Deep Learning

Lecture 1: Fundamentals of Learning

University of Agder,
Kristiansand, Norway

Prepared by: Hadi Ghauch*, Hossein S. Ghadikolaei†

* Telecom ParisTech

† Royal Institute of Technology, KTH

https://sites.google.com/view/fundl/home

April 2019

Outline

Outline

Logistics

- number of credits TBD
- 10 lectures, $\sim$ 2 hours/lecutre:
  Fundamentals (lectures 1-6), Special Topics (lectures 7-10)
- Student groups for homework
  4-5 students per group
  Deadline for groups formation: end of lecture 2
- All homeworks are available at start of course
  the deadline for all **HWK submission is Jun 7th, 2019**
- Paper presentations

Logistics cont.

- Slides available before each lecture
- Lectures recorded and uploaded on YouTube afterward
- 3 homework and 1 paper presentation
- No homeworks nor presentation required for students auditing the course
- Email: hadi.ghauch@telecom-paristech.fr, hshokri@kth.se
- Course website: `https://sites.google.com/view/fundl/home`
- Please **use this form for registration:** `https://goo.gl/forms/oD0OaALGvVy1P31y2`

Course Organization

Part I: Fundamentals

- Lecture 1: Fundamentals of Learning
- Lecture 2: Learning and Convex Optimization
- Lecture 3: Large-scale Convex learning
- Lecture 4: Non-convex optimization for learning (part 1)
- Lecture 5: Non-convex optimization for learning (part 2)
- Lecture 6: Fundamentals of Deep Neural Networks

Break: May 1st - May 26th

- use break to work on homeworks and paper presentation

Part II: Special topics and Application

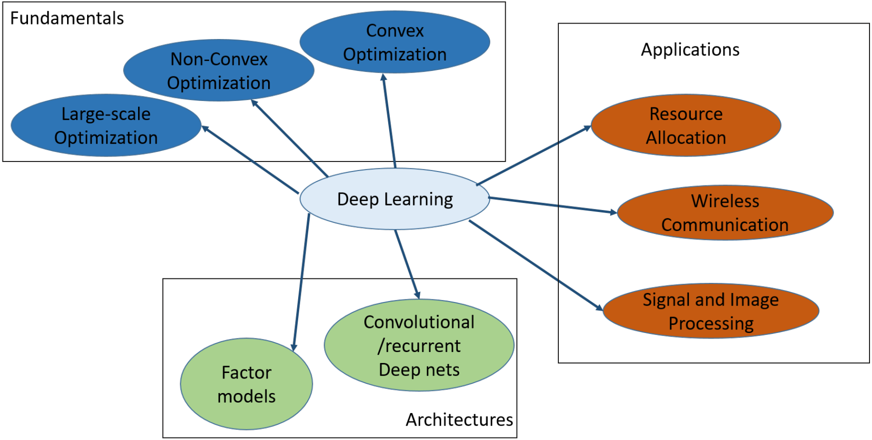
- Lecture 7: Large-scale training of Deep Neural Networks
- Lecture 8: Architectures for Deep Neural Networks
- Lectures 9, 10: Student Presentations

Paper Presentations

- *PhD students:* will divide in teams to present one paper. Each paper presentation is 30 min: 20 min for presentation + 10 min for questions session (lead by another team member who acts as the session chair). This part is mandatory for PhD students.
  PhD students are encouraged to find their own paper about the topic they choose.
- Anyone may check the course web for some suggested topics

Outline

Keywords and Contents



Motivation for Deep Learning

Deep Learning is everywhere!

Outline

Taxonomy of Machine Learning

- *Supervised Learning:* learning from labeled data:
  training samples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$

- regression, classification, deep learning, etc

- *Unsupervised Learning:* learning from unlabeled data
  training samples $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^N$

- clustering, matrix factorization, etc

- *Reinforcement learning:* (online learning)
  learning by interacting with an unknown environment (modeled by a Markov decision process)

- Q-learning

Problem Setting

Empirical Risk Minimization problem

$$\min_{\boldsymbol{w} \in \mathcal{W}} \; \ell(\boldsymbol{w}) := \frac{1}{N} \sum_{i=1}^{N} \ell_i \left((\boldsymbol{x}_i, y_i); \boldsymbol{w} \right)$$

