

# Deep learning in the news

The Man Behind the Google Brain: Andrew Ng and the Quest for the New AI  
BY DANIELA HERNANDEZ 01/07/13 6:30 AM

**WIRED**

Researcher Dreams Up Machines That Learn Without Humans

**MIT Technology Review**

**10 BREAKTHROUGH TECHNOLOGIES 2013**

Meet Facebook's Head of Artificial Intelligence

**FACEBOOK HEAD OF A.I.**

Yoshua Bengio. Image: C

Recomm...  
Why Isn't...  
Growing as...  
Facebook?  
U.S. Says...  
Shows Rus...  
Into Ukrain...

**Deep Learning**  
With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.

**Temporary Social Media**  
Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous.

**Prenatal DNA Sequencing**  
Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child?

**THE GLOBE AND MAIL**  
CANADA'S NATIONAL NEWSPAPER • FOUNDED 1844

Google taps U of T professor to teach context to computers  
03.11.13

**The New York Times**  
Scientists See Promise in Deep-Learning Programs  
John Markoff November 23, 2012

# Review of fundamentals

Notes taken from Hugo Larochelle (IFT 725 - Réseaux neuronaux)

# TO LEARN MORE...

**Topics:** online videos on neural networks

Click with the mouse or tablet to draw with pen 2

## RESTRICTED BOLTZMANN MACHINE

**Topics:** RBM, visible layer; hidden layer; energy function

Diagram illustrating the structure of a Restricted Boltzmann Machine (RBM). It shows two layers of binary units: the hidden layer ( $\mathbf{h}$ ) and the visible layer ( $\mathbf{x}$ ). The hidden layer has four units, labeled  $b_j$ , and the visible layer has four units, labeled  $c_k$ . The layers are connected by weights  $\mathbf{W}$ . A bias  $b_j$  is associated with each unit in the hidden layer, and a bias  $c_k$  is associated with each unit in the visible layer. The diagram is annotated with labels: "hidden layer (binary units)" for  $\mathbf{h}$ , "visible layer (binary units)" for  $\mathbf{x}$ , "bias" for  $b_j$ , and "connections" for  $\mathbf{W}$ .

Energy function: 
$$\begin{aligned} E(\mathbf{x}, \mathbf{h}) &= -\mathbf{h}^T \mathbf{W} \mathbf{x} - \mathbf{c}^T \mathbf{x} - \mathbf{b}^T \mathbf{h} \\ &= -\sum_j \sum_k W_{j,k} h_j x_k - \sum_k c_k x_k - \sum_j b_j h_j \end{aligned}$$

Distribution:  $p(\mathbf{x}, \mathbf{h}) = \exp(-E(\mathbf{x}, \mathbf{h}))/Z$

partition function (intractable)

A video player interface showing a thumbnail of a video. The thumbnail features a man with dark hair, glasses, and a beard, wearing a grey shirt over a red sweater. The video player has a black header bar with the text "Click with the mouse or tablet to draw with pen 2" and a black footer bar.

[http://info.usherbrooke.ca/hlarochelle/neural\\_networks](http://info.usherbrooke.ca/hlarochelle/neural_networks)

- ▶ used in his graduate course in Sherbrooke (flipped class)
- ▶ covers many other topics: convolutional networks, neural language model, restricted Boltzmann machines, autoencoders, sparse coding, etc.

# LINEAR ALGEBRA

**Topics:** matrix, vector, norms, products

- Vector:  $\mathbf{x} = [x_1, \dots, x_d]^\top = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$ 
  - product:  $\langle \mathbf{x}^{(1)}, \mathbf{x}^{(2)} \rangle = \mathbf{x}^{(1)^\top} \mathbf{x}^{(2)} = \sum_{i=1}^d x_i^{(1)} x_i^{(2)}$
  - norm:  $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathbf{x}} = \sqrt{\sum_i x_i^2}$  (Euclidean)
- Matrix:  $\mathbf{X} = \begin{bmatrix} X_{1,1} & \dots & X_{1,m} \\ \vdots & \vdots & \vdots \\ X_{n,1} & \dots & X_{n,m} \end{bmatrix}$ 
  - product:  $(\mathbf{X}^{(1)} \mathbf{X}^{(2)})_{i,j} = \mathbf{X}_{i,\cdot}^{(1)} \mathbf{X}_{\cdot,j}^{(2)} = \sum_k X_{i,k}^{(1)} X_{k,j}^{(2)}$
  - norm:  $\|\mathbf{X}\|_F = \sqrt{\text{trace}(\mathbf{X}^\top \mathbf{X})} = \sqrt{\sum_i \sum_j X_{i,j}^2}$  (Frobenius)

# LINEAR ALGEBRA

**Topics:** special matrices

- Identity matrix  $\mathbf{I}$ :  $I_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$
- Diagonal matrix  $\mathbf{X}$ :  $X_{i,j} = 0$  if  $i \neq j$
- Lower triangular matrix  $\mathbf{X}$ :  $X_{i,j} = 0$  if  $i < j$
- Symmetric matrix  $\mathbf{X}$ :  $X_{i,j} = X_{j,i}$  (i.e.  $\mathbf{X}^T = \mathbf{X}$ )
- Square matrix: matrix with same number of rows and columns

# LINEAR ALGEBRA

**Topics:** operations on matrices

- Trace of matrix:  $\text{trace}(\mathbf{X}) = \sum_i X_{i,i}$

- ▶ trace of products:

$$\text{trace}(\mathbf{X}^{(1)} \mathbf{X}^{(2)} \mathbf{X}^{(3)}) = \text{trace}(\mathbf{X}^{(3)} \mathbf{X}^{(1)} \mathbf{X}^{(2)}) = \text{trace}(\mathbf{X}^{(2)} \mathbf{X}^{(3)} \mathbf{X}^{(1)})$$

- Inverse of matrix:  $\mathbf{X}^{-1} \mathbf{X} = \mathbf{X} \mathbf{X}^{-1} = \mathbf{I}$

- ▶ doesn't exist if determinant is 0
  - ▶ inverse of product:  $(\mathbf{X}^{(1)} \mathbf{X}^{(2)})^{-1} = \mathbf{X}^{(2)-1} \mathbf{X}^{(1)-1}$

- Transpose of matrix:  $(\mathbf{X}^\top)_{i,j} = \mathbf{X}_{j,i}$

- ▶ transpose of product:  $(\mathbf{X}^{(1)} \mathbf{X}^{(2)})^\top = \mathbf{X}^{(2)\top} \mathbf{X}^{(1)\top}$

# LINEAR ALGEBRA

**Topics:** operations on matrices

- Determinant
  - ▶ of triangular matrix:  $\det(\mathbf{X}) = \prod_i \mathbf{X}_{i,i}$
  - ▶ of transpose of matrix:  $\det(\mathbf{X}^\top) = \det(\mathbf{X})$
  - ▶ of inverse of matrix:  $\det(\mathbf{X}^{-1}) = \det(\mathbf{X})^{-1}$
  - ▶ of product of matrix:  $\det(\mathbf{X}^{(1)}\mathbf{X}^{(2)}) = \det(\mathbf{X}^{(1)})\det(\mathbf{X}^{(2)})$

# LINEAR ALGEBRA

**Topics:** properties of matrices

- Orthogonal matrix:  $\mathbf{X}^\top = \mathbf{X}^{-1}$
- Positive definite matrix:  $\mathbf{v}^\top \mathbf{X} \mathbf{v} > 0 \quad \forall \mathbf{v} \in \mathbb{R}^d$ 
  - ▶ if  $\leq$ , then positive semi-definite

