[DLT 3-4]
Optimization and generalization of deep networks: a *highly biased* margin perspective

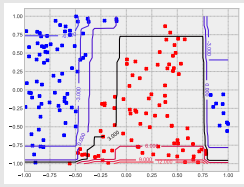
Matus Telgarsky <mjt@illinois.edu>

(with help from many friends!)

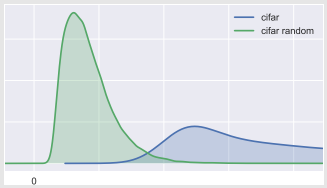
(Thanks for inviting me to Russia!)

Scope of these 4 lectures:

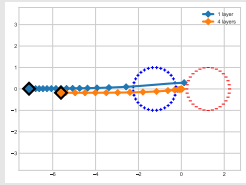
- 1-2: approximation theory.
- 3-4: margin perspective on optimization and generalization.
  This part is highly biased!



Margins.



Generalization.



Optimization.

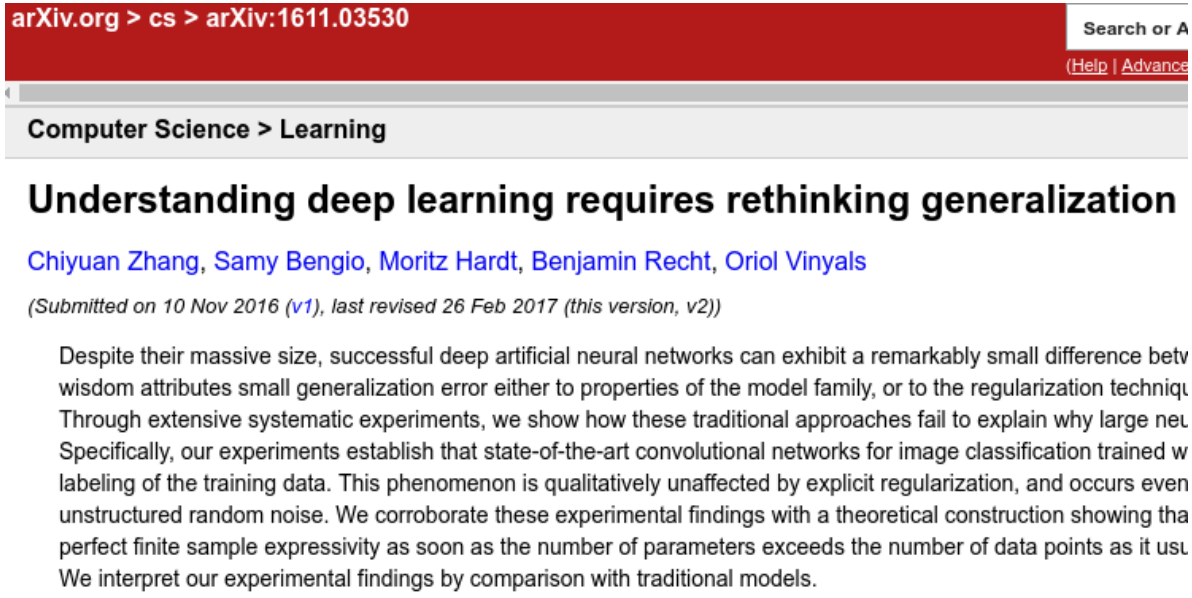
Margins?

Margins?

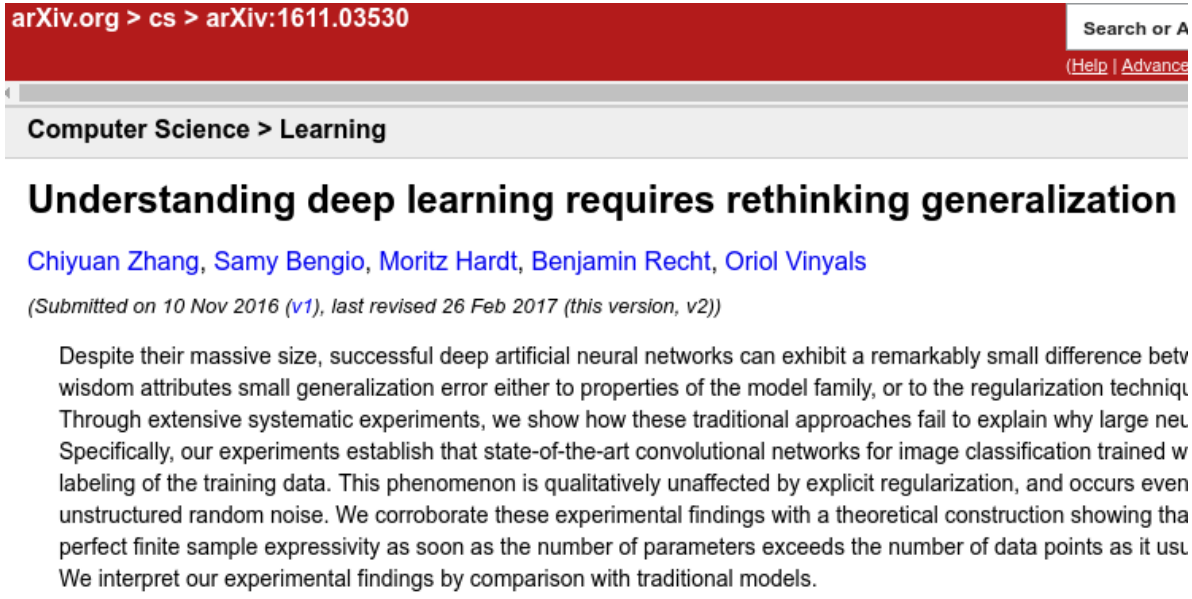
- Margins are an existing theory for generalization/optimization;
  we can apply this theory to deep learning.
- There are other ways to study optimization generalization.
- As compared with yesterday, today will be **highly biased**, and include much work I am involved with.

(Some backstory from 2017...)

(Some backstory from 2017...)



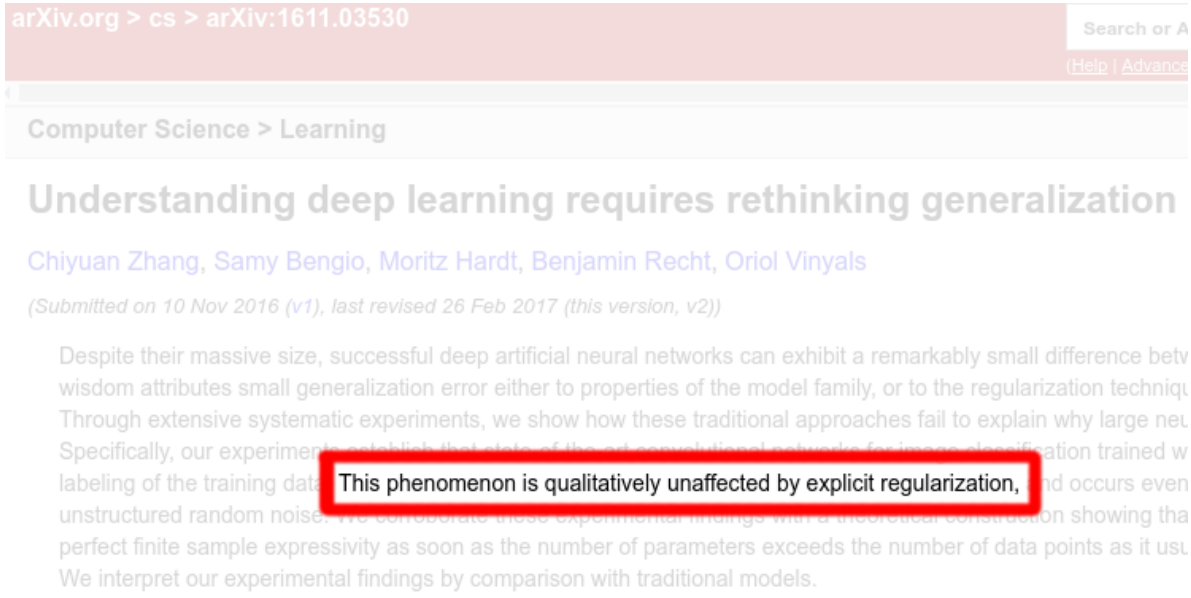
(Some backstory from 2017...)



My immediate feelings:

- Wow! *Always* get 0 training error. (Even random labels!)
- Still generalizes.

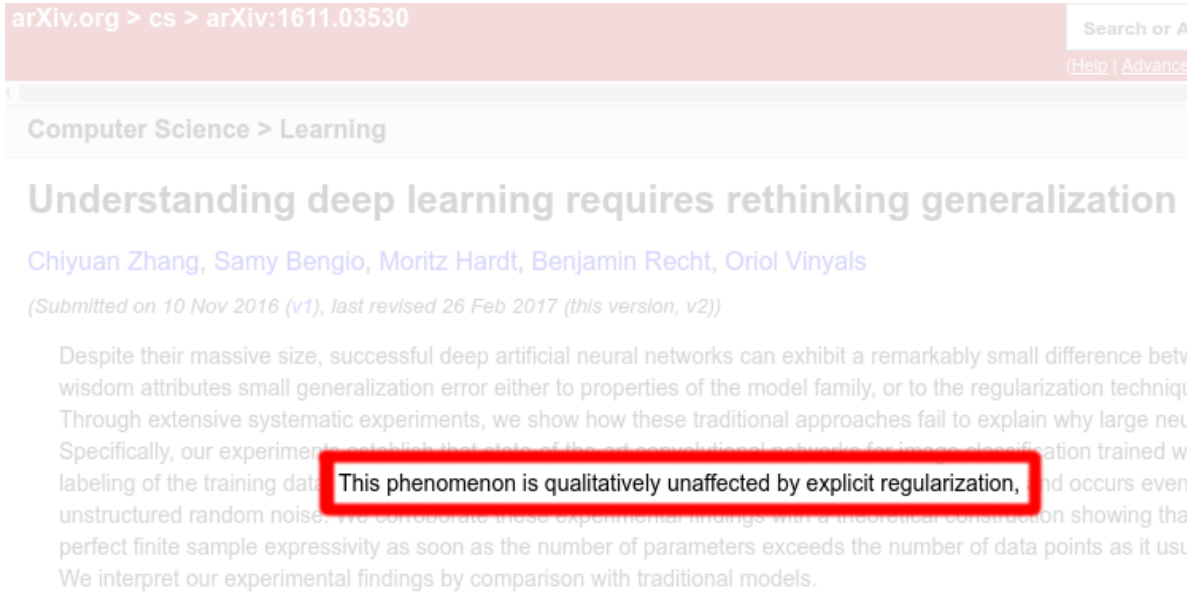
(Some backstory from 2017...)



My immediate feelings:

- Wow! *Always* get 0 training error. (Even random labels!)
- Still generalizes.

(Some backstory from 2017...)



My immediate feelings:

- Wow! *Always* get 0 training error. (Even random labels!)
- Still generalizes.
- **Explicit regularization not needed;**
  sounds like boosting and margin theory!
- **Remark:** there are other ways to reason about this puzzle!

. . . so what are margins?

. . . so what are margins?

Standard classifiers discretize real-valued predictions:

$$x \mapsto \operatorname{sgn}(f(x)), \qquad \text{or} \qquad x \mapsto \arg\max_j f(x)_j.$$

Unnormalized margin measures "how correct/confident":

$$f(x)y, \qquad \text{or} \qquad f(x)_y - \arg\max_{j \neq y} f(x)_j.$$

. . . so what are margins?

Standard classifiers discretize real-valued predictions:

$$x \mapsto \operatorname{sgn}(f(x)), \qquad \text{or} \qquad x \mapsto \arg\max_j f(x)_j.$$

Unnormalized margin measures "how correct/confident":

$$f(x)y, \qquad \text{or} \qquad f(x)_y - \arg\max_{j \neq y} f(x)_j.$$

Margins require normalization to be meaningful;
otherwise can scale by $c > 0$ and "improve".

. . . so what are margins?

Standard classifiers discretize real-valued predictions:

$$x \mapsto \operatorname{sgn}(f(x)), \qquad \text{or} \qquad x \mapsto \arg\max_j f(x)_j.$$

Unnormalized margin measures "how correct/confident":

$$f(x)y, \qquad \text{or} \qquad f(x)_y - \arg\max_{j \neq y} f(x)_j.$$

Margins require normalization to be meaningful;
otherwise can scale by $c > 0$ and "improve".

History:

- Margins & optimization: Novikoff '62.
- Margins & generalization: Bartlett '96
  "For valid generalization the size of the weights is more important than the size of the network".
- Margins prominently used as an explanation of boosting.

Margins, boosting, and deep networks.

- Standard boosting methods can guarantee 0 training error.
  (Empirically observed with deep networks!)
- Standard boosting methods select large margin predictors.
  (Is this empirically observed with deep networks? Can we prove/disprove it?)
- Boosting methods have margin-based generalization bounds.
  (How about deep networks?)

Boosting overview.

- Greedily select classifiers $(h_i)_{i=1}^t$ and weights $w \in \mathbb{R}^t$ and output predictor $f_t(x) := \sum_j w_j h_j(x)$.
- Define margin $\min_i \frac{y_i f_t(x_i)}{\sum_j |w_j|}$.
- **Remarks.**
  - Standard method is **coordinate descent** on some risk; *not explicitly maximizing margins.*
  - $\sum_j |w_j| = \|w\|_1$ is due to coordinate descent; can prove theorems with this choice.

Large margin boosted classifiers generalize.

Theorem (Schapire-Freund-Bartlett-Lee, 1998).
Fix classifiers $\mathcal{H}$ with VC dimension $\text{VC}(\mathcal{H})$.
Given margin γ, with probability $\geq 1 - \delta$ over data $((x_i, y_i))^n$,
for any weights $w \in \mathbb{R}^t$ over $h_1, \ldots, h_t \in \mathcal{H}$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\frac{1}{\gamma}\sqrt{\frac{\text{VC}(\mathcal{H})\ln(n) + \ln(1/\delta)}{n}}\right).$$

Large margin boosted classifiers generalize.

Theorem (Schapire-Freund-Bartlett-Lee, 1998).
Fix classifiers $\mathcal{H}$ with VC dimension $\text{VC}(\mathcal{H})$.
Given margin γ, with probability $\geq 1-\delta$ over data $((x_i, y_i))^n$,
for any weights $w \in \mathbb{R}^t$ over $h_1, \ldots, h_t \in \mathcal{H}$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\frac{1}{\gamma}\sqrt{\frac{\text{VC}(\mathcal{H})\ln(n) + \ln(1/\delta)}{n}}\right).$$

Remarks.

- Interpretation: treating $\|w\|_1$ as "complexity"
  ($\|w\|_1$ arises in $\widehat{\Pr}\left[\gamma > {}^{Yf_w(X)}/_{\|w\|_1}\right]$),
  this says low complexity boosted classifiers will generalize.

Large margin boosted classifiers generalize.

Theorem (Schapire-Freund-Bartlett-Lee, 1998).
Fix classifiers $\mathcal{H}$ with VC dimension $\text{VC}(\mathcal{H})$.
Given margin γ, with probability $\geq 1-\delta$ over data $((x_i, y_i))^n$,
for any weights $w \in \mathbb{R}^t$ over $h_1, \ldots, h_t \in \mathcal{H}$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\frac{1}{\gamma}\sqrt{\frac{\text{VC}(\mathcal{H})\ln(n) + \ln(1/\delta)}{n}}\right).$$

Remarks.

- Interpretation: treating $\|w\|_1$ as "complexity"
  ($\|w\|_1$ arises in $\widehat{\Pr}\left[\gamma > {}^{Yf_w(X)}/_{\|w\|_1}\right]$),
  this says low complexity boosted classifiers will generalize.
- With random data, $\|w\|_1$ will be large and kill the bound.
- It is okay if $|\mathcal{H}| = \infty$!

Large margin boosted classifiers generalize.

Theorem (Schapire-Freund-Bartlett-Lee, 1998).
Fix classifiers $\mathcal{H}$ with VC dimension $\text{VC}(\mathcal{H})$.
Given margin γ, with probability $\geq 1-\delta$ over data $((x_i, y_i))^n$,
for any weights $w \in \mathbb{R}^t$ over $h_1, \ldots, h_t \in \mathcal{H}$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\frac{1}{\gamma}\sqrt{\frac{\text{VC}(\mathcal{H})\ln(n) + \ln(1/\delta)}{n}}\right).$$

Remarks.

- Interpretation: treating $\|w\|_1$ as "complexity"
  ($\|w\|_1$ arises in $\widehat{\Pr}\left[\gamma > {}^{Yf_w(X)}/_{\|w\|_1}\right]$),
  this says low complexity boosted classifiers will generalize.
- With random data, $\|w\|_1$ will be large and kill the bound.
- It is okay if $|\mathcal{H}| = \infty$!
- Story seems consistent with deep networks:
  can have big boosted classifier (big network),
  zero training error, and still generalize (if not random!).

Boosting methods find large margin boosted classifiers.

Theorem (T '13).
Let $\bar{\gamma}$ denote the maximum possible margin.
After $8\ln(n)/\bar{\gamma}^2$ iterations,
AdaBoost finds a classifier with margin $\bar{\gamma}/4$.

Boosting methods find large margin boosted classifiers.

Theorem (T '13).
Let $\bar{\gamma}$ denote the maximum possible margin.
After $8\ln(n)/\bar{\gamma}^2$ iterations,
AdaBoost finds a classifier with margin $\bar{\gamma}/4$.

Remarks.

- **Bias warning:** so I've been working on margins forever...

Boosting methods find large margin boosted classifiers.

Theorem (T '13).
Let $\bar{\gamma}$ denote the maximum possible margin.
After $8\ln(n)/\bar{\gamma}^2$ iterations,
AdaBoost finds a classifier with margin $\bar{\gamma}/4$.

Remarks.

- **Bias warning:** so I've been working on margins forever...
- **Proof.** Adaboost uses exponential loss;
  thus $\ln \sum$ loss is like unnormalized minimum margin.
  Can then directly show it grows at least like $\bar{\gamma}\|w\|_1$.

Boosting methods find large margin boosted classifiers.

Theorem (T '13).
Let $\bar{\gamma}$ denote the maximum possible margin.
After $8\ln(n)/\bar{\gamma}^2$ iterations,
AdaBoost finds a classifier with margin $\bar{\gamma}/4$.

Remarks.

- **Bias warning:** so I've been working on margins forever...
- **Proof.** Adaboost uses exponential loss;
  thus $\ln \sum$ loss is like unnormalized minimum margin.
  Can then directly show it grows at least like $\bar{\gamma}\|w\|_1$.
- Together with previous slide, get low test error for boosting.
- A long history!
  - Novikoff '62: perceptron convergence rate via margins.
  - Schapire-Freund-Bartlett-Lee '98: asymptotic convergence to $\gamma/2$ margins (largest posible).
  - Zhang-Yu '05: more careful step sizes gives margin maximization.

Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot CDF of

$$(x_i, y_i) \mapsto \frac{f(x_i) y_i}{C_f}.$$

This captures margins of whole training set.

