Optimization for Large Scale Machine Learning

Francis Bach

INRIA - Ecole Normale Supérieure, Paris, France

Hausdorff School - September 2020

Slides available at `www.di.ens.fr/~fbach/hausdorff2020.pdf`

Scientific context

- **Proliferation of digital data**
  - Personal data
  - Industry
  - Scientific: from bioinformatics to humanities
- **Need for automated processing of massive data**

Scientific context

- **Proliferation of digital data**
  - Personal data
  - Industry
  - Scientific: from bioinformatics to humanities
- **Need for automated processing of massive data**
- **Series of "hypes"**

  Big data $\rightarrow$ Data science $\rightarrow$ Machine Learning
  $\rightarrow$ Deep Learning $\rightarrow$ Artificial Intelligence

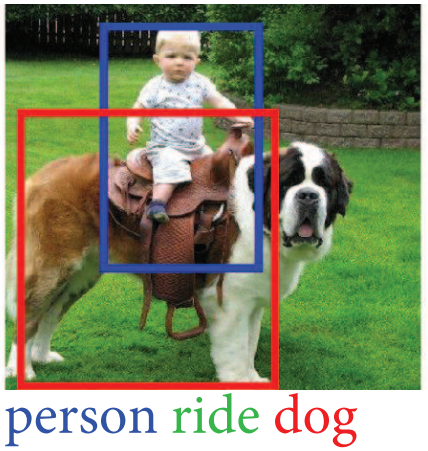
Scientific context

- **Proliferation of digital data**
  - Personal data
  - Industry
  - Scientific: from bioinformatics to humanities
- **Need for automated processing of massive data**
- **Series of "hypes"**

  Big data $\rightarrow$ Data science $\rightarrow$ Machine Learning
  $\rightarrow$ Deep Learning $\rightarrow$ Artificial Intelligence
- **Healthy interactions between theory, applications, and hype?**

Recent progress in perception (vision, audio, text)

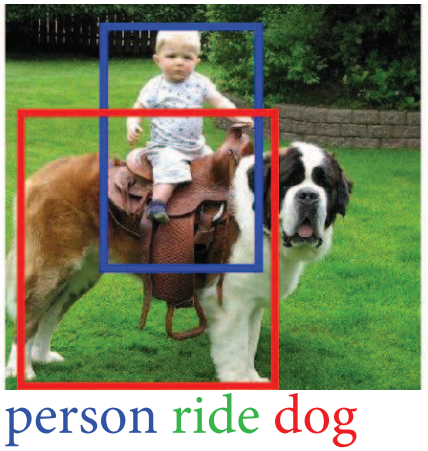
From translate.google.fr



From Peyré et al. (2017)

Recent progress in perception (vision, audio, text)

From translate.google.fr



From Peyré et al. (2017)

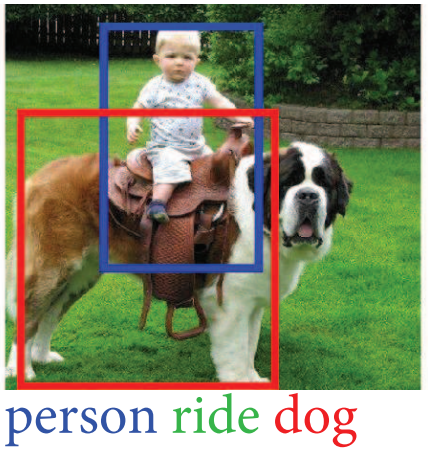
(1) **Massive data**

(2) **Computing power**

(3) **Methodological and scientific progress**

Recent progress in perception (vision, audio, text)

From translate.google.fr



From Peyré et al. (2017)

(1) **Massive data**

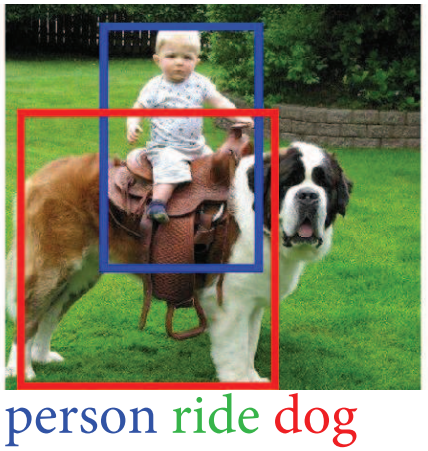
(2) **Computing power**

(3) **Methodological and scientific progress**

"Intelligence" = models + algorithms + data
+ computing power

Recent progress in perception (vision, audio, text)

From translate.google.fr



From Peyré et al. (2017)

(1) **Massive data**

(2) **Computing power**

(3) **Methodological and scientific progress**

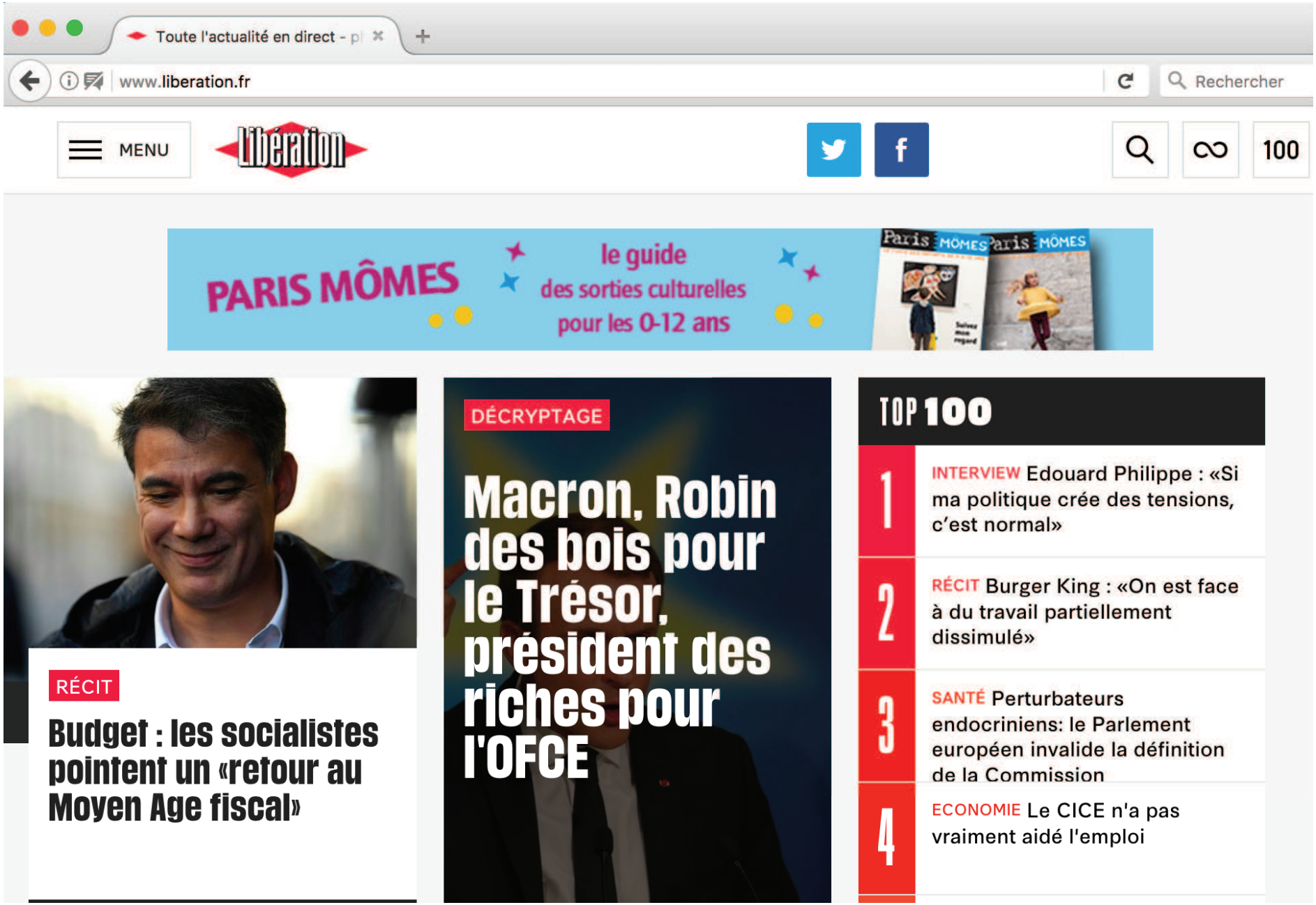
"Intelligence" = models + algorithms + data
+ computing power

Machine learning for large-scale data

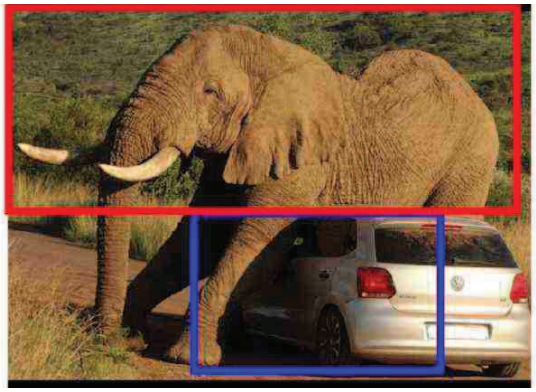
- **Large-scale supervised machine learning**: **large d, large n**
  - d : dimension of each observation (input) or number of parameters
  - n : number of observations

- **Examples**: computer vision, advertising, bioinformatics, etc.

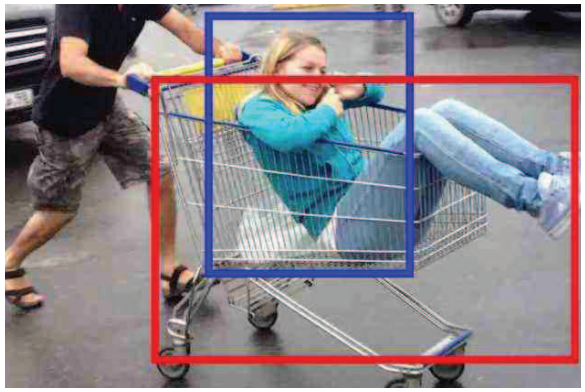
Advertising



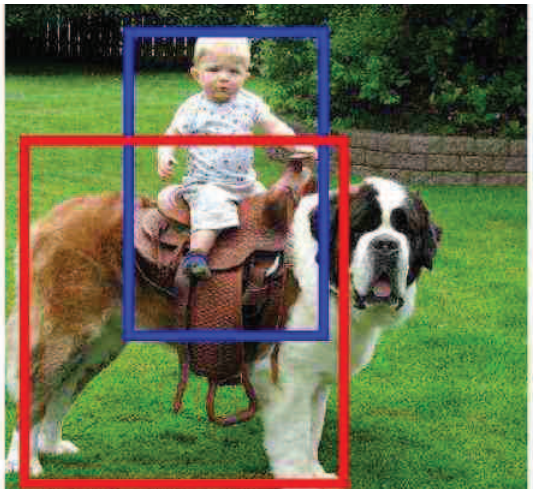
Object / action recognition in images



car under elephant



person in cart

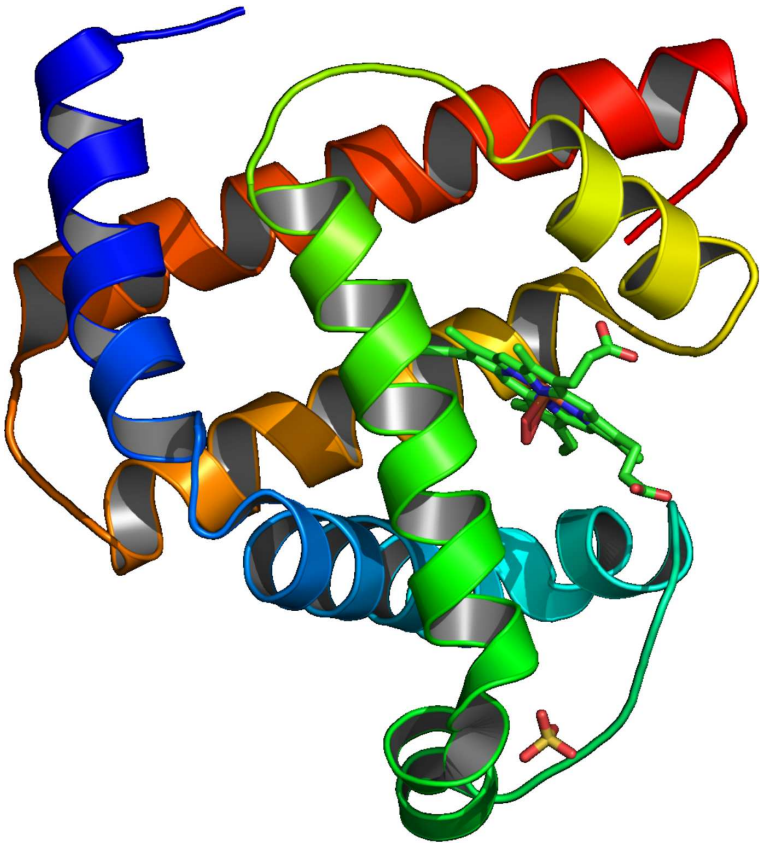


person ride dog

person on top of traffic light

From Peyré, Laptev, Schmid and Sivic (2017)

Bioinformatics



- Predicting multiple functions and interactions of **proteins**
- **Massive data**: up to 1 millions for humans!
- **Complex data**
  - Amino-acid sequence
  - Link with DNA
  - Tri-dimensional molecule

Machine learning for large-scale data

- **Large-scale supervised machine learning**: **large d, large n**
  - d : dimension of each observation (input), or number of parameters
  - n : number of observations

- **Examples**: computer vision, advertising, bioinformatics, etc.

- **Ideal running-time complexity**: $O(dn)$

Machine learning for large-scale data

- **Large-scale supervised machine learning**: **large d, large n**
  - d : dimension of each observation (input), or number of parameters
  - n : number of observations
- **Examples**: computer vision, advertising, bioinformatics, etc.
- **Ideal running-time complexity**: $O(dn)$
- **Going back to simple methods**
  - Stochastic gradient methods (Robbins and Monro, 1951)
- **Goal: Present classical algorithms and some recent progress**

Outline

1. **Introduction/motivation: Supervised machine learning**
   - Machine learning $\approx$ optimization of finite sums
   - Batch optimization methods

2. **Fast stochastic gradient methods for convex problems**
   - Variance reduction: for *training* error
   - Constant step-sizes: for *testing* error

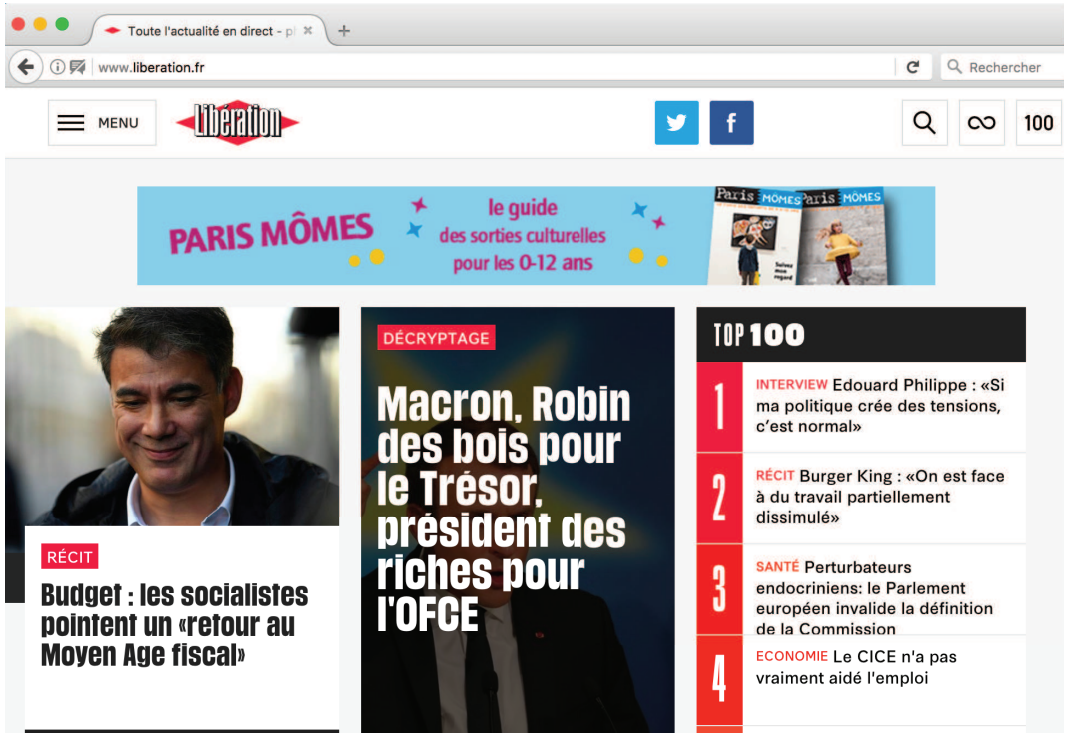
3. **Beyond convex problems**
   - Generic algorithms with generic "guarantees"
   - Global convergence for over-parameterized neural networks

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

Parametric supervised machine learning

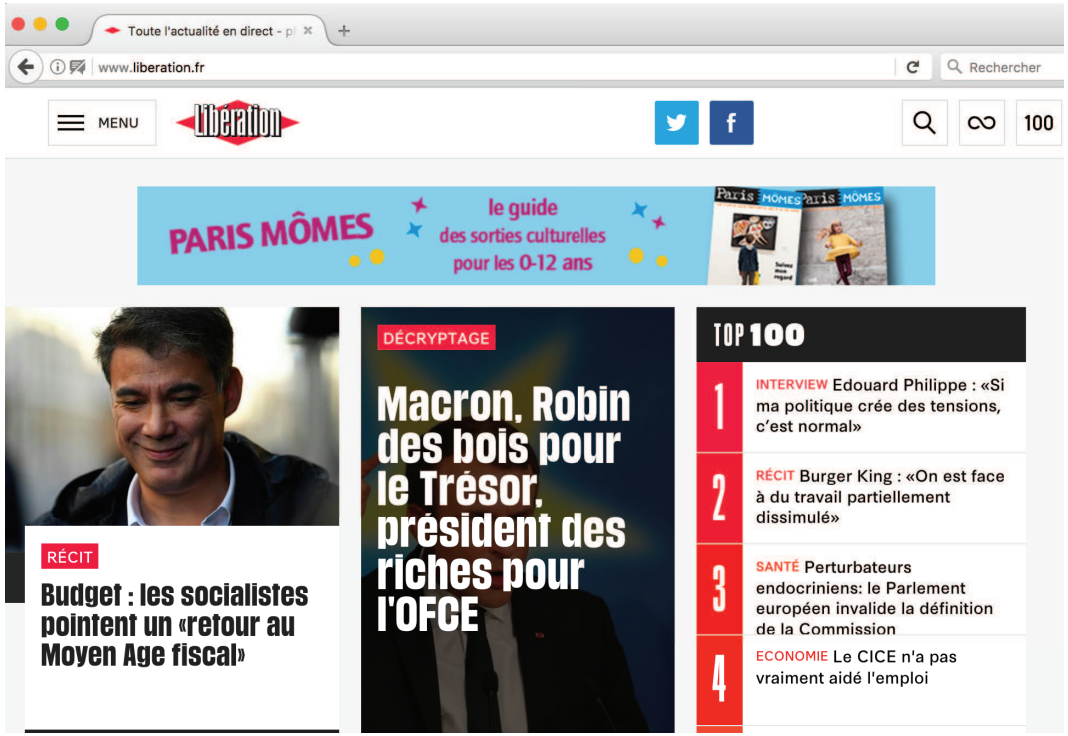
- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$



- **Advertising**: $n > 10^9$
  - $\Phi(x) \in \{0, 1\}^d$, $d > 10^9$
  - Navigation history + ad

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$



- **Advertising**: $n > 10^9$
  - $\Phi(x) \in \{0, 1\}^d$, $d > 10^9$
  - Navigation history + ad
- Linear predictions
  - $h(x, \theta) = \theta^\top \Phi(x)$

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

x_1 x_2 x_3 x_4 x_5 x_6

$y_1 = 1$ $y_2 = 1$ $y_3 = 1$ $y_4 = -1$ $y_5 = -1$ $y_6 = -1$

Parametric supervised machine learning

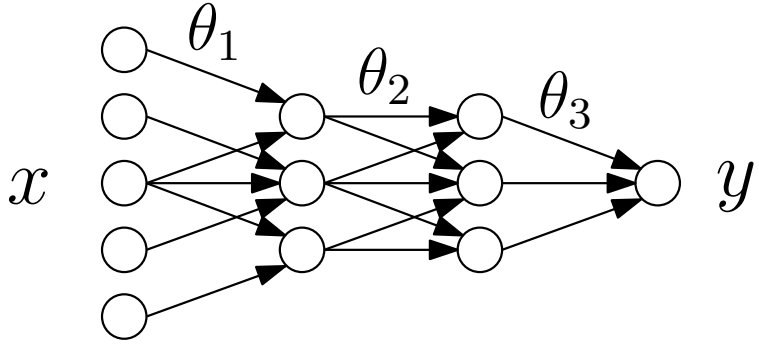
- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$

- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

x_1 x_2 x_3 x_4 x_5 x_6

$y_1 = 1$ $y_2 = 1$ $y_3 = 1$ $y_4 = -1$ $y_5 = -1$ $y_6 = -1$

  - Neural networks $(n, d > 10^6)$: $h(x, \theta) = \theta_m^\top \sigma(\theta_{m-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x))$



Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta)$$

data fitting term + regularizer

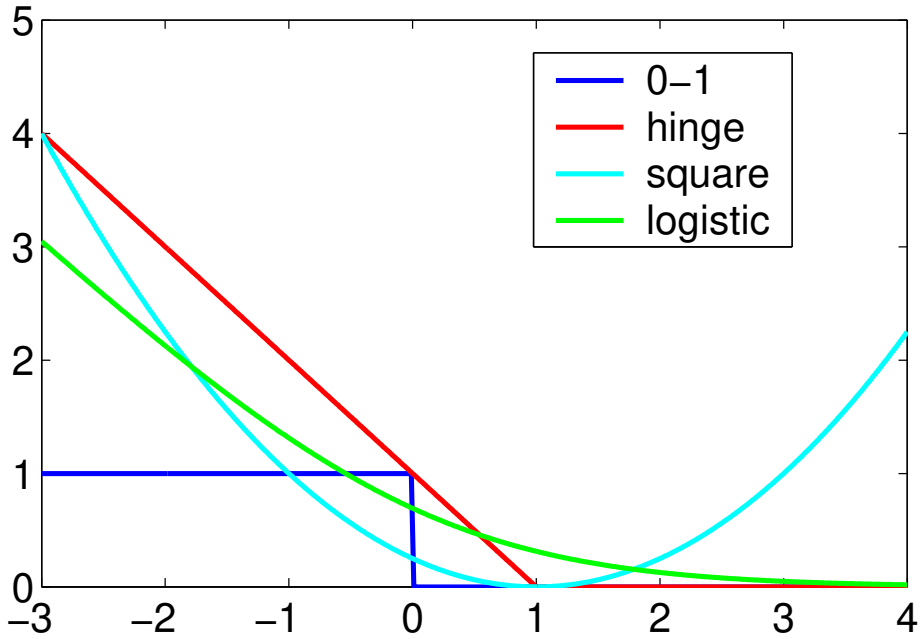
Usual losses

- **Regression**: $y \in \mathbb{R}$, prediction $\hat{y} = h(x, \theta)$
  - quadratic loss $\frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(y - h(x, \theta))^2$

Usual losses

- **Regression**: $y \in \mathbb{R}$, prediction $\hat{y} = h(x, \theta)$
  - quadratic loss $\frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(y - h(x, \theta))^2$

- **Classification** : $y \in \{-1, 1\}$, prediction $\hat{y} = \text{sign}(h(x, \theta))$
  - loss of the form $\ell(y\, h(x, \theta))$
  - "True" 0-1 loss: $\ell(y\, h(x, \theta)) = 1_{y\, h(x,\theta)<0}$
  - Usual convex losses:



Main motivating examples

- **Support vector machine** (hinge loss): non-smooth

$$\ell(Y, h(X\theta)) = \max\{1 - Yh(X, \theta), 0\}$$

- **Logistic regression**: smooth

$$\ell(Y, h(X\theta)) = \log(1 + \exp(-Yh(X, \theta)))$$

- **Least-squares regression**

$$\ell(Y, h(X\theta)) = \frac{1}{2}(Y - h(X, \theta))^2$$

- **Structured output regression**
  - See Tsochantaridis et al. (2005); Lacoste-Julien et al. (2013)

Usual regularizers

- **Main goal**: avoid overfitting
- **(squared) Euclidean norm**: $\|\theta\|_2^2 = \sum_{j=1}^{d} |\theta_j|^2$
  - Numerically well-behaved if $h(x, \theta) = \theta^\top \Phi(x)$
  - Representer theorem and kernel methods : $\theta = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$
  - See, e.g., Schölkopf and Smola (2001); Shawe-Taylor and Cristianini (2004)

Usual regularizers

- **Main goal**: avoid overfitting
- **(squared) Euclidean norm**: $\|\theta\|_2^2 = \sum_{j=1}^d |\theta_j|^2$
  - Numerically well-behaved if $h(x, \theta) = \theta^\top \Phi(x)$
  - Representer theorem and kernel methods : $\theta = \sum_{i=1}^n \alpha_i \Phi(x_i)$
  - See, e.g., Schölkopf and Smola (2001); Shawe-Taylor and Cristianini (2004)
- **Sparsity-inducing norms**
  - Main example: ℓ_1-norm $\|\theta\|_1 = \sum_{j=1}^d |\theta_j|$
  - Perform model selection as well as regularization
  - Non-smooth optimization and structured sparsity
  - See, e.g., Bach, Jenatton, Mairal, and Obozinski (2012a,b)

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta)$$

data fitting term + regularizer

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \Big\{ \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta) \Big\} \qquad = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

data fitting term + regularizer

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \Big\{ \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta) \Big\} \qquad = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

data fitting term + regularizer

- Optimization: optimization of regularized risk training cost

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, **i.i.d.**
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \Big\{ \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta) \Big\} \qquad = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