# LINEAR ALGEBRA

**Topics:** linear dependence, rank, range and nullspace

- Set of linearly dependent vectors  $\{\mathbf{x}^{(t)}\}$ :

$$\exists \mathbf{w}, t^* \text{ such that } \mathbf{x}^{(t^*)} = \sum_{t \neq t^*} w_t \mathbf{x}^{(t)}$$

- Rank of matrix: number of linear independent columns
- Range of a matrix:

$$\mathcal{R}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n \mid \exists \mathbf{w} \text{ such that } \mathbf{x} = \sum_j w_j \mathbf{A}_{\cdot,j}\}$$

- Null space of a matrix:

$$\text{Null}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} = \mathbf{0}\}$$

# LINEAR ALGEBRA

**Topics:** eigenvalues and eigenvectors of a matrix

- Eigenvalues and eigenvectors

$$\{\lambda_i, \mathbf{u}_i \mid \mathbf{X}\mathbf{u}_i = \lambda_i \mathbf{u}_i \text{ and } \mathbf{u}_i^\top \mathbf{u}_j = 1_{i=j}\}$$

- Properties

- can write  $\mathbf{X} = \sum_i \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$
  - determinant of **any** matrix:  $\det(\mathbf{X}) = \prod_i \lambda_i$
  - positive definite if  $\lambda_i > 0 \quad \forall i$
  - rank of matrix is the number of non-zero eigenvalues
- For more, see the Matrix Cookbook
    - [http://www2.imm.dtu.dk/pubdb/views/edoc\\_download.php/3274/pdf/imm3274.pdf](http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf)

# DIFFERENTIAL CALCULUS

**Topics:** derivative, partial derivative

- Derivative:

$$\frac{d}{dx} f(x) = \lim_{\Delta \rightarrow 0} \frac{f(x + \Delta) - f(x)}{\Delta}$$

- ▶ direction and rate of increase of function
- Partial derivative:

$$\frac{\partial}{\partial x} f(x, y) = \lim_{\Delta \rightarrow 0} \frac{f(x + \Delta, y) - f(x, y)}{\Delta}$$

$$\frac{\partial}{\partial y} f(x, y) = \lim_{\Delta \rightarrow 0} \frac{f(x, y + \Delta) - f(x, y)}{\Delta}$$

- ▶ direction and rate of increase for variable assuming others are fixed

# DIFFERENTIAL CALCULUS

**Topics:** derivative, partial derivative

- Example:

$$f(x, y) = \frac{x^2}{y}$$

$$\frac{\partial f(x, y)}{\partial x} = \frac{2x}{y}$$

treat  $y$   
as constant

$$\frac{\partial f(x, y)}{\partial y} = \frac{-x^2}{y^2}$$

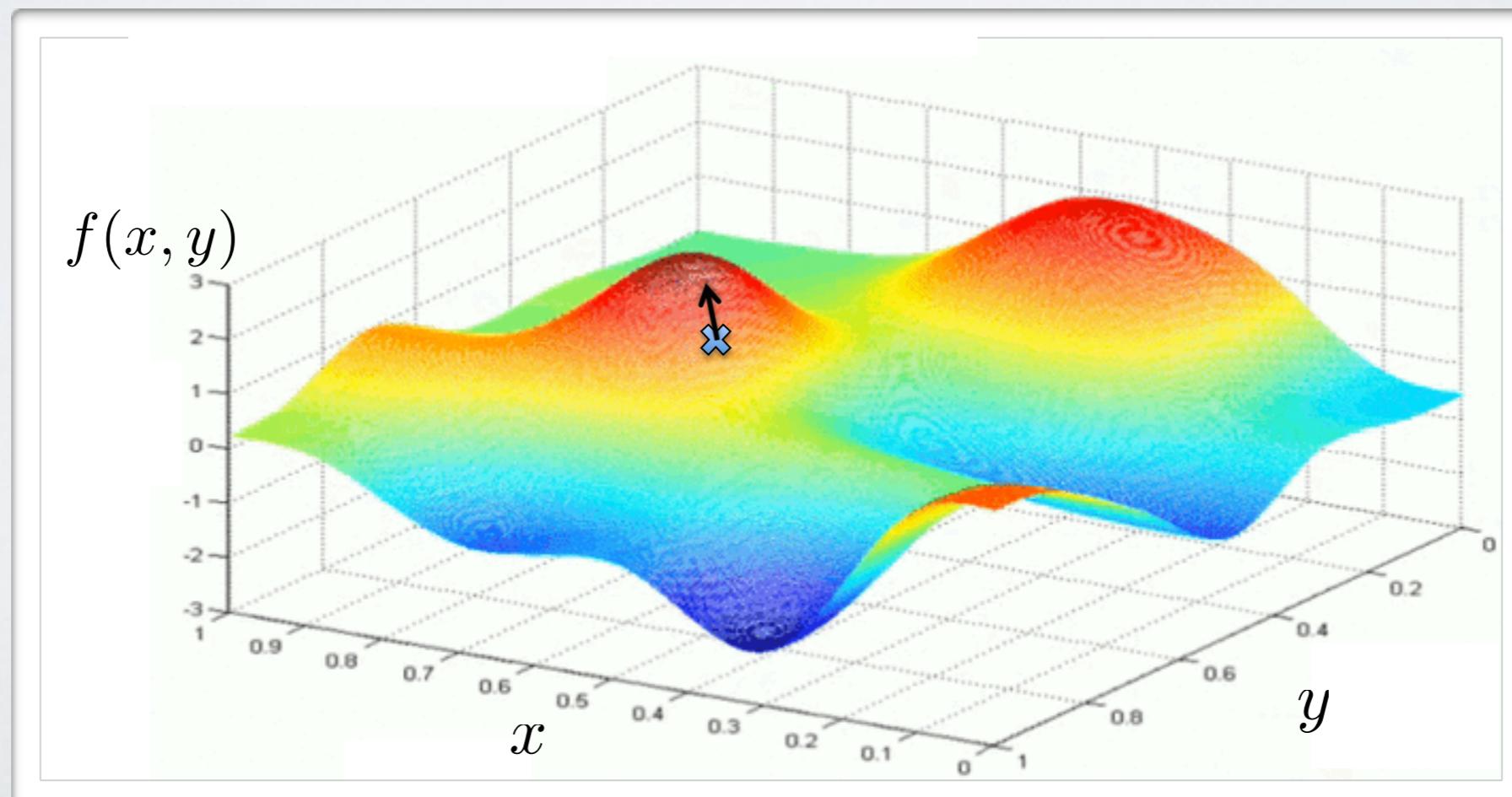
treat  $x$   
as constant

# DIFFERENTIAL CALCULUS

**Topics:** gradient

- Gradient:

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \left[ \frac{\partial}{\partial x_1} f(\mathbf{x}), \dots, \frac{\partial}{\partial x_d} f(\mathbf{x}) \right]^{\top} = \begin{bmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_d} f(\mathbf{x}) \end{bmatrix}$$



# DIFFERENTIAL CALCULUS

**Topics:** Jacobian, Hessian

- Hessian:

$$\nabla_{\mathbf{x}}^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2}{\partial x_1^2} f(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_d} f(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_d \partial x_1} f(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_d^2} f(\mathbf{x}) \end{bmatrix}$$

- $\mathbf{f}(\mathbf{x}) = [f(\mathbf{x})_1, \dots, f(\mathbf{x})_k]^\top$  is a vector; the Jacobian is:

$$\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x})_1 & \cdots & \frac{\partial}{\partial x_d} f(\mathbf{x})_1 \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1} f(\mathbf{x})_k & \cdots & \frac{\partial}{\partial x_d} f(\mathbf{x})_k \end{bmatrix}$$

