# Lecture 4

## Gradient Descent, Regularization, and Momentum

Bao-Jun Cai, 3/25/2026

Introduction to Algorithms for Data Science and Physics IMP@Fudan, 2026

Topics of this lecture:

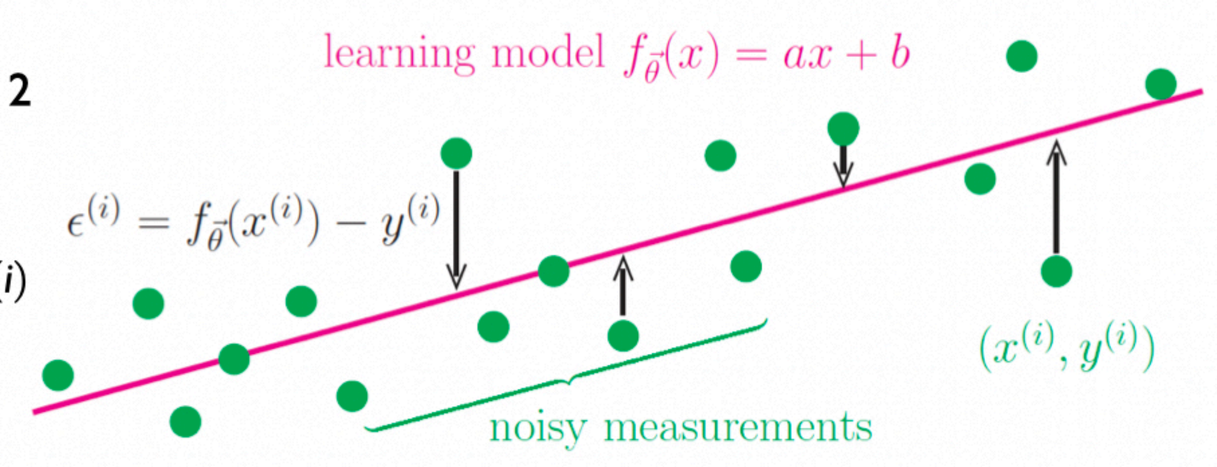
- gradient descent $\mathbf{x} \leftarrow \mathbf{x} - \epsilon \mathbf{g}$
- learning rate/search step size $\epsilon = \mathbf{g}^\top \mathbf{g} / \mathbf{g}^\top \mathbf{H} \mathbf{g}$
- singular behavior of Hessian, regularization $\det \mathbf{H} \approx 0, \mathbf{H} + \lambda \mathbf{I} \to \mathbf{H}$
- mass and momentum mechanism $\frac{d\mathbf{x}}{dt} = -\epsilon \nabla f(\mathbf{x}) + \cdots$
- Polyak&Nesterov momentum
- extensions of gradient descent (first-order in nature)

# Basic aim of optimization

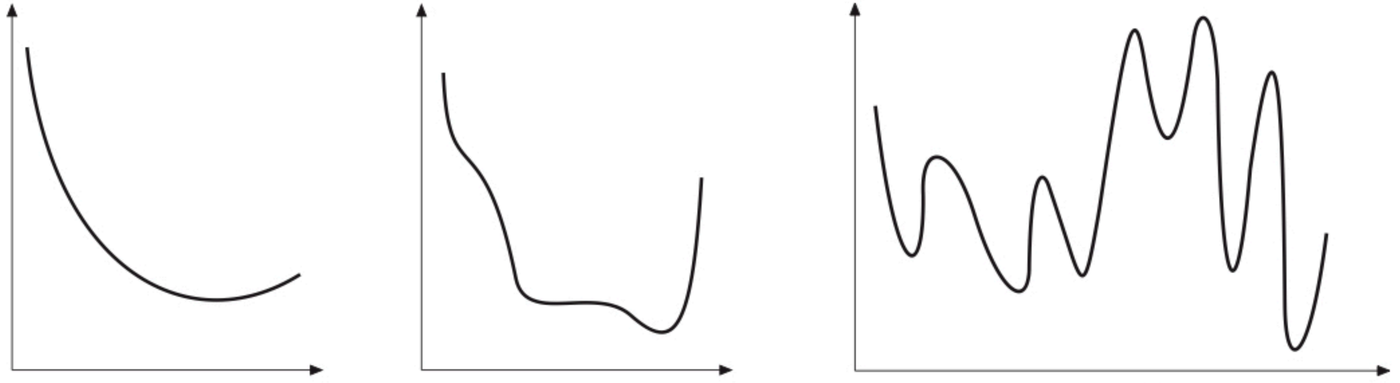
modern neural networks: $\dim \vec{\theta} \gg 10^{4\sim5}$

loss/cost/error function

$$J(\vec{\theta}) = J(a,b) = \frac{1}{2}\sum_{i=1}^{n}\left(f_{\vec{\theta}}(x^{(i)} - y^{(i)}\right)^2$$




* $J$ is a function of $a$ and $b$ instead of $x^{(i)}$ or $y^{(i)}$
* semi-postiveness: $J \geq 0$
* optimizing $J$ to be as small as possible
* $J$ in some cases is very complicated