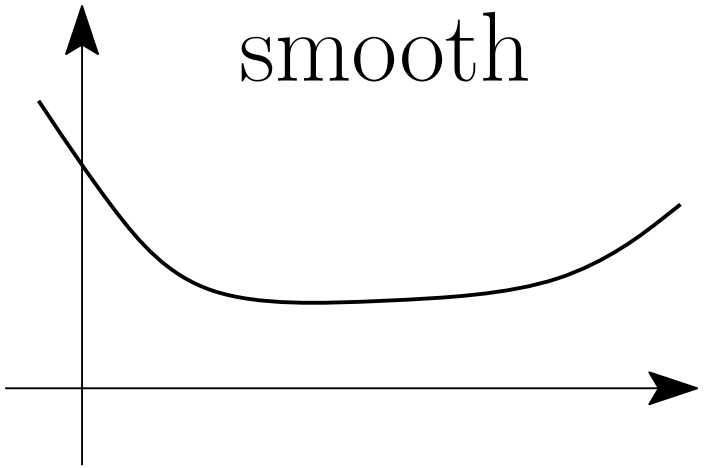
data fitting term + regularizer

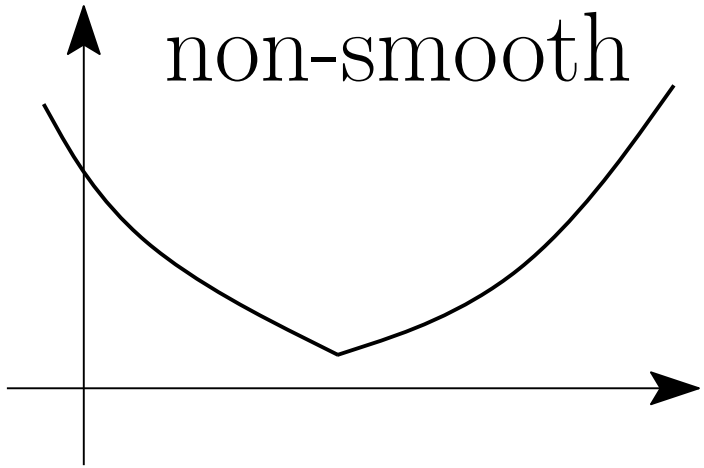
- Optimization: optimization of regularized risk — training cost
- Statistics: guarantees on $\mathbb{E}_{p(x,y)} \ell(y, h(x, \theta))$ — testing cost

Smoothness and (strong) convexity

- A function $g : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if and only if it is twice differentiable and

$$\forall \theta \in \mathbb{R}^d, \ \left|\text{eigenvalues}\left[g''(\theta)\right]\right| \leqslant L$$





Smoothness and (strong) convexity

- A function $g : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if and only if it is twice differentiable and

$$\forall \theta \in \mathbb{R}^d, \ \big|\text{eigenvalues}\big[g''(\theta)\big]\big| \leqslant L$$

- **Machine learning**
  - with $g(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Smooth prediction function $\theta \mapsto h(x_i, \theta)$ + smooth loss
  - *(see board)*

Board

- Function $g(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i))$

Board

- Function $g(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i))$
- Gradient $g'(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell'(y_i, \theta^\top \Phi(x_i)) \Phi(x_i)$

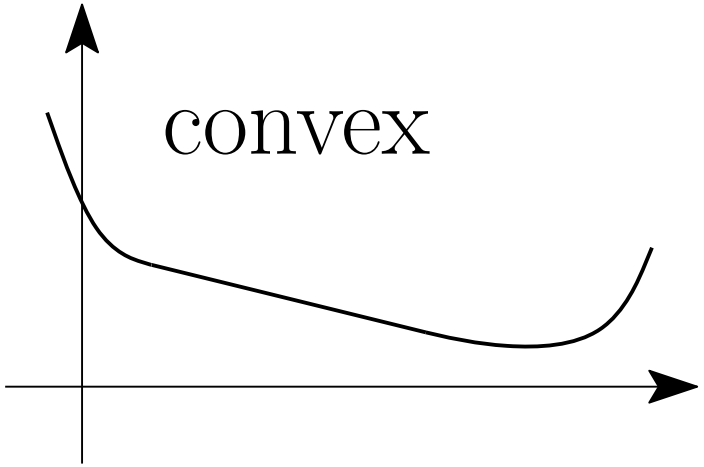
Board

- Function $g(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i))$
- Gradient $g'(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell'(y_i, \theta^\top \Phi(x_i)) \Phi(x_i)$
- Hessian $g''(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell''(y_i, \theta^\top \Phi(x_i)) \Phi(x_i) \Phi(x_i)^\top$
  - Smooth loss $\Rightarrow \ell''(y_i, \theta^\top \Phi(x_i))$ bounded

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if

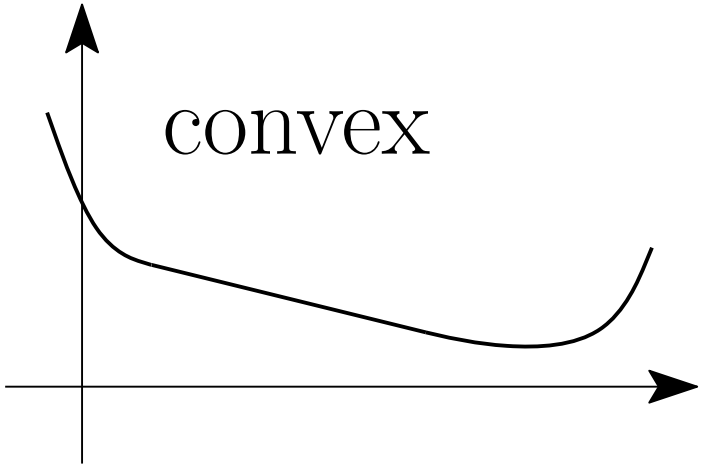
$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant 0$$

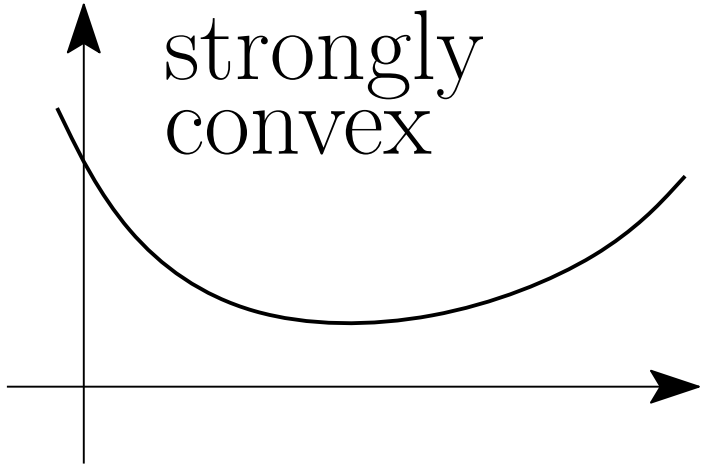


Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$



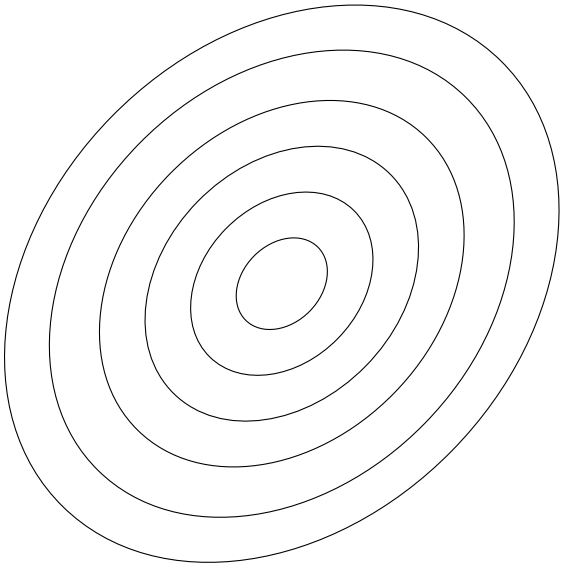


Smoothness and (strong) convexity

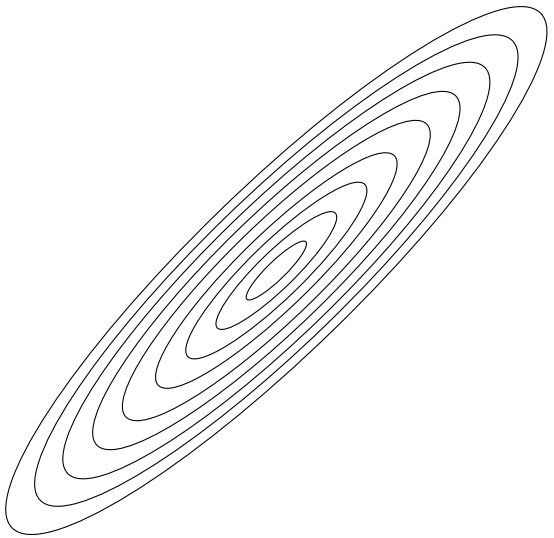
- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

  - Condition number $\kappa = L/\mu \geqslant 1$



(small $\kappa = L/\mu$)



(large $\kappa = L/\mu$)

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

- **Convexity in machine learning**
  - With $g(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

- **Convexity in machine learning**
  - With $g(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$

- **Relevance of convex optimization**
  - Easier design and analysis of algorithms
  - Global minimum vs. local minimum vs. stationary points
  - Gradient-based algorithms only need convexity for their analysis

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

- **Strong convexity in machine learning**
  - With $g(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Strongly convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

- **Strong convexity in machine learning**
  - With $g(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Strongly convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$
  - Invertible covariance matrix $\frac{1}{n} \sum_{i=1}^n \Phi(x_i) \Phi(x_i)^\top \Rightarrow n \geqslant d$ *(board)*
  - Even when $\mu > 0$, μ may be arbitrarily small!

Board

- Function $g(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i))$
- Gradient $g'(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell'(y_i, \theta^\top \Phi(x_i)) \Phi(x_i)$
- Hessian $g''(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell''(y_i, \theta^\top \Phi(x_i)) \Phi(x_i) \Phi(x_i)^\top$
  - Smooth loss $\Rightarrow \ell''(y_i, \theta^\top \Phi(x_i))$ bounded
- Square loss $\Rightarrow \ell''(y_i, \theta^\top \Phi(x_i)) = 1$
  - Hessian proportional to $\frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \Phi(x_i)^\top$

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

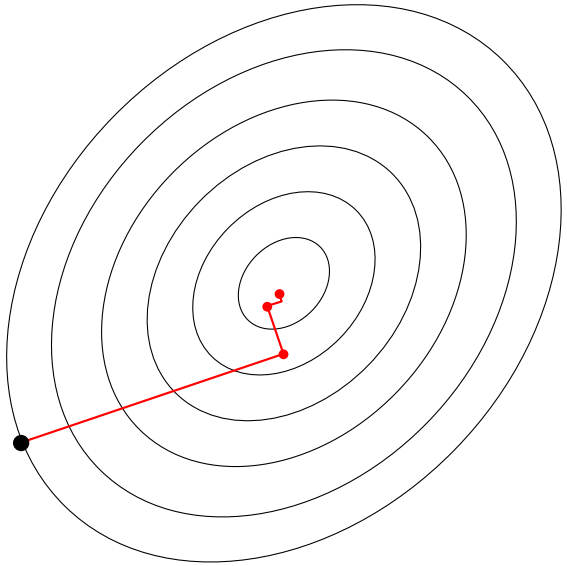
$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

- **Strong convexity in machine learning**
  - With $g(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Strongly convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$
  - Invertible covariance matrix $\frac{1}{n}\sum_{i=1}^n \Phi(x_i)\Phi(x_i)^\top \Rightarrow n \geqslant d$ *(board)*
  - Even when $\mu > 0$, μ may be arbitrarily small!

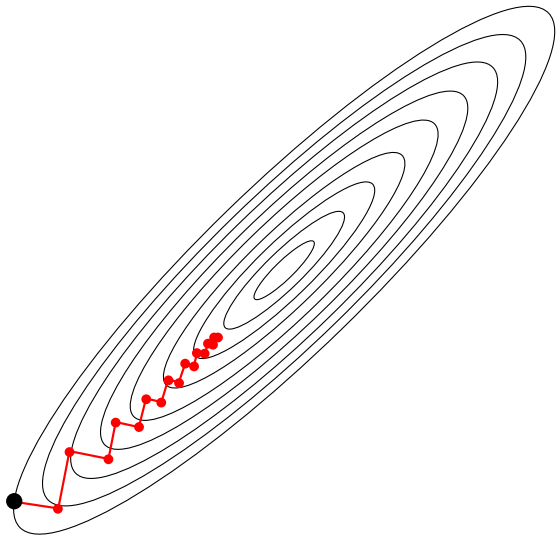
- **Adding regularization by $\frac{\mu}{2}\|\theta\|^2$**
  - creates additional bias unless μ is small, but reduces variance
  - Typically $L/\sqrt{n} \geqslant \mu \geqslant L/n$

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$ *(line search+adaptivity)*



(small $\kappa = L/\mu$)

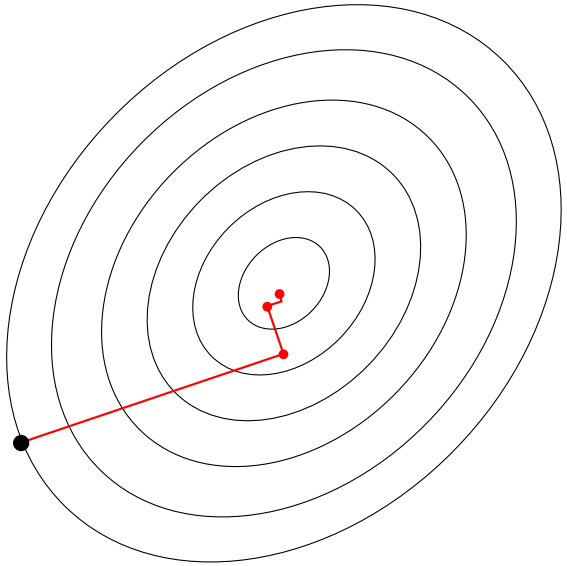


(large $\kappa = L/\mu$)

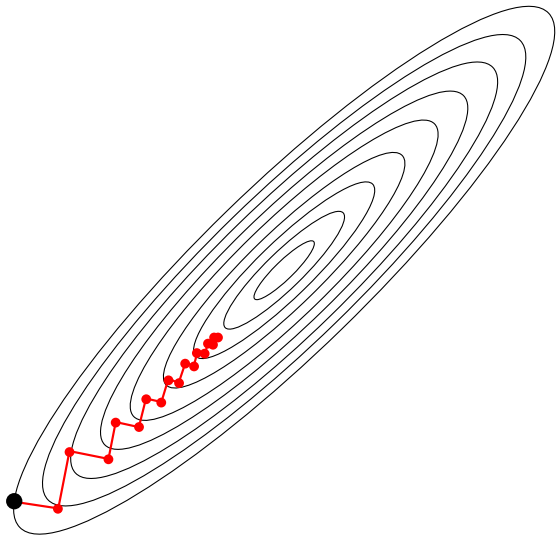
Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$ *(line search+adaptivity)*

$$g(\theta_t) - g(\theta_*) \leqslant O(1/t)$$
$$g(\theta_t) - g(\theta_*) \leqslant O((1-\mu/L)^t) = O(e^{-t(\mu/L)}) \text{ if } \mu\text{-strongly convex}$$



(small $\kappa = L/\mu$)



(large $\kappa = L/\mu$)

Gradient descent - Proof for quadratic functions

- Quadratic convex function: $g(\theta) = \frac{1}{2}\theta^\top H\theta - c^\top\theta$
  - μ and L are the smallest and largest eigenvalues of H
  - Global optimum $\theta_* = H^{-1}c$ (or $H^\dagger c$) such that $H\theta_* = c$

Gradient descent - Proof for quadratic functions

- Quadratic convex function: $g(\theta) = \frac{1}{2}\theta^\top H\theta - c^\top\theta$
  - μ and L are the smallest and largest eigenvalues of H
  - Global optimum $\theta_* = H^{-1}c$ (or $H^\dagger c$) such that $H\theta_* = c$

- Gradient descent with $\gamma = 1/L$:

$$\begin{aligned}\theta_t &= \theta_{t-1} - \frac{1}{L}(H\theta_{t-1} - c) = \theta_{t-1} - \frac{1}{L}(H\theta_{t-1} - H\theta_*) \\ \theta_t - \theta_* &= (I - \frac{1}{L}H)(\theta_{t-1} - \theta_*) = (I - \frac{1}{L}H)^t(\theta_0 - \theta_*)\end{aligned}$$

Gradient descent - Proof for quadratic functions

- Quadratic convex function: $g(\theta) = \frac{1}{2}\theta^\top H\theta - c^\top\theta$
  - μ and L are the smallest and largest eigenvalues of H
  - Global optimum $\theta_* = H^{-1}c$ (or $H^\dagger c$) such that $H\theta_* = c$

- Gradient descent with $\gamma = 1/L$:

$$\begin{aligned}\theta_t &= \theta_{t-1} - \frac{1}{L}(H\theta_{t-1} - c) = \theta_{t-1} - \frac{1}{L}(H\theta_{t-1} - H\theta_*) \\ \theta_t - \theta_* &= (I - \frac{1}{L}H)(\theta_{t-1} - \theta_*) = (I - \frac{1}{L}H)^t(\theta_0 - \theta_*)\end{aligned}$$

- **Strong convexity** $\mu > 0$: eigenvalues of $(I - \frac{1}{L}H)^t$ in $[0, (1 - \frac{\mu}{L})^t]$
  - Convergence of iterates: $\|\theta_t - \theta_*\|^2 \leqslant (1 - \mu/L)^{2t}\|\theta_0 - \theta_*\|^2$
  - Function values: $g(\theta_t) - g(\theta_*) \leqslant (1 - \mu/L)^{2t}\big[g(\theta_0) - g(\theta_*)\big]$

Gradient descent - Proof for quadratic functions

- Quadratic convex function: $g(\theta) = \frac{1}{2}\theta^\top H\theta - c^\top\theta$
  - μ and L are the smallest and largest eigenvalues of H
  - Global optimum $\theta_* = H^{-1}c$ (or $H^\dagger c$) such that $H\theta_* = c$

- Gradient descent with $\gamma = 1/L$:

$$\begin{aligned}\theta_t &= \theta_{t-1} - \frac{1}{L}(H\theta_{t-1} - c) = \theta_{t-1} - \frac{1}{L}(H\theta_{t-1} - H\theta_*) \\ \theta_t - \theta_* &= (I - \frac{1}{L}H)(\theta_{t-1} - \theta_*) = (I - \frac{1}{L}H)^t(\theta_0 - \theta_*)\end{aligned}$$

- **Convexity** $\mu = 0$: eigenvalues of $(I - \frac{1}{L}H)^t$ in $[0, 1]$
  - No convergence of iterates: $\|\theta_t - \theta_*\|^2 \leqslant \|\theta_0 - \theta_*\|^2$
  - Function values: $g(\theta_t) - g(\theta_*) \leqslant \max_{v\in[0,L]} v(1-v/L)^{2t}\|\theta_0 - \theta_*\|^2$
    $g(\theta_t) - g(\theta_*) \leqslant \frac{L}{t}\|\theta_0 - \theta_*\|^2$ *(board)*

Board

- No convergence of iterates: $\|\theta_t - \theta_*\|^2 \leqslant \|\theta_0 - \theta_*\|^2$
- $g(\theta_t) - g(\theta_*) = \frac{1}{2}(\theta_t - \theta_*)^\top H(\theta_t - \theta_*)$, which is equal to

$$\frac{1}{2}(\theta_0 - \theta_*)^\top H(I - \frac{1}{L}H)^{2t}(\theta_0 - \theta_*)$$

- Function values: $g(\theta_t) - g(\theta_*) \leqslant \max_{v \in [0,L]} v(1 - v/L)^{2t}\|\theta_0 - \theta_*\|^2$

Board

- No convergence of iterates: $\|\theta_t - \theta_*\|^2 \leqslant \|\theta_0 - \theta_*\|^2$
- $g(\theta_t) - g(\theta_*) = \frac{1}{2}(\theta_t - \theta_*)^\top H(\theta_t - \theta_*)$, which is equal to

$$\frac{1}{2}(\theta_0 - \theta_*)^\top H(I - \frac{1}{L}H)^{2t}(\theta_0 - \theta_*)$$

- Function values: $g(\theta_t) - g(\theta_*) \leqslant \max_{v \in [0,L]} v(1 - v/L)^{2t}\|\theta_0 - \theta_*\|^2$

$$\begin{aligned} v(1 - v/L)^{2t} &\leqslant v\exp(-v/L)^{2t} = v\exp(-2tv/L) \\ &\leqslant (2tv/L)\exp(-2tv/L) \times \frac{L}{2t} \\ &\leqslant \max_{\alpha \geqslant 0} \alpha\exp(-\alpha) \times \frac{L}{2t} = O(\frac{L}{2t}) \end{aligned}$$

Proof for strongly-convex functions

- Using the correct inequality is key! Here Kurdyka-Lojasiewicz:

$$g(\theta) - g(\theta_*) \leqslant \frac{1}{2\mu}\|g'(\theta)\|^2$$

Proof for strongly-convex functions

- Using the correct inequality is key! Here Kurdyka-Lojasiewicz:

$$g(\theta) - g(\theta_*) \leqslant \frac{1}{2\mu}\|g'(\theta)\|^2$$

- Assuming $\gamma \leqslant 1/L$

$$\begin{aligned}
g(\theta_t) &\leqslant g(\theta_{t-1}) + g'(\theta_{t-1})^\top(-\gamma g'(\theta_{t-1})) + \frac{L}{2}\|-\gamma g'(\theta_{t-1})\|^2 \\
g(\theta_t) &\leqslant g(\theta_{t-1}) - \gamma(1 - L\gamma/2)\|g'(\theta_{t-1})\|^2 \\
g(\theta_t) &\leqslant g(\theta_{t-1}) - \frac{\gamma}{2}\|g'(\theta_{t-1})\|^2 \leqslant g(\theta_{t-1}) - \gamma\mu(g(\theta_{t-1}) - g(\theta_*))
\end{aligned}$$

  leading to $g(\theta_t) - g(\theta_*) \leqslant (1 - \gamma\mu)(g(\theta_{t-1}) - g(\theta_*))$

- See *automatic* proofs by Taylor et al. (2017); Taylor and Bach (2019)

Accelerated gradient methods (Nesterov, 1983)

- **Assumptions**
  - g convex with L-Lipschitz-cont. gradient , min. attained at θ_*
- **Algorithm**:

$$\begin{aligned} \theta_t &= \eta_{t-1} - \frac{1}{L} g'(\eta_{t-1}) \\ \eta_t &= \theta_t + \frac{t-1}{t+2}(\theta_t - \theta_{t-1}) \end{aligned}$$

- **Bound**:

$$g(\theta_t) - g(\theta_*) \leqslant \frac{2L\|\theta_0 - \theta_*\|^2}{(t+1)^2}$$

- Ten-line proof (see, e.g., Schmidt, Le Roux, and Bach, 2011)
- Not improvable
- Extension to strongly-convex functions

Accelerated gradient methods - strong convexity

- **Assumptions**
  - g convex with L-Lipschitz-cont. gradient , min. attained at θ_*
  - g μ-strongly convex

- **Algorithm**:

$$\begin{aligned}\theta_t &= \eta_{t-1} - \frac{1}{L} g'(\eta_{t-1}) \\ \eta_t &= \theta_t + \frac{1-\sqrt{\mu/L}}{1+\sqrt{\mu/L}}(\theta_t - \theta_{t-1})\end{aligned}$$

- **Bound**: $g(\theta_t) - g(\theta_*) \leqslant L\|\theta_0 - \theta_*\|^2 (1 - \sqrt{\mu/L})^t$
  - Ten-line proof (see, e.g., Schmidt, Le Roux, and Bach, 2011)
  - Not improvable
  - Relationship with conjugate gradient for quadratic functions

Optimization for sparsity-inducing norms
(see Bach, Jenatton, Mairal, and Obozinski, 2012a)

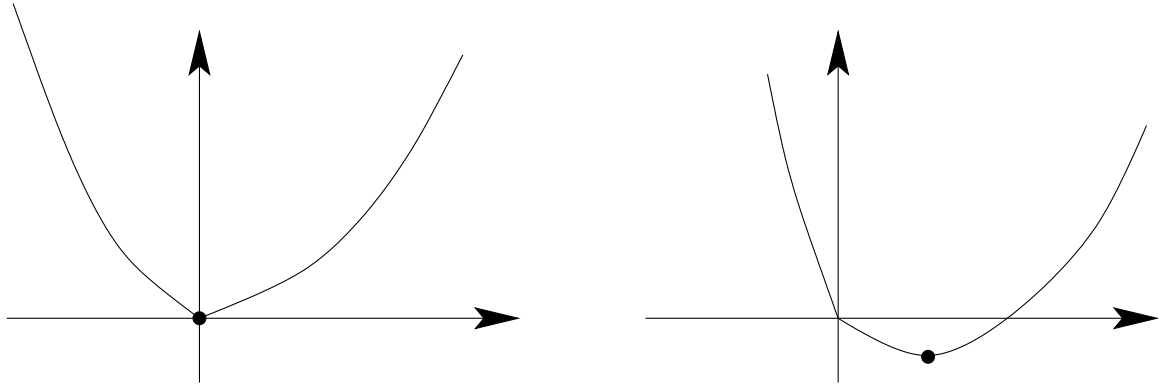
- Gradient descent as a **proximal method** (differentiable functions)
  - $\theta_{t+1} = \arg\min_{\theta \in \mathbb{R}^d} f(\theta_t) + (\theta - \theta_t)^\top f'(\theta_t) + \frac{L}{2}\|\theta - \theta_t\|_2^2$
  - $\theta_{t+1} = \theta_t - \frac{1}{L} f'(\theta_t)$

Optimization for sparsity-inducing norms
(see Bach, Jenatton, Mairal, and Obozinski, 2012a)

- Gradient descent as a **proximal method** (differentiable functions)
  - $\theta_{t+1} = \arg\min_{\theta \in \mathbb{R}^d} f(\theta_t) + (\theta - \theta_t)^\top f'(\theta_t) + \frac{L}{2}\|\theta - \theta_t\|_2^2$
  - $\theta_{t+1} = \theta_t - \frac{1}{L} f'(\theta_t)$
- Problems of the form: $\boxed{\min_{\theta \in \mathbb{R}^d} f(\theta) + \mu\Omega(\theta)}$
  - $\theta_{t+1} = \arg\min_{\theta \in \mathbb{R}^d} f(\theta_t) + (\theta - \theta_t)^\top f'(\theta_t) + \mu\Omega(\theta) + \frac{L}{2}\|\theta - \theta_t\|_2^2$
  - $\Omega(\theta) = \|\theta\|_1 \Rightarrow$ **Thresholded gradient descent**
- Similar convergence rates than smooth optimization
  - Acceleration methods (Nesterov, 2007; Beck and Teboulle, 2009)

Soft-thresholding for the ℓ_1-norm

- **Example 1**: quadratic problem in 1D, i.e. $\boxed{\min_{x\in\mathbb{R}} \frac{1}{2}x^2 - xy + \lambda|x|}$
- Piecewise quadratic function with a kink at zero
  - Derivative at $0+$: $g_+ = \lambda - y$ and $0-$: $g_- = -\lambda - y$