# DIFFERENTIAL CALCULUS

**Topics:** gradient for matrices

- If scalar function  $f(\mathbf{X})$  takes a matrix  $\mathbf{X}$  as input

$$\nabla_{\mathbf{X}} f(\mathbf{X}) = \begin{bmatrix} \frac{\partial}{\partial X_{1,1}} f(\mathbf{X}) & \cdots & \frac{\partial}{\partial X_{1,m}} f(\mathbf{X}) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial X_{n,1}} f(\mathbf{X}) & \cdots & \frac{\partial}{\partial X_{n,m}} f(\mathbf{X}) \end{bmatrix}$$

- For functions that output functions and take matrices as input, we organize into 3D tensors

# PROBABILITY

**Topics:** probability space

- Probability space: triplet  $(\Omega, \mathcal{F}, P)$

- $\Omega$  is the space of possible outcomes
- $\mathcal{F}$  is the space of possible events
- $P$  is a probability measure mapping an **event** to its probability  $[0, 1]$
- example: throwing a die

- $\Omega = \{1, 2, 3, 4, 5, 6\}$
- $e = \{1, 5\} \in \mathcal{F}$  (i.e. die is either 1 or 5)
- $P(\{1, 5\}) = \frac{2}{6}$

- Properties:

$$1. P(\{\omega\}) \geq 0 \quad \forall \omega \in \Omega \quad 2. \sum_{\omega \in \Omega} P(\{\omega\}) = 1$$

# PROBABILITY

**Topics:** random variable

- Random variable: a function on outcomes
- Examples:
  - $X$  is the value of the outcome
  - $O$  is 1 if the outcome is 1, 3 or 5, otherwise it's 0
  - $S$  is 1 if the outcome is smaller than 4, otherwise it's 0

# PROBABILITY

**Topics:** distributions (joint, marginal, conditional)

- Joint distribution:  $p(X = x, O = o, S = s)$  ( $p(x, s, o)$  for short)
  - ▶ the probability of a complete assignment of many random variables
  - ▶ example:  $p(X = 1, O = 1, S = 0) = 0$
- Marginal distribution:  $p(o, s) = \sum_x p(x, o, s)$ 
  - ▶ the probability of a partial assignment
  - ▶ example:  $p(O = 1, S = 0) = \frac{1}{6}$
- Conditional distribution:  $p(S = s | O = o)$ 
  - ▶ the probability of some variables, assuming an assignment of other variables
  - ▶ example:  $p(S = 1 | O = 1) = \frac{2}{3}$

# PROBABILITY

**Topics:** probability chain rule, Bayes rule

- Probability chain rule:  $p(s, o) = p(s|o)p(o) = p(o|s)p(s)$ 
  - ▶ in general:

$$p(\mathbf{x}) = \prod_i p(x_i | x_1, \dots, x_{i-1})$$

- Bayes rule:

$$p(O = o | S = s) = \frac{p(S=s | O=o)p(O=o)}{\sum_{o'} p(S=s | O=o')p(O=o')}$$

# PROBABILITY

**Topics:** independence between variables

- Independence: variables  $X_1$  and  $X_2$  are independent if

$$p(x_1, x_2) = p(x_1)p(x_2)$$

or  $p(x_1|x_2) = p(x_1)$

or  $p(x_2|x_1) = p(x_2)$

- Conditional independence: variables  $X_1$  and  $X_2$  are independent given  $X_3$  if

$$p(x_1, x_2|x_3) = p(x_1|x_3)p(x_2|x_3)$$

or  $p(x_1|x_2, x_3) = p(x_1|x_3)$

or  $p(x_2|x_1, x_3) = p(x_2|x_3)$

# PROBABILITY

**Topics:** expectation, variance

- Expectation:  $E[X] = \sum_x x p(X = x)$

► properties:

- $E[X + Y] = E[X] + E[Y]$
- $E[f(X)] = \sum_x f(x) p(X = x)$
- if independent,  $E[XY] = E[X]E[Y]$

- Variance:  $\text{Var}[X] = \sum_x (x - E(X))^2 p(X = x)$

► properties:

- $\text{Var}[X] = E[X^2] - E[X]^2$
- if independent,  $\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y]$

# PROBABILITY

**Topics:** covariance matrix

- Covariance:

$$\begin{aligned}\text{Cov}(X_1, X_2) &= \mathbb{E}[(X_1 - \mathbb{E}[X_1])(X_2 - \mathbb{E}[X_2])] \\ &= \sum_{x_1} \sum_{x_2} (x_1 - \mathbb{E}[X_1])(x_2 - \mathbb{E}[X_2]) p(x_1, x_2)\end{aligned}$$

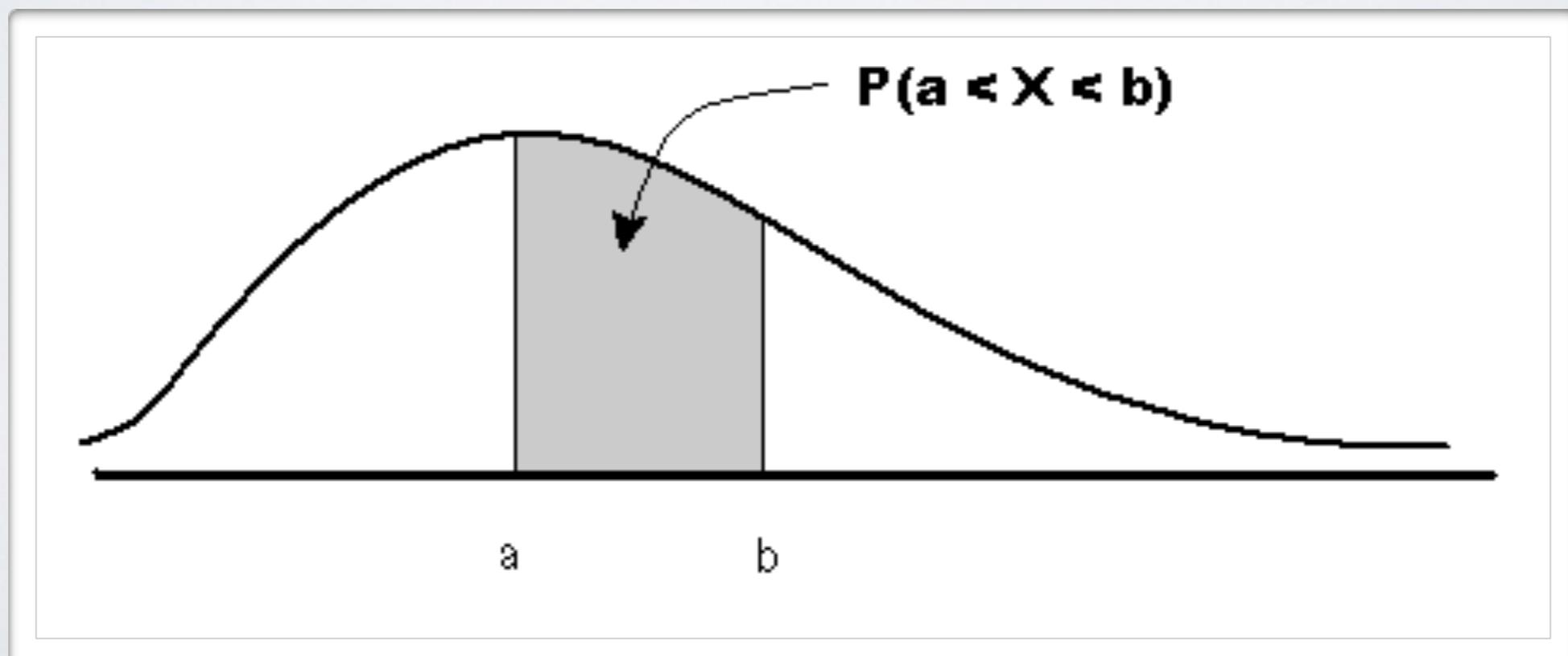
- if independent  $\text{Cov}(X_1, X_2) = 0$
- $\text{Var}(X) = \text{Cov}(X, X)$
- Covariance matrix:

$$\text{Cov}(\mathbf{X}) = \begin{bmatrix} \text{Cov}(X_1, X_1) & \dots & \text{Cov}(X_1, X_d) \\ \vdots & \vdots & \vdots \\ \text{Cov}(X_d, X_1) & \dots & \text{Cov}(X_d, X_d) \end{bmatrix}$$

