高等机器学习

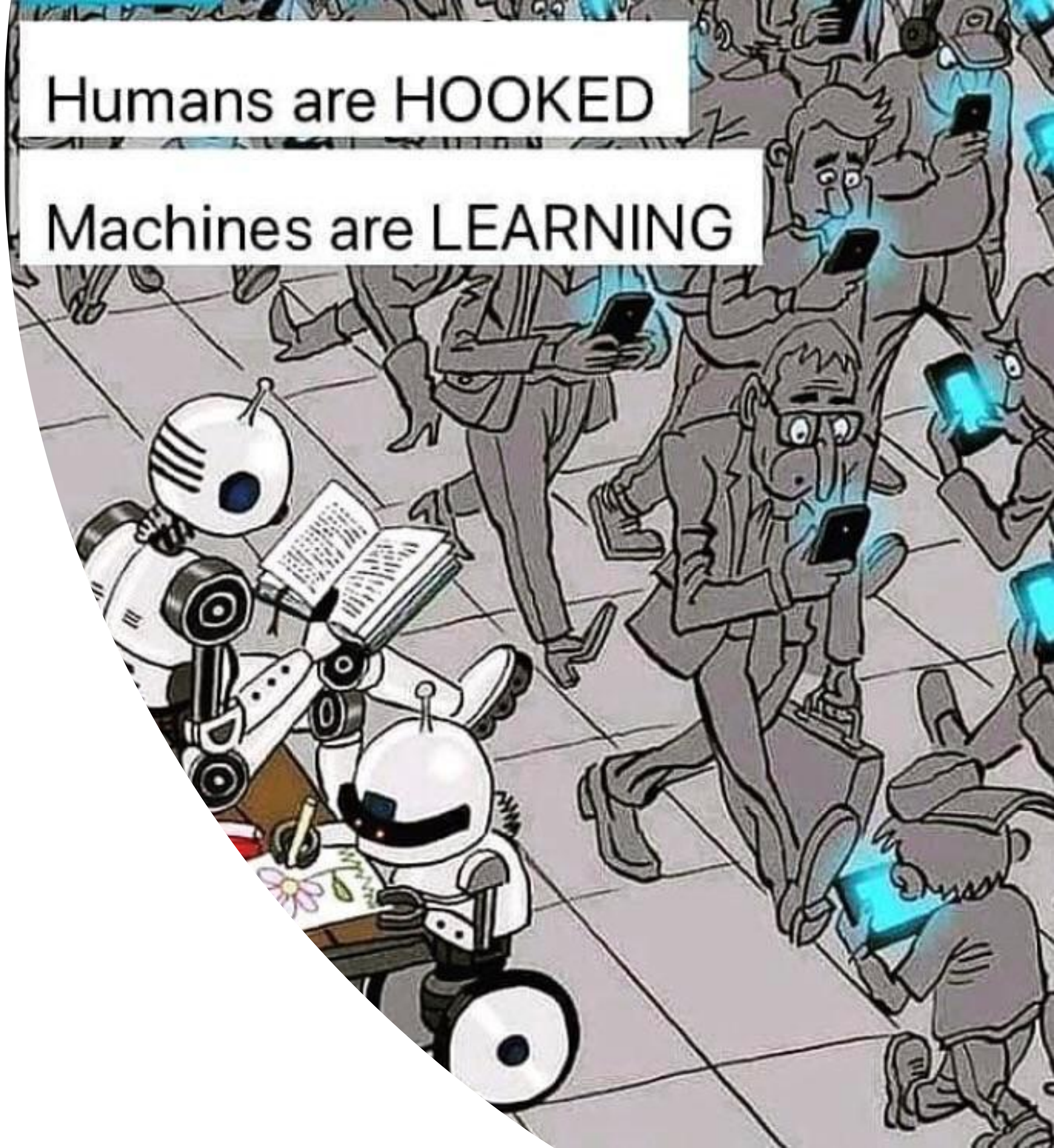
机器学习理论

张卫强

Microsoft

清華大學
Tsinghua University

Statistical Learning Theory



An example: Diagnostics for learning

- Formulation: maximize likelihood

$$\max_{\theta} \sum_{i=1}^{m} \log p\left(y^i \middle| x^i, \theta\right) - \lambda \|\theta\|^2$$

- High test error. Try to improve the algorithm!
  - Try getting more training examples.
  - Try a smaller set of features.
  - Try a larger set of features.
  - Try changing the features
  - Run gradient descent more iterations.
  - Try Newton's method.
  - Try another model (e.g., larger/smaller depth/width)
  - Use a different value of λ.