data samples: $(x^{(i}, y^{(i)})$

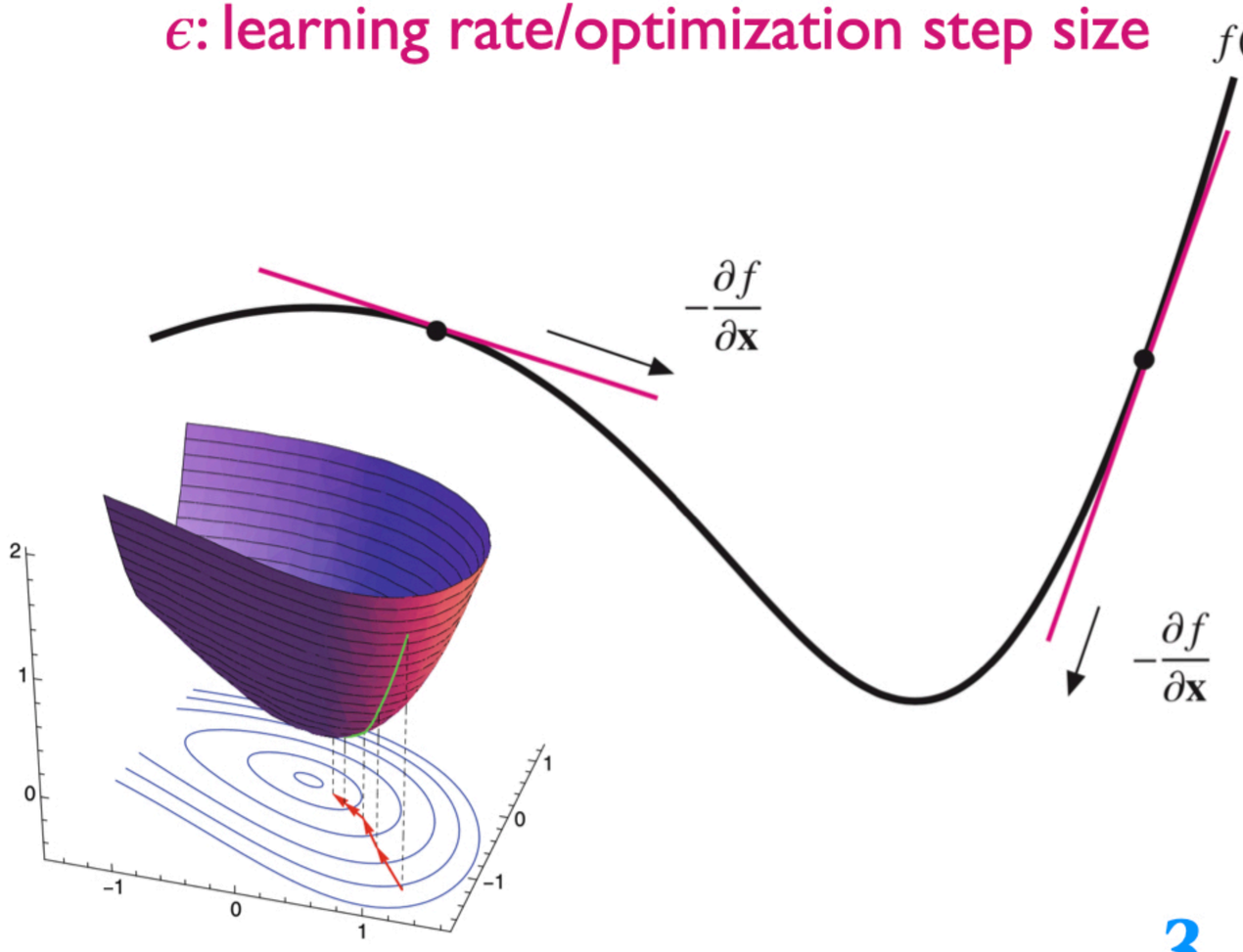
fitting model: $f_{\vec{\theta}}(x) = ax + b$

model parameters: $\vec{\theta} = (a, b)$

# How to select the search direction?

$\mathbf{x} \leftarrow \mathbf{x} + \epsilon \mathbf{d}$ (iterative idea)

$\epsilon$: learning rate/optimization step size



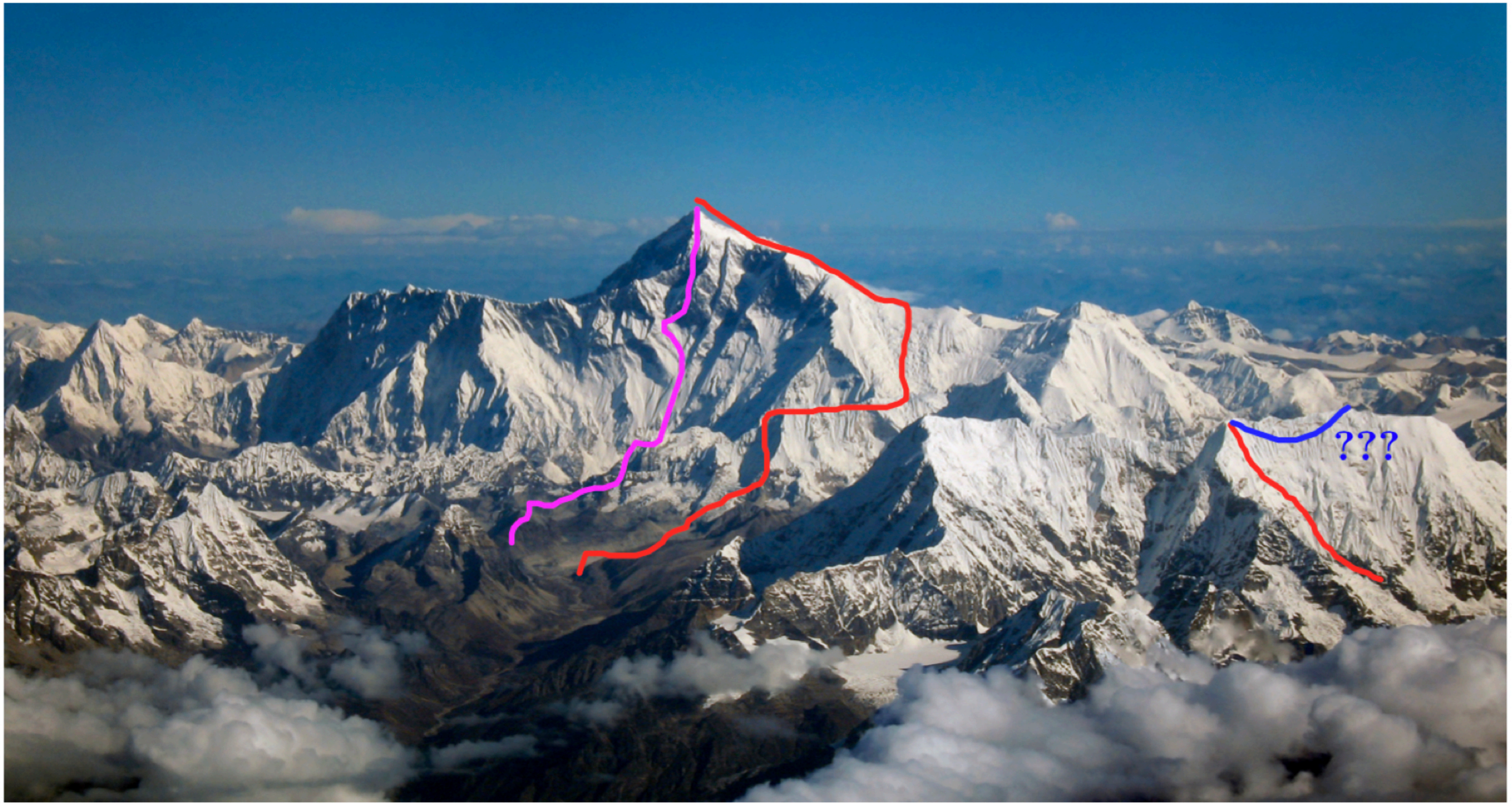

motivation/aim: make the function $f(\mathbf{x})$ as small as possible from $\mathbf{x} \leftarrow \mathbf{x} + \epsilon \mathbf{d}$

the direction with fastest decreasing on $f(\mathbf{x})$ is the negative gradient

