\mathbf{w}_*))^{\theta},$$

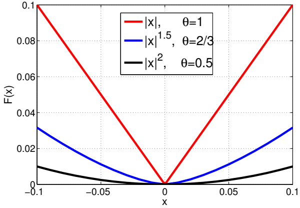


$$O\left(\frac{1}{\epsilon^{2(1-\theta)}} \log\left(\frac{1}{\epsilon}\right)\right) \quad \text{vs.} \quad O\left(\frac{1}{\epsilon^2}\right)$$

Theoretical Result: Faster Convergence

Growth/Sharpness Condition: $\exists c < \infty$ and $\theta \in (0, 1]$ such that:

$$\|\mathbf{w} - \mathbf{w}_*\|_2 \leq c(F(\mathbf{w}) - F(\mathbf{w}_*))^{\theta},$$



$O\left(\frac{1}{\epsilon^{2(1-\theta)}} \log\left(\frac{1}{\epsilon}\right)\right)$ vs. $O(\frac{1}{\epsilon^2})$

SSGD SGD

Machine Learning Problems satisfy GC

- SVM for high-dimensional data: $\theta = 1$

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i) + \lambda \|\mathbf{w}\|_1$$

$O(\log(1/\epsilon))$ vs $O(1/\epsilon^2)$

- LASSO: $\theta = 1/2$

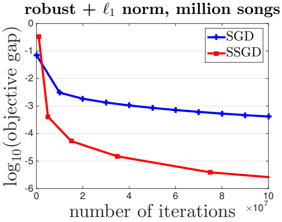
$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_1$$

$\widetilde{O}(1/\epsilon)$ vs $O(1/\epsilon^2)$

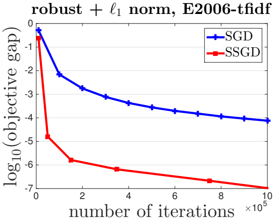
- many many more

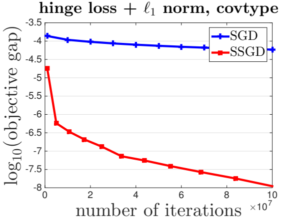
Empirical Results: SSGD vs SGD

million songs: $n = 463,715$

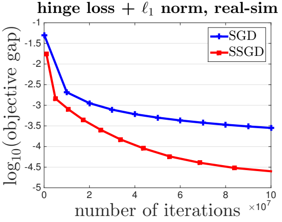


E2006-tfidf: $n = 16,087$



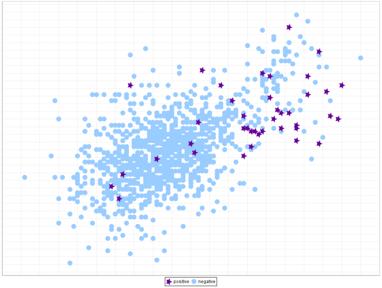


covtype: $n = 581,012$



real-sim: $n = 72,309$

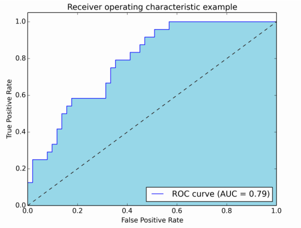
From Balanced Data to Imbalanced Data



Minimizing Error Rate is not a Good idea!

Optimization of Suitable Measures for Imbalanced Data

- maximize AUC (area under ROC curve):



$$\text{AUC} = \text{Prob.}(\text{score of} + > \text{score of} -)$$

- maximize F-measure

$$F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- etc

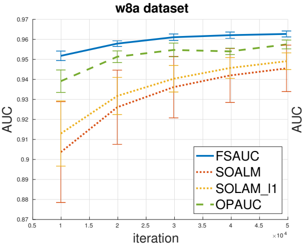
Non-Decomposable over individual examples

Our Contributions

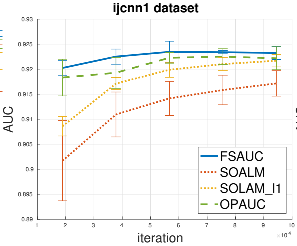
Our Approaches: low memory, low computation, low iteration complexity.