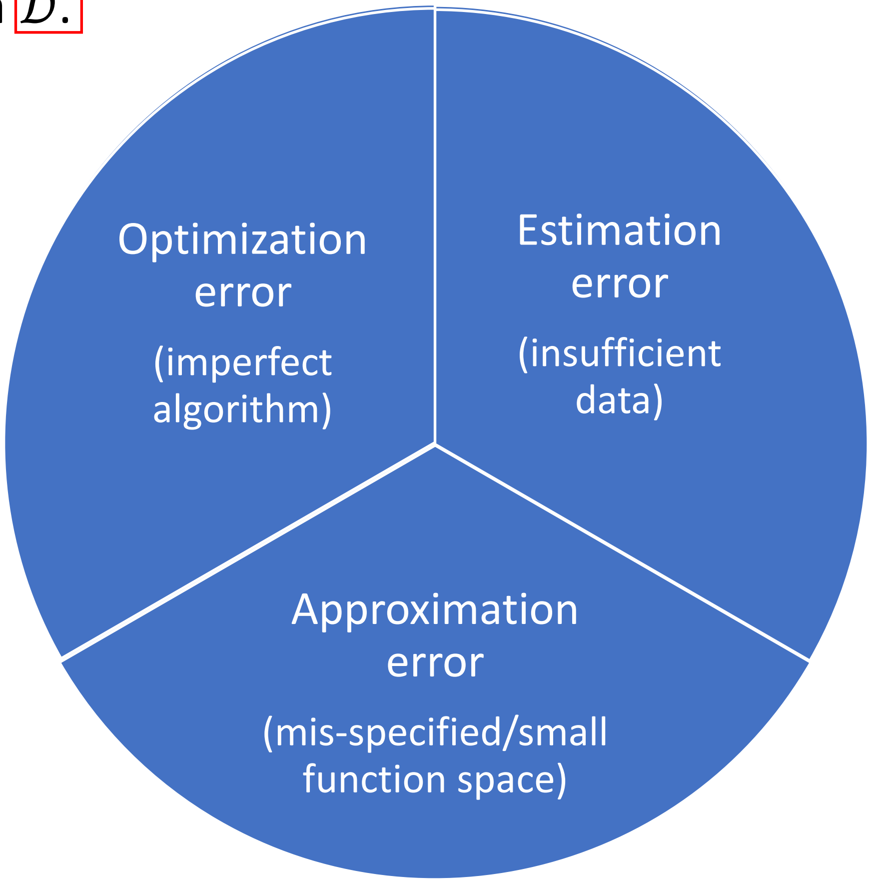
- A better approach: Diagnose the possible problem based on the observation.

Train loss	validation loss	Test loss	diagnosis
high	high	high	?
low	Low	low	done
low	high	high	?
low	low	high	?

Statistical Learning Theory

- Reality: Find a function f from a class $\mathcal{F}$ based on training data $\mathcal{D}$.
- Goal: f performs well on test data from a distribution $\mathcal{P}$?

- Where is the gap?
  - $\mathcal{D} \rightarrow \mathcal{P}$: estimation error
  - **Hypothesis space $\mathcal{F}$**: approximation error
  - **Find**: optimization error



A mathematical formulation

- Training data set:

$$S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$$

  Contains n i.i.d. copies of a random variable (x, y) with distribution D, where $x \in \mathcal{X}$ is feature, $y \in \mathcal{Y}$ is label

- Hypothesis class $\mathcal{F}$:

$$\mathcal{F} = \{f\colon \mathcal{X} \to \mathcal{Y}\} \text{(e.g. neural networks)}$$

- Loss function l:

  Prediction error $l(f(x), y)$ of f for a sample (x, y)

- Example: Least squares regression

$$x \in R^d, y \in R, \mathcal{F} = \left\{f \middle| f(x) = \theta^\top x, \theta \in R^d\right\}, l(f(x), y) = (f(x) - y)^2$$

A mathematical formulation

- Hypothesis set $\mathcal{F}$:

$$\mathcal{F} = \{f\colon \mathcal{X} \to \mathcal{Y}\} \text{ (e.g. neural networks)}$$

- Loss function l:

Prediction error $l(f(x), y)$ of f for a sample (x, y)

- $L(f) = L_D(f) = \mathbb{E}_{x,y \in D} l(f(x), y)$ → *Population Risk. The goal of learning is to minimize the risk.*
- $L_S(f) = \frac{1}{n}\sum_{i=1}^{n} l(f; x_i, y_i)$ → *Empirical Risk. Estimate of the risk: law of large number*
- For a hypothesis class $\mathcal{F}$, and the data distribution D, the population risk minimization is defined as

$$f_{\mathcal{F}}^* = \mathrm{argmin}_f L_D(f)$$

- For a hypothesis class $\mathcal{F}$, and the dataset S, the empirical risk minimization (ERM) is defined as

$$\hat{f}_{\mathcal{F}}^{\;*} = \mathrm{argmin}_{f \in \mathcal{F}} L_S(f)$$

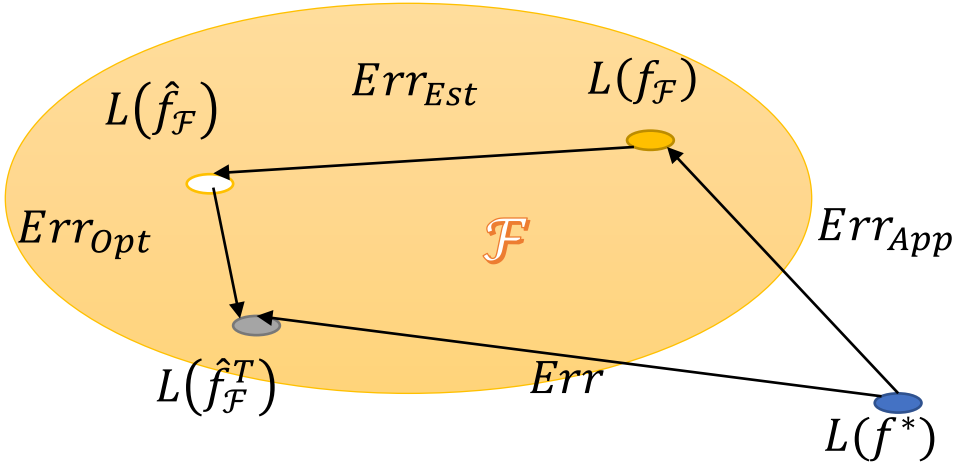
- For a the data distribution D, the bayes risk minimization is defined as

$$f^* = \mathrm{argmin}_f L_D(f)$$

Error Decomposition

Excess risk: Optimization Error Estimation Error Approximation Error

$$L(\hat{f}_{\mathcal{F}}^{T}) - L(f^*) = \boxed{\left(L(\hat{f}_{\mathcal{F}}^{T}) - L(\hat{f}_{\mathcal{F}}^{*})\right)} + \boxed{\left(L(\hat{f}_{\mathcal{F}}^{*}) - L(f_{\mathcal{F}}^{*})\right)} + \boxed{\left(L(f_{\mathcal{F}}^{*}) - L(f^*)\right)}$$



