



# EP3260: Machine Learning Over Networks

## Lecture 1: Introduction

Hossein S. Ghadikolaei

Division of Network and Systems Engineering  
School of Electrical Engineering and Computer Science  
KTH Royal Institute of Technology, Stockholm, Sweden

<https://sites.google.com/view/mlons2020/home>

February 2020

# Outline

1. Logistics
2. Course Contents
3. Lectures

# Outline

1. Logistics

2. Course Contents

3. Lectures

# Logistics

- 10 credits advanced Ph.D. course
- 16 lectures:
  - Fundamentals (Lectures 1-10), Special Topics (Lectures 11-16)
- Student groups for homework (HW) and computer assignments (CAs)
  - 2-3 students per group
  - Deadline for groups formation: end of Lecture 2**
- 3 HW and 6 CAs (for groups)
  - HW due in one week, CA due in two weeks
  - peer-to-peer review of HW and CAs
- Optional assignments and final research project

## Logistics cont.

- Last round of the course:  
<https://sites.google.com/view/mlons2019/home>
- 55 participants (35 outside Sweden)
- Email: hshokri@kth.se, jmbdsj@kth.se, carlofi@kth.se  
(please **use “MLoN-2020:” in the email subject**)
- Course website:  
<https://sites.google.com/view/mlons2020/home>
- YouTube channel: [https://www.youtube.com/channel/UCoFj1tFuK4b\\_Wh21-KQoU5g?view\\_as=subscriber](https://www.youtube.com/channel/UCoFj1tFuK4b_Wh21-KQoU5g?view_as=subscriber)
- GitHub account for HW and CA submissions:  
<https://github.com/hshokrig/EP3260-MLoNs-2020>

# Outline

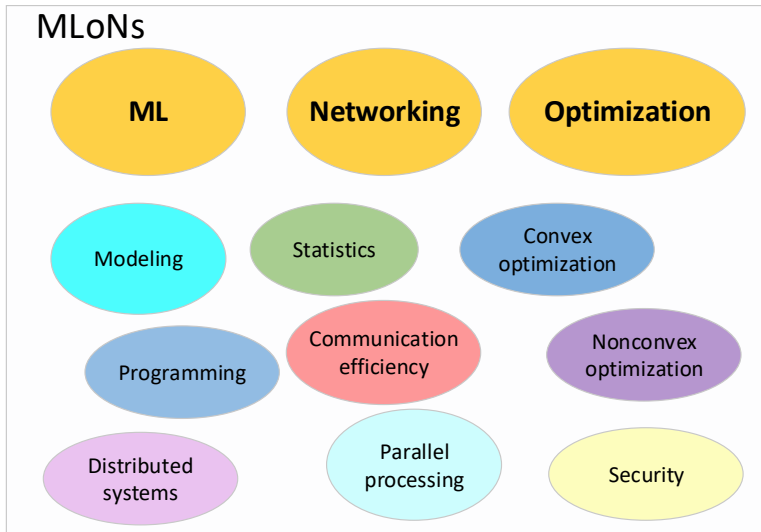
1. Logistics

2. Course Contents

3. Lectures

# Course contents

## MLOs



# Machine learning!

- Unsupervised learning (e.g.,  $k$ -means)

learning from unlabeled data: identifies commonalities

- Supervised learning (e.g., deep neural networks)

learning from labeled data: regression and classification

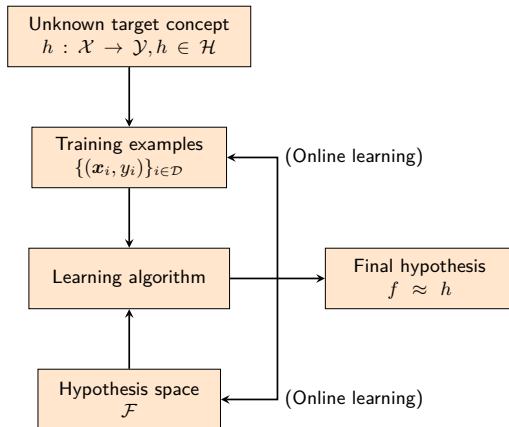
- Reinforcement learning (e.g.,  $Q$ -learning)

learning by interacting with an unknown environment (modeled by a Markov decision process)

sequential decision making, lack of correct dataset a priori, suboptimal actions are allowed in the learning process



# Supervised learning



- $\mathcal{F}$  instead of  $\mathcal{H}$ , e.g., an easier class of mappings like linear regression or neural networks

# Supervised learning

- A dataset of  $N$  training samples  $\mathcal{D} = \{(\mathbf{x}_i, y_i = h(\mathbf{x}_i))\}_{i=1}^N$
- Our prediction:  $\hat{y} = f(\mathbf{x}), f \in \mathcal{F}$
- Loss on a single observation:  $\ell(\mathbf{x}, h(\mathbf{x}), f(\mathbf{x}))$
- **Expected risk (test error):**  $L = \mathbb{E}_{(\mathbf{x}, y)} [\ell(\mathbf{x}, h(\mathbf{x}), f(\mathbf{x}))]$
- **Empirical risk (training error):**  $\hat{L} = \frac{1}{N} \sum_{i \in [N]} \ell(\mathbf{x}_i, h(\mathbf{x}_i), f(\mathbf{x}_i))$
- Assume  $\mathbf{w}$  parameterizes both  $h$  and  $f$ , and  $\mathbf{w}^*$  is the solution of our algorithm.