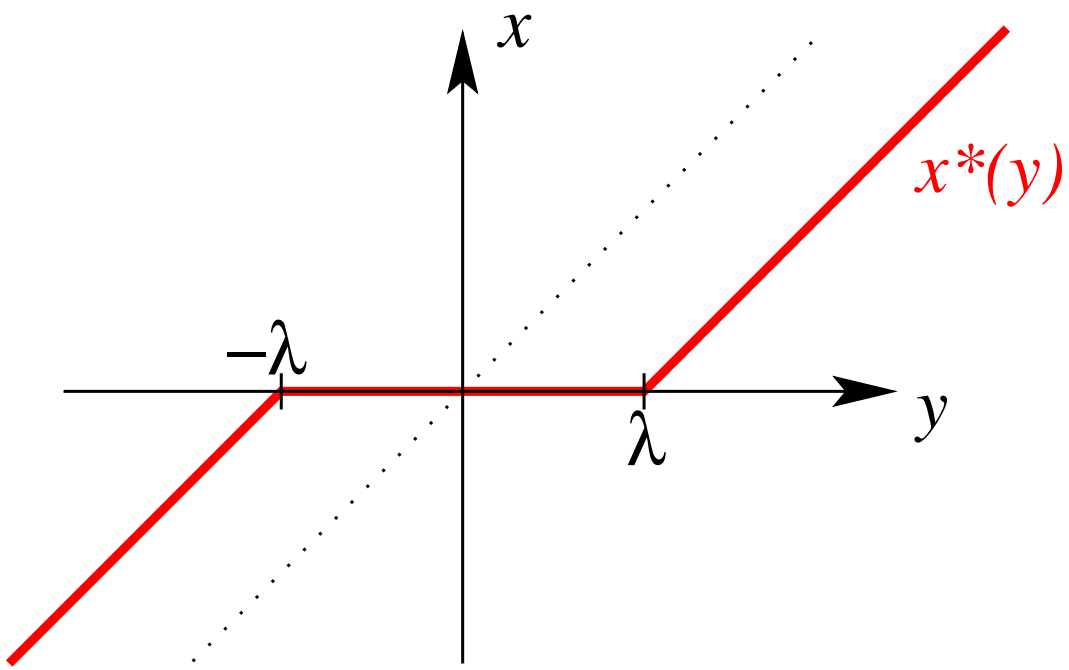


  - $x = 0$ is the solution iff $g_+ \geqslant 0$ and $g_- \leqslant 0$ (i.e., $|y| \leqslant \lambda$)
  - $x \geqslant 0$ is the solution iff $g_+ \leqslant 0$ (i.e., $y \geqslant \lambda$) $\Rightarrow x^* = y - \lambda$
  - $x \leqslant 0$ is the solution iff $g_- \geqslant 0$ (i.e., $y \leqslant -\lambda$) $\Rightarrow x^* = y + \lambda$

- Solution $\boxed{x^* = \mathrm{sign}(y)(|y| - \lambda)_+}$ = soft thresholding

Soft-thresholding for the ℓ_1-norm

- **Example 1**: quadratic problem in 1D, i.e. $\boxed{\min_{x\in\mathbb{R}} \frac{1}{2}x^2 - xy + \lambda|x|}$
- Piecewise quadratic function with a kink at zero
- Solution $\boxed{x^* = \mathrm{sign}(y)(|y| - \lambda)_+}$ = soft thresholding



Projected gradient descent

- Problems of the form: $\boxed{\min_{\theta \in \mathcal{K}} f(\theta)}$
  - $\theta_{t+1} = \arg\min_{\theta \in \mathcal{K}} f(\theta_t) + (\theta - \theta_t)^\top f'(\theta_t) + \frac{L}{2} \|\theta - \theta_t\|_2^2$
  - $\theta_{t+1} = \arg\min_{\theta \in \mathcal{K}} \frac{1}{2} \left\| \theta - \left(\theta_t - \frac{1}{L} f'(\theta_t) \right) \right\|_2^2$
  - Projected gradient descent
- Similar convergence rates than smooth optimization
  - Acceleration methods (Nesterov, 2007; Beck and Teboulle, 2009)

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex
- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate *(see board)*

Board

- Second-order Taylor expansion

$$g(\theta) \approx g(\theta_{t-1}) + g'(\theta_{t-1})^\top (\theta - \theta_{t-1}) + \frac{1}{2}(\theta - \theta_{t-1})^\top g''(\theta_{t-1})(\theta - \theta_{t-1})$$

  - Minimization by zeroing gradient:

$$g'(\theta_{t-1}) + g''(\theta_{t-1})(\theta - \theta_{t-1}) = 0$$

  - Iteration: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$

- Local quadratic convergence: $\|\theta_t - \theta_*\| = O(\|\theta_{t-1} - \theta_*\|^2)$

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex $\Leftrightarrow$ $O(\kappa \log \frac{1}{\varepsilon})$ iterations
- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate $\Leftrightarrow$ $O(\log \log \frac{1}{\varepsilon})$ iterations

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$
- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate $\Leftrightarrow$ complexity $= O((nd^2 + d^3) \cdot \log \log \frac{1}{\varepsilon})$

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$
- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate $\Leftrightarrow$ complexity $= O((nd^2 + d^3) \cdot \log \log \frac{1}{\varepsilon})$
- **Key insights for machine learning (Bottou and Bousquet, 2008)**
  1. No need to optimize below statistical error
  2. Objective functions are averages
  3. Testing error is more important than training error

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$
- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate $\Leftrightarrow$ complexity $= O((nd^2 + d^3) \cdot \log \log \frac{1}{\varepsilon})$
- **Key insights for machine learning (Bottou and Bousquet, 2008)**
  1. No need to optimize below statistical error
  2. Objective functions are averages
  3. Testing error is more important than training error

Outline

1. **Introduction/motivation: Supervised machine learning**
   - Machine learning $\approx$ optimization of finite sums
   - Batch optimization methods

2. **Fast stochastic gradient methods for convex problems**
   - Variance reduction: for *training* error
   - Constant step-sizes: for *testing* error

3. **Beyond convex problems**
   - Generic algorithms with generic "guarantees"
   - Global convergence for over-parameterized neural networks

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, **i.i.d.**
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \Big\{ \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta) \Big\} \qquad = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

data fitting term + regularizer

- Optimization: optimization of regularized risk — training cost
- Statistics: guarantees on $\mathbb{E}_{p(x,y)} \ell(y, h(x, \theta))$ — testing cost

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$
- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate $\Leftrightarrow$ complexity $= O((nd^2 + d^3) \cdot \log \log \frac{1}{\varepsilon})$
- **Key insights for machine learning (Bottou and Bousquet, 2008)**
  1. No need to optimize below statistical error
  2. Objective functions are averages
  3. Testing error is more important than training error

Stochastic gradient descent (SGD) for finite sums

$$\min_{\theta \in \mathbb{R}^d} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

- **Iteration**: $\theta_t = \theta_{t-1} - \gamma_t f'_{i(t)}(\theta_{t-1})$
  - Sampling with replacement: $i(t)$ random element of $\{1, \dots, n\}$
  - Polyak-Ruppert averaging: $\bar{\theta}_t = \frac{1}{t+1} \sum_{u=0}^{t} \theta_u$

Stochastic gradient descent (SGD) for finite sums

$$\min_{\theta \in \mathbb{R}^d} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

- **Iteration**: $\theta_t = \theta_{t-1} - \gamma_t f'_{i(t)}(\theta_{t-1})$
  - Sampling with replacement: $i(t)$ random element of $\{1, \dots, n\}$
  - Polyak-Ruppert averaging: $\bar{\theta}_t = \frac{1}{t+1} \sum_{u=0}^{t} \theta_u$

- **Convergence rate** if each f_i is convex L-smooth and g μ-strongly-convex:

$$\mathbb{E} g(\bar{\theta}_t) - g(\theta_*) \leqslant \begin{cases} O(1/\sqrt{t}) & \text{if } \gamma_t = 1/(L\sqrt{t}) \\ O(L/(\mu t)) = O(\kappa/t) & \text{if } \gamma_t = 1/(\mu t) \end{cases}$$

  - No adaptivity to strong-convexity in general
  - Running-time complexity: $O(d \cdot \kappa/\varepsilon)$

Where does κ/t come from? - I

- Iteration: $\theta_t = \theta_{t-1} - \gamma_t f'_{i(t)}(\theta_{t-1})$, with bounded gradients

$$\begin{aligned}
\|\theta_t - \theta_*\|^2 &= \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top f'_{i(t)}(\theta_{t-1}) + \gamma_t^2 \|f'_{i(t)}(\theta_{t-1})\|^2 \\
&\leqslant \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top f'_{i(t)}(\theta_{t-1}) + \gamma_t^2 B^2
\end{aligned}$$

Where does κ/t come from? - I

- Iteration: $\theta_t = \theta_{t-1} - \gamma_t f'_{i(t)}(\theta_{t-1})$, with bounded gradients

$$\begin{aligned}\|\theta_t - \theta_*\|^2 &= \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top f'_{i(t)}(\theta_{t-1}) + \gamma_t^2 \|f'_{i(t)}(\theta_{t-1})\|^2 \\ &\leqslant \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top f'_{i(t)}(\theta_{t-1}) + \gamma_t^2 B^2\end{aligned}$$

  - leading to, with the proper conditional expectation:

$$\mathbb{E}\big[\|\theta_t - \theta_*\|^2 | i(1), \dots, i(t-1)\big] \leqslant \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top g'(\theta_{t-1}) + \gamma_t^2 B^2$$

Where does κ/t come from? - I

- Iteration: $\theta_t = \theta_{t-1} - \gamma_t f'_{i(t)}(\theta_{t-1})$, with bounded gradients

$$\begin{aligned}\|\theta_t - \theta_*\|^2 &= \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top f'_{i(t)}(\theta_{t-1}) + \gamma_t^2 \|f'_{i(t)}(\theta_{t-1})\|^2 \\ &\leqslant \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top f'_{i(t)}(\theta_{t-1}) + \gamma_t^2 B^2\end{aligned}$$

  - leading to, with the proper conditional expectation:

$$\mathbb{E}\big[\|\theta_t - \theta_*\|^2 | i(1), \dots, i(t-1)\big] \leqslant \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top g'(\theta_{t-1}) + \gamma_t^2 B^2$$

  - Using another magical identity $g(\theta) + (\theta_* - \theta)^\top g'(\theta) + \frac{\mu}{2}\|\theta - \theta_*\|^2 \leqslant g(\theta_*)$:

$$\mathbb{E}\big[\|\theta_t - \theta_*\|^2 | i(1), \dots, i(t-1)\big] \leqslant (1 - \gamma_t\mu)\|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t[g(\theta_{t-1}) - g(\theta_*)] + \gamma_t^2 B^2$$

Where does κ/t come from? - I

- Iteration: $\theta_t = \theta_{t-1} - \gamma_t f'_{i(t)}(\theta_{t-1})$, with bounded gradients

$$\begin{aligned}\|\theta_t - \theta_*\|^2 &= \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top f'_{i(t)}(\theta_{t-1}) + \gamma_t^2 \|f'_{i(t)}(\theta_{t-1})\|^2 \\ &\leqslant \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top f'_{i(t)}(\theta_{t-1}) + \gamma_t^2 B^2\end{aligned}$$

  - leading to, with the proper conditional expectation:

$$\mathbb{E}\big[\|\theta_t - \theta_*\|^2 | i(1), \dots, i(t-1)\big] \leqslant \|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top g'(\theta_{t-1}) + \gamma_t^2 B^2$$

  - Using another magical identity $g(\theta) + (\theta_* - \theta)^\top g'(\theta) + \frac{\mu}{2}\|\theta - \theta_*\|^2 \leqslant g(\theta_*)$:

$$\mathbb{E}\big[\|\theta_t - \theta_*\|^2 | i(1), \dots, i(t-1)\big] \leqslant (1 - \gamma_t\mu)\|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t[g(\theta_{t-1}) - g(\theta_*)] + \gamma_t^2 B^2$$

  - Taking with full expectations:

$$\mathbb{E}\|\theta_t - \theta_*\|^2 \leqslant (1 - \gamma_t\mu)\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t\mathbb{E}[g(\theta_{t-1}) - g(\theta_*)] + \gamma_t^2 B^2$$

Where does κ/t come from? - II

- Starting from

$$\mathbb{E}\|\theta_t - \theta_*\|^2 \leqslant (1 - \gamma_t \mu)\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t \mathbb{E}[g(\theta_{t-1}) - g(\theta_*)] + \gamma_t^2 B^2$$

  - Isolate $\mathbb{E}[g(\theta_{t-1}) - g(\theta_*)] \leqslant \frac{1-\gamma_t\mu}{2\gamma_t}\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \frac{1}{2\gamma_t}\mathbb{E}\|\theta_t - \theta_*\|^2 + \frac{\gamma_t}{2}B^2$

Where does κ/t come from? - II

- Starting from

$$\mathbb{E}\|\theta_t - \theta_*\|^2 \leqslant (1-\gamma_t\mu)\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t\mathbb{E}[g(\theta_{t-1}) - g(\theta_*)] + \gamma_t^2 B^2$$

  - Isolate $\mathbb{E}[g(\theta_{t-1}) - g(\theta_*)] \leqslant \frac{1-\gamma_t\mu}{2\gamma_t}\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \frac{1}{2\gamma_t}\mathbb{E}\|\theta_t - \theta_*\|^2 + \frac{\gamma_t}{2}B^2$
  - With the step-size $\gamma_t = 1/(\mu t)$, we get a telescoping sum!

$$\mathbb{E}[g(\theta_{t-1}) - g(\theta_*)] \leqslant \mu\frac{t-1}{2}\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \mu\frac{t}{2}\mathbb{E}\|\theta_t - \theta_*\|^2 + \frac{1}{2\mu t}B^2$$

Where does κ/t come from? - II

- Starting from

$$\mathbb{E}\|\theta_t - \theta_*\|^2 \leqslant (1-\gamma_t\mu)\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - 2\gamma_t\mathbb{E}[g(\theta_{t-1}) - g(\theta_*)] + \gamma_t^2 B^2$$

  - Isolate $\mathbb{E}[g(\theta_{t-1}) - g(\theta_*)] \leqslant \frac{1-\gamma_t\mu}{2\gamma_t}\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \frac{1}{2\gamma_t}\mathbb{E}\|\theta_t - \theta_*\|^2 + \frac{\gamma_t}{2}B^2$
  - With the step-size $\gamma_t = 1/(\mu t)$, we get a telescoping sum!

$$\mathbb{E}[g(\theta_{t-1}) - g(\theta_*)] \leqslant \mu\frac{t-1}{2}\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \mu\frac{t}{2}\mathbb{E}\|\theta_t - \theta_*\|^2 + \frac{1}{2\mu t}B^2$$

  - Taking the sum from $t = 1$ to T, and using Jensen's inequality:

$$\mathbb{E}\Big[g\Big(\frac{1}{T}\sum_{t=1}^{T}\theta_{t-1}\Big) - g(\theta_*)\Big] \leqslant \frac{1}{2}\frac{B^2}{\mu}\frac{\log T}{T} = \frac{1}{2}\kappa\frac{\log T}{T}$$

  - Use $\gamma_t = 2/[\mu(t+1)]$ to remove $\log T$ factor (Lacoste-Julien et al., 2012)

Impact of averaging (Bach and Moulines, 2011)

- Stochastic gradient descent with learning rate $\gamma_t = Ct^{-\alpha}$
- **Strongly convex smooth objective functions**
  - Non-asymptotic analysis with explicit constants
  - Forgetting of initial conditions
  - Robustness to the choice of C

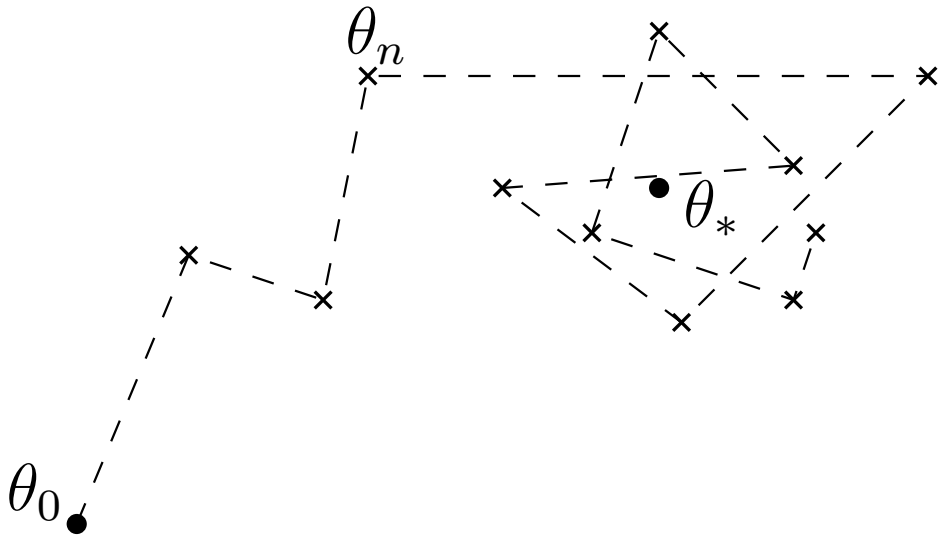
Impact of averaging (Bach and Moulines, 2011)

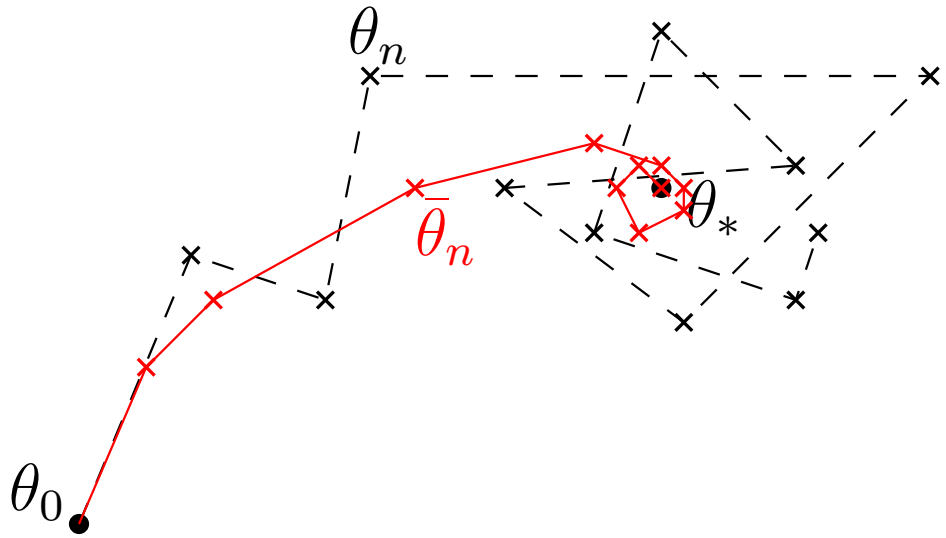
- Stochastic gradient descent with learning rate $\gamma_t = Ct^{-\alpha}$
- **Strongly convex smooth objective functions**
  - Non-asymptotic analysis with explicit constants
  - Forgetting of initial conditions
  - Robustness to the choice of C
- **Convergence rates** for $\mathbb{E}\|\theta_t - \theta_*\|^2$ and $\mathbb{E}\|\bar{\theta}_t - \theta_*\|^2$
  - no averaging: $O\Big(\frac{\sigma^2\gamma_t}{\mu}\Big) + O(e^{-\mu t\gamma_t})\|\theta_0 - \theta_*\|^2$
  - averaging: $\frac{\operatorname{tr} H(\theta_*)^{-1}}{t} + O\Big(\frac{\|\theta_0 - \theta_*\|^2}{\mu^2 t^2}\Big)$

    (see board)

Board

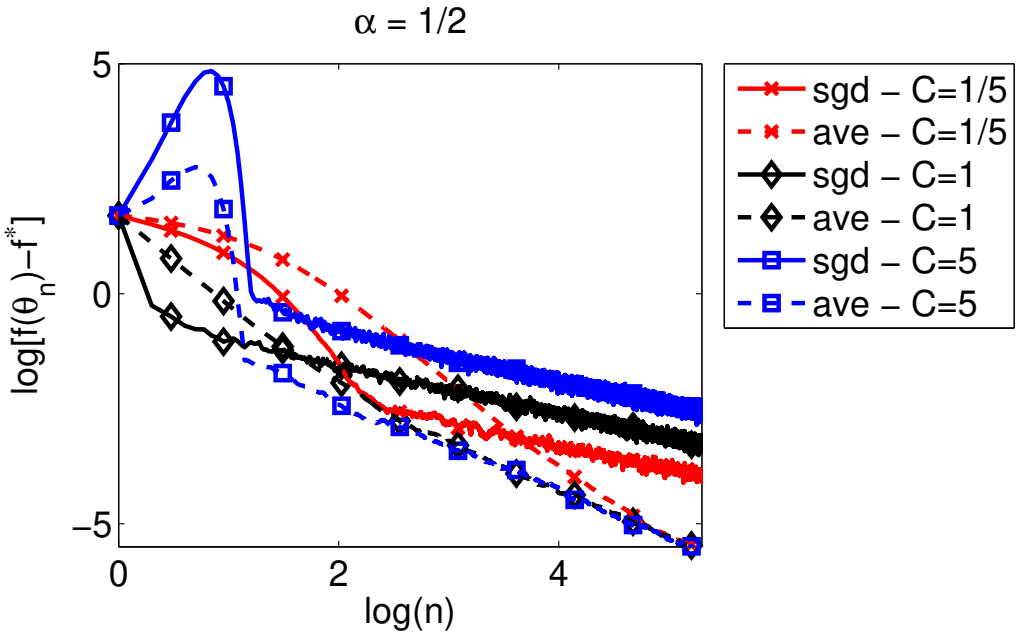
- Leaving initial point θ_0 to reach θ_*
- Impact of averaging

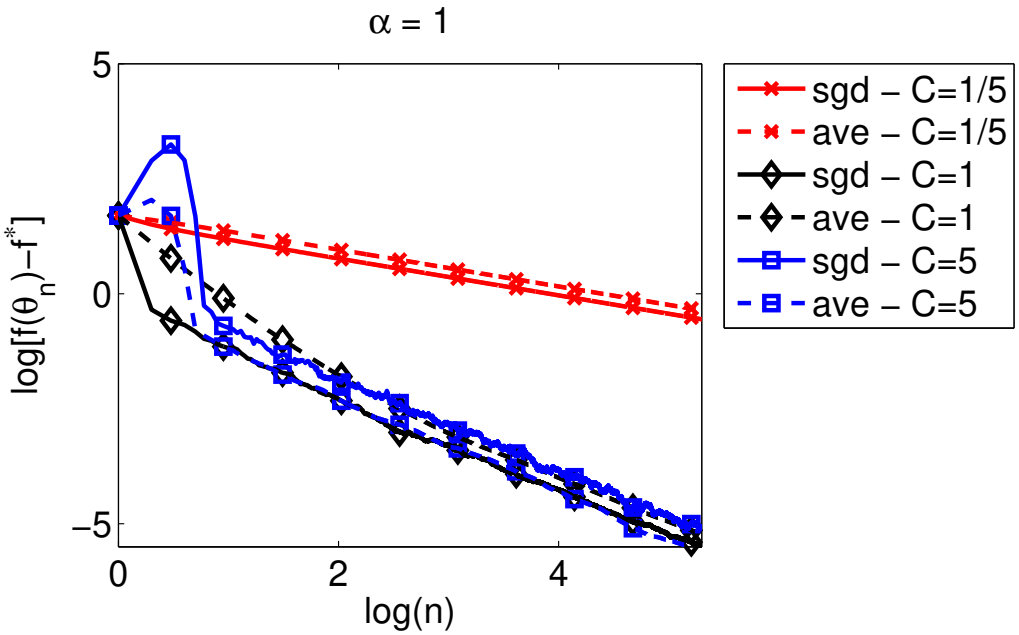




Robustness to wrong constants for $\gamma_t = Ct^{-\alpha}$

- $f(\theta) = \frac{1}{2}|\theta|^2$ with i.i.d. Gaussian noise ($d = 1$)
- Left: $\alpha = 1/2$
- Right: $\alpha = 1$





- See also `http://leon.bottou.org/projects/sgd`

Stochastic vs. deterministic methods

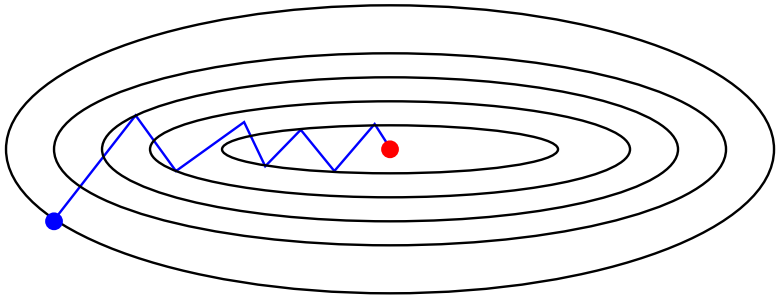
- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$

Stochastic vs. deterministic methods

- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$
- Batch gradient descent: $\theta_t = \theta_{t-1} - \gamma_t g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_t}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$
  - Linear (e.g., exponential) convergence rate in $O(e^{-t/\kappa})$
  - Iteration complexity is linear in n

Stochastic vs. deterministic methods

- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$
- Batch gradient descent: $\theta_t = \theta_{t-1} - \gamma_t g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_t}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$

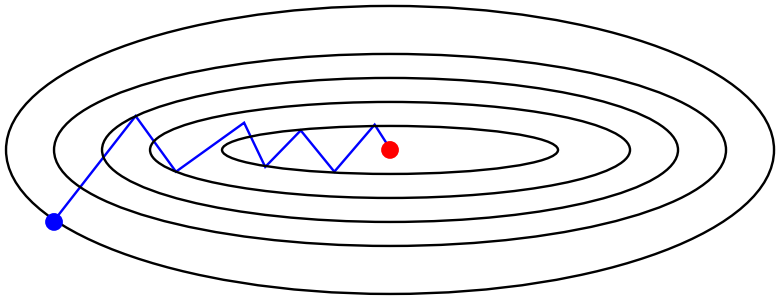


Stochastic vs. deterministic methods

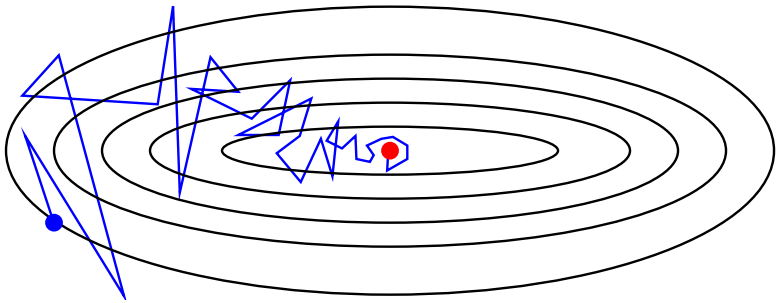
- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$
- Batch gradient descent: $\theta_t = \theta_{t-1} - \gamma_t g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_t}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$
  - Linear (e.g., exponential) convergence rate in $O(e^{-t/\kappa})$
  - Iteration complexity is linear in n
- Stochastic gradient descent: $\theta_t = \theta_{t-1} - \gamma_t f_{i(t)}'(\theta_{t-1})$
  - Sampling with replacement: $i(t)$ random element of $\{1, \ldots, n\}$
  - Convergence rate in $O(\kappa/t)$
  - Iteration complexity is independent of n