1. Fast Stochastic/Online AUC Maximization (ICML 2018)
   - based on a zero-sum convex-concave game formulation
   - stagewise learning for solving a convex-concave game
   - improves complexity from $O(1/\epsilon^2)$ to $O(1/\epsilon)$
2. Fast Stochastic/Online F-measure Maximization (NeurIPS 2018)
   - decomposes into two tasks
   - learning a posterior probability and learning a threshold
   - stagewise learning for optimizing the threshold faster
   - improves complexity from $O(1/\epsilon^2)$ to $O(1/\epsilon)$

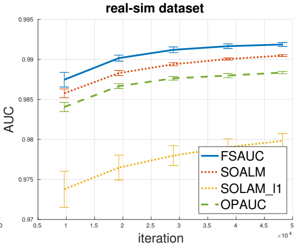
Experiments: Stochastic AUC Maximization



$p = 2.97\%$,

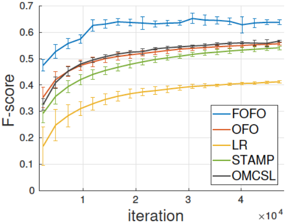


$p = 9.49\%$,

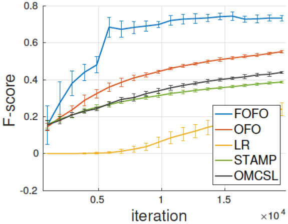


$p = 30.68\%$

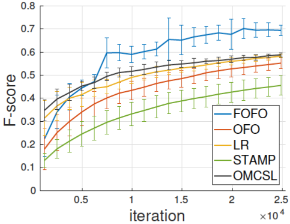
Experiments: Stochastic F-measure Optimization



(d) ijcnn1 (p=9.49%)



(e) Sensorless (1 vs o) (p=9.09%)

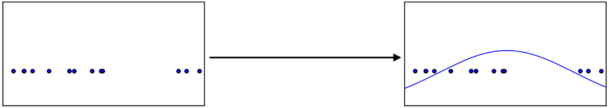


(f) w8a (p=2.97%)

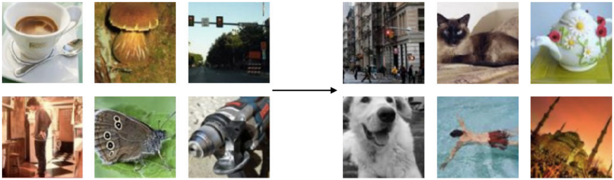
Outline

Generative Adversarial Nets (GAN)

- Generative Modeling (Density Estimation)



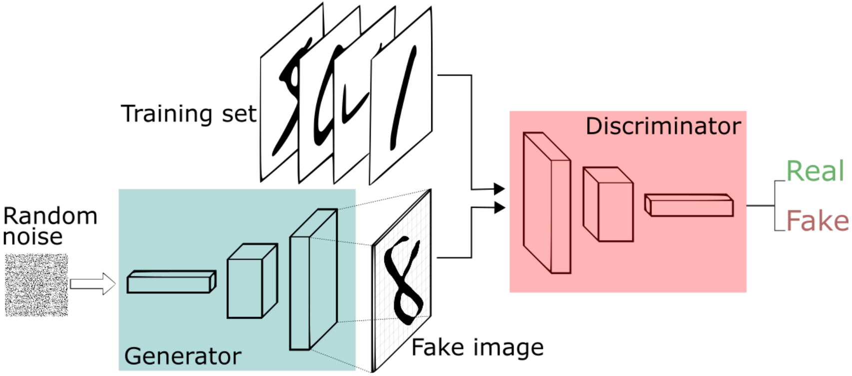
- Sample Generation



Training examples   Model samples

slides courtesy of Ian Goodfellow NIPS 2016 tutorial

Generative Adversarial Nets



Formulation of Generative Adversarial Nets

Zero-Sum Game Formulation (Goodfellow et al. (2014))

$$\min_{G}\max_{D} \mathrm{E}_{\mathbf{x}\sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathrm{E}_{\mathbf{z}\sim p_{\text{random}}}[\log(1 - D(G(\mathbf{z})))]$$

- $\mathbf{z}$ random noise, $\mathbf{x}$ real data
- G generator: $G(\mathbf{z})$ fake image
- D discriminator: $D(\mathbf{x})$ (e.g., probability of being real image)
- Ideally: at Nash Equilibrium: $p(G(\mathbf{z})) = p(\mathbf{x})$

Formulation of Generative Adversarial Nets

prob. of being real

Zero-Sum Game Formulation (Goodfellow et al. (2014))

$$\min_G \max_D \mathrm{E}_{\mathbf{x}\sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathrm{E}_{\mathbf{z}\sim p_{\text{random}}}[\log(1 - D(G(\mathbf{z})))]$$

- $\mathbf{z}$ random noise, $\mathbf{x}$ real data
- G generator: $G(\mathbf{z})$ fake image
- D discriminator: $D(\mathbf{x})$ (e.g., probability of being real image)
- Ideally: at Nash Equilibrium: $p(G(\mathbf{z})) = p(\mathbf{x})$