Ex.: Prove it!

gradient-descent (GD) algorithm

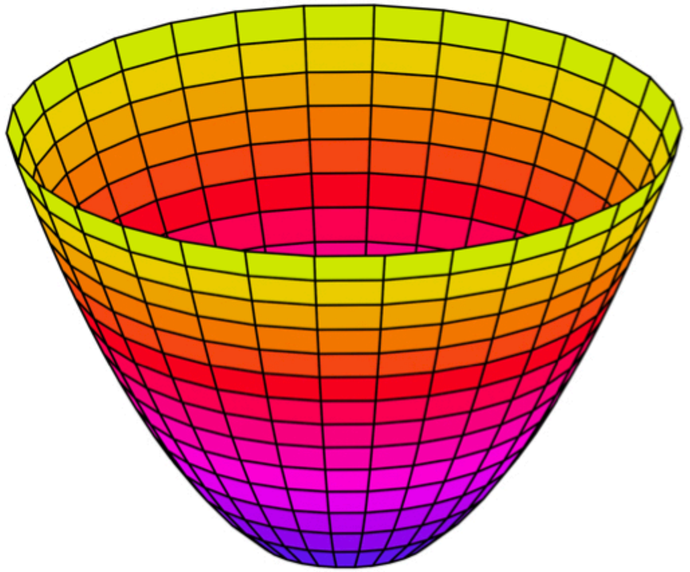
$$\mathbf{x} \leftarrow \mathbf{x} - \epsilon \mathbf{g}$$

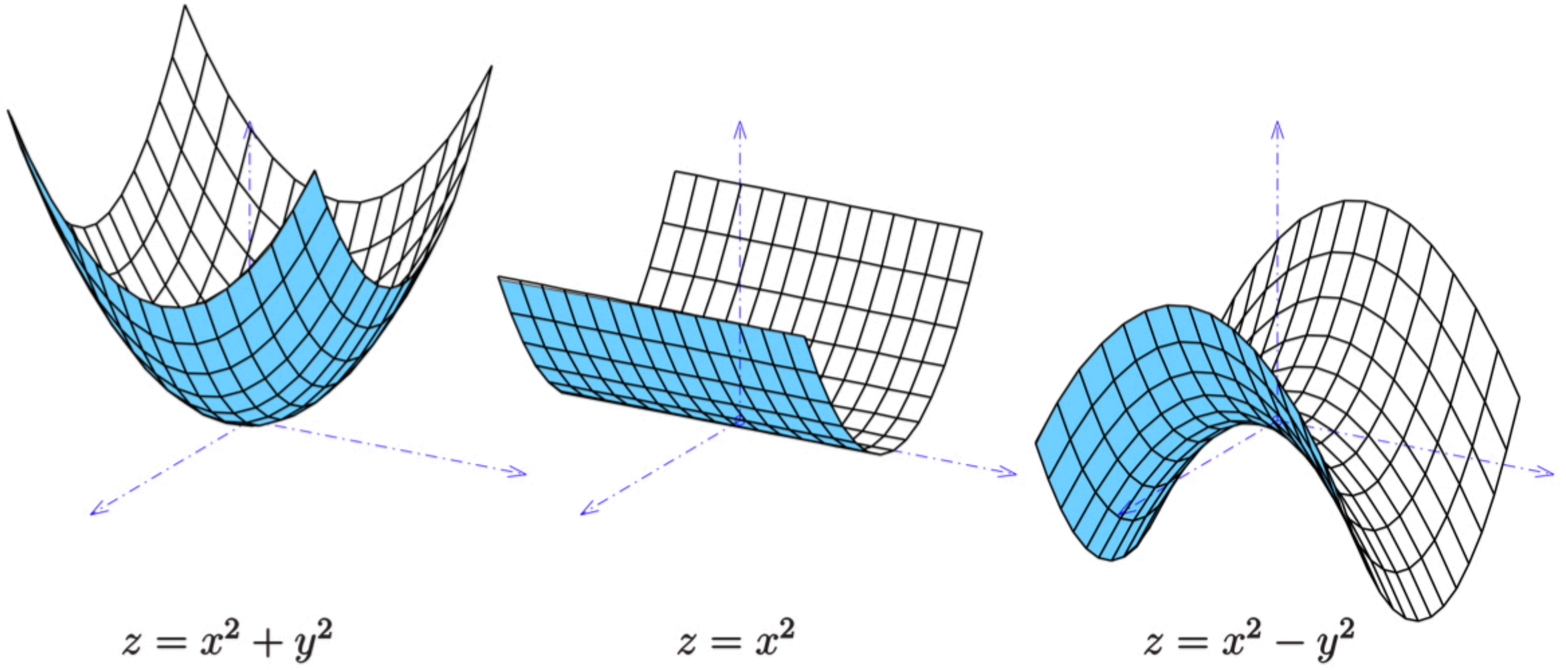


# High-dimensional objects/2D surface

## Tool: Taylor's expansion

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \delta\mathbf{x}^\top \mathbf{g}(\mathbf{x}_0) + \frac{1}{2}\delta\mathbf{x}^\top \mathbf{H}(\mathbf{x}_0)\delta\mathbf{x}$$





$z = x^2 + y^2$ $\quad z = x^2$ $\quad z = x^2 - y^2$

$$f(x, y) = x^2 + y^2$$

$$\mathbf{g} = \begin{pmatrix} 2x \\ 2y \end{pmatrix}, \mathbf{H} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} - \epsilon \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} (1 - 2\epsilon)x \\ (1 - 2\epsilon)y \end{pmatrix}$$

# Designing a good learning rate

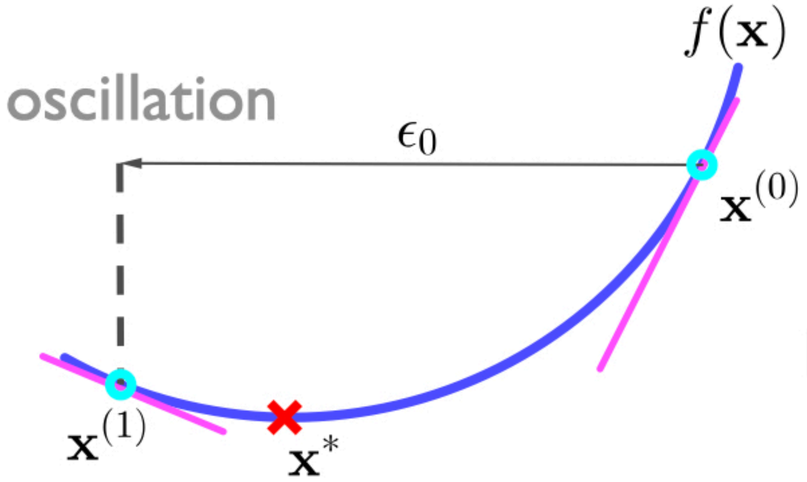
search direction: $\mathbf{x} \leftarrow \mathbf{x} - \epsilon \mathbf{g}$

$$K(\epsilon) \equiv f(\mathbf{x} \overbrace{-\epsilon \mathbf{g}}^{\text{small}}) \approx f(\mathbf{x}) - \epsilon \mathbf{g}^\top \mathbf{g} + \frac{1}{2}\epsilon^2 \mathbf{g}^\top \mathbf{H} \mathbf{g}$$