# PROBABILITY

**Topics:** continuous variables

- for continuous variable  $X$ ,  $p(x)$  is a density function
  - ▶  $P(X \in A) = \int_{x \in A} p(x)dx$
  - ▶ the probability  $P(X = x)$  is zero for continuous variables
  - ▶ in previous equations, summations would be replaced by integrals



# PROBABILITY

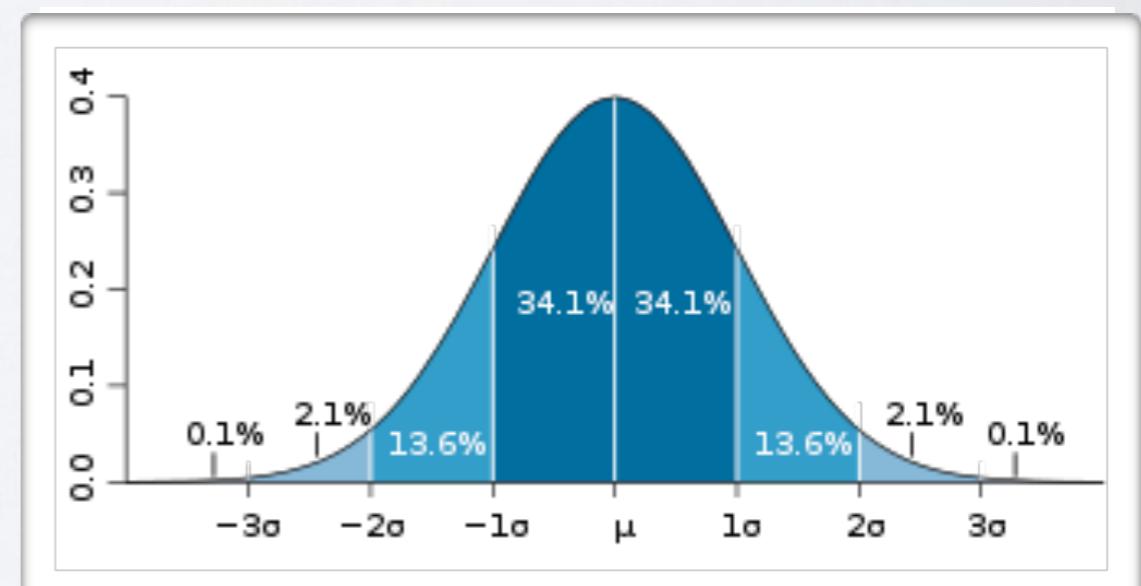
**Topics:** Bernoulli, Gaussian distributions

- Bernoulli variable:  $X \in \{0, 1\}$

- $p(X = 1) = \mu$
- $p(X = 0) = 1 - \mu$
- $E[X] = \mu$
- $\text{Var}[X] = \mu(1 - \mu)$

- Gaussian variable:  $X \in \mathbb{R}$

- $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
- $E[X] = \mu$
- $\text{Var}[X] = \sigma^2$

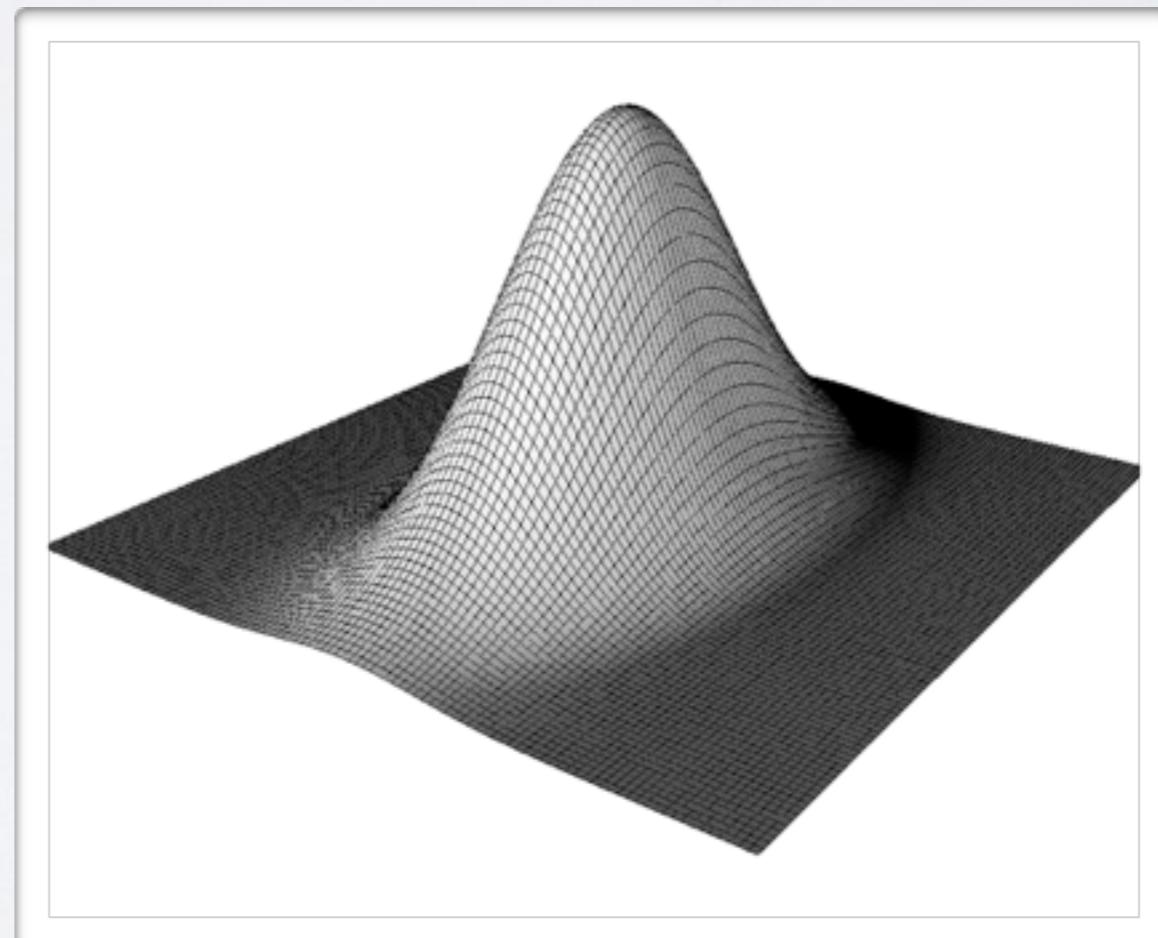


# PROBABILITY

**Topics:** Multivariate Gaussian distributions

- Gaussian variable:  $\mathbf{X} \in \mathbb{R}^d$

- $p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$
- $E[\mathbf{X}] = \boldsymbol{\mu}$
- $\text{Cov}[\mathbf{X}] = \boldsymbol{\Sigma}$



