Deep learning in the news

The Man Behind the Google Brain: Andrew Ng and the Quest for the New AI

BY DANIELA HERNANDEZ 05.07.13 6:30 AM

WIRED

Researcher Dreams Up Machines That Learn Without Humans

Yoshua Bengio. *Image: C*

MIT Technology Review

10 BREAKTHROUGH TECHNOLOGIES 2013

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 c fetuses will be the next frontier of the genomic revolution But do you really w to know about the genetic problems o musical aptitude of your unborn child?

Meet Facebook's Head of Artificial Intelligence

Recomme

Why Isn't Growing as Facebook?

U.S. Says Shows Rus Into Ukrain

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

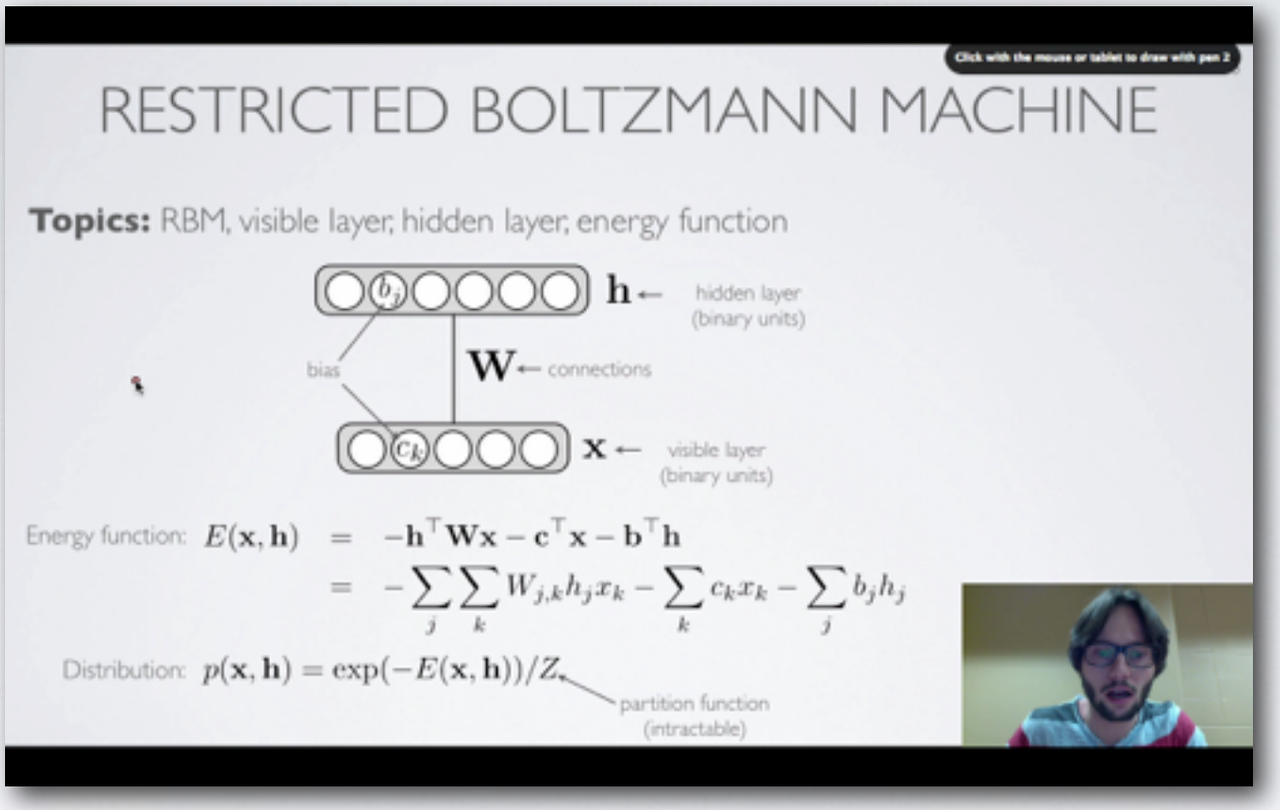
John Markoff November 23, 2012

Review of fundamentals

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

TO LEARN MORE...

Topics: online videos on neural networks



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: $\mathbf{x} = [x_1, \ldots, x_d]^\top = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$
  - product: $< \mathbf{x}^{(1)}, \mathbf{x}^{(2)} >= {\mathbf{x}^{(1)}}^\top \mathbf{x}^{(2)} = \sum_{i=1}^{d} x_i^{(1)} x_i^{(2)}$
  - norm: $||\mathbf{x}||_2 = \sqrt{\mathbf{x}^\top \mathbf{x}} = \sqrt{\sum_i x_i^2}$ (Euclidean)
- Matrix: $\mathbf{X} = \begin{bmatrix} X_{1,1} & \ldots & X_{1,m} \\ \vdots & \vdots & \vdots \\ X_{n,1} & \ldots & X_{n,m} \end{bmatrix}$
  - product: $(\mathbf{X}^{(1)}\mathbf{X}^{(2)})_{i,j} = \mathbf{X}^{(1)}_{i,\cdot}\mathbf{X}^{(2)}_{\cdot,j} = \sum_k X^{(1)}_{i,k} X^{(2)}_{k,j}$
  - 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 $\mathbf{I}$: $\mathbf{I}_{i,j} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$

