Introduction to Statistical Learning

Nicolas Vayatis

Session 6 - Statistical analysis of mainstream ML algorithms

Part II - Feedforward Neural Networks, Bagging, Random Forests

Overview of Chapter 3

0. (Consistency of local methods : k-NN, decision trees, local averaging)

1. Consistency of global methods
   a. (Boosting)
   b. Support Vector Machines
   c. Neural networks

2. Consistency of ensemble methods
   + Bagging, Random Forests

Main theoretical objectives of the course

- Take a well-known ML algorithm which operates in $\mathcal{F}$: it produces a (random) sequence of decision rules $(\widehat{f}_n)_{n\geq 1}$ in $\mathcal{F}$. Then show :
- **Convergence of estimation error** :

$$L(\widehat{f}_n) \to \inf_{\mathcal{F}} L \text{ almost surely as } n \to \infty \ ,$$

- **Upper bounds :** with probability at least $1-\delta$, there exists some constant c such that :

$$L(\widehat{f}_n) - \inf_{\mathcal{F}} L \leq C(\mathcal{F}, n) + c\sqrt{\frac{\log(1/\delta)}{n}} \ ,$$

where $C(\mathcal{F}, n)) = O(1/\sqrt{n})$ after processing some complexity/stability measure

Chapter 3

1. Consistency of global methods

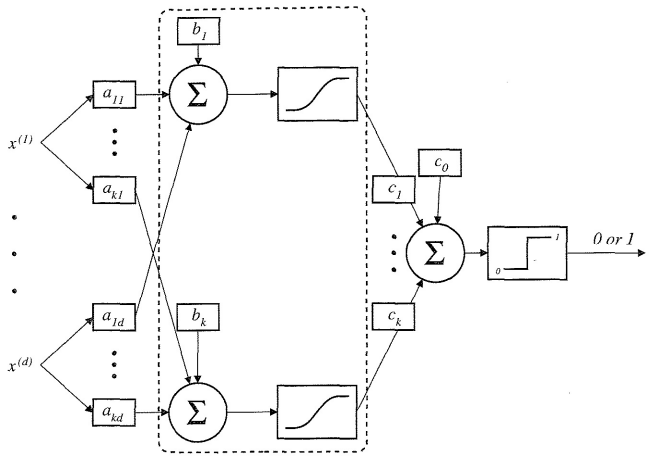
c. Neural networks

Historical perspective on neural networks

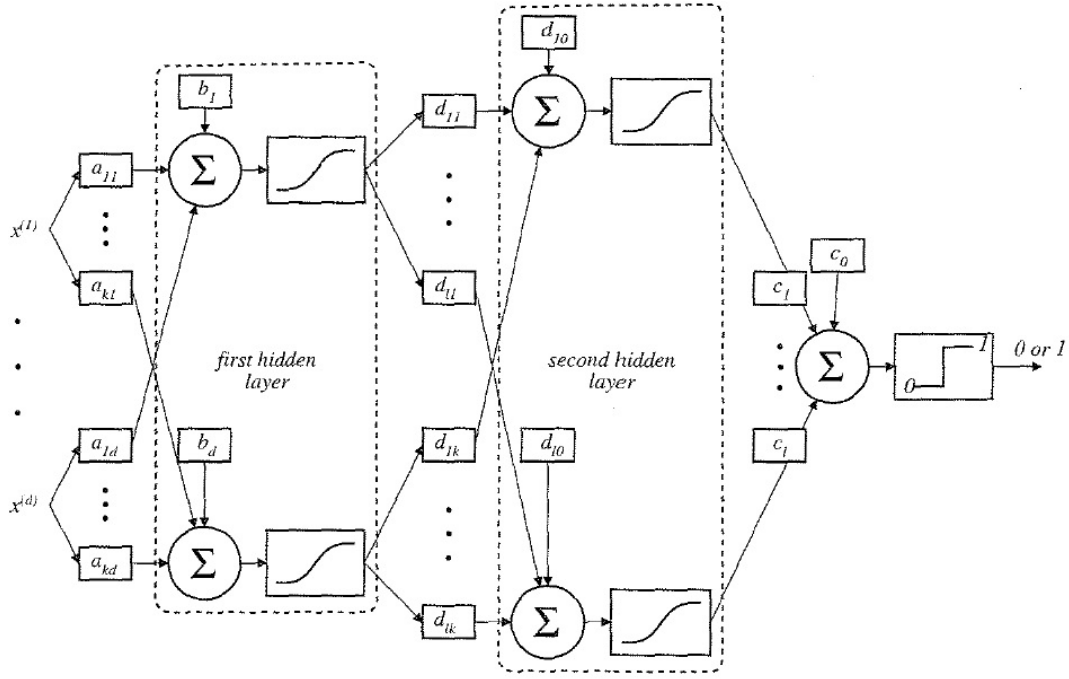
- Cybernetics (1940s-1960s)
  - Achievement : modeling and training one neuron
  - Key algorithm : Linear Perceptron
  - Paper : Rosenblatt (1958)
- Connectionism (1980s)
  - Achievement : training one or two hidden layers
  - Key algorithm : Backpropagation
  - Paper : Rumelhart-Hinton-Williams (1986)
- Deep Learning (2007-....)
  - Achievement : training multiple layers of representation
  - Key algorithm : Stochastic gradient
  - Papers : Hinton (2006), Bengio-LeCun (2007)