Stochastic vs. deterministic methods

- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$
- Batch gradient descent: $\theta_t = \theta_{t-1} - \gamma_t g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_t}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$

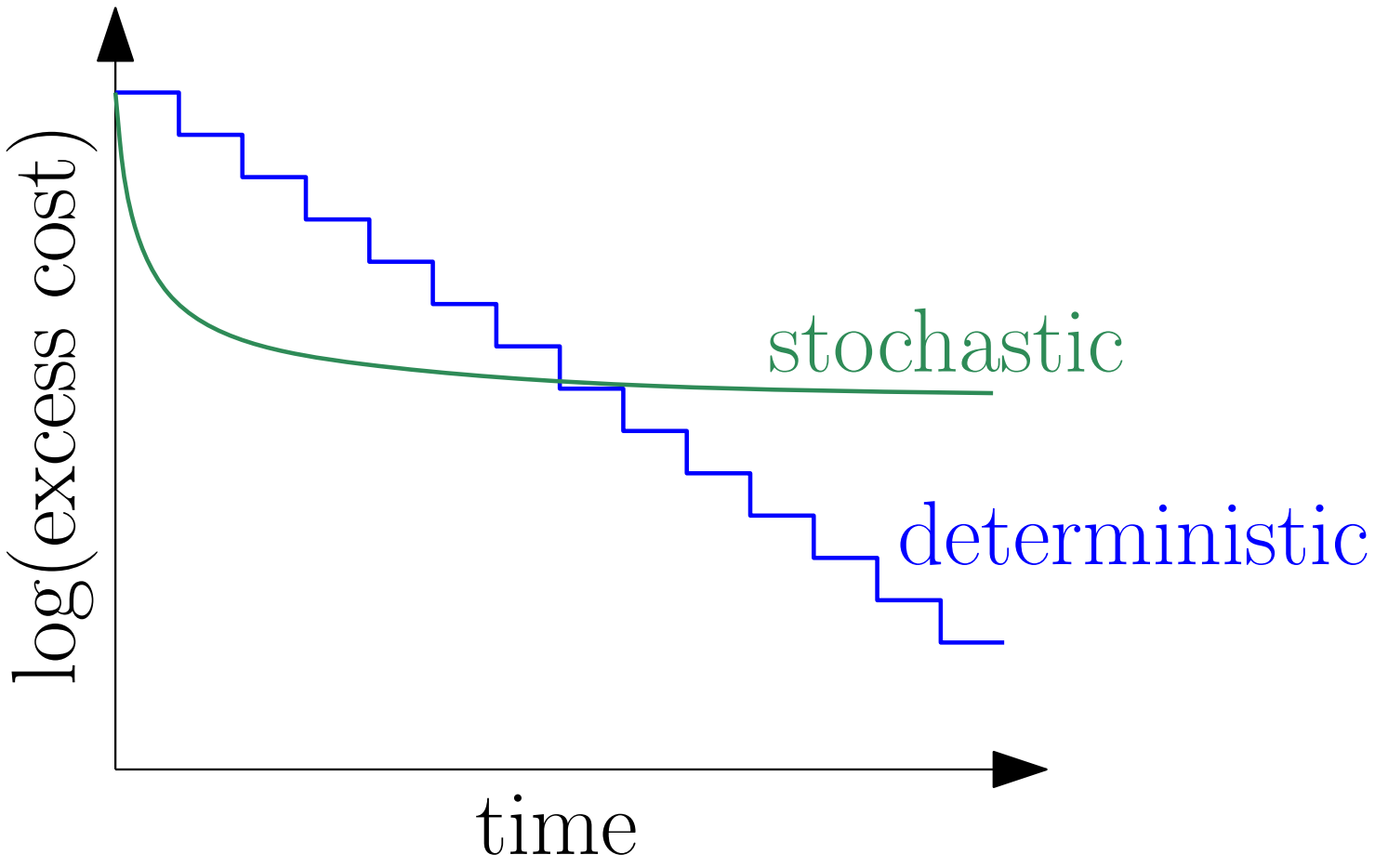


- Stochastic gradient descent: $\theta_t = \theta_{t-1} - \gamma_t f_{i(t)}'(\theta_{t-1})$



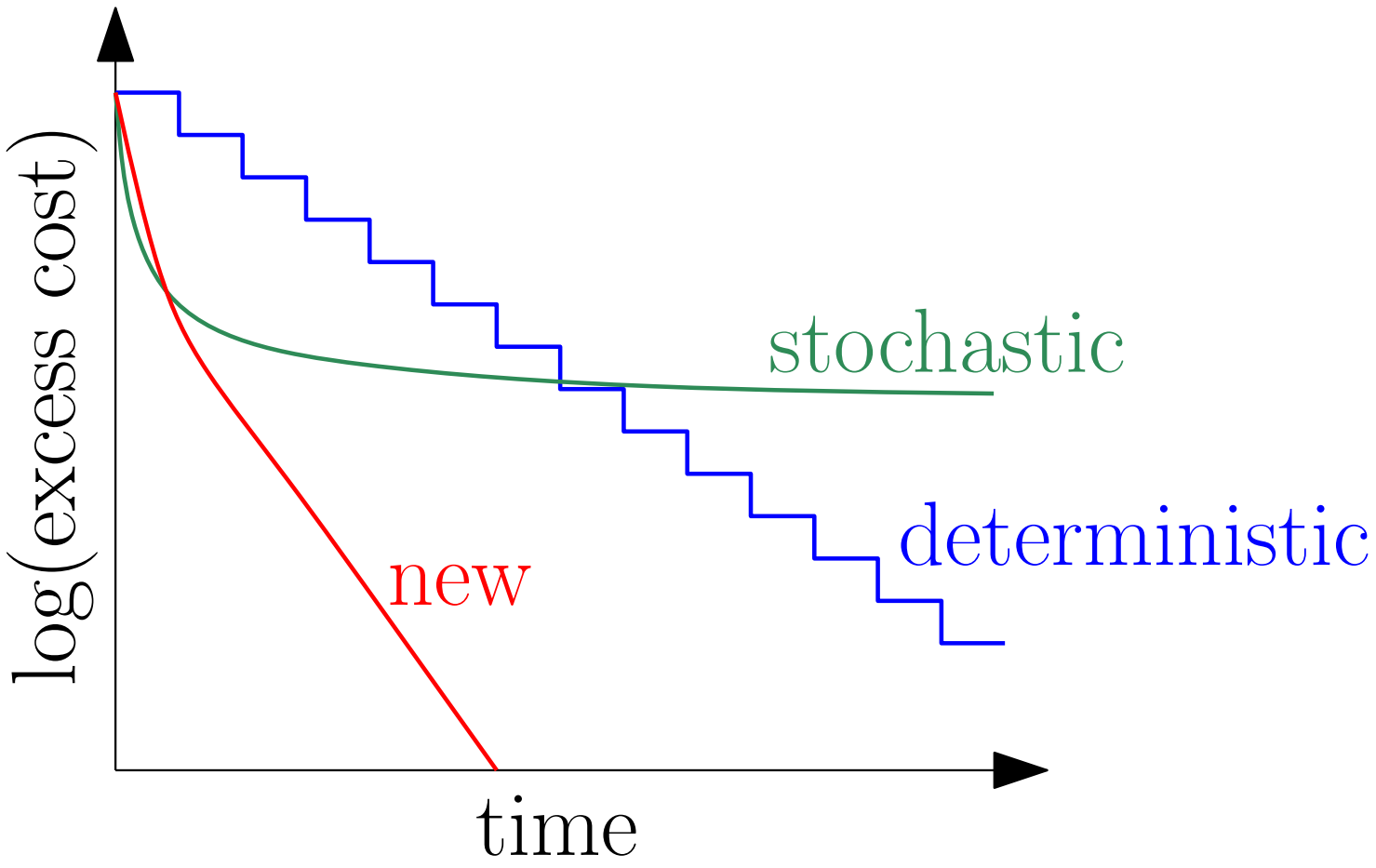
Stochastic vs. deterministic methods

- **Goal = best of both worlds**: Linear rate with $O(d)$ iteration cost
  Simple choice of step size



Stochastic vs. deterministic methods

- **Goal = best of both worlds**: Linear rate with $O(d)$ iteration cost
  Simple choice of step size



Accelerating gradient methods - Related work

- **Generic acceleration** (Nesterov, 1983, 2004)

$$\theta_t = \eta_{t-1} - \gamma_t g'(\eta_{t-1}) \text{ and } \eta_t = \theta_t + \delta_t(\theta_t - \theta_{t-1})$$

Accelerating gradient methods - Related work

- **Generic acceleration** (Nesterov, 1983, 2004)

$$\theta_t = \eta_{t-1} - \gamma_t g'(\eta_{t-1}) \text{ and } \eta_t = \theta_t + \delta_t(\theta_t - \theta_{t-1})$$

  - Good choice of momentum term $\delta_t \in [0, 1)$

    $g(\theta_t) - g(\theta_*) \leqslant O(1/t^2)$

    $g(\theta_t) - g(\theta_*) \leqslant O(e^{-t\sqrt{\mu/L}}) = O(e^{-t/\sqrt{\kappa}})$ if μ-strongly convex
  - Optimal rates after $t = O(d)$ iterations (Nesterov, 2004)

Accelerating gradient methods - Related work

- **Generic acceleration** (Nesterov, 1983, 2004)

$$\theta_t = \eta_{t-1} - \gamma_t g'(\eta_{t-1}) \text{ and } \eta_t = \theta_t + \delta_t(\theta_t - \theta_{t-1})$$

  - Good choice of momentum term $\delta_t \in [0, 1)$

    $g(\theta_t) - g(\theta_*) \leqslant O(1/t^2)$

    $g(\theta_t) - g(\theta_*) \leqslant O(e^{-t\sqrt{\mu/L}}) = O(e^{-t/\sqrt{\kappa}})$ if μ-strongly convex
  - Optimal rates after $t = O(d)$ iterations (Nesterov, 2004)
  - Still $O(nd)$ iteration cost: complexity $= O(nd \cdot \sqrt{\kappa} \log \frac{1}{\varepsilon})$

Accelerating gradient methods - Related work

- **Constant step-size stochastic gradient**
  - Solodov (1998); Nedic and Bertsekas (2000)
  - Linear convergence, but only up to a fixed tolerance

Accelerating gradient methods - Related work

- **Constant step-size stochastic gradient**
  - Solodov (1998); Nedic and Bertsekas (2000)
  - Linear convergence, but only up to a fixed tolerance

- **Stochastic methods in the dual (SDCA)**
  - Shalev-Shwartz and Zhang (2013)
  - Similar linear rate but limited choice for the f_i's
  - Extensions without duality: see Shalev-Shwartz (2016)

Accelerating gradient methods - Related work

- **Constant step-size stochastic gradient**
  - Solodov (1998); Nedic and Bertsekas (2000)
  - Linear convergence, but only up to a fixed tolerance

- **Stochastic methods in the dual (SDCA)**
  - Shalev-Shwartz and Zhang (2013)
  - Similar linear rate but limited choice for the f_i's
  - Extensions without duality: see Shalev-Shwartz (2016)

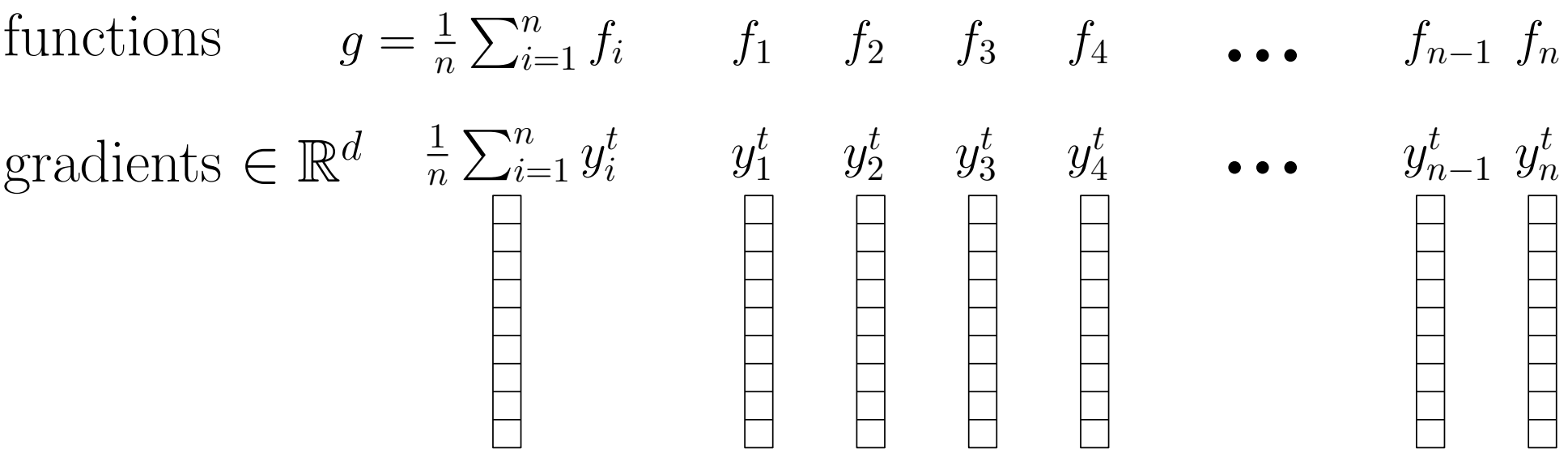
- **Stochastic version of accelerated batch gradient methods**
  - Tseng (1998); Ghadimi and Lan (2010); Xiao (2010)
  - Can improve constants, but still have sublinear $O(1/t)$ rate

Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \ldots, n$
  - Random selection $i(t) \in \{1, \ldots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \dfrac{\gamma_t}{n} \displaystyle\sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$

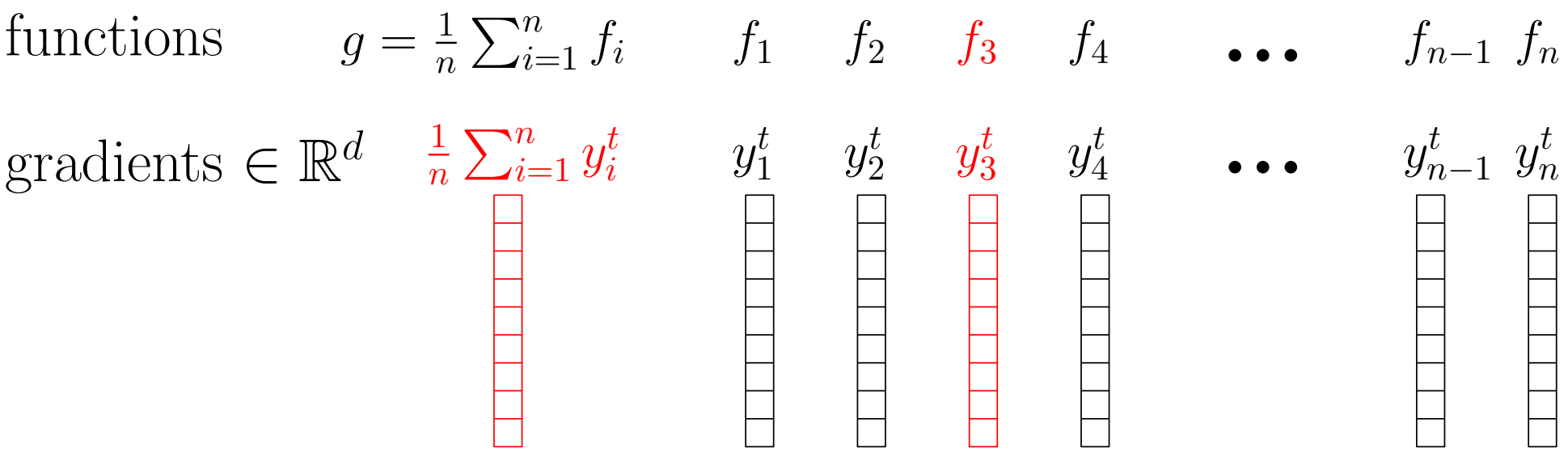
Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \ldots, n$
  - Random selection $i(t) \in \{1, \ldots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \dfrac{\gamma_t}{n} \displaystyle\sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$



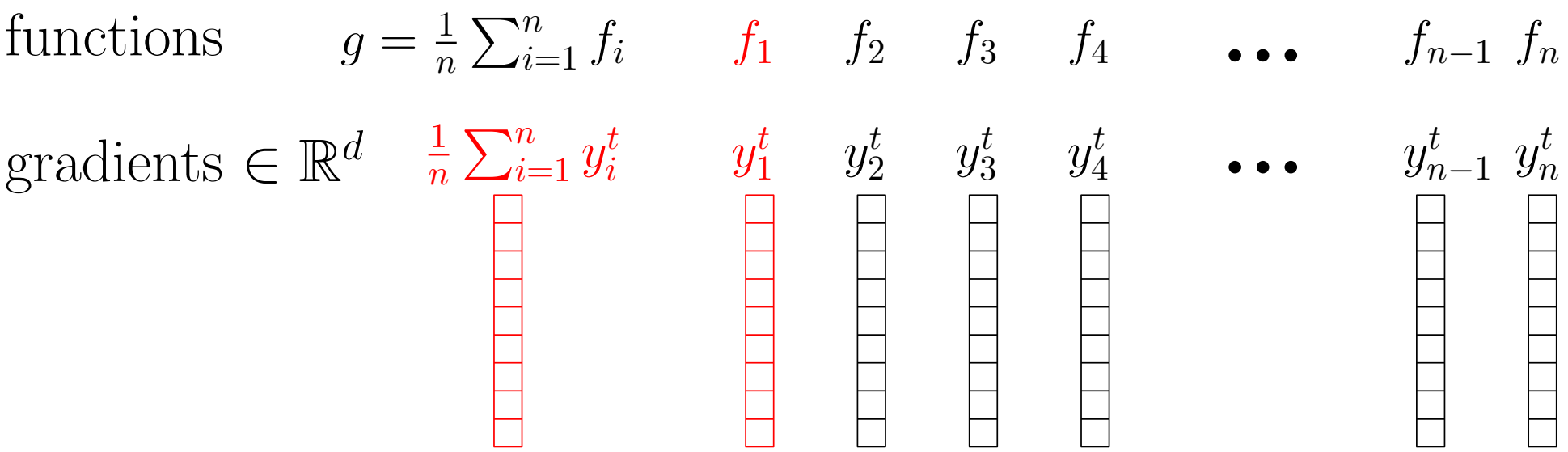
Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \ldots, n$
  - Random selection $i(t) \in \{1, \ldots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \dfrac{\gamma_t}{n} \displaystyle\sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$



Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \ldots, n$
  - Random selection $i(t) \in \{1, \ldots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \dfrac{\gamma_t}{n} \displaystyle\sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$



Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \ldots, n$
  - Random selection $i(t) \in \{1, \ldots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \dfrac{\gamma_t}{n} \displaystyle\sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$
- Stochastic version of incremental average gradient (Blatt et al., 2008)

Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \ldots, n$
  - Random selection $i(t) \in \{1, \ldots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \dfrac{\gamma_t}{n} \displaystyle\sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$
- Stochastic version of incremental average gradient (Blatt et al., 2008)
- **Extra memory requirement**: n gradients in $\mathbb{R}^d$ in general
- **Linear supervised machine learning**: only n real numbers
  - If $f_i(\theta) = \ell(y_i, \Phi(x_i)^\top \theta)$, then $f_i'(\theta) = \ell'(y_i, \Phi(x_i)^\top \theta)\, \Phi(x_i)$

Running-time comparisons (strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each f_i convex L-smooth and g μ-strongly convex, $\kappa = L/\mu$

Stochastic gradient descent	$d \times \frac{L}{\mu} \times \frac{1}{\varepsilon}$
Gradient descent	$d \times n\frac{L}{\mu} \times \log\frac{1}{\varepsilon}$
Accelerated gradient descent	$d \times n\sqrt{\frac{L}{\mu}} \times \log\frac{1}{\varepsilon}$
SAG	$d \times (n + \frac{L}{\mu}) \times \log\frac{1}{\varepsilon}$

NB: slightly different (smaller) notion of condition number for batch methods

Running-time comparisons (strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each f_i convex L-smooth and g μ-strongly convex, $\kappa = L/\mu$

Stochastic gradient descent	$d \times \frac{L}{\mu} \times \frac{1}{\varepsilon}$
Gradient descent	$d \times n\frac{L}{\mu} \times \log\frac{1}{\varepsilon}$
Accelerated gradient descent	$d \times n\sqrt{\frac{L}{\mu}} \times \log\frac{1}{\varepsilon}$
SAG	$d \times (n + \frac{L}{\mu}) \times \log\frac{1}{\varepsilon}$

- **Beating two lower bounds** (Nemirovski and Yudin, 1983; Nesterov, 2004): with additional assumptions

(1) stochastic gradient: exponential rate for finite sums
(2) full gradient: better exponential rate using the sum structure

Running-time comparisons (non-strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each f_i convex L-smooth
  - Ill conditioned problems: g may not be strongly-convex ($\mu = 0$)

Stochastic gradient descent	$d\times \quad 1/\varepsilon^2$
Gradient descent	$d\times \quad n/\varepsilon$
Accelerated gradient descent	$d\times \quad n/\sqrt{\varepsilon}$
SAG	$d\times \quad \sqrt{n}/\varepsilon$

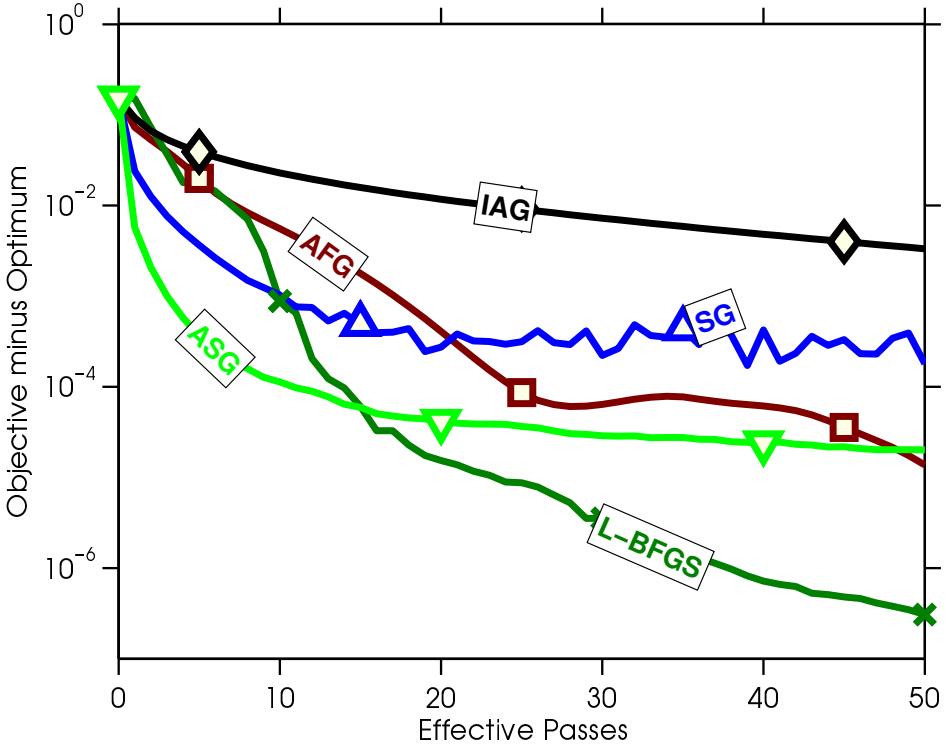
- Adaptivity to potentially hidden strong convexity
- No need to know the local/global strong-convexity constant

Stochastic average gradient
Implementation details and extensions

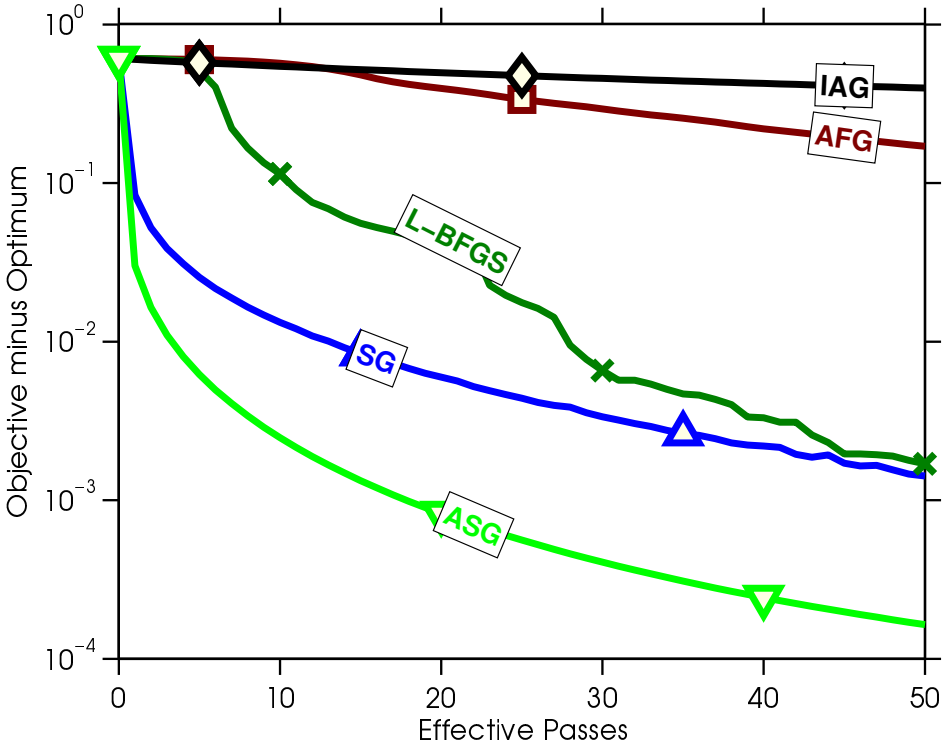
- **Sparsity in the features**
  - Just-in-time updates $\Rightarrow$ replace $O(d)$ by number of non zeros
  - See also Leblond, Pedregosa, and Lacoste-Julien (2016)
- **Mini-batches**
  - Reduces the memory requirement + block access to data
- **Line-search**
  - Avoids knowing L in advance
- **Non-uniform sampling**
  - Favors functions with large variations
- See `www.cs.ubc.ca/~schmidtm/Software/SAG.html`

Experimental results (logistic regression)

`quantum` dataset
($n = 50\ 000$, $d = 78$)