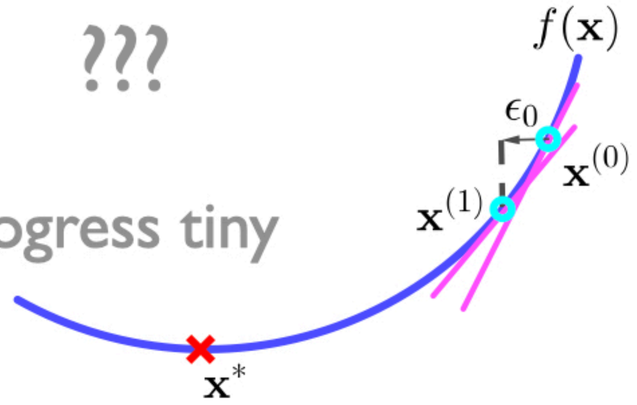
$f(\mathbf{x} - \epsilon \mathbf{g})$ as small as possible

$$\epsilon^* \equiv \text{argmin}_\epsilon f(\mathbf{x} - \epsilon \mathbf{g})$$

$$\epsilon^* \leftarrow \min K(\epsilon)$$




???




$$\epsilon^* = \frac{\mathbf{g}^\top \mathbf{g}}{\mathbf{g}^\top \mathbf{H} \mathbf{g}}$$

exact-line search

it is better to design the $\epsilon$ encapsulating information of the objective functions as much as possible (adaptively)

$$\mathbf{x} \leftarrow \mathbf{x} - \frac{\mathbf{g}^\top \mathbf{g}}{\mathbf{g}^\top \mathbf{H} \mathbf{g}} \mathbf{g}$$

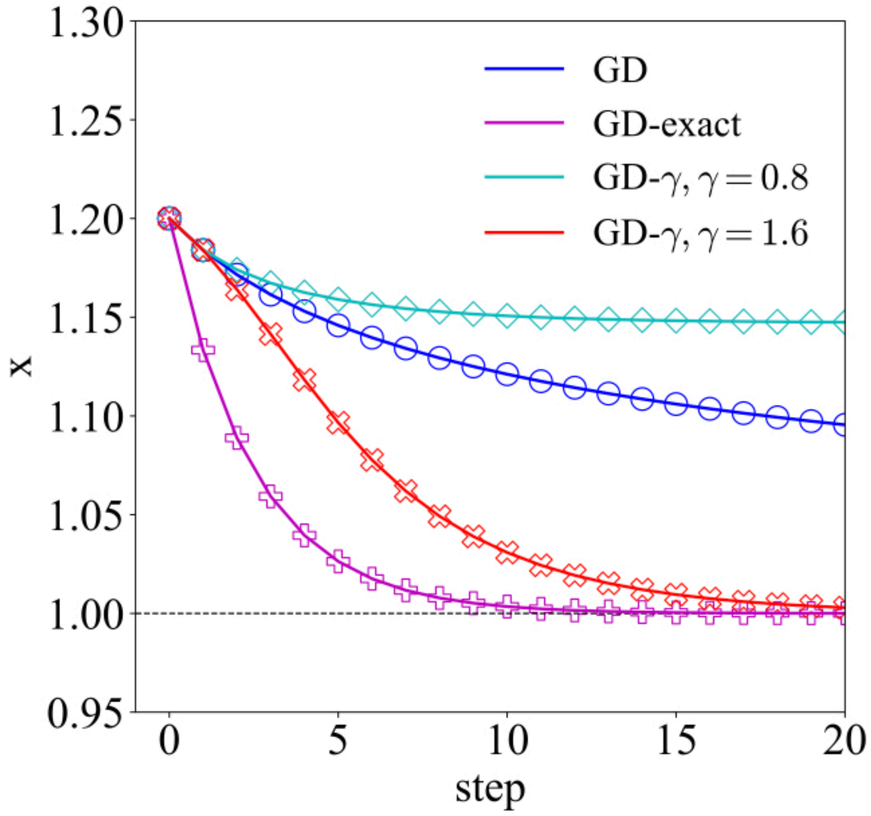
Ex.: Write down the result for 1D case.

# 1D example

$$f(x) = (x-1)^4$$

GD: $x \leftarrow x - 4\epsilon(x-1)^3$

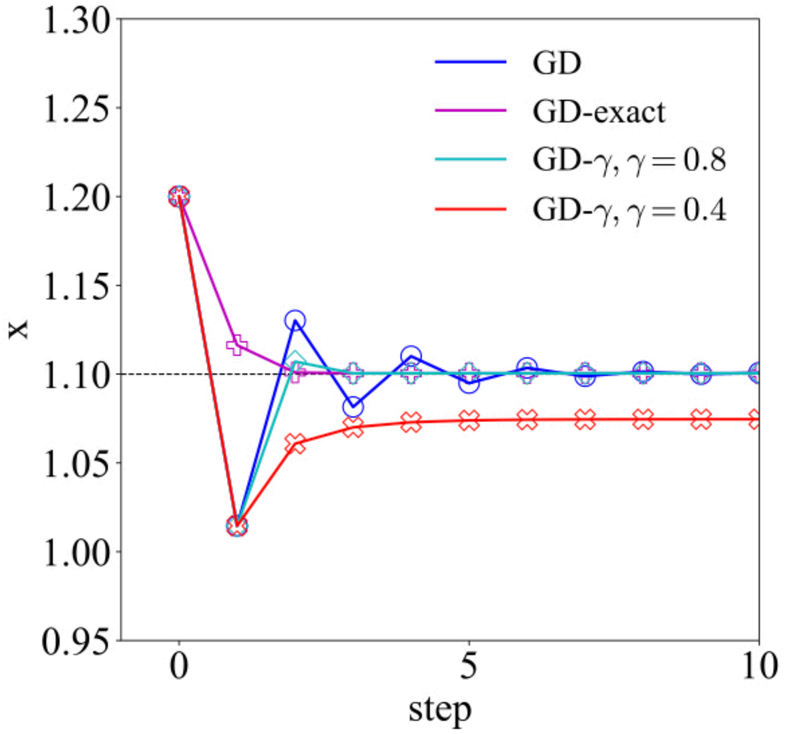
GD with exact-line search: $x \leftarrow \frac{2}{3}x + \frac{1}{3}$