Principle of feedforward neural networks
Single-layer

- Form of classifier implemented by a one-hidden layer perceptron : $g = \mathrm{sgn}(f - 1/2)$, where :
  - $f(x) = c_0 + \sum_{i=1}^{p} c_i \cdot \sigma \circ \psi_i(x)\ , \quad \forall x \in \mathbb{R}^d$
  - σ is a sigmoid,
  - the ψ_i's are linear : $\psi_i(x) = b_i + \sum_{i=1}^{p} a_{i,j} x^{(j)}$



Principle of feedforward neural networks
Multiple-layer

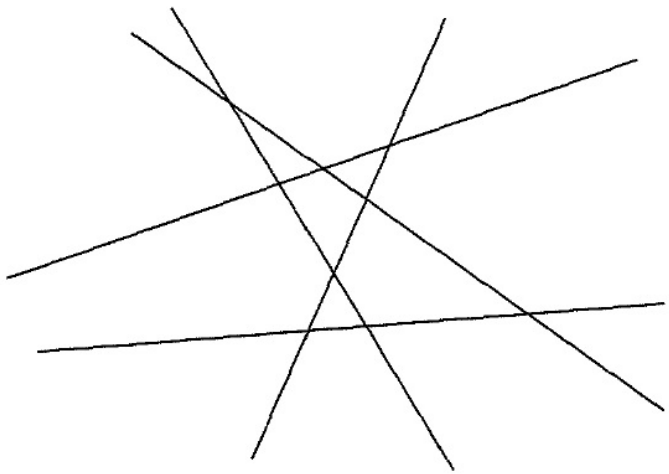


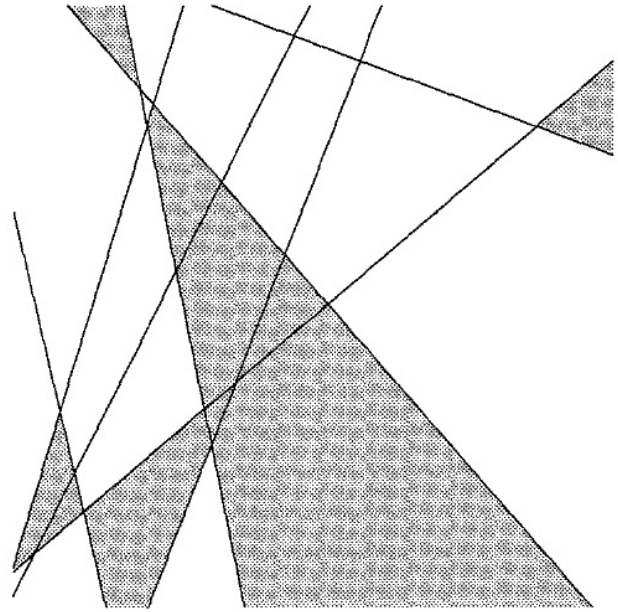
- Hypothesis space : functions of the form

$$f(x, \theta) = \sigma_m \circ A_m \circ \sigma_{m-1} \circ \ldots \circ A_2 \circ \sigma_1 \circ A_1 x$$

where $\theta = (A_1, \ldots, A_m)$ and $A_1, \ldots, A_m$ are matrices

Intuition for complexity analysis : linear arrangements





Arrangements : definition and key property

Definition.

A simple arrangement *is a collection $\mathcal{A}$ of hyperplanes in dimension d such that :*
(i) any d hyperplanes of $\mathcal{A}$ have a unique point in common, and
(ii) any $d+1$ hyperplanes of $\mathcal{A}$ have no point in common.