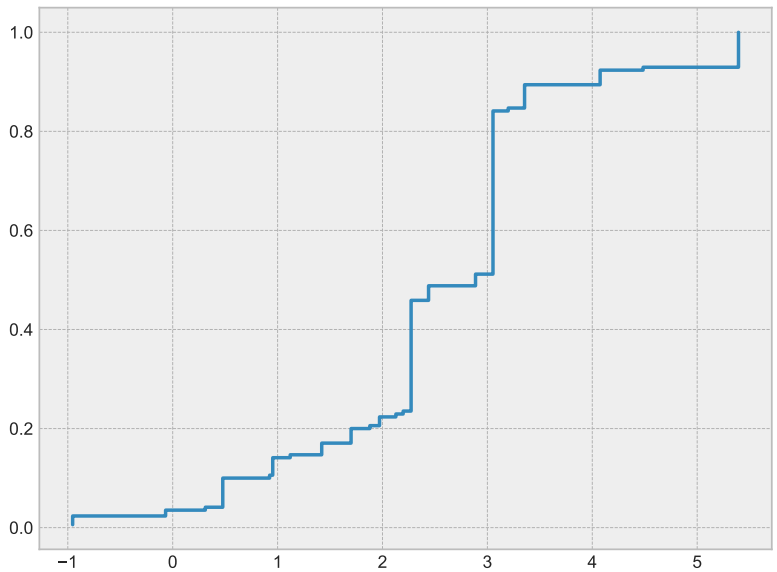
Margin plots

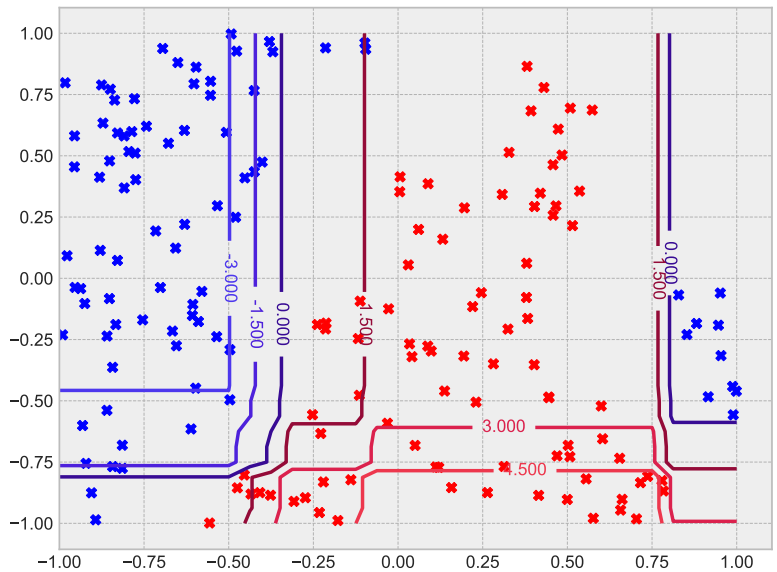
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot CDF of

$$(x_i, y_i) \mapsto \frac{f(x_i)y_i}{C_f}.$$

This captures margins of whole training set.

Example: boosted decision stumps.





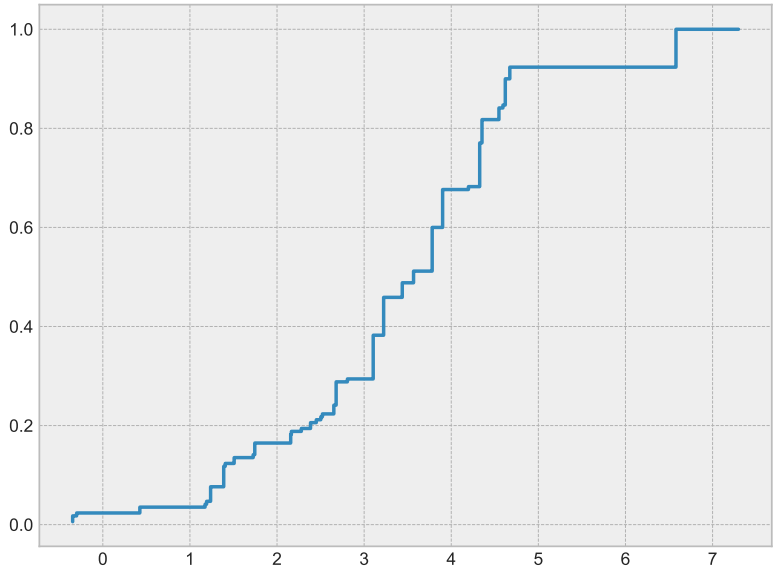
Margin plots

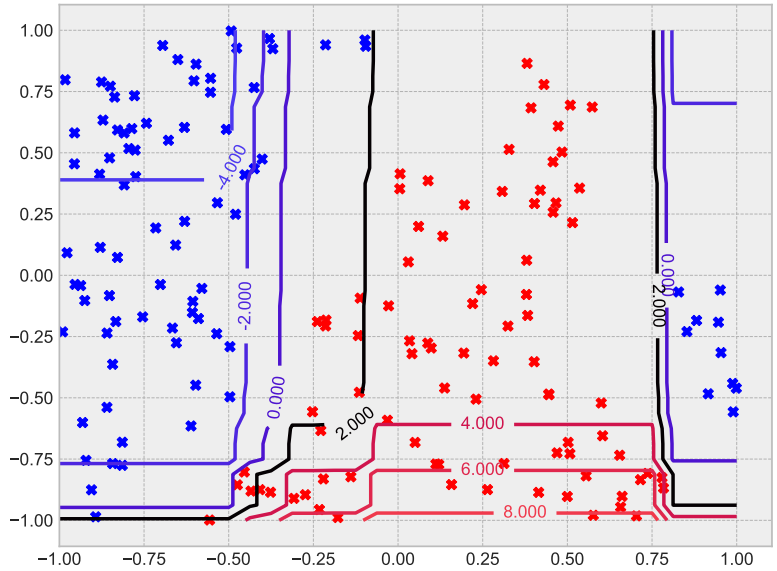
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot CDF of

$$(x_i, y_i) \mapsto \frac{f(x_i)y_i}{C_f}.$$

This captures margins of whole training set.

Example: boosted decision stumps.





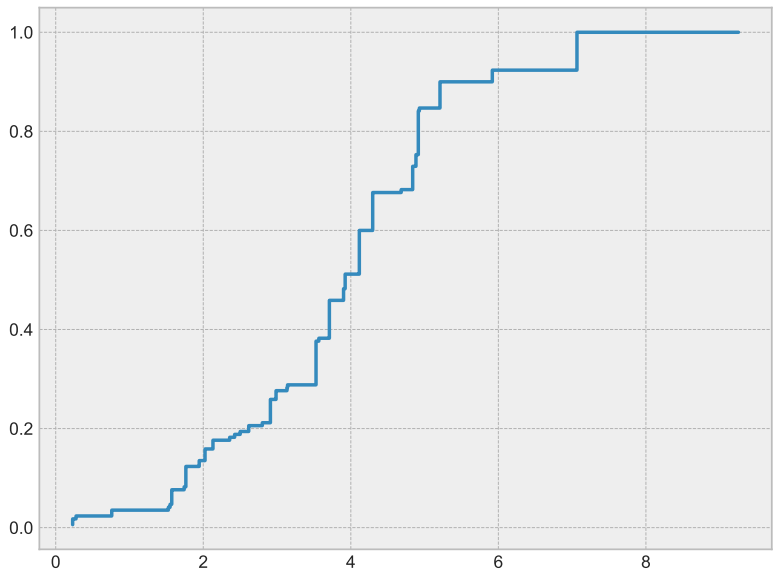
Margin plots

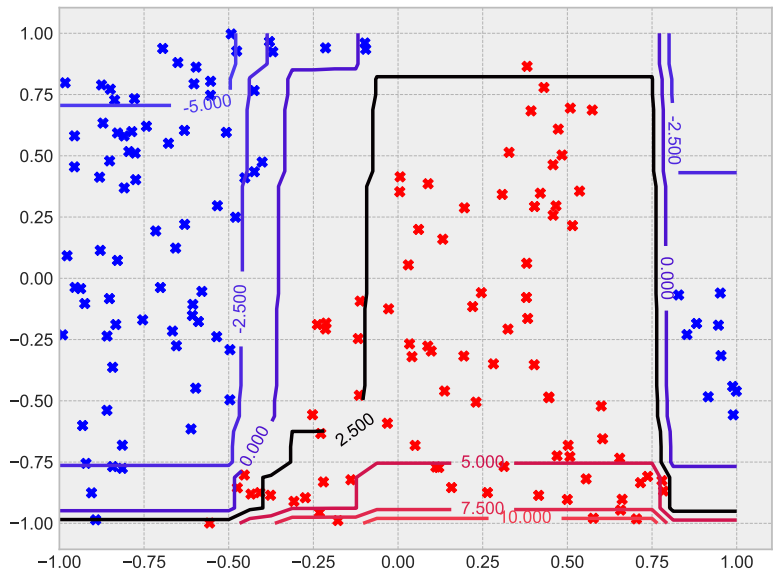
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot CDF of

$$(x_i, y_i) \mapsto \frac{f(x_i)y_i}{C_f}.$$

This captures margins of whole training set.

Example: boosted decision stumps.





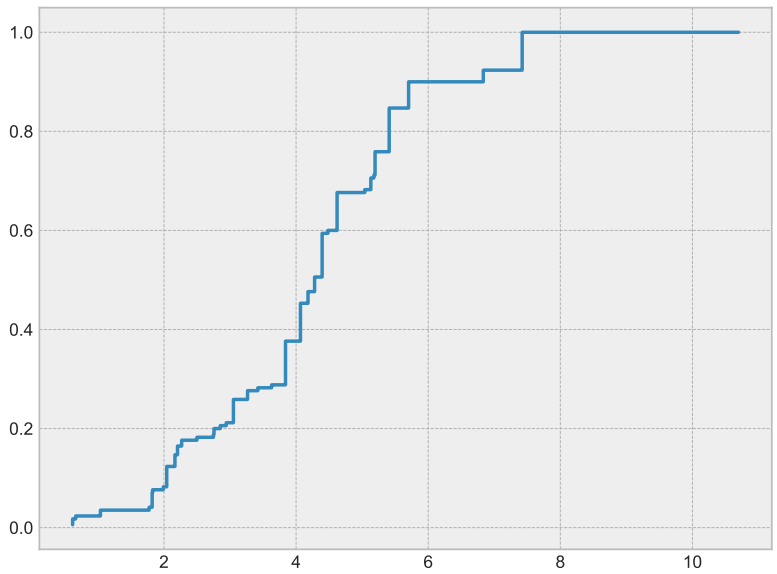
Margin plots

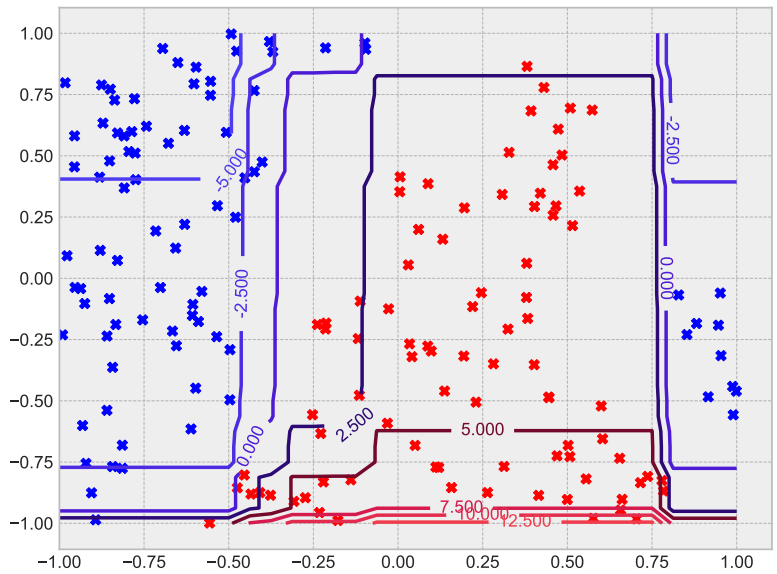
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot CDF of

$$(x_i, y_i) \mapsto \frac{f(x_i)y_i}{C_f}.$$

This captures margins of whole training set.

Example: boosted decision stumps.





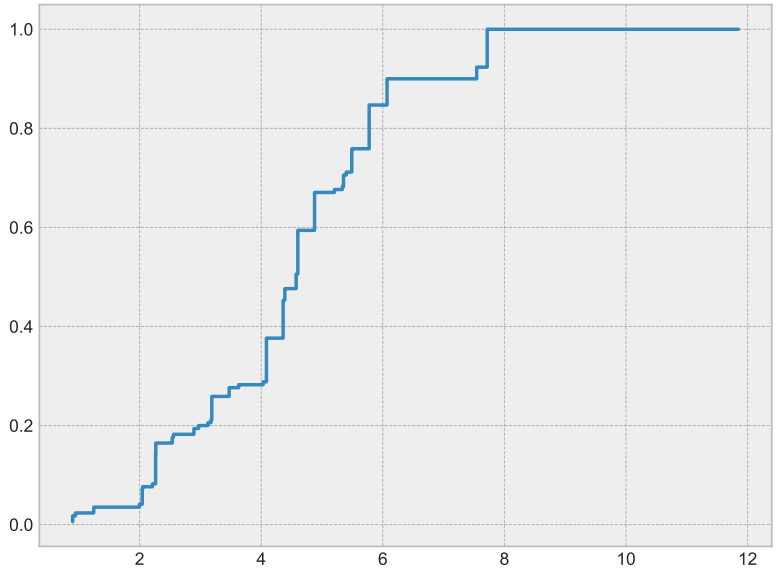
Margin plots

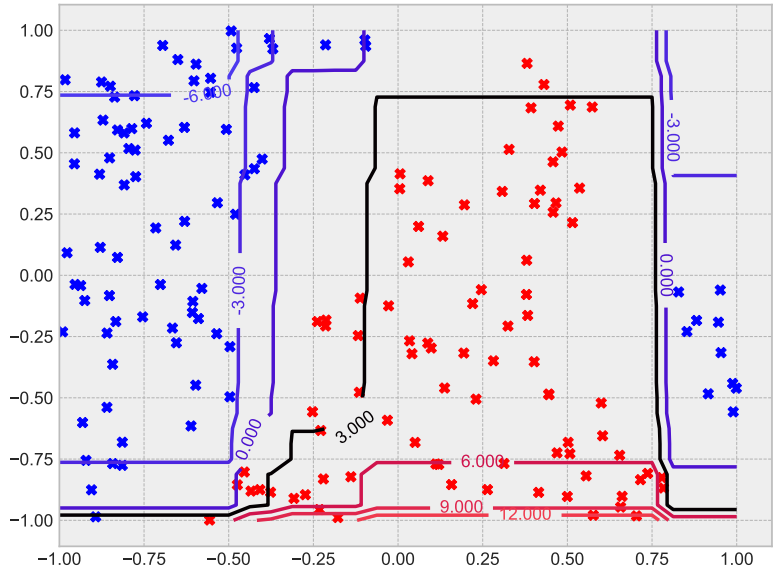
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot CDF of

$$(x_i, y_i) \mapsto \frac{f(x_i)y_i}{C_f}.$$

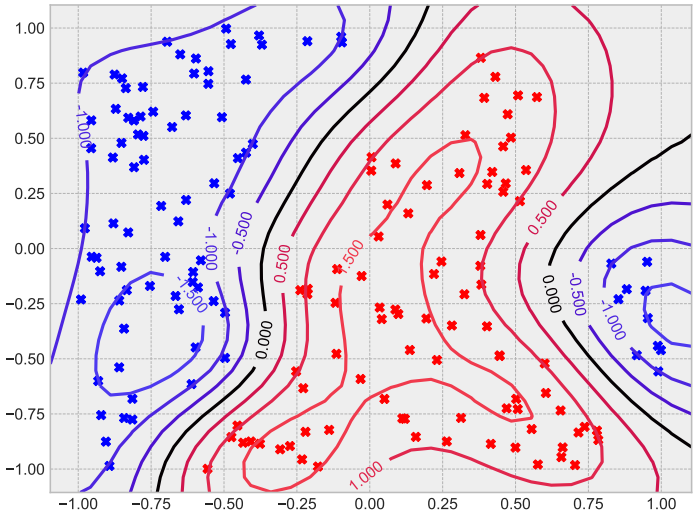
This captures margins of whole training set.

Example: boosted decision stumps.

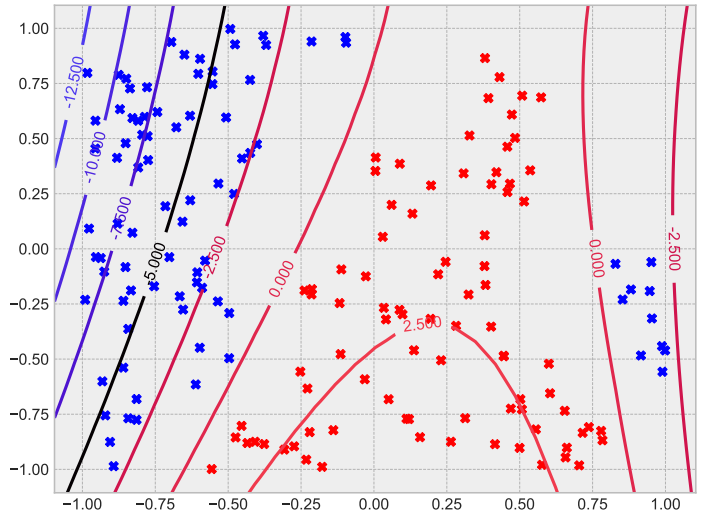




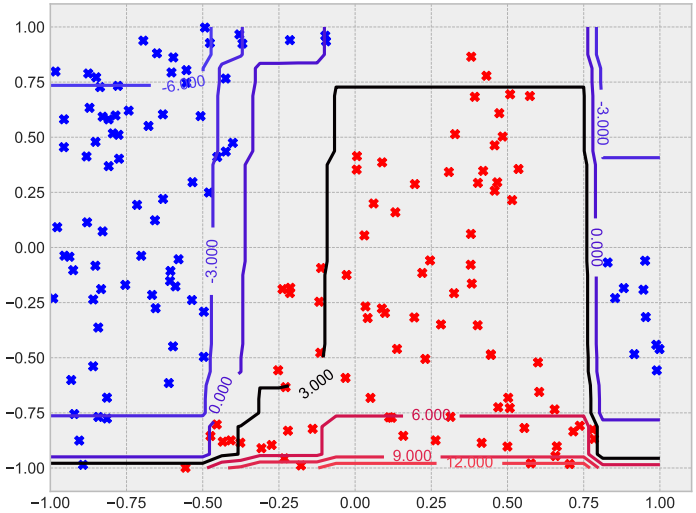
Margins appear throughout machine learning.



RBF SVM.



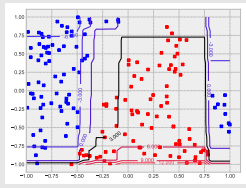
Quadratic SVM.



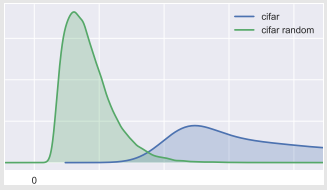
Boosted stumps.

?

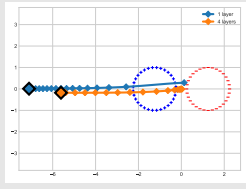
ReLU network.



Margins.

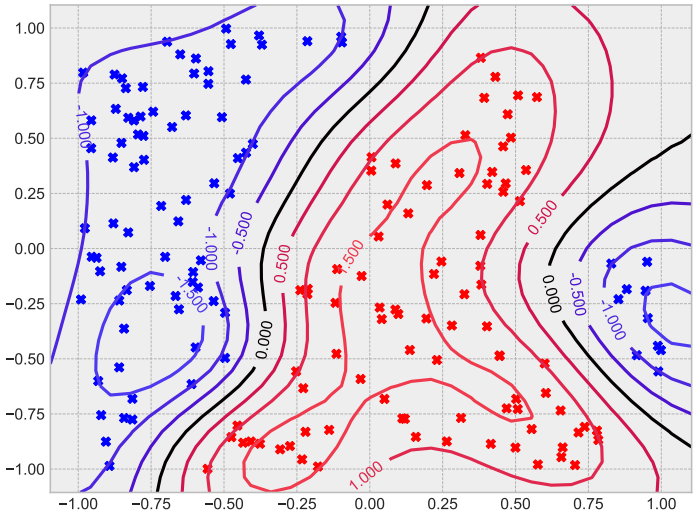


Generalization.