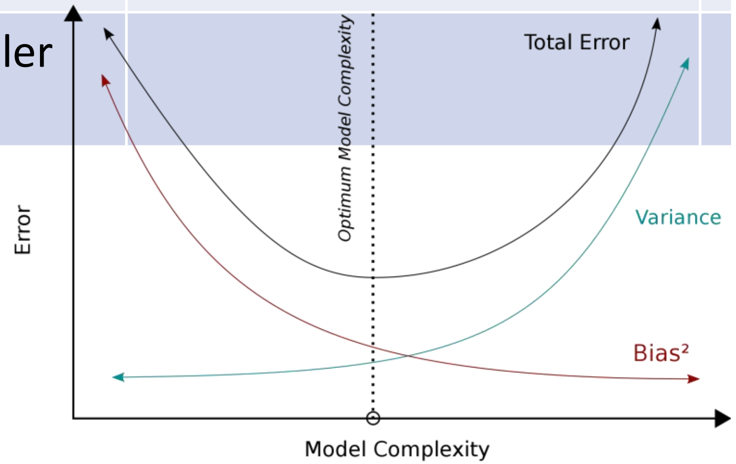
Error Decomposition

$$\underbrace{L(\hat{f}_{\mathcal{F}}^{T}) - L(f^{*})}_{\text{Excess risk:}} = \underbrace{\boxed{\left(L(\hat{f}_{\mathcal{F}}^{T}) - L(\hat{f}_{\mathcal{F}}^{*})\right)}}_{\text{Optimization Error}} + \underbrace{\boxed{\left(L(\hat{f}_{\mathcal{F}}^{*}) - L(f_{\mathcal{F}}^{*})\right)}}_{\text{Estimation Error}} + \underbrace{\boxed{\left(L(f_{\mathcal{F}}^{*}) - L(f^{*})\right)}}_{\text{Approximation Error}}$$

- Concept check:
  - Can excess risk ever be negative?
  - Is approximation error a random or non-random variable?
  - Is estimation error a random or non-random variable?
  - Can optimization error be negative?
  - Can we have a concrete example of finding $\hat{f}_{\mathcal{F}}^{T}$ and checking its error decomposition?

More Discussion

	Optimization Error	Estimation Error	Approximation Error
Definition	$L(\hat{f}_{\mathcal{F}}^{T}) - L(\hat{f}_{\mathcal{F}}^{*})$	$L(\hat{f}_{\mathcal{F}}^{*}) - L(f_{\mathcal{F}}^{*})$	$L(f_{\mathcal{F}}^{*}) - L(f^{*})$
Caused by	Approximate Optimization Algorithm	Finite Training Data	Limited Hypothesis Space
Hypothesis space $\mathcal{F}$	Not clear	the larger, the larger	the larger, the smaller
Number of training instances n	In general, the smaller, the smaller	the larger, the smaller	
Opt Algorithm and Iteration number T	the better/larger, the smaller		



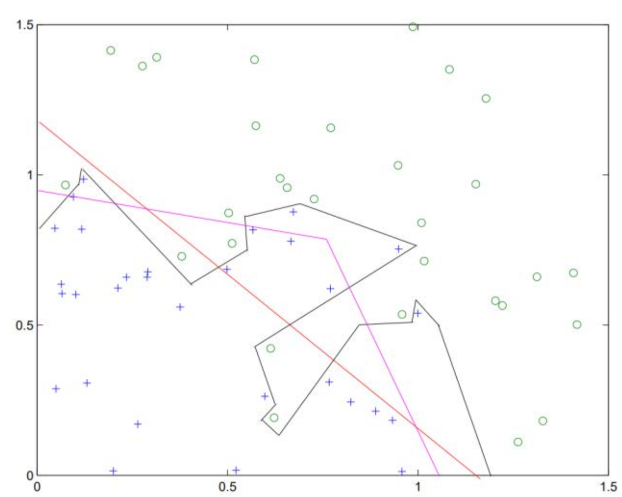


Fig. 1. Trade-off between fit and complexity.

Guarantees for Three Errors

- Optimization error <= = Convergence rate of optimization algorithms

$$L\left(\hat{f}_{\mathcal{F}}^{T}\right) - L\left(\hat{f}_{\mathcal{F}}\right) \leq \epsilon(Alg, \mathcal{F}, n, T)$$

- Estimation/generalization error <= = Upper bound in terms of capacity

$$L\left(\hat{f}_{\mathcal{F}}\right) - L(f_{\mathcal{F}}) \leq 2 \sup_{f \in \mathcal{F}} \left|\hat{L}(f) - L(f)\right| \leq \epsilon(Cap(\mathcal{F}), n)$$

- Approximation error (cannot be controllable in general) for neural networks <== Universal approximation theorem of neural networks

Outline

- Optimization theory
- Generalization theory
- Approximation theory

Definition of Convergence Rate

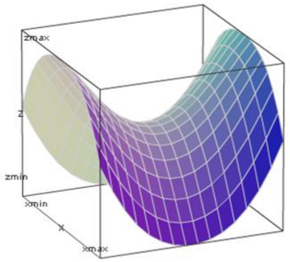
Assume the optimization error$L\left(\hat{f}_{\mathcal{F}}^{T}\right) - L\left(\hat{f}_{\mathcal{F}}\right) \leq \epsilon(Alg, \mathcal{F}, n, T)$