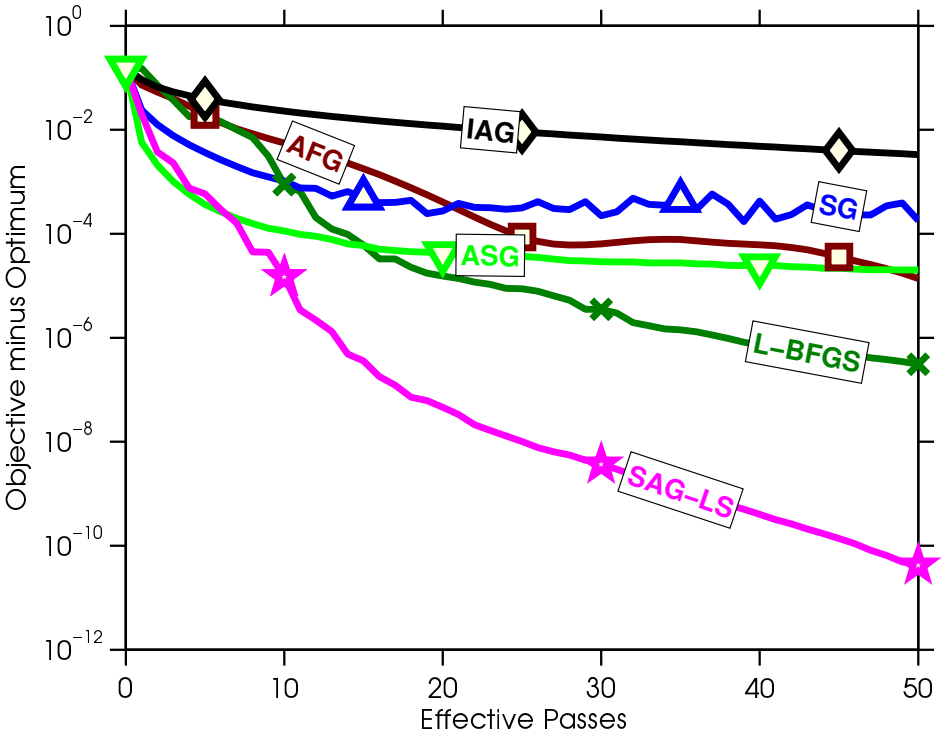


`rcv1` dataset
($n = 697\ 641$, $d = 47\ 236$)

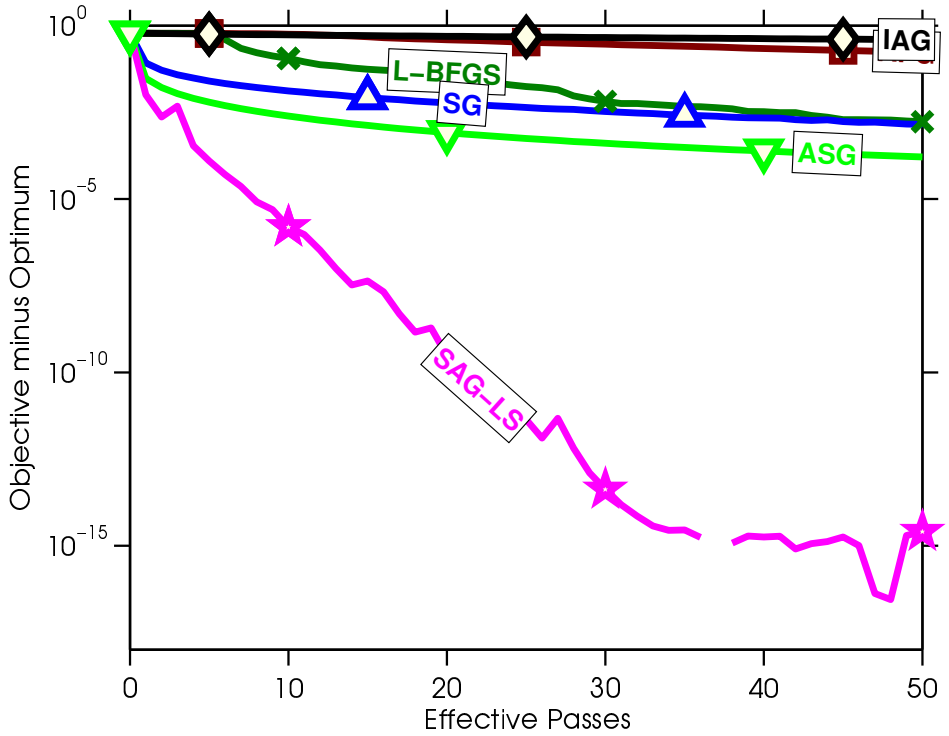


Experimental results (logistic regression)

`quantum` dataset
$(n = 50\ 000,\ d = 78)$

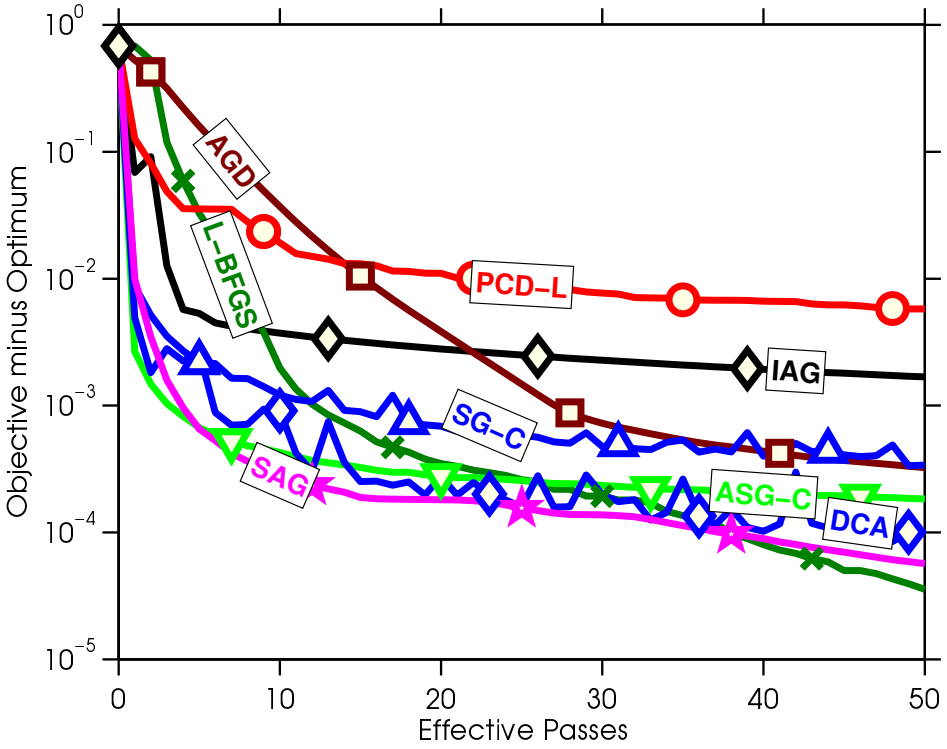


`rcv1` dataset
$(n = 697\ 641,\ d = 47\ 236)$

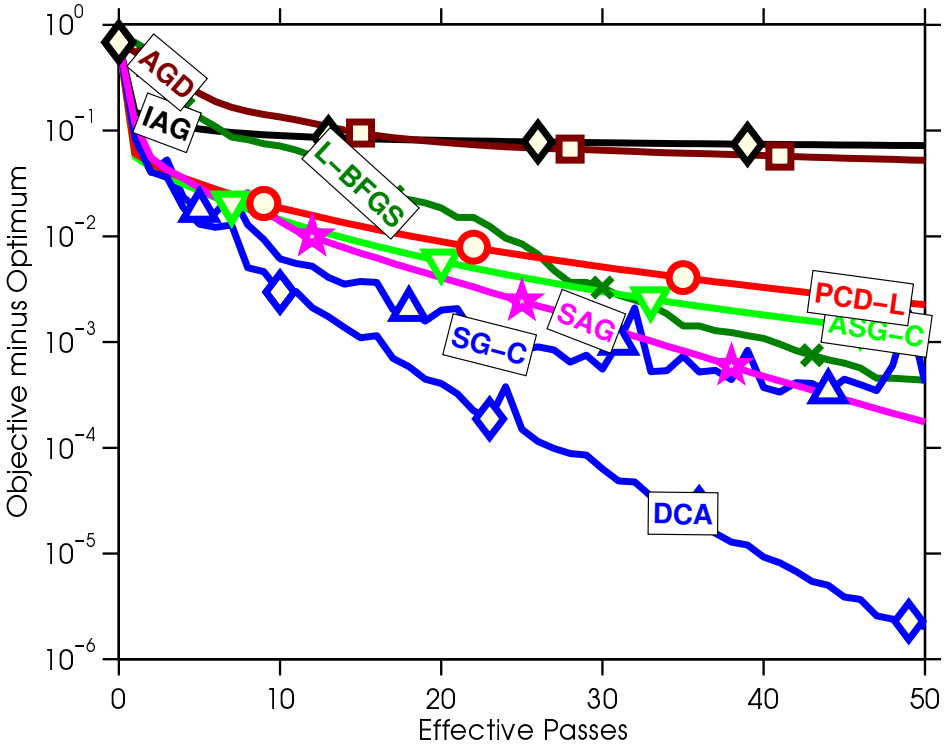


Before non-uniform sampling

`protein` dataset
$(n = 145\ 751,\ d = 74)$

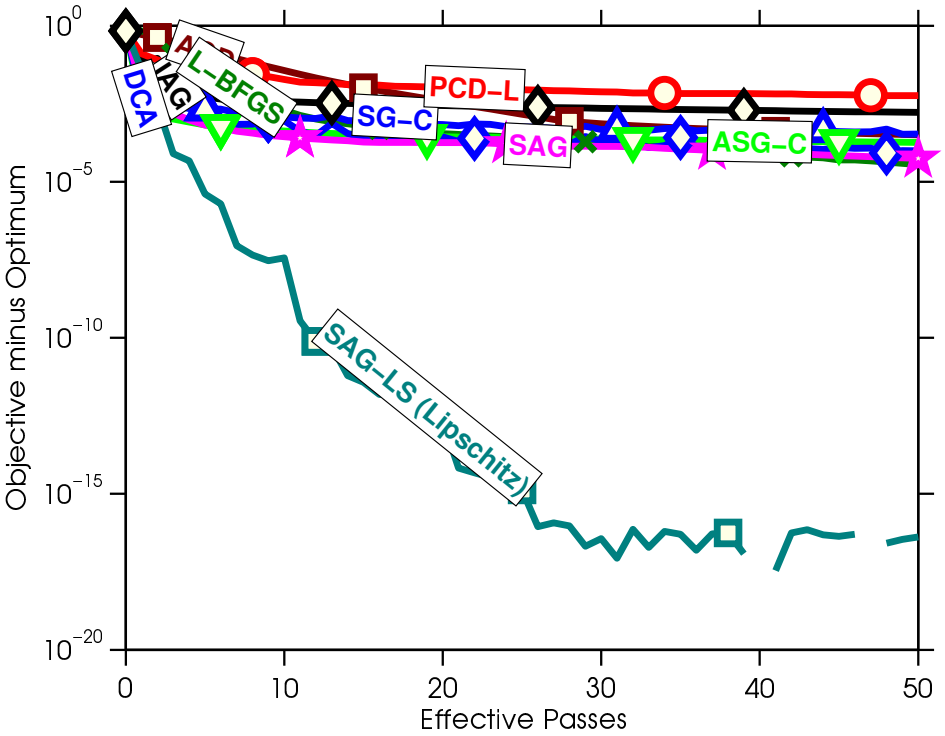


`sido` dataset
$(n = 12\ 678,\ d = 4\ 932)$

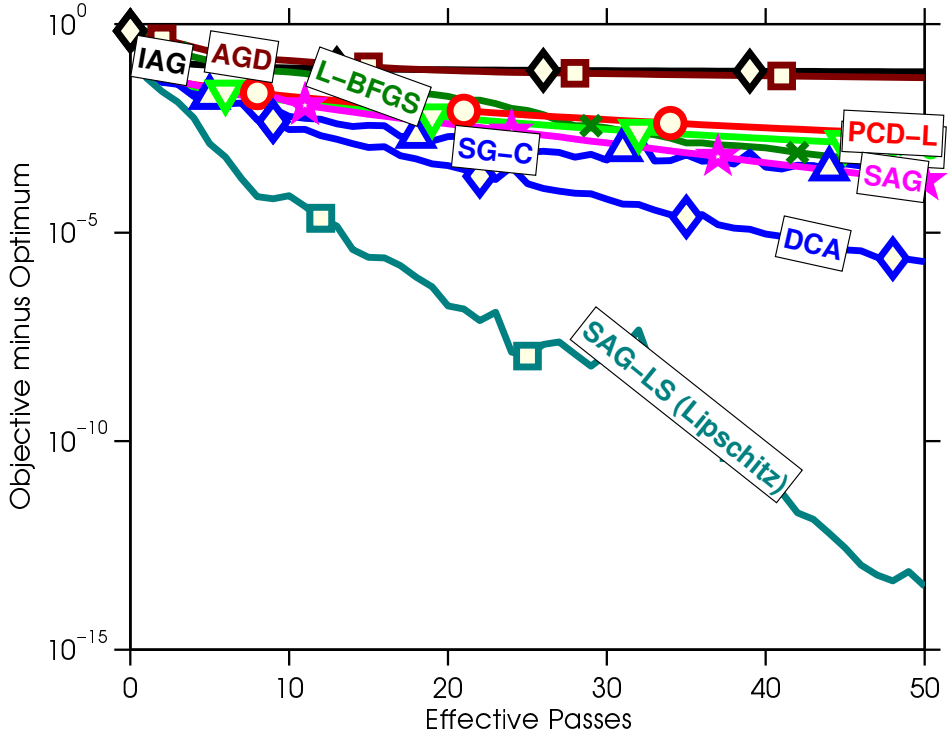


After non-uniform sampling

`protein` dataset
$(n = 145\ 751,\ d = 74)$

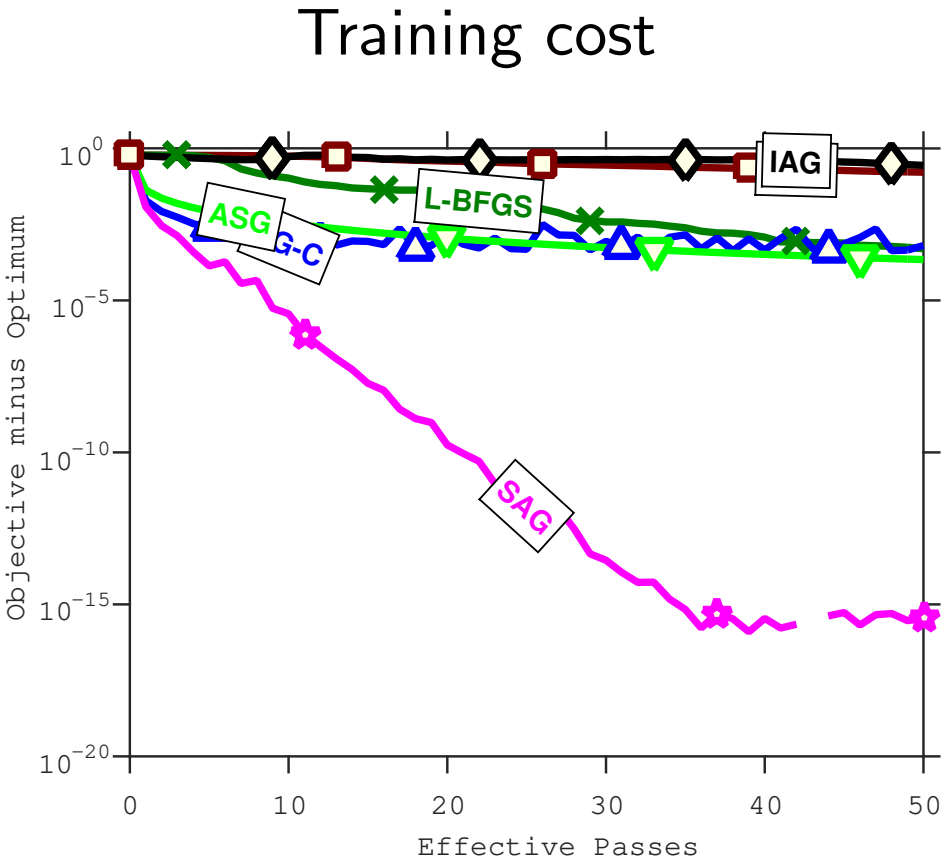


`sido` dataset
$(n = 12\ 678,\ d = 4\ 932)$



From training to testing errors

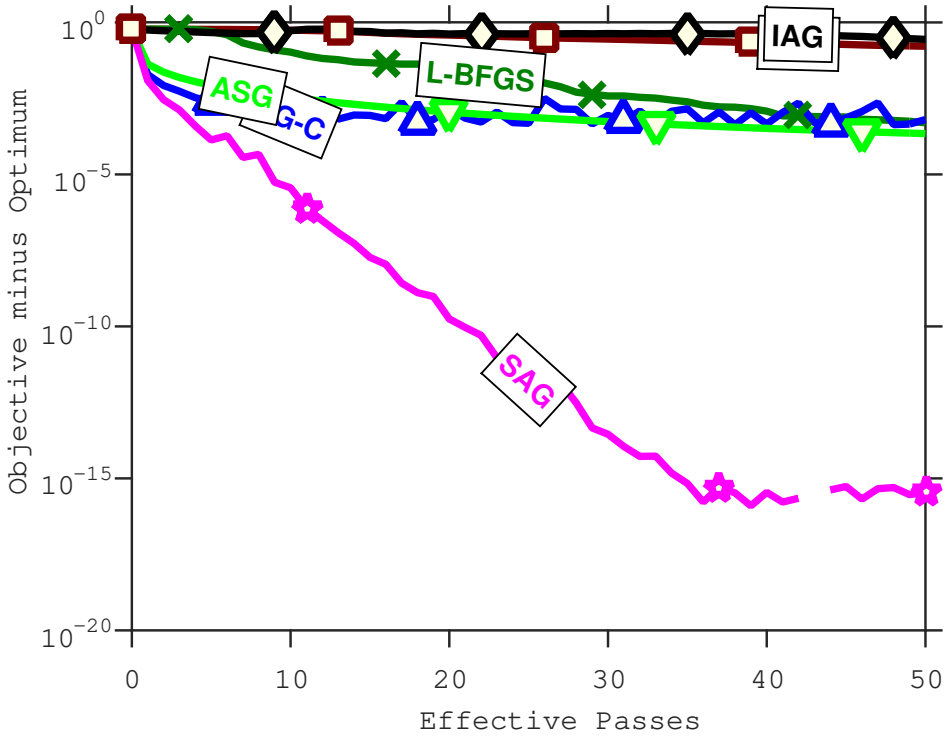
- `rcv1` dataset ($n = 697\,641$, $d = 47\,236$)
  - NB: IAG, SG-C, ASG with optimal step-sizes in hindsight



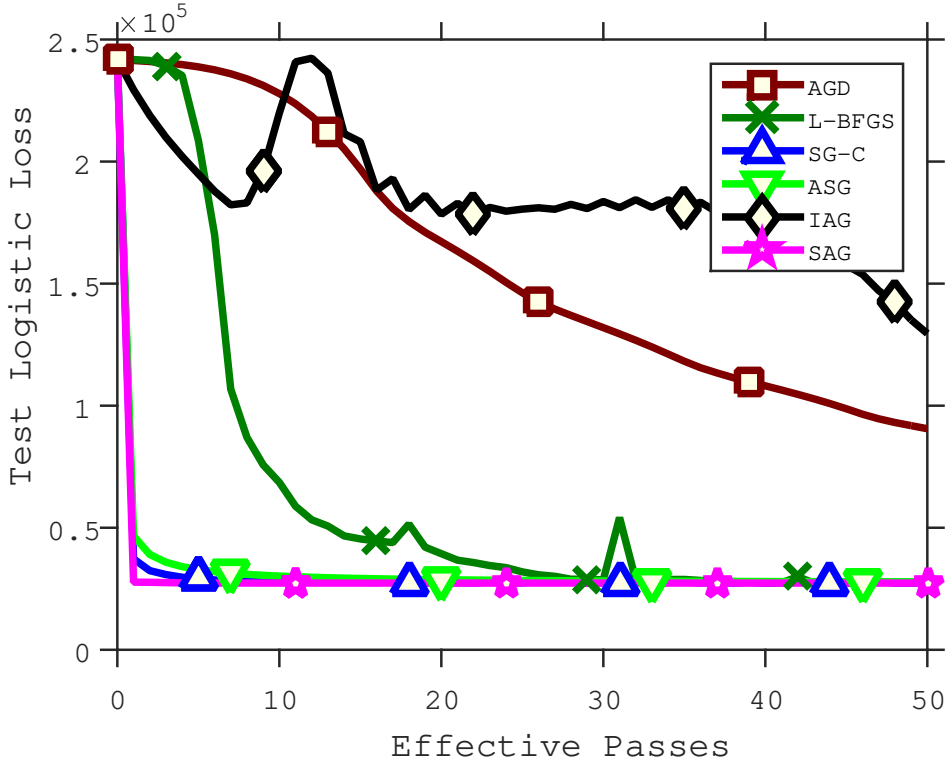
From training to testing errors

- `rcv1` dataset ($n = 697\,641$, $d = 47\,236$)
  - NB: IAG, SG-C, ASG with optimal step-sizes in hindsight

Training cost



Testing cost



Linearly convergent stochastic gradient algorithms

- **Many related algorithms**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SDCA (Shalev-Shwartz and Zhang, 2013)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - MISO (Mairal, 2015)
  - Finito (Defazio et al., 2014b)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014a)
  - ...
- **Similar rates of convergence and iterations**

Linearly convergent stochastic gradient algorithms

- **Many related algorithms**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SDCA (Shalev-Shwartz and Zhang, 2013)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - MISO (Mairal, 2015)
  - Finito (Defazio et al., 2014b)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014a)
  - ...
- **Similar rates of convergence and iterations**
- **Different interpretations and proofs / proof lengths**
  - Lazy gradient evaluations
  - Variance reduction: $\theta_t = \theta_{t-1} - \gamma \left[\nabla f_{i(t)}(\theta_{t-1}) \right.$ $\left. \right]$

Linearly convergent stochastic gradient algorithms

- **Many related algorithms**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SDCA (Shalev-Shwartz and Zhang, 2013)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - MISO (Mairal, 2015)
  - Finito (Defazio et al., 2014b)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014a)
  - ...
- **Similar rates of convergence and iterations**
- **Different interpretations and proofs / proof lengths**
  - Lazy gradient evaluations
  - Variance reduction: $\theta_t = \theta_{t-1} - \gamma \left[\nabla f_{i(t)}(\theta_{t-1}) + \frac{1}{n} \sum_{i=1}^{n} y_i^{t-1} - y_{i(t)}^{t-1} \right]$

Acceleration

- **Similar guarantees for finite sums**: SAG, SDCA, SVRG (Xiao and Zhang, 2014), SAGA, MISO (Mairal, 2015)

Gradient descent	$d\times \quad n\frac{L}{\mu} \quad \times \log\frac{1}{\varepsilon}$
Accelerated gradient descent	$d\times \quad n\sqrt{\frac{L}{\mu}} \quad \times \log\frac{1}{\varepsilon}$
SAG(A), SVRG, SDCA, MISO	$d\times \quad (n+\frac{L}{\mu}) \quad \times \log\frac{1}{\varepsilon}$
Accelerated versions	$d\times (n+\sqrt{n\frac{L}{\mu}}) \quad \times \log\frac{1}{\varepsilon}$

- **Acceleration for special algorithms** (e.g., Shalev-Shwartz and Zhang, 2014; Nitanda, 2014; Lan, 2015; Defazio, 2016)

- **Catalyst** (Lin, Mairal, and Harchaoui, 2015)
  - Widely applicable generic acceleration scheme

SGD minimizes the testing cost!

- **Goal**: minimize $f(\theta) = \mathbb{E}_{p(x,y)}\ell(y, h(x, \theta))$
  - Given n independent samples (x_i, y_i), $i = 1, \ldots, n$ from $p(x, y)$
  - Given a single pass of stochastic gradient descent
  - Bounds on the excess testing cost $\mathbb{E}f(\bar{\theta}_n) - \inf_{\theta \in \mathbb{R}^d} f(\theta)$

SGD minimizes the testing cost!

- **Goal**: minimize $f(\theta) = \mathbb{E}_{p(x,y)}\ell(y, h(x, \theta))$
  - Given n independent samples (x_i, y_i), $i = 1, \ldots, n$ from $p(x, y)$
  - Given a single pass of stochastic gradient descent
  - Bounds on the excess testing cost $\mathbb{E}f(\bar{\theta}_n) - \inf_{\theta \in \mathbb{R}^d} f(\theta)$

- **Optimal convergence rates**: $O(1/\sqrt{n})$ and $O(1/(n\mu))$
  - Optimal for non-smooth losses (Nemirovski and Yudin, 1983)
  - Attained by averaged SGD with decaying step-sizes

SGD minimizes the testing cost!

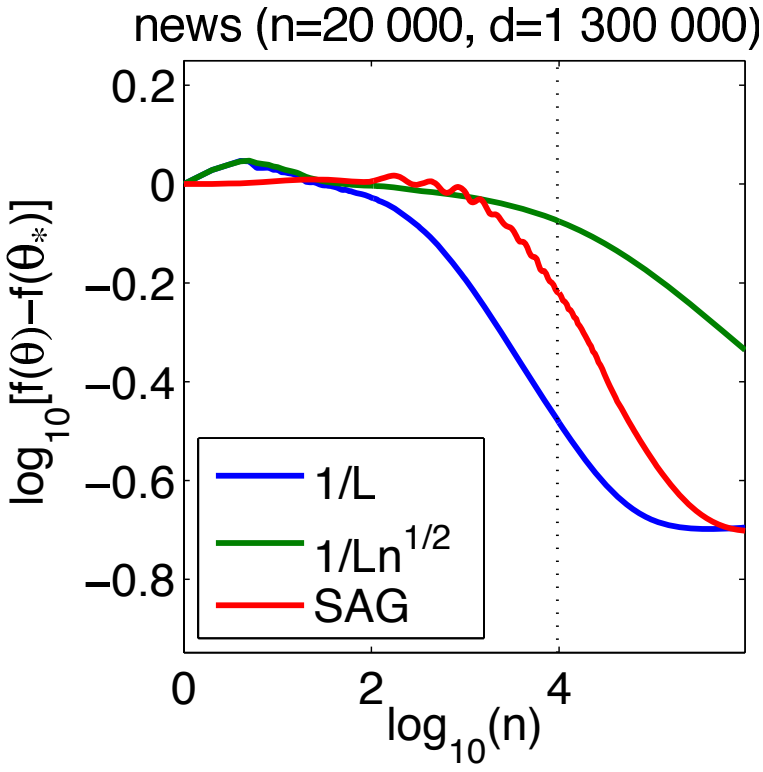
- **Goal**: minimize $f(\theta) = \mathbb{E}_{p(x,y)}\ell(y, h(x,\theta))$
  - Given n independent samples (x_i, y_i), $i = 1, \ldots, n$ from $p(x,y)$
  - Given a single pass of stochastic gradient descent
  - Bounds on the excess testing cost $\mathbb{E}f(\bar{\theta}_n) - \inf_{\theta \in \mathbb{R}^d} f(\theta)$

- **Optimal convergence rates**: $O(1/\sqrt{n})$ and $O(1/(n\mu))$
  - Optimal for non-smooth losses (Nemirovski and Yudin, 1983)
  - Attained by averaged SGD with decaying step-sizes

- **Constant-step-size SGD**
  - Linear convergence up to the noise level for strongly-convex problems (Solodov, 1998; Nedic and Bertsekas, 2000)
  - Full convergence and robustness to ill-conditioning?

