KAUST 2019

Generalization Error and Implicit Bias of Gradient Methods for Deep Learning

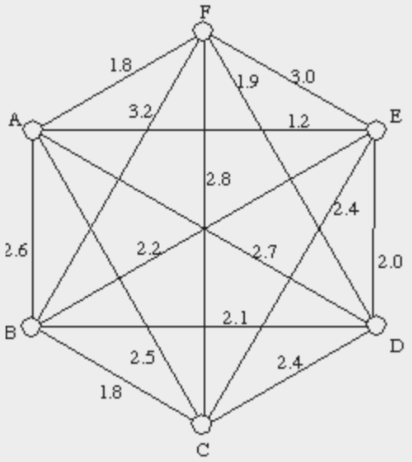
Jian Li

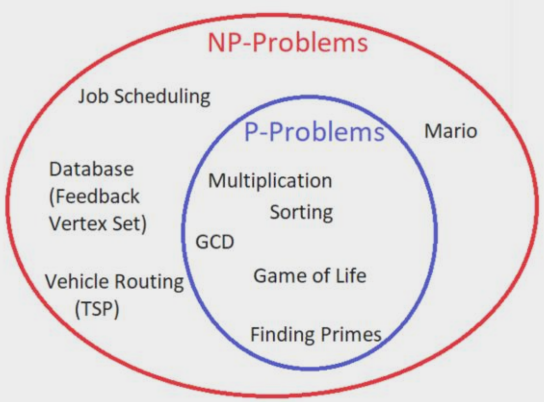
Institute of Interdisciplinary Information Science
Tsinghua University

Joint work with Xuyuan Luo, Mingda Qiao and Kaifeng Lv

Research interests

- **Theoretical Computer Science**
  - Algorithm design
  - Computational complexity



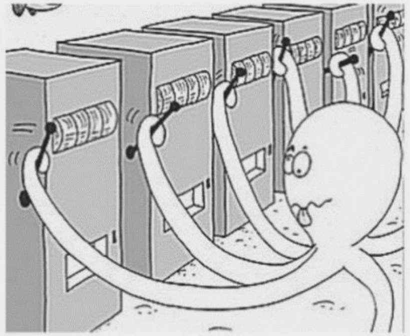


- **Databases**
  - Uncertain data management
  - Crowdsourcing

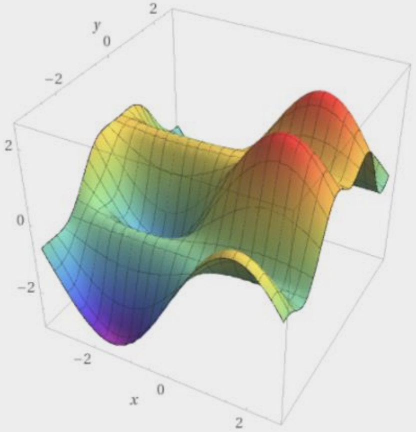
Key	Product ID	Price ($)	Prob.
a_1	a	120	0.7
a_2	a	80	0.3
b_1	b	110	0.6
b_2	b	90	0.4
c_1	c	140	0.5
c_2	c	110	0.3
c_3	c	100	0.2
d_1	d	10	1

Machine Learning

- Online learning, Bandits
- Optimization
- Learning Theory (esp. for deep learning)



source: Microsoft Research



- Applications in spatial-temporal data prediction, financial data analysis

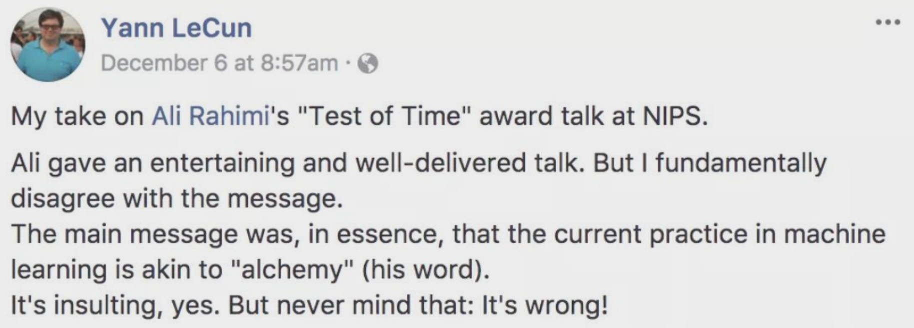
Why Deep Neural Networks Work So Well?

- Tremendous success in practice
- Theory, several exciting recent results (still not so satisfying)

Ali Rahimi, winner of the Test-of-Time award at a recent NIPS conference:

"Machine learning has become alchemy."

The Rahimi – LeCun debate:


Yann LeCun
December 6 at 8:57am ·

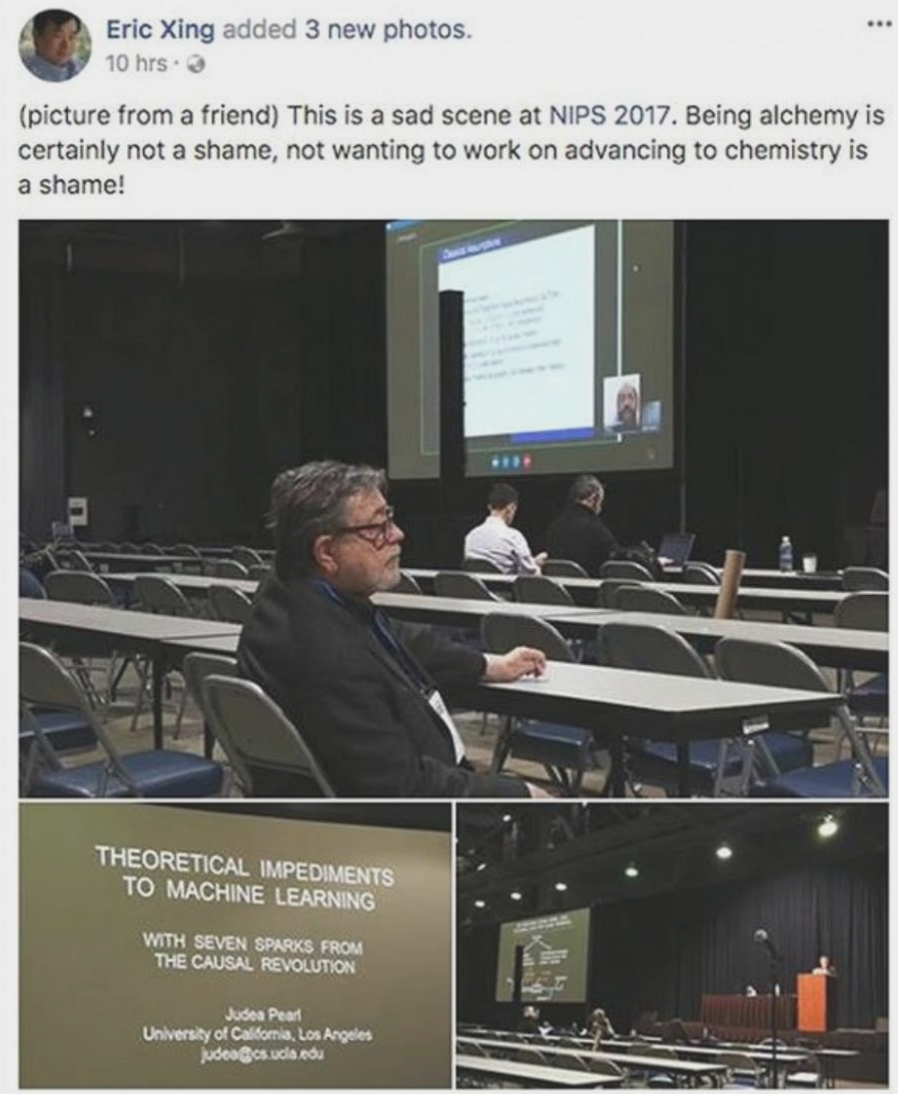
My take on Ali Rahimi's "Test of Time" award talk at NIPS.

Ali gave an entertaining and well-delivered talk. But I fundamentally disagree with the message.

The main message was, in essence, that the current practice in machine learning is akin to "alchemy" (his word).