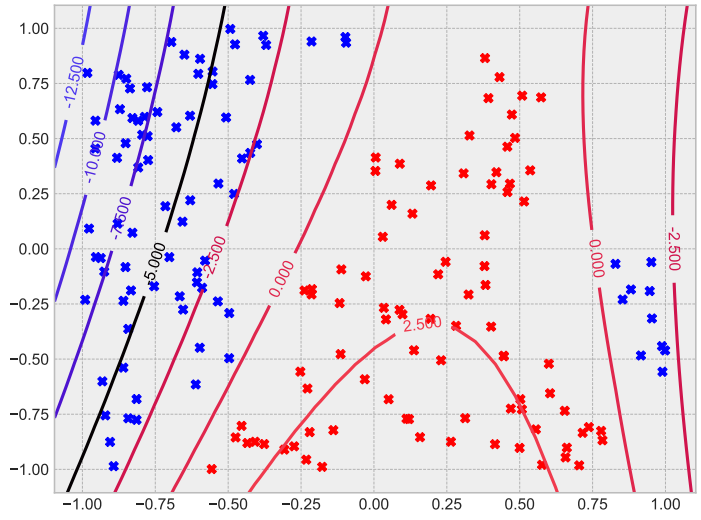


Optimization.

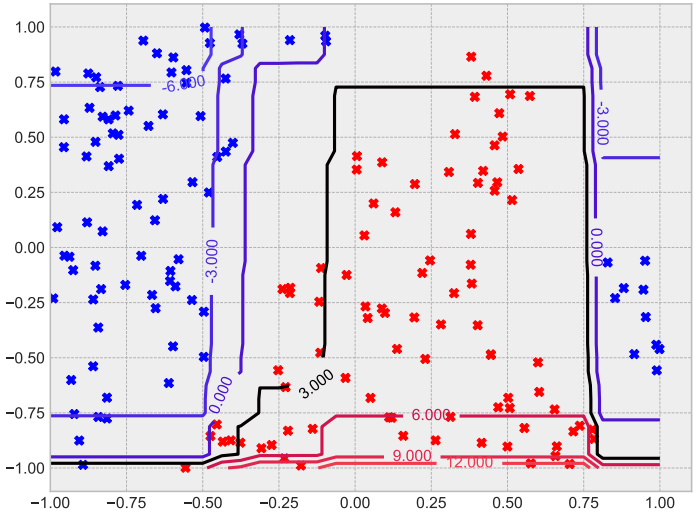
Margins appear throughout machine learning.



RBF SVM.



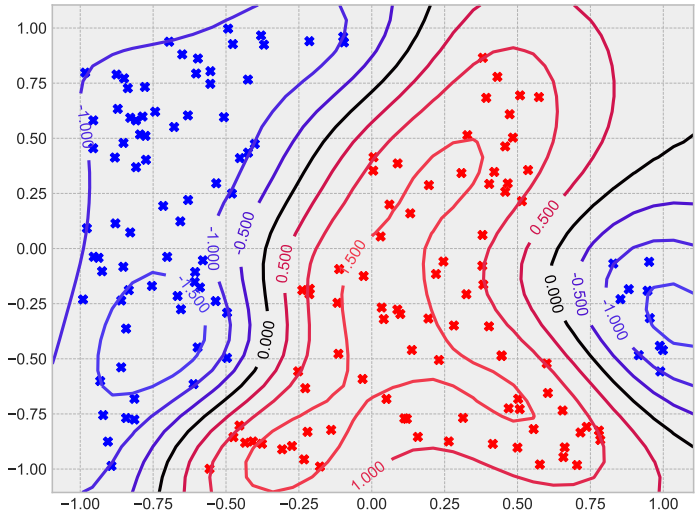
Quadratic SVM.



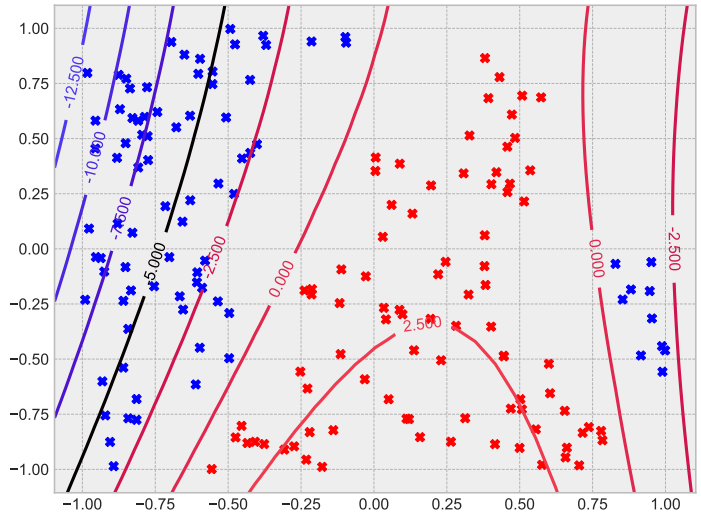
Boosted stumps.

ReLU network.

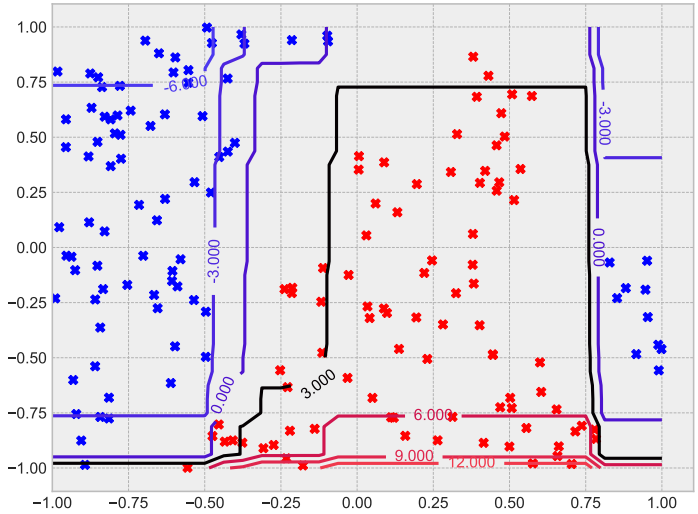
Margins appear throughout machine learning.



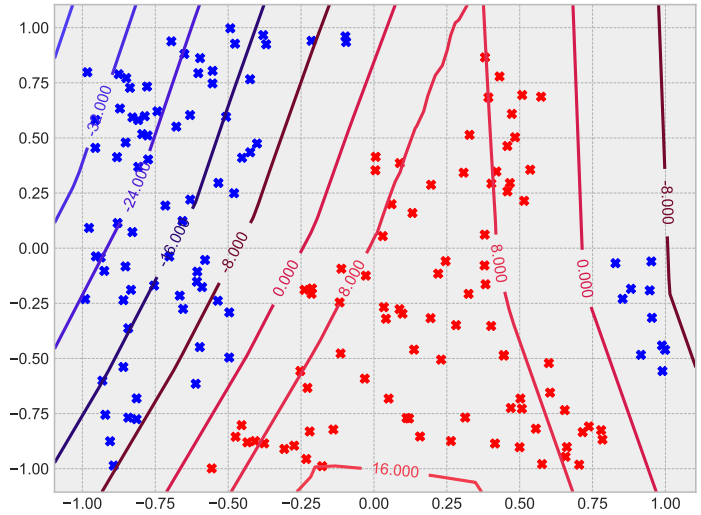
RBF SVM.



Quadratic SVM.



Boosted stumps.



ReLU network.

Margin plots

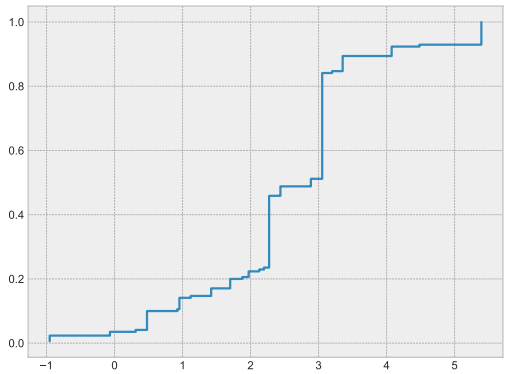
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

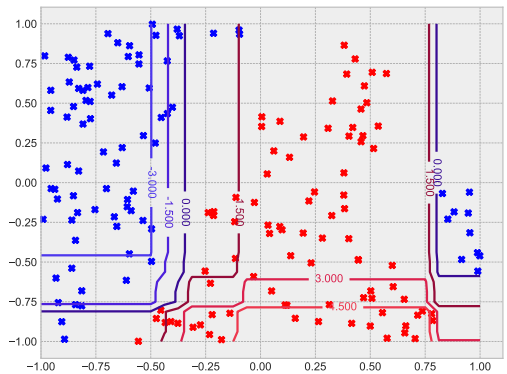
$$\frac{f(x_i)y_i}{C_f} \qquad \text{or} \qquad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$

Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



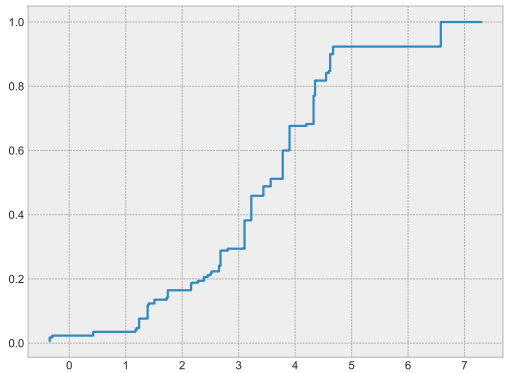


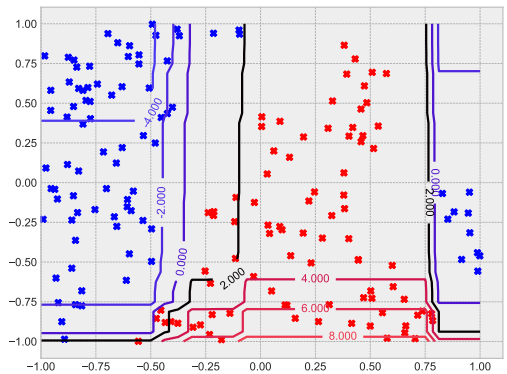
Boosted stumps.
($\mathcal{O}(n)$ param.)

Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



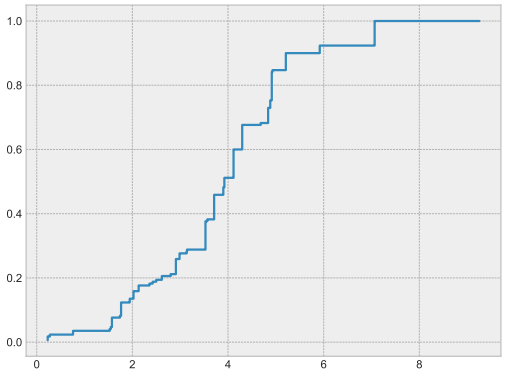


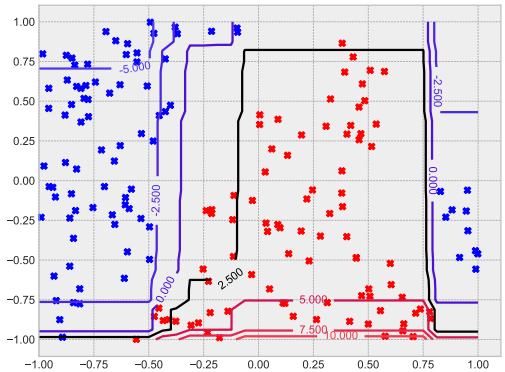
Boosted stumps.
($\mathcal{O}(n)$ param.)

Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



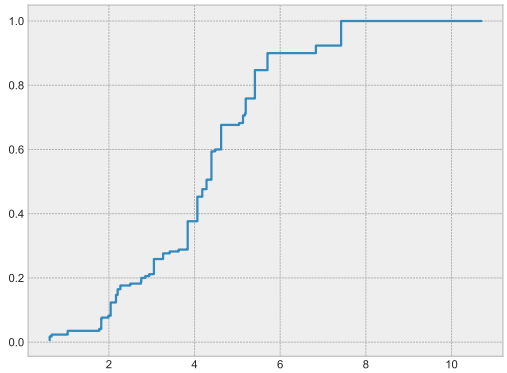


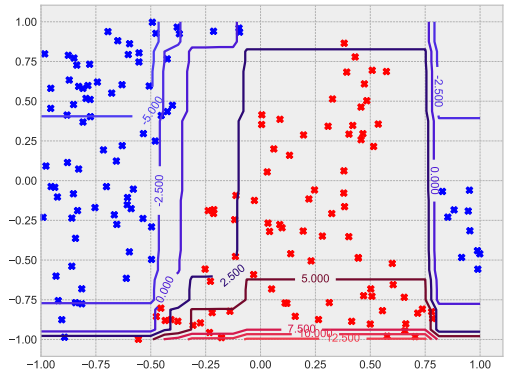
Boosted stumps.
($\mathcal{O}(n)$ param.)

Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



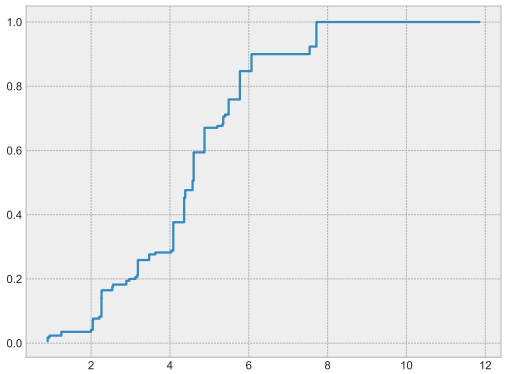


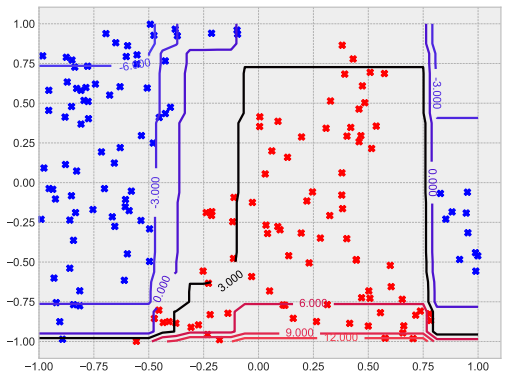
Boosted stumps.
($\mathcal{O}(n)$ param.)

Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



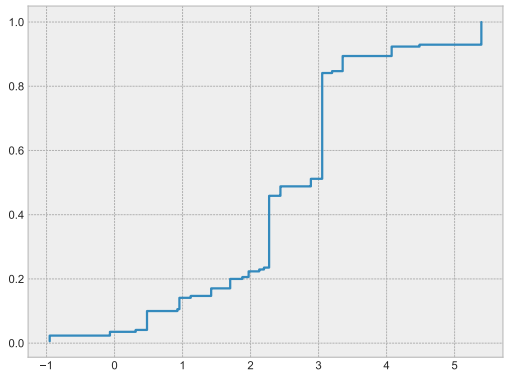


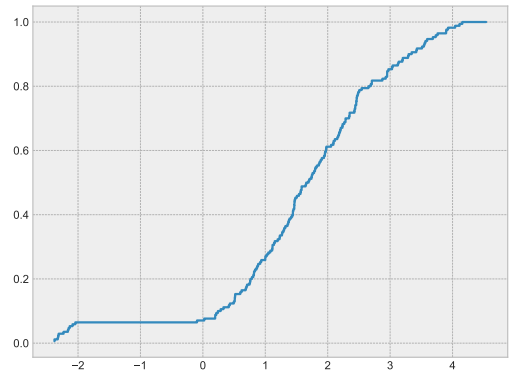
Boosted stumps.
($\mathcal{O}(n)$ param.)

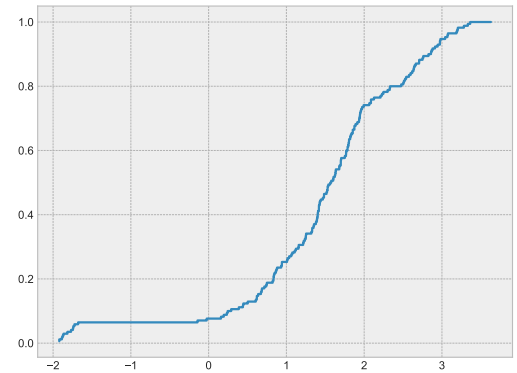
Margin plots

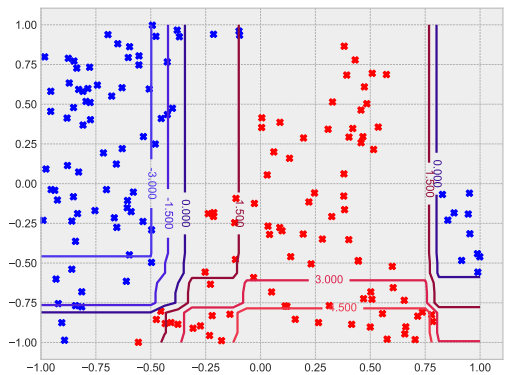
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

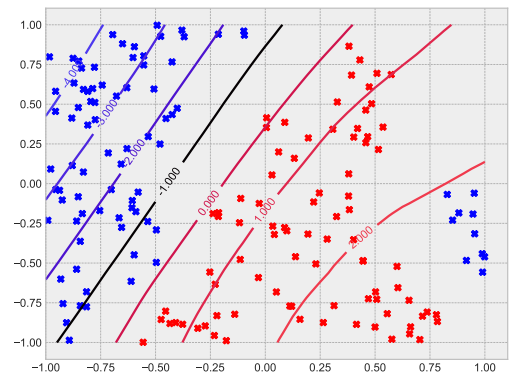
$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$

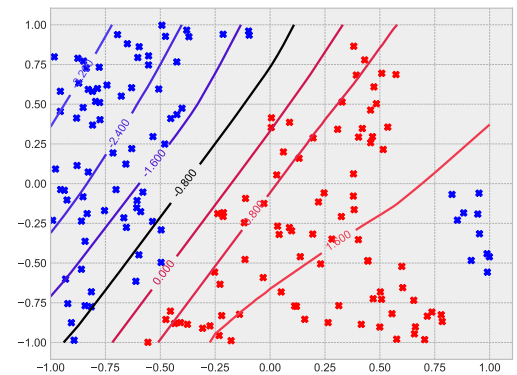












Boosted stumps.
($\mathcal{O}(n)$ param.)

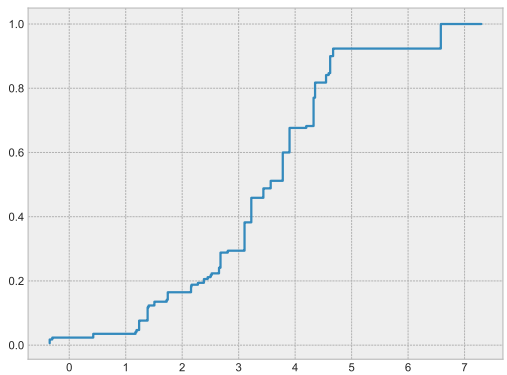
2-layer ReLU.
($\mathcal{O}(n)$ param.)

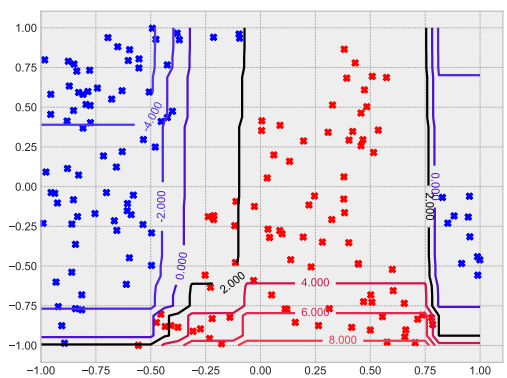
3-layer ReLU.
($\mathcal{O}(n)$ param.)