decaying factor

$\epsilon \leftarrow \gamma\epsilon$

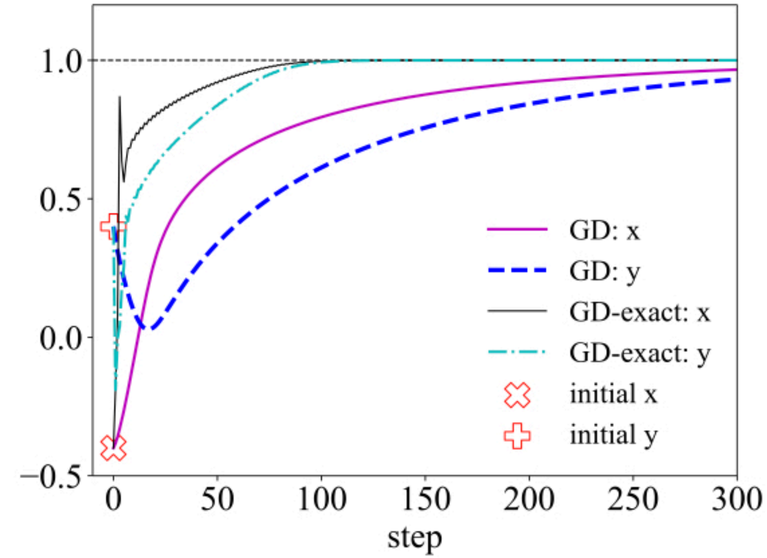
Ex.: Why?




$$f(x) = e^{x^2-2} - x$$

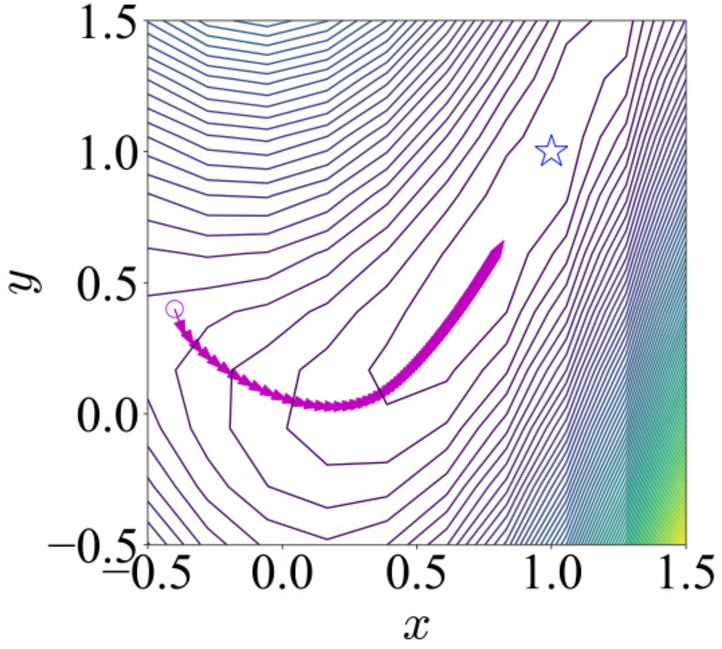
# 2D example

Rosenbrock: $f(\mathbf{x}) = (x-a)^2 + b(x^2-y)^2, b > 0, \mathbf{x}^* = \begin{pmatrix} a \\ a^2 \end{pmatrix}$



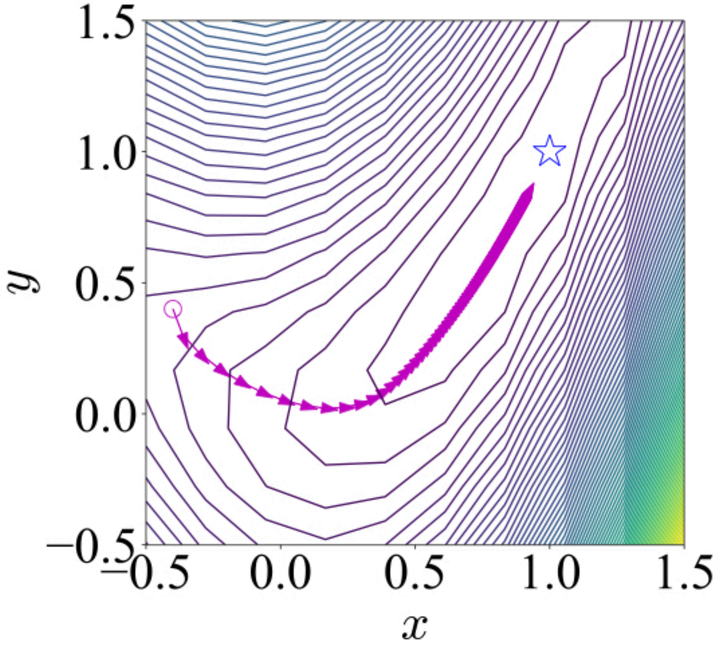

exact-line

$\epsilon = 0.02$

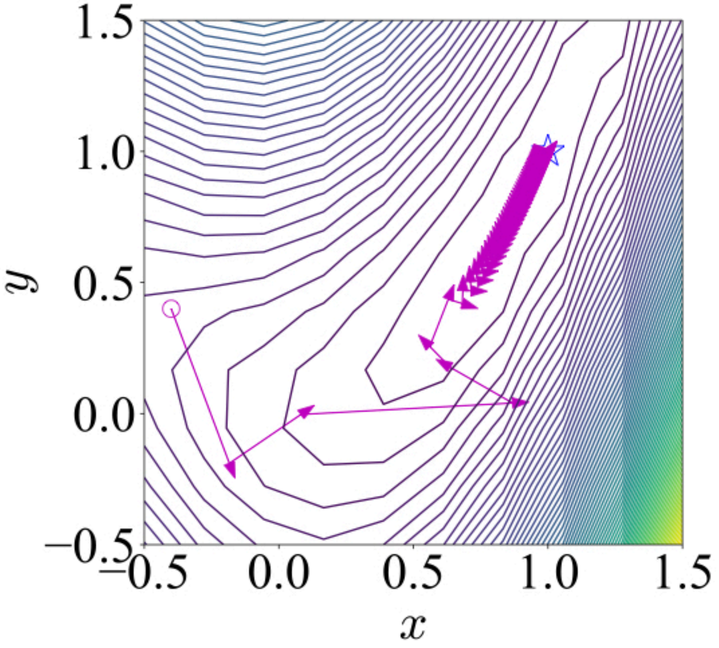



1042

$\epsilon = 0.04, \gamma = 0.999$




734




138

$|x^{(i+1)} - x^{(i)}| \lesssim 10^{-4}, |y^{(i+1)} - y^{(i)}| \lesssim 10^{-4}$

# Two realistic considerations

(1) number of data $\mathrm{m}$ is very large

$$J(\vec{\theta}) = J(a,b) = \frac{1}{2}\sum_{i=1}^{n}\left(f_{\vec{\theta}}(x^{(i)} - y^{(i)}\right)^2$$