It's insulting, yes. But never mind that: It's wrong!

Theory of Deep Learning



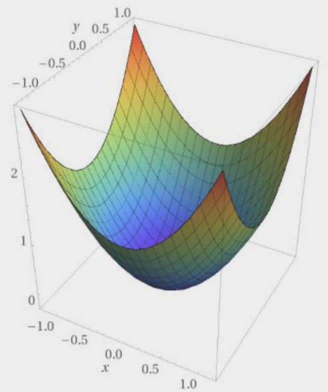
Judea Pearl, 2011 Turing award winner

- Develop theory of nonconvex learning and deep learning
  - Understand what happens in the blackbox
- Use theory to develop better algorithms
- Motivate important theoretical/mathematical questions

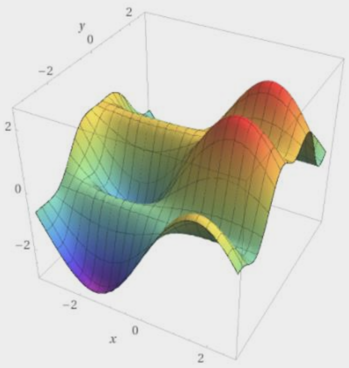
The mathematics of machine learning and deep learning – Sanjeev Arora – ICM2018

Why Deep Neural Networks Work So Well?

- Convex Learning （linear, logistic, SVM etc.）
  - Convex objectives
  - Optimization (optimal rate, well studied)
  - Generalization （PAC, VC-dimension, Rademacher Complexity, Margin bounds）
  - $\mathrm{err}_{\mathrm{gen}} \approx O(\sqrt{\mathrm{complexity}/n})$
  - Traditional complexity measure $\geq \#parameters >> n$



- Nonconvex
  - **Deep Learning**, topic modeling, matrix/tensor completion
  - Optimization
  - Traditional learning theory does not suffices



Understanding deep learning requires rethinking generalization， ICLR 17 best paper

Why Deep Neural Networks Work So Well?

Mysteries:

- Over-parametrized (traditional theories do not work directly)
- Highly Nonconvex, many local/global minima
- Commonly believed that the training algorithms (gradient-based algorithms) play important roles (not just the network architectures)
  - Algorithm-dependent generalization
  - Implicit bias (towards local/global min with interesting properties)
- Inductive bias
  - Why CNN works so well for image data?
- Many useful tricks
  - Dropout, batchnorm, layernorm, initialization

Outline

- Generalization
  - SGD,SGLD
  - Bayes-Stability
  - Extensions
- Implicit Bias
  - Smoothed Normalized Margin
  - Main Results
  - Robustness

Generalization error

- Measure how well a hypothesis obtained from the training data can generalize to a new test data point
- A central concept in machine learning
- Well studied in convex setting [uniform convergence, ERM, huge literature]

Formal definition:

$$\mathrm{err}_{\mathrm{gen}} = \mathrm{E}_S \mathrm{E}_A [\, \underline{f(A(S))} - \underline{f(A(S), S)} \,]$$

Training data set

$S = (z_1, z_2, \ldots, z_n)$

Learning algorithm

Population loss:

$f(w) = \mathrm{E}_z[f(w, z)]$

This is what we truly want to minimize

Training loss:

$f(w, S) = \frac{1}{n}\sum_{i=1}^{n} f(w, z_i)$

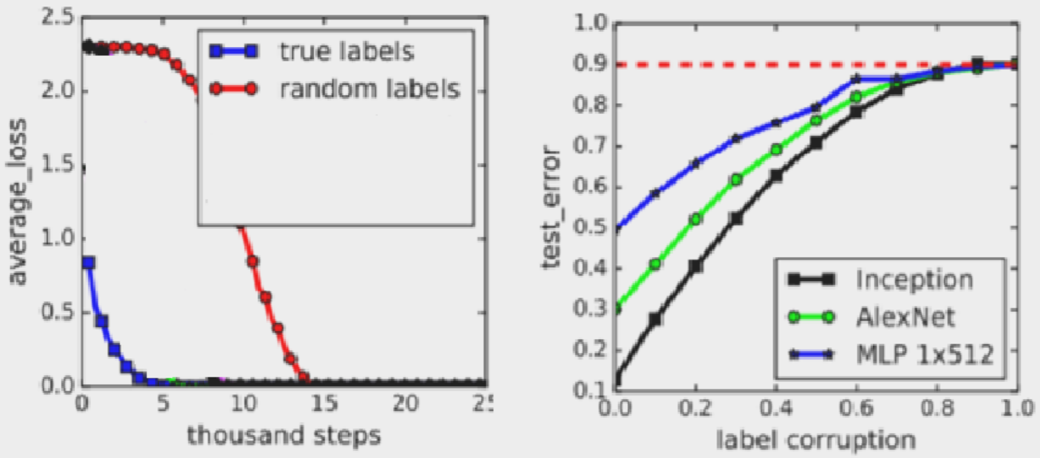
This is what we can optimize in practice, using training data

Generalization error

- Classical learning theory
  - VC-dimension, Rademacher Complexity, etc
$$\mathbf{err}_{\mathbf{gen}} = O(\sqrt{\mathbf{complexity}/n})$$
    - Only depends on the complexity of the hypothesis class
  - Traditional complexity measure > #parameters >> n
  - We need data dependent bound: Otherwise, we can't explain the random label experiment [Zhang et al.] (next page)

Understanding deep learning requires rethinking generalization [Zhang et al. 16]

Random label experiments: **choose a random label for each image**



(a) learning curves

(c) generalization error growth

Previous Argument:

Random-labeled instances requires more time to train, hence worse generalization

Training faster, generalize better [Hardt et al. 15][Mou et al. 18]

(generalization bound only depends on T)

What data characteristics makes random labeled data different from normal data?

Several other perspectives (e.g., [Bartlett et al. 17]…….[Arora et al. 19][Oymak et al. 19])

Related Work

Generalization error in nonconvex settings/Deep learning

- Random label experiment [Zhang et al. 16]
- Flat/Sharp local min [Kerskar et al. 16] [Dinh et al. 17]
- Norm/Margin based [Neyshabur et al. 17][Bartlett et al. 17][Wei et al. 18]
- Rademacher complexity [Kawaguchi et al. 17]
- PAC Bayesian [Neyshabur et al. 17, London 17, Mou et al. 18]
- Compression based [Brutzkus et al. 17][Arora et al. 18]
- Information Bottlenek [Shwartz-Ziv and Tishby 17]
- Algorithmic stability: Training faster, generalize better [Hardt et al. 15][Mou et al. 18][Pensia et al. 18]
- Overparametrization [Brutzkus et al. 17][Li et al. 18] [Du et al. 18] [Allen-Zhu et al. 18][Alon et al. 18] [Arora et al. 19]
- ……

Outline

- Generalization
  - SGD,SGLD
  - Bayes-Stability
  - Extensions
- Implicit Bias
  - Smoothed Normalized Margin
  - Main Results
  - Robustness

SGD and SGLD

GD/SGD

(full or stochastic) gradient

$$W_t \leftarrow W_{t-1} - \gamma_t g_t(W_{t-1})$$

The most popular algorithm for nonconvex objectives.
May be difficult to analyze due to the noise structure.