Theorem. (Edelsbrunner (1987))

The number of cells of a simple arrangement with cardinality $|\mathcal{A}| = M$ is given by :

$$\begin{cases} 2^M & \text{if } d \geq M \\ \sum_{i=1}^{d} \binom{M}{i} & \text{if } d < M \end{cases}$$

Consistency result for ERM on arrangements

Theorem. (Devroye, Györfi, Lugosi (1996))

The ERM classifier $\hat{g}_n^M$ on all possible arrangements of size at most M has expected error which converges to the Bayes error :

$$\mathbb{E}\big(L\big(\hat{g}_n^M\big)\big) \to L^*$$

for all distributions, as soon as $M \to \infty$ and $M = o(n/\log n)$.

The key argument of the proof relies on exact computation of shattering coefficient :

$$\gamma(\mathcal{G}, n) = \big(2(n^d + 1)\big)^M$$

Comments

- More work needed to deal with data-driven arrangements (with or without optimization)
- One hidden layer neural nets are universal approximators (denseness results)
- Upper bounds on VC dimension available
- More theory in [Devroye, Lugosi, and Györfi, 1996] !

Consistency result for L^1-error minimization

Theorem 30.9. (LUGOSI AND ZEGER (1995)). *Let σ be an arbitrary sigmoid. Define the class $\mathcal{F}_n$ of neural networks by*

$$\mathcal{F}_n = \left\{ \sum_{i=1}^{k_n} c_i \sigma(a_i^T x + b_i) + c_0 : \ a_i \in \mathcal{R}^d, b_i \in \mathcal{R}, \sum_{i=0}^{k_n} |c_i| \leq \beta_n \right\},$$

and let ψ_n be a function that minimizes the empirical L_1 error

$$J_n(\psi) = \frac{1}{n} \sum_{i=1}^{n} |\psi(X_i) - Y_i|$$

over $\psi \in \mathcal{F}_n$. If k_n and β_n satisfy

$$\lim_{n\to\infty} k_n = \infty, \quad \lim_{n\to\infty} \beta_n = \infty, \quad \textit{and} \quad \lim_{n\to\infty} \frac{k_n \beta_n^2 \log(k_n \beta_n)}{n} = 0,$$

then the classification rule

$$g_n(x) = \begin{cases} 0 & \textit{if } \psi_n(x) \leq 1/2 \\ 1 & \textit{otherwise} \end{cases}$$

is universally consistent.

Rademacher complexity of neural networks (Setup)

Consider an IID sample $X_1, \ldots, X_n$ of observations over some space $\mathcal{X}$ and $\mathcal{F}_0$ is a set of real-valued functions over $\mathcal{X}$ that includes the zero function.

Assume $\psi : \mathbb{R} \to \mathbb{R}$ is k-Lispchitz and define, for fixed positive real numbers V and B :

- a one-layer network as :

$$\mathcal{F}_1 = \left\{ x \mapsto \psi\left(v + \sum_{j=1}^{m} w_j f_j(x) \right) : |v| \leq V, \|w\|_1 \leq B, f_j \in \mathcal{F}_0 \right\}$$

- a p-layer network as (iterative definition with fixed layer size) :

$$\mathcal{F}_p = \left\{ x \mapsto \psi\left(v + \sum_{j=1}^{m} w_j f_j(x) \right) : |v| \leq V, \|w\|_1 \leq B, f_j \in \mathcal{F}_{p-1} \right\}$$

Rademacher complexity of neural networks (Exercise)

Prove the following upper bounds on the empirical Rademacher average :

1. $\hat{R}_n(\mathcal{F}_1) \leq k\left(\frac{V}{\sqrt{n}} + 2B\hat{R}_n(\mathcal{F}_0)\right)$.
2. Assume in addition that $\psi(-u) = -\psi(u)$ and $k = 1$ then show that on $\mathcal{X} = \{x \in \mathbb{R}^d,\ \|x\|_\infty \leq 1\}$:

$$\hat{R}_n(\mathcal{F}_p) \leq \frac{1}{\sqrt{n}}\left(B^p\sqrt{2\ln(2d)} + V\sum_{l=1}^{p-1} B^l\right) .$$