Non-Convex Non-Concave Games

$$\min_{\mathbf{w}} \max_{\mathbf{u}} F(\mathbf{w}, \mathbf{u}) := \mathrm{E}_{\mathbf{x} \sim p_{\text{data}}, \mathbf{z} \sim p_{\text{random}}}[\ell(\mathbf{w}, \mathbf{u}; \mathbf{x}, \mathbf{z})]$$

- Non-convex w.r.t **w**, non-concave w.r.t **u**
- Finding Nash-Equilibrium is NP-hard
- Existing studies mostly heuristics learned from convex-concave games
- No Convergence Guarantee (could be divergent)

First Convergence Theory for Finding Nearly Stationary Points:

Stationary Point: $\nabla F(\mathbf{w}^*, \mathbf{u}^*) = 0$

(Lin-Liu-Rafique-Y., arXiv 2018, presented at NeurIPS 2018 SGO&ML)

A Stagewise Learning Algorithm for GAN

for $k = 0, \ldots, K-1$ **do**

$$F_k(\mathbf{w}, \mathbf{u}) = F(\mathbf{w}, \mathbf{u}) + \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_k\|^2 - \frac{\lambda}{2}\|\mathbf{u} - \mathbf{u}_k\|^2$$

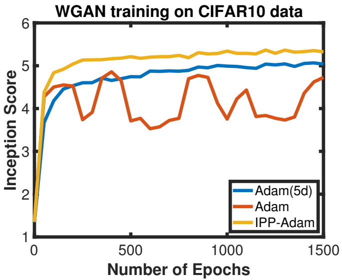
$$(\mathbf{w}_{k+1}, \mathbf{u}_{k+1}) = \mathcal{A}(F_k, \mathbf{w}_k, \mathbf{u}_k, \eta_k, T_k)$$

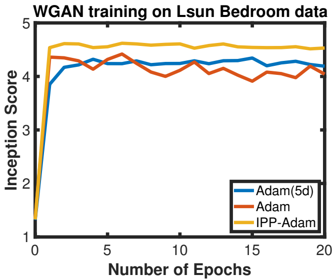
end for

$$(\mathbf{w}^*, \mathbf{u}^*) = \arg\min_{\mathbf{w}} \max_{\mathbf{u}} F(\mathbf{w}, \mathbf{u}) + \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}^*\|^2 - \frac{\lambda}{2}\|\mathbf{u} - \mathbf{u}^*\|^2$$

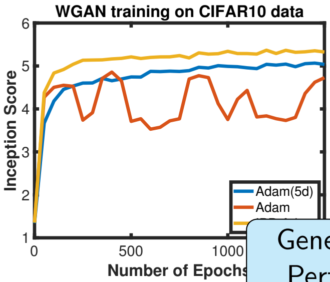
1. $\lambda > 0$ is an algorithmic regularization parameter
2. Different $\mathcal{A}$ (e.g., Primal-Dual SGD, Adam)
3. Use variational inequality for analysis

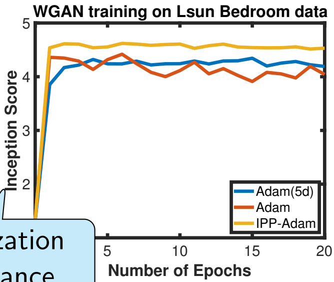
Experiments: Image Generation





Experiments: Image Generation





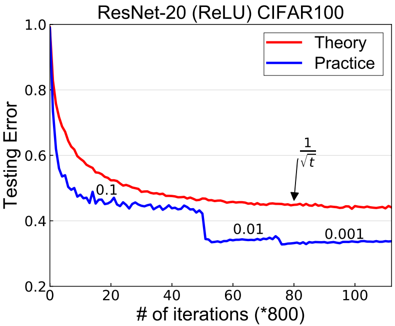
Generalization Performance

Why Does Stagewise Learning Improves Testing Error?

Optimizing Deep Neural Networks

Non-Convex

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) \triangleq \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}; \mathbf{x}_i, y_i)$$



A Stagewise Learning Algorithm for DNN

for $k = 0, \ldots, K-1$ **do**

$$F_k(\mathbf{w}) = F(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_k\|^2$$
$$\mathbf{w}_{k+1} = \mathrm{SGD}(F_k, \mathbf{w}_k, \eta_k, T_k)$$
$$\eta_{k+1} = \eta_k / 2$$

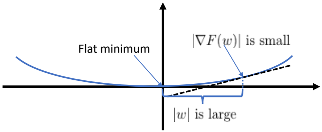
end for

1. $\lambda \geq 0$: algorithmic regularization
2. step size decreases geometrically in stages
3. is a more general framework
4. convergence to a stationary point established in ICLR 2019

Why Does Stagewise Learning Improves Testing Error?