SGLD (Stochastic Gradient Langevin Dynamics)

$$W_t \leftarrow W_{t-1} - \gamma_t g_t(W_{t-1}) + \frac{\sigma_t}{\sqrt{2}}\mathcal{N}(0, I_d)$$

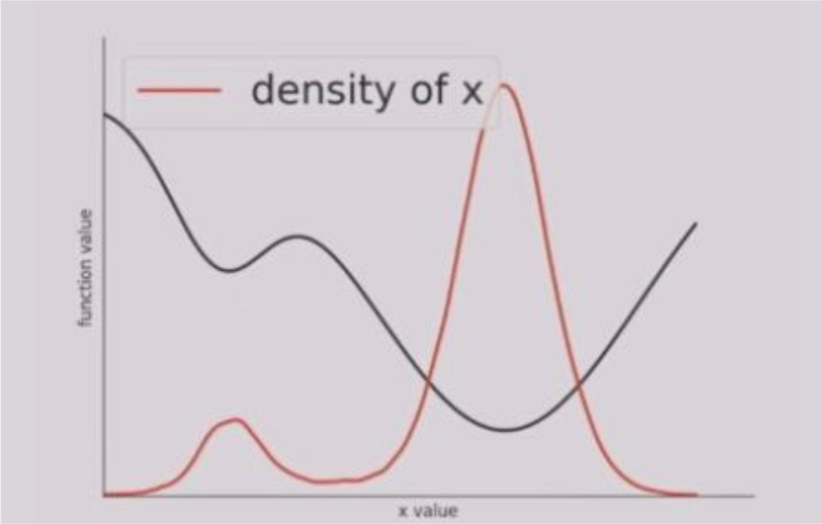
With the extra Gaussian noise, the theoretical analysis can be much easier sometimes
The Gaussian noise is useful sometimes in practice (sometimes not) [Zhu et at. 2019]

SGLD

The continuous case (Langevin Monte Carlo)

Langevin dynamics: $dw(t) = -\nabla f(w)dt + \sqrt{2/\beta}dB(t)$

Stationary distribution: $\pi(x) \propto e^{-\beta f(x)}$



Related to Bayesian inference [Welling, Teh. 11]….

It hits a (nearly) stationary point in poly-time [Zhang et al. 17][Du et al. 19]

Excess risk is small when the distr close to stationary [Raginsky et al. 17]

(but it may take exponential time to mix)

Outline

- Generalization
    - SGD,SGLD
    - Bayes-Stability
    - Extensions
- Implicit Bias
    - Smoothed Normalized Margin
    - Main Results
    - Robustness

Bayes-Stability Framework

A new framework combining algorithm stability and some ideas from PAC Bayesian

P : prior distr, independent of training data S

Q_S : distribution of W_T for a given dataset S

$$Q_{z_n} \leftarrow \mathrm{E}_{(z_1,\ldots,z_{n-1})}[Q_{(z_1,\ldots,z_n)}]$$

Theorem Assuming the loss is bounded by C, the generalization can be bounded by

$$2C\,\mathbb{E}_z\left[\sqrt{2\mathrm{KL}(P,Q_z)}\right] \quad \text{or} \quad 2C\,\mathbb{E}_z\left[\sqrt{2\mathrm{KL}(Q_z,P)}\right]$$

Our Result

SGLD with mini batch $$W_t \leftarrow W_{t-1} - \gamma_t g_t(W_{t-1}) + \frac{\sigma_t}{\sqrt{2}}\mathcal{N}(0, I_d)$$

Theorem

Suppose loss function f is C-bounded. The Batch size is less equal to n/2, learning rate is γ_t.
The generalization error of SGLD can be bounded by

$$\mathrm{err}_{gen} = O\left(\frac{C}{n}\sqrt{\mathop{\mathbb{E}}_{S\sim\mathcal{D}^n}\left[\sum_{t=1}^{T}\frac{\gamma_t^2}{\sigma_t^2}\mathrm{g_e}(t)\right]}\right)$$

$$\mathrm{g_e}(t) = \mathbb{E}_{w\sim W_{t-1}}\left[\frac{1}{n}\sum_{i=1}^{n}\|\nabla f(w, z_i)\|_2^2\right]$$

Average Gradient Norm wrt training data/population along the optimization path

- Independent of #parameters
- Typically, $T \ll O(n^2)$
- Larger σ^2 is good for generalization, but hurts optimization

One cannot obtain such bound using the standard stability framework

Comparison with previous results

Previous bound for SGD in [Hardt et al. ICML16]

- Convex: $O(\frac{L^2}{n}\sum_t \gamma_t)$
- Nonconvex: $O(T^{1-\frac{1}{\beta c+1}}/n)$ (step size $\gamma_t \leq c/t$, β-smooth)

Typical practice in deep learning: the constant step size for several epochs, then decrease the step size, and then repeat. So the above assumption doesn't really apply

Comparison with previous results

Previous approach in [Mou et al. COLT18] (only for b=1)

Their bound= $O\left(\frac{LC}{n}\sqrt{\sum_t \frac{\gamma_t^2}{\sigma_t^2}}\right)$ $\quad \mathrm{err}_{gen} = O\left(\frac{C}{n}\sqrt{\mathbb{E}_{S\sim\mathcal{D}^n}\left[\sum_{t=1}^{T}\frac{\gamma_t^2}{\sigma_t^2}\mathrm{g}_e(t)\right]}\right)$

L: Worst case Lipschitz constant
Unknown, very large for NN

$\leq L^2$

Their technique:

- Interpolate SGLD steps using SDE
- Use Fokker-Planck to derive a bound for $\partial H(W_t, W_t')/\partial t$

$$\frac{\partial P_t}{\partial t} = \Delta P_t + \nabla \cdot (P_t \nabla f)$$

- Using FP, we can only get information about the distr P_t (only) at time t
- Hence, it is a pointwise proof (doesn't work if the final output dependent on the path)
- In practice, we take the average of all steps, or the average of the suffix of certain length [e.g., Shamir&Zhang]

We can obtain their result with a much simpler proof, and our proof is pathwise.

Comparison with previous results

In practice, we take the average of all steps, or the average of the suffix of certain length [e.g., Shamir&Zhang]

Previous bound in [Pensia et al. ISIT18] (only for b=1)

Pathwise analysis, works for the averaging schemes.

Their Bound:

$O\left(\sqrt{I(S;W)/n}\right) \leq O(\sqrt{\sum \gamma_t^2/n})$ (only scales with $1/\sqrt{n}$)

Some additional results

Our bounds

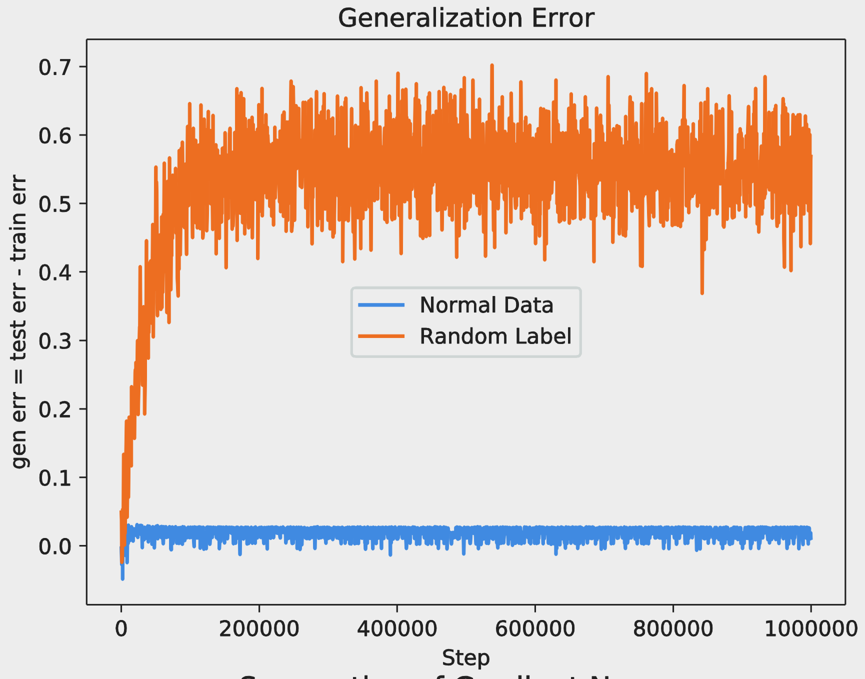
- Proof simpler
- Work for arbitrary averaging scheme (pathwise)
- Easily extended to momentum, aceleration and other variants (e.g., Entropy-SGD [Chaudhari et al. 2016])
- Extended to other continuous noises (log-Lipschitz)
- Can better explain the experiments in [Zhang et al. 2016]