Margin plots

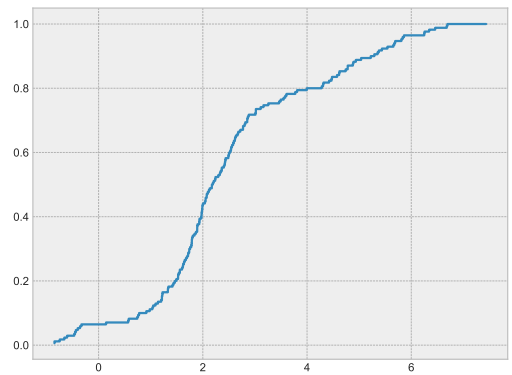
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

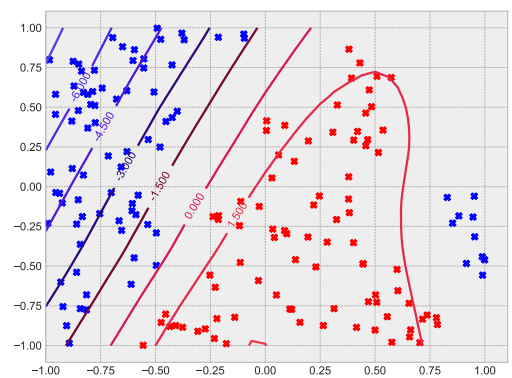
$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



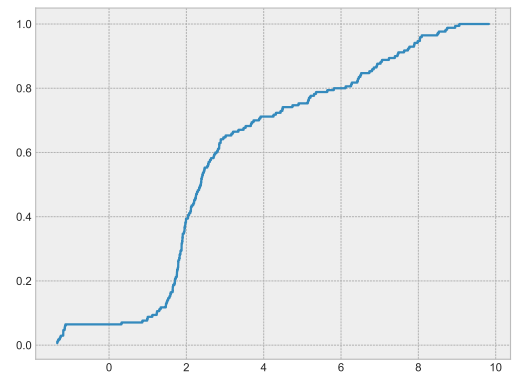


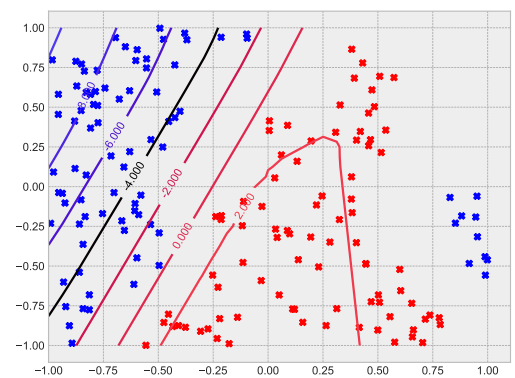
Boosted stumps.
($\mathcal{O}(n)$ param.)





2-layer ReLU.
($\mathcal{O}(n)$ param.)



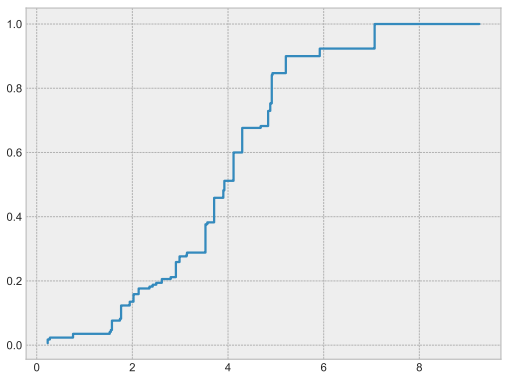


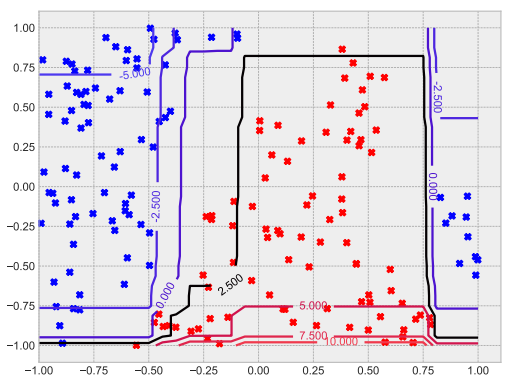
3-layer ReLU.
($\mathcal{O}(n)$ param.)

Margin plots

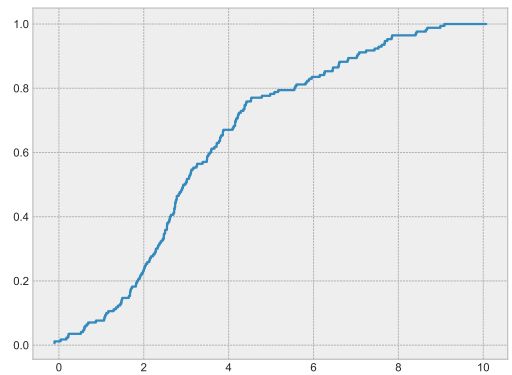
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

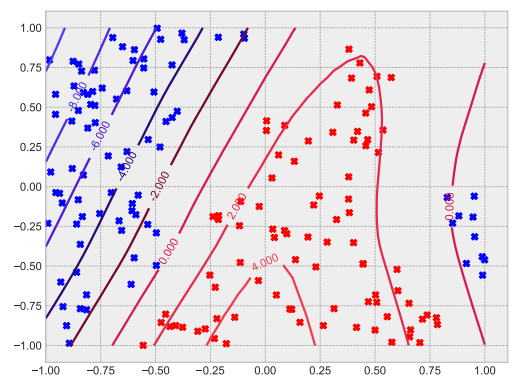
$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



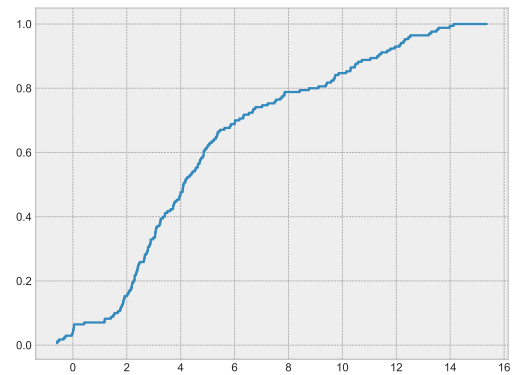


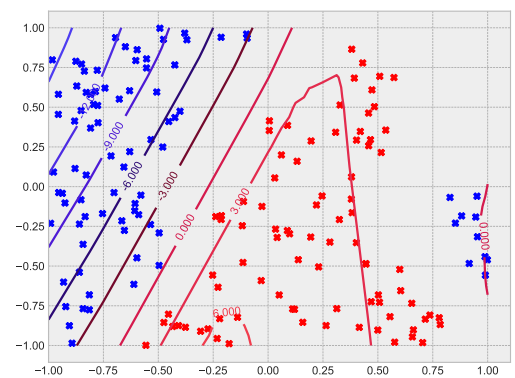
Boosted stumps.
($\mathcal{O}(n)$ param.)





2-layer ReLU.
($\mathcal{O}(n)$ param.)



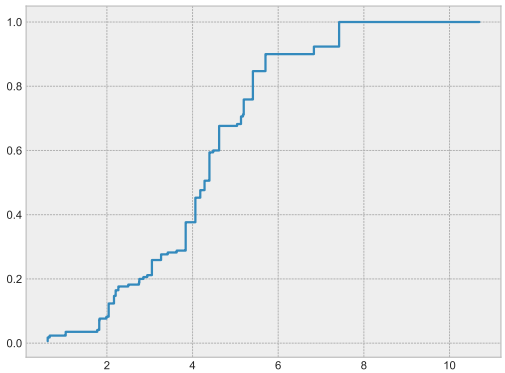


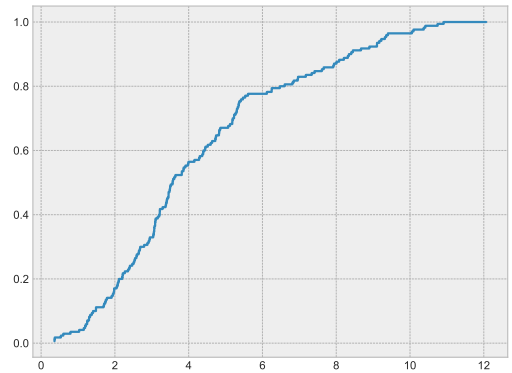
3-layer ReLU.
($\mathcal{O}(n)$ param.)

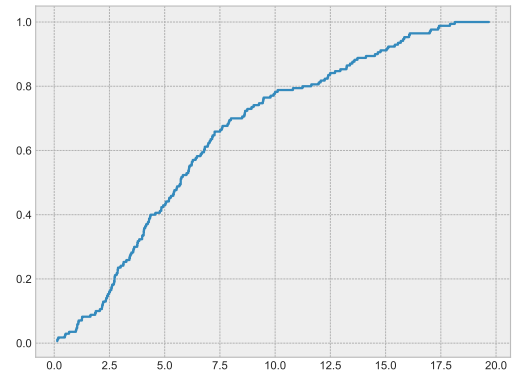
Margin plots

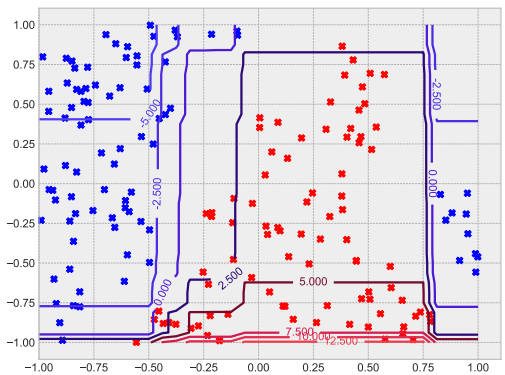
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

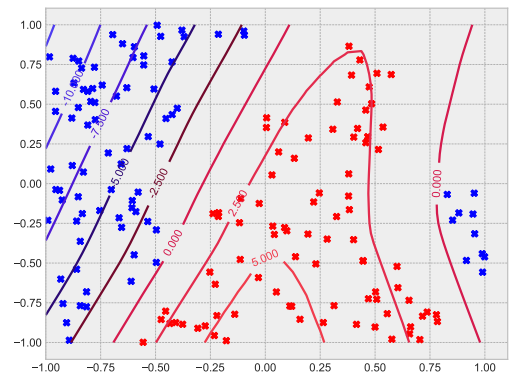
$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$

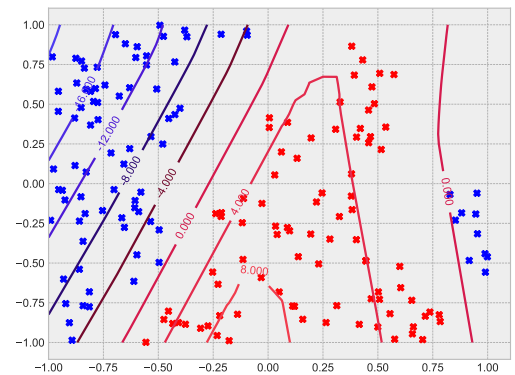












Boosted stumps.
($\mathcal{O}(n)$ param.)

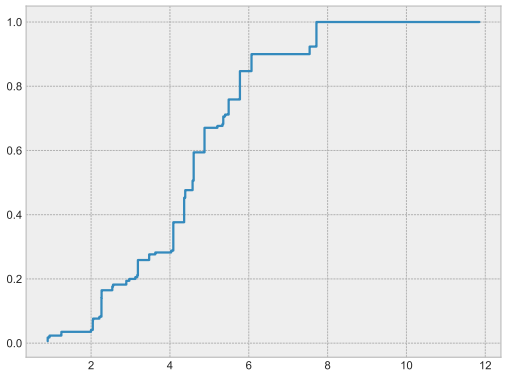
2-layer ReLU.
($\mathcal{O}(n)$ param.)

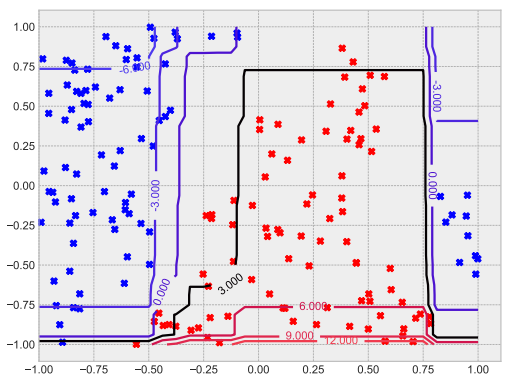
3-layer ReLU.
($\mathcal{O}(n)$ param.)

Margin plots

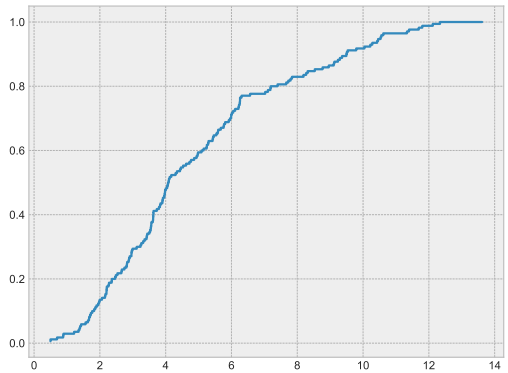
Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

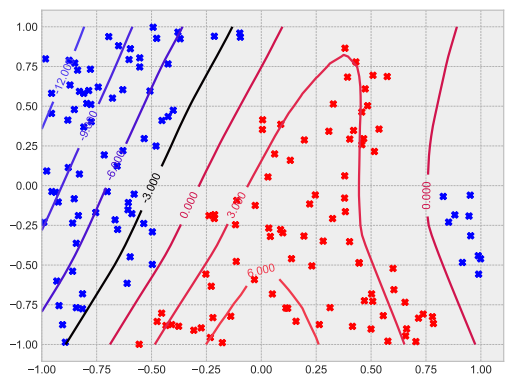
$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



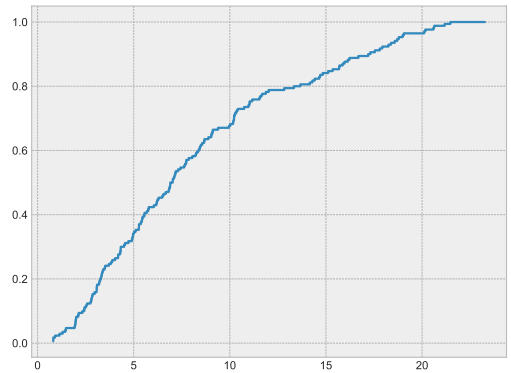


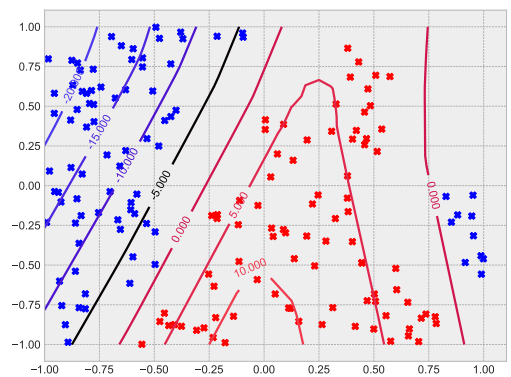
Boosted stumps.
($\mathcal{O}(n)$ param.)





2-layer ReLU.
($\mathcal{O}(n)$ param.)



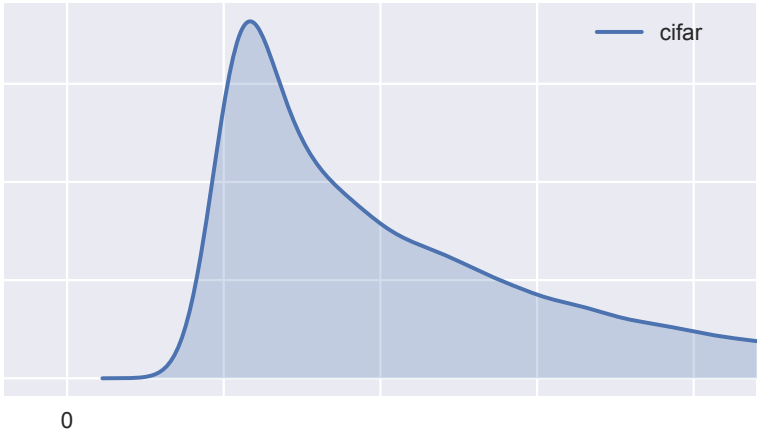


3-layer ReLU.
($\mathcal{O}(n)$ param.)

Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



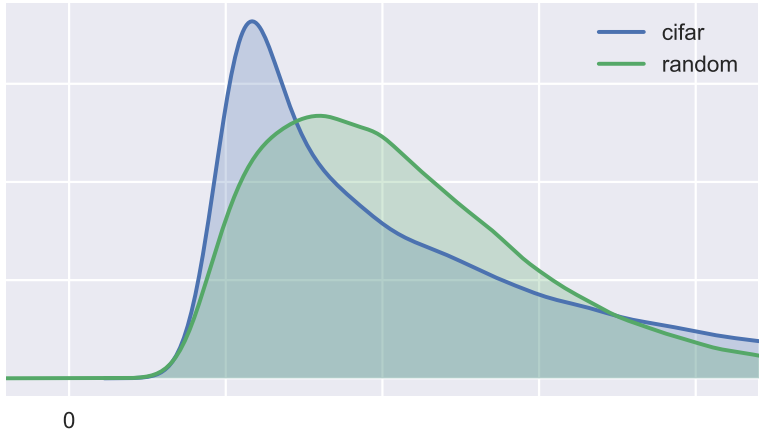
Unnormalized margins ($C_f = 1$).

Unnormalized margins can't distinguish random and true labels.

Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



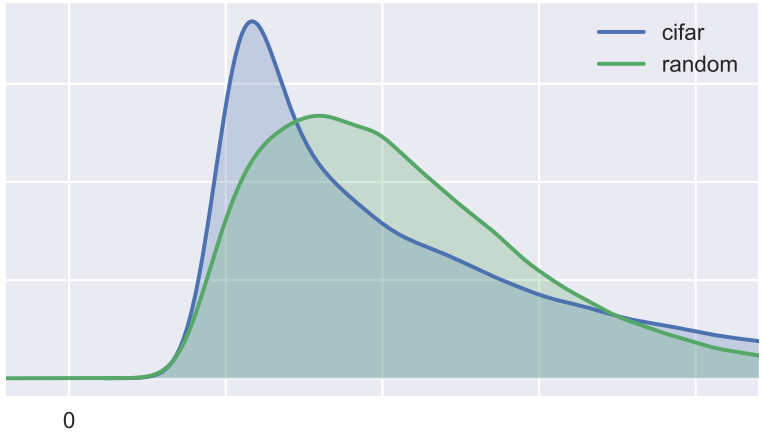
Unnormalized margins ($C_f = 1$).

Unnormalized margins can't distinguish random and true labels.

Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



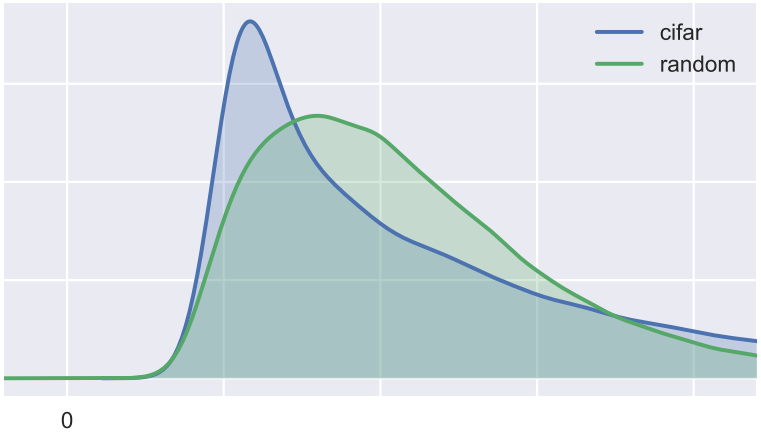
Unnormalized margins ($C_f = 1$).

Unnormalized margins can't distinguish random and true labels.