- Diagonal matrix $\mathbf{X}$: $X_{i,j} = 0 \text{ if } i \neq j$

- Lower triangular matrix $\mathbf{X}$: $X_{i,j} = 0 \text{ if } i < j$

- Symmetric matrix $\mathbf{X}$: $X_{i,j} = X_{j,i}$ (i.e. $\mathbf{X}^\top = \mathbf{X}$)
- Square matrix: matrix with same number of rows and columns

LINEAR ALGEBRA

Topics: operations on matrices

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

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

- 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 « $\geq$ » , 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:

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

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

DIFFERENTIAL CALCULUS

Topics: derivative, partial derivative

- Derivative:

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

  - direction and rate of increase of function

- Partial derivative:

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

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

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

DIFFERENTIAL CALCULUS

Topics: derivative, partial derivative

- Example:

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

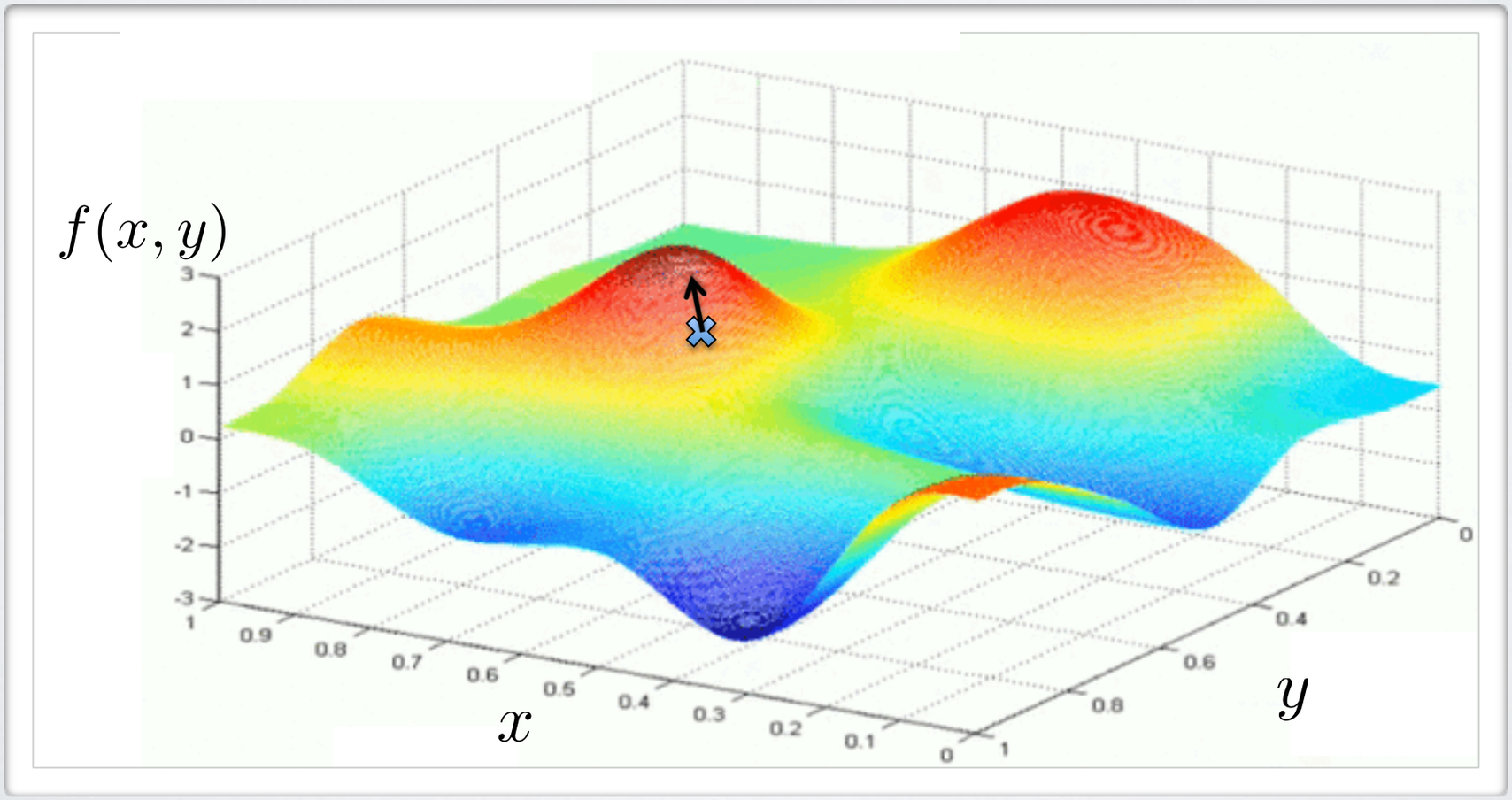
$$\frac{\partial f(x,y)}{\partial x} = \underbrace{\frac{2x}{y}}_{\text{treat } y \text{ as constant}} \qquad \frac{\partial f(x,y)}{\partial y} = \underbrace{\frac{-x^2}{y^2}}_{\text{treat } x \text{ as constant}}$$

DIFFERENTIAL CALCULUS

Topics: gradient

- Gradient:

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



DIFFERENTIAL CALCULUS

Topics: Jacobian, Hessian

- Hessian:

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

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

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

DIFFERENTIAL CALCULUS

Topics: gradient for matrices

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

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

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

PROBABILITY

Topics: probability space

- Probability space: triplet $(\Omega, \mathcal{F}, P)$
  - Ω 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. $P(\{\omega\}) \geq 0 \quad \forall \omega \in \Omega$ 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:

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

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

- Variance: $\mathrm{Var}[X] = \sum_x (x - \mathrm{E}(X))^2\ p(X = x)$
  - properties:
    - $\mathrm{Var}[X] = \mathrm{E}[X^2] - \mathrm{E}[X]^2$
    - if independent, $\mathrm{Var}[X + Y] = \mathrm{Var}[X] + \mathrm{Var}[Y]$

PROBABILITY

Topics: covariance matrix

- Covariance:

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

  - if independent $\mathrm{Cov}(X_1, X_2) = 0$
  - $\mathrm{Var}(X) = \mathrm{Cov}(X, X)$

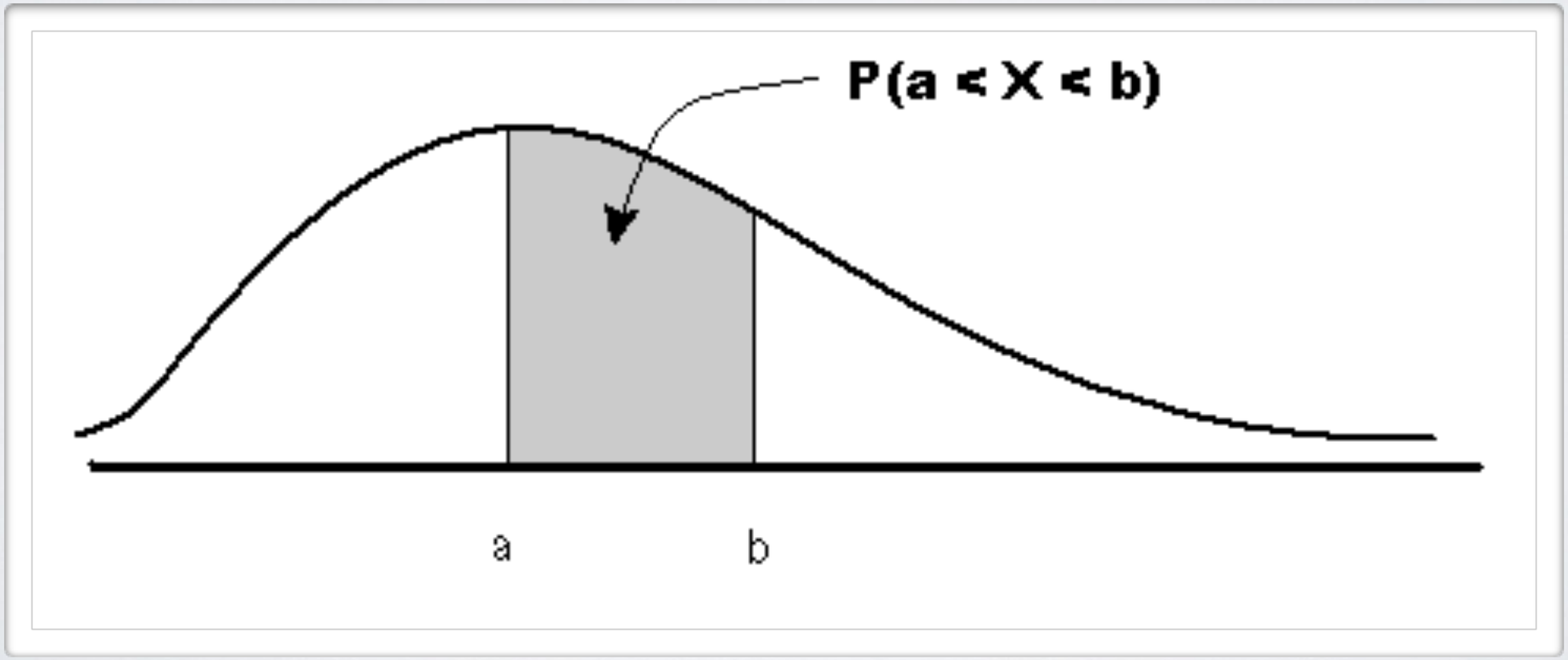
- Covariance matrix:

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

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

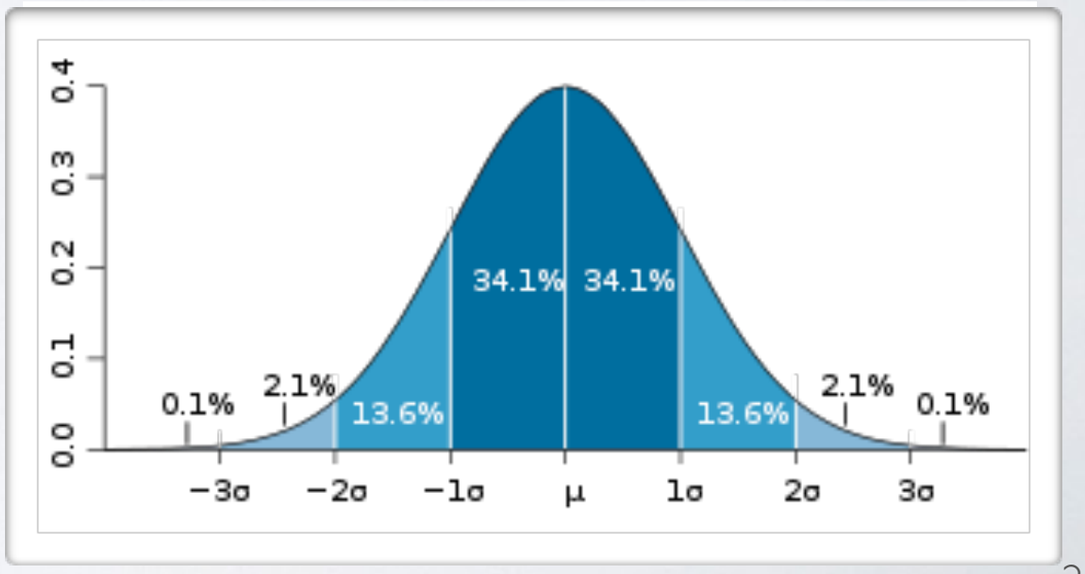


PROBABILITY

Topics: Bernoulli, Gaussian distributions

- Bernoulli variable: $X \in \{0, 1\}$
  - $p(X = 1) = \mu$
  - $p(X = 0) = 1-\mu$
  - $\mathrm{E}[X] = \mu$
  - $\mathrm{Var}[X] = \mu(1-\mu)$

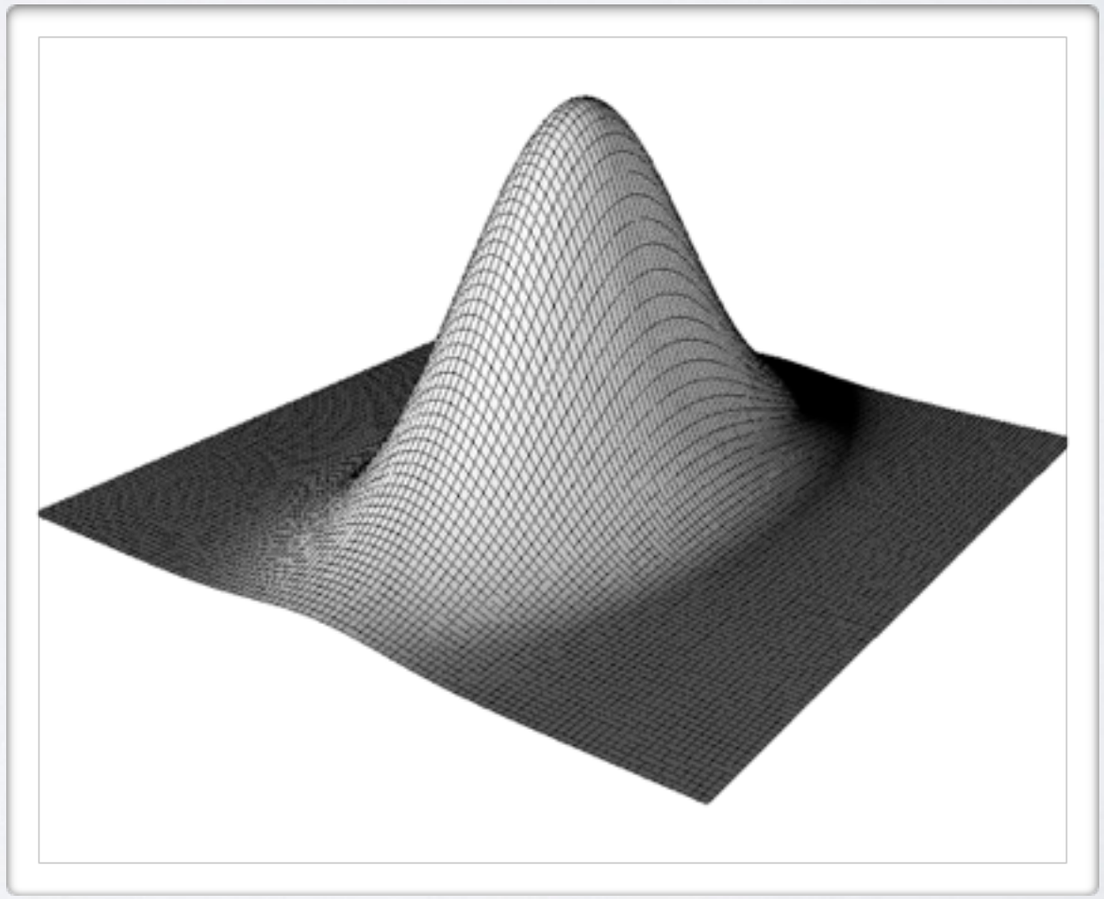
- Gaussian variable: $X \in \mathbb{R}$
  - $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
  - $\mathrm{E}[X] = \mu$
  - $\mathrm{Var}[X] = \sigma^2$



PROBABILITY

Topics: Multivariate Gaussian distributions

- Gaussian variable: $\mathbf{X} \in \mathbb{R}^d$
  - $p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$
  - $\mathrm{E}[\mathbf{X}] = \boldsymbol{\mu}$
  - $\mathrm{Cov}[\mathbf{X}] = \Sigma$



STATISTICS

Topics: estimate of the expectation and covariance matrix

- Sample mean:

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

- Sample variance:

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

- Sample covariance matrix:

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

- These estimators are unbiased, i.e.:

$$\mathrm{E}[\widehat{\boldsymbol{\mu}}] = \boldsymbol{\mu} \quad \mathrm{E}[\widehat{\sigma}^2] = \sigma^2 \quad \mathrm{E}\left[\widehat{\Sigma}\right] = \Sigma$$