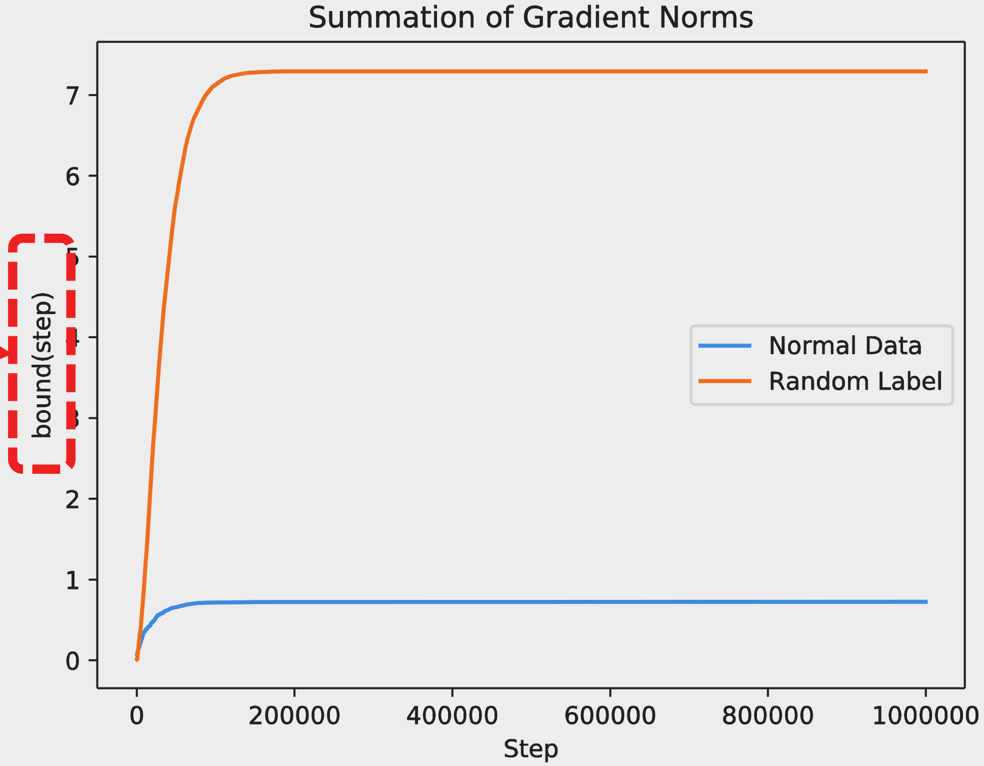
Bayes-Stability bound (SGLD)

flatness of f

$$\mathrm{err}_{gen} = O\left(\frac{C}{n}\sqrt{\mathop{\mathbb{E}}_{S\sim\mathcal{D}^n}\boxed{\left[\sum_{t=1}^{T}\frac{\gamma_t^2}{\sigma_t^2}\mathrm{g}_e(t)\right]}}\right)$$

Flatter training path leads to better generalization
Normal data has a much flatter training path



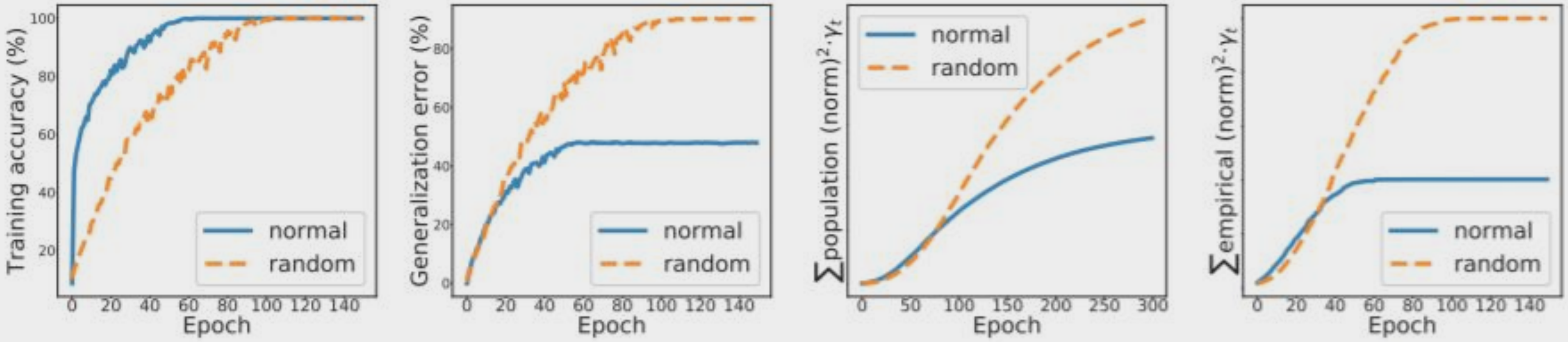


Random Label : y is drawn uniformly from {-1,1}

Normal Data : If label < 5 then y = -1 else y = 1

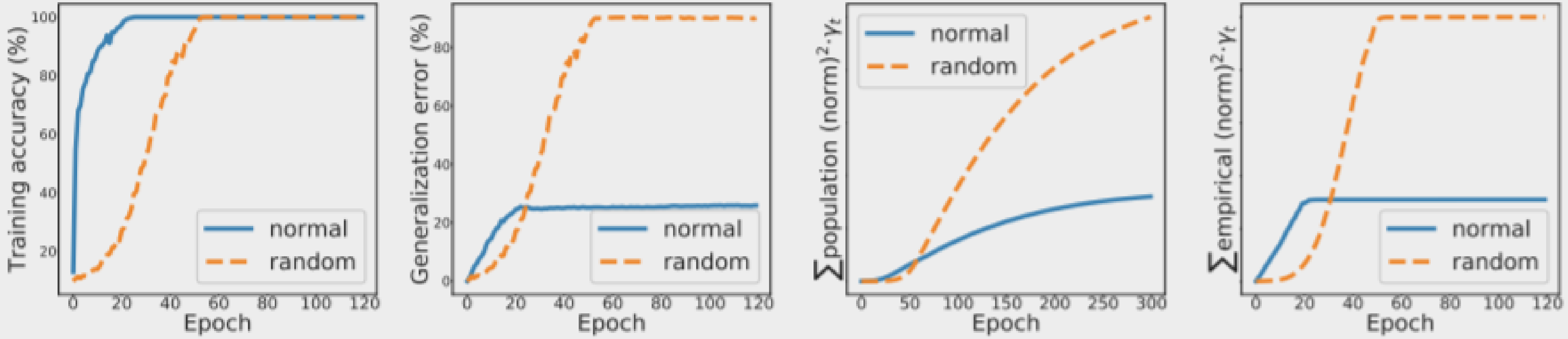
Training CNN on Mnist Dataset.
Convert to binary classification problem,
(x,y) is a data point, y = -1 or 1.

Bayes-Stability bound (SGLD)



(a) (b) (c) (d)

MLP on CIFAR10 with SGLD



(a) (b) (c) (d)

AlexNet on CIFAR10 with SGLD

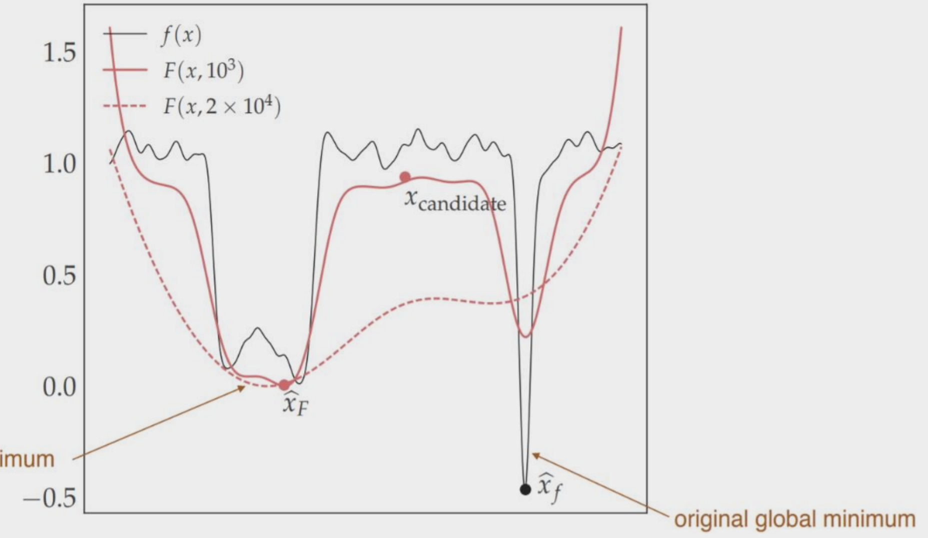
Entropy-SGD [Pratik et al. 2017]

Local entropy: $f_\gamma(w) = -\log \int_{w'} \exp\left(-f(w') - \frac{1}{2\gamma}\|w - w'\|^2\right) dw'$

$$\operatorname*{argmin}_w \; -\log\left(G_\gamma * e^{-f(w)}\right)$$

Gaussian kernel of variance γ

focuses on the neighborhood of w



Picture from [Pratik et al.]