$$f(\mathbf{w}^*) - \min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w}) = \underbrace{\left( f(\mathbf{w}^*) - \min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) \right)}_{\text{estimation error}} + \underbrace{\left( \min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) - \min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w}) \right)}_{\text{approximation error}}$$

## Some examples

**Linear ridge regression:**

$$f(\mathbf{x}; \mathbf{w}) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \left( y_i - \mathbf{w}^T \mathbf{x}_i \right)^2 + \lambda \|\mathbf{w}\|_2^2$$

data fitting      +      regularizer

**Linear LASSO regression:**

$$f(\mathbf{x}; \mathbf{w}) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \left( y_i - \mathbf{w}^T \mathbf{x}_i \right)^2 + \lambda \|\mathbf{w}\|_1$$

**Support vector machine (binary classification):**

$$f(\mathbf{x}; \mathbf{w}) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \max \left( 0, 1 - y_i \left( \mathbf{w}^T \mathbf{x}_i - b \right) \right) + \lambda \|\mathbf{w}\|_2^2$$

# Optimization

- Convexity

**convex set:**  $\mathcal{X} \subseteq \mathbb{R}^d$  is convex if

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}, \theta \in [0, 1], \theta \mathbf{x}_1 + (1 - \theta) \mathbf{x}_2 \in \mathcal{X}$$

**convex function:**  $f : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\}$  for convex  $\mathcal{X}$  is convex if

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \lambda \in [0, 1], f(\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y})$$

**convex function:** its epigraph  $\{(t, \mathbf{x}) : f(\mathbf{x}) \leq t\}$  is a convex set

**strictly convex function:** convex  $f$  for which  $<$  holds

**Useful forms of Jensen's inequality:**  $f$  is convex,  $\{x_i\}_i$  are deterministic real numbers,  $a_i > 0$ ,  $X$  is random variable (**proof?**):

$$f\left(\frac{\sum a_i x_i}{\sum a_i}\right) \leq \frac{\sum a_i f(x_i)}{\sum a_i}, \quad f(E[X]) \leq E[f(X)]$$

# Optimization

- **Convex optimization**

$f$  and  $\mathcal{W}$  are convex, then:  $\underset{w \in \mathcal{W}}{\text{minimize}} f(w)$

local optimum  $\Rightarrow$  global optimum

**Linear convergence** with strongly convex and smooth  $f$

- **Efficient solvers.** Let  $f(w) := \frac{1}{N} \sum_{i=1}^N f(x_i; w)$ .

Gradient descent:  $w_{k+1} = w_k - \alpha_k \nabla_w f(w_k)$

Stochastic gradient descent (SGD):  $w_{k+1} = w_k - \alpha_k \nabla_w f(x_\zeta; w_k)$

SGD with memory, e.g., stochastic average gradient

Acceleration:  $v_{k+1} = \gamma v_k - \alpha_k \nabla_w f(w_k), w_k = w_{k-1} - v_k$

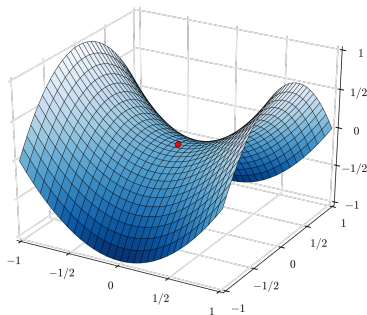
# Optimization

- **on-convex optimization**

local optimum  $\nrightarrow$  global optimum

saddle points:  $f(x, y) = y^2 - x^2$

perturbed gradient descent

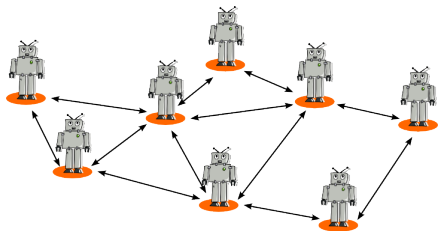


# Networked systems

- Graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$

$\mathcal{V}$ : set of vertices

$\mathcal{E}$ : set of edges

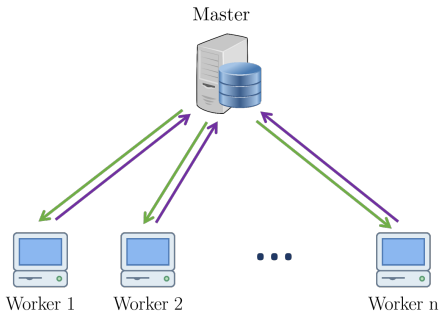


Example	$v_i \in \mathcal{V}$	$e_{ij} \in \mathcal{E}$
Computer networks	worker $i$	communication link $v_i \rightarrow v_j$
Wireless networks	link $i$	interference from $v_i$ to $v_j$
Biological networks	sensor $i$	communication link $v_i \rightarrow v_j$

# Example 1: Large-scale ML

$$\underset{\mathbf{w} \in \mathbb{R}^d}{\text{minimize}} \quad \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i; \mathbf{w})$$

- Large  $N$   
parallel processing?  
random sampling?
- Large  $d$ :  
sparse solutions?  
quantization?



