Rethinking Generalization and Robustness in Neural Networks: Breiman's Dilemma and Huber's Agnostic Contamination Model

Yuan Yao

HKUST

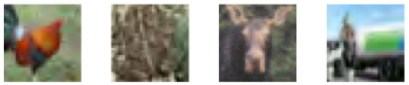
March 8, 2019

Acknowledgements


- PhD students at HKUST: *Weizhi Zhu*, *Yifei Huang*
- Discussions: Tommy Poggio (MIT), Peter Bartlett (UC Berkeley)

Why big models generalize well?

Courtesy of Ben Recht 2016



CIFAR10

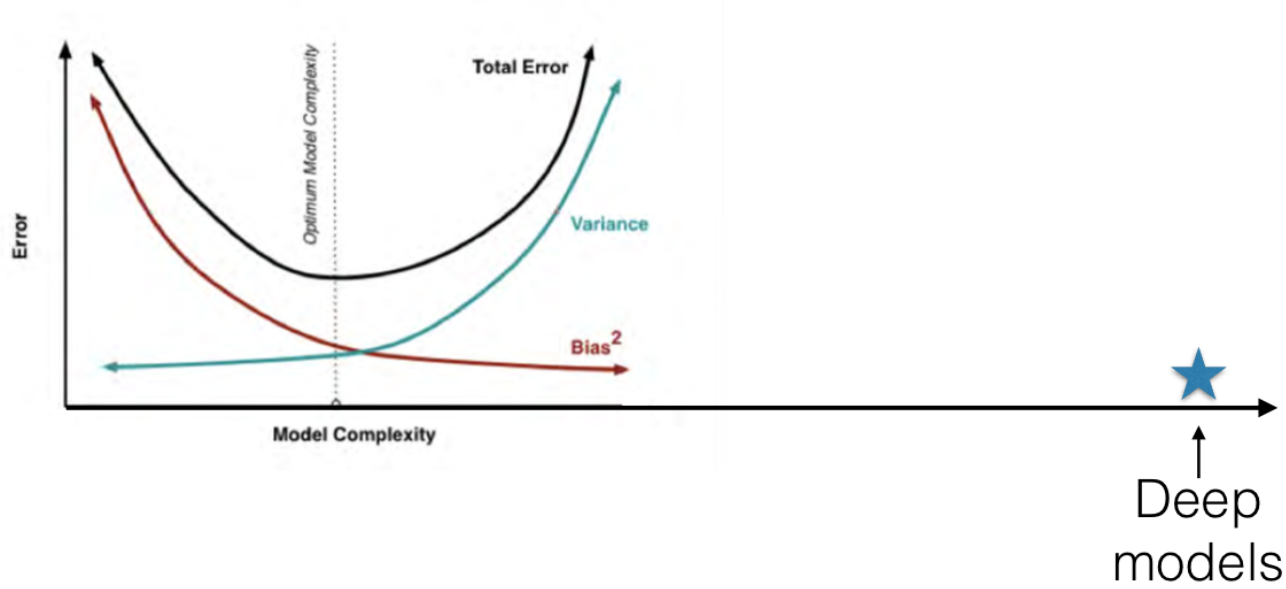
n=50,000
d=3,072
k=10

What happens when I turn off the regularizers?

Model	parameters	p/n	Train ***loss***	Test ***error***
CudaConvNet	145,578	2.9	0	23%
CudaConvNet (with regularization)	145,578	2.9	0.34	18%
MicroInception	1,649,402	33	0	14%
ResNet	2,401,440	48	0	13%

Where is the Bias-Variance Tradeoff?

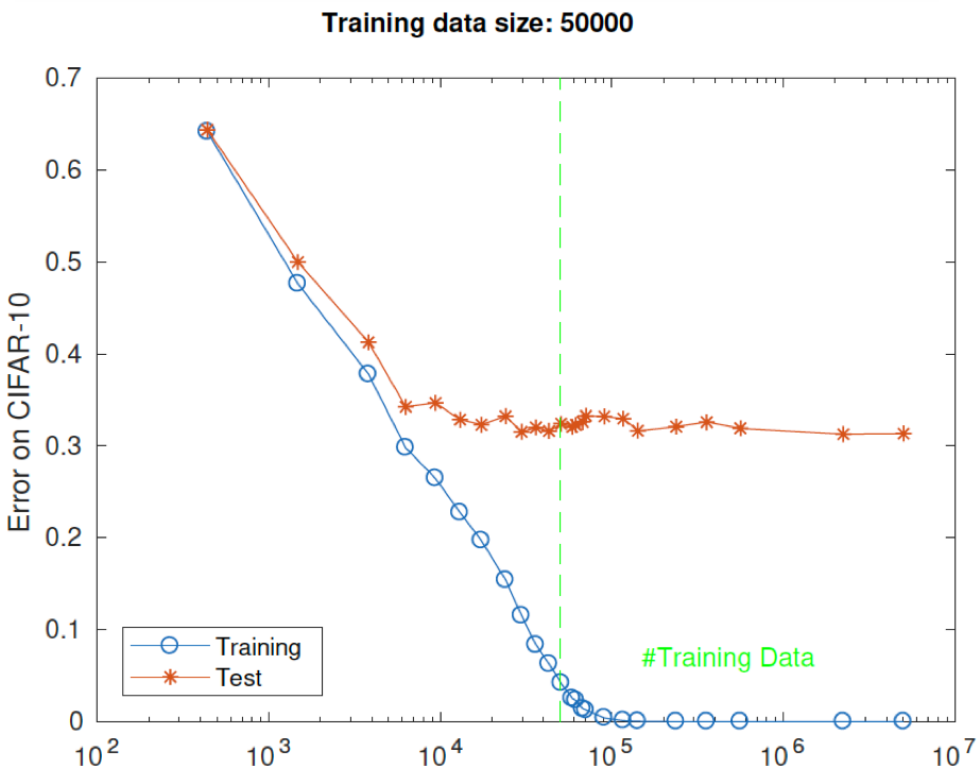
Courtesy of Ben Recht 2016



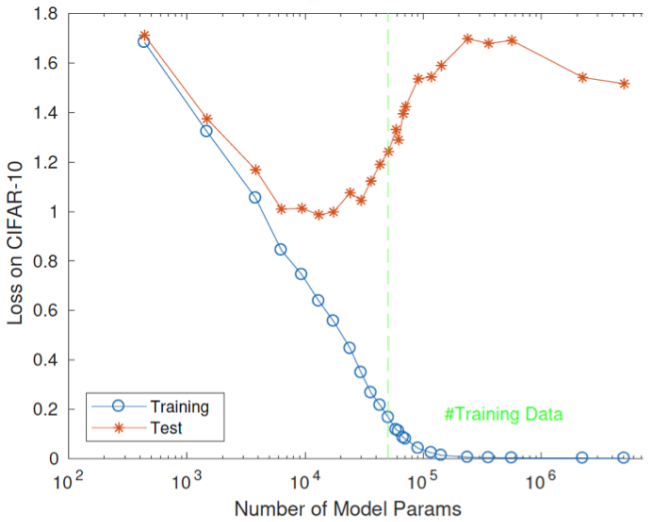
Models where p>20n are common

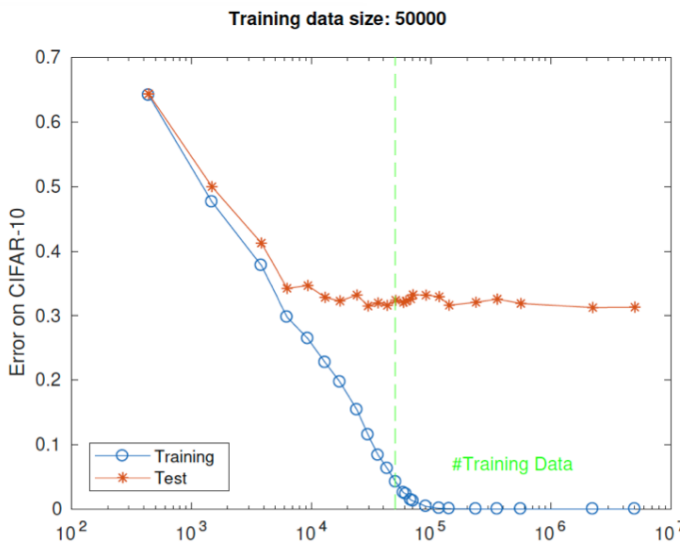
Resistance of Test Error against Overfitting

As model complexity grows ($p > n$), training error goes down to zero, but test error does not increase. Why over-parameterized models do not overfit here? – Tommy Poggio, 2018



Overfitting of Test Loss but not Test Error?