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

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

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

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

STATISTICS

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

- maximum likelihood estimator (MLE):

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

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

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

- Independent and identically distributed variables

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

SAMPLING

Topics: Monte Carlo estimate

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

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

  - 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

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

  - $q(\mathbf{x})$ 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

$$p(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$$

    - return $\mathbf{x}'$

  - often, we simply cycle through the variables, in random order

Machine Learning

MACHINE LEARNING

Topics: supervised learning

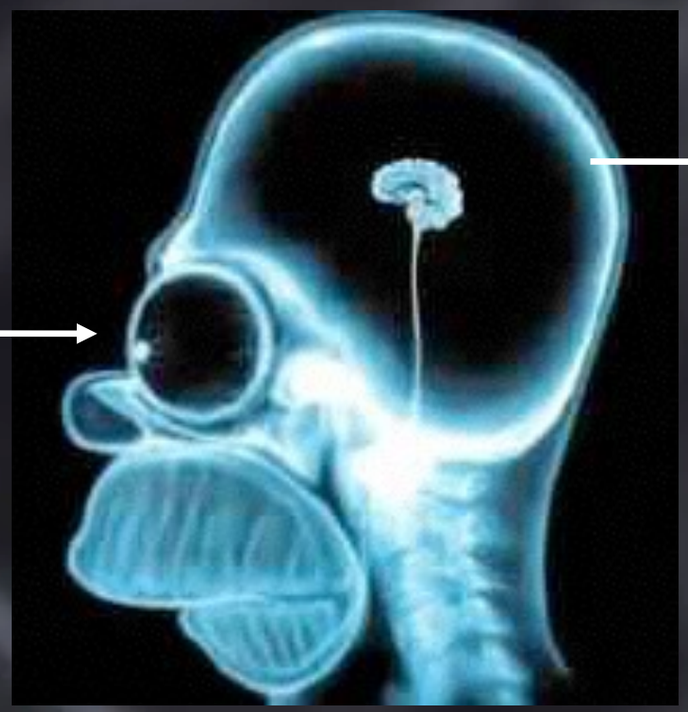
- Learning example: $(\mathbf{x}, y)$
- Task to solve: predict target y from input $\mathbf{x}$
  - classification: target is a class ID (from 0 to nb. of class - 1)
  - regression: target is a real number

Supervised Training Example

Entrée X

Sortie f(X)

Cible Y



six

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!