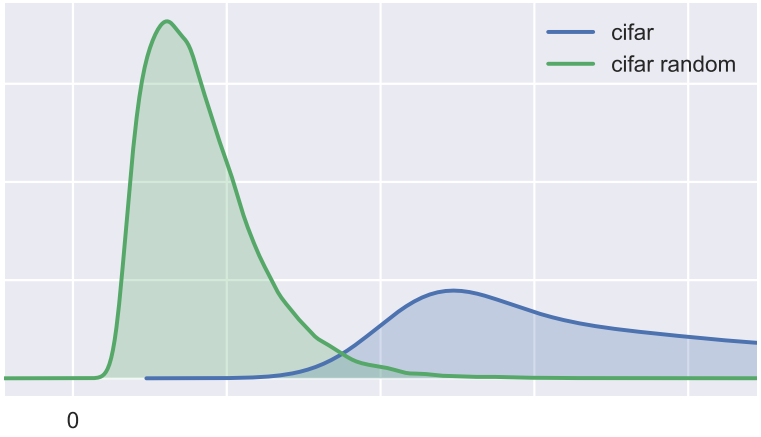
Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

$$\frac{f(x_i)y_i}{C_f} \quad \text{or} \quad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



Unnormalized margins ($C_f = 1$).



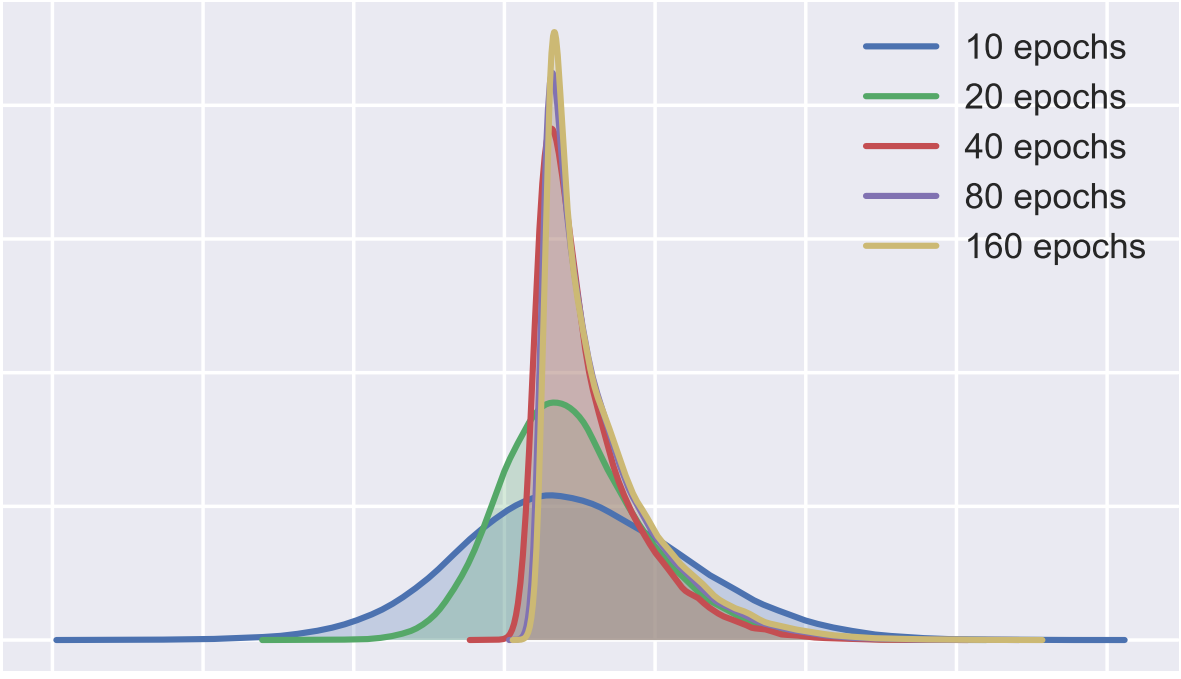
Normalized margins!

Unnormalized margins can't distinguish random and true labels.

Margin plots

Given $((x_i, y_i))_{i=1}^n$ and $f : \mathbb{R}^d \to \mathbb{R}^k$, plot distribution of

$$\frac{f(x_i)y_i}{C_f} \qquad \text{or} \qquad \frac{f(x_i)_{y_i} - \max_{y \neq y_i} f(x_i)_y}{C_f}.$$



GD improves normalized margins!

Summary of margin plots. (Purely empirical.)

Summary of margin plots. (Purely empirical.)

- Gradient descent seems to select large margin deep networks.

Summary of margin plots. (Purely empirical.)

- Gradient descent seems to select large margin deep networks.
- Margins are worse on random labels; as with boosting, there is a hope for good bounds on "reasonable data".

Summary of margin plots. (Purely empirical.)

- Gradient descent seems to select large margin deep networks.
- Margins are worse on random labels; as with boosting, there is a hope for good bounds on "reasonable data".

What about a generalization bound?
(And how is it normalized?)

Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[(\text{data})\left(\frac{\text{Lipschitz}}{\text{margin}}\right)\left(\text{nuisance}\right)\right].$$

Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\text{Lipschitz}}{\text{margin}}\right)\left(\text{nuisance}\right)\right].$$

Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\text{nuisance}\right)\right].$$

Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\sum_{i=1}^L \frac{\|W_i - W_i(0)\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)^{3/2}\right].$$

Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\sum_{i=1}^L \frac{\|W_i - W_i(0)\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)^{3/2}\right].$$

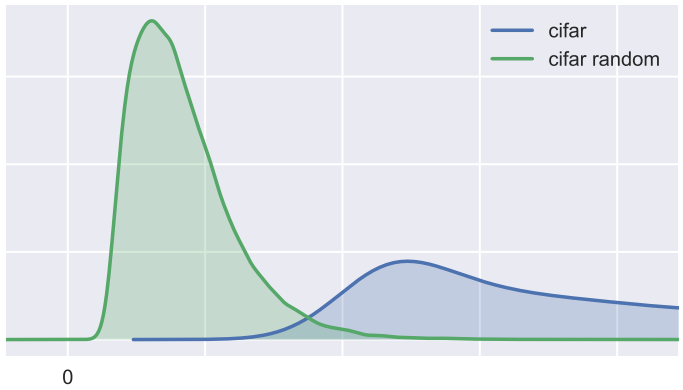
Remarks.

- $\prod_i \rho_i \|W_i\|$ is (upper bound on) Lipschitz constant.
- *Predictive* of empirical performance. (Next slide.)
- No explicit combinatorial factors. (But $\|X\|_F \ldots$)
- Multivariate gates (e.g., ReLU+maxpool) and output.

Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\sum_{i=1}^{L} \frac{\|W_i - W_i(0)\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)^{3/2}\right].$$

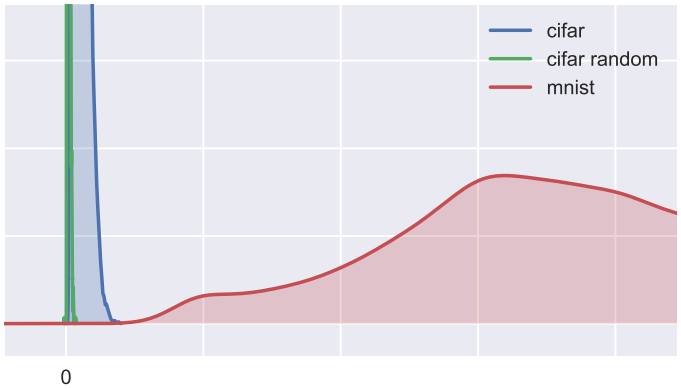
Predictive?

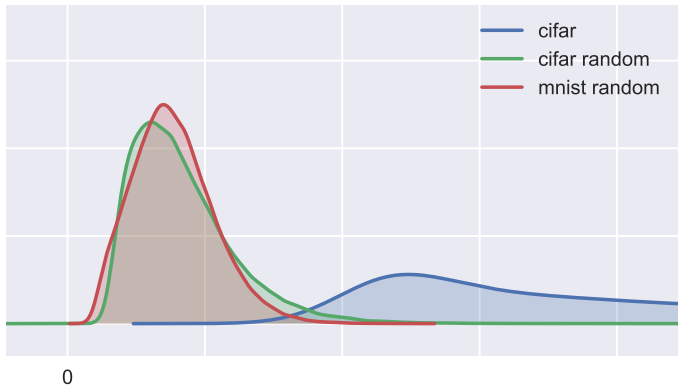


Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\sum_{i=1}^{L} \frac{\|W_i - W_i(0)\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)^{3/2}\right].$$

Predictive?

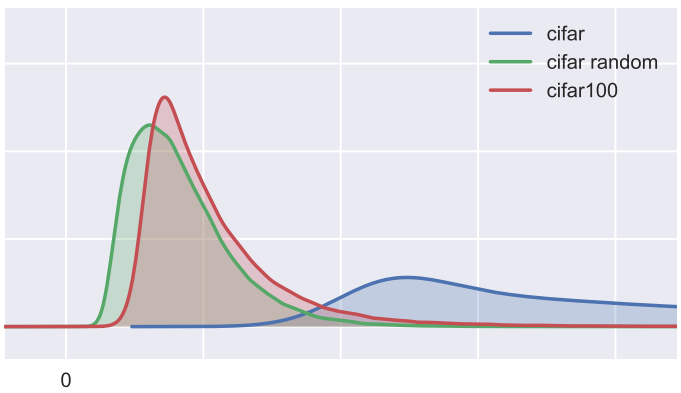


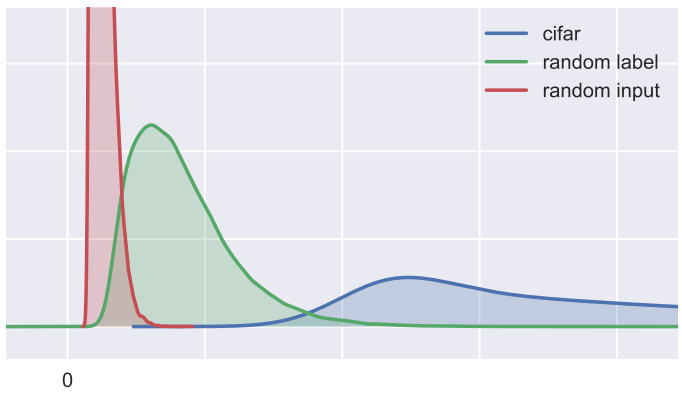


Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\sum_{i=1}^{L} \frac{\|W_i - W_i(0)\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)^{3/2}\right].$$

Predictive?

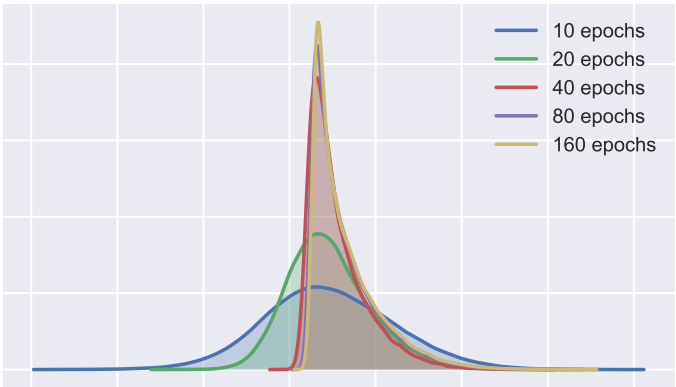


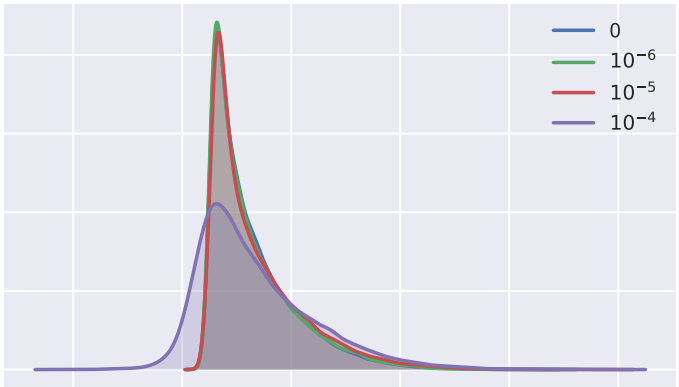


Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\sum_{i=1}^{L} \frac{\|W_i - W_i(0)\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)^{3/2}\right].$$

Predictive?





Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\sum_{i=1}^L \frac{\|W_i - W_i(0)\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)^{3/2}\right].$$

Negative remark: "nuisance term".

- Not in (linear!) lower bound.
- Linear lower bound has aligned matrices,
  upper bound has "maximally misaligned" matrices;
  "misaligned nonlinearities" quickly increase #pieces.
- Ratio term $\lesssim L^{3/2} w \approx L\sqrt{p}$; useless for ResNet, RNNs, etc.

Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\sum_{i=1}^L \frac{\|W_i - W_i(0)\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)^{3/2}\right].$$

Proof:
Layer-by-layer "matrix covering"
(avoids combinatorial parameters).

Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\sum_{i=1}^L \frac{\|W_i - W_i(0)\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)^{3/2}\right].$$

"Uniform devations?"

- This methodology equally controls all predictors in some set.
- Set can be made small
  via sensitivity to data, predictors, and algorithm.

Theorem (Bartlett-Foster-Telgarsky, 2017).
Fix any architecture with ρ-Lipschitz σ_i, $\sigma_i(0) = 0$.
With probability $\geq 1 - \delta$ over data $X \in \mathbb{R}^{n \times d}$,
for any margin γ and weights $(W_i)_{i=1}^L$,

$$\Pr[\text{error}] \leq \widehat{\Pr}[\gamma \text{ margin violation}] + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right)$$
$$+ \widetilde{\mathcal{O}}\left[\left(\frac{\|X\|_F}{n}\right)\left(\frac{\prod_i \rho_i \|W_i\|_2}{\gamma}\right)\left(\sum_{i=1}^L \frac{\|W_i - W_i(0)\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)^{3/2}\right].$$

Follow-up work and open problems.

- How to avoid degradation with depth and width?
- (Wei-Ma '19) and (Nagarajan-Kolter '19)
  depend on Jacobian matrices
  and bounds on within-network representations.
- (Golowich-Rakhlin-Shamir '18) improve dependency on Lipschitz, but $\sqrt{n}$ becomes $n^{1/5}$.

Comparison: VC Theory.

Theorem (Bartlett-Harvey-Liaw-Mehrabian '17).
Fix a ReLU architecture with L layers, p params.

$$\text{VC}(\text{architecture}) = \Theta\left(pL \ln p\right).$$

Comparison: VC Theory.

Theorem (Bartlett-Harvey-Liaw-Mehrabian '17).
Fix a ReLU architecture with L layers, p params.

$$\text{VC}(\text{architecture}) = \Theta\left(pL \ln p\right).$$

Remarks.

- This means: test error $\leq$ train error $+ \widetilde{\mathcal{O}}(\sqrt{pL/n})$.

Comparison: VC Theory.

Theorem (Bartlett-Harvey-Liaw-Mehrabian '17).
Fix a ReLU architecture with L layers, p params.

$$\text{VC}(\text{architecture}) = \Theta\left(pL \ln p\right).$$

Remarks.

- This means: test error $\leq$ train error $+ \widetilde{\mathcal{O}}(\sqrt{pL/n})$.
- This is "tight"; but $p > n$ in standard setups.
- Even though loose,
  proofs still illuminative.

Comparison: VC Theory.

Theorem (Bartlett-Harvey-Liaw-Mehrabian '17).
Fix a ReLU architecture with L layers, p params.

$$\text{VC}(\text{architecture}) = \Theta\left(pL \ln p\right).$$

Remarks.

- This means: test error $\leq$ train error $+ \widetilde{\mathcal{O}}(\sqrt{pL/n})$.
- This is "tight"; but $p > n$ in standard setups.
- Even though loose,
  proofs still illuminative.
- With threshold gates, becomes $\widetilde{\mathcal{O}}(p)$.
  Polynomial gates: $\widetilde{\mathcal{O}}(pL^2)$.
  Sigmoid: $\widetilde{\mathcal{O}}(m^2p^2)$ where m is # nodes.
  Arbitrary convex-concave: ∞.

Generalization summary.

- Empirically, GD seems to find large margin networks.
- Existing margin generalization bounds seem to correlate with observed generalization, but are still very loose.
- All existing bounds too sensitive to p and L...

Generalization summary.

- Empirically, GD seems to find large margin networks.
- Existing margin generalization bounds seem to correlate with observed generalization, but are still very loose.
- All existing bounds too sensitive to p and L...

Where next?

- In practice, increasing p and L can *help* generalization!
- Further understanding of gradient descent needed to better pin down "practically observed" networks.
- Going beyond classification?
- Regularization?

Generalization summary.

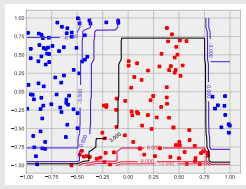
- Empirically, GD seems to find large margin networks.
- Existing margin generalization bounds seem to correlate with observed generalization, but are still very loose.
- All existing bounds too sensitive to p and L...

Where next?

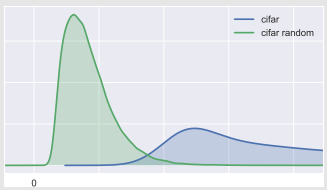
- In practice, increasing p and L can *help* generalization!
- Further understanding of gradient descent needed to better pin down "practically observed" networks.
- Going beyond classification?
- Regularization?

This lecture:

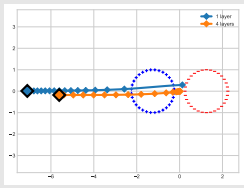
still need to *prove* gradient descent finds large margin networks.



Margins.



Generalization.



Optimization.

The linear case.

Single node neural network, no nonlinearity: $x \mapsto w^{\mathsf{T}} x$.

The linear case.

Single node neural network, no nonlinearity: $x \mapsto w^\mathsf{T} x$.

- Coordinate descent gets max margin predictor.

The linear case.

Single node neural network, no nonlinearity: $x \mapsto w^{\mathsf{T}} x$.

- Coordinate descent gets max margin predictor.
- Same proof (T '13) works for gradient descent (T-Ji '18), see also (Soudry-Hoffer-Nacson-Gunasekar-Srebro '17).

The linear case.

Single node neural network, no nonlinearity: $x \mapsto w^{\mathsf{T}} x$.

- Coordinate descent gets max margin predictor.
- Same proof (T '13) works for gradient descent (T-Ji '18), see also (Soudry-Hoffer-Nacson-Gunasekar-Srebro '17).
- What about something deeper?

Deep linear networks

No nonlinearities:

$$x \mapsto W_L \cdots W_1 x.$$

Deep linear networks

No nonlinearities:

$$x \mapsto W_L \cdots W_1 x.$$

- Linear in the input x;
  multi-linear/polynomial in parameters $(W_L, \ldots, W_1)$.