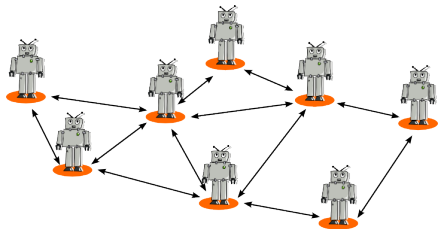


## Example 2: Multiagent systems

$$\underset{\mathbf{w} \in \mathbb{R}^d}{\text{minimize}} \quad \frac{1}{N} \sum_{i=1}^N f_i(\mathbf{w})$$

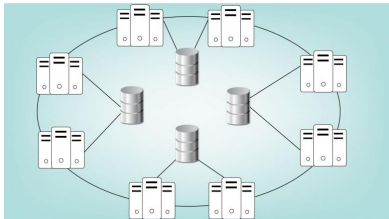
$d$  combined decision variables

- Local variables:  $\mathbf{w}_1 \neq \mathbf{w}_2$
- Private information:  
 $f_i(\mathbf{w}) = \frac{1}{N_i} \sum_{j=1}^{N_i} h(\mathbf{w}; \mathbf{x}_{ij})$
- Consensus form (separable 😊)

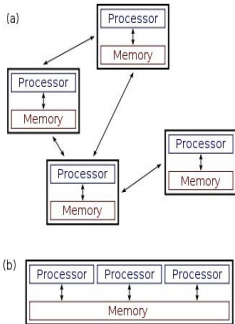


$$\underset{\{z_i\}}{\text{minimize}} \quad \sum_{i=1}^N f_i(z_i)$$
$$\text{s.t. } z_i = z_j \in \mathbb{R}^d$$

## Example 3: Distributed systems



- Local information
- Privacy constraints
- Security challenges



## Example 4: Intra-body sensor networks

- Abstractly, same as before
- Low processing power
- Harsh communication environment
- Higher system dynamics
- Time-sensitive decisions

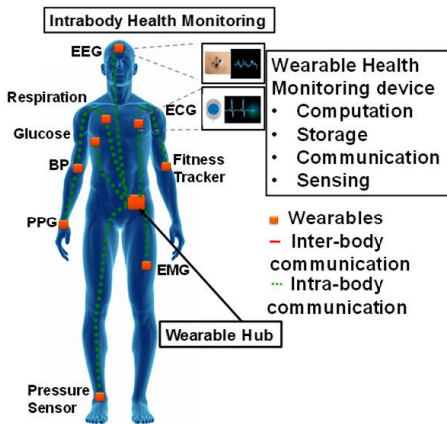


Image source: "Wearable Health Monitoring Using Capacitive Voltage-Mode Human Body Communication," arXiv'17.

# Outline

1. Logistics

2. Course Contents

3. Lectures

# Lectures

- Lecture 1: Introduction, – Today!
- Lecture 2: Centralized Convex ML (deterministic algorithms), Feb. 12, 2020, 10:00-12:00.
- Lecture 3: Centralized Convex ML (stochastic algorithms), Feb. 19, 2020, 10:00-12:00.
- Lecture 4: Computer Assignment Session and Homework (part 1), Feb. 26, 2020, 10:00-12:00.
- Lecture 5: Centralized Nonconvex ML, Mar. 4, 2020, 10:00-12:00.
- Lecture 6: Distributed ML, Mar. 6, 2020, 10:00-12:00.
- Lecture 7: ADMM, Mar. 12, 2020, 10:00-12:00.
- Lecture 8: Communication Efficiency, Mar. 18, 2020, 10:00-12:00.
- Lecture 9: Computer Assignment Session and Homework (part 2), Mar. 25, 2020, 10:00-12:00.
- Lecture 10: Deep Neural Networks, Apr. 1, 2020, 10:00-12:00.
- Lecture 11: Special Topic 1: Large-scale MLoN
- Lecture 12: Special Topic 2: Application areas: Federated learning and privacy-preserving distributed MLoN
- Lecture 13: Special Topic 3: Security in MLoN
- Lecture 14: Special Topic 4: Online MLoNs
- Lecture 15: Special Topic 5: Robust MLoN
- Lecture 16: Application areas and open research problems

## Special topics: two-days workshop

- Poster workshop for Lectures 11–16
- Date: April 23 and 24, 2020, 10:00–18:00
- Some invited talks, one 30-min oral presentation per group, integrated into poster sessions
- Panel discussion on recent progresses and the future of machine learning
- Networking!

## 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.
- 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.



# EP3260: Machine Learning Over Networks

## Lecture 1: Introduction

Hossein S. Ghadikolaei

Division of Network and Systems Engineering  
School of Electrical Engineering and Computer Science  
KTH Royal Institute of Technology, Stockholm, Sweden

<https://sites.google.com/view/mlons2020/home>

February 2020