Difficult to estimate the gradient of Local entropy:

- use MCMC
- The resulting algorithm is similar to SGLD
- We can show similar generalization bound

$$\mathrm{err}_{\mathrm{gen}} \leq 2C\sqrt{\frac{1}{2}\mathrm{KL}(W_{T,L+1}, W'_{T,L+1})} = O\left(\frac{C\sqrt{\eta}L_g}{\varepsilon n}\sqrt{TL}\right)$$

Outline

- Generalization
  - SGD,SGLD
  - Bayes-Stability
  - Extensions
- Implicit Bias
  - Smoothed Normalized Margin
  - Main Results
  - Robustness

Implicit Bias

- A traditional wisdom in ML
  - Many models tend to overfit if you train longer (increase the complexity of the model)
  - Trick: Early Stopping or adding l2 –regularizations (capacity control)
- **Mystery in DL: Early stopping/ l2-regularization is not so useful.**
- For DNN, the training objective has many global minima.
  - (For overparameterized super-wide NN, there is a global optimal near every initialization point [Du et al. 18] [Jacot et al. 2018][Arora et al. 19])
- The optimization algorithm may **implicitly bias** the solutions to global minima with special properties.
  - Implicit bias is particularly important in learning deep neural networks as "it introduces **effective capacity control** not directly specified in the objective" [Gunasekar et al. 18] (without explicit regularization and early stopping)

Related Work

- For 2-layer overparametrized network (with leakyReLU activation and linearly separable data), [Brutzkus et al. 17] show SGD can find global optimum for hinge loss.
- For (deep) linear logistic regression, there is no attainable global minima.
  - So the solution does not converge.
  - But for linear separable data, the direction of the solution (hence decision boundary) converges to the hard margin support vector machine solution [Soudry et al., 2018] [Nacson et al., 2018].
- [Ji and Telgarsky, 2018] characterized the convergence of weight without assuming separability;
- [Gunasekar et al., 2018] characterized the convergence of weight direction for other optimization methods, and provided results for (full-width) deep linear convolutional networks (biases toward linear separators that are sparse in the frequency domain).
- The regularization path $\Theta_r(\lambda) = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_2^2$ converges to a max margin solution for homogeneous DNN with cross entropy or logistic loss [Wei et al. 18].

The setting

Deep Homogeneous Networks (binary classification for this talk):

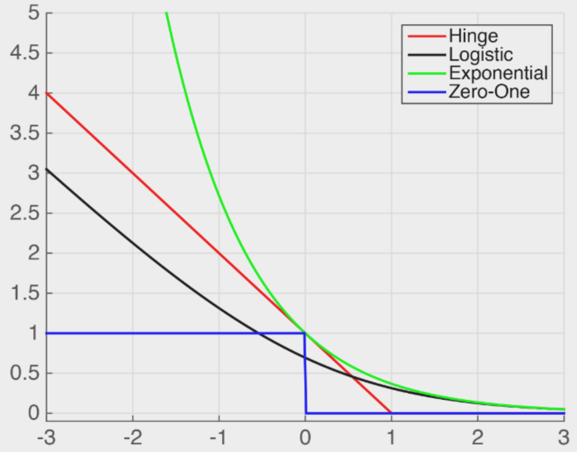
- A function *F(x)* is *k*-homogeneous if for all input x

$$F(\alpha \boldsymbol{x}) = \alpha^k F(\boldsymbol{x})$$

- Output of the neural network: $\Phi(\boldsymbol{\theta}; \boldsymbol{x}) \in \mathbb{R}$
  - For ReLU (or leakyReLU) network (without bias terms), the output is *k*-homogeneous if there are *k* layers

- Training loss: $\mathcal{L}(\boldsymbol{\theta}) := \sum_{n=1}^{N} \ell(y_n \Phi(\boldsymbol{\theta}; \boldsymbol{x}_n))$

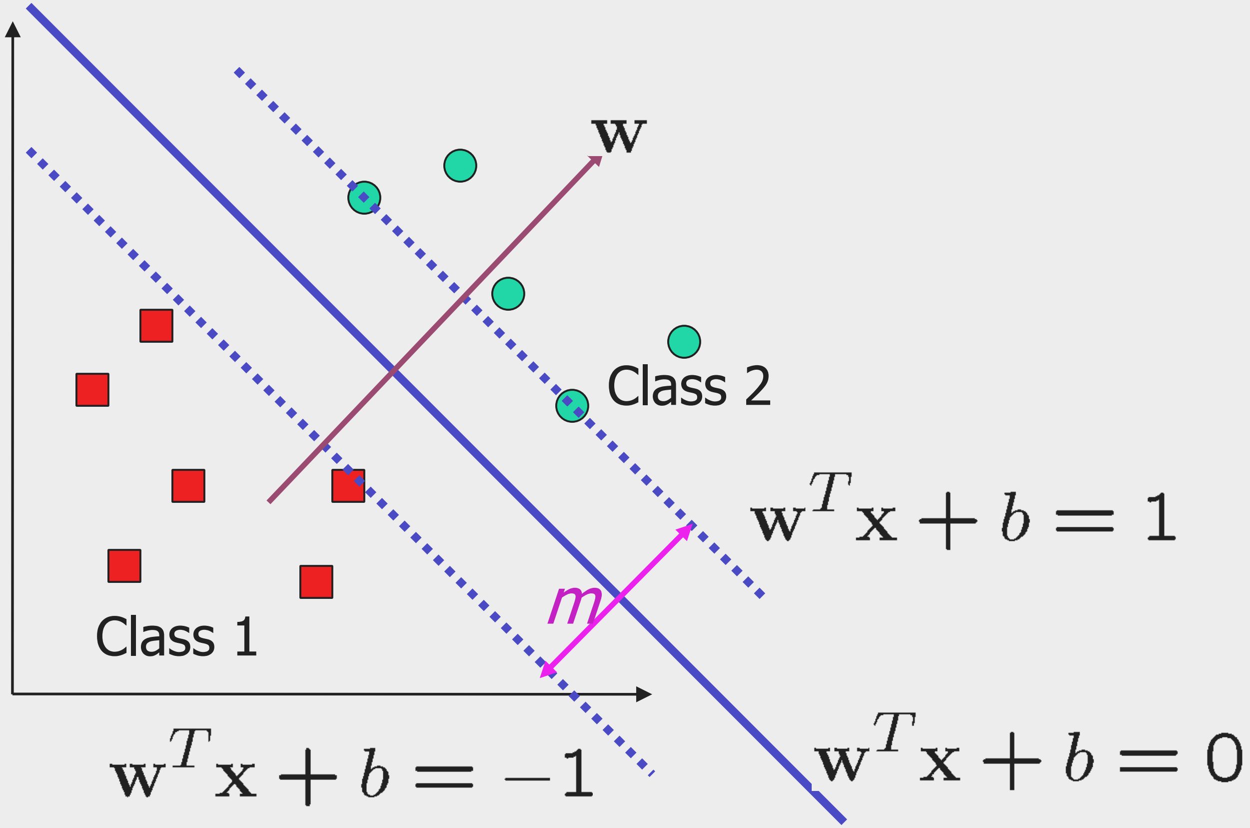
- We mainly consider the following loss func
  - Exponential loss: $\ell(q) := e^{-q}$
  - Logistic loss: $\ell(q) = \log(1 + e^{-q})$
  - Note that such loss has no global min



Outline

- Generalization
  - SGD,SGLD
  - Bayes-Stability
  - Extensions
- Implicit Bias
  - Smoothed Normalized Margin
  - Main Results
  - Robustness