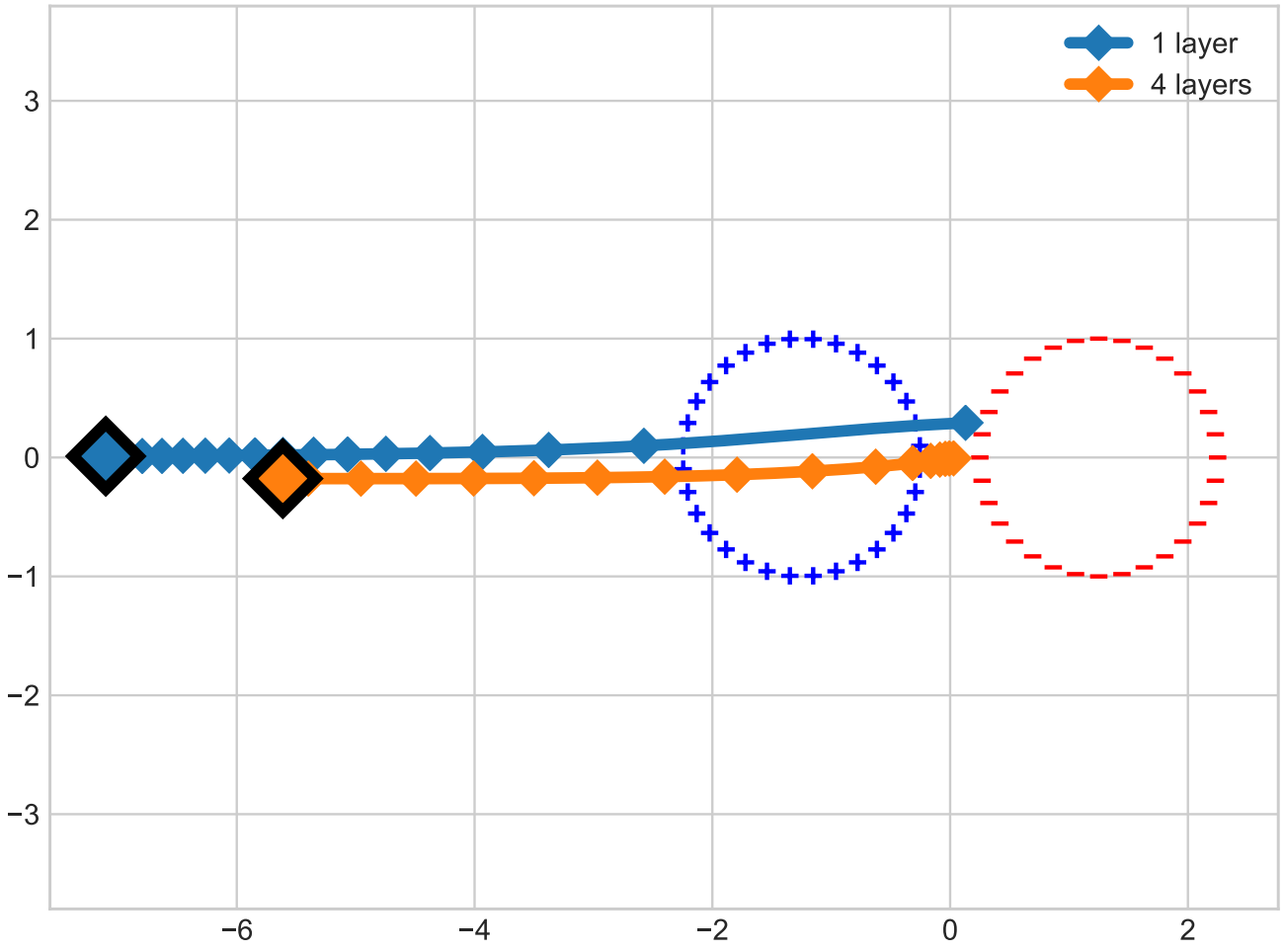
Deep linear networks

No nonlinearities:

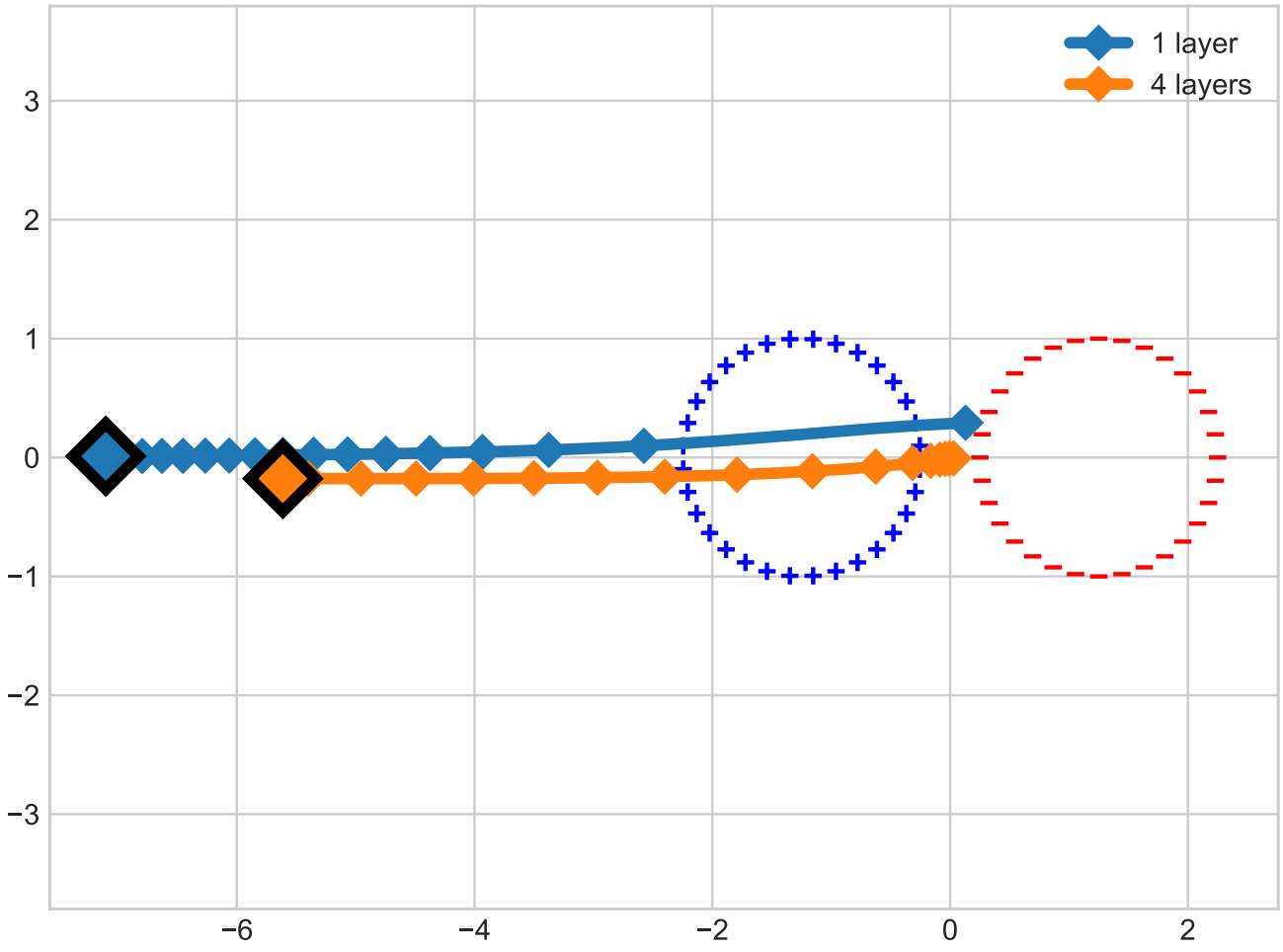
$$x \mapsto W_L \cdots W_1 x.$$

- Linear in the input x;
  multi-linear/polynomial in parameters $(W_L, \ldots, W_1)$.
- Not convex! What can we hope for?

Margin maximization..



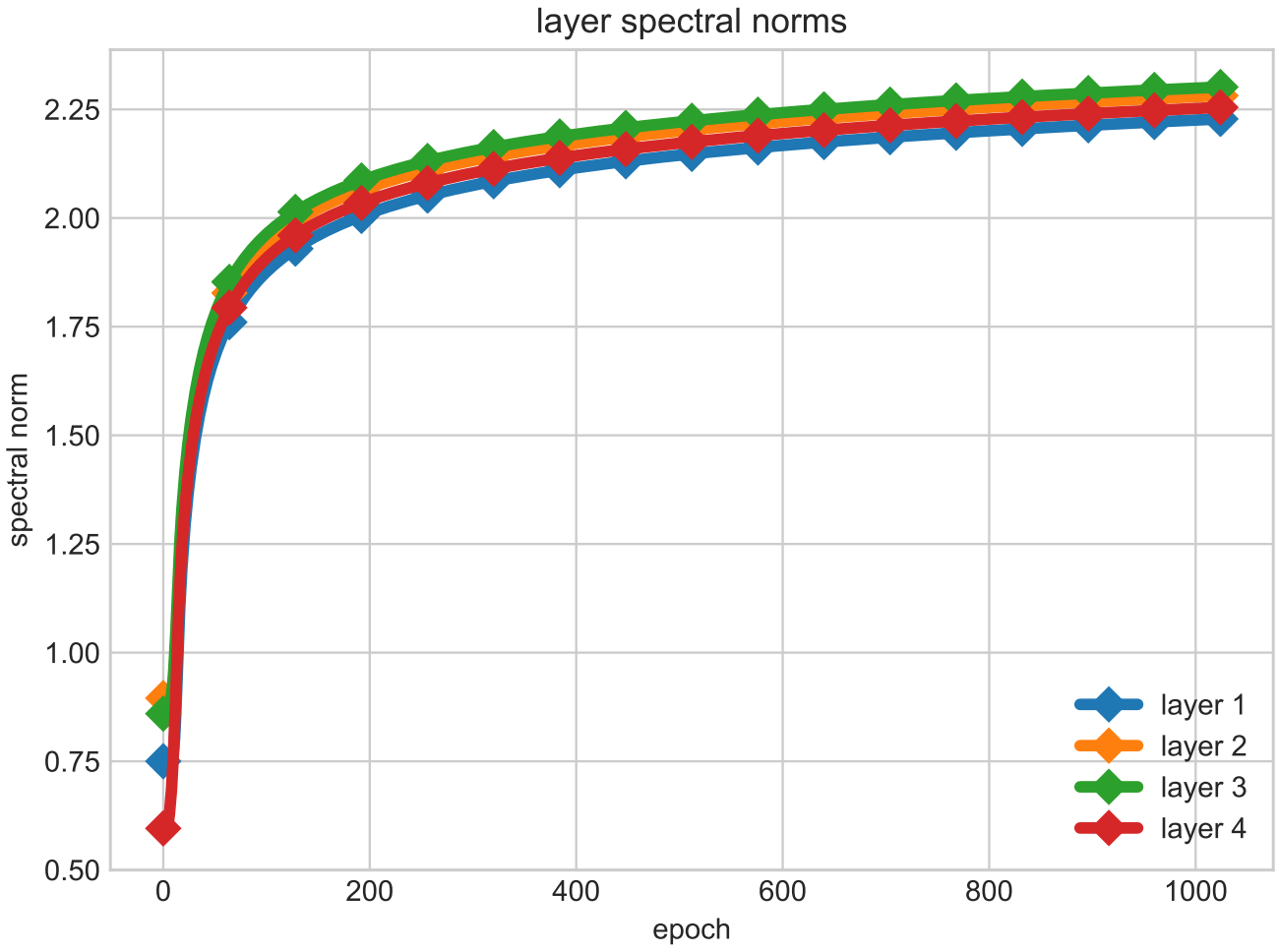
Margin maximization..



(This implies *at least one* weight matrix $\uparrow \infty$.)

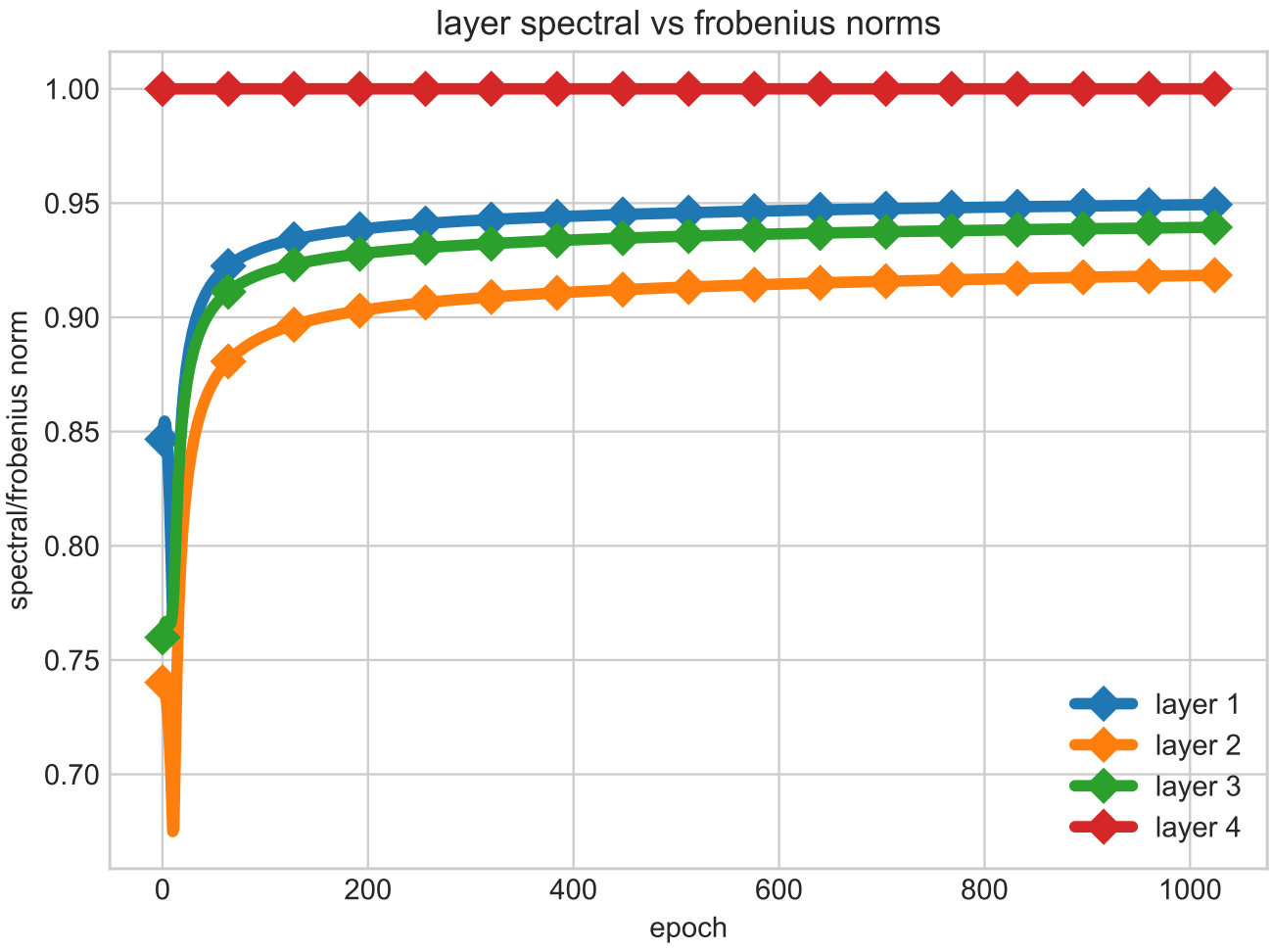
What do individual weight matrices do?

$\|W_i\|_2 \to \infty$ **for all layers** i**...** (simultaneously!)...



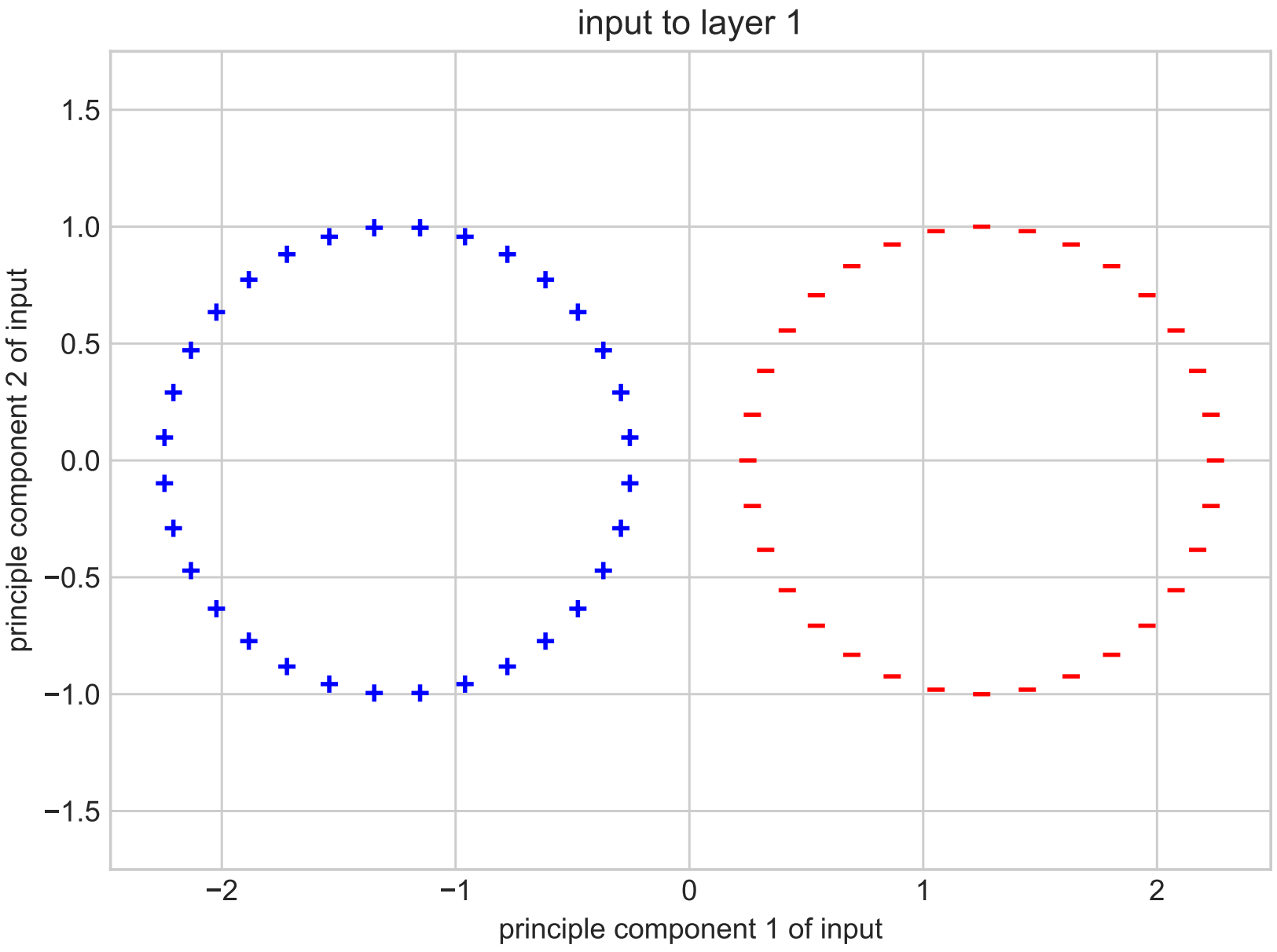
What are their ranks?

Layers are asymptotically rank 1!