Why?

Towards a Deeper Understanding of Deep Learning

- Margin-Based Complexity Theories for Classification
  - Novikoff 1962: Perceptron halts in finite steps if training data is separable with a margin
  - Vapnik 1994: Large margin classifier as Support Vector Machines
  - Bartlett 1997: generalization error of neural networks is bounded by products of weight norms, as an upper bound of fat shattering dimension for large margin hyperplanes
  - Schapire, Freund, Bartlett, Lee, 1998: resistance to overfitting in AdaBoost due to margin improvement during training
  - Koltchinskii, Panchenko, 2003: improved margin-based generalization error bound of boosting, using Rademacher complexity
  - Bartlett, Foster, Telgarsky, 2017: return of the margin theory for deep neural networks, using Lipschitz upper bound as products of spectral norms of weights

A Dilemma by Leo Breiman

- Summary: Empirical Margin Distributions, measure the stability of classifier, hence improvement on margins will lead to better generalization performance
- Yet, Leo Breiman 1999:
  - designed margin maximization via prediction games and discovered examples that margin improvement leads to worse generalization ability.
  - "The results above leave us in a quandary. ... if we try too hard to make the margins larger, then overfitting sets in. ... My sense of it is that we just do not understand enough about what is going on."

Example: Train 5-layer CNN(n) on Cifar10

- 5 convolutional layers of 3x3 filters, stride 2, padding 1, *n* channels
- Batch normalization (batchsize 100), ReLU
- Fully connected last layer, cross-entropy loss
- Cifar10: 50K train, 10K test, 10% labels are randomly permuted

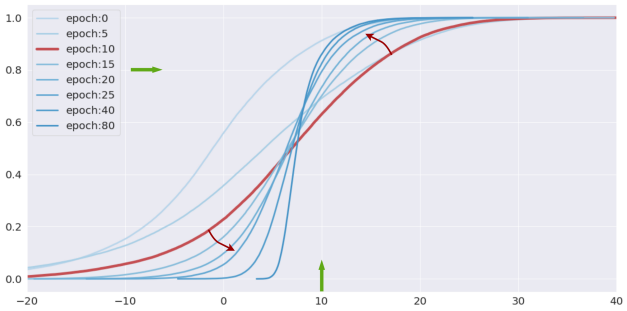
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1           [-1, 50, 16, 16]           1,400
       BatchNorm2d-2           [-1, 50, 16, 16]             100
              ReLU-3           [-1, 50, 16, 16]               0
            Conv2d-4             [-1, 50, 8, 8]          22,550
       BatchNorm2d-5             [-1, 50, 8, 8]             100
              ReLU-6             [-1, 50, 8, 8]               0
            Conv2d-7             [-1, 50, 4, 4]          22,550
       BatchNorm2d-8             [-1, 50, 4, 4]             100
              ReLU-9             [-1, 50, 4, 4]               0
           Conv2d-10             [-1, 50, 2, 2]          22,550
      BatchNorm2d-11             [-1, 50, 2, 2]             100
             ReLU-12             [-1, 50, 2, 2]               0
           Conv2d-13             [-1, 50, 1, 1]          22,550
      BatchNorm2d-14             [-1, 50, 1, 1]             100
             ReLU-15             [-1, 50, 1, 1]               0
           Linear-16                   [-1, 10]             510
================================================================
Total params: 92,610
Trainable params: 92,610
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 0.39
Params size (MB): 0.35
Estimated Total Size (MB): 0.76
----------------------------------------------------------------
```

CNN(50) (#params=93K)

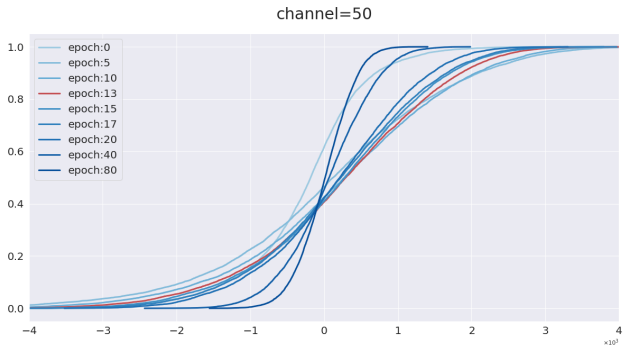
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 400, 16, 16]          11,200
       BatchNorm2d-2          [-1, 400, 16, 16]             800
              ReLU-3          [-1, 400, 16, 16]               0
            Conv2d-4            [-1, 400, 8, 8]       1,440,400
       BatchNorm2d-5            [-1, 400, 8, 8]             800
              ReLU-6            [-1, 400, 8, 8]               0
            Conv2d-7            [-1, 400, 4, 4]       1,440,400
       BatchNorm2d-8            [-1, 400, 4, 4]             800
              ReLU-9            [-1, 400, 4, 4]               0
           Conv2d-10            [-1, 400, 2, 2]       1,440,400
      BatchNorm2d-11            [-1, 400, 2, 2]             800
             ReLU-12            [-1, 400, 2, 2]               0
           Conv2d-13            [-1, 400, 1, 1]       1,440,400
      BatchNorm2d-14            [-1, 400, 1, 1]             800
             ReLU-15            [-1, 400, 1, 1]               0
           Linear-16                  [-1, 10]           4,010
================================================================
Total params: 5,780,810
Trainable params: 5,780,810
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 3.12
Params size (MB): 22.05
Estimated Total Size (MB): 25.19
----------------------------------------------------------------
```

CNN(400) (#params=5.8M)

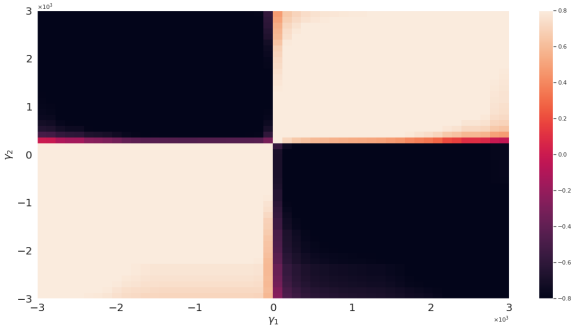
Phase Transitions in Margin Dynamics: nonoverfitting CNN(50)



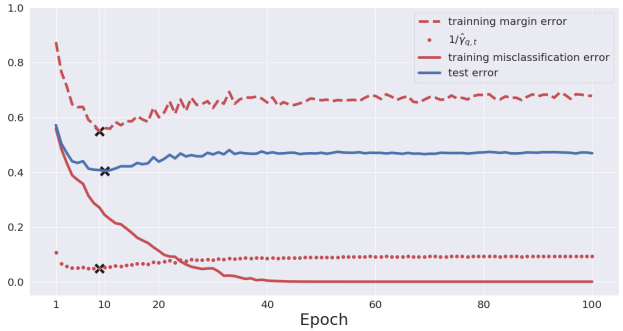
(a) Training Margin Distributions



(b) Test Margin Distributions

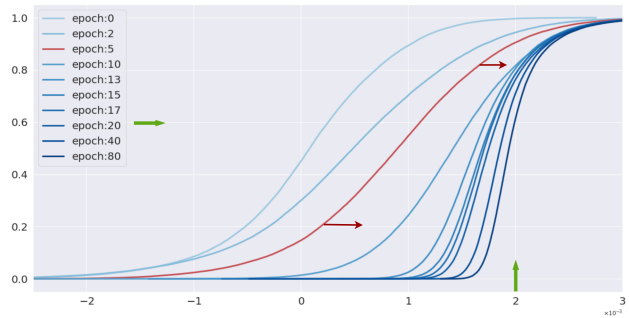


(c) Rank correlations (Spearman-ρ)