Good Decision Boundary: Linear case (SVM)



- Maximize the margin, m

$$m = \frac{2}{||\mathbf{w}||} \qquad \|\mathbf{x}\| := \sqrt{x_1^2 + \cdots + x_n^2}.$$

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

Smoothed Normalized Margin

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

$$\mathcal{L}(\boldsymbol{\theta}) := \sum_{n=1}^{N} \ell(y_n \Phi(\boldsymbol{\theta}; \boldsymbol{x}_n))$$

- Margin of (x_n, y_n): $q_n(\boldsymbol{\theta}) := y_n \Phi(\boldsymbol{\theta}; \boldsymbol{x}_n)$.
- Margin: $q_{\min}(\boldsymbol{\theta}) := \min_{n \in [N]} q_n(\boldsymbol{\theta})$
  - We hope the margin to be large (smaller loss, better classification)
  - But the margin can approach to infinity (due to homogeneity)

Smoothed Normalized Margin

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

- Margin of (x_n, y_n): $q_n(\boldsymbol{\theta}) := y_n\Phi(\boldsymbol{\theta}; \boldsymbol{x}_n)$.

$$\mathcal{L}(\boldsymbol{\theta}) := \sum_{n=1}^{N} \ell(y_n\Phi(\boldsymbol{\theta}; \boldsymbol{x}_n))$$

- Margin: $q_{\min}(\boldsymbol{\theta}) := \min_{n\in[N]} q_n(\boldsymbol{\theta})$
  - We hope the margin to be large (smaller loss, better classification)
  - But the margin can approach to infinity (due to homogeneity)

$$\text{Maximize } m$$

$$\text{subject to } \frac{y_i(w^T x_i + b)}{||w||} \geq \frac{m}{2} \qquad \forall i$$

- So we consider the normalized margin (only consider the direction since the direction is enough to determine the prediction, due to homogeneity):

$$\bar{\gamma}(\boldsymbol{\theta}) := q_{\min}(\hat{\boldsymbol{\theta}}) = q_{\min}(\boldsymbol{\theta})/\rho^L \qquad \rho := \|\boldsymbol{\theta}\|_2 \qquad \hat{\boldsymbol{\theta}} := \boldsymbol{\theta}/\rho \in \mathcal{S}^{d-1}$$

Smoothed Normalized Margin

- But the normalized margin is difficult to analyze
- Consider smoothed normalized margin (change min to softmin)

$$\tilde{\gamma}(\boldsymbol{\theta}) := \rho^{-L} \log \frac{1}{\mathcal{L}} \qquad \log \frac{1}{\mathcal{L}} = -\log \left(\sum_{n=1}^{N} e^{-q_n} \right)$$

- One can easily show

$$\bar{\gamma} - \rho^{-L} \log N \le \tilde{\gamma} \le \bar{\gamma}$$

- So, as $\rho \to +\infty$, we have $\tilde{\gamma} \to \bar{\gamma}$.
- In fact, we will show $\rho \to +\infty$.

Outline

- Generalization
  - SGD,SGLD
  - Bayes-Stability
  - Extensions
- Implicit Bias
  - Smoothed Normalized Margin
  - Main Results
  - Robustness

Our Results

- Consider the gradient flow

$$\frac{d\boldsymbol{\theta}(t)}{dt} \in -\partial^{\circ}\mathcal{L}(\boldsymbol{\theta}(t)) \qquad \text{for a.e. } t \geq 0.$$

Clarke subdifferential

- Assume that we have fitted the training data at time t_0.

Theorem 1: SNM increases monotonically.

1. *For a.e.* $t > t_0$, $\frac{d\tilde{\gamma}}{dt} \geq 0$;
2. *For a.e.* $t > t_0$, *either* $\frac{d\tilde{\gamma}}{dt} > 0$ *or* $\frac{d\hat{\boldsymbol{\theta}}}{dt} = 0$;
3. $\mathcal{L} \to 0$ *and* $\rho \to \infty$ *as* $t \to +\infty$; *therefore,* $|\bar{\gamma}(t) - \tilde{\gamma}(t)| \to 0$.

If $\ell(\cdot)$ *is the exponential or logistic loss, then for* $t > t_0$,

$$\mathcal{L}(t) = \Theta\left(\frac{1}{t(\log t)^{2-2/L}}\right) \quad \text{and} \quad \rho = \Theta((\log t)^{1/L}).$$

Our Results

Max-Margin Problem: (P)

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{\theta}\|_2^2 \\ \text{s.t.} \quad & q_n(\boldsymbol{\theta}) \geq 1 \qquad \forall n \in [N] \end{aligned}$$

Classical SVM

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

Theorem 2: For every limit point of the direction $\widehat{\boldsymbol{\theta}}$, $\widehat{\boldsymbol{\theta}}/q_{\min}(\widehat{\boldsymbol{\theta}})^{1/L}$ is a KKT point of (P).

Definition A feasible point $\boldsymbol{\theta}$ of (P) is a KKT point if there exist $\lambda_1, \dots, \lambda_N \geq 0$ such that

1. $\boldsymbol{\theta} - \sum_{n=1}^{N} \lambda_n \boldsymbol{h}_n = \mathbf{0}$ for some $\boldsymbol{h}_1, \dots, \boldsymbol{h}_N$ satisfying $\boldsymbol{h}_n \in \partial^\circ q_n(\boldsymbol{\theta})$;
2. $\forall n \in [N] : \lambda_n(q_n(\boldsymbol{\theta}) - 1) = 0$.

First order (necessary) condition for a local optimal solution in a constrained optimization problem

Comparing to an independent recent work [Nacson et al. 19], we use much weaker assumptions. They have some other results. E.g., convergence to "lexicographic max-margin" solution.

Our Results

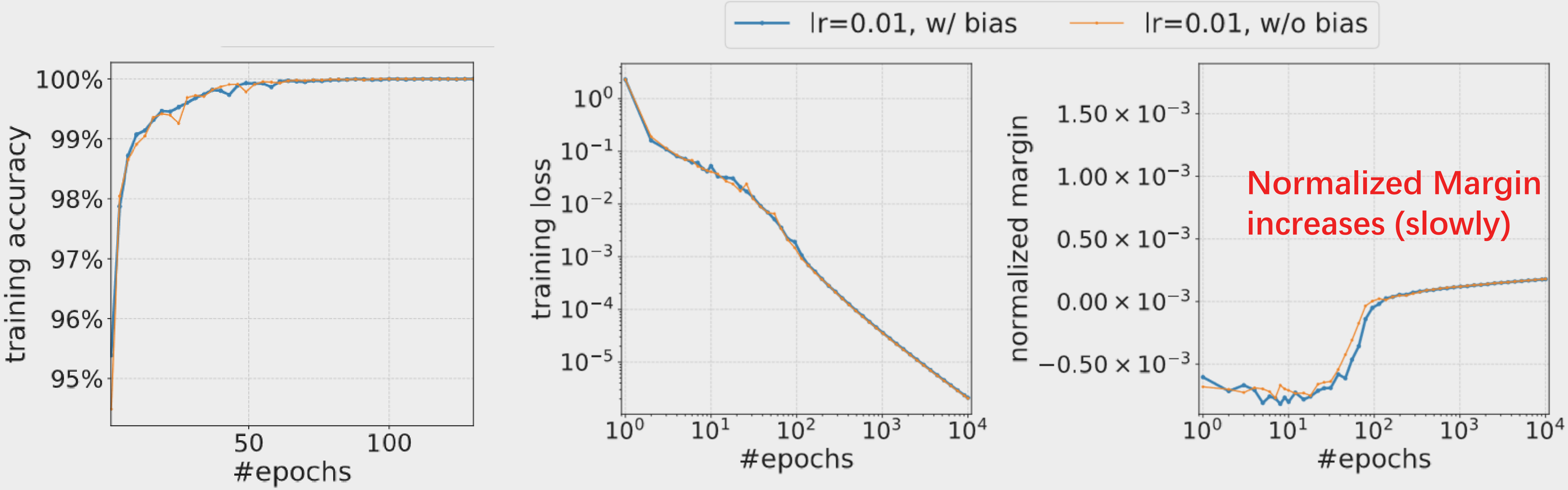
Corollary: For every limit point of the direction $\hat{\theta}$ is along the max-margin direction **for the Kernel SVM with neural tangent kernel** (NTK, introduced in [Jacot et al. 2018])

$$K_{\bar{\boldsymbol{\theta}}}(\boldsymbol{x}, \boldsymbol{x}') = \left\langle \nabla \Phi_{\boldsymbol{x}}(\bar{\boldsymbol{\theta}}), \nabla \Phi_{\boldsymbol{x}'}(\bar{\boldsymbol{\theta}}) \right\rangle$$