# STATISTICS

**Topics:** estimate of the expectation and covariance matrix

- Sample mean:

$$\hat{\mu} = \frac{1}{T} \sum_t \mathbf{x}^{(t)}$$

- Sample variance:

$$\hat{\sigma}^2 = \frac{1}{T-1} \sum_t (\mathbf{x}^{(t)} - \hat{\mu})^2$$

- Sample covariance matrix:

$$\hat{\Sigma} = \frac{1}{T-1} \sum_t (\mathbf{x}^{(t)} - \hat{\mu})(\mathbf{x}^{(t)} - \hat{\mu})^\top$$

- These estimators are unbiased, i.e.:

$$E[\hat{\mu}] = \mu \quad E[\hat{\sigma}^2] = \sigma^2 \quad E[\hat{\Sigma}] = \Sigma$$

# STATISTICS

**Topics:** confidence interval

- Confidence interval of the sample mean (1D):
  - ▶ if  $T$  is big, the following estimator is approx. Gaussian with mean 0 and variance 1

$$\frac{\hat{\mu} - \mu}{\sqrt{\hat{\sigma}^2 / T}}$$

- ▶ then we have that, with 95% probability, that

$$\mu \in \hat{\mu} \pm -1.96 \sqrt{\hat{\sigma}^2 / T}$$

# STATISTICS

**Topics:** maximum likelihood, I.I.D. hypothesis

- maximum likelihood estimator (MLE):

$$\hat{\theta} = \arg \max_{\theta} p(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(T)})$$

- ▶ the sample mean is the MLE for a Gaussian distribution
- ▶ the sample covariance matrix is not, but this is

$$\frac{T-1}{T} \hat{\Sigma} = \frac{1}{T} \sum_t (\mathbf{x}^{(t)} - \hat{\mu})(\mathbf{x}^{(t)} - \hat{\mu})^T$$

- Independent and identically distributed variables

$$p(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(T)}) = \prod_t p(\mathbf{x}^{(t)})$$

# SAMPLING

**Topics:** Monte Carlo estimate

- Monte Carlo estimate:
  - ▶ a method to approximate an expensive expectation

$$\mathbb{E}[f(\mathbf{X})] = \sum_{\mathbf{x}} f(\mathbf{x}) p(\mathbf{x}) \approx \frac{1}{K} \sum_k f(\mathbf{x}^{(k)})$$

- ▶ the  $\mathbf{x}^{(k)}$  must be sampled from  $p(\mathbf{x})$

# SAMPLING

**Topics:** importance sampling

- Importance sampling:

- ▶ a sampling method for when  $p(\mathbf{x})$  is expensive to sample from

$$\mathbb{E}[f(\mathbf{X})] = \sum_{\mathbf{x}} f(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) \approx \frac{1}{K} \sum_k f(\mathbf{x}^{(k)}) \frac{p(\mathbf{x})}{q(\mathbf{x})}$$

- ▶  $q(\mathbf{x})$  is easier to sample from and should be as similar as possible to  $p(\mathbf{x})$ 
    - designing a good  $q(\mathbf{x})$  is often hard to do

# SAMPLING

**Topics:** Markov Chain Monte Carlo (MCMC)

- MCMC:

- iterative method to generate the sequence of  $\mathbf{x}^{(k)}$
- the set of  $\mathbf{x}^{(k)}$  will be dependent of each other  $\mathbf{x}^{(k)}$

$$\mathbf{x}^{(1)} \xrightarrow{T(\mathbf{x}' \leftarrow \mathbf{x})} \mathbf{x}^{(2)} \xrightarrow{T(\mathbf{x}' \leftarrow \mathbf{x})} \mathbf{x}^{(3)} \xrightarrow{T(\mathbf{x}' \leftarrow \mathbf{x})} \dots \xrightarrow{T(\mathbf{x}' \leftarrow \mathbf{x})} \mathbf{x}^{(K)}$$

- $T(\mathbf{x}' \leftarrow \mathbf{x})$  is a transition operator, that must satisfy certain properties
- $K$  must be big for the set of samples be representative of distribution
- usually, we drop the first samples, which are not reliable

# SAMPLING

## Topics: Gibbs sampling

- Gibbs sampling:
  - ▶ MCMC method which uses the following transition operator  $T(\mathbf{x}' \leftarrow \mathbf{x})$ 
    - pick a variable  $x_i$
    - obtain  $\mathbf{x}'$  by only resampling this variable according to
$$p(x_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)$$
    - return  $\mathbf{x}'$
  - ▶ often, we simply cycle through the variables, in random order

Machine Learning

# MACHINE LEARNING

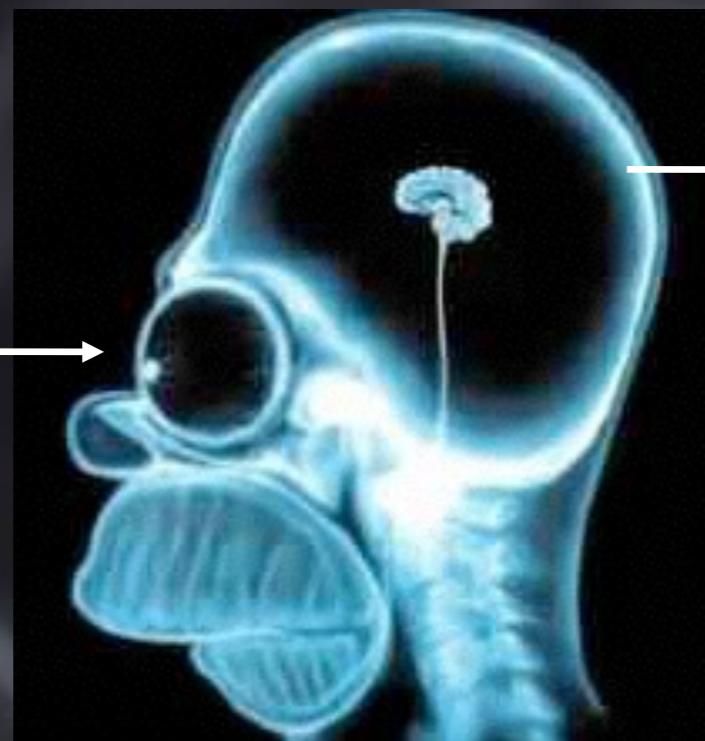
**Topics:** supervised learning

- Learning example:  $(\mathbf{x}, y)$
- Task to solve: predict target  $y$  from input  $\mathbf{x}$ 
  - ▶ classification: target is a class ID (from 0 to nb. of class - 1)
  - ▶ regression: target is a real number

# Supervised Training Example

Entrée X

e



Sortie  $f(X)$

six

Cible Y

deux!