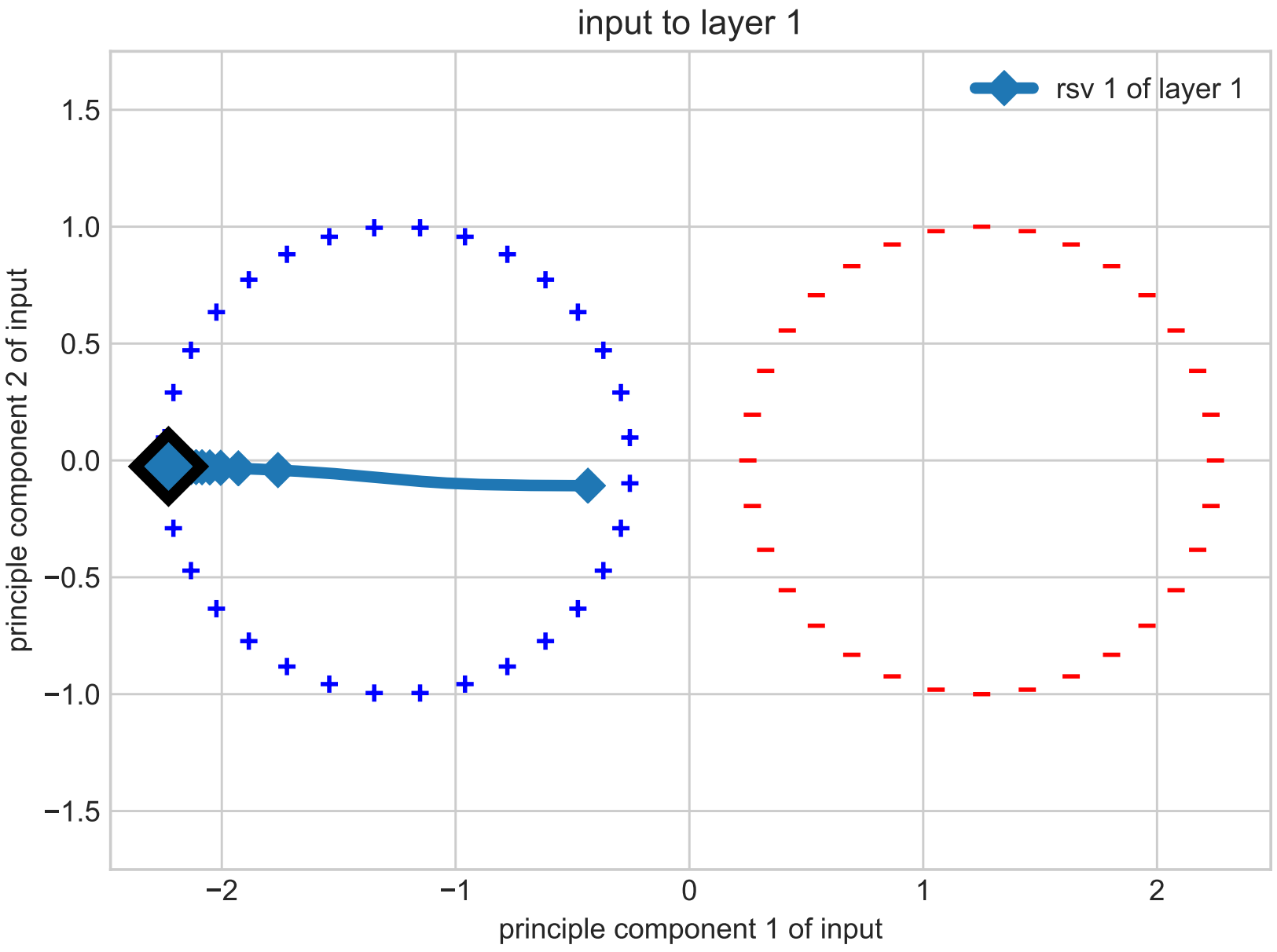


Are they "wasting norm" layer-to-layer?

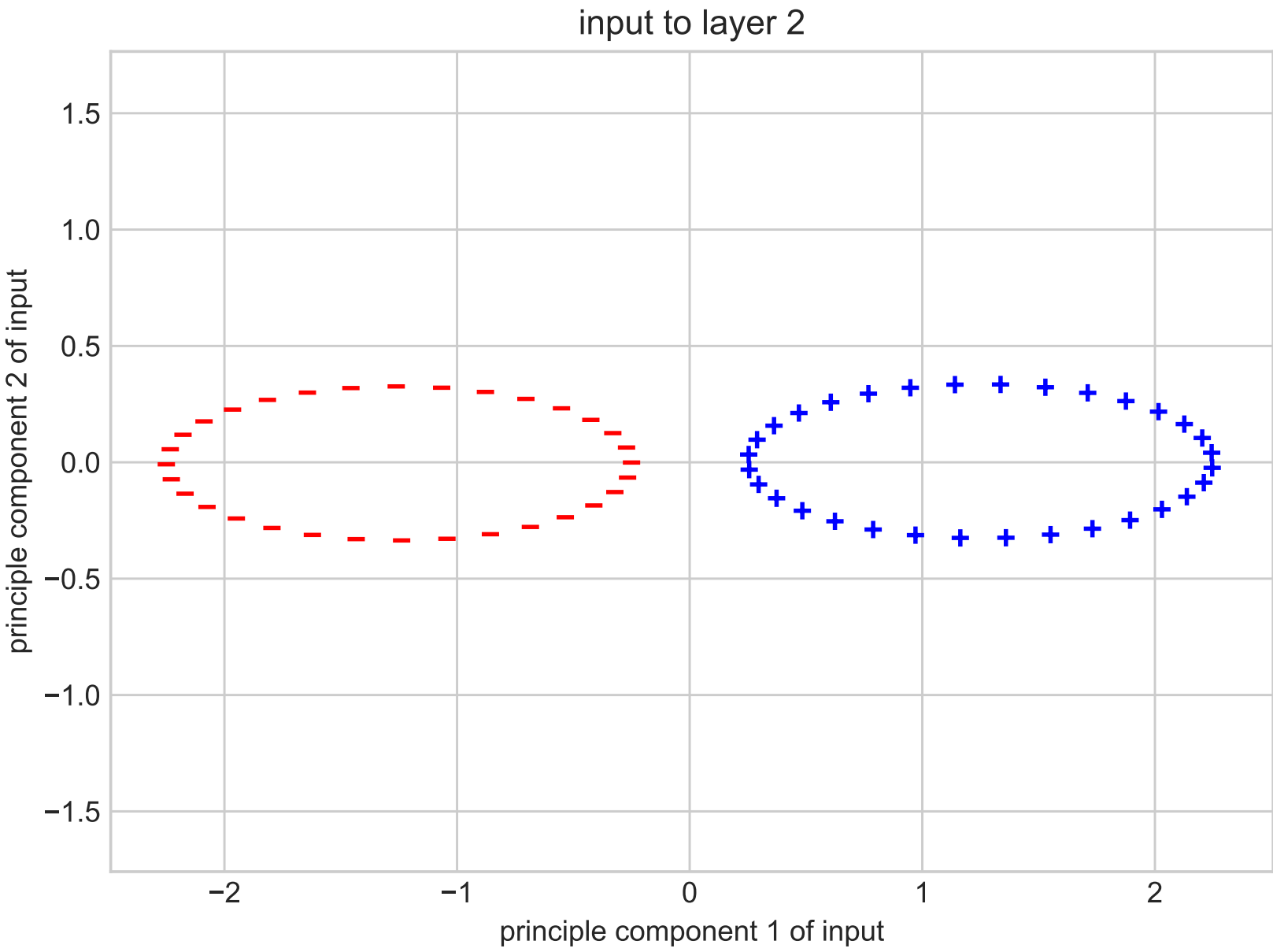
The layers *are aligned*.



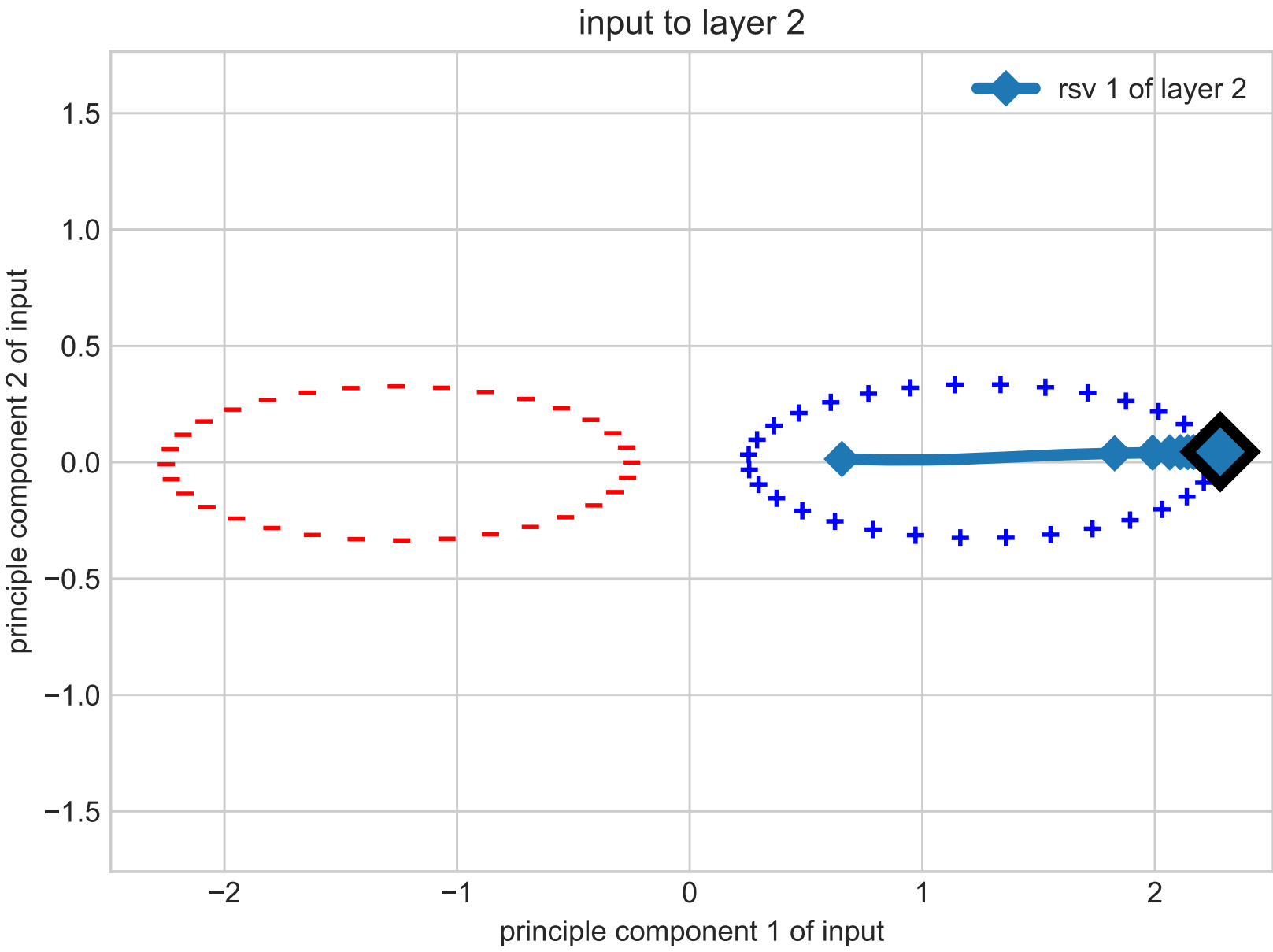
The layers *are aligned*.



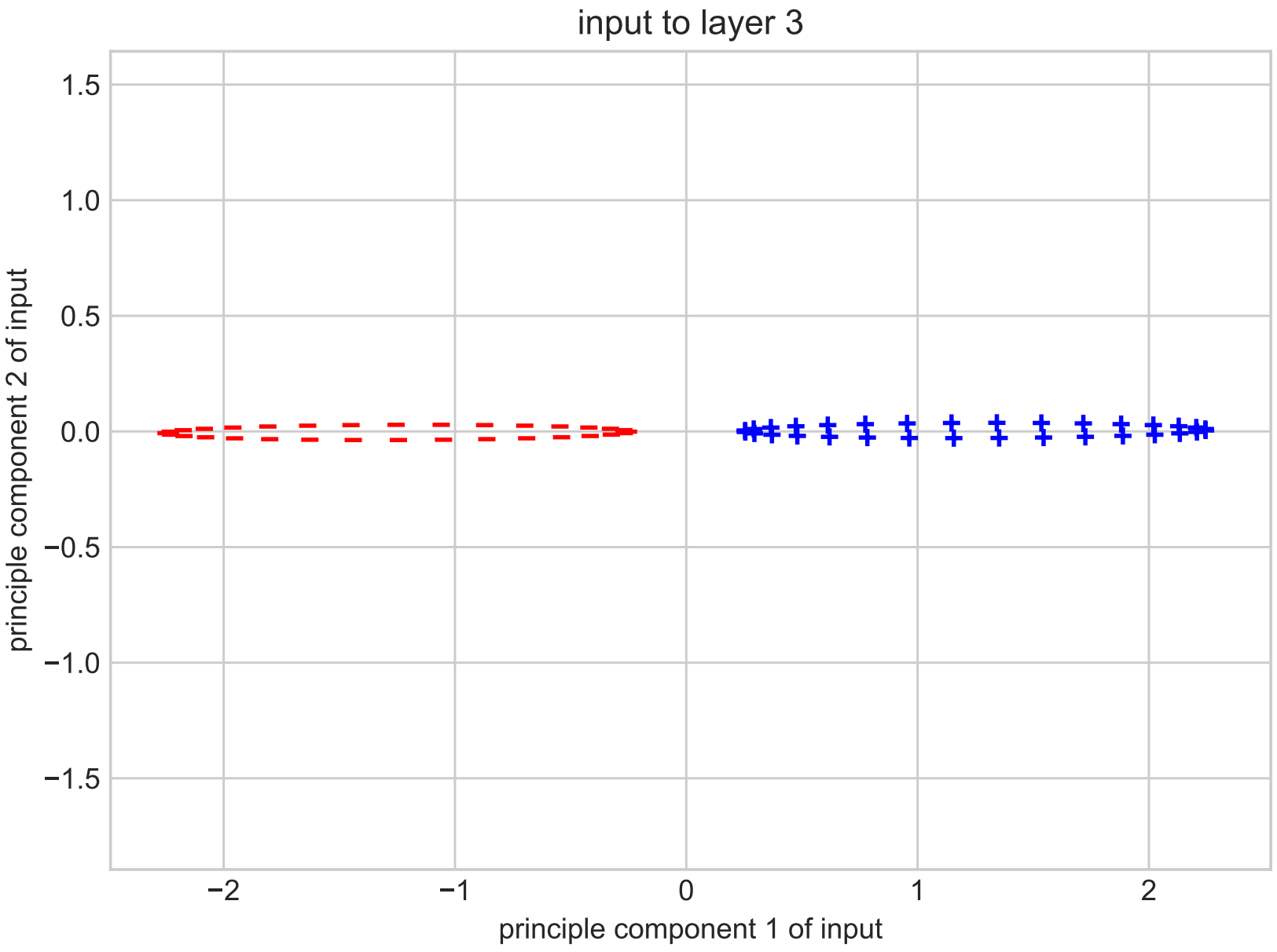
The layers *are aligned*.



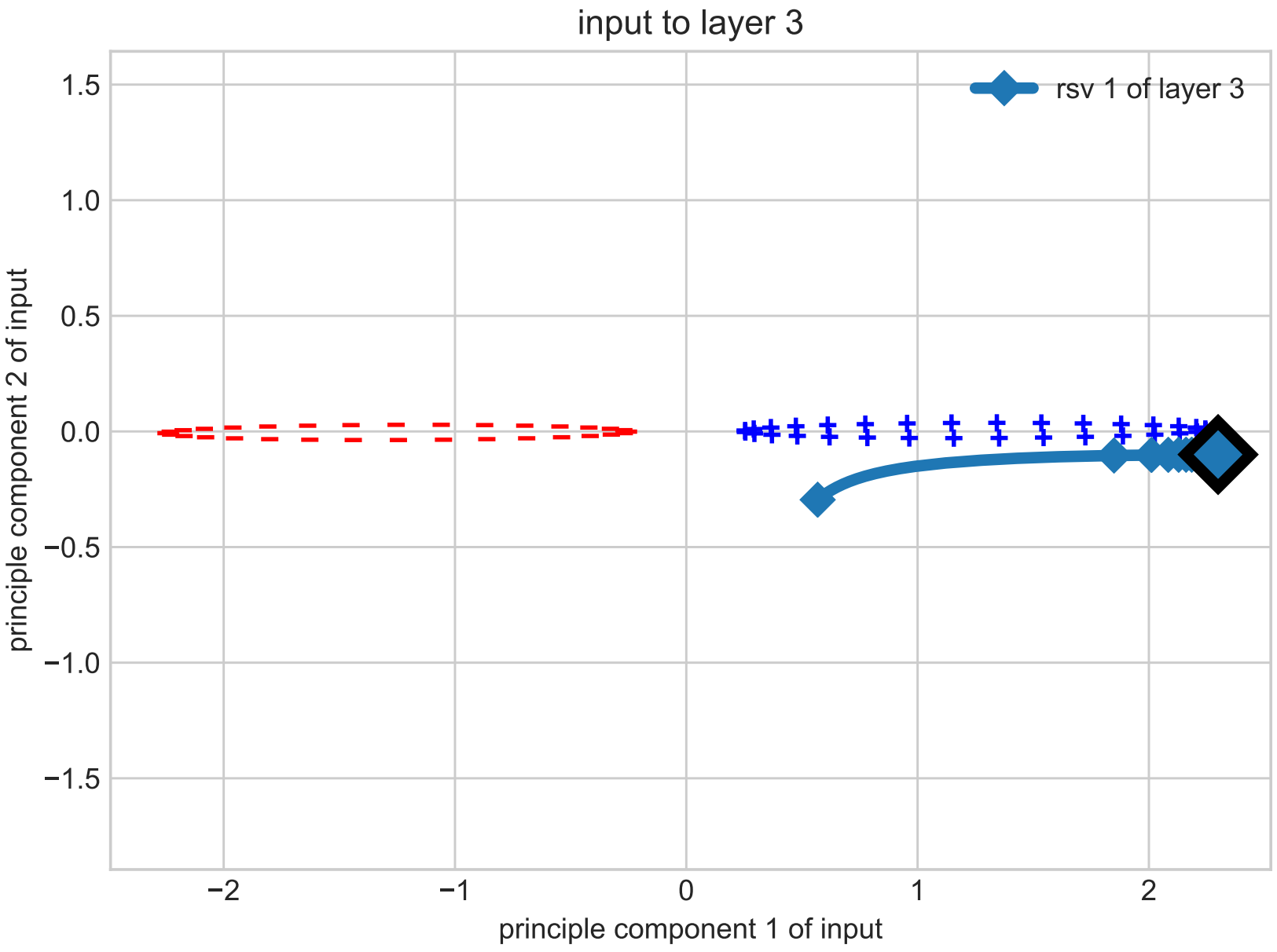
The layers *are aligned*.



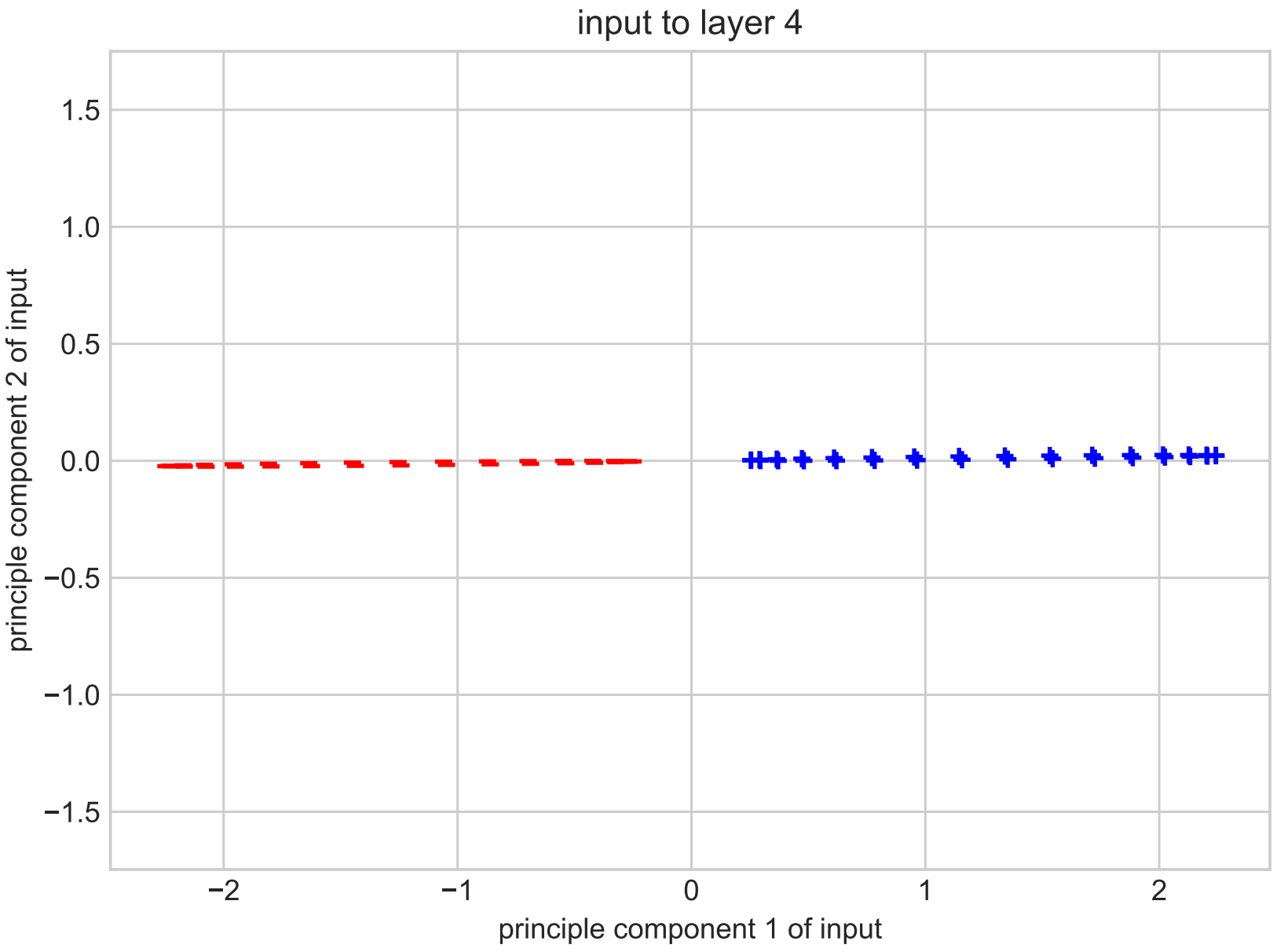
The layers *are aligned*.