Chapter 3

2. Consistency of ensemble methods

Bagging and Random Forests

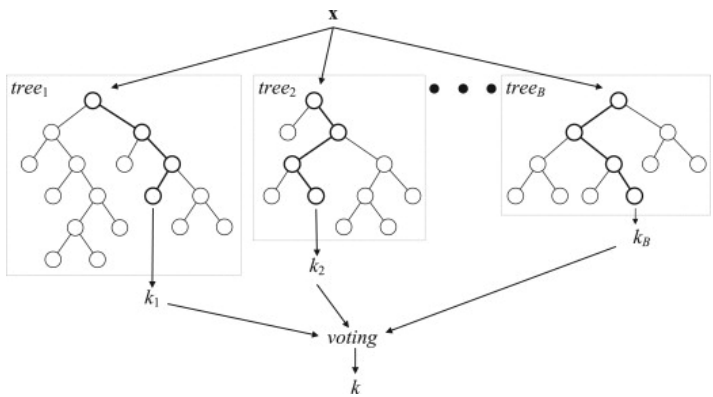
Ensemble methods
Starting point

- Consider we already have a machine learning algorithm with reasonable performance that we want to improve, e.g. decision tree, k-NN, SVM, ...
- The idea of the ensemble is to generate different functions from the same training data and the same hypothesis space
- In the illustration coming next and most of the discussion, the basic hypothesis space is the one with decision trees obtained with orthogonal splits (such splits are called decision stumps).

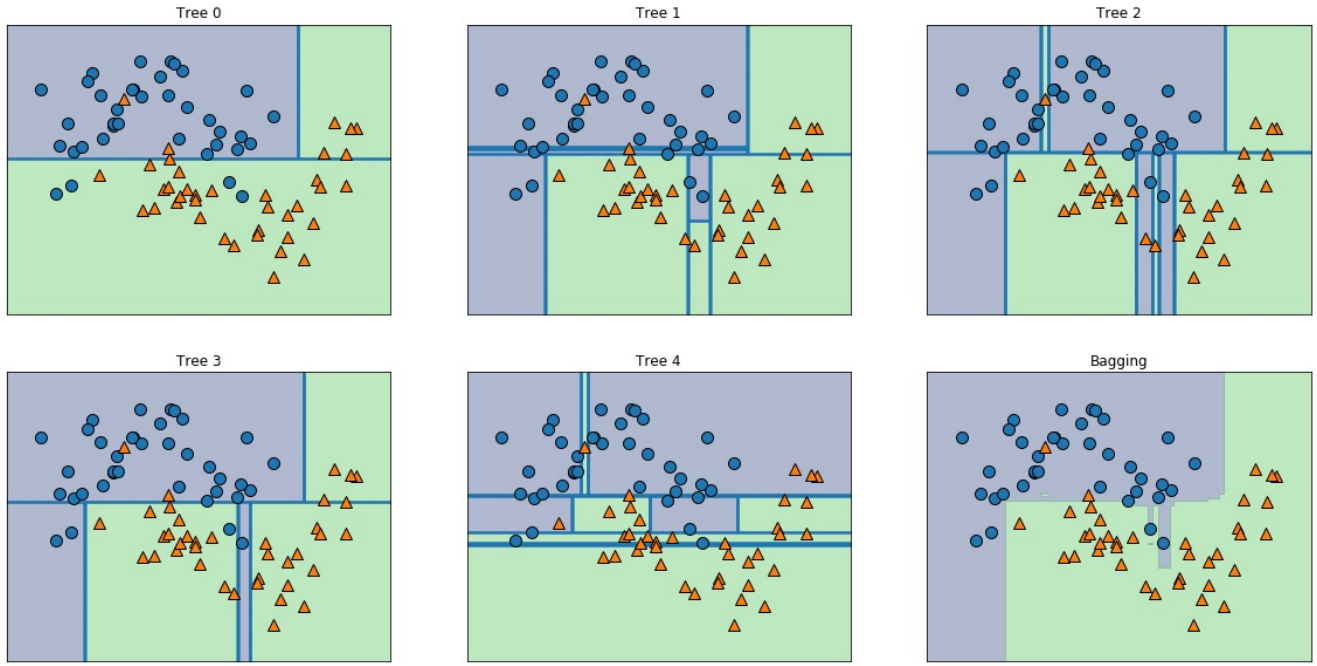
Ensembles of decision trees
General principle

Randomization (training) + averaging (decision rule)

- Generate a collection of *weak* predictors (ensemble) obtained with a basic Machine Learning algorithm (e.g. decision tree)
- For every point x, compute their individual predictions
- Take an average or a majority vote of the individual predictions to determine the prediction of the ensemble



Ensembles of decision trees
Resulting classifier



Ensembles of decision trees
Three popular methods

- Bagging (Breiman, 1996)
- Random forests (Amit-Geman, 1997 ; Breiman, 2000)
- ... and also Boosting (Freund-Schapire, 1996) just seen before

Randomized rules (1/2)

- For a given sample $D_n = \{(X_i, Y_i) \ : \ i = 1, \ldots, n\}$
- Introduce $\mathcal{Z}$ a measurable space and Z a random variable over $\mathcal{Z}$
- Conditionally on the sample D_n and on (X, Y), draw independent sequences $Z_1, \ldots, Z_B$ of B copies of Z
- Design a pool of decision rules $\widehat{g}_{n,b}(x) = \widehat{g}_{n,b}(x, Z_b, D_n)$ for $b = 1, \ldots, B$