Does the log error $\log \epsilon(T)$ *decrease faster than* $-T$*?*

- Equal to: linear convergence rate, e.g., $O(e^{-T})$
- Faster than: super-linear convergence rate, e.g., $O\left(e^{-T^2}\right)$
  - Quadratic: $\log \log \epsilon(T)$ deceasing in the same order with $-T$, e.g. $O\left(e^{-2^T}\right)$
- Slower than: sub-linear convergence rate, e.g., $O\left(\frac{1}{T}\right)$

Convexity

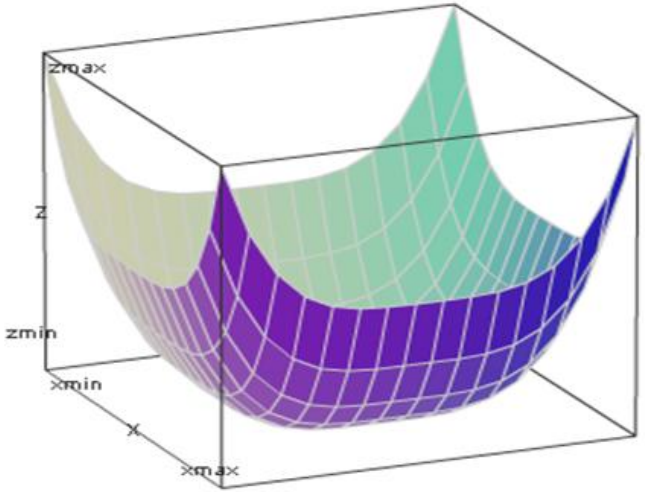
$$g(w) = w_1^2 - w_2^2$$

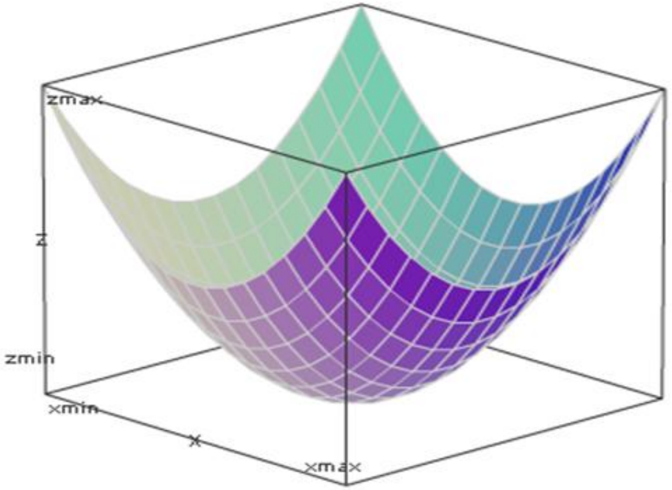


$$g(w) - g(v) \geq \nabla g(v)^\tau (w - v)$$
$$\forall w, v \in \mathcal{W},$$

$$g(w) - g(v) \geq \nabla g(v)^\tau (w - v) + \frac{\alpha}{2}||w - v||^2$$
$$\forall w, v \in \mathcal{W},$$

$$g(w) = w_1^4 + w_2^4$$





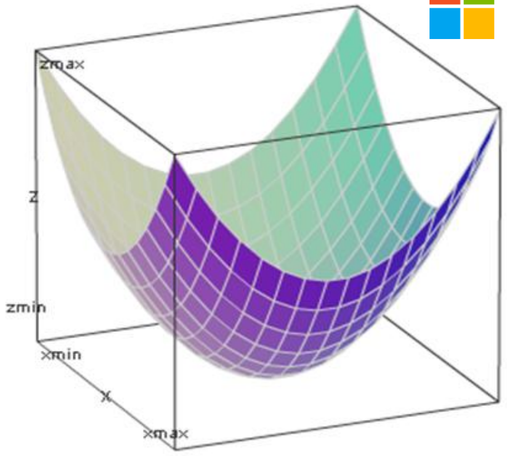
$$g(w) = w_1^2 + w_2^2$$

Convex

Strongly-Convex

Microsoft

Smoothness



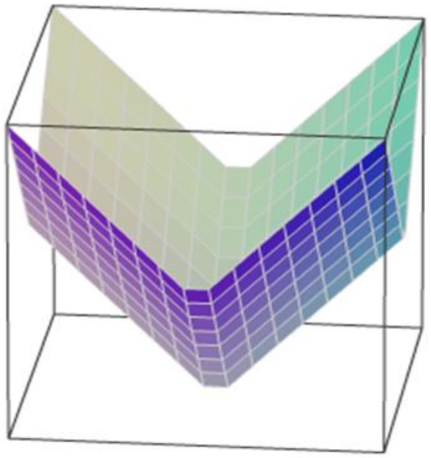
Smooth

β-smooth: $\left|\left|\nabla g(w) - \nabla g(w)\right|\right| \leq \beta\left|\left|w - v\right|\right|$
$\forall w, v \in \mathcal{W},$

$$g(w) - g(v) \leq \nabla g(v)^{\tau}(w - v) + \frac{\beta}{2}\left|\left|w - v\right|\right|^2$$

Lipschitz

L-Lipschitz: $|g(w) - g(v)| \leq L\left|\left|w - v\right|\right|$
$\forall w, v \in \mathcal{W}$



Convergence Rate of GD

Theorem 1: Assume the objective g is **convex** and β-smooth on R^d.
With step size $\eta = \frac{1}{\beta}$, Gradient Descent satisfies:

$$g(x_{T+1}) - g(x^*) \leq \frac{2\beta\|x_1 - x^*\|^2}{T}.$$

Sub-linear Convergence

Theorem 2: Assume the objective g is **α-strongly convex** and β-smooth on R^d.
With step size $\eta = \frac{2}{\alpha+\beta}$, Gradient Descent satisfies:

$$g(x_{T+1}) - g(x^*) \leq \frac{\beta}{2} \exp\left(-\frac{4T}{Q+1}\right) \|x_1 - x^*\|^2,$$

Linear Convergence

where $Q = \frac{\beta}{\alpha}$.