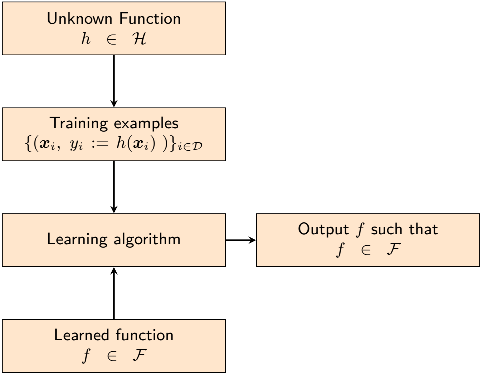
- ℓ: loss function, ℓ_i: loss of sample i
- $\boldsymbol{w}$: the model
  Linear/logistic regression, SVMs: $\boldsymbol{w}$ vector in $\mathbb{R}^d$
  Nonlinear regression (DNN): $\boldsymbol{w}$ is set of all weights
- N: size of training set
  N large $\Rightarrow$ large-scale learning
- Training set: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$
  $\boldsymbol{x}_i$ (resp y_i) feature vector (resp label) of sample i

Outline

1. Logistics
2. Course Contents
3. Machine Learning
4. Statistical Learning Theory
5. Background and Definitions

Statistical Learning Theory

Statistical Learning Perspective on Supervised Learning ?



$\mathcal{H}$: loss class of hidden function, h (subspace of Banach/Hilbert space)
$\mathcal{F}$: loss class of learned function, f (subspace of Banach/Hilbert space)
e.g., for a linear classification task:

$$\mathcal{F} = \mathcal{H} = \{\ \mathbb{I}[\ \sum_{j}^{d} w_j x_j \geq c\ ]\ |\ w_j \in \mathbb{R}_{-0}, j \in [d]\ \}$$

Statistical Learning Theory

- A dataset of N training samples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i = h(\boldsymbol{x}_i))\}_{i=1}^N$
  training set: $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N) \overset{\text{i.i.d.}}{\sim} P_{\boldsymbol{x},y}$ ($P_{\boldsymbol{x},y}$ unknown)
  $h \in \mathcal{H}$: **hidden unknown fnc** from which training set is generated
  $f \in \mathcal{F}$: **learned fnc** from the training set
- Prediction on sample i: $\hat{y}_i := f(\boldsymbol{x}_i), f \in \mathcal{F}$
- Loss on sample i: $\ell(\underbrace{y_i}_{\text{true}}, \underbrace{f(\boldsymbol{x}_i)}_{\text{predicted}})$, using 0 / 1 loss: $\mathbb{I}(y_i \neq f(\boldsymbol{x}_i))$

Definitions:

- **Empirical Risk:** $L(f_N) := N^{-1} \sum_{i=1}^N \ell(y_i, f(\boldsymbol{x}_i))$ (training error)
  $f_N()$: fnc learned after N training samples
- **Expected Risk:** $L(f) := \mathbb{E}_{(\boldsymbol{x},y)} [\ell(y, f(\boldsymbol{x}))]$
- **Oracle Risk:** $L(f^\star) := \inf_{f \in \mathcal{F}} \ L(f)$
  Bayes Risk: $L^\star := \inf_f \ L(f)$

Statistical Learning Theory

- Ideally: difference b/w **expected (true) risk**, $L(f)$, and **empirical risk**, $L(f_N)$ to be 'small'
- Decompose the risk as:

$$L(f_N) - L^\star = \underbrace{(L(f_N) - L(f^\star))}_{\text{bias}} + \underbrace{(L(f^\star) - L^\star)}_{\text{variance}}$$

bias $=$ approximation error of estimator,
variance $=$ stochastic error of estimation

Probably Approximately Correct (PAC) learning [Vailant' 81]:

There exists (ϵ, δ) such that: $\mathbb{P}[|L(f_N) - L(f^\star)| \leq \epsilon] \geq 1 - \delta$

- $L(f_N)$ is ϵ-close to $L(f^\star)$ with probability at least $1 - \delta$
- probabilistic upperbound on bias
- PAC framework not often used in SL, but in multi-armed bandits

Statistical Learning Theory

Another approach is to bound the difference between **expected risk**, $L(f)$, with **empirical risk**, $L(f_N)$

Hoeffding bound: [Hoeffding 63]

Assume that $f \in \mathcal{F}$ is fixed. Then,

$$\mathbb{P}[|L(f) - L(f_N)| \geq \epsilon] \leq 2\exp^{-2n\epsilon^2}$$

- minimizing empirical risk and expected risk are equivalent a.s.
- Not applicable: f is not fixed in learning but optimized

In practice most bounds are loose.