# MACHINE LEARNING

**Topics:** unsupervised learning

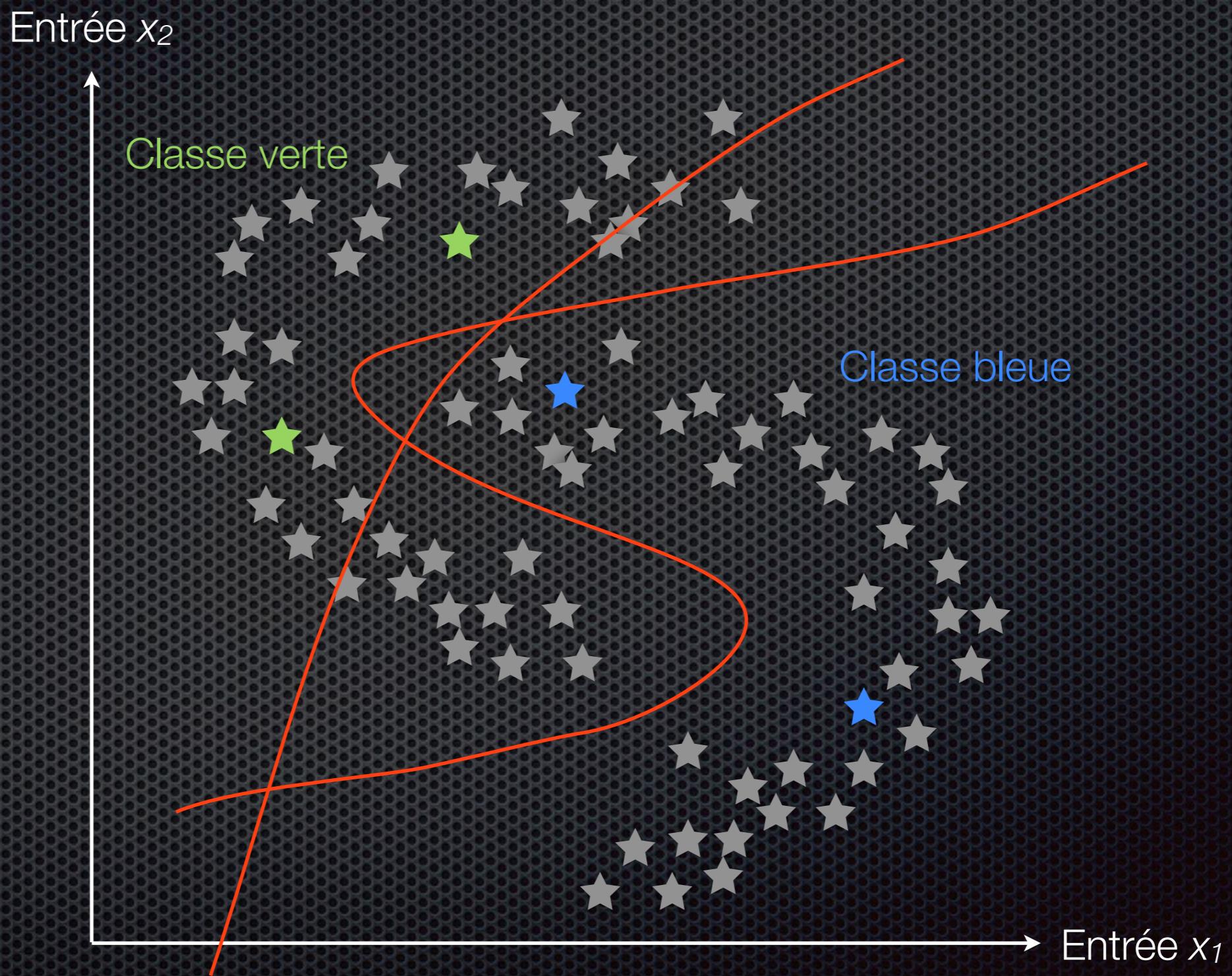
- Learning example:  $\mathbf{x}$
- No explicit target to predict
  - ▶ clustering: partition data into groups
  - ▶ feature extraction: learn meaningful features automatically
  - ▶ dimensionality reduction: learning a lower-dimensional representation of input

# MACHINE LEARNING

**Topics:** semi-supervised learning

- Few supervised learning examples:  $(\mathbf{x}, y)$
- Many unsupervised learning examples:  $\mathbf{x}$
- Task to solve: predict target  $y$  from input  $\mathbf{x}$ 
  - ▶ classification: target is a class ID (from 0 to nb. of class - 1)
  - ▶ regression: target is a real number
- Can the extra unsupervised data help?

# Apprentissage semi-supervisé



# MACHINE LEARNING

**Topics:** learning algorithm, model, training set

- Learning algorithm
  - ▶ takes as input a training set  $\mathcal{D}^{\text{train}} = \{(\mathbf{x}^{(t)}, y^{(t)})\}$
  - ▶ outputs a model  $f(\mathbf{x}; \boldsymbol{\theta})$
- We then say the model  $f(\mathbf{x}; \boldsymbol{\theta})$  was trained on  $\mathcal{D}^{\text{train}}$ 
  - ▶ the model has learned the information present in  $\mathcal{D}^{\text{train}}$
- We can now use the model  $f(\mathbf{x}; \boldsymbol{\theta})$  on new inputs

# MACHINE LEARNING

**Topics:** training, validation and test sets, generalization

- Training set  $\mathcal{D}^{\text{train}}$  serves to train a model
- Validation set  $\mathcal{D}^{\text{valid}}$  serves to select hyper-parameters
- Test set  $\mathcal{D}^{\text{test}}$  serves to estimate the generalization performance (error)
- Generalization is the behavior of the model on **unseen examples**
  - ▶ this is what we care about in machine learning!

# MACHINE LEARNING

**Topics:** capacity of a model, underfitting, overfitting, hyper-parameter, model selection

- **Capacity:** flexibility of a model
- **Hyper-parameter:** a parameter of a model that is not trained (specified before training)
- **Underfitting:** state of model which could improve generalization with more training or capacity
- **Overfitting:** state of model which could improve generalization with less training or capacity
- **Model selection:** process of choosing the best hyper-parameters on validation set

# MACHINE LEARNING

**Topics:** capacity of a model, underfitting, overfitting, hyper-parameter, model selection



# MACHINE LEARNING

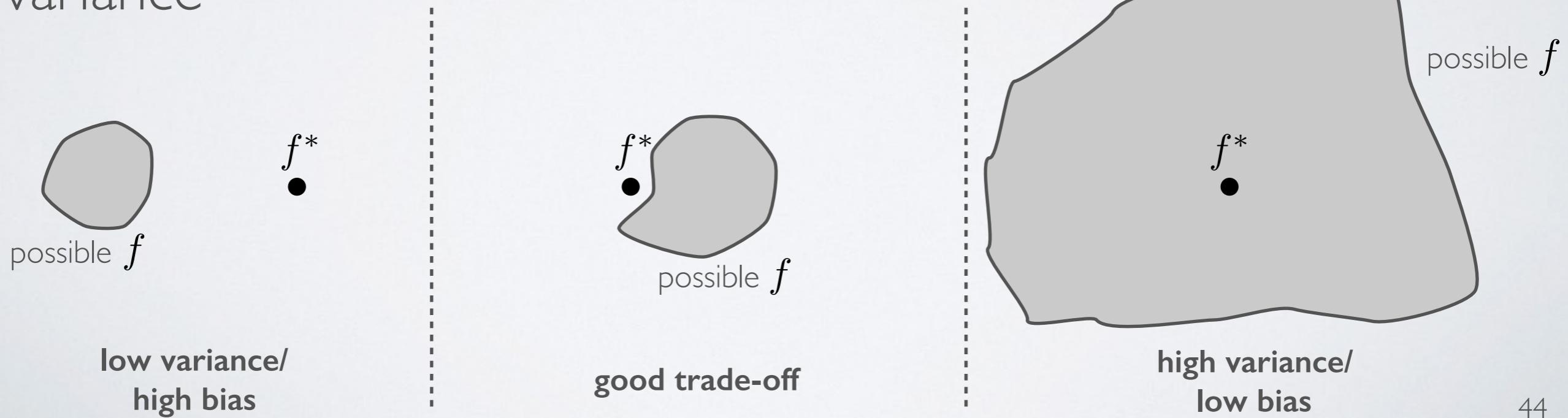
**Topics:** interaction between training set size/capacity/training time and training error/generalization error

- If capacity increases:
  - ▶ training error will decrease
  - ▶ generalization error will increase or decrease
- If training time increases:
  - ▶ training error will decrease
  - ▶ generalization error will increase or decrease
- If training set size increases:
  - ▶ generalization error will decrease (or maybe stay the same)
  - ▶ difference between the training and generalization error will decrease