Robust averaged stochastic gradient (Bach and Moulines, 2013)

- **Constant-step-size SGD is convergent for least-squares**
  - Convergence rate in $O(1/n)$ without any dependence on μ
  - Simple choice of step-size (equal to $1/L$) *(see board)*

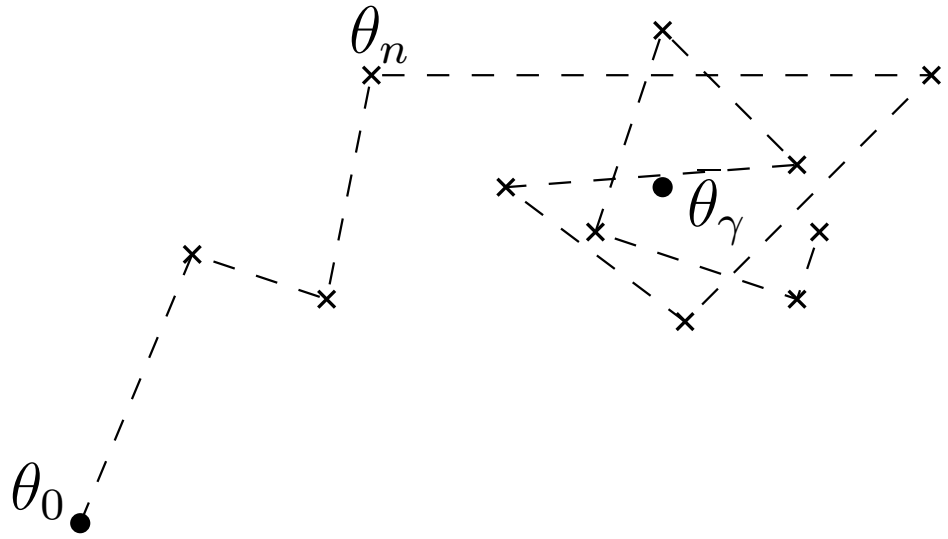


Markov chain interpretation of constant step sizes

- LMS recursion for $f_n(\theta) = \frac{1}{2}\big(y_n - \langle \Phi(x_n), \theta \rangle\big)^2$

$$\theta_n = \theta_{n-1} - \gamma\big(\langle \Phi(x_n), \theta_{n-1} \rangle - y_n\big)\Phi(x_n)$$

- The sequence $(\theta_n)_n$ is a homogeneous Markov chain
  - convergence to a stationary distribution π_γ
  - with expectation $\bar{\theta}_\gamma \stackrel{\text{def}}{=} \int \theta \pi_\gamma(\mathrm{d}\theta)$



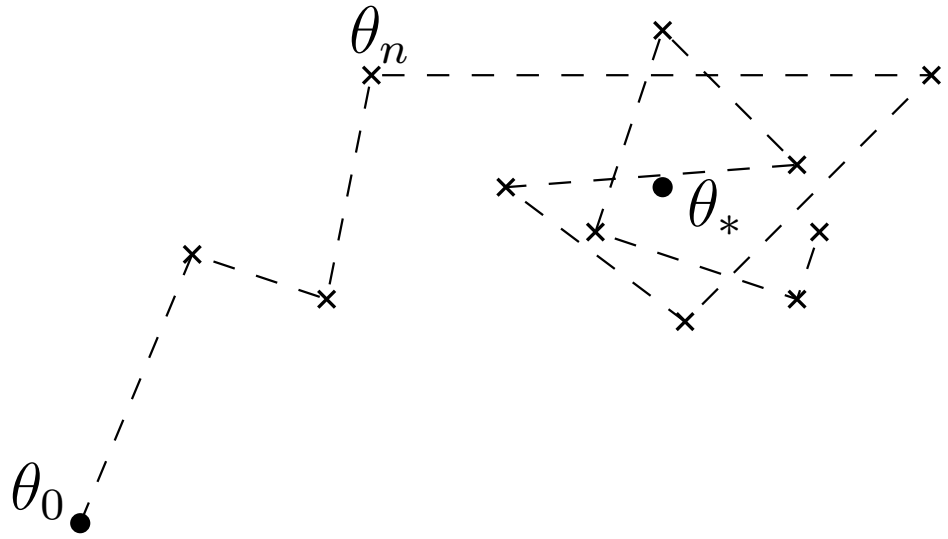
Markov chain interpretation of constant step sizes

- LMS recursion for $f_n(\theta) = \frac{1}{2}\big(y_n - \langle \Phi(x_n), \theta \rangle\big)^2$

$$\theta_n = \theta_{n-1} - \gamma\big(\langle \Phi(x_n), \theta_{n-1} \rangle - y_n\big)\Phi(x_n)$$

- The sequence $(\theta_n)_n$ is a homogeneous Markov chain
  - convergence to a stationary distribution π_γ
  - with expectation $\bar{\theta}_\gamma \stackrel{\text{def}}{=} \int \theta \pi_\gamma(\mathrm{d}\theta)$

- **For least-squares,** $\bar{\theta}_\gamma = \theta_*$



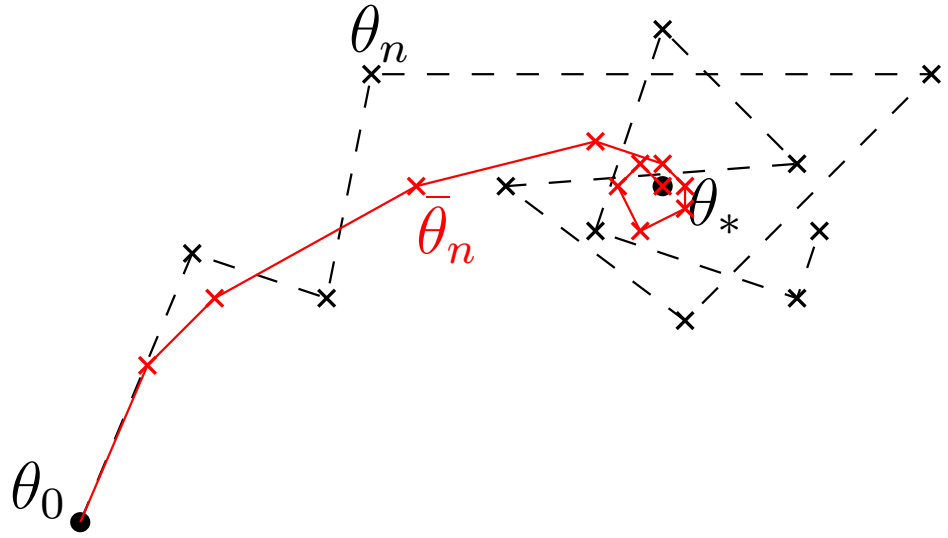
Markov chain interpretation of constant step sizes

- LMS recursion for $f_n(\theta) = \frac{1}{2}\big(y_n - \langle \Phi(x_n), \theta \rangle\big)^2$

$$\theta_n = \theta_{n-1} - \gamma\big(\langle \Phi(x_n), \theta_{n-1} \rangle - y_n\big)\Phi(x_n)$$

- The sequence $(\theta_n)_n$ is a homogeneous Markov chain
  - convergence to a stationary distribution π_γ
  - with expectation $\bar{\theta}_\gamma \stackrel{\text{def}}{=} \int \theta \pi_\gamma(\mathrm{d}\theta)$

- **For least-squares, $\bar{\theta}_\gamma = \theta_*$**



Markov chain interpretation of constant step sizes

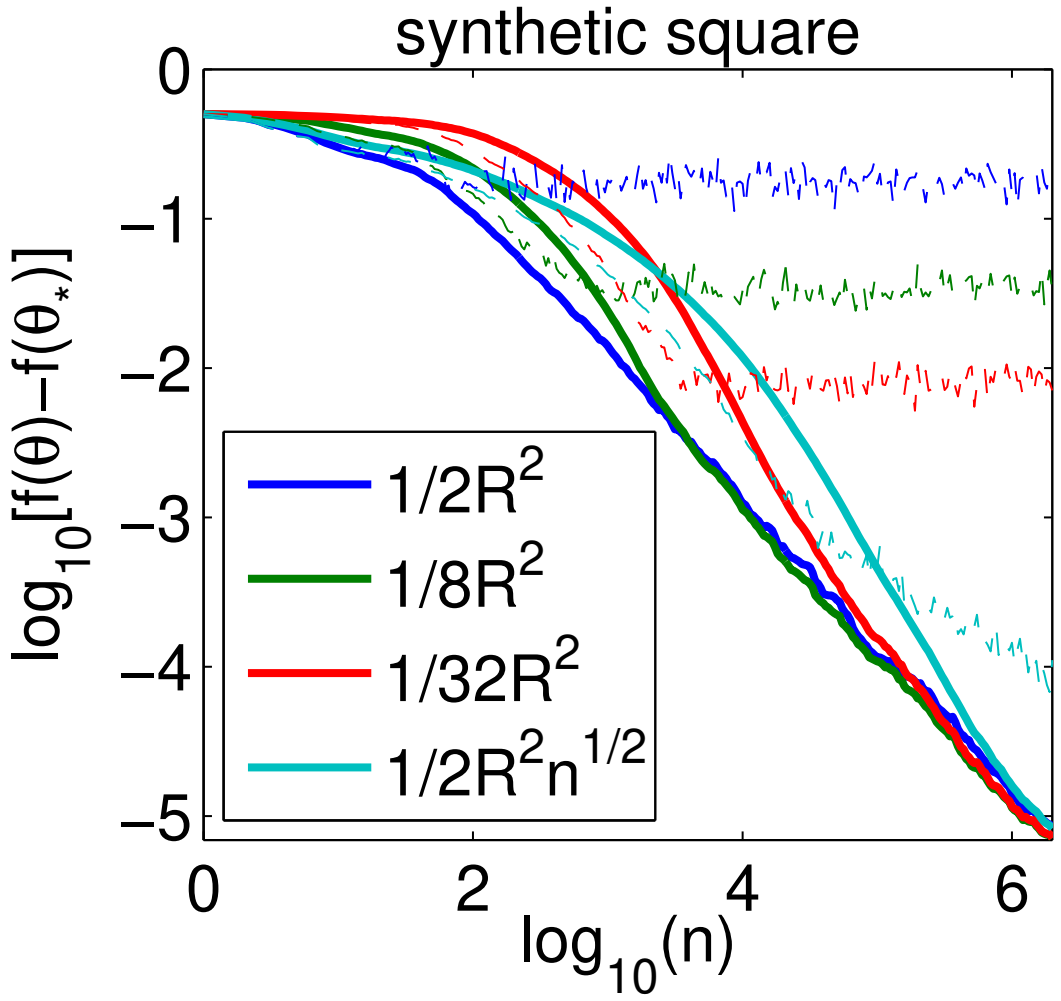
- LMS recursion for $f_n(\theta) = \frac{1}{2}\big(y_n - \langle \Phi(x_n), \theta \rangle\big)^2$

$$\theta_n = \theta_{n-1} - \gamma\big(\langle \Phi(x_n), \theta_{n-1} \rangle - y_n\big)\Phi(x_n)$$

- The sequence $(\theta_n)_n$ is a homogeneous Markov chain
  - convergence to a stationary distribution π_γ
  - with expectation $\bar{\theta}_\gamma \stackrel{\text{def}}{=} \int \theta \pi_\gamma(\mathrm{d}\theta)$
- **For least-squares, $\bar{\theta}_\gamma = \theta_*$**
  - θ_n does not converge to θ_* but oscillates around it
  - oscillations of order $\sqrt{\gamma}$
- **Ergodic theorem:**
  - Averaged iterates converge to $\bar{\theta}_\gamma = \theta_*$ at rate $O(1/n)$

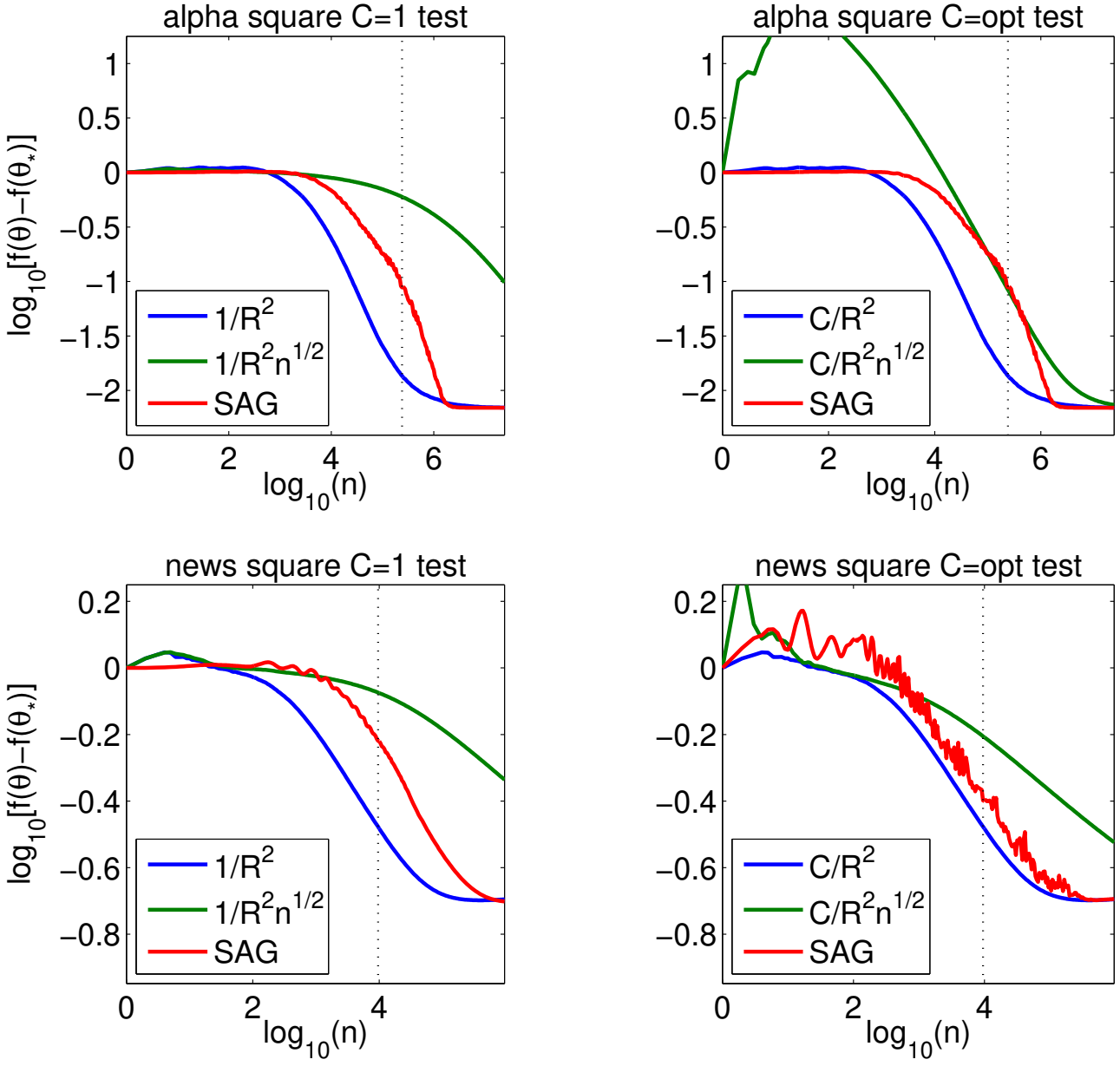
Simulations - synthetic examples

- Gaussian distributions - $d = 20$



Simulations - benchmarks

- *alpha* ($d = 500$, $n = 500\ 000$), *news* ($d = 1\ 300\ 000$, $n = 20\ 000$)



Robust averaged stochastic gradient (Bach and Moulines, 2013)

- **Constant-step-size SGD is convergent for least-squares**
  - Convergence rate in $O(1/n)$ without any dependence on μ
  - Simple choice of step-size (equal to $1/L$)

- **Constant-step-size SGD can be made convergent**
  - Online Newton correction with same complexity as SGD
  - Replace $\theta_n = \theta_{n-1} - \gamma f_n'(\theta_{n-1})$
    by $\theta_n = \theta_{n-1} - \gamma \big[f_n'(\bar{\theta}_{n-1}) + f''(\bar{\theta}_{n-1})(\theta_{n-1} - \bar{\theta}_{n-1}) \big]$
  - Simple choice of step-size and convergence rate in $O(1/n)$

Robust averaged stochastic gradient (Bach and Moulines, 2013)

- **Constant-step-size SGD is convergent for least-squares**
  - Convergence rate in $O(1/n)$ without any dependence on μ
  - Simple choice of step-size (equal to $1/L$)
- **Constant-step-size SGD can be made convergent**
  - Online Newton correction with same complexity as SGD
  - Replace $\theta_n = \theta_{n-1} - \gamma f_n'(\theta_{n-1})$
    by $\theta_n = \theta_{n-1} - \gamma \big[f_n'(\bar{\theta}_{n-1}) + f''(\bar{\theta}_{n-1})(\theta_{n-1} - \bar{\theta}_{n-1}) \big]$
  - Simple choice of step-size and convergence rate in $O(1/n)$
- **Multiple passes still work better in practice**
  - See Pillaud-Vivien, Rudi, and Bach (2018)

Perspectives

- **Linearly-convergent stochastic gradient methods**
  - Provable and precise rates
  - Improves on two known lower-bounds (by using structure)
  - Several extensions / interpretations / accelerations

Perspectives

- **Linearly-convergent stochastic gradient methods**
  - Provable and precise rates
  - Improves on two known lower-bounds (by using structure)
  - Several extensions / interpretations / accelerations

- **Extensions and future work**
  - Lower bounds for finite sums (Lan, 2015)
  - Sampling without replacement (Gurbuzbalaban et al., 2015)

Perspectives

- **Linearly-convergent stochastic gradient methods**
  - Provable and precise rates
  - Improves on two known lower-bounds (by using structure)
  - Several extensions / interpretations / accelerations

- **Extensions and future work**
  - Lower bounds for finite sums (Lan, 2015)
  - Sampling without replacement (Gurbuzbalaban et al., 2015)
  - Bounds on testing errors for incremental methods

Perspectives

- **Linearly-convergent stochastic gradient methods**
  - Provable and precise rates
  - Improves on two known lower-bounds (by using structure)
  - Several extensions / interpretations / accelerations

- **Extensions and future work**
  - Lower bounds for finite sums (Lan, 2015)
  - Sampling without replacement (Gurbuzbalaban et al., 2015)
  - Bounds on testing errors for incremental methods
  - Parallelization (Leblond, Pedregosa, and Lacoste-Julien, 2016; Hendrikx, Bach, and Massoulié, 2019)
  - Non-convex problems (Reddi et al., 2016)
  - Other forms of acceleration (Scieur, d'Aspremont, and Bach, 2016)

Perspectives

- **Linearly-convergent stochastic gradient methods**
  - Provable and precise rates
  - Improves on two known lower-bounds (by using structure)
  - Several extensions / interpretations / accelerations

- **Extensions and future work**
  - Lower bounds for finite sums (Lan, 2015)
  - Sampling without replacement (Gurbuzbalaban et al., 2015)
  - Bounds on testing errors for incremental methods
  - Parallelization (Leblond, Pedregosa, and Lacoste-Julien, 2016; Hendrikx, Bach, and Massoulié, 2019)
  - Non-convex problems (Reddi et al., 2016)
  - Other forms of acceleration (Scieur, d'Aspremont, and Bach, 2016)
  - Pre-conditioning

Outline

1. **Introduction/motivation: Supervised machine learning**
   - Machine learning $\approx$ optimization of finite sums
   - Batch optimization methods

2. **Fast stochastic gradient methods for convex problems**
   - Variance reduction: for *training* error
   - Constant step-sizes: for *testing* error

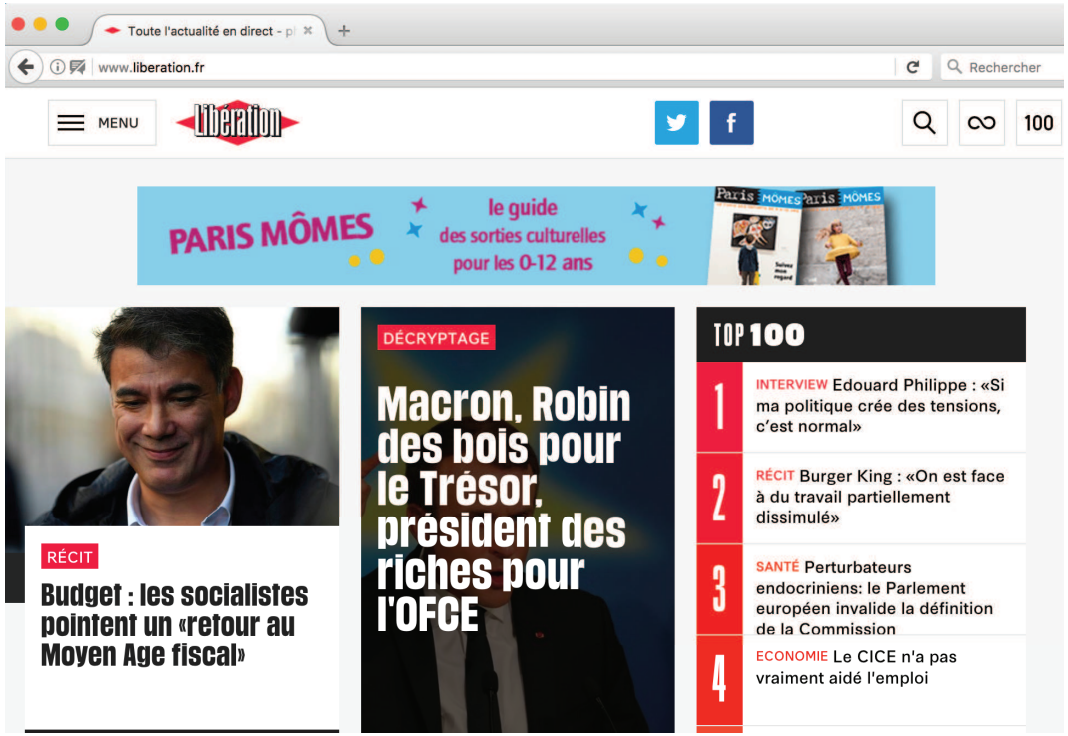
2. **Beyond convex problems**
   - Generic algorithms with generic "guarantees"
   - Global convergence for over-parameterized neural networks

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$



- Linear predictions
  - $h(x, \theta) = \theta^\top \Phi(x)$
- E.g., **advertising**: $n > 10^9$
  - $\Phi(x) \in \{0, 1\}^d$, $d > 10^9$
  - Navigation history + ad

Parametric supervised machine learning

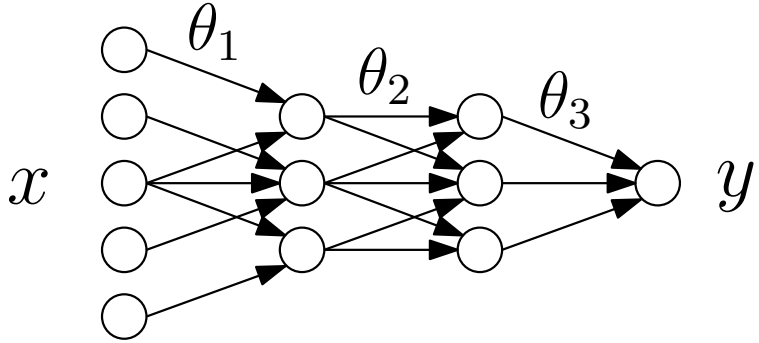
- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$

- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

x_1 x_2 x_3 x_4 x_5 x_6

$y_1 = 1$ $y_2 = 1$ $y_3 = 1$ $y_4 = -1$ $y_5 = -1$ $y_6 = -1$

  - Neural networks $(n, d > 10^6)$: $h(x, \theta) = \theta_r^\top \sigma(\theta_{r-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x))$



Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**:

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta)$$

data fitting term + regularizer

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**:

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta)$$

data fitting term + regularizer

- **Actual goal**: minimize test error $\mathbb{E}_{p(x,y)} \ell(y, h(x, \theta))$

Convex optimization problems

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta)$$

- **Conditions**: Convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$
- **Consequences**
  - Efficient algorithms (typically gradient-based)
  - Quantitative runtime and prediction performance guarantees

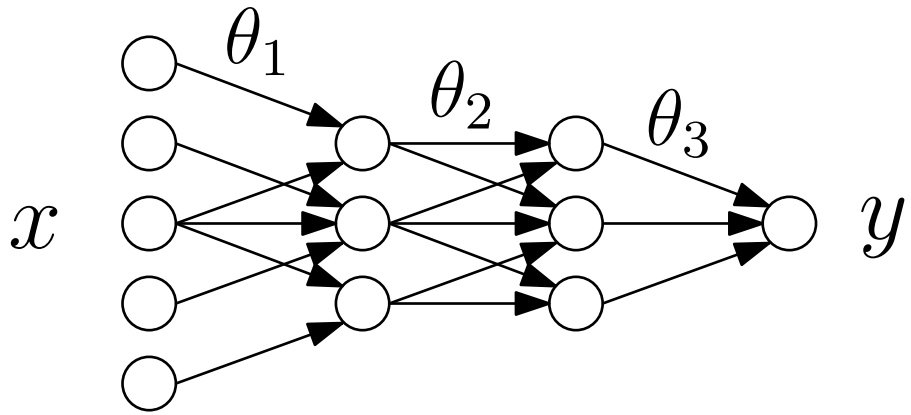
Convex optimization problems

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta)$$

- **Conditions**: Convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$
- **Consequences**
  - Efficient algorithms (typically gradient-based)
  - Quantitative runtime and prediction performance guarantees
- **Golden years of convexity in machine learning** (1995 to 2020)
  - Support vector machines and kernel methods
  - Sparsity / low-rank models with first-order methods
  - Optimal transport
  - Stochastic methods for large-scale learning and online learning
  - etc.