heavy computational resources

stochastic gradient-descent (SGD)

$$m' \ll m : J(m' \to m)$$

(2) dimension of $\mathbf{x} \in \mathrm{R}^{\mathrm{d}}$ is very large

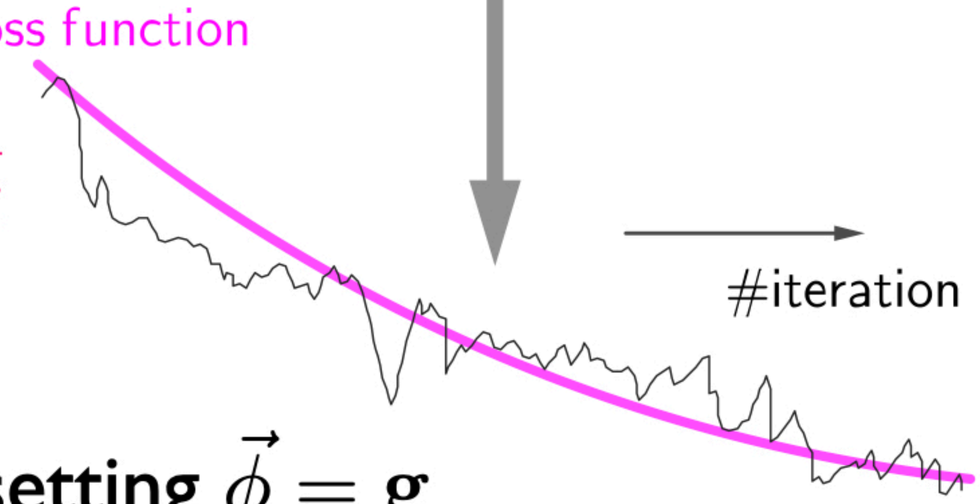
$\mathbf{H} \in \mathrm{R}^{d \times d}$ is hard to obtain

$$\epsilon^* = \frac{\mathbf{g}^\top \mathbf{g}}{\mathbf{g}^\top \mathbf{H} \mathbf{g}}$$

$$\mathbf{H}\vec{\phi} \approx \frac{\mathbf{g}(\mathbf{x} + \Delta\vec{\phi}) - \mathbf{g}(\mathbf{x} + \Delta\vec{\phi})}{2\Delta}$$

central scheme for differencing

by setting $\vec{\phi} = \mathbf{g}$



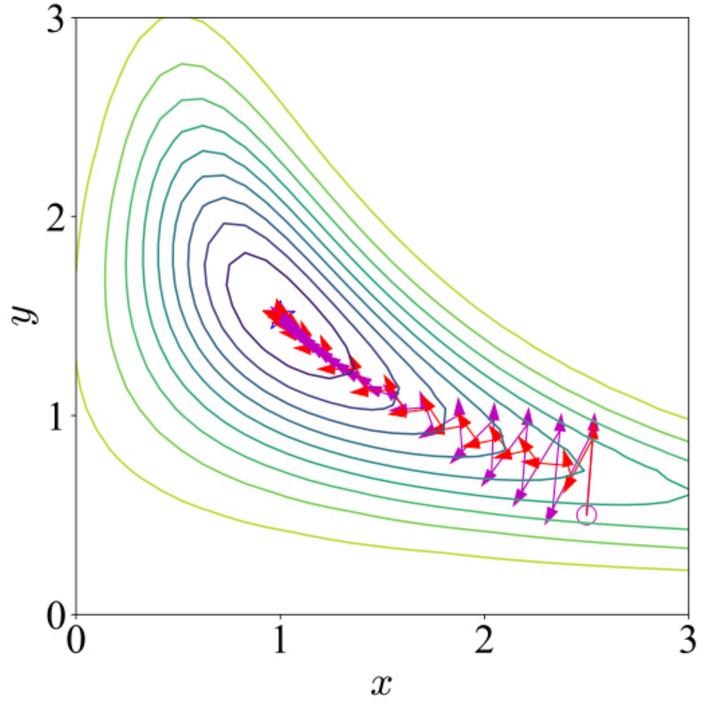
# Singular behavior of Hessian

$$f(\mathbf{x}) = \exp[-(xy-a)^2 - (y-a)^2], \quad a = 3/2.$$

$$f(x,y) \sim \mathcal{O}(P(x,y))$$

$$\det \mathbf{H} = 0$$

$$\mathbf{H} + \lambda \vec{\mathbf{I}} \to \mathbf{H}$$

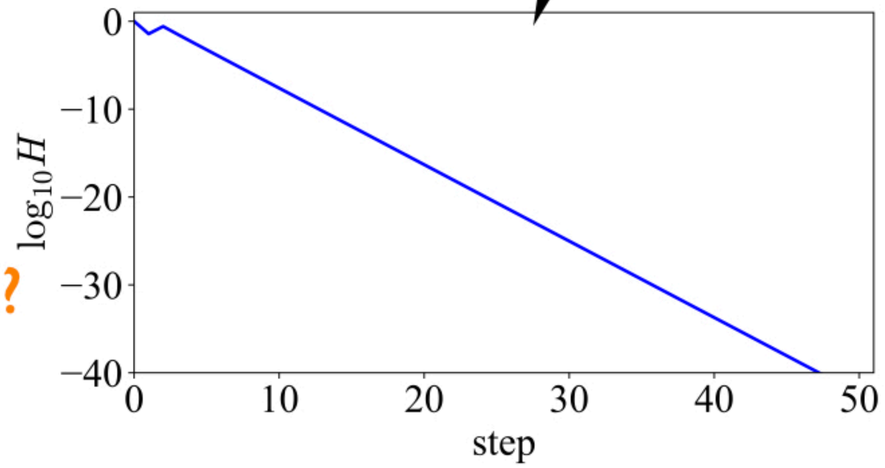



gradient-descent

$$\epsilon^* = \frac{\mathbf{g}^\top \mathbf{g}}{\mathbf{g}^\top \mathbf{H} \mathbf{g}} = \frac{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}{\frac{\partial^2 f}{\partial x^2}\left(\frac{\partial f}{\partial x}\right)^2 + 2\frac{\partial^2 f}{\partial x \partial y}\left(\frac{\partial f}{\partial x}\right)\left(\frac{\partial f}{\partial y}\right) + \frac{\partial^2 f}{\partial y^2}\left(\frac{\partial f}{\partial y}\right)^2} \sim \frac{P(x,y)}{f(x,y)}$$