Kernel SVM:

$$\min \quad \frac{1}{2}\|\boldsymbol{\theta}\|_2^2 \qquad s.t. \quad y_n \left\langle \boldsymbol{\theta}, \nabla \Phi_{\boldsymbol{x}_n}(\bar{\boldsymbol{\theta}}) \right\rangle \geq 1 \qquad \forall n \in [N]$$

Experiments



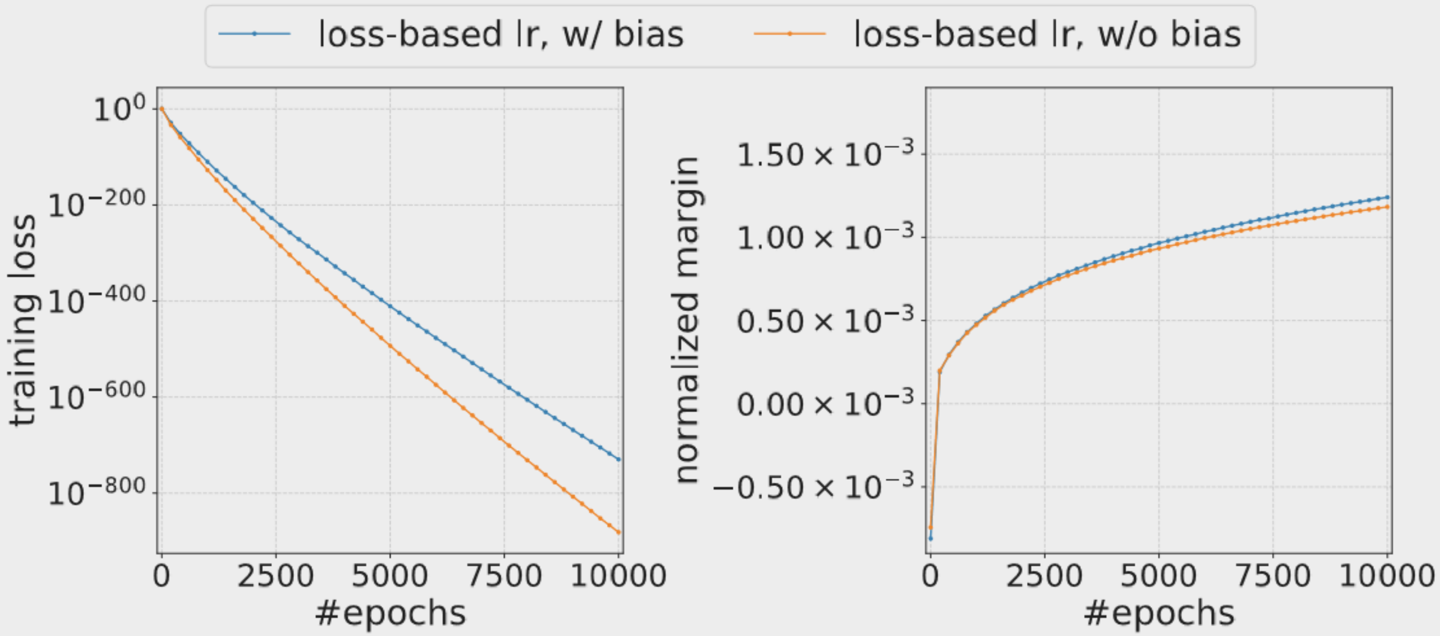
(a)

CNN, MNIST, constant learning rate

conv-32 with filter size 5×5, max-pool, conv-64 with filter size 3×3, max-pool, fc-1024, fc-10

Standard architecture used in MNIST Adversarial Examples Challenge

Experiments



(b)

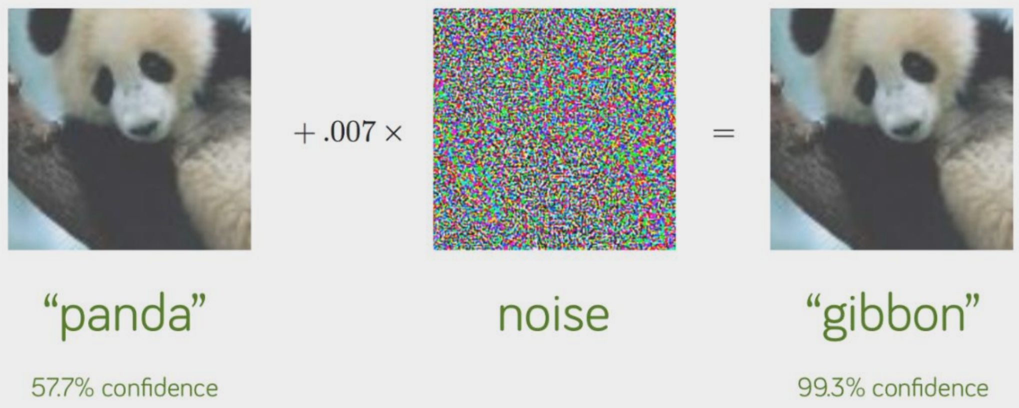
- Constant LR: Gradient very small, loss decreases very slowly
- We can increase the learning rate! (based on the loss)
- SGD with Loss-based Learning Rate.
  - Training loss so small. We even have to modify Tensorflow to deal with numerical issues

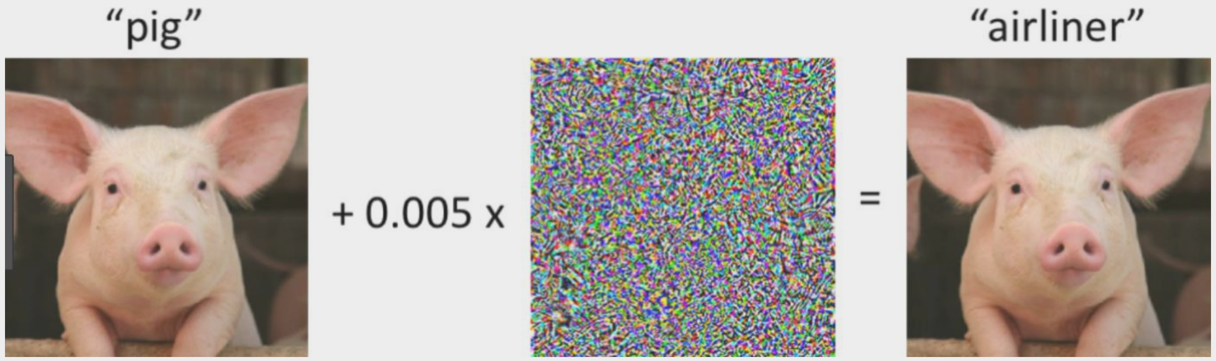
Outline

- Generalization
  - SGD,SGLD
  - Bayes-Stability
  - Extensions
- Implicit Bias
  - Smoothed Normalized Margin
  - Main Results
  - Robustness

Robustness

- Adversarial examples in deep learning (first found in [Szegedy et al. 13])





- Accuracy drops to nearly zero in the presence of small adversarial perturbations
- Geometrically, every training sample (as well as testing sample) is very close to the decision boundary.