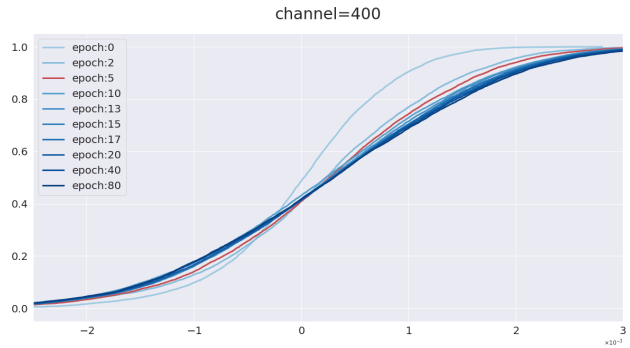


(d) Test error prediction

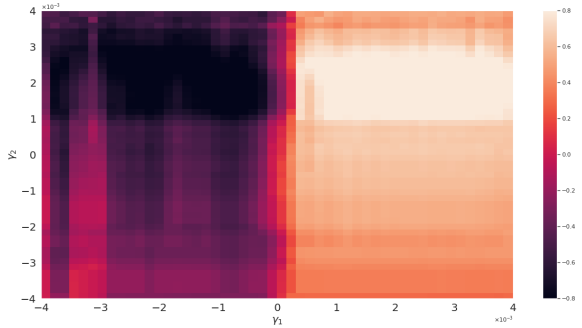
Phase Transitions in Margin Dynamics: overfitting CNN(400)



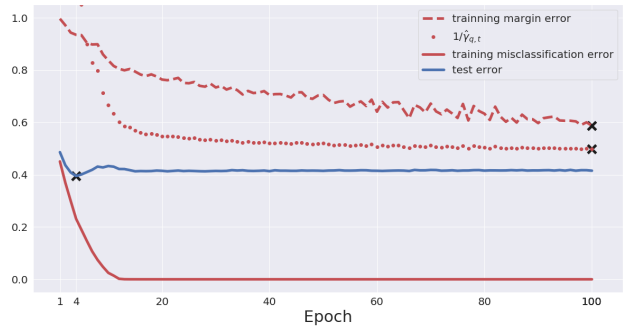
(a) Training Margin Distributions



(b) Test Margin Distributions



(c) Rank correlations (Spearman-ρ)



(d) Overfitting

Space of Neural Network Mappings

Let $\mathcal{X}$ be the input space and $\mathcal{Y} := \{1, \ldots, K\}$ be the output space of K classes. Define $\mathcal{F}$ to be the space of functions $f : \mathcal{X} \to \mathbb{R}^K$ represented by neural networks,

$$\begin{cases} x_0 &= x, \\ x_i &= \sigma_i(W_i x_{i-1} + b_i), \quad i = 1, \ldots, l-1, \\ f(x) &= W_l x_{l-1} + b_l, \end{cases} \tag{1}$$

where l is the depth of the network, W_i is the weight matrix corresponding to a linear operator on x_i and σ_i stands for either element-wise activation function (e.g. ReLU) or pooling operator that are assumed to be Lipschitz bounded with constant L_{σ_i}.

Lipschitz semi-norm

Consider the Lipschitz semi-norm,

$$\|f\|_{\mathcal{F}} := \sup_{x \neq x'} \frac{\|f(x) - f(x')\|_\infty}{\|x - x'\|_2} \leq L_\sigma \|W_l\|_{\infty,2} \prod_{i=1}^{l-1} \|W_i\|_\sigma, \tag{2}$$

where $\|\cdot\|_\sigma$ is the spectral norm, $\|W_l\|_{\infty,2}$ is the operator $\ell_2 \to \ell_\infty$ norm, and $L_\sigma = \prod_{i=1}^{l} L_{\sigma_i}$. Without loss of generality, we assume $L_\sigma = 1$ for simplicity.

Margin

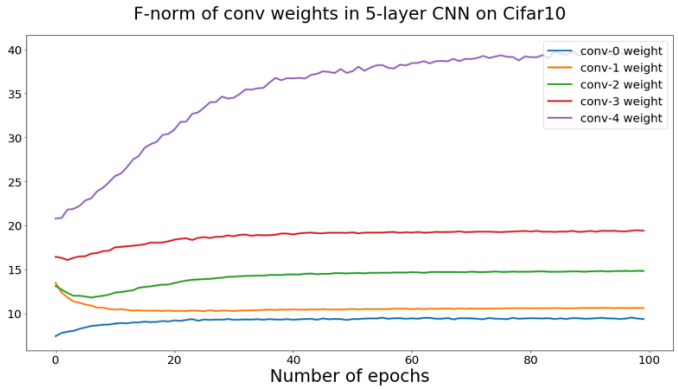
Prediction *margin* of $f(x)$ given label y:

$$\zeta(f(x), y) = [f(x)]_y - \max_{\{j:j\neq y\}} [f(x)]_j$$

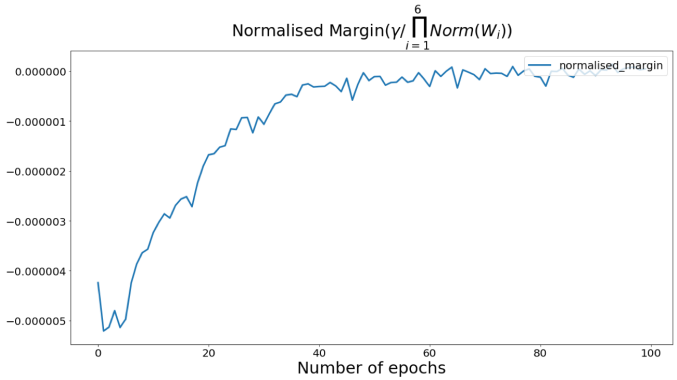
Normalized margins:

$$\zeta(\widehat{f}(x), y) = \frac{[f(x)]_y - \max_{\{j:j\neq y\}}[f(x)]_j}{c\|f\|_{\mathcal{F}}},$$

such that $\widehat{f} = f/(c\|f\|_{\mathcal{F}}) = f/(\|W\|_{\infty,2}\prod_{i=1}^{l-1}\|W_i\|_\sigma)$.



(a) Growth of weights



(b) Normalized Margins

Main Theorem 1

Theorem (Zhu-Huang-Y.'18)

Let $\mathbb{P}_n = \frac{1}{n}\sum_{i=1}^n \delta_{(x_i,y_i)} \sim \mathbb{P}$. *Consider* γ_1 *and* γ_2 *such that* $\gamma_2 > \gamma_1 \geq 0$ *and* $\Delta := \gamma_2 - \gamma_1 \geq 0$, *for any* $\delta > 0$, *with probability at least* $1-\delta$, *there holds*

$$\mathbb{P}[\zeta(\widehat{f}_t(x), y) < \gamma_1] \leq \mathbb{P}_n[\zeta(\widehat{f}_t(x), y) < \gamma_2] + \frac{4K\mathcal{R}_n(\mathcal{F}_1)}{\Delta} + \sqrt{\frac{\log(1/\delta)}{2n}} \tag{3}$$

where the Rademacher complexity of function class $\mathcal{F}_1$ *is defined by*

$$\mathcal{R}_n(\mathcal{F}_1) := \max_{y\in\mathcal{Y}} \mathcal{R}_n(\bar{\mathcal{F}}_1^y) = \max_{y\in\mathcal{Y}} \left\{ \mathbb{E}_{x_i,\varepsilon_i} \sup_{\|f\|_{\mathcal{F}}\leq 1} \frac{1}{n}\sum_{i=1}^n \varepsilon_i [f(x_i)]_y \right\}, \tag{4}$$

the expectation is taken over x_i, ε_i, $i = 1, ..., n$, *and* $\bar{\mathcal{F}}_L^y$ *is the class of evaluation functional of* $f(x) \in \mathbb{R}^K$ *at the* y*-th output.*

Remarks

- When $\mathcal{F}$ is not over-expressive where $\mathcal{R}_n(\mathcal{F}_1)$ can be regarded as a constant, then one can predict the test margin using the training margin via the linear inequality

$$\mathbb{P}[\zeta(\widehat{f}_t(x), y) < \gamma_1] \leq \mathbb{P}_n[\zeta(\widehat{f}_t(x), y) < \gamma_2] + C,$$

  where $\gamma_1 = 0$ gives the test error.

- Yet when $\mathcal{F}$ is over-expressive with monotonic improvement of training margins, $\mathcal{R}_n(\mathcal{F}_1)$ leads to a blow-up upper bound and the prediction above fails (Breiman's Dilemma occurs).