$$\epsilon^* = \frac{\mathbf{g}^\top \mathbf{g}}{\mathbf{g}^\top \mathbf{H} \mathbf{g}} \to \frac{\mathbf{g}^\top \mathbf{g}}{\mathbf{g}^\top (\mathbf{H} + \lambda \vec{\mathbf{I}}) \mathbf{g}} \approx \frac{\mathbf{g}^\top \mathbf{g}}{\mathbf{g}^\top \lambda \vec{\mathbf{I}} \mathbf{g}} = \frac{1}{\lambda}$$

regularization




Ex.: What happens if $\det \mathbf{H} < 0$?

# Mass and momentum

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \epsilon \nabla f(\mathbf{x}^{(i)})$$

gradient descent

continuous version

$$\frac{d\mathbf{x}}{dt} = -\epsilon \nabla f(\mathbf{x})$$

massless particle

$$\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)} = -\epsilon \nabla f(\mathbf{x}^{(i)}) + p\left(\mathbf{x}^{(i)} - \mathbf{x}^{(i-1)}\right)$$

momentum term

discrete version

EOM of damped oscillator with mass $m$

"force"

$$m\frac{d^2\mathbf{x}}{dt^2} + \mu\frac{d\mathbf{x}}{dt} = -\nabla f(\mathbf{x})$$

$$\mathbf{x}(t+\delta t) - \mathbf{x}(t) = -\frac{\delta t^2}{m+\mu\delta t}\nabla f(\mathbf{x}) + \frac{m}{m+\mu\delta t}[\mathbf{x}(t) - \mathbf{x}(t-\delta t)]$$

$$= -\epsilon f(\mathbf{x}) + p[\mathbf{x}(t) - \mathbf{x}(x-\delta t)]$$

momentum parameter

$$\epsilon = \frac{\delta t^2}{m+\mu\delta t}, \quad p = \frac{m}{m+\mu\delta t} = \frac{1}{1+\mu\delta t/m}$$

stability condition

$-1<p<1$

$p \lesssim 1$ (pratical)

$$\frac{d^2\mathbf{x}}{dt^2} \approx \frac{\mathbf{x}(t+\delta t) + \mathbf{x}(t-\delta t) - 2\mathbf{x}(t)}{\delta t^2}$$

$$\frac{d\mathbf{x}}{dt} \approx \frac{\mathbf{x}(t+\delta t) - \mathbf{x}(t)}{\delta t}$$

# Momentum mechanism: continued

under force: $-\epsilon\nabla f(\mathbf{x}^{(i)})$

particle moves with $p\mathbf{v}^{(i)}$

Polyak momentum

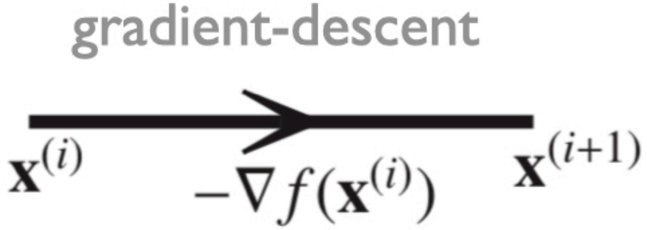
$$\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)} = -\epsilon\nabla f(\mathbf{x}^{(i)}) + p\left(\mathbf{x}^{(i)} - \mathbf{x}^{(i-1)}\right)$$

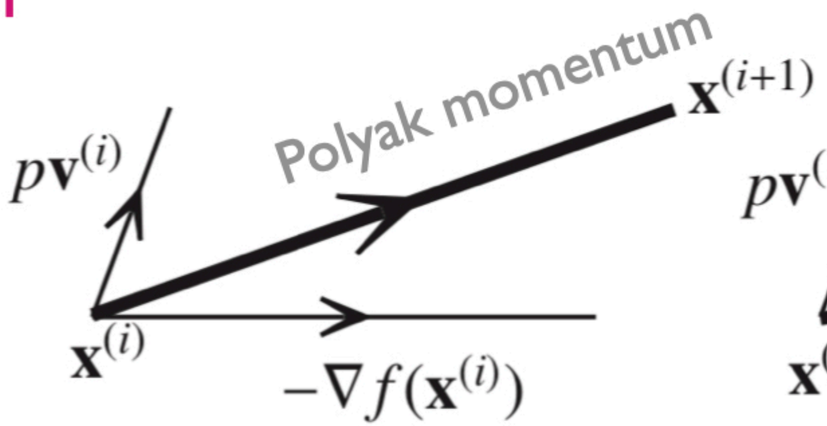
$$\sim p\mathbf{v}^{(i)}$$

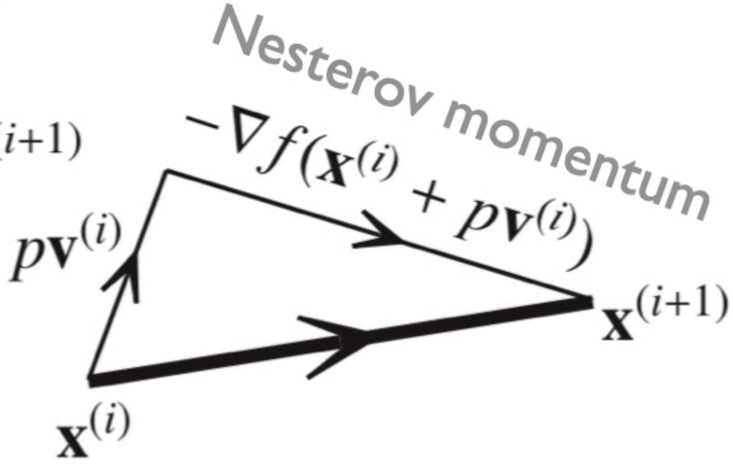
$$\mathbf{v}^{(i+1)} = p\mathbf{v}^{(i)} - \epsilon\nabla f(\mathbf{x}^{(i)}),\ \mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \mathbf{v}^{(i+1)}$$