Robustness

- Robustness

$$R_{\boldsymbol{\theta}}(\boldsymbol{z}) := \inf_{\boldsymbol{x}' \in X} \{\|\boldsymbol{x} - \boldsymbol{x}'\| : (\boldsymbol{x}', y) \text{ is misclassified}\}$$

- Robustness and normalized margin
  - If q is β-Lipschitz, it is easy to see that (see e.g.,[Sokolic et al., 2017])

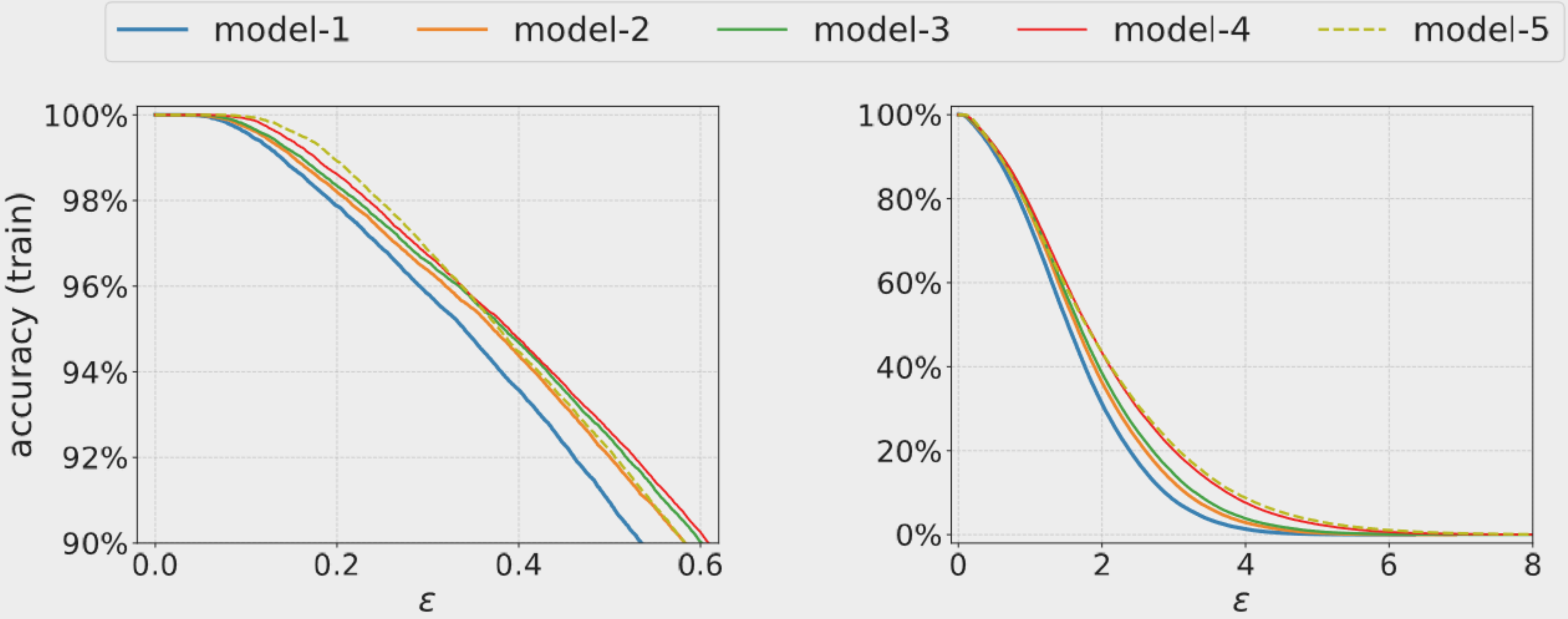
$$R_{\boldsymbol{\theta}}(\boldsymbol{z}) \geq \frac{q_{\hat{\boldsymbol{\theta}}}(\boldsymbol{z})}{\beta}$$

  - So larger normalized margin perhaps implies better robustness

Robustness

The robust accuracy
(the percentage of data with robustness $\geq \epsilon$)

model name	number of epochs	train loss	normalized margin
model-1	38	$10^{-10.04}$	5.65×10^{-5}
model-2	75	$10^{-15.12}$	9.50×10^{-5}
model-3	107	$10^{-20.07}$	1.30×10^{-4}
model-4	935	$10^{-120.01}$	4.61×10^{-4}
model-5	10000	$10^{-881.51}$	1.18×10^{-3}



Hence, training longer may be useful in improving the robustness.
Hopefully, it can be used in combination with other methods (data augmentation, regularization, ensemble, robust optimization etc.) (future work)

Outline

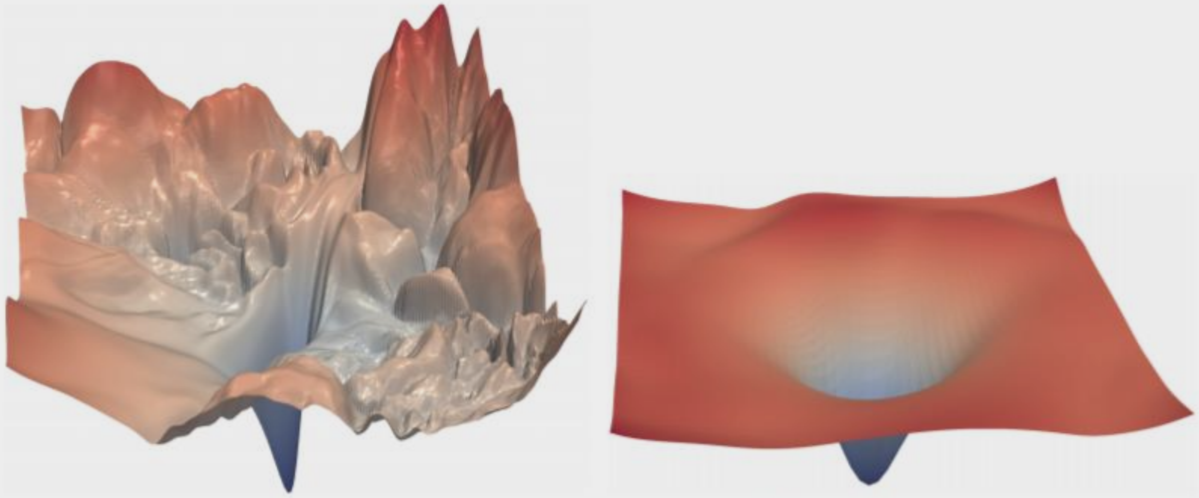
- Generalization
  - SGD,SGLD
  - Bayes-Stability
  - Extensions
- Implicit Bias
  - Smoothed Normalized Margin
  - Main Results
  - Robustness
- Conclusion

Concluding Remarks

- Generalization of SGLD:
  - Bayes-stability framework
  - Generalization error
    - Connection to the sum of gradient variance over the training trajectory
    - Data dependent: can explain the random label experiment
- Implicit bias of GD
  - GD maximizes the normalized margin
  - Equivalent to kernel SVM (with Neural Tangent Kernel)
  - Training longer can potentially improve robustness

Picture from [Li et al.]

Open Problems



(a) without skip connections (b) with skip connections

Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

"Conjecture":
If the landscape of the loss function is "nice", SGLD generalizes.

Handling **discrete noise** like in SGD
The noise structure of SGD is ill-conditioned (very different from isotropic Gaussian noise)

- Mini-batch and Dropout help (make the noise less ill-conditioned)
- But SGD is fairly good even without extra noise (Zhu et al. 19)

Thanks

Jian Li
lapordge@gmail.com

References

- Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: stability of stochastic gradient descent. In International Conference on Machine Learning (ICML), pages 1225–1234, 2016.
- Wenlong Mou, Liwei Wang, Xiyu Zhai, and Kai Zheng. Generalization bounds of sgld for nonconvex learning: Two theoretical viewpoints. In Conference on Learning Theory (COLT), pages 605–638, 2018.
- Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In International Conference on Machine Learning (ICML), pages 1139–1147, 2013.
- Flemming Topsoe. Some inequalities for information divergence and related measures of discrimination. IEEE Transactions on Information Theory, 46(4):1602–1609, 2000.
- Chaudhari P, Choromanska A, Soatto S, et al. Entropy-sgd: Biasing gradient descent into wide valleys. ICLR 2017.
- Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning requires rethinking generalization.“ ICLR 2017.
- Gradient Descent Maximizes the Margin of Homogeneous Neural Networks. Kaifeng Lyu, Jian Li. 2019 （under review)
- On Generalization Error Bounds of Noisy Gradient Methods for Non-Convex Learning. Jian Li, Xuanyuan Luo, Mingda Qiao. 2019. (under review)