Convergence Rate of Newton's Method

$$f\left(x^k+d^k\right)=f\left(x^k\right)+\nabla f\left(x^k\right)^{\mathrm{T}} d^k+\frac{1}{2}\left(d^k\right)^{\mathrm{T}} \nabla^2 f\left(x^k\right) d^k+o\left(\left\|d^k\right\|^2\right)$$

$$x^{k+1}=x^k-\alpha_k \nabla^2 f\left(x^k\right)^{-1} \nabla f\left(x^k\right).$$

Theorem 3: Suppose the function g is continuously differentiable, its derivative is not 0 at its optimum x^*, and it has a second derivative at x^*, then the convergence is quadratic:

$$||x_t - x^*|| \leq O\left(e^{-2^T}\right)$$

Advantage:
We have a more accurate local approximation of the objective, the convergence is much faster.
Disadvantage:
We need to compute the inverse of Hessian, which is time/storage consuming.

Convergence Rate of GD and SGD

Overall Complexity (ϵ) = Convergence Rate$^{-1}(\epsilon) * $ Complexity of each iteration

	Strongly Convex + Smooth			**Convex + Smooth**		
	Convergence Rate	Complexity of each iteration	Overall Complexity	Convergence Rate	Complexity of each iteration	Overall Complexity
GD	$O\left(\exp\left(-\frac{t}{Q}\right)\right)$	$O(n \cdot d)$	$O\left(nd \cdot Q \cdot \log\left(\frac{1}{\epsilon}\right)\right)$	$O\left(\frac{\beta}{t}\right)$	$O(n \cdot d)$	$O\left(nd \cdot \beta \cdot \left(\frac{1}{\epsilon}\right)\right)$
SGD	$O\left(\frac{1}{t}\right)$	$O(d)$	$O\left(\frac{d}{\epsilon}\right)$	$O\left(\frac{1}{\sqrt{t}}\right)$	$O(d)$	$O\left(\frac{d}{\epsilon^2}\right)$

When data size n is very large, SGD is faster than GD.

Outline

- Optimization theory
- Generalization theory
- Approximation theory

Generalization error upper bound

- Estimation/generalization error: $L_D\left(\hat{f}_{\mathcal{F}}^*\right) - L_D\left(f_{\mathcal{F}}^*\right)$

- Error decomposition

$$L_D\left(\hat{f}_{\mathcal{F}}^*\right) - L_D\left(f_{\mathcal{F}}^*\right)$$
$$= L_D\left(\hat{f}_{\mathcal{F}}^*\right) - L_S\left(\hat{f}_{\mathcal{F}}^*\right) + L_S\left(\hat{f}_{\mathcal{F}}^*\right) - L_S\left(f_{\mathcal{F}}^*\right) + L_S\left(f_{\mathcal{F}}^*\right) - L_D\left(f_{\mathcal{F}}^*\right)$$

$L_S\left(\hat{f}_{\mathcal{F}}^*\right) - L_S\left(f_{\mathcal{F}}^*\right)$: non-positive

$L_S\left(f_{\mathcal{F}}^*\right) - L_D\left(f_{\mathcal{F}}^*\right)$: zero mean

$L_D\left(\hat{f}_{\mathcal{F}}^*\right) - L_S\left(\hat{f}_{\mathcal{F}}^*\right)$: not zero mean, but $< \sup_{f \in \mathcal{F}}\left(L_D(f) - L_S(f)\right)$

Finite Hypothesis Class

Union bound

$$p(A_i) > 1 - \delta$$
$$p(\cap_{i=1}^n A_i) = 1 - p(\cup_{i=1}^n \overline{A_i}) > 1 - n\delta$$

Hoeffding's inequality

$$s_n = x_1 + \cdots x_n, \Delta_i = b_i - a_i, a_i \le x_i \le b_i$$
$$p(s_n - Es_n \ge \boldsymbol{t}) \le \mathbf{exp}\left(-\frac{\mathbf{2t^2}}{\sum_{i=1}^n \mathbf{\Delta}_i^2}\right)$$

Theorem: If $\mathcal{F}$ is finite and $l(f(x), y) \in [0,1]$, we have

(1) For any fixed $f \in \mathcal{F}$ and $\epsilon > 0$,

$$\Pr\left[\left|\hat{L}(f) - L(f)\right| \le \epsilon\right] \ge 1 - 2e^{-2n\epsilon^2}$$

(2) For any $\epsilon > 0$,

$$\Pr\left[\forall f \in \mathcal{F}, \quad \left|\hat{L}(f) - L(f)\right| \le \epsilon\right] \ge 1 - 2|\mathcal{F}|e^{-2n\epsilon^2}$$

(3) With prob. at least $1 - \delta$,

$$\sup_{f \in \mathcal{F}}\left|\hat{L}(f) - L(f)\right| \le \sqrt{\frac{\log|\mathcal{F}| + \log\frac{2}{\delta}}{2n}}$$