Randomized rules (2/2)

Two options :

- *Voting classifier* :

$$\widehat{g}_n^B(x) = \mathbb{I}\left\{\sum_{b=1}^{B} \widehat{g}_{n,b}(x) > B/2\right\} \ ,$$

- *Averaging classifier* (which is not a randomized classifier) :

$$\overline{g}_n^B(x) = \mathbb{I}\left\{\mathbb{E}_Z \widehat{g}_n(x, Z) > 1/2\right\} \ .$$

Bagging - Breiman, 1996

- Randomization through bootstrap replicates of D_n
- Randomized rule through bagging :

$$g_n(x, Z, D_n) = g_n(x, D_n(Z))$$

- ... and $D_n(Z) = \{(X_i^*, Y_i^*) \ : \ i = 1, \ldots, n\}$ where the points are drawn through random sampling from D_n
- Typical sampling is sampling with replacement and $|D_n(Z)| = n$

Bagging - a consistency result

- Special case with subsampling and without replicates in the bootstrap sample
- $|D_n(Z)| = N \leq n$ and ...
- ... we assume $N \sim \mathrm{Bin}(n, q_n)$
- ... therefore $q_n = \mathbb{P}((X_i, Y_i) \in D_n(Z))$
- Consistency of both voting classifier and averaging classifier under assumptions :
  - $\{g_n\}$ sequence of classifiers that is consistent for P
  - $nq_n \to \infty$ when $n \to \infty$

Bagging can render consistent rules that are inconsistent

- Biau, Devroye and Lugosi (2008) have considered bagging 1-NN
- 1-NN is consistent if and only if $L^* \in \{0, 1/2\}$
- Bagging averaged 1-NN classifier is consistent for any P if and only if $q_n \to 0$ and $nq_n \to \infty$ when $n \to \infty$
- Proof follows the lines of Stone theorem (cf. Devroye, Györfi, Lugosi (1996))

Plain Random Forests - Breiman 2001

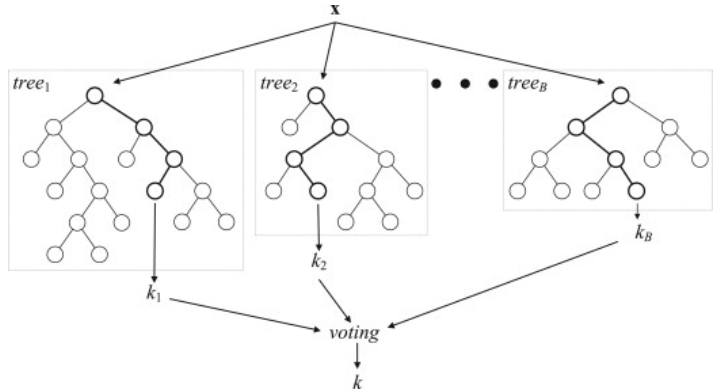
- Main ingredient 1 - Build randomized tree classifiers

  $\Rightarrow$ perform splits in random subspaces (parameter `mtry` in `R`), use trees with small depth

- Main ingredient 2 - Booststrap samples of data set D_n

  $\Rightarrow$ number of trees has limited impact above 100

- Main ingredient 3 - Aggregation through averaging classifier



- Credits to Amit and Geman (1997) ! !

RF consistency - Biau, Devroye, Lugosi (2008)

- Simplified model : the *purely random forest*
- Start with a tree classifier with rectangular cells with orthogonal splits over $[0,1]^d$ and build randomized versions as follows :
  - Draw a leave according to a uniform distribution over the set of leaves of the tree
  - Draw one split variable among the d dimensions of input space with a uniform
  - Position the split at random (uniform distribution again)
  - Repeat k times splitting of terminal cells of the tree
- Result : The averaged classifier is consistent if $k_n \to \infty$ and $k_n/n \to 0$ when $n \to \infty$
- Variant developed in Biau (2012) for regression setup with non-uniform split variable selection and fixed splitting value

Take-home message (on aggregation)

- Aggregation of consistent rules leads to consistent rules
- Aggregation of inconsistent rules may lead to consistent rules
- Conjecture : Breiman's original RF belongs to this case
- 'Oldies' like Stone theorem are useful!!
- Lots of known *'unknowns'*!

Cherry on the cake : a mirror descent algorithm

Motivations

- Ensembles are defined by a set of predictors and their weights
- For bagging and random forests, weights are uniform
- In boosting weights reflect individual performance but are determined iteratively
- Knowing the predictors, what are the optimal weights ?