Theoretical analysis of deep learning

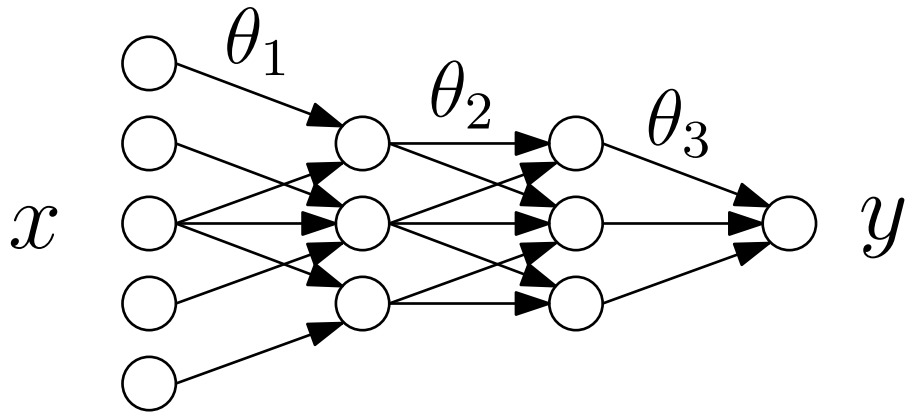
- **Multi-layer neural network** $h(x, \theta) = \theta_r^\top \sigma(\theta_{r-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x))$



  - NB: already a simplification

Theoretical analysis of deep learning

- **Multi-layer neural network** $h(x, \theta) = \theta_r^\top \sigma(\theta_{r-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x))$



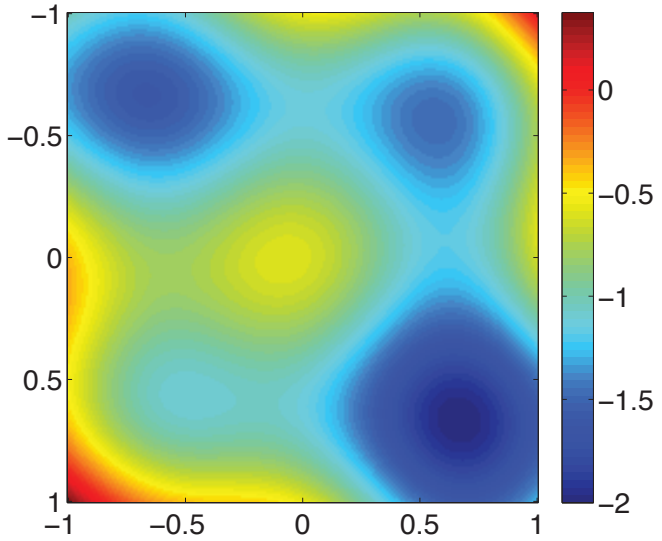
  - NB: already a simplification

- **Main difficulties**

  1. Non-convex optimization problems
  2. Generalization guarantees in the overparameterized regime

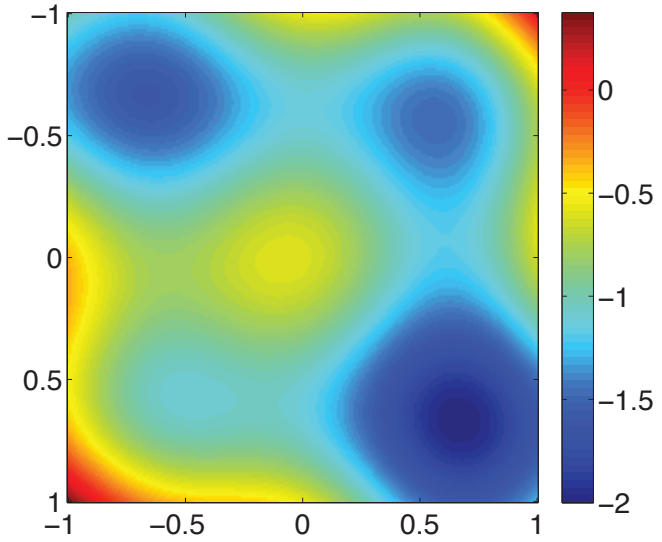
Optimization for multi-layer neural networks

- **What can go wrong with non-convex optimization problems?**
  - Local minima
  - Stationary points
  - Plateaux
  - Bad initialization
  - etc...



Optimization for multi-layer neural networks

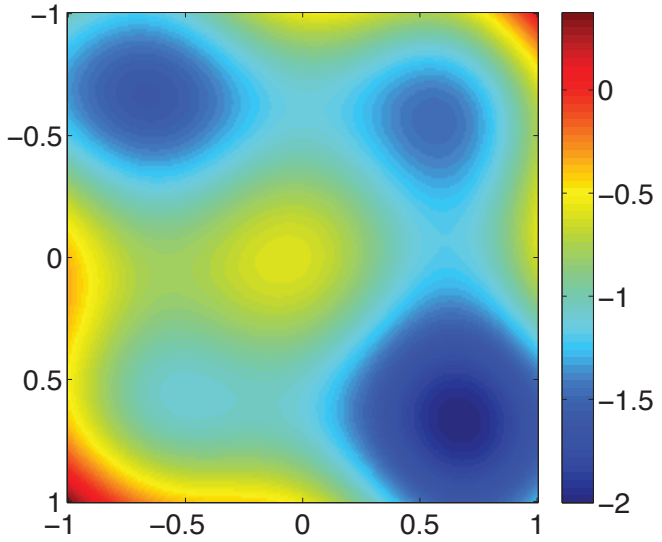
- **What can go wrong with non-convex optimization problems?**
  - Local minima
  - Stationary points
  - Plateaux
  - Bad initialization
  - etc...



- **Generic local theoretical guarantees**
  - Convergence to stationary points or local minima
  - See, e.g., Lee et al. (2016); Jin et al. (2017)

Optimization for multi-layer neural networks

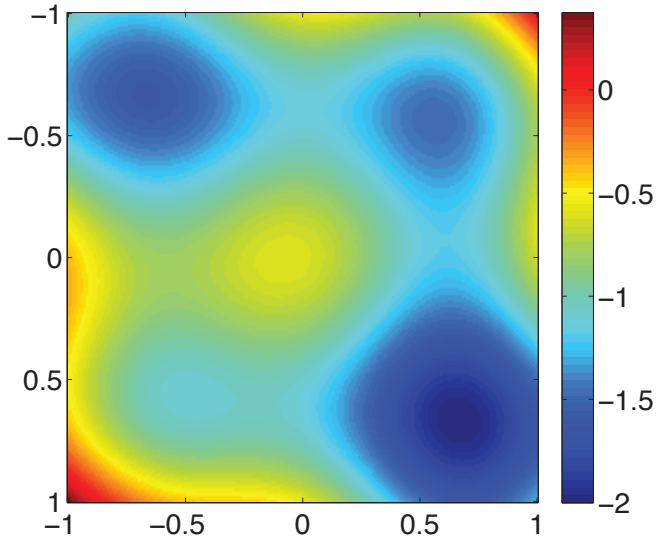
- **What can go wrong with non-convex optimization problems?**
  - Local minima
  - Stationary points
  - Plateaux
  - Bad initialization
  - etc...



- **General global performance guarantees impossible to obtain**

Optimization for multi-layer neural networks

- **What can go wrong with non-convex optimization problems?**
  - Local minima
  - Stationary points
  - Plateaux
  - Bad initialization
  - etc...



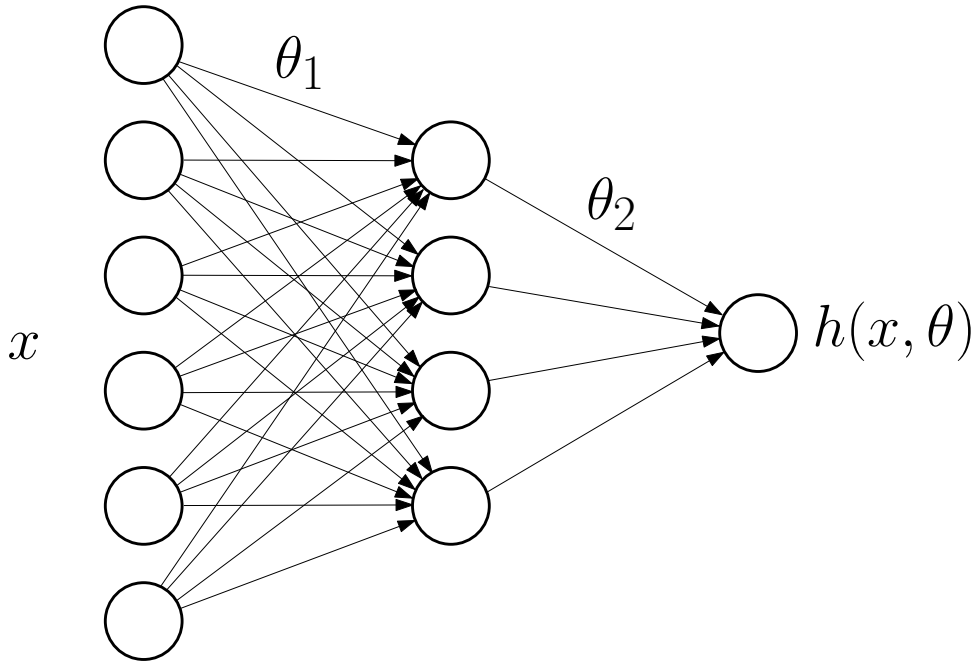
- **General global performance guarantees impossible to obtain**

- **Special case of (deep) neural networks**
  - Most local minima are equivalent (Choromanska et al., 2015)
  - No spurrious local minima (Soltanolkotabi et al., 2018)

Gradient descent for a single hidden layer

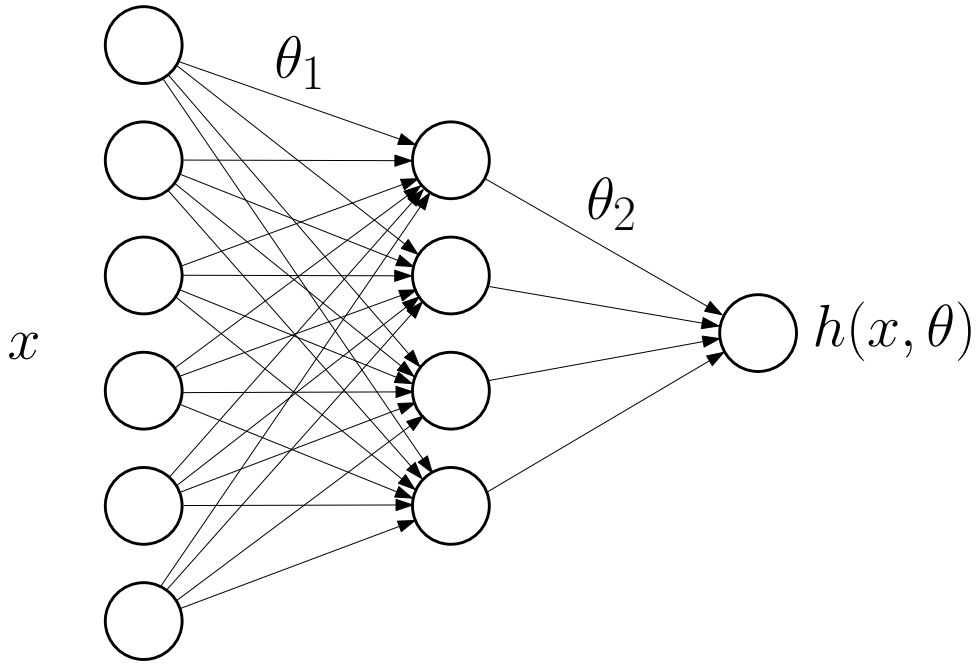
- **Predictor**: $h(x) = \frac{1}{m}\theta_2^\top \sigma(\theta_1^\top x) = \frac{1}{m}\sum_{j=1}^m \theta_2(j) \cdot \sigma\left[\theta_1(\cdot, j)^\top x\right]$

- **Goal**: minimize $R(h) = \mathbb{E}_{p(x,y)}\ell(y, h(x))$, with R convex



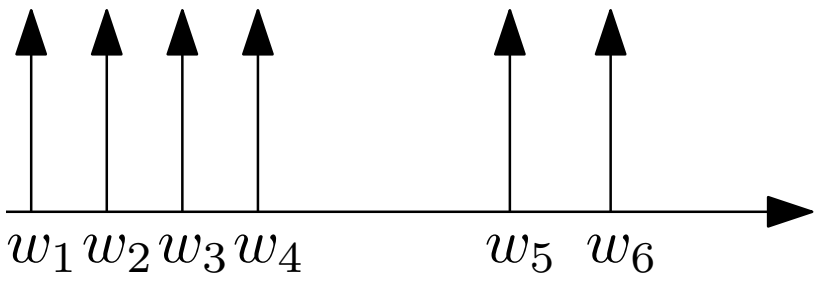
Gradient descent for a single hidden layer

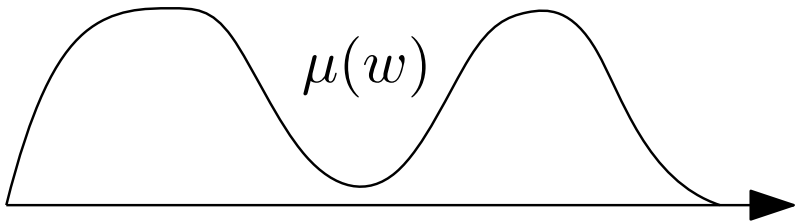
- **Predictor**: $h(x) = \frac{1}{m}\theta_2^\top \sigma(\theta_1^\top x) = \frac{1}{m}\sum_{j=1}^m \theta_2(j) \cdot \sigma\left[\theta_1(\cdot, j)^\top x\right]$
  - Family: $h = \dfrac{1}{m}\displaystyle\sum_{j=1}^{m} \Psi(w_j)$ with $\Psi(w_j)(x) = \theta_2(j) \cdot \sigma\left[\theta_1(\cdot, j)^\top x\right]$
- **Goal**: minimize $R(h) = \mathbb{E}_{p(x,y)}\ell(y, h(x))$, with R convex



Gradient descent for a single hidden layer

- **Predictor**: $h(x) = \frac{1}{m}\theta_2^\top \sigma(\theta_1^\top x) = \frac{1}{m}\sum_{j=1}^m \theta_2(j) \cdot \sigma\left[\theta_1(\cdot, j)^\top x\right]$
  - Family: $h = \dfrac{1}{m}\displaystyle\sum_{j=1}^m \Psi(w_j)$ with $\Psi(w_j)(x) = \theta_2(j) \cdot \sigma\left[\theta_1(\cdot, j)^\top x\right]$
- **Goal**: minimize $R(h) = \mathbb{E}_{p(x,y)}\ell(y, h(x))$, with R convex
- **Main insight**
  - $h = \dfrac{1}{m}\displaystyle\sum_{j=1}^m \Psi(w_j) = \int_{\mathcal{W}} \Psi(w) d\mu(w)$ with $d\mu(w) = \dfrac{1}{m}\displaystyle\sum_{j=1}^m \delta_{w_j}$





Gradient descent for a single hidden layer

- **Predictor**: $h(x) = \frac{1}{m}\theta_2^\top \sigma(\theta_1^\top x) = \frac{1}{m}\sum_{j=1}^m \theta_2(j) \cdot \sigma\big[\theta_1(\cdot, j)^\top x\big]$
  - Family: $h = \dfrac{1}{m}\displaystyle\sum_{j=1}^m \Psi(w_j)$ with $\Psi(w_j)(x) = \theta_2(j) \cdot \sigma\big[\theta_1(\cdot, j)^\top x\big]$
- **Goal**: minimize $R(h) = \mathbb{E}_{p(x,y)}\ell(y, h(x))$, with R convex
- **Main insight**
  - $h = \dfrac{1}{m}\displaystyle\sum_{j=1}^m \Psi(w_j) = \int_{\mathcal{W}} \Psi(w) d\mu(w)$ with $d\mu(w) = \dfrac{1}{m}\displaystyle\sum_{j=1}^m \delta_{w_j}$
  - Overparameterized models with m large $\approx$ measure μ with densities
  - Barron (1993); Kurkova and Sanguineti (2001); Bengio et al. (2006); Rosset et al. (2007); Bach (2017)

Optimization on measures

- **Minimize with respect to measure** μ: $R\Big(\int_{\mathcal{W}} \Psi(w) d\mu(w)\Big)$
  - Convex optimization problem on measures
  - Frank-Wolfe techniques for incremental learning
  - Non-tractable (Bach, 2017), not what is used in practice

Optimization on measures

- **Minimize with respect to measure μ:** $R\Big(\int_{\mathcal{W}} \Psi(w) d\mu(w)\Big)$
  - Convex optimization problem on measures
  - Frank-Wolfe techniques for incremental learning
  - Non-tractable (Bach, 2017), not what is used in practice

- **Represent μ by a finite set of "particles"** $\mu = \frac{1}{m} \sum_{j=1}^{m} \delta_{w_j}$
  - Backpropagation = gradient descent on $(w_1, \ldots, w_m)$

- **Three questions**:
  - Algorithm limit when number of particles m gets large
  - Global convergence to a global minimizer
  - Prediction performance

Many particle limit and global convergence (Chizat and Bach, 2018)

- **General framework**: minimize $F(\mu) = R\Big(\int_{\mathcal{W}} \Psi(w) d\mu(w)\Big)$
  - Algorithm: minimizing $F_m(w_1, \ldots, w_m) = R\Big(\frac{1}{m}\sum_{j=1}^{m} \Psi(w_j)\Big)$

Many particle limit and global convergence (Chizat and Bach, 2018)

- **General framework**: minimize $F(\mu) = R\Big(\int_{\mathcal{W}} \Psi(w) d\mu(w)\Big)$
  - Algorithm: minimizing $F_m(w_1, \ldots, w_m) = R\Big(\frac{1}{m}\sum_{j=1}^{m} \Psi(w_j)\Big)$
  - Gradient flow $\dot{W} = -m\nabla F_m(W)$, with $W = (w_1, \ldots, w_m)$
  - Idealization of (stochastic) gradient descent

Many particle limit and global convergence (Chizat and Bach, 2018)

- **General framework**: minimize $F(\mu) = R\Big(\int_{\mathcal{W}} \Psi(w) d\mu(w)\Big)$
  - Algorithm: minimizing $F_m(w_1, \ldots, w_m) = R\Big(\frac{1}{m}\sum_{j=1}^{m} \Psi(w_j)\Big)$
  - Gradient flow $\dot{W} = -m\nabla F_m(W)$, with $W = (w_1, \ldots, w_m)$
  - Idealization of (stochastic) gradient descent
    1. Single pass SGD on the unobserved expected risk
    2. Multiple pass SGD or full GD on the empirical risk

Many particle limit and global convergence (Chizat and Bach, 2018)

- **General framework**: minimize $F(\mu) = R\Big(\int_{\mathcal{W}} \Psi(w) d\mu(w)\Big)$
  - Algorithm: minimizing $F_m(w_1, \ldots, w_m) = R\Big(\frac{1}{m}\sum_{j=1}^{m} \Psi(w_j)\Big)$
  - Gradient flow $\dot{W} = -m\nabla F_m(W)$, with $W = (w_1, \ldots, w_m)$
  - Idealization of (stochastic) gradient descent
- **Limit when m tends to infinity**
  - Wasserstein gradient flow (Nitanda and Suzuki, 2017; Chizat and Bach, 2018; Mei, Montanari, and Nguyen, 2018; Sirignano and Spiliopoulos, 2018; Rotskoff and Vanden-Eijnden, 2018)
- NB: for more details on gradient flows, see Ambrosio et al. (2008)

Many particle limit and global convergence (Chizat and Bach, 2018)

- **(informal) theorem**: when the number of particles tends to infinity, the gradient flow converges to the global optimum

Many particle limit and global convergence (Chizat and Bach, 2018)

- **(informal) theorem**: when the number of particles tends to infinity, the gradient flow converges to the global optimum
  - See precise definitions and statement in paper
  - Two key ingredients: homogeneity and initialization

Many particle limit and global convergence (Chizat and Bach, 2018)

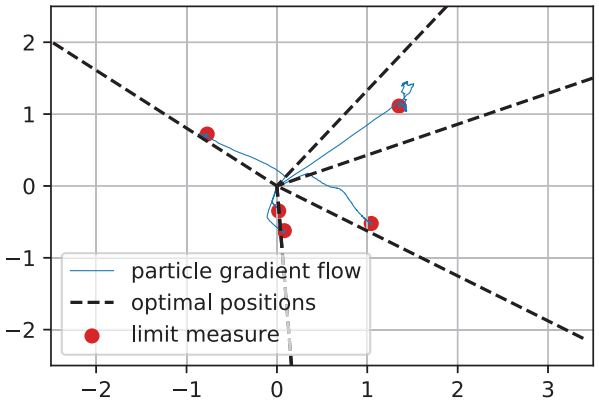
- **(informal) theorem**: when the number of particles tends to infinity, the gradient flow converges to the global optimum
  - See precise definitions and statement in paper
  - Two key ingredients: homogeneity and initialization
- **Homogeneity** (see, e.g., Haeffele and Vidal, 2017; Bach et al., 2008)
  - Full or partial, e.g., $\Psi(w_j)(x) = m\theta_2(j) \cdot \sigma\left[\theta_1(\cdot, j)^\top x\right]$
  - Applies to rectified linear units (but also to sigmoid activations)
- **Sufficiently spread initial measure**
  - Needs to cover the entire sphere of directions

Many particle limit and global convergence (Chizat and Bach, 2018)

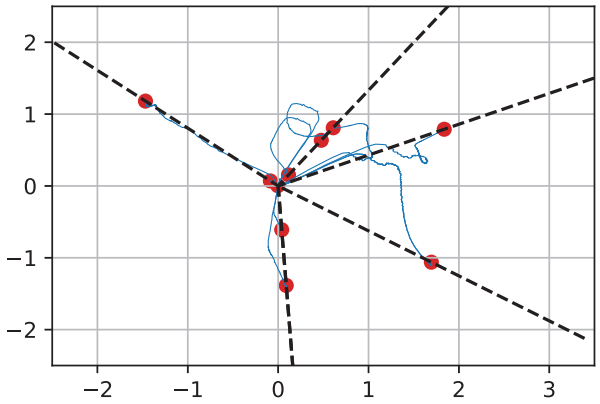
- **(informal) theorem**: when the number of particles tends to infinity, the gradient flow converges to the global optimum
  - See precise definitions and statement in paper
  - Two key ingredients: homogeneity and initialization
- **Homogeneity** (see, e.g., Haeffele and Vidal, 2017; Bach et al., 2008)
  - Full or partial, e.g., $\Psi(w_j)(x) = m\theta_2(j) \cdot \sigma\left[\theta_1(\cdot, j)^\top x\right]$
  - Applies to rectified linear units (but also to sigmoid activations)
- **Sufficiently spread initial measure**
  - Needs to cover the entire sphere of directions
- **Only qualititative!**

Simple simulations with neural networks

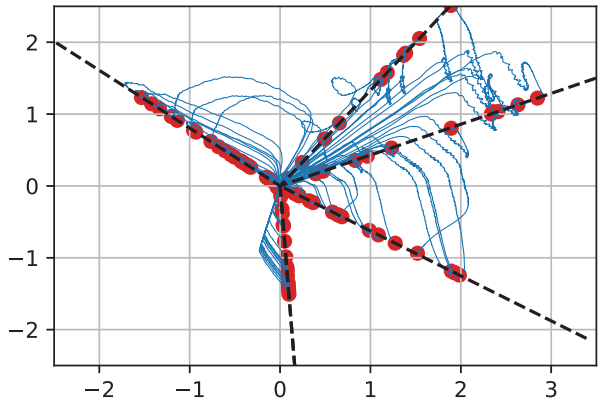
- ReLU units with $d = 2$ (optimal predictor has 5 neurons)



5 neurons



10 neurons



100 neurons

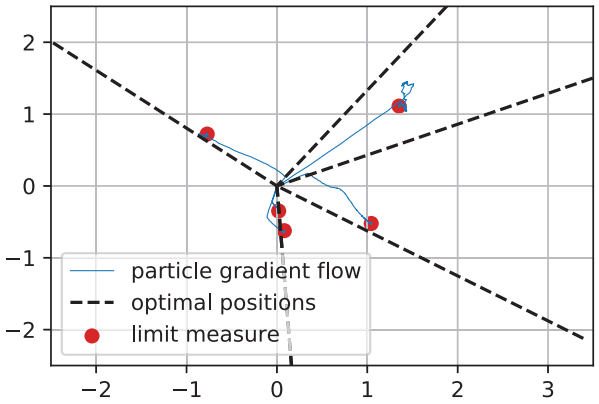
$$h(x) = \frac{1}{m}\sum_{j=1}^{m} \Psi(w_j)(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$$

(plotting $|\theta_2(j)|\theta_1(\cdot, j)$ for each hidden neuron j)

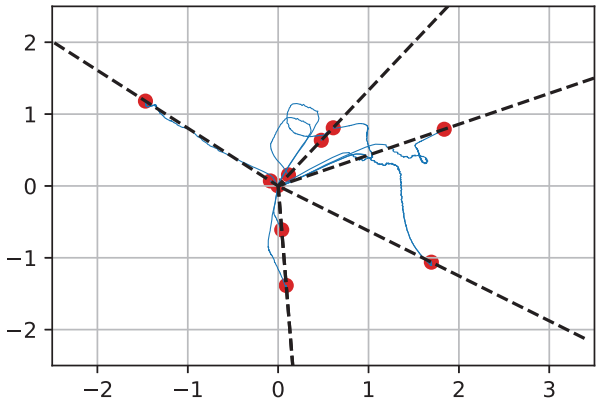
NB : also applies to spike deconvolution

Simple simulations with neural networks

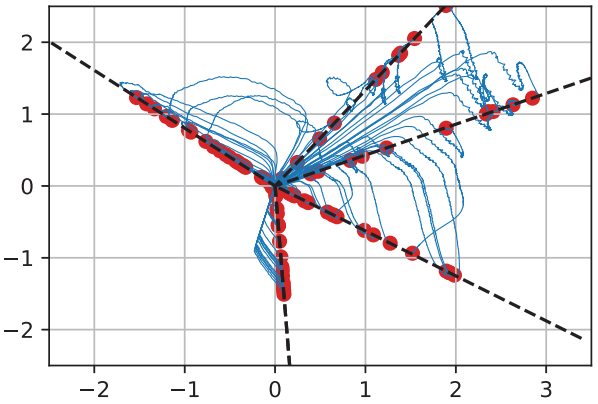
- ReLU units with $d = 2$ (optimal predictor has 5 neurons)



5 neurons



10 neurons



100 neurons

NB : also applies to spike deconvolution

From optimization to statistics

- **Summary**: with $h(x) = \frac{1}{m}\sum_{j=1}^{m} \Psi(w_j)(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - If m tends to infinity, the gradient flow converges to a global minimizer of the risk $R(h) = \mathbb{E}_{p(x,y)}\ell(y, h(x))$
  - Requires well-spread initialization, no quantitative results

From optimization to statistics

- **Summary**: with $h(x) = \frac{1}{m}\sum_{j=1}^{m} \Psi(w_j)(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - If m tends to infinity, the gradient flow converges to a global minimizer of the risk $R(h) = \mathbb{E}_{p(x,y)}\ell(y, h(x))$
  - Requires well-spread initialization, no quantitative results

- **Single-pass SGD** with R the (unobserved) expected risk
  - Converges to an optimal predictor on the testing distribution
  - Tends to underfit

From optimization to statistics

- **Summary**: with $h(x) = \frac{1}{m}\sum_{j=1}^{m} \Psi(w_j)(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - If m tends to infinity, the gradient flow converges to a global minimizer of the risk $R(h) = \mathbb{E}_{p(x,y)}\ell(y, h(x))$
  - Requires well-spread initialization, no quantitative results

- **Single-pass SGD** with R the (unobserved) expected risk
  - Converges to an optimal predictor on the testing distribution
  - Tends to underfit

- **Multiple-pass SGD or full GD** with R the empirical risk
  - Converges to an optimal predictor on the training distribution
  - Should overfit?