Main Theorem 2

Let $\hat{\gamma}_{q,f}$ be the q^{th} *quantile margin* of the network f with respect to sample S,

$$\hat{\gamma}_{q,f} = \inf\{\gamma : \mathbb{P}_n[\zeta(f(x_i), y_i) \leq \gamma] \geq q\}. \tag{5}$$

Theorem (Inverse Quantile Margin Bound, Zhu-Huang-Y.'18)

Assume the input space is bounded by $M > 0$, that is $\|x\|_2 \leq M$, $\forall x \in \mathcal{X}$. Given a quantile $q \in [0,1]$, for any $\delta \in (0,1)$ and $\tau > 0$, the following holds with probability at least $1-\delta$ for all f_t satisfying $\hat{\gamma}_{q,\widehat{f_t}} > \tau$,

$$\mathbb{P}[\zeta(f_t(x), y) < 0] \leq C_q + \frac{C_{\mathcal{F}}}{\hat{\gamma}_{q,\widehat{f_t}}} \tag{6}$$

$C_q = q + \sqrt{\frac{\log(2/\delta)}{2n}} + \sqrt{\frac{\log\log_2(4(M+l)/\tau)}{n}}$ *and* $C_{\mathcal{F}} = 8K\mathcal{R}_n(\mathcal{F}_1)$.

Examples

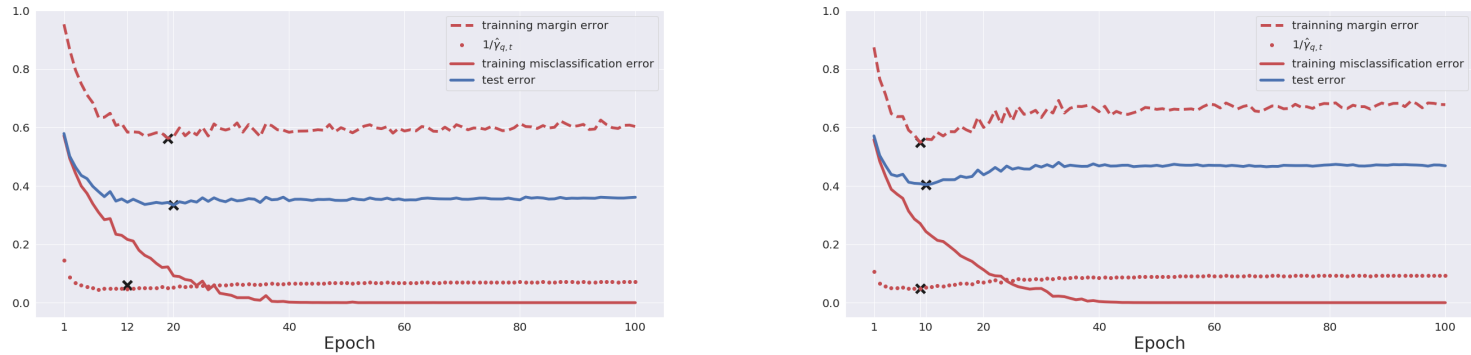


Figure: Success examples. Net structure: basic CNN (50). Dataset: Original CIFAR10 (Left) and CIFAR10 with 10 percents label corrupted (Right). In each figure, we show training error (red solid), training margin error $\gamma = 9.8$ (red dash) and inverse quantile margin (red dotted) with $q = 0.6$ and generalization error (blue solid). The marker "x" in each curve indicates the global minimum along epoch $1, \ldots, T$. Both training margin error and inverse quantile margin successfully predict the tendency of generalization error.

Inverse Quantile Margin can be More Accurate

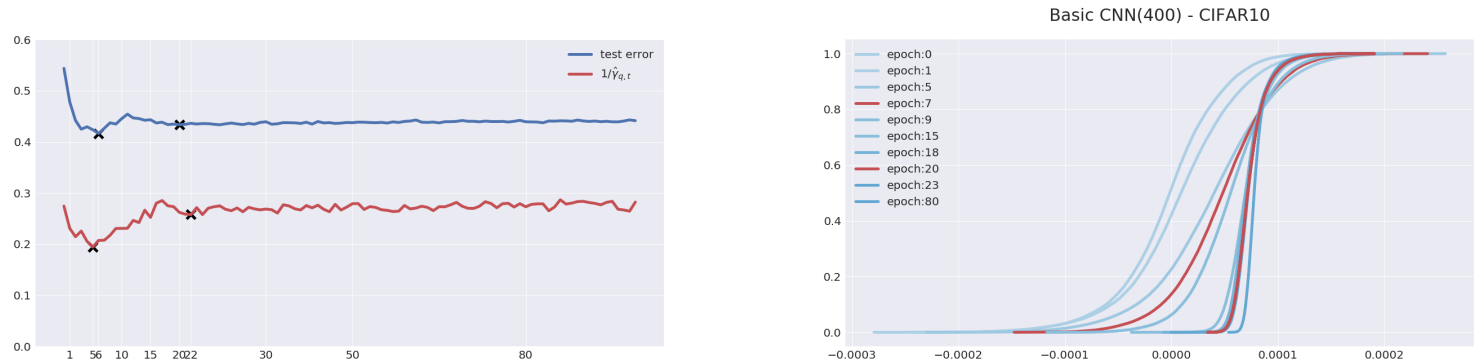


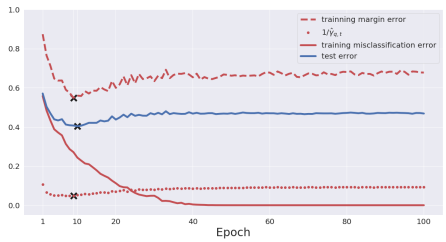
Figure: Inverse quantile margin. Net structure: CNN(200). Dataset: CIFAR10 with 10 percents label corrupted. Left: the dynamics of test error (blue) and inverse quantile margin with $q = 0.95$ (red). Two local minima are marked by "x" in each curve. Right: dynamics of training margin distributions, where two distributions in red color correspond to when the two local minima occur. The inverse quantile margin successfully captures two local minima of test error.

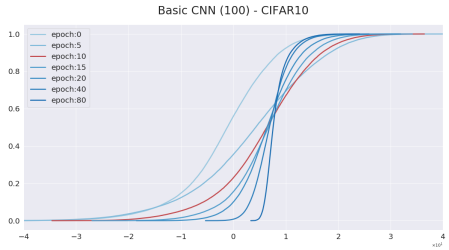
Rank Correlations of Training Margin and Test Error

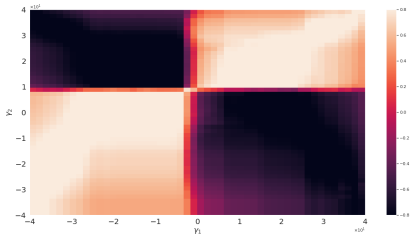
Figure: Spearman's ρ rank correlations between training (or quantile) margins and training errors, as well as training (or quantile) margins and test errors, at different γ (or q, respectively). Net structure: Basic CNN(50). Dataset: CIFAR10. Left: Blue curves show rank correlations between training margin error and test (generalization) error, while Red curves show that between the training margin error and training error, at different γ. Right: Blue curves show rank correlations between inverse quantile margin and test error, and Red curves show that between inverse quantile margin and training error, at different q. Spearman's ρ (Kendall's τ) shows qualitatively that dynamics of large margins are closely related to the test errors in the sense that they have similar trends marked by large rank correlations, while small margins are close to training errors in trend.

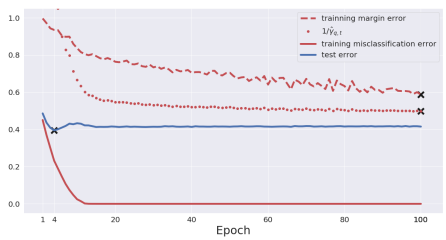
More examples: Cifar10

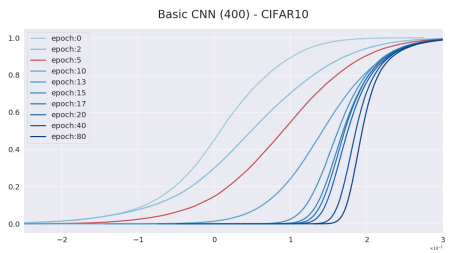
Comparisons of Basic CNNs, AlexNet, VGG16, and ResNet-18 in CIFAR10/100, and Mini-ImageNet. Left: curves of training error, generalization/test error, training margin error and inverse quantile margin. Middle: dynamics of training margin distributions. Right: heatmaps are Spearman-ρ rank correlation coefficients between dynamics of training margin error ($\mathbb{P}_n[\zeta(\widehat{f}(x_i), y_i) < \gamma_2]$) and dynamics of test margin error ($\mathbb{P}[\zeta(\widehat{f}_t(x), y) < \gamma_1]$) drawn on the (γ_1, γ_2) plane.