Problem formulation

- Convex risk minimization under an ℓ_1 constraint
- Set $w \mapsto A(w)$ to be the convex objective
- $w \in \mathbb{R}_+^T$ is the weight vector s.t. $\|w\|_1 = C$, for some $C > 0$
- Approach : fast iterative algorithm to optimize w over the C-simplex

Setup for learning and optimization

- Vector of predictors $H(x) \in \mathbb{R}^T$
- Loss function $\varphi : \mathbb{R} \to \mathbb{R}_+$ is convex with derivative φ' (monotone version)
- Optimization objective $A(w) = \mathbb{E}\big(\varphi(Yw^T H(X))\big)$ (not necessarily accessible)
- "Observable" : stochastic subgradient at observation (X_i, Y_i)

$$u_i(w) = \varphi'(Y_i w^T H(X_i)) Y_i H(X_i) \in \mathbb{R}^T$$

- Insight : (fast) gradient descent in the dual space

The "mirror"

- Potential or proxy function V convex over $E = (\mathbb{R}^T, \|\|_1)$
- Dual space : $E^* = (\mathbb{R}^T, \|\|_\infty)$
- C-simplex $\Delta(C) = \{w \in \mathbb{R}_+^T \ : \ \|w\|_1 = C\}$
- The β-convex conjugate of V is defined as : for any $z \in E^*$, for any $\beta > 0$

$$W_\beta(z) = \sup_{w \in \Delta(C)} (w^T z - \beta V(w))$$

- Here : special case where V is the entropy proxy function, $\forall w \in \Delta(C)$

$$V(w) = C \ln(T/C) + \sum_{j=1}^{T} w^{(j)} \ln w^{(j)}$$

Mirror descent algorithm for convex risk optimization

For $i = 1, \ldots$

- "Temperature" parameter : Let $\beta_i = \beta_0 \sqrt{i+1}$
- Gradient updates $\zeta_i = \zeta_{i-1} + u_i(w_{i-1})$
- Mirror step : $w_i = -\nabla W_{\beta_i}(\zeta_i)$
- Averaging step : $\hat{w}_{i+1} = \hat{w}_i - \frac{1}{i+1}(\hat{w}_i - w_i)$

NB : constant stepsize equal to one in this version

Upper bound on numerical convergence

- Consider a positive and convex loss φ and $T \geq 2$, then

$$\mathbb{E}(A(\hat{w}_n)) - \min_{w \in \Delta(C)} A(w) \leq \kappa(\varphi, C) \frac{\sqrt{(n+1)\ln T}}{n}$$

  where $\kappa(\varphi, C)$ is explicit and tight

- Classification case : $\kappa(\varphi, C) = 2C \sup_{x \in [-C,C]} |\varphi'(x)|$

Background on mirror descent algorithms

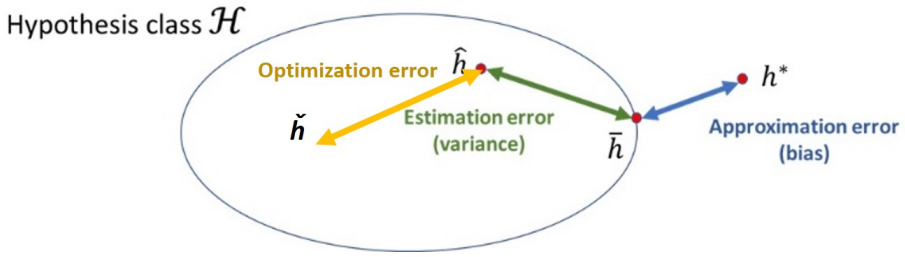
- A.S. Nemirovski and D.B. Yudin. Problem Complexity andMethodEfficiency in Optimization. Wiley-Interscience, 1983.
- B.T. Polyak and A.B. Juditsky. Acceleration of stochastic approxima-tion by averaging. SIAM J. Control Optim., 30(4) :838–855, 1992
- A. Beck and M. Teboulle. Mirror descent and nonlinear projected sub-gradient methods for convex optimization. Operations Research Letters,31 :167–175, 2003.
- J. Kivinen and M.K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. Information and computation, 132(1) :1–64, 1997.
- A. Juditsky, A. Nazin, A. Tsybakov, and N. Vayatis. Recursive Aggregation of Estimators via the Mirror Descent Algorithm with averaging. Problems of Information Transmission , 41(4) : 368-384, 2005.