Interpolation regime

- Minimizing $R(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i))$ for $h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - When $m(d+1) > n$, typically there exist many h such that

$$\forall i \in \{1, \ldots, n\}, \quad h(x_i) = y_i \qquad (\text{or } \ell(y_i, h(x_i)) = 0)$$

  - See Belkin et al. (2018); Ma et al. (2018); Vaswani et al. (2019)

Interpolation regime

- Minimizing $R(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i))$ for $h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - When $m(d+1) > n$, typically there exist many h such that

$$\forall i \in \{1, \dots, n\}, \quad h(x_i) = y_i \qquad (\text{or } \ell(y_i, h(x_i)) = 0)$$

  - See Belkin et al. (2018); Ma et al. (2018); Vaswani et al. (2019)

- **Which h is the gradient flow converging to?**
  - Implicit bias of (stochastic) gradient descent

Interpolation regime

- Minimizing $R(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i))$ for $h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - When $m(d+1) > n$, typically there exist many h such that

$$\forall i \in \{1, \dots, n\}, \quad h(x_i) = y_i \qquad (\text{or } \ell(y_i, h(x_i)) = 0)$$

  - See Belkin et al. (2018); Ma et al. (2018); Vaswani et al. (2019)

- **Which h is the gradient flow converging to?**
  - Implicit bias of (stochastic) gradient descent
  - Typically minimum Euclidean norm solution (Gunasekar et al., 2017; Soudry et al., 2018; Gunasekar et al., 2018)

Interpolation regime

- Minimizing $R(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i))$ for $h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - When $m(d+1) > n$, typically there exist many h such that

$$\forall i \in \{1, \dots, n\}, \quad h(x_i) = y_i \qquad (\text{or } \ell(y_i, h(x_i)) = 0)$$

  - See Belkin et al. (2018); Ma et al. (2018); Vaswani et al. (2019)

- **Which h is the gradient flow converging to?**
  - Implicit bias of (stochastic) gradient descent
  - Typically minimum Euclidean norm solution (Gunasekar et al., 2017; Soudry et al., 2018; Gunasekar et al., 2018)
  - Surprisingly difficult for the square loss
  - Surprisingly easy for the logistic loss

Maximum margin and logistic regression

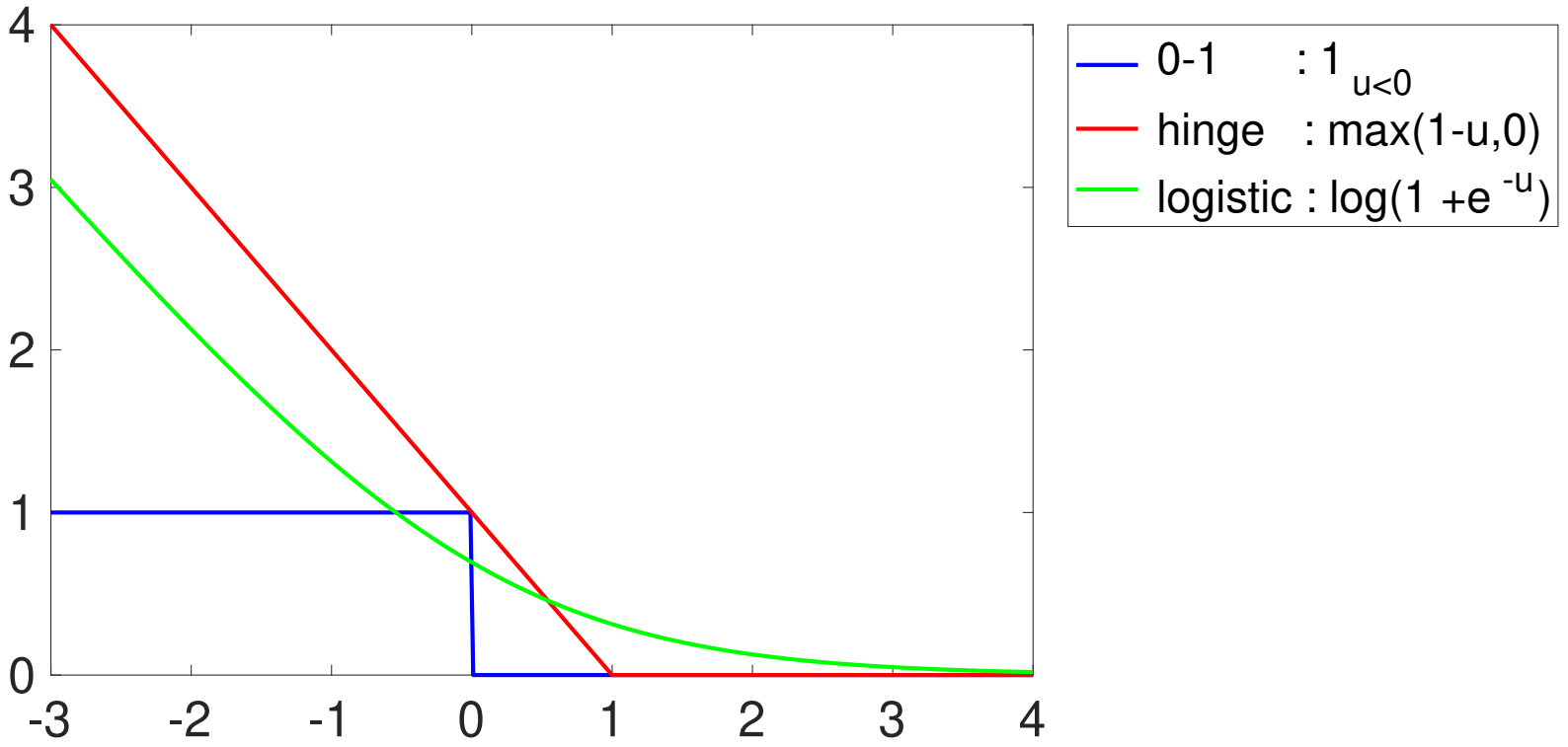
- **Logistic regression:** $\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \theta^\top x_i))$
  - Separable data: $\exists \theta \in \mathbb{R}^d, \quad \forall i \in \{1, \dots, n\}, \; y_i \theta^\top x_i > 0$

Maximum margin and logistic regression

- **Logistic regression:** $\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \theta^\top x_i))$
  - Separable data: $\exists \theta \in \mathbb{R}^d, \quad \forall i \in \{1, \ldots, n\}, \; y_i \theta^\top x_i > 1$

Maximum margin and logistic regression

- **Logistic regression:** $\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \theta^\top x_i))$
  - Separable data: $\exists \theta \in \mathbb{R}^d, \quad \forall i \in \{1, \dots, n\}, \; y_i \theta^\top x_i > 1$
  - $0 =$ infimum of the risk, attained for infinitely large $\|\theta\|_2$



(with $u = y_i \theta^\top x_i$)

Maximum margin and logistic regression

- **Logistic regression:** $\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \theta^\top x_i))$
  - Separable data: $\exists \theta \in \mathbb{R}^d, \quad \forall i \in \{1, \dots, n\}, \; y_i \theta^\top x_i > 1$
  - $0 =$ infimum of the risk, attained for infinitely large $\|\theta\|_2$

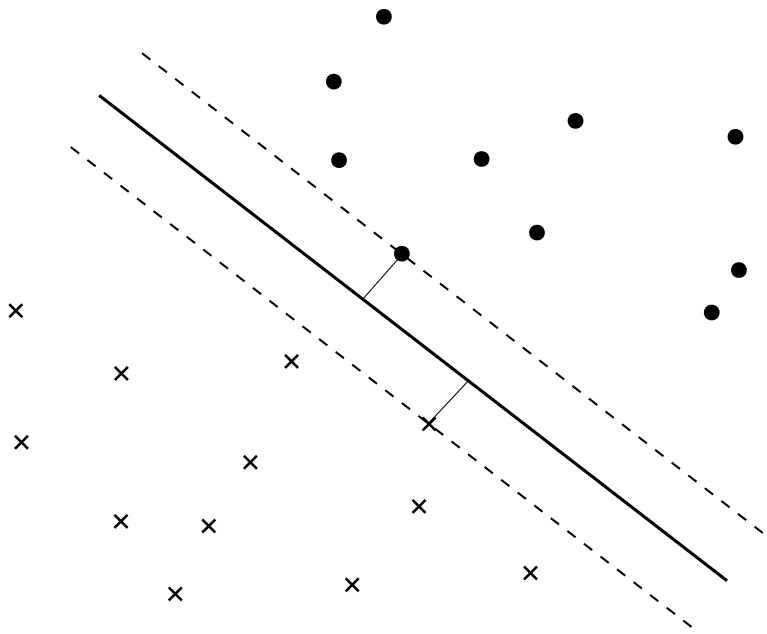
- **Implicit bias of gradient descent** (Soudry et al., 2018)
  - GD diverges but $\frac{1}{\|\theta_t\|_2} \theta_t$ converges to maximum margin separator

$$\max_{\|\eta\|_2 = 1} \quad \min_{i \in \{1, \dots, n\}} y_i \eta^\top x_i$$

  - often written as

$$\min \|\theta\|_2^2 \text{ such that } \forall i, y_i \theta^\top x_i > 1$$

  - Separable support vector machine (Vapnik and Chervonenkis, 1964)



Logistic regression for two-layer neural networks

$$h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$$

- **Overparameterized regime** $m \to +\infty$
  - Will converge to well-defined "maximum margin" separator

Logistic regression for two-layer neural networks

$$h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$$

- **Overparameterized regime** $m \to +\infty$
  - Will converge to well-defined "maximum margin" separator
- **Two different regimes** (Chizat and Bach, 2020)
  1. Optimizing over output layer only θ_2: random feature kernel
  2. Optimizing over all layers θ_1, θ_2: feature learning

Random feature kernel regime - I

- **Prediction function** $h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - Input weights $\theta_1(\cdot, j)$, $j = 1, \ldots, m$, random and fixed
  - Optimize over output weights $\theta_2 \in \mathbb{R}^m$
  - Corresponds to linear predictor with $\Phi(x)_j = \frac{1}{\sqrt{m}}\big(\theta_1(\cdot, j)^\top x\big)_+$

Random feature kernel regime - I

- **Prediction function** $h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - Input weights $\theta_1(\cdot, j)$, $j = 1, \ldots, m$, random and fixed
  - Optimize over output weights $\theta_2 \in \mathbb{R}^m$
  - Corresponds to linear predictor with $\Phi(x)_j = \frac{1}{\sqrt{m}}\big(\theta_1(\cdot, j)^\top x\big)_+$

- **Converges to separator with minimum norm** $\|\theta_2\|_2^2$
  - Direct application of results from Soudry et al. (2018)
  - Limit when m tends to infinity?

Random feature kernel regime - I

- **Prediction function** $h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - Input weights $\theta_1(\cdot, j)$, $j = 1, \ldots, m$, random and fixed
  - Optimize over output weights $\theta_2 \in \mathbb{R}^m$
  - Corresponds to linear predictor with $\Phi(x)_j = \frac{1}{\sqrt{m}}\big(\theta_1(\cdot, j)^\top x\big)_+$

- **Converges to separator with minimum norm** $\|\theta_2\|_2^2$
  - Direct application of results from Soudry et al. (2018)
  - Limit when m tends to infinity?

- **Kernel** $\Phi(x)^\top \Phi(x') = \frac{1}{m}\sum_{j=1}^{m} \big(\theta_1(\cdot, j)^\top x\big)_+ \big(\theta_1(\cdot, j)^\top x'\big)_+$
  - Converges to $\mathbb{E}_\eta \big(\eta^\top x\big)_+ \big(\eta^\top x'\big)_+$
  - "Random features" (Neal, 1995; Rahimi and Recht, 2007)

Random feature kernel regime - II

- **Limiting kernel** $\mathbb{E}_\eta \left(\eta^\top x\right)_+ \left(\eta^\top x'\right)_+$
  - Reproducing kernel Hilbert spaces (RKHS) (see, e.g., Schölkopf and Smola, 2001)
  - Space of (very) smooth functions (Bach, 2017)

Random feature kernel regime - II

- **Limiting kernel** $\mathbb{E}_\eta (\eta^\top x)_+ (\eta^\top x')_+$
  - Reproducing kernel Hilbert spaces (RKHS) (see, e.g., Schölkopf and Smola, 2001)
  - Space of (very) smooth functions (Bach, 2017)

- **(informal) theorem** (Chizat and Bach, 2020): when $m \to +\infty$, the gradient flow converges to the function in the RKHS that separates the data with minimum RKHS norm
  - Quantitative analysis available
  - Letting $m \to +\infty$ is useless in practice
  - See Montanari et al. (2019) for related work in the context of "double descent"

From RKHS norm to variation norm

- **Alternative definition of the RKHS norm**

$$\|f\|^2 = \inf_{a(\cdot)} \int_{\mathbb{S}^d} |a(\eta)|^2 d\tau(\eta) \quad \text{such that} \quad f(x) = \int_{\mathbb{S}^d} (\eta^\top x)_+ a(\eta) d\tau(\eta)$$

  - Input weigths uniformly distributed on the sphere (Bach, 2017)
  - Smooth functions (does not allow single hidden neuron)

From RKHS norm to variation norm

- **Alternative definition of the RKHS norm**

$$\|f\|^2 = \inf_{a(\cdot)} \int_{\mathbb{S}^d} |a(\eta)|^2 d\tau(\eta) \quad \text{such that} \quad f(x) = \int_{\mathbb{S}^d} (\eta^\top x)_+ a(\eta) d\tau(\eta)$$

  - Input weigths uniformly distributed on the sphere (Bach, 2017)
  - Smooth functions (does not allow single hidden neuron)

- **Variation norm** (Kurkova and Sanguineti, 2001)

$$\Omega(f) = \inf_{a(\cdot)} \int_{\mathbb{S}^d} |a(\eta)| d\tau(\eta) \quad \text{such that} \quad f(x) = \int_{\mathbb{S}^d} (\eta^\top x)_+ a(\eta) d\tau(\eta)$$

  - Larger space including non-smooth functions
  - Allows single hidden neuron
  - Adaptivity to linear structures (Bach, 2017)

Feature learning regime

- **Prediction function** $h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - Optimize over all weights θ_1, θ_2

Feature learning regime

- **Prediction function** $h(x) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j)\big(\theta_1(\cdot, j)^\top x\big)_+$
  - Optimize over all weights θ_1, θ_2
- **(informal) theorem** (Chizat and Bach, 2020): when $m \to +\infty$, the gradient flow converges to the function that separates the data with minimum **variation norm**
  - Actual learning of representations
  - Adaptivity to linear structures (see Chizat and Bach, 2020)
  - No known convex optimization algorithms in polynomial time
  - End of the curve of double descent (Belkin et al., 2018)

Optimizing over two layers

- Two-dimensional classification with "bias" term

Space of parameters

- Plot of $|\theta_2(j)|\theta_1(\cdot, j)$
- Color depends on sign of $\theta_2(j)$
- "tanh" radial scale

Space of predictors

- $(+/-)$ training set
- One color per class
- Line shows 0 level set of h

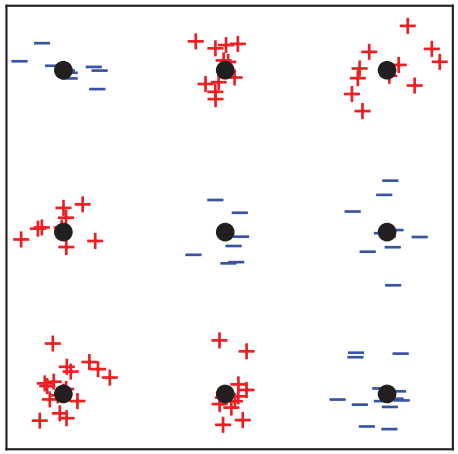
Comparison of kernel and feature learning regimes

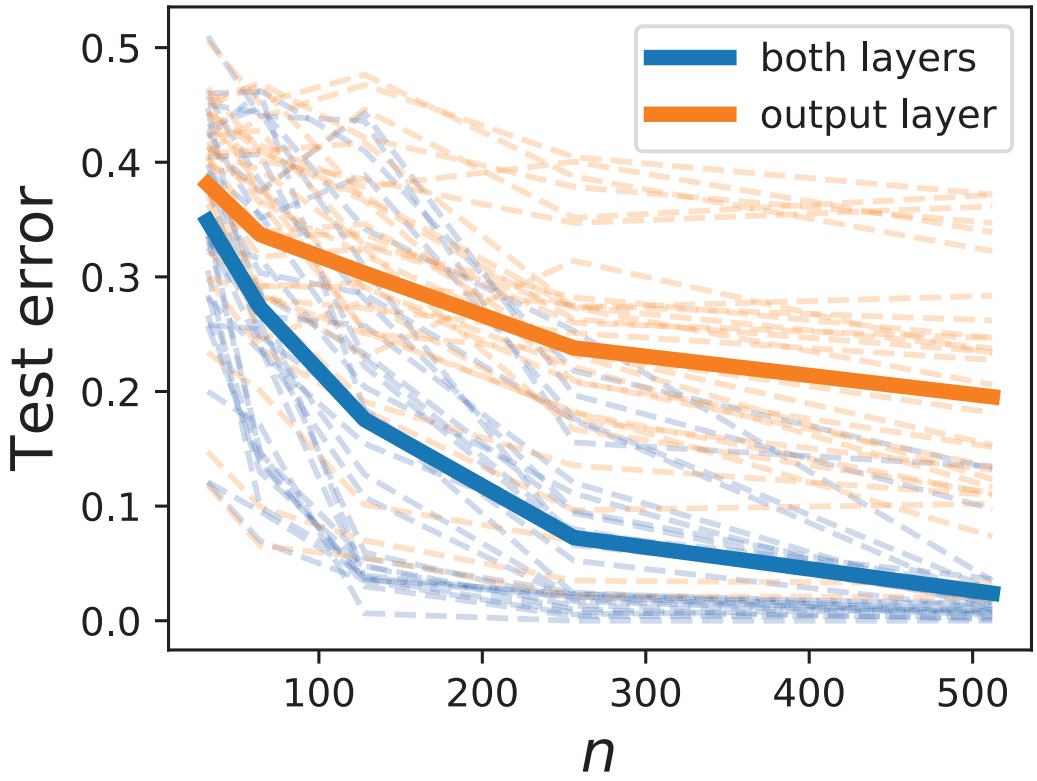
- ℓ_2 (left: kernel) vs. ℓ_1 (right: feature learning and variation norm)

Comparison of kernel and feature learning regimes

- **Adaptivity to linear structures**

- **Two-class classification in dimension** $d = 15$
  - Two first coordinates as shown below
  - All other coordinates uniformly at random





Conclusion

- **Summary**
  - Qualitative analysis of gradient descent for 2-layer neural networks
  - Global convergence with infinitely many neurons
  - Convergence to maximum margin separators in well-defined function spaces
  - Only qualitative

Conclusion

- **Summary**
  - Qualitative analysis of gradient descent for 2-layer neural networks
  - Global convergence with infinitely many neurons
  - Convergence to maximum margin separators in well-defined function spaces
  - Only qualitative

- **Open problems**
  - Quantitative analysis in terms of number of neurons m and time t
  - Extension to convolutional neural networks
  - Extension to deep neural networks

Healthy interactions between theory, applications, and hype?

Healthy interactions between theory, applications, and hype?

- **Empirical successes of deep learning cannot be ignored**

Healthy interactions between theory, applications, and hype?

- **Empirical successes of deep learning cannot be ignored**
- **Scientific standards should not be lowered**
  - Critics and limits of theoretical and empirical results
  - Rigor beyond mathematical guarantees

Healthy interactions between theory, applications, and hype?

- **Empirical successes of deep learning cannot be ignored**
- **Scientific standards should not be lowered**
  - Critics and limits of theoretical and empirical results
  - Rigor beyond mathematical guarantees
- **Some wisdom from physics:**

  Physical Review adheres to the following policy with respect to use of terms such as "new" or "novel:" All material accepted for publication in the Physical Review is expected to contain new results in physics. Phrases such as "new," "for the first time," etc., therefore should normally be unnecessary; they are not in keeping with the journal's scientific style. Furthermore, such phrases could be construed as claims of priority, which the editors cannot assess and hence must rule out.

Conclusions
Optimization for machine learning

- **Well understood**
  - Convex case with a single machine
  - Matching lower and upper bounds for variants of SGD
  - Non-convex case: SGD for local risk minimization

Conclusions
Optimization for machine learning

- **Well understood**
  - Convex case with a single machine
  - Matching lower and upper bounds for variants of SGD
  - Non-convex case: SGD for local risk minimization

- **Not well understood**: many open problems
  - Step-size schedules and acceleration
  - Dealing with non-convexity
    (global minima vs. local minima and stationary points)
  - Distributed learning: multiple cores, GPUs, and cloud (see, e.g., Hendrikx, Bach, and Massoulié, 2019, and references therein)

References

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

F. Bach and E. Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Adv. NIPS*, 2011.

F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Adv. NIPS*, 2013.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012a.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization, 2012b.

Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(1):629–681, 2017.

Francis Bach, Julien Mairal, and Jean Ponce. Convex sparse matrix factorizations. Technical Report 0812.1869, arXiv, 2008.

A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118*, 2018.

Y. Bengio, N. Le Roux, P. Vincent, O. Delalleau, and P. Marcotte. Convex neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

D. Blatt, A. O. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2008.

L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Adv. NIPS*, 2008.

Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046, 2018.

Lénaïc Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. *arXiv preprint arXiv:2002.04486*, 2020.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014a.

A. Defazio, J. Domke, and T. S. Caetano. Finito: A faster, permutable incremental gradient method for big data problems. In *Proc. ICML*, 2014b.

Aaron Defazio. A simple practical accelerated method for finite sums. In *Advances in Neural Information Processing Systems*, pages 676–684, 2016.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization. *Optimization Online*, July, 2010.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pages 6151–6159, 2017.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841, 2018.

M. Gurbuzbalaban, A. Ozdaglar, and P. Parrilo. On the convergence rate of incremental aggregated gradient algorithms. Technical Report 1506.02081, arXiv, 2015.

Benjamin D. Haeffele and René Vidal. Global optimality in neural network training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7331–7339, 2017.

Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. Asynchronous accelerated proximal stochastic gradient for strongly convex distributed finite sums. Technical Report 1901.09865, arXiv, 2019.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. *arXiv preprint arXiv:1703.00887*, 2017.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013.

V. Kurkova and M. Sanguineti. Bounds on rates of variable-basis and neural-network approximation. *IEEE Transactions on Information Theory*, 47(6):2659–2665, Sep 2001.

S. Lacoste-Julien, M. Schmidt, and F. Bach. A simpler approach to obtaining an o (1/t) convergence

rate for projected stochastic subgradient descent. Technical Report 1212.2002, ArXiv, 2012.

Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-coordinate {Frank-Wolfe} optimization for structural {SVMs}. In *Proceedings of The 30th International Conference on Machine Learning*, pages 53–61, 2013.

G. Lan. An optimal randomized incremental gradient method. Technical Report 1507.02000, arXiv, 2015.

N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

R. Leblond, F. Pedregosa, and S. Lacoste-Julien. Asaga: Asynchronous parallel Saga. Technical Report 1606.04809, arXiv, 2016.

Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016.

H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning. In *International Conference on Machine Learning*, pages 3325–3334, 2018.

J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layers

neural networks. Technical Report 1804.06561, arXiv, 2018.

Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv preprint arXiv:1911.01544*, 2019.

R. M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1995.

A. Nedic and D. Bertsekas. Convergence rate of incremental subgradient algorithms. *Stochastic Optimization: Algorithms and Applications*, pages 263–304, 2000.

A. S. Nemirovski and D. B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley & Sons, 1983.

Y. Nesterov. A method for solving a convex programming problem with rate of convergence $O(1/k^2)$. *Soviet Math. Doklady*, 269(3):543–547, 1983.

Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Kluwer, 2004.

Y. Nesterov. Gradient methods for minimizing composite objective function. *Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Tech. Rep*, 76, 2007.

A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Atsushi Nitanda and Taiji Suzuki. Stochastic particle gradient descent for infinite ensembles. *arXiv preprint arXiv:1712.05438*, 2017.

Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. Technical Report 1805.10074, arXiv, 2018.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20:1177–1184, 2007.

S. J. Reddi, A. Hefny, S. Sra, B. Póczós, and A. Smola. Stochastic variance reduction for nonconvex optimization. Technical Report 1603.06160, arXiv, 2016.

H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statistics*, 22:400–407, 1951.

S. Rosset, G. Swirszcz, N. Srebro, and J. Zhu. ℓ_1-regularization in infinite dimensional feature spaces. In *Proceedings of the Conference on Learning Theory (COLT)*, 2007.

Grant M. Rotskoff and Eric Vanden-Eijnden. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *arXiv preprint arXiv:1805.00915*, 2018.

M. Schmidt, N. Le Roux, and F. Bach. Convergence rates for inexact proximal-gradient method. In *Adv. NIPS*, 2011.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2001.

Damien Scieur, Alexandre d'Aspremont, and Francis Bach. Regularized nonlinear acceleration. In *Advances In Neural Information Processing Systems*, pages 712–720, 2016.

S. Shalev-Shwartz. Sdca without duality, regularization, and individual convexity. Technical Report 1602.01582, arXiv, 2016.

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.

S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized

loss minimization. In *Proc. ICML*, 2014.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks. *arXiv preprint arXiv:1805.01053*, 2018.

M. V. Solodov. Incremental gradient algorithms with stepsizes bounded away from zero. *Computational Optimization and Applications*, 11(1):23–35, 1998.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 2018.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1): 2822–2878, 2018.

Adrien Taylor and Francis Bach. Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions. *arXiv preprint arXiv:1902.00947*, 2019.

Adrien B Taylor, Julien M Hendrickx, and François Glineur. Performance estimation toolbox (pesto): automated worst-case analysis of first-order optimization methods. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1278–1283. IEEE, 2017.

P. Tseng. An incremental gradient(-projection) method with momentum term and adaptive stepsize rule. *SIAM Journal on Optimization*, 8(2):506–531, 1998.

I. Tsochantaridis, Thomas Joachims, T., Y. Altun, and Y. Singer. Large margin methods for structured

and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

V. N. Vapnik and A. Ya. Chervonenkis. On a perceptron class. *Avtomat. i Telemekh.*, 25(1):112–120, 1964.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1195–1204, 2019.

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 9:2543–2596, 2010.

L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

L. Zhang, M. Mahdavi, and R. Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, 2013.