What about the infinite hypothesis class

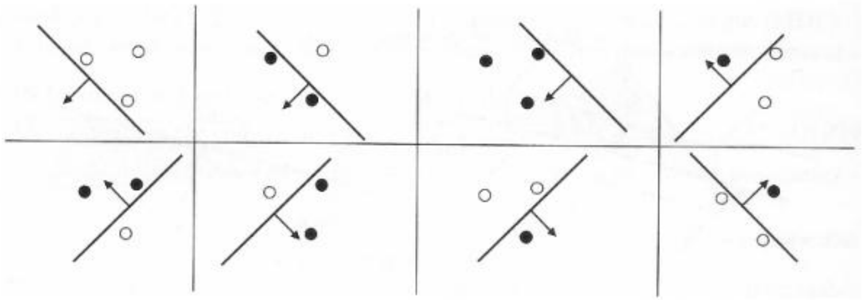
- VC dimension
- Covering number
- Rademacher Average
- Margin bound

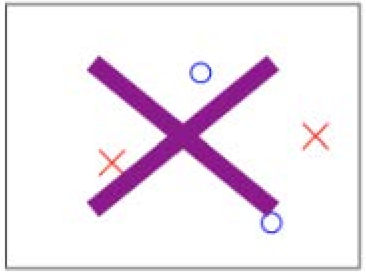
VC dimension

- Growth function
  - The growth function of $\mathcal{F}$ with n points is maximum number of ways that n points can be classified by the hypothesis class $\mathcal{F}$

$$S_{\mathcal{F}}(n) = \sup_{(z_1,\cdots,z_n)} \left|\left\{\left(f(z_1),\cdots,f(z_n)\right): f \in \mathcal{F}\right\}\right|$$

- VC dimension
  - The VC dimension h of a class G is the largest n such that $S_{\mathcal{F}}(n) = 2^n$.





VC dimension

- Generalization bound

Theorem 2 (Vapnik-Chervonenkis). *For any $\delta > 0$, with probability at least $1-\delta$,*

$$\forall g \in \mathcal{G},\ R(g) \le R_n(g) + 2\sqrt{2\frac{\log S_{\mathcal{G}}(2n) + \log\frac{2}{\delta}}{n}}.$$

$$\forall g \in \mathcal{G},\ R(g) \le R_n(g) + 2\sqrt{2\frac{h\log\frac{2en}{h} + \log\frac{2}{\delta}}{n}}.$$

With probability at least $1-\delta$,

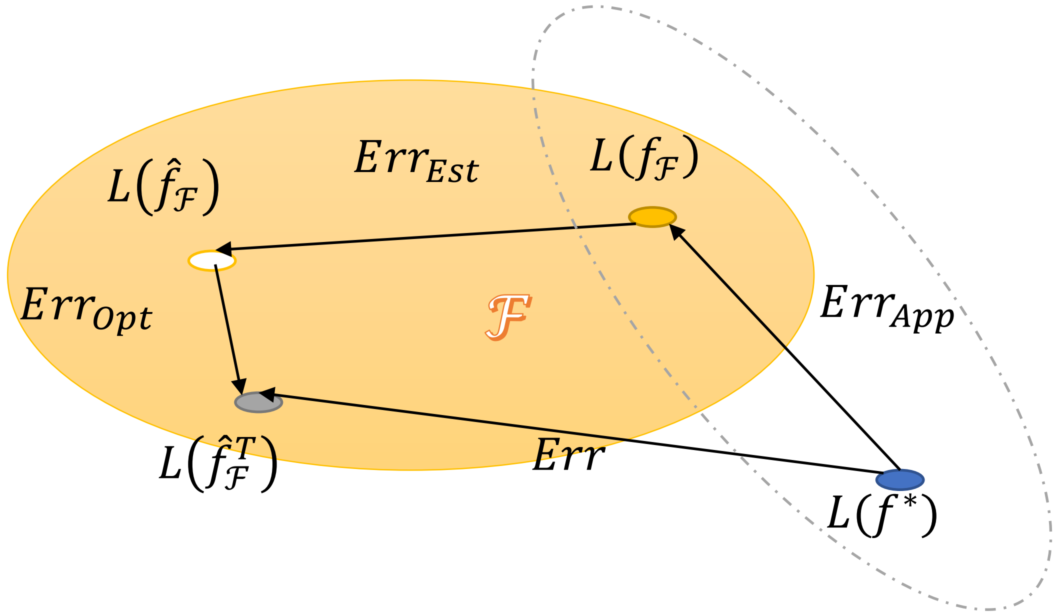
in infinite hypothesis class : $\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f) - R(f)\right| \le 2\sqrt{2\frac{h\log\frac{2en}{h}+\log\frac{2}{\delta}}{n}}$

in finite hypothesis class: $\sup_{f\in\mathcal{F}}\left|\hat{L}(f) - L(f)\right| \le \sqrt{\frac{\log|\mathcal{F}|+\log\frac{2}{\delta}}{2n}}$

Outline

- Optimization theory
- Generalization theory
- Approximation theory

Approximation Error



Continuous function on compact set.