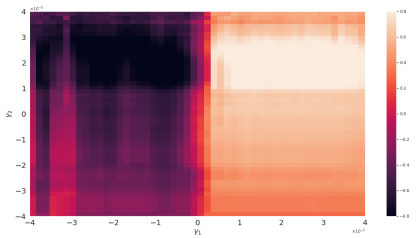




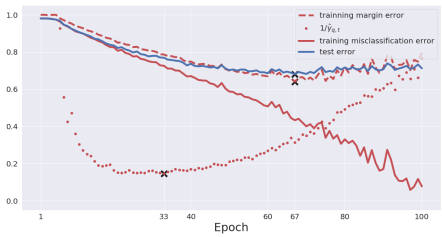


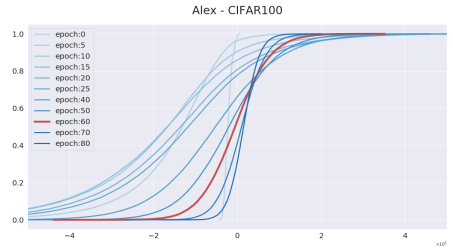


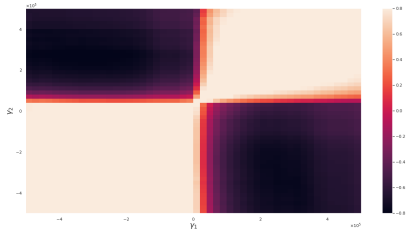


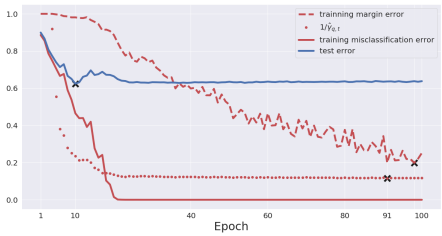


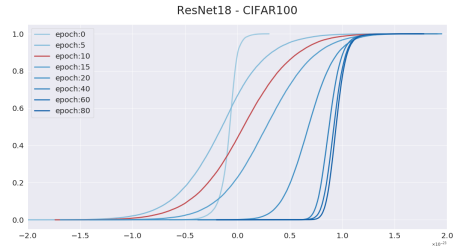
More examples: Cifar100

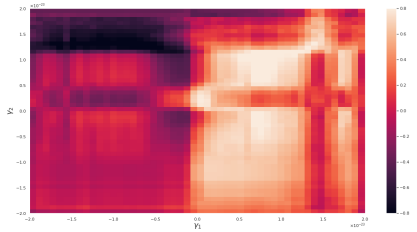


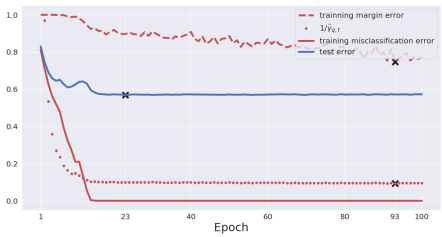


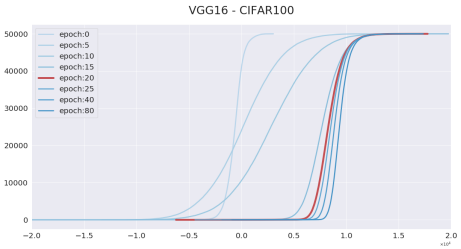


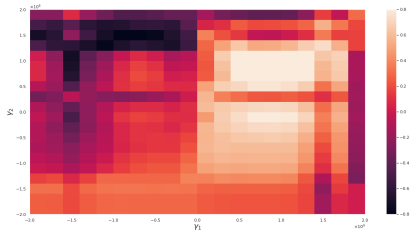




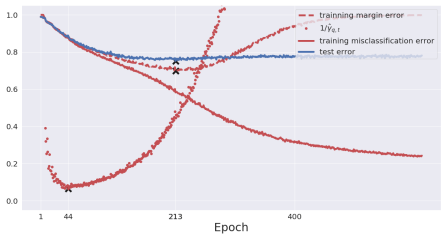


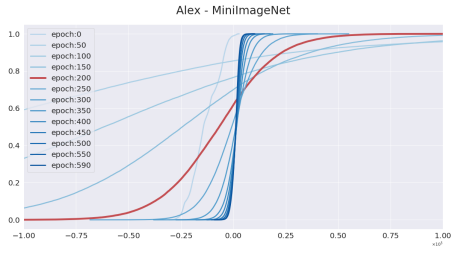


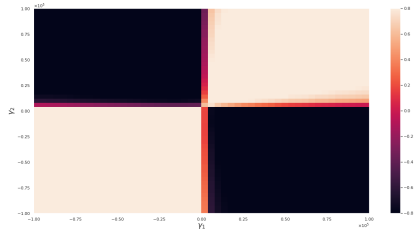


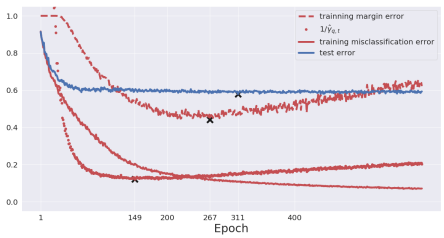


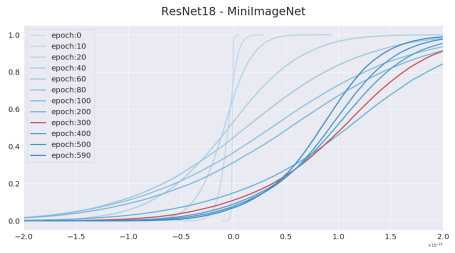
More examples: Mini-ImageNet

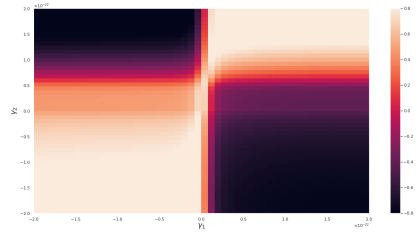


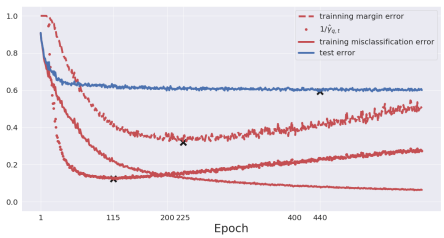


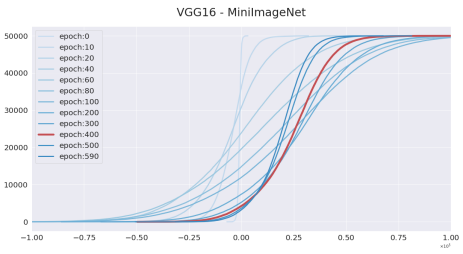


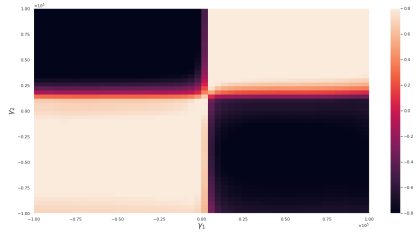










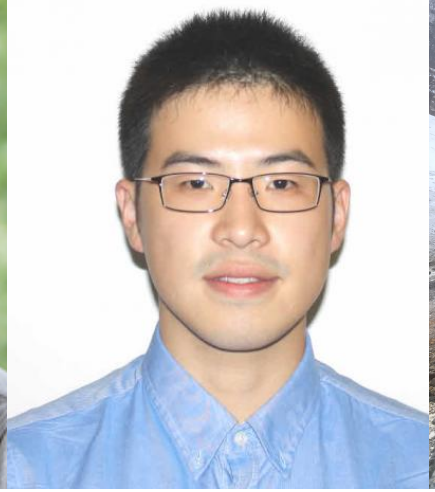


Summary

- Breiman's Dilemma in Margin-based bounds on generalization error
  - Phase transitions of normalized margin dynamics shed light on model expressiveness against data complexity
  - When model expressiveness is comparable to data complexity, such that training margins and test margins share similar phase transitions, one can predict test error using training margin dynamics by restricted Rademacher complexity bounds
  - When model is over-expressive against data, such that training margins are monotonically improved in training, training margins will fail to predict test error

Acknowledgements

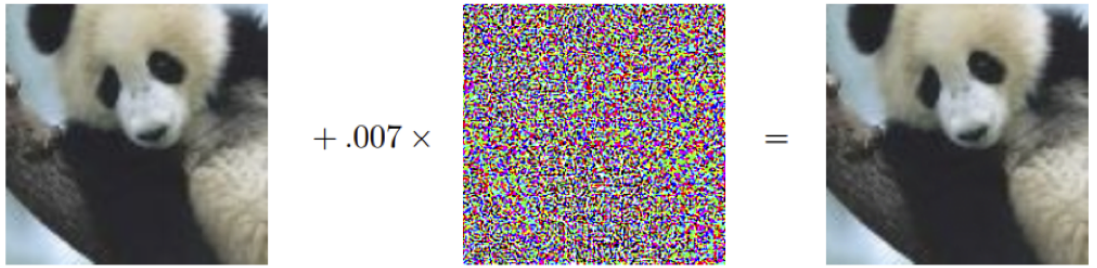
Chao Gao (Chicago)



Jiyu Liu (Yale)

Weizhi Zhu (HKUST)

Deep Neural Networks are Notoriously not Robust



[Goodfellow et al., 2014]

- Imperceivable adversarial examples are ubiquitous to fail neural networks.
- How can one achieve stability or robustness against adversarial?