Beyond this course

Theory of ML : From 2000 to Today

- Consistency/rates/fast rates of convergence of the estimation error for regularized learning methods : SVM (Steinwart, 2005), Boosting (Lugosi-Vayatis, 2004)(Zhang, 2004), general surrogate losses (Bartlett, Jordan, McAuliffe, 2006)
- Theory of ranking and scoring algorithms with advanced concentration inequalities (Clémençon-Lugosi-Vayatis, 2008)
- Other tracks with theoretical advances : multiclass classification, ranking, multitask learning
- Other setups : online learning, learning view on game theory, transfer learning, active learning...

Bias-variance revisited



$$\begin{aligned}\mathcal{E} &= \mathbb{E}\left[E(f^*_{\mathcal{F}}) - E(f^*)\right] + \mathbb{E}\left[E(f_n) - E(f^*_{\mathcal{F}})\right] + \mathbb{E}\left[E(\tilde{f}_n) - E(f_n)\right] \\ &= \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{est}} + \mathcal{E}_{\text{opt}}.\end{aligned}$$

Trade-off wrt : Search space $\mathcal{F}$, sample size n, numerical tolerance ρ

		$\mathcal{F}$	n	ρ
$\mathcal{E}_{\text{app}}$	(approximation error)	$\searrow$		
$\mathcal{E}_{\text{est}}$	(estimation error)	$\nearrow$	$\searrow$	
$\mathcal{E}_{\text{opt}}$	(optimization error)	$\cdots$	$\cdots$	$\nearrow$
T	(computation time)	$\nearrow$	$\nearrow$	$\searrow$

[The trade-offs of Large Scale Learning, L. Bottou, O. Bousquet, 2011]

Shallow learning vs. Deep Learning

Mysteries about deep learning

- Approximation : deep better than shallow ?
- Optimization : nonconvex with millions of dimensions (!)
- Overfitting : huge complexity

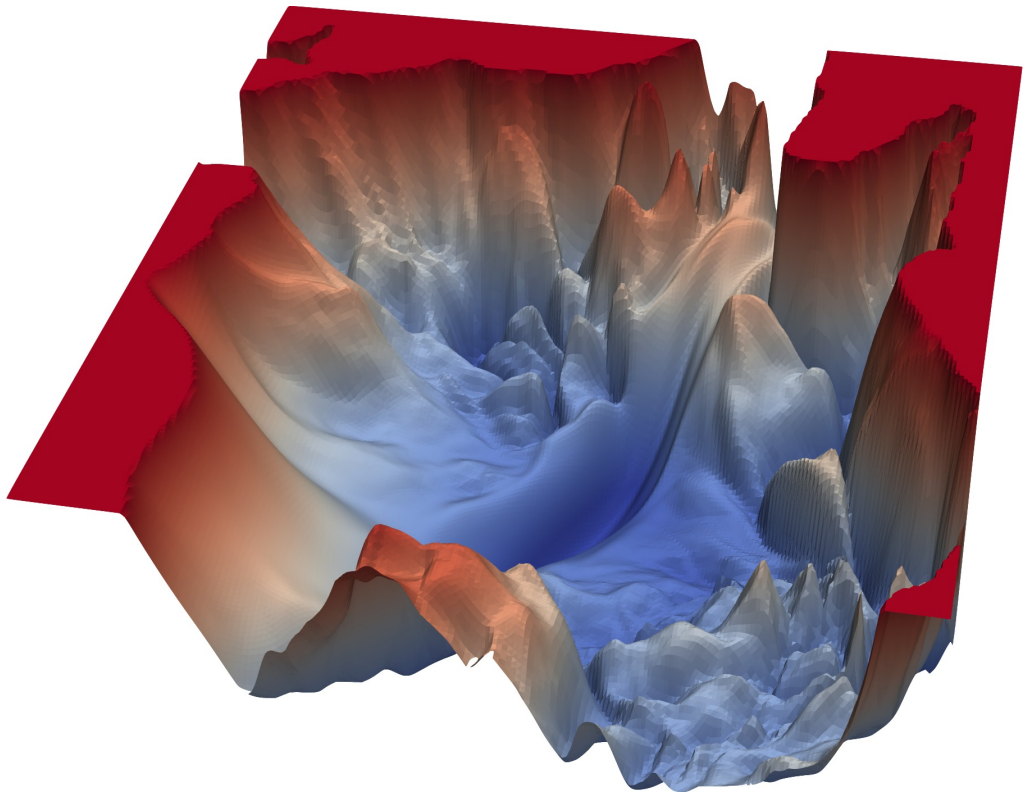
Facts about approximation theory

Comparison of Shallow vs. Deep Networks

- Poggio and Liao (2018) : approximation of compositional functions
- Liang and Srikant (2017) : approximation of polynomial functions
- Similar findings :
  " the number of neurons needed by a shallow network to approximate a function is exponentially larger than the corresponding number of neurons needed by a deep network for a given degree of function approximation. "