Assume $L^* = L(\boxed{f^*})$,

if $\exists f_m \in \boxed{\mathcal{F}}$ and $\boxed{f_m \to f^*}$,

Then $L(f_\mathcal{F}) - L^* = 0$

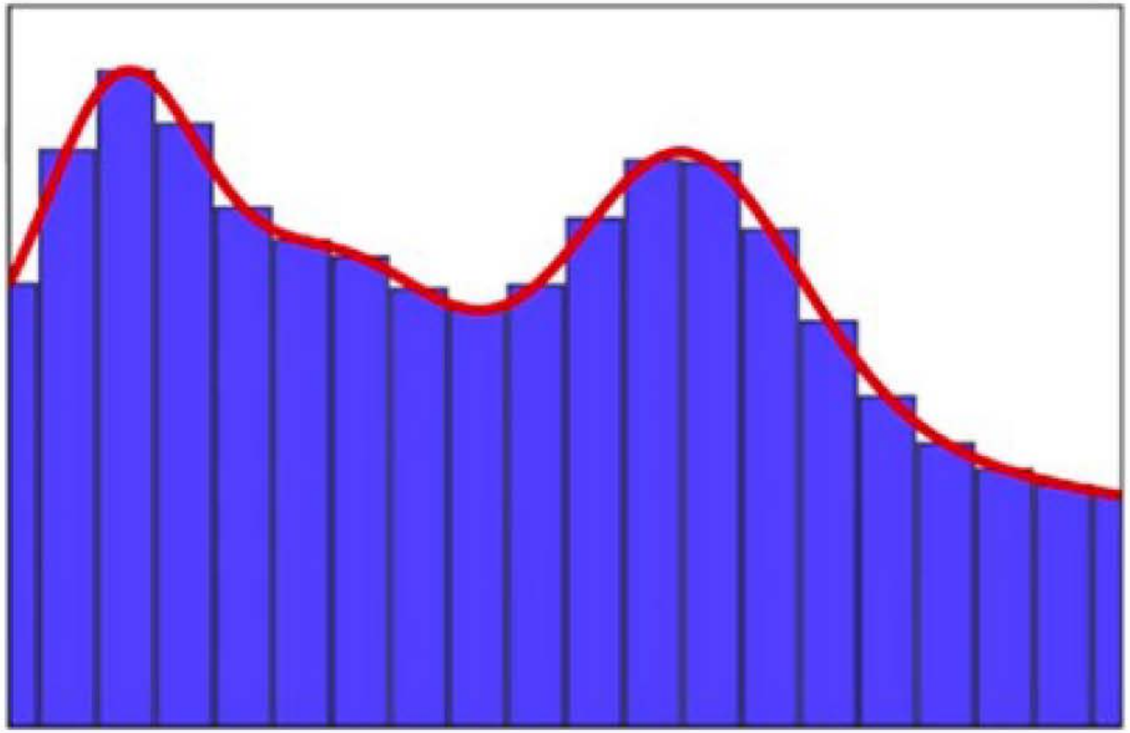
L_∞- convergence

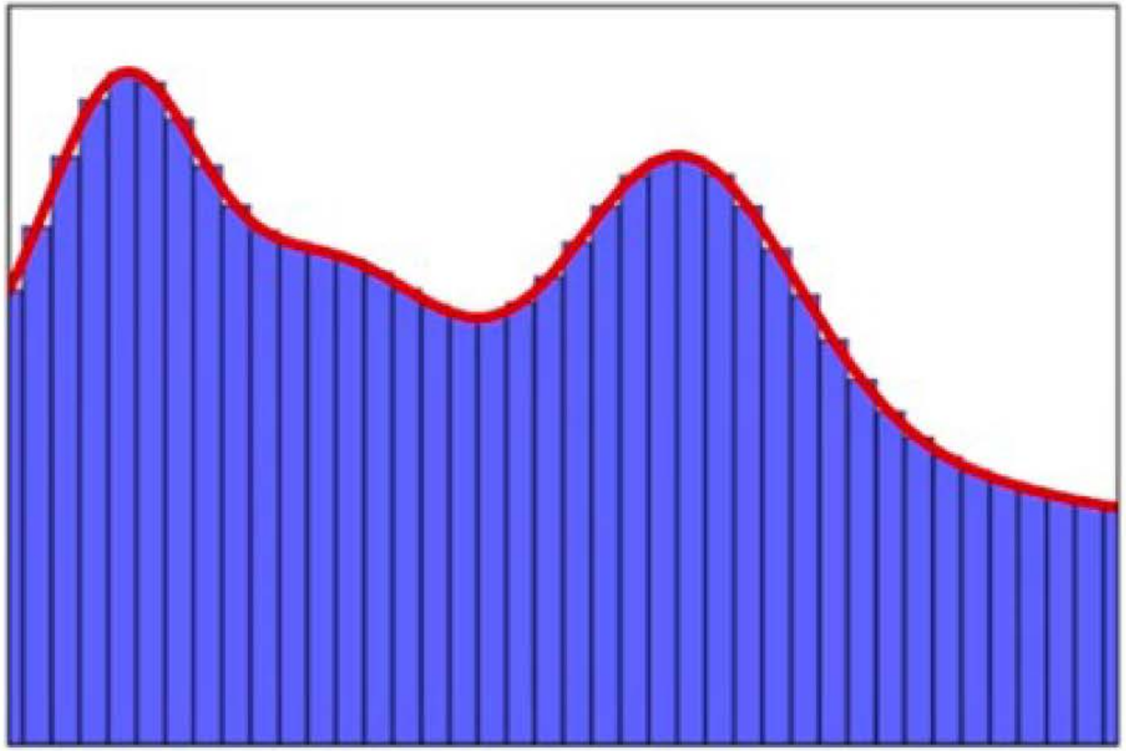
2-layer neural networks with finite hidden units