Stability is the Invariance

- Consider an input-output map

$$f : \mathcal{X} \to \mathcal{Y}$$

  - f is a classification neural network of images
  - f is the area or metric of shapes (geometry)
  - f is a convex/nonconvex classification of shapes
  - f is the number of connected components of shapes (topology)
- How much variation does the input allow without changing the output?

$$f(x + \delta x) = f(x)$$

  - the larger is δx, the larger is the invariance or stability

Robust Optimization

- Traditional training:

$$\min_{\theta} J_n(\theta, \mathbf{z} = (x_i, y_i)_{i=1}^n)$$

  - e.g. square or cross-entropy loss as negative log-likelihood of logit models

- Robust optimization:

$$\min_{\theta} \max_{\|\epsilon_i\| \le \delta} J_n(\theta, \mathbf{z} = (x_i + \epsilon_i, y_i)_{i=1}^n)$$

  - robust to any distributions, yet perhaps too conservative

- Distributional Robust Optimization:

$$\min_{\theta} \max_{\epsilon} \mathbb{E}_{\mathbf{z} \sim P_\epsilon \in \mathcal{D}}[J_n(\theta, \mathbf{z})]$$

  - $\mathcal{D}$ is a set of ambiguous distributions, e.g. Wasserstein ambiguity set
  - intermediate approach with statistically contaminated distributions
  - *sometimes, contamination might be unstructured...*

Huber's Agnostic Contamination Model

[Huber 1964]

$$P_\epsilon = (1-\epsilon)P_\theta + \epsilon Q$$

where Q is arbitrary unknown contamination distribution, ϵ is the probability of contamination, and can you recover the parameter θ?.

- For example, robust estimate of normal mean θ,

$$X_1, \ldots, X_n \sim P_\epsilon = (1-\epsilon)\mathcal{N}(\theta, I_p) + \epsilon Q$$

  - Coordinatewise Median

$$\widehat{\theta^C} = (\hat{\theta}_j)_{j=1}^p, \text{ where } \hat{\theta}_j = \text{Median}(\{X_{ij}\}_{i=1}^n).$$

  - Tukey's Median [Tukey 1975]

$$\widehat{\theta^T} = \arg\max_{\eta \in \mathbb{R}^p} \min_{\|u\|=1} \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{u^T(X_i - \eta) \geq 0\}, \tag{7}$$

$$\ell(\theta, Q) = \text{negative log-likelihood} = \sum_{i=1}^{n} (\theta - X_i)^2$$

$$\sim (1-\epsilon)\mathbb{E}_{\mathcal{N}(\theta)}(\theta - X)^2 + \epsilon \mathbb{E}_Q (\theta - X)^2$$

the sample mean

$$\hat{\theta}_{mean} = \frac{1}{n}\sum_{i=1}^{n} X_i = \arg\min_{\theta} \ell(\theta, Q)$$

$$\min_{\theta} \max_{Q} \ell(\theta, Q) \geq \max_{Q} \min_{\theta} \ell(\theta, Q) = \max_{Q} \ell(\hat{\theta}_{mean}, Q) = \infty$$

Statistical Accuracy

	Coordinatewise Median	Tukey's Median
breakdown point	$1/2$	$1/3$
statistical precision (no contamination)	$\frac{p}{n}$	$\frac{p}{n}$
statistical precision (with contamination)	$\frac{p}{n} + p\epsilon^2$	$\frac{p}{n} + \epsilon^2$: minimax [Chen-Gao-Ren'15]
computational complexity	Polynomial	NP-hard [Amenta et al. '00]

Computational Complexity

- Polynomial algorithms are proposed [Diakonikolas et al.'16, Lai et al. 16] of minimax optimal statistical precision
  - needs information on second or higher order of moments
  - some priori knowledge about ϵ
- Tukey's median has a wider adaptivity,
  - does not need to know ϵ
  - does not need to know second (or higher) order of moments
  - optimal for any elliptical distribution even when moments are not defined
  - find saddle points of mini-max optimization
  - any computational facility for it?

Computational Complexity

- Polynomial algorithms are proposed [Diakonikolas et al.'16, Lai et al. 16] of minimax optimal statistical precision
  - needs information on second or higher order of moments
  - some priori knowledge about ϵ
- Tukey's median has a wider adaptivity,
  - does not need to know ϵ
  - does not need to know second (or higher) order of moments
  - optimal for any elliptical distribution even when moments are not defined
  - find saddle points of mini-max optimization
  - any computational facility for it?

Generative Adversarial Networks (GANs)!

f-GAN

Given a strictly convex function f that satisfies $f(1) = 0$, the f-divergence between two probability distributions P and Q is defined by

$$D_f(P\|Q) = \int f\left(\frac{p}{q}\right) dQ. \tag{8}$$

Let f^* be the convex conjugate of f. A variational lower bound of (8) is

$$D_f(P\|Q) \geq \sup_{T\in\mathcal{T}} \left[\mathbb{E}_P T(X) - \mathbb{E}_Q f^*(T(X))\right]. \tag{9}$$

where equality holds whenever the class $\mathcal{T}$ contains the function $f'(p/q)$.

[Nowozin-Cseke-Tomioka'16] f-GAN minimizes the variational lower bound (9)

$$\widehat{P} = \arg\min_{Q\in\mathcal{Q}} \sup_{T\in\mathcal{T}} \left[\frac{1}{n}\sum_{i=1}^{n} T(X_i) - \mathbb{E}_Q f^*(T(X))\right]. \tag{10}$$

with i.i.d. observations $X_1, ..., X_n \sim P$.

From f-GAN to Tukey's Median

Consider the special case

$$\mathcal{T} = \left\{ f'\left(\frac{\widetilde{q}}{q}\right) : \widetilde{q} \in \widetilde{\mathcal{Q}} \right\}. \tag{11}$$

which is tight if $P \in \widetilde{\mathcal{Q}}$. The sample version leads to the following f-learning

$$\widehat{P} = \arg\min_{Q \in \mathcal{Q}} \sup_{\widetilde{Q} \in \widetilde{\mathcal{Q}}} \left[\frac{1}{n} \sum_{i=1}^{n} f'\left(\frac{\widetilde{q}(X_i)}{q(X_i)}\right) - \mathbb{E}_Q f^*\left(f'\left(\frac{\widetilde{q}(X)}{q(X)}\right)\right) \right]. \tag{12}$$

- If $f(x) = x \log x$, $\mathcal{Q} = \widetilde{\mathcal{Q}}$, (12) $\Rightarrow$ Maximum Likelihood Estimate
- If $f(x) = (x-1)+$, then $D_f(P\|Q) = \frac{1}{2}\int |p-q|$ is the TV-distance, $f^*(t) = t\mathbb{I}\{0 \le t \le 1\}$, f-GAN $\Rightarrow$ TV-GAN
  - $\mathcal{Q} = \{N(\eta, I_p) : \eta \in \mathbb{R}^p\}$ and $\widetilde{\mathcal{Q}} = \{N(\widetilde{\eta}, I_p) : \|\widetilde{\eta} - \eta\| \le r\}$, (12) $\overset{r \to 0}{\Rightarrow}$ Tukey's Median

TV-GAN

For $f(x) = (x-1)+$, $f^*(t) = t\mathbb{I}\{0 \le t \le 1\}$, consider normal mean estimate

$$\widehat{\theta} = \arg\min_{\eta \in \mathbb{R}^p} \max_{D \in \mathcal{D}} \left[\frac{1}{n} \sum_{i=1}^{n} D(X_i) - \mathbb{E}_{\mathcal{N}(\eta, I_p)} D(X) \right]. \tag{13}$$

with the logistic regression for Discriminator,

$$\mathcal{D} = \left\{ D(x) = \text{sigmoid}(w^T x + b) : w \in \mathbb{R}^p, b \in \mathbb{R} \right\}. \tag{14}$$

Theorem (Gao-Liu-Y.-Zhu'18)

There exists some $C > 0$,

$$\|\widehat{\theta} - \theta\|_2^2 \le C \left(\frac{p}{n} \vee \epsilon^2 \right)$$

with high probability uniformly over $\theta \in \mathbb{R}^p$ and Q.