# MACHINE LEARNING

**Topics:** bias-variance trade-off

- **Variance** of trained model: does it vary a lot if the training set changes
- **Bias** of trained model: is the average model close to the true solution?
- Generalization error can be seen as the sum of bias and the variance



# MACHINE LEARNING

**Topics:** empirical risk minimization, regularization

- Empirical risk minimization
  - ▶ framework to design learning algorithms

$$\arg \min_{\boldsymbol{\theta}} \frac{1}{T} \sum_t l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)}) + \lambda \Omega(\boldsymbol{\theta})$$

- ▶  $l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)})$  is a loss function
- ▶  $\Omega(\boldsymbol{\theta})$  is a regularizer (penalizes certain values of  $\boldsymbol{\theta}$ )
- Learning is cast as optimization
  - ▶ ideally, we'd optimize classification error, but it's not smooth
  - ▶ loss function is a surrogate for what we truly should optimize (e.g. upper bound)

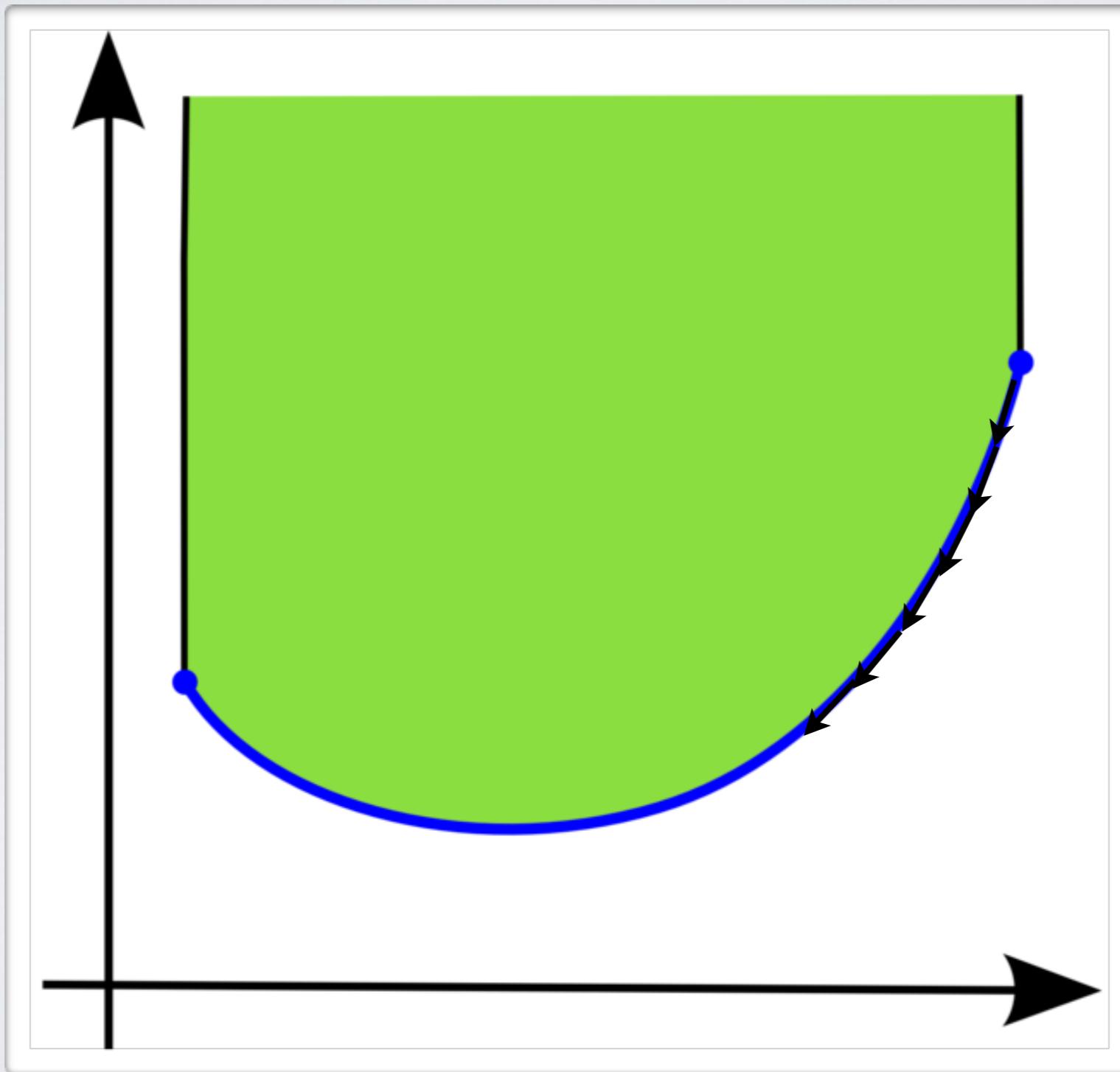
# MACHINE LEARNING

**Topics:** gradient descent

- Gradient descent: procedure to minimize a function
  - ▶ compute gradient
  - ▶ take step in opposite direction