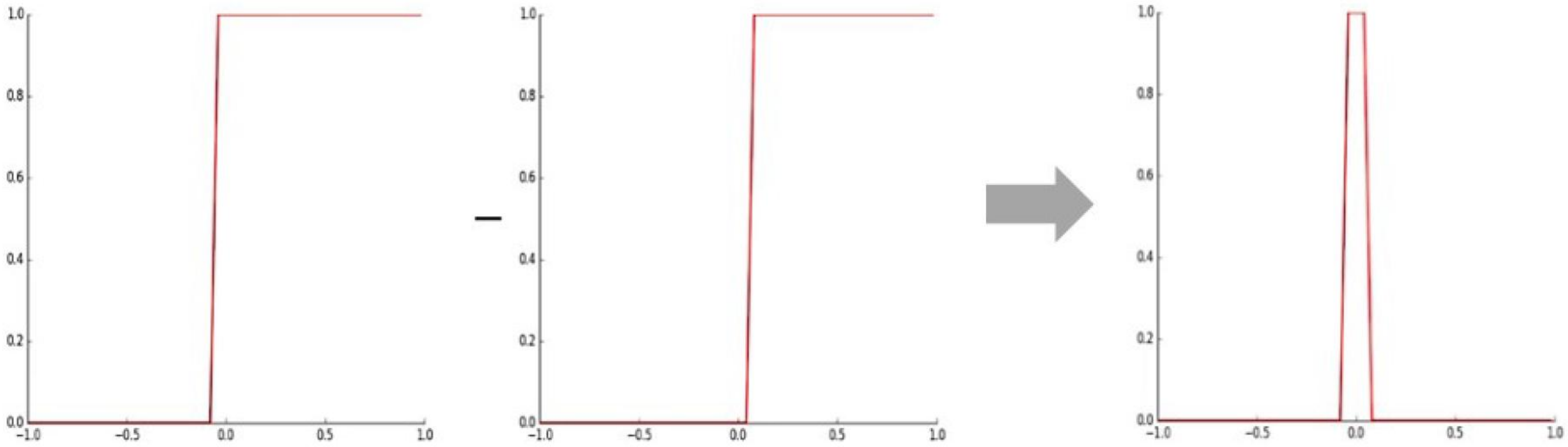
Universal Approximation of Neural Networks

- (Hornik 1989) Feedforward networks with only a single hidden layer can approximate any continuous function **uniformly** on any compact set and any measurable function arbitrarily well.
- For example, $\forall\ f \in C\left([0,1]^d\right), \forall \epsilon > 0, \exists$ 2-layer neural network $NN, s.t.$

$$\forall x \in [0,1]^d, |NN(x) - f(x)| \leq \epsilon.$$






1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0

Overall Picture of Statistical Learning Theory

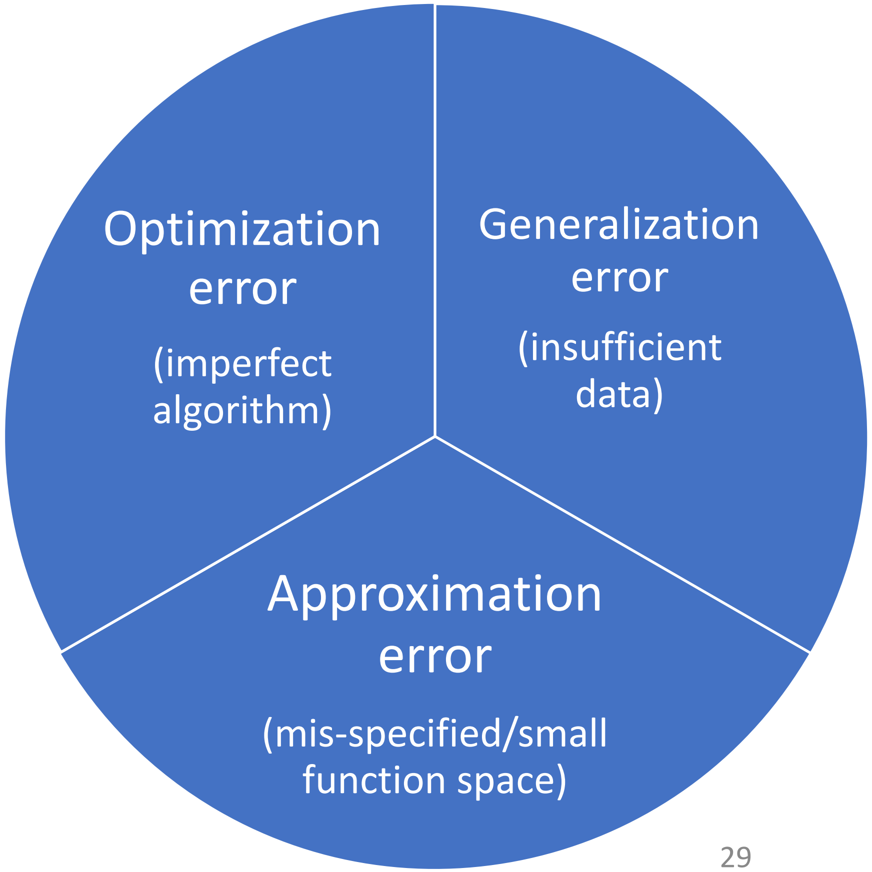
- Training: Find a function f from a function class $\mathcal{F}$ based on training dataset $\mathcal{D}$.

$$\arg\min_{f\in\boldsymbol{\mathcal{F}}} \mathbb{E}_{(x_i,y_i)\in\mathcal{D}}\, L(f(x_i), y_i)$$

- Evaluation: How does f perform on test data: good or not?

$$\mathbb{E}_{(x_i,y_i)\in\boldsymbol{\mathcal{P}}}\, L(f(x_i), y_i)$$

- Where is the gap?
  - $\arg\min$: optimization error → convergence of the algorithm
  - $\mathcal{D} \rightarrow \mathcal{P}$: generalization error → hypothesis space capacity
  - Hypothesis space $\mathcal{F}$: approximation error → hypothesis space capacity



Reference

- 周志华，机器学习，清华出版社
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press
- Vapnik, The Nature of Statistical Learning Theory, Springer, 1999
- Shalev-Shwartz, Shai, and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- Bousquet, Olivier, Stéphane Boucheron, and Gábor Lugosi. "Introduction to statistical learning theory." Springer, Berlin, Heidelberg, 2003.
- Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators[J]. Neural networks, 1989, 2(5): 359-366.

Thanks!

http://web.ee.tsinghua.edu.cn/wqzhang