- Most quantities cannot be computed in closed-form:
  data-generating distribution $P_{\boldsymbol{x},y}$ needed (not known in supervised learning)
- Theory developed by V. Vapnik more prevalent in SLT

Statistical Learning Theory

Let $\mathcal{N}^{\mathcal{F}}(z_1, ..., z_N) := |\{\ell(z_1), ..., \ell(z_N)\}|$, $z_i = (\boldsymbol{x}_i, y_i), i \in [N]$
$\{\ell(z_1), ..., \ell(z_N)\}$: N-dimensional binary vector
$\mathcal{N}^{\mathcal{F}}(z_1, ..., z_N)$: counts num of possible patterns (random)

Vapnik-Chervonenkis (VC) Entropy:
$H^{\mathcal{F}}(N) := \mathbb{E}[\log_2 \mathcal{N}^{\mathcal{F}}(\boldsymbol{z}_1, ..., \boldsymbol{z}_N)]$

Theorem 1: If the VC entropy converges uniformly ($\lim_{N\to\infty} \frac{H^{\mathcal{F}}(N)}{N} \to 0$)
$\Rightarrow$ **expected risk**, $L(f)$, and **empirical risk**, $L(f_N)$ are uniformly close.
"converges uniformly" $\Rightarrow$ VC entropy grows sub-linearly in N
need VC entropy bounded to ensure true and emp risk are close

- $H^{\mathcal{F}}(N)$ cannot be computed analytically ($P_{\boldsymbol{x},y}$ unknown)
- $S^{\mathcal{F}}(N)$ **shattering coefficient** for loss class $\mathcal{F}$:

$$S^{\mathcal{F}}(N) := \sup_{(\boldsymbol{z}_1,...,\boldsymbol{z}_N)} \mathcal{N}^{\mathcal{F}}(\boldsymbol{z}_1, ..., \boldsymbol{z}_N)$$

necessary cond for Theorem 1
use the upperbound: $H^{\mathcal{F}}(N) \leq \log_2(S^F(N))$
showing that $\lim_{N\to\infty} \frac{\log_2 S^{\mathcal{F}}(N)}{N} \to 0$ is necessary
illustrate with an example (2D classifier)

Statistical Learning Theory

Theorem 2: Shattering coeff satisfies one of the following:

a) $\log_2(S^{\mathcal{F}}(N)) = N$, for $\log_2(N) \geq 0$
shattering coeff exponentially increasing:
necessary cond for Theorem 1 does not hold: cannot say if empirical and expected risk are close

b) $\log_2(S^{\mathcal{F}}(N)) = N$, for $\log_2(N) \geq \log_2(D)$ and
$\log_2(S^{\mathcal{F}}(N)) \leq D \log_2 \frac{cN}{D}$, for $\log_2(N) < \log_2(D)$
$S^F(N)$ exponentially increasing until D, and polynomial for $N > D$
necessary cond for Theorem 1 holds: empirical and expected risk are uniformly close

VC Dimension $V(\mathcal{F})$: Largest integer D such that Theorem 2-b) holds.
$V(\mathcal{F}) = \infty$ for Theorem 2-a).

- Intuition: max number of features, D, that can be shattered by $\mathcal{F}$
  shattered = approximated with zero error

Applications

VC dimension: max number of features shattered (classified with zero error)
capacity of a model (e.g. channel capacity)

Example: VC dimension of linear classifier in $\mathbb{R}^2$,
$\mathcal{F} = \{\mathbb{I}[\ w_1x_1 + w_2x_2 \geq c\ ] \mid (w_1, w_2) \in \mathbb{R}_{-0}\}$

- $V(\mathcal{F}) = 3$. Proof ? see board

Example: VC dimension of linear classifier in $\mathbb{R}^d$,
$\mathcal{F} = \{\ \mathbb{I}[\ \sum_j^d\ w_jx_j \geq c\ ] \mid w_j \in \mathbb{R}_{-0}, j \in [d]\ \}$

- $V(\mathcal{F}) = d + 1$: $V(\mathcal{F})$ increasing with model size

Theorem 1 satisfied for above cases.

- empirical risk and expected risk uniformly close
- empirical risk min ‘equivalent’ to expected risk min

Model Complexity and Overfitting

Empirical risk minimization (from SLT perspective):

$$\arg\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i))$$

Consider two func classes: $\mathcal{F}_1$ and $\mathcal{F}_2$, where $\mathcal{F}_1 \subset \mathcal{F}_2$

$\Rightarrow$ **model complexity** of $f_1 \in \mathcal{F}_1$ lower than $f_2 \in \mathcal{F}_2$

Constrain ERM prob. to control model complexity

$$f_1 = \arg\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i)) \quad \text{s. t. } \; r(f) \le c_1$$

$$f_2 = \arg\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i)) \quad \text{s. t. } \; r(f) \le c_2,\; c_1 < c_2$$

$r(f)$ is **regularization**: increases as complexity of f increases

$\Rightarrow$ model complexity of f_1 lower that f_2

Regularized ERM problem equiv to

$$\arg\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(\boldsymbol{x}_i)) + r(f),$$

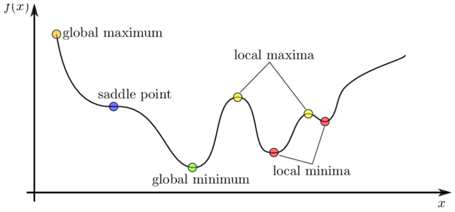
ex: $r(f) = \lambda\|f\|_2^2$. increase λ reduces model complexity