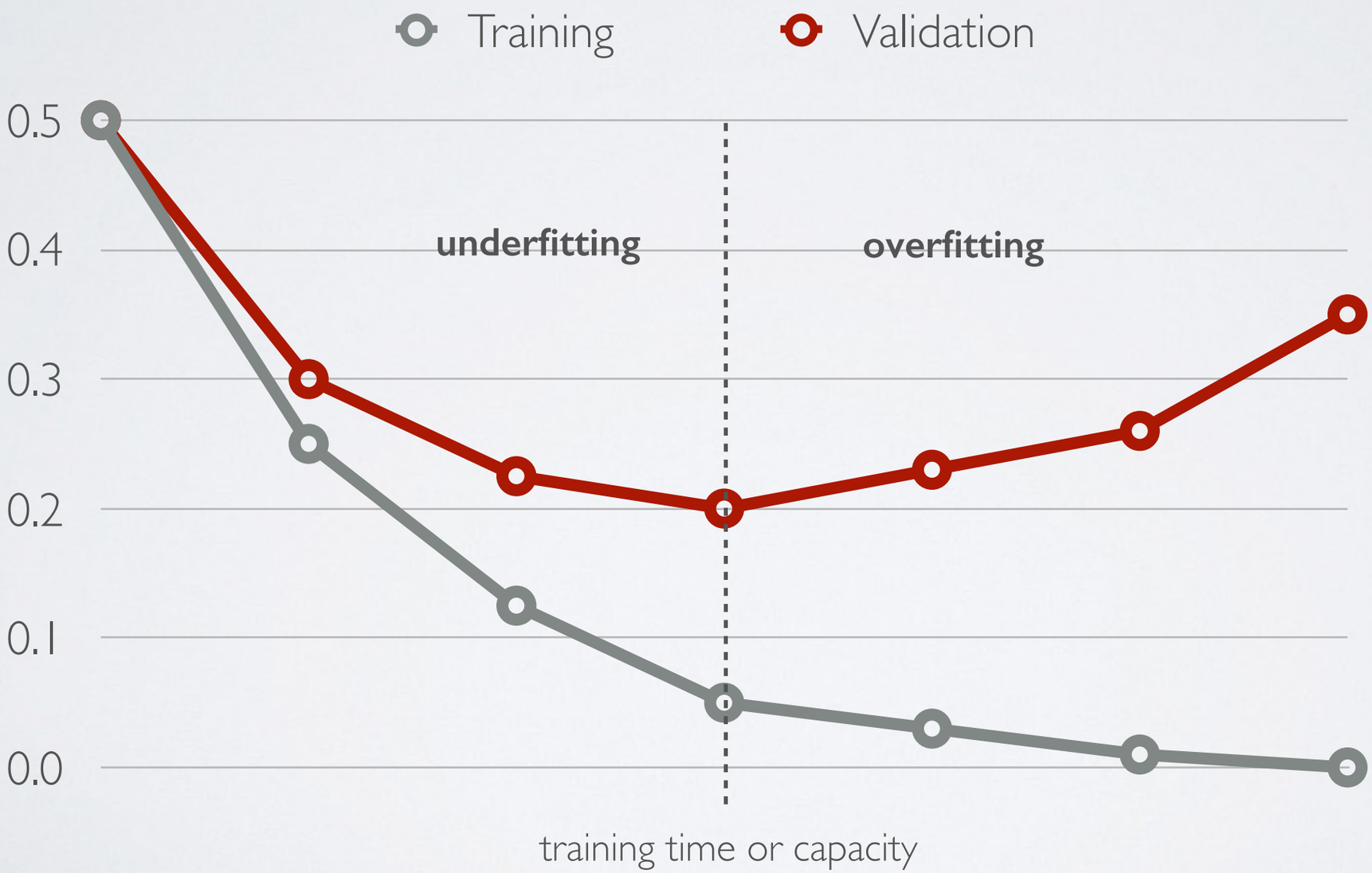
MACHINE LEARNING

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

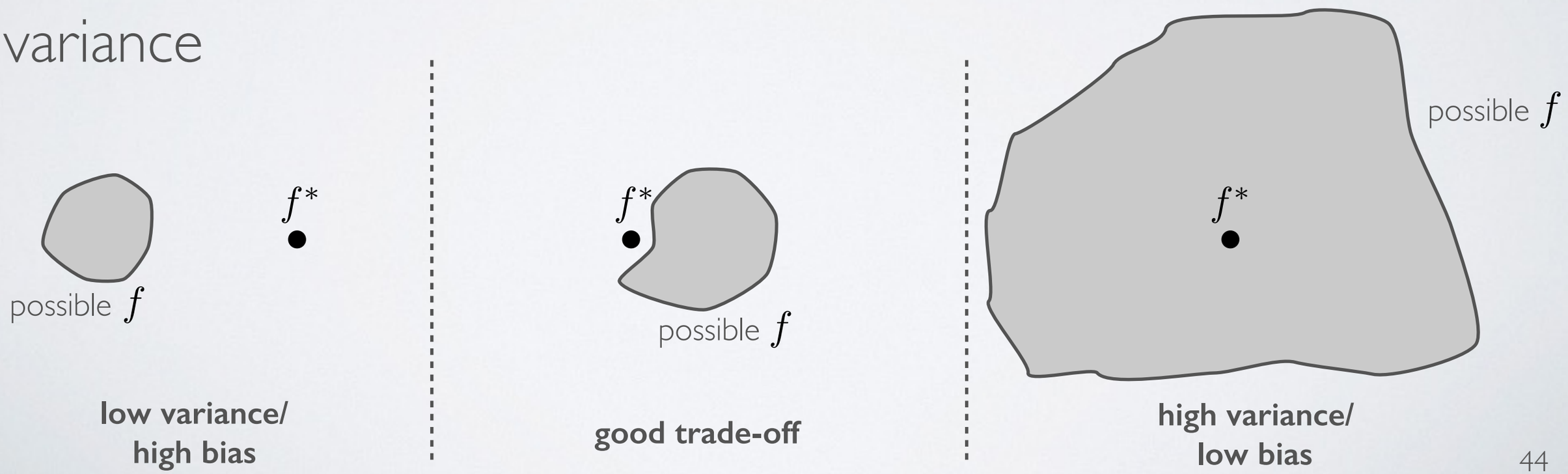
- If capacity increases:
  - training error will decrease
  - generalization error will increase or decrease
- If training time increases:
  - training error will decrease
  - generalization error will increase or decrease
- If training set size increases:
  - generalization error will decrease (or maybe 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)