The Landscape of TV-GAN can be Bad!

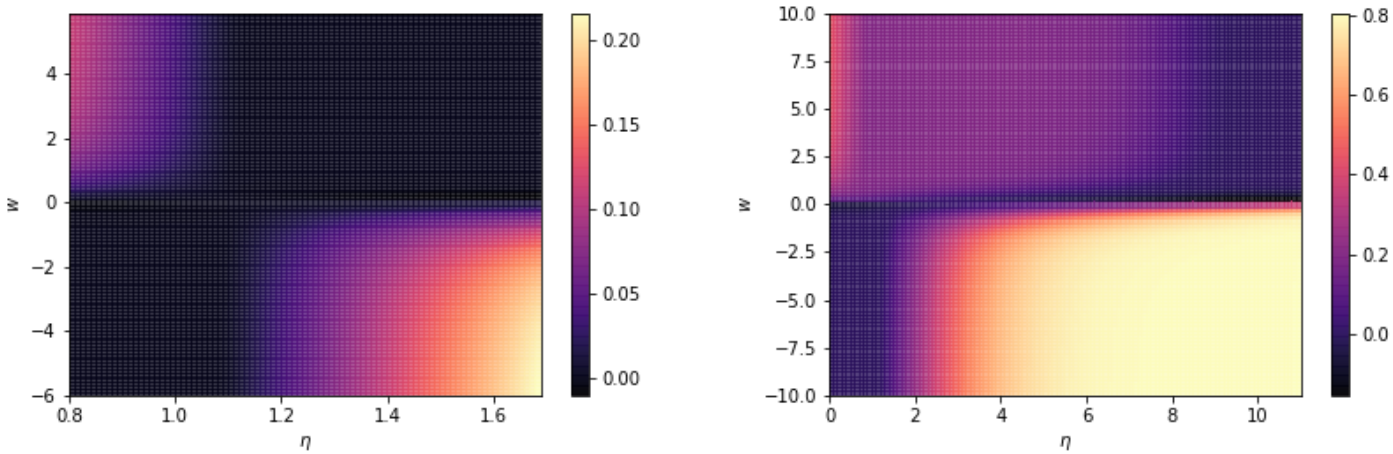


Figure: Heatmaps of the landscape of $F(\eta, w) = \sup_b [E_P \text{sigmoid}(wX + b) - E_{N(\eta,1)} \text{sigmoid}(wX + b)]$, where b is maximized out for visualization. Left: samples are drawn from $P = (1 - \epsilon)N(1, 1) + \epsilon N(1.5, 1)$ with $\epsilon = 0.2$. Right: samples are drawn from $P = (1 - \epsilon)N(1, 1) + \epsilon N(10, 1)$ with $\epsilon = 0.2$. Left: the landscape is good in the sense that no matter whether we start from the left-top area or the right-bottom area of the heatmap, gradient ascent on η does not consistently increase or decrease the value of η. This is because the signal becomes weak when it is close to the saddle point around $\eta = 1$. Right: it is clear that $\tilde{F}(w) = F(\eta, w)$ has two local maxima for a given η, achieved at $w = +\infty$ and $w = -\infty$. In fact, the global maximum for $\tilde{F}(w)$ has a phase transition from $w = +\infty$ to $w = -\infty$ as η grows. For example, the maximum is achieved at $w = +\infty$ when $\eta = 1$ (blue solid) and is achieved at $w = -\infty$ when $\eta = 5$ (red solid). Unfortunately, even if we initialize with $\eta_0 = 1$ and $w_0 > 0$, gradient ascents on η will only increase the value of η (green dash), and thus as long as the discriminator cannot reach the global maximizer, w will be stuck in the positive half space $\{w : w > 0\}$ and further increase the value of η.

The original Jenson-Shannon GAN

[Goodfellow et al. 2014] For $f(x) = x \log x - (x+1) \log \frac{x+1}{2}$,

$$\widehat{\theta} = \arg\min_{\eta \in \mathbb{R}^p} \max_{D \in \mathcal{D}} \left[\frac{1}{n} \sum_{i=1}^{n} \log D(X_i) + \mathbb{E}_{\mathcal{N}(\eta, I_p)} \log(1 - D(X)) \right] + \log 4. \quad (15)$$

What are $\mathcal{D}$, the class of discriminators?

- Single layer (no hidden layer):

$$\mathcal{D} = \left\{ D(x) = \text{sigmoid}(w^T x + b) : w \in \mathbb{R}^p, b \in \mathbb{R} \right\}$$

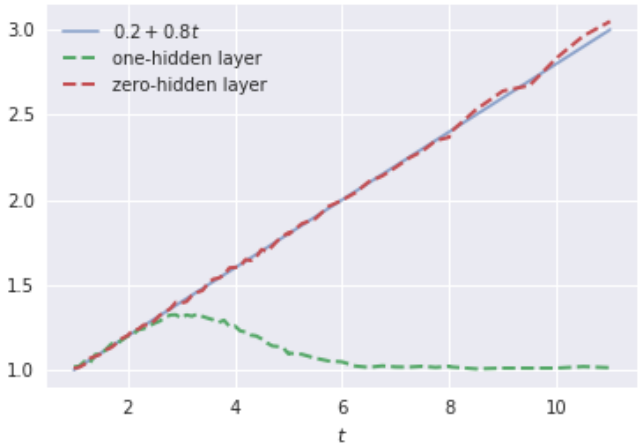
- One-hidden or Multiple layer:

$$\mathcal{D} = \left\{ D(x) = \text{sigmoid}(w^T g(X)) \right\}$$

Multiple Layer JS-GAN might Work

$$P \sim (1-\epsilon)\mathcal{N}(\theta, I_p) + \epsilon\mathcal{N}(t, I_p)$$

- For single layer (no hidden layer):
  - $\widehat{\theta} \approx (1-\epsilon)\theta + t$, not robust...
- For one-hidden or Multiple layer:
  - $\widehat{\theta} \approx \theta$ robust!



JS-GAN as feature (generalized moment) matching

$$\mathrm{JS}_g(P, Q) = \max_{w \in \mathcal{W}} \left[E_P \log \mathrm{sigmoid}(w^T g(X)) + \mathbb{E}_Q \log(1 - \mathrm{sigmoid}(w^T g(X))) \right] + \log 4.$$

Proposition (Gao-Liu-Y.-Zhu'18)

Assume $\mathcal{W}$ is a convex set that contains an open neighborhood of 0. Then,

$$\mathrm{JS}_g(P, Q) = 0 \Leftrightarrow \mathbb{E}_P g(X) = \mathbb{E}_Q g(X).$$

- For $g = Id$, first moment matching gives sample mean estimate
- For g being indicator of half spaces, it may lead to Tukey median estimate
- *How about feature maps in neural networks?*

Neural Network Features

$$\widehat{\theta} = \arg\min_{\eta} \max_{D \in \mathcal{D}} \left[E_P \log D(X) + \mathbb{E}_{Q=\mathcal{N}(\eta, I_p)} \log(1 - D(X)) \right] + \log 4.$$

Theorem (Gao-Liu-Y.-Zhu'18)