Outline

Definitions and Notations

- **Notation convention:**
  x is scalar, $\boldsymbol{x}$ is vector, $\boldsymbol{X}$ is matrix
  $\|\boldsymbol{x}\|_2$: Euclidean norm , $\|\boldsymbol{X}\|_F$: Frobenius norm
  $\nabla f(\boldsymbol{x}) \in \mathbb{R}^d$: gradient of $f(\boldsymbol{x})$
  $\nabla^2 f(\boldsymbol{x}) \in \mathbb{R}^{d\times d}$: Hessian of $f(\boldsymbol{x})$ (symmetric matrix)
- **Inequalities:** $\boldsymbol{x} \leq \boldsymbol{y}$ and $\boldsymbol{x} \geq \boldsymbol{y}$ holds element-by-element
- **Positive Definite (PD) Matrix**
  $\boldsymbol{X} \in \mathbb{R}^{N\times N}$ is PD matrix iff $\lambda_i[\boldsymbol{X}] > 0,\ \forall\ i \in [N]$: $\boldsymbol{X} \succ 0$
- **Positive Semi-definite (PSD) Matrix**
  $\boldsymbol{X} \in \mathbb{R}^{N\times N}$ is PSD matrix iff $\lambda_i[\boldsymbol{X}] \geq 0,\ \forall\ i \in [N]$: $\boldsymbol{X} \succeq 0$
- **Inequalities on semi-definite cone:**
  Set of PSD (PD) matrices is a cone. Any two PSD (PD) matrices can be ordered using the '$\succeq$'
  $\boldsymbol{X} \succeq \boldsymbol{Y} \Leftrightarrow \boldsymbol{X} - \boldsymbol{Y} \succeq 0$, for $\boldsymbol{X} \succeq 0$, $\boldsymbol{Y} \succeq 0$.

Optimization Definitions



$\boldsymbol{w}^\star$ global minimum iff
$\nabla f(\boldsymbol{w}^\star) = 0,\ \nabla^2 f(\boldsymbol{w}^\star) \succeq 0, f(\boldsymbol{w}^\star) \leq f(\boldsymbol{w}),\ \forall\ \boldsymbol{w} \in\ \mathcal{W}$

$\boldsymbol{w}^\star$ local minimum iff
$\nabla f(\boldsymbol{w}^\star) = 0,\ \nabla^2 f(\boldsymbol{w}^\star) \succeq 0, f(\boldsymbol{w}^\star) \leq f(\boldsymbol{w}),\ \forall\ \boldsymbol{w} \in\ \mathcal{W}$

$\boldsymbol{w}^\star$ saddle point iff $\nabla f(\boldsymbol{w}^\star) = 0,\ \nabla^2 f(\boldsymbol{w}^\star)$ indefinite

$\boldsymbol{w}^\star$ stationary point iff $\nabla f(\boldsymbol{w}^\star) = 0,\ \boldsymbol{w} \in \mathcal{W}$

Optimization Nomenclature

Convex optimization (Lec 2)

f and $\mathcal{W}$ are convex, then: $\min_{\boldsymbol{w} \in \mathcal{W}} f(\boldsymbol{w})$

Stationary point ($\nabla f(\boldsymbol{w}) = 0$) $\Leftrightarrow$ global optimum

Gradient descent: $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)$

Stochastic gradient descent (SGD): $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla_i f(\boldsymbol{w}_k)$

Non-convex optimization (Lec 4)

Stationary point ($\nabla f(\boldsymbol{w}) = 0$) $\Rightarrow$ Local optima, saddle points, global optimum

Some references

- S. Bubeck, "Convex optimization: Algorithms and complexity," Foundations and Trends in Machine Learning, 2015.
- L. Bottou, F. Curtis, and J. Norcedal, "Optimization methods for large-scale machine learning," SIAM Rev., 2018.
- T. Weissman, "EE 378A: Statistical signal processing", Lecture notes, 2016.
- S. Boyd, et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations and Trends in Machine Learning, 2011.
- M.I. Jordan, J.D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," Journal of the American Statistical Association, 2018.
- M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," Mathematical Programming, 2017.
- Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT press 2016.
- S. Sra, S. Nowozin, and S.J. Wright (eds), "Optimization for machine learning" Mit Press, 2012.

Deep Learning

Lecture 1: Fundamentals of Learning

University of Agder,
Kristiansand, Norway

Prepared by: Hadi Ghauch*, Hossein S. Ghadikolaei†

* Telecom ParisTech

† Royal Institute of Technology, KTH

https://sites.google.com/view/fundl/home

April 2019