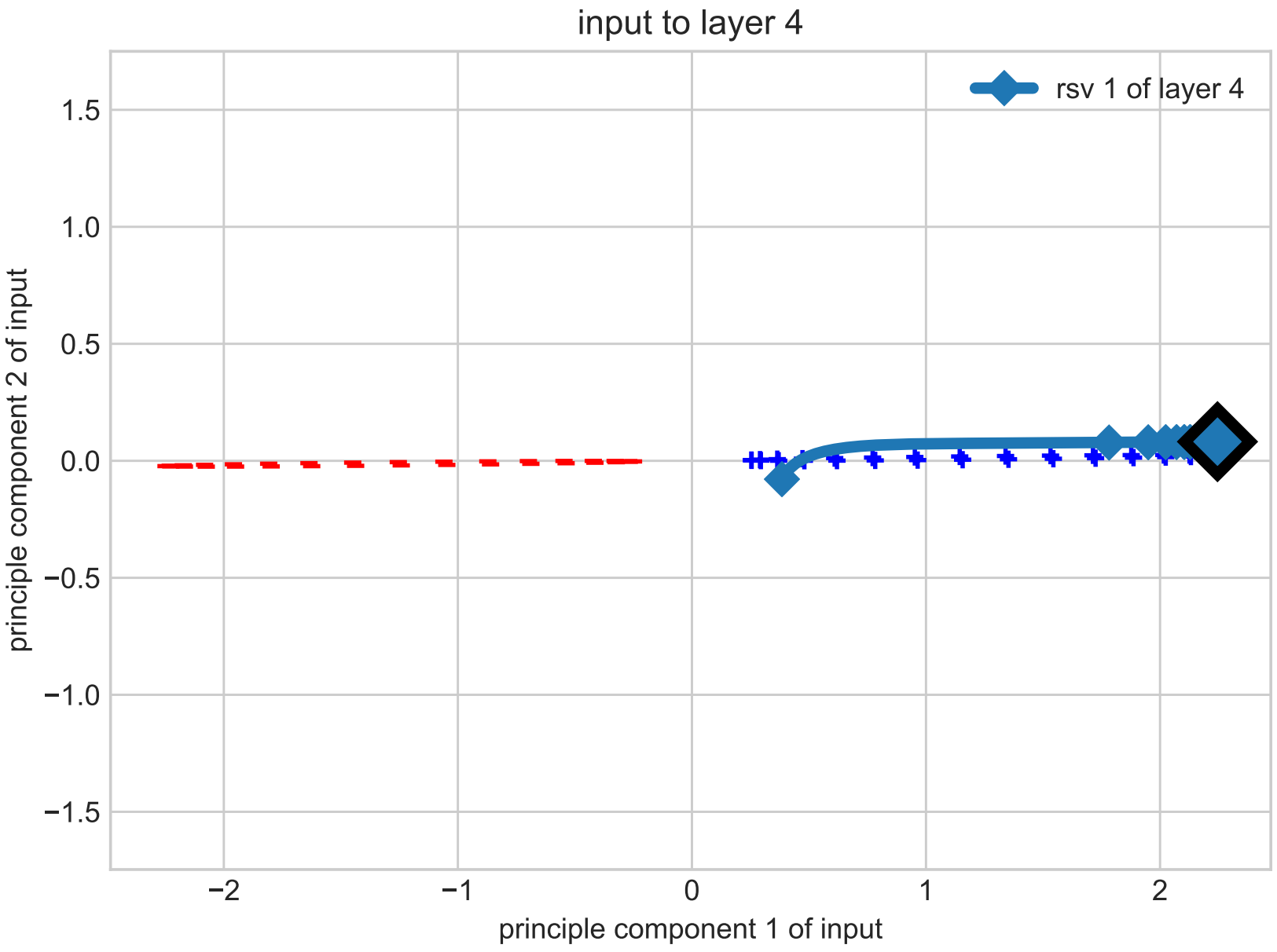
The layers *are aligned*.



The layers *are aligned*.



The layers *are aligned*.



Theorem (Ji-Telgarsky '18).

Suppose: *Strongly separable*, initially not saddle; gradient flow or GD+linesearch; logistic loss;

Then:

Theorem (Ji-Telgarsky '18).

Suppose: *Strongly separable*, initially not saddle; gradient flow or GD+linesearch; logistic loss;

Then: Normalized gradient descent iterates $\left(\frac{W_1}{\|W_1\|_{\mathrm{F}}}, \cdots, \frac{W_L}{\|W_L\|_{\mathrm{F}}}\right)$ converge to

$$\max_{\substack{W_1 \in \mathbb{R}^{d_2 \times d_1} \\ \|W_1\|_{\mathrm{F}}=1}} \cdots \max_{\substack{W_L \in \mathbb{R}^{1 \times d_L} \\ \|W_L\|_{\mathrm{F}}=1}} \min_i \; y_i \left(W_L \cdots W_1\right) x_i$$

Theorem (Ji-Telgarsky '18).

Suppose: *Strongly separable*, initially not saddle; gradient flow or GD+linesearch; logistic loss;

Then: Normalized gradient descent iterates $\left(\frac{W_1}{\|W_1\|_{\mathrm{F}}}, \cdots, \frac{W_L}{\|W_L\|_{\mathrm{F}}}\right)$ converge to

$$\max_{\substack{W_1 \in \mathbb{R}^{d_2 \times d_1} \\ \|W_1\|_{\mathrm{F}}=1}} \cdots \max_{\substack{W_L \in \mathbb{R}^{1 \times d_L} \\ \|W_L\|_{\mathrm{F}}=1}} \min_i \; y_i \left(W_L \cdots W_1\right) x_i$$

Remarks.

- Related work with convolution layers (Gunasekar-Lee-Soudry-Srebro '18).
- Generalization consequence: not just margins, can shrink "nuisance term"!
- Empirically on ReLU networks, *some alignment occurs*: ratios $\|W_i\|_{\mathrm{F}}/\|W_i\|_2$ are small.

Theorem (Ji-Telgarsky '18).

Suppose: *Strongly separable*, initially not saddle; gradient flow or GD+linesearch; logistic loss;

Then: Normalized gradient descent iterates $\left(\frac{W_1}{\|W_1\|_{\mathrm{F}}}, \cdots, \frac{W_L}{\|W_L\|_{\mathrm{F}}}\right)$ converge to

$$\max_{\substack{W_1 \in \mathbb{R}^{d_2 \times d_1} \\ \|W_1\|_{\mathrm{F}}=1}} \cdots \max_{\substack{W_L \in \mathbb{R}^{1 \times d_L} \\ \|W_L\|_{\mathrm{F}}=1}} \min_i \; y_i \left(W_L \cdots W_1\right) x_i$$

Proof remarks.

- Never underestimate simply writing down the gradient!

$$-\frac{\mathrm{d}W_i}{\mathrm{d}t} = \frac{\mathrm{d}\widehat{\mathcal{R}}}{\mathrm{d}W_i} = W_{i+1}^{\mathsf{T}} \cdots W_L^{\mathsf{T}} \nabla \widehat{\mathcal{R}}(w_{\mathrm{prod}}) W_1^{\mathsf{T}} \cdots W_{i-1}^{\mathsf{T}}.$$

  Implies: gradients rank 1, and $W_i^{\mathsf{T}} \left(\frac{\mathrm{d}\widehat{\mathcal{R}}}{\mathrm{d}W_i}\right) = \left(\frac{\mathrm{d}\widehat{\mathcal{R}}}{\mathrm{d}W_{i-1}}\right) W_{i-1}^{\mathsf{T}}$ (Du et al., 2018; Arora et al, 2018; Folklore).

- Can use perceptron proof ideas!

One approach to nonlinearities:

NTK (neural tangent kernel).

- $f(x; w)$ denotes network output with params w, input x.
- Let $\ell_i'(w) := \ell'\left(f(x_i; w), y_i\right)$; gradient step has form

$$w' := w - \eta \frac{1}{n} \sum_{i=1}^{n} \ell_i'(w) \nabla_w f(x_i; w).$$

One approach to nonlinearities:

NTK (neural tangent kernel).

- $f(x; w)$ denotes network output with params w, input x.
- Let $\ell_i'(w) := \ell'\left(f(x_i; w), y_i\right)$; gradient step has form

$$w' := w - \eta \frac{1}{n} \sum_{i=1}^{n} \ell_i'(w) \nabla_w f(x_i; w).$$

- If η is small, *predictions* evolve as

$$\begin{aligned} f(x; w') &\approx f(x; w) + \nabla_w f(x; w)^\mathsf{T} (w' - w) \\ &= f(x; w) - \eta \frac{1}{n} \sum_{i=1}^{n} \ell_i'(w) \nabla_w f(x; w)^\mathsf{T} \nabla_w f(x_i; w). \end{aligned}$$

One approach to nonlinearities:

NTK (neural tangent kernel).

- $f(x; w)$ denotes network output with params w, input x.
- Let $\ell_i'(w) := \ell'\left(f(x_i; w), y_i\right)$; gradient step has form

$$w' := w - \eta \frac{1}{n} \sum_{i=1}^{n} \ell_i'(w) \nabla_w f(x_i; w).$$

- If η is small, *predictions* evolve as

$$\begin{aligned} f(x; w') &\approx f(x; w) + \nabla_w f(x; w)^\mathsf{T}(w' - w) \\ &= f(x; w) - \eta \frac{1}{n} \sum_{i=1}^{n} \ell_i'(w) \nabla_w f(x; w)^\mathsf{T} \nabla_w f(x_i; w). \end{aligned}$$

- If $\eta \approx 0$, evolution of predictions is controlled by *kernel*

$$(x, x') \mapsto \nabla_w f(x; w)^\mathsf{T} \nabla_w f(x'; w).$$

The initial kernel

What if we replace $\nabla_w f(x; w)^\mathsf{T} \nabla_w f(x'; w)$ with $\mathbb{E}_{w_0} \nabla_w f(x; w_0)^\mathsf{T} \nabla_w f(x'; w_0)$?

- Sanity check: if $f(x; w) = w^\mathsf{T} x$, get $x^\mathsf{T} x'$.

The initial kernel

What if we replace $\nabla_w f(x; w)^\mathsf{T} \nabla_w f(x'; w)$
with $\mathbb{E}_{w_0} \nabla_w f(x; w_0)^\mathsf{T} \nabla_w f(x'; w_0)$?

- Sanity check: if $f(x; w) = w^\mathsf{T} x$, get $x^\mathsf{T} x'$.
- More generally:
  this is a linear model with "NTK features",
  a convex problem.

The initial kernel

What if we replace $\nabla_w f(x; w)^\mathsf{T} \nabla_w f(x'; w)$
with $\mathbb{E}_{w_0} \nabla_w f(x; w_0)^\mathsf{T} \nabla_w f(x'; w_0)$?

- Sanity check: if $f(x; w) = w^\mathsf{T} x$, get $x^\mathsf{T} x'$.
- More generally:
  this is a linear model with "NTK features",
  a convex problem.
- The expectation nicely handles cancellations in weights;
  can often be explicitly analyzed (Cho-Saul '09).

The initial kernel

What if we replace $\nabla_w f(x; w)^\mathsf{T} \nabla_w f(x'; w)$
with $\mathbb{E}_{w_0} \nabla_w f(x; w_0)^\mathsf{T} \nabla_w f(x'; w_0)$?

- Sanity check: if $f(x; w) = w^\mathsf{T} x$, get $x^\mathsf{T} x'$.
- More generally:
  this is a linear model with "NTK features",
  a convex problem.
- The expectation nicely handles cancellations in weights;
  can often be explicitly analyzed (Cho-Saul '09).
- Key question: does this relate to deep networks?

Initial and intermediate kernels

Initial and intermediate kernels

- Can prove: as $width \to \infty$ and $\eta \downarrow 0$,
  individual node weights change less and less,
  and deep network approaches kernel.

Initial and intermediate kernels

- Can prove: as $width \to \infty$ and $\eta \downarrow 0$,
  individual node weights change less and less,
  and deep network approaches kernel.
  - Clean analysis: "...Lazy Training..." (Chizat-Bach '19).

Initial and intermediate kernels

- Can prove: as $width \to \infty$ and $\eta \downarrow 0$,
  individual node weights change less and less,
  and deep network approaches kernel.
  - Clean analysis: "...Lazy Training..." (Chizat-Bach '19).
  - Many researchers have contributed many key ideas;
    "NTK" term is due to (Jacot-Gabriel-Hongler '18), but key ideas developed then and before by Zeyuan Allen-Zhu, Simon Du, Jason Lee, Yuanzhi Li, and many others.

Initial and intermediate kernels

- Can prove: as $width \to \infty$ and $\eta \downarrow 0$,
  individual node weights change less and less,
  and deep network approaches kernel.
  - Clean analysis: "...Lazy Training..." (Chizat-Bach '19).
  - Many researchers have contributed many key ideas;
    "NTK" term is due to (Jacot-Gabriel-Hongler '18), but key ideas developed then and before by Zeyuan Allen-Zhu, Simon Du, Jason Lee, Yuanzhi Li, and many others.
  - High level reason: standard initialization means increasing width rescales function; risk surface becomes flat.

Initial and intermediate kernels

- Can prove: as $width \to \infty$ and $\eta \downarrow 0$,
  individual node weights change less and less,
  and deep network approaches kernel.
  - Clean analysis: "...Lazy Training..." (Chizat-Bach '19).
  - Many researchers have contributed many key ideas;
    "NTK" term is due to (Jacot-Gabriel-Hongler '18), but key ideas developed then and before by Zeyuan Allen-Zhu, Simon Du, Jason Lee, Yuanzhi Li, and many others.
  - High level reason: standard initialization means increasing width rescales function; risk surface becomes flat.
  - ("Mean field" analysis related but different.)

Initial and intermediate kernels

- Can prove: as $width \to \infty$ and $\eta \downarrow 0$,
  individual node weights change less and less,
  and deep network approaches kernel.
  - Clean analysis: "...Lazy Training..." (Chizat-Bach '19).
  - Many researchers have contributed many key ideas;
    "NTK" term is due to (Jacot-Gabriel-Hongler '18), but key ideas developed then and before by Zeyuan Allen-Zhu, Simon Du, Jason Lee, Yuanzhi Li, and many others.
  - High level reason: standard initialization means increasing width rescales function; risk surface becomes flat.
  - ("Mean field" analysis related but different.)
- Unfortunately, these proofs require width $\Omega(n)$.

Initial and intermediate kernels

- Can prove: as $width \to \infty$ and $\eta \downarrow 0$,
  individual node weights change less and less,
  and deep network approaches kernel.
  - Clean analysis: "...Lazy Training..." (Chizat-Bach '19).
  - Many researchers have contributed many key ideas;
    "NTK" term is due to (Jacot-Gabriel-Hongler '18), but key ideas developed then and before by Zeyuan Allen-Zhu, Simon Du, Jason Lee, Yuanzhi Li, and many others.
  - High level reason: standard initialization means increasing width rescales function; risk surface becomes flat.
  - ("Mean field" analysis related but different.)
- Unfortunately, these proofs require width $\Omega(n)$.
- Empirically:
  - True for initial iterations on standard architectures.
  - In some rare cases, explicit NTK can match or slightly outperform deep network (!).

NTK remarks

- Proofs use smoothness to relate DL and NTK; ReLU can be handled with lots of work (AllenZhu-Li-Song '18).

NTK remarks

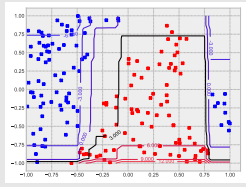
- Proofs use smoothness to relate DL and NTK;
  ReLU can be handled with lots of work
  (AllenZhu-Li-Song '18).
- NTK initial phase can still pick decent parameters;
  margin analysis can then say loss goes to 0 (Lyu-Li '19).

NTK remarks

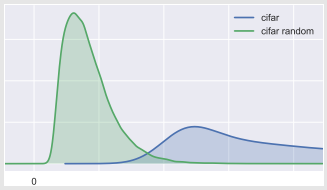
- Proofs use smoothness to relate DL and NTK; ReLU can be handled with lots of work (AllenZhu-Li-Song '18).
- NTK initial phase can still pick decent parameters; margin analysis can then say loss goes to 0 (Lyu-Li '19).
- These analyses require large width, kill generalization; preceding margin analysis doesn't guarantee sufficient margin.

NTK remarks

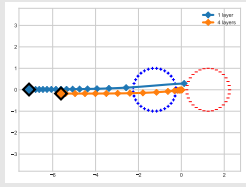
- Proofs use smoothness to relate DL and NTK; ReLU can be handled with lots of work (AllenZhu-Li-Song '18).
- NTK initial phase can still pick decent parameters; margin analysis can then say loss goes to 0 (Lyu-Li '19).
- These analyses require large width, kill generalization; preceding margin analysis doesn't guarantee sufficient margin.
- This analysis (and "mean field") are only ways I know to really take advantage of random initialization!



Margins.



Generalization.



Optimization.

Summary.

- Optimization and generalization are still open.

Summary.

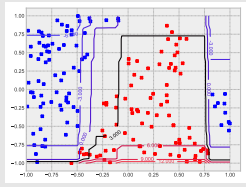
- Optimization and generalization are still open.
- Margin theory was effective with boosting;
  deep learning has similar traits,
  but the bounds aren't nearly adequate yet.

Summary.

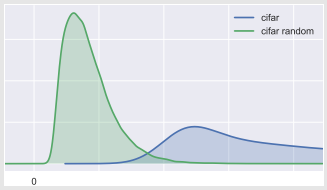
- Optimization and generalization are still open.
- Margin theory was effective with boosting;
  deep learning has similar traits,
  but the bounds aren't nearly adequate yet.
- Freezing the initial kernel ("NTK")
  does prove arbitrarily low training error in some nonlinear cases,
  but does not generalize.

Summary.

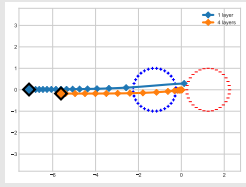
- Optimization and generalization are still open.
- Margin theory was effective with boosting;
  deep learning has similar traits,
  but the bounds aren't nearly adequate yet.
- Freezing the initial kernel ("NTK")
  does prove arbitrarily low training error in some nonlinear cases,
  but does not generalize.
  Moreover, it is known deep learning diverges from NTK,
  but only in limited constructions (AllenZhu-Li '19).



Margins.



Generalization.



Optimization.