The loss landscape of Deep Learning

View on a 56-layer neural network without skip-connection



From [Visualizing the Loss Landscape of Neural Nets,
H. Li, Z. Xu1, G. Taylor, C. Studer, T. Goldstein, 2018]

Facts about optimization in Deep Learning

- Under certain conditions, no poor local minima

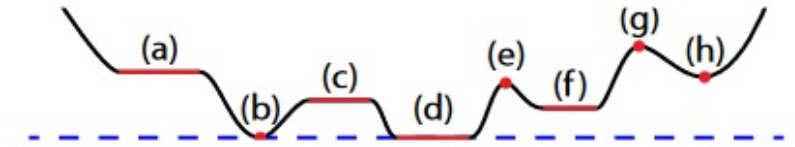


Figure 1. Example critical points of a non-convex function (shown in red). (a,c) Plateaus. (b,d) Global minima. (e,g) Local maxima. (f,h) Local minima.

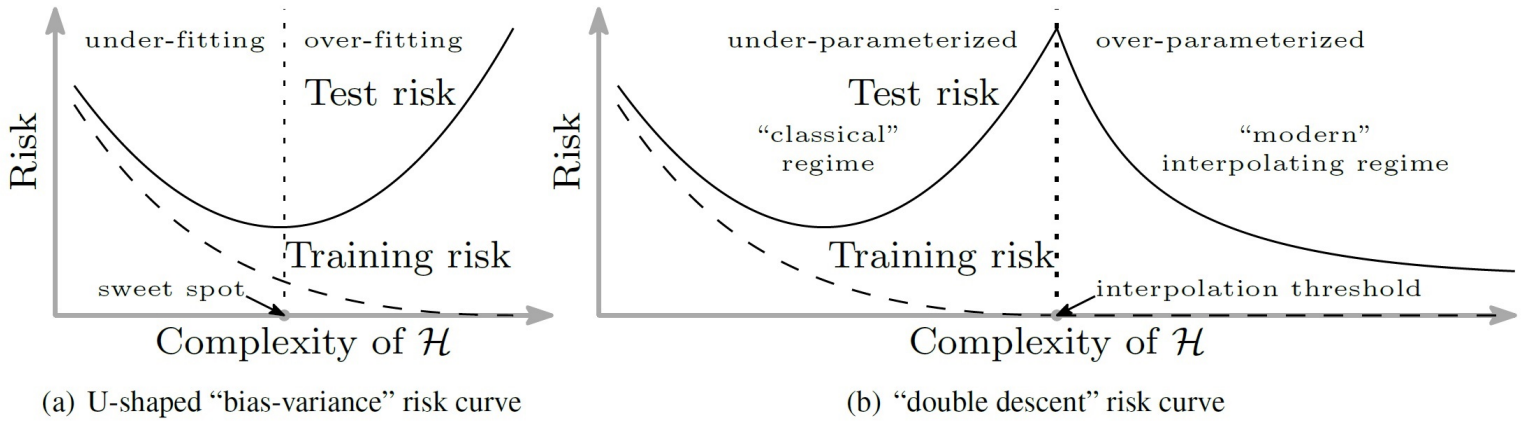
- SGD avoids bad critical points
- Larger networks are better behaved (local minima are global)

References :

Soudry and Carmon (2016), "No bad local minima : Data independent training error guarantees for multilayer neural networks".
Kawaguchi (2016), "Deep learning without poor local minima".
Haeffele and Vidal (2017), "Global optimality in neural network training".
Janzamin, Sedghi, and Anandkumar (2015), "Beating the perils of non-convexity : Guaranteed training of neural networks using tensor methods".
Panageas and Piliouras (2016), "Gradient descent only converges to minimizers : Non-isolated critical points and invariant regions".
Brutzkus, Alon et al. (2017), "SGD Learns Over-parameterized Networks that Provably Generalize on Linearly Separable Data".

The theory of a double descent risk curve

How Deep Learning (and random forests) avoid overfitting



(a) U-shaped "bias-variance" risk curve

(b) "double descent" risk curve

From [Reconciling modern machine learning and the bias-variance trade-off, M. Belkin, D. Hsu, S. Ma, S. Mandal, 2018]

If you liked this course, you will/should like...

- Fondements théoriques du deep learning
- Sequential learning
- Kernel methods for machine learning
- Représentations parcimonieuses
- Graphs in machine learning
- Apprentissage pour les séries temporelles

Thank you !