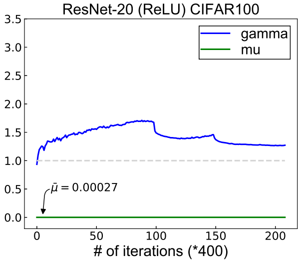
1. Explore Growth Condition!

$$\mu\|\mathbf{w} - \mathbf{w}_*\|_2^2 \leq F(\mathbf{w}) - F(\mathbf{w}_*),$$



2. Explore Almost-Convexity Condition

$$\frac{-\nabla F(\mathbf{w})^\top(\mathbf{w}_* - \mathbf{w})}{F(\mathbf{w}) - F(\mathbf{w}_*)} \geq \gamma > 0,$$



3. Use stability for generalization analysis

Testing Error = Training Error + Generalization Error

Why Does Stagewise Learning Improves Testing Error?

- Faster Convergence of Training Error: $O(\frac{1}{\mu\epsilon})$ vs $O(\frac{1}{\mu^2\epsilon})$
- With the same number of iterations $T = \sqrt{\frac{n}{\mu}}$
- Testing Error Comparison

$$O\left(\frac{1}{\sqrt{n}\mu^{1/2}}\right) \quad \text{vs.} \quad O\left(\frac{1}{\sqrt{n}\mu^{3/2}}\right)$$

First Theory for Explaining Stagewise Learning for DNN

(Y.-Yan-Yuan-Jin, arXiv 2018)

Why Does Stagewise Learning Improves Testing Error?

- Faster Convergence of Training Error: $O(\frac{1}{\mu\epsilon})$ vs $O(\frac{1}{\mu^2\epsilon})$
- With the same number of iterations $T = \sqrt{\frac{n}{\mu}}$
- Testing Error Comparison

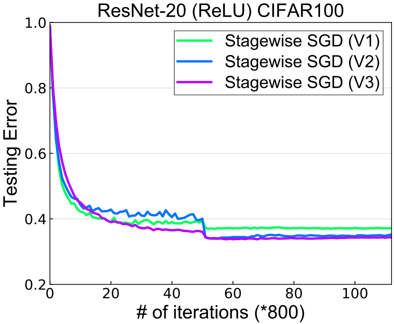
$$O\left(\frac{1}{\sqrt{n}\mu^{1/2}}\right) \quad \text{vs.} \quad O\left(\frac{1}{\sqrt{n}\mu^{3/2}}\right)$$

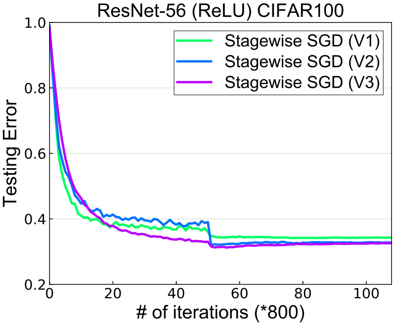
Stagewise Step Size — Decreasing Step Size

First Theory for Explaining Stagewise Learning for DNN

(Y.-Yan-Yuan-Jin, arXiv 2018)

Better Stagewise Learning Algorithms





- V1: standard, no alg. regularization, restart at last solution
- V2: algorithmic regularization, restart at last solution
- V3: algorithmic regularization, restart at averaged solution

Conclusions

Stagewise Learning is Powerful

- Theory: faster convergence for both convex and non-convex methods
- Practice: SVM, AUC, F-measure, DNN, GAN
- Open problems: e.g., Generalization of SL for GAN

Acknowledgements

- Students: Yi Xu, Mingrui Liu, Xiaoxuan Zhang, Zhuoning Yuan, Yan Yan, Hassan Rafique
- Collaborators: Qihang Lin (UIowa), Rong Jin (Alibaba Group)
- Funding Agency: NSF (CRII, Big Data)

Thank You!

Questions?

References I

Brock, Andrew, Donahue, Jeff, and Simonyan, Karen. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

Cortes, Corinna and Vapnik, Vladimir. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995. ISSN 0885-6125.

Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 2672–2680, 2014.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1106–1114, 2012.

References II

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Robbins, Herbert and Monro, Sutton. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22:400–407, 1951.