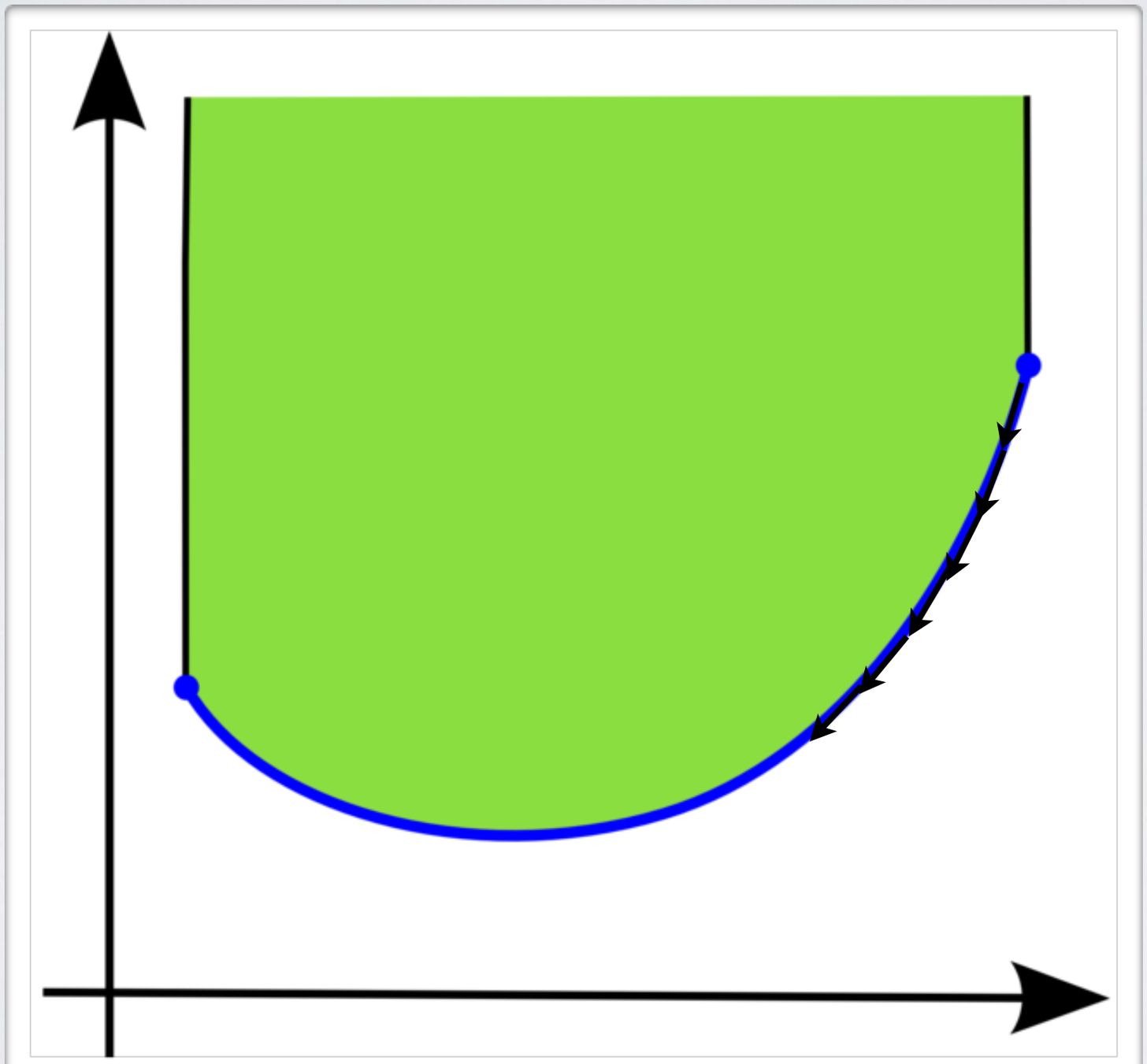
MACHINE LEARNING

Topics: gradient descent

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

MACHINE LEARNING

Topics: gradient descent



Descent direction

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

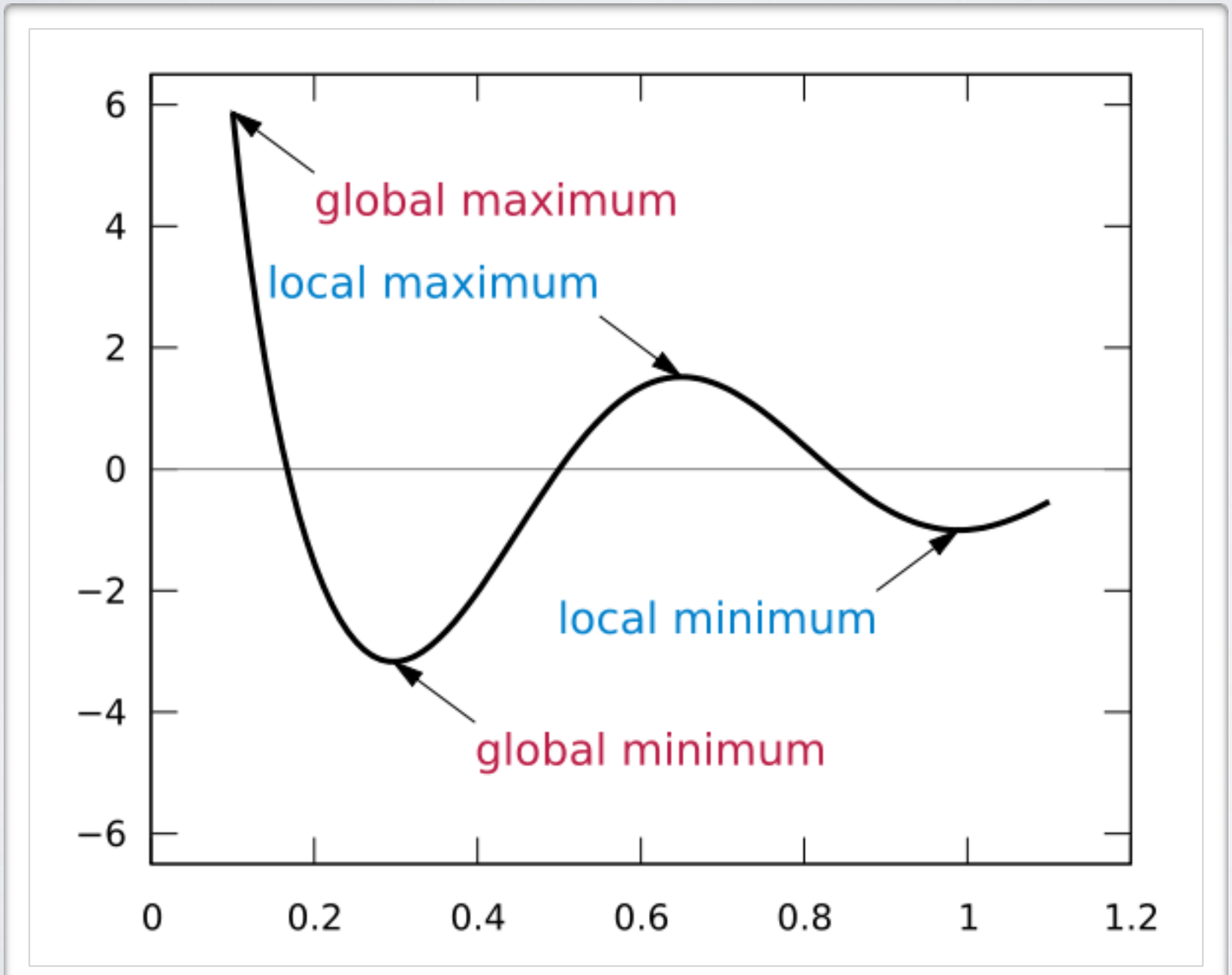
MACHINE LEARNING

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

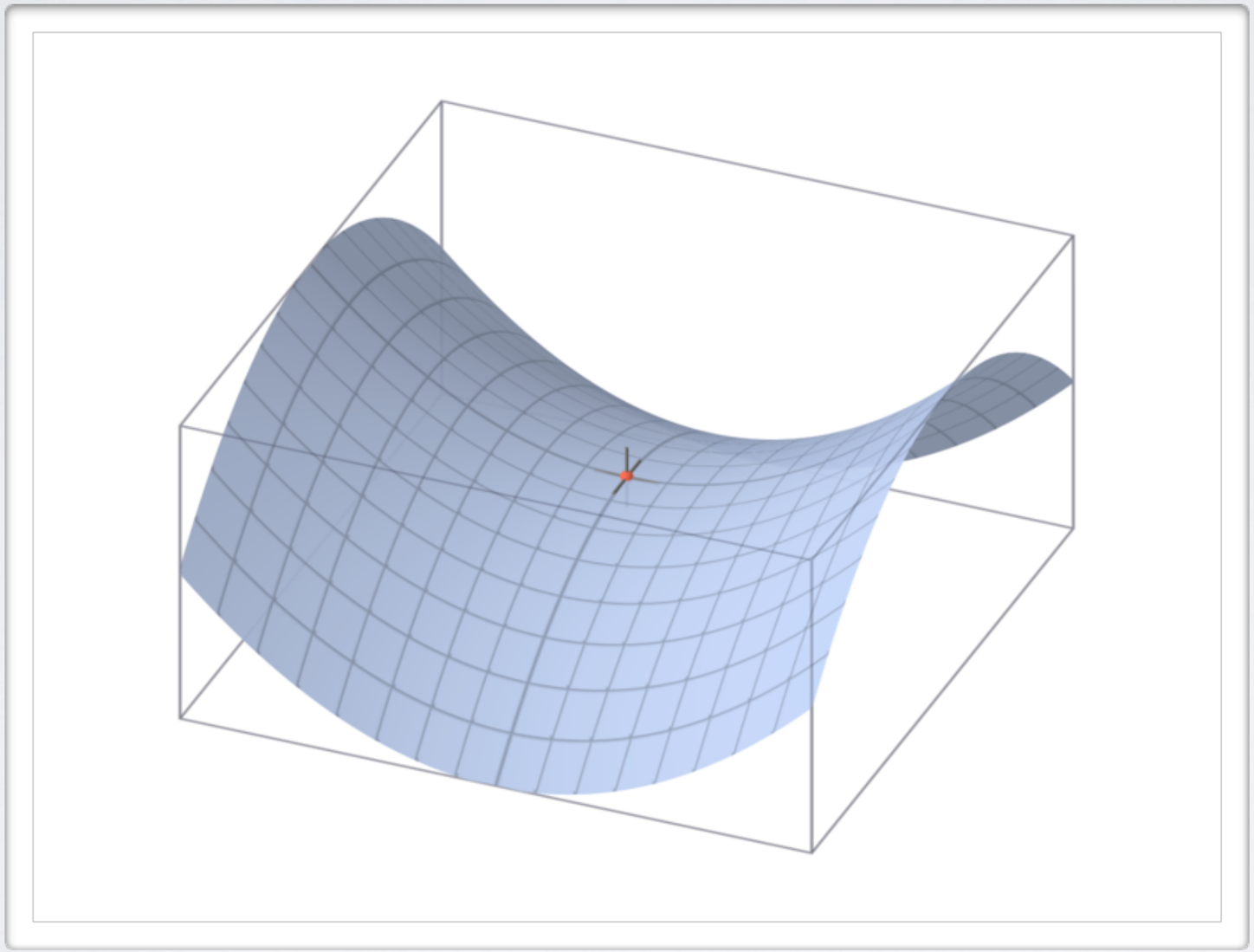
- Critical points: $\{\mathbf{x} \in \mathbb{R}^d \mid \nabla_{\mathbf{x}} f(\mathbf{x}) = 0\}$

- Curvature in direction $\mathbf{v}$: $\mathbf{v}^\top \nabla^2_{\mathbf{x}} f(\mathbf{x}) \mathbf{v}$

- Types of critical points:
  - local minima: $\mathbf{v}^\top \nabla^2_{\mathbf{x}} f(\mathbf{x}) \mathbf{v} > 0 \quad \forall \mathbf{v}$ (i.e. $\nabla^2_{\mathbf{x}} f(\mathbf{x})$ positive definite)
  - local maxima: $\mathbf{v}^\top \nabla^2_{\mathbf{x}} f(\mathbf{x}) \mathbf{v} < 0 \quad \forall \mathbf{v}$ (i.e. $\nabla^2_{\mathbf{x}} f(\mathbf{x})$ 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.