# MACHINE LEARNING

**Topics:** gradient descent



Descent  
direction

$$-\frac{\partial f(x)}{\partial x}$$

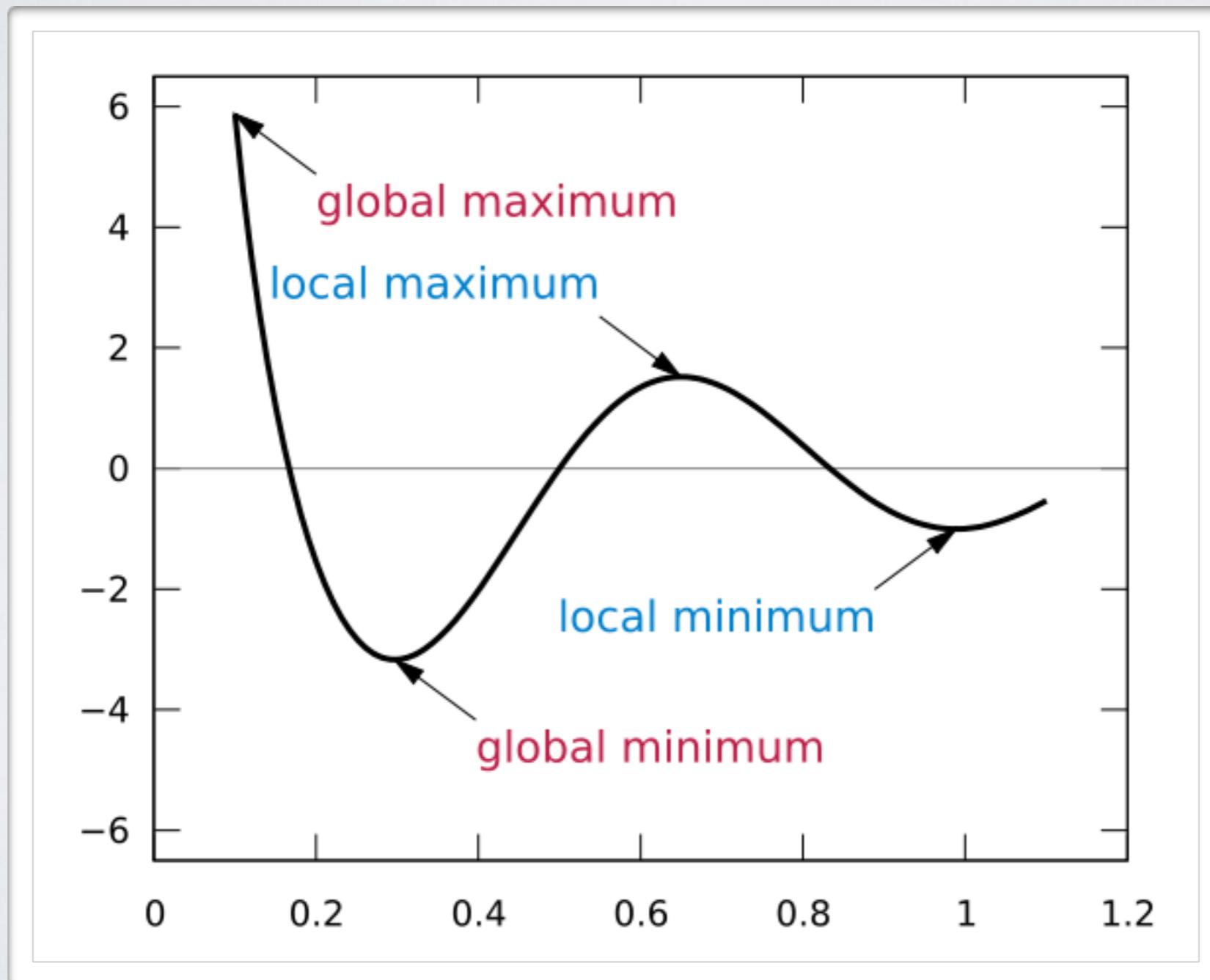
# MACHINE LEARNING

**Topics:** gradient descent

- Gradient descent for empirical risk minimization
  - initialize  $\boldsymbol{\theta}$
  - for N iterations
    - $\Delta = -\frac{1}{T} \sum_t \nabla_{\boldsymbol{\theta}} l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)}) - \lambda \nabla_{\boldsymbol{\theta}} \Omega(\boldsymbol{\theta})$
    - $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \Delta$

# MACHINE LEARNING

**Topics:** local and global optima



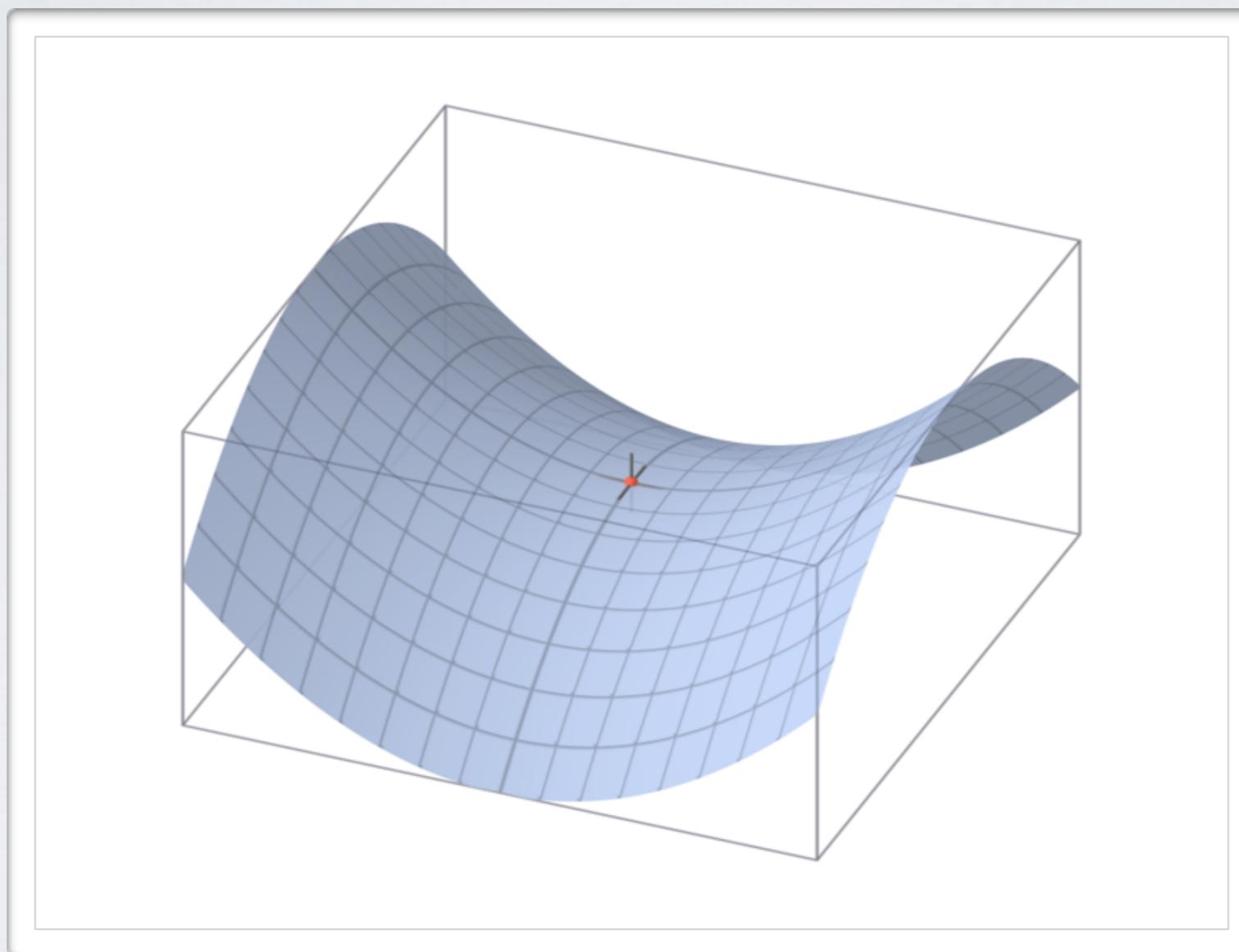
# MACHINE LEARNING

**Topics:** critical points, local optima, saddle point, curvature

- Critical points:  $\{\mathbf{x} \in \mathbb{R}^d \mid \nabla_{\mathbf{x}} f(\mathbf{x}) = 0\}$
- Curvature in direction  $\mathbf{v}$ :  $\mathbf{v}^\top \nabla_{\mathbf{x}}^2 f(\mathbf{x}) \mathbf{v}$
- Types of critical points:
  - local minima:  $\mathbf{v}^\top \nabla_{\mathbf{x}}^2 f(\mathbf{x}) \mathbf{v} > 0 \quad \forall \mathbf{v}$  (i.e.  $\nabla_{\mathbf{x}}^2 f(\mathbf{x})$  positive definite)
  - local maxima:  $\mathbf{v}^\top \nabla_{\mathbf{x}}^2 f(\mathbf{x}) \mathbf{v} < 0 \quad \forall \mathbf{v}$  (i.e.  $\nabla_{\mathbf{x}}^2 f(\mathbf{x})$  negative definite)
  - saddle point: curvature is positive in certain directions and negative in others

# MACHINE LEARNING

**Topics:** saddle point



# MACHINE LEARNING

**Topics:** stochastic gradient descent

- Algorithm that performs updates after each example
  - ▶ initialize  $\boldsymbol{\theta}$
  - ▶ for N iterations
    - for each training example  $(\mathbf{x}^{(t)}, y^{(t)})$ 
      - ✓  $\Delta = -\nabla_{\boldsymbol{\theta}} l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)}) - \lambda \nabla_{\boldsymbol{\theta}} \Omega(\boldsymbol{\theta})$
      - ✓  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \Delta$

# MACHINE LEARNING

**Topics:** parametric vs. non-parametric

- Parametric model: its capacity is fixed and does not increase with the amount of training data
  - ▶ examples: linear classifier, neural network with fixed number of hidden units, etc.
- Non-parametric model: the capacity increases with the amount of training data
  - ▶ examples: k nearest neighbors classifier, neural network with adaptable hidden layer size, etc.