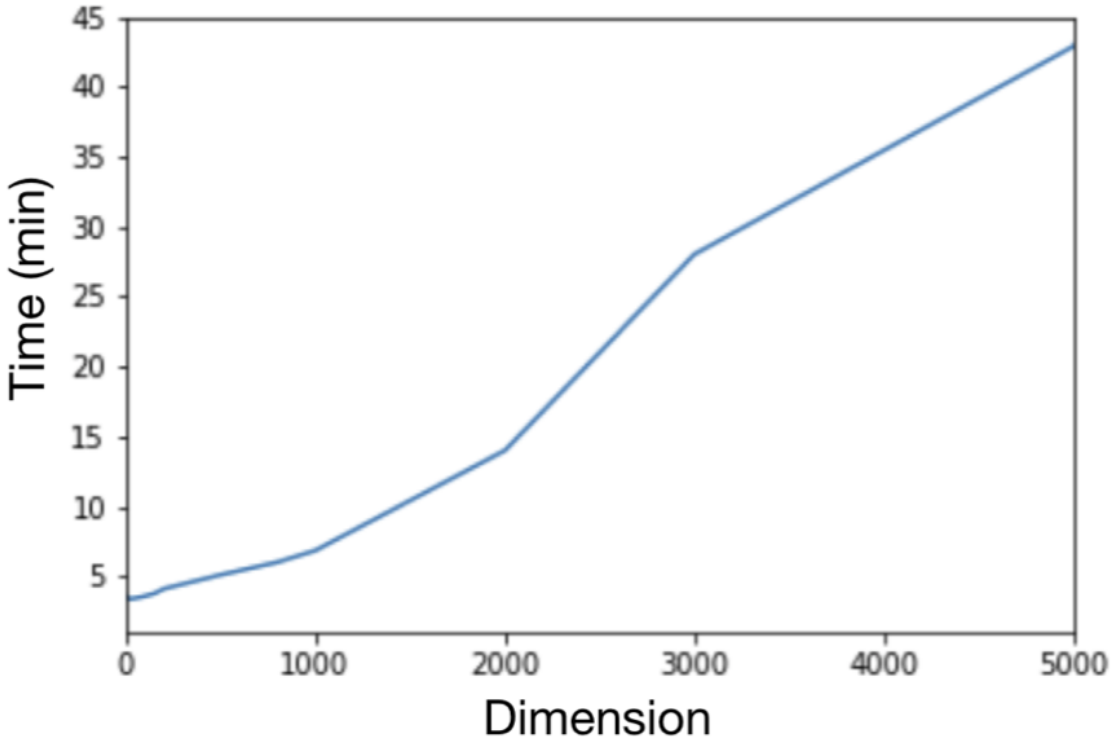
With proper neural network $\mathcal{D}$ of at least one hidden layer and regularization on weight norms,

$$\|\widehat{\theta} - \theta\|^2 \lesssim \begin{cases} \frac{p}{n} \vee \epsilon^2, & \text{bounded activations} \\ \frac{p \log p}{n} \vee \epsilon^2, & \text{ReLU+last sigmoid} \end{cases}$$

with high probability uniformly over all $\theta \in \mathbb{R}^p$ and all Q.

Experiments: run time of JS-GAN

- initialization: coordinatewise median
- run time with single GPU (1080Ti)



Experiments: Comparisons

Q	n	p	ϵ	TV-GAN	JS-GAN	Dimension Halving	Iterative Filtering
$N(0.5 * 1_p, I_p)$	50,000	100	.2	**0.0953 (0.0064)**	0.1144 (0.0154)	0.3247 (0.0058)	0.1472 (0.0071)
$N(0.5 * 1_p, I_p)$	5,000	100	.2	**0.1941 (0.0173)**	0.2182 (0.0527)	0.3568 (0.0197)	0.2285 (0.0103)
$N(0.5 * 1_p, I_p)$	50,000	200	.2	**0.1108 (0.0093)**	0.1573 (0.0815)	0.3251 (0.0078)	0.1525 (0.0045)
$N(0.5 * 1_p, I_p)$	50,000	100	.05	0.0913 (0.0527)	0.1390 (0.0050)	0.0814 (0.0056)	**0.0530 (0.0052)**
$N(5 * 1_p, I_p)$	50,000	100	.2	2.7721 (0.1285)	**0.0534 (0.0041)**	0.3229 (0.0087)	0.1471 (0.0059)
$N(0.5 * 1_p, \Sigma)$	50,000	100	.2	0.1189 (0.0195)	**0.1148 (0.0234)**	0.3241 (0.0088)	0.1426 (0.0113)
Cauchy$(0.5 * 1_p)$	50,000	100	.2	0.0738 (0.0053)	**0.0525 (0.0029)**	0.1045 (0.0071)	0.0633 (0.0042)

Table: Comparison of various robust mean estimation methods. The smallest error of each case is highlighted in bold.

- *Dimension Halving:* [Lai et al.'16]
  `https://github.com/kal2000/AgnosticMeanAndCovarianceCode`.
- *Iterative Filtering:* [Diakonikolas et al.'17]
  `https://github.com/hoonose/robust-filter`.

Experiments: the error rates

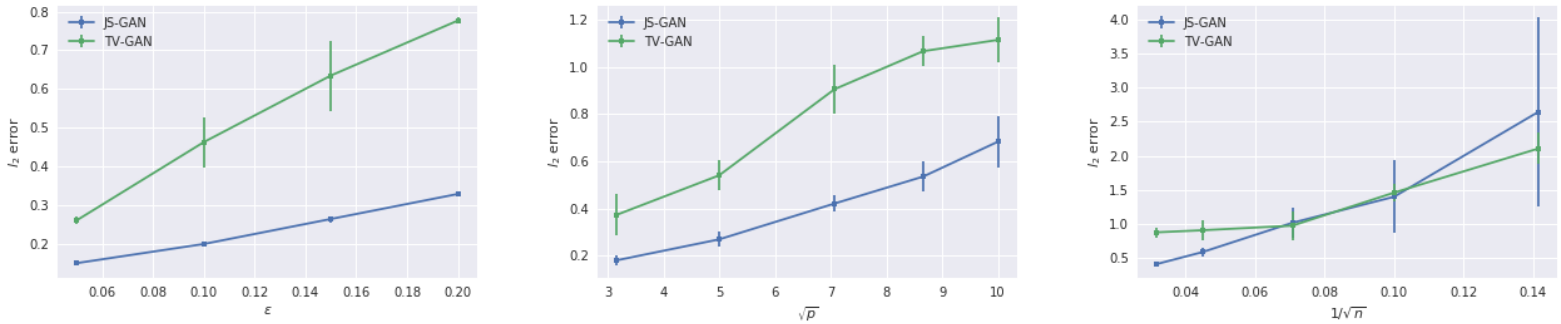


Figure: ℓ_2 error $\|\widehat{\theta} - \theta\|$ against ϵ (left), $\sqrt{p}$ (middle) and $1/\sqrt{n}$ (right), respectively. The vertical bars indicate $\pm$ standard deviations. In all cases, the errors are approximately linear with respect to the corresponding numbers, which empirically verifies the theoretical rates.

Summary

- Robust estimate via GANs under Huber's Agnostic Contamination model
  - Both TV-GAN and JS-GAN may provably achieve statistical optimality under Huber's model
  - Yet the landscape of TV-GAN might be hard to find the desired saddle point
  - The landscape of JS-GAN is easier to handle in practice
- Future directions
  - provable robust GANs for co-variance and regression, etc., in high dimensional statistics
  - *does it lead to an alternative approach against adversarial samples in neural networks?* If machines can discriminate sparse *human imperceivable* abnormal samples from normal ones, then robust (invariant) estimates set in

Some Reference

- Zhu, Huang, and Yao, "On Breiman's Dilemma in Neural Networks: Phase Transitions of Margin Dynamics", arXiv:1810.03389.
- Gao, Liu, Yao, and Zhu, "Robust Estimation and Generative Adversarial Networks", arXiv:1810.02030.

The END


thank you
danke
謝謝
gracias
merci
obrigado
dziękuję
bedankt
go raibh maith agat
teşekkür ederim
спасибо
ngiyabonga
dank je
mochchakkeram
sukriya
kop khun krap
terima kasih
grazie
arigatō
dakujem
tapadh leat
mauruuru
hvala
sagolun
takk
merci
мерси
рахмат
Баярлалаа
vinaka
faafetai lava
kiitos
dankie
dhanyavad
nandri
nanni
köszönöm
bayarlalaa
gràcies
enkosi
chnorakaloutioun
gratias ago
gràcies
akun
tankon
aciū
sulpáy
djiere dieuf
tau
дякую
mamnun
chokrane
murakoze
teiki
obrigada
asante
manana
хвала
paldies
grazzi
misaotra
matondo
mahalo
mersi
barka
welalin
tack
ora
ida
spas
blagodaram
спасибі
sobodi
dēkuji
mësi
didi madloba
najis tuke
kam sah hamnida
rahmat
ありがとう
tanemirt
rahmet
diolch
dhanyavadagalu
shukriya
trugarez
mercè
ευχαριστώ
xièxie
감사합니다
তোমাকে ধন্যবাদ