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \epsilon\nabla f\left(\mathbf{x}^{(i)} + p\mathbf{v}^{(i)}\right) + p\left(\mathbf{x}^{(i)} - \mathbf{x}^{(i-1)}\right)$$

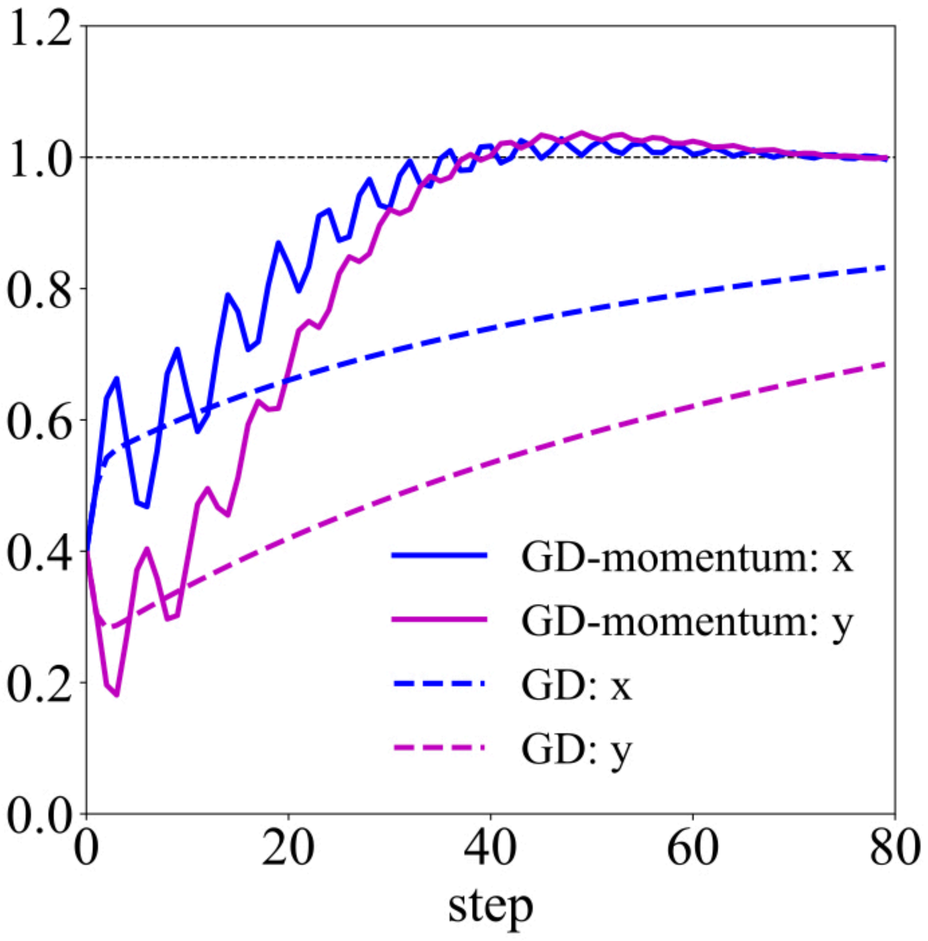
Nesterov momentum

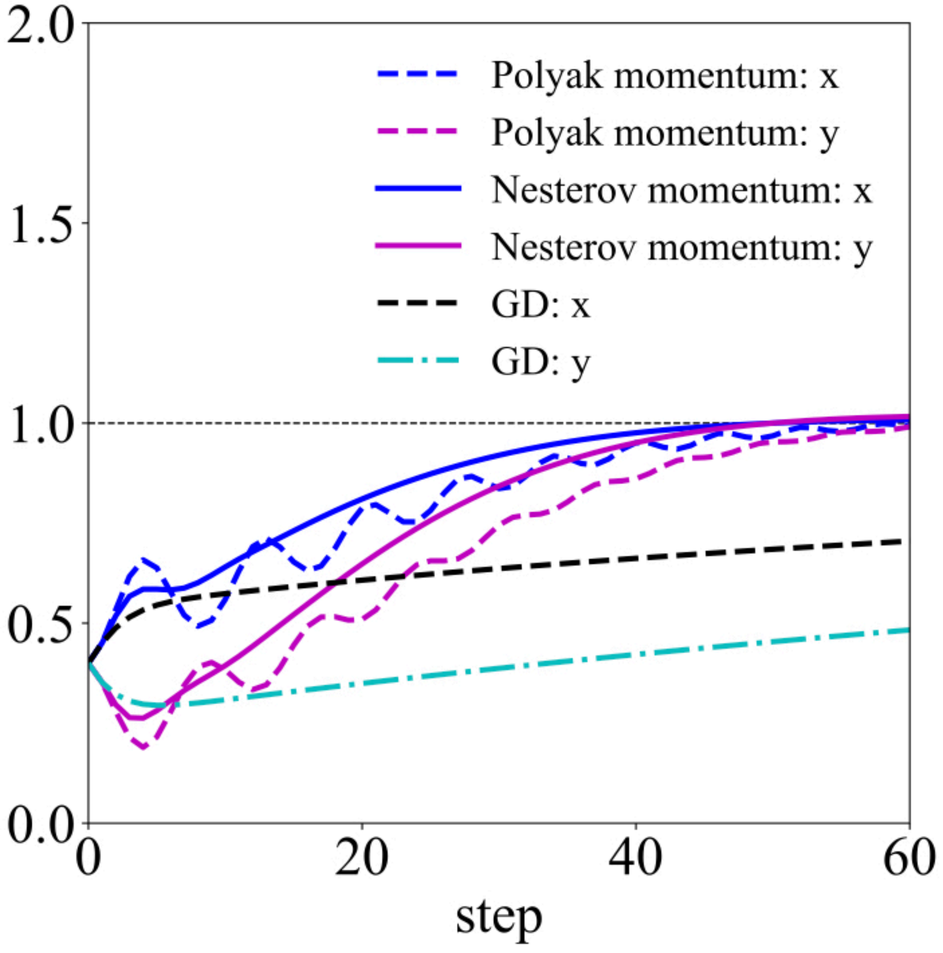


# Example: Rosenbrock function, $p$=0.9




Nesterov momentum is more stable!

# Extensions (1st): learning rate or search direction

**AdaGrad** accumulating the gradient to avoid the $\epsilon$ to be too large

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - \overbrace{\frac{\epsilon}{\delta + \sqrt{\mathbf{s}^{(i+1)}}}}^{\text{effective } \widetilde{\epsilon}_i} \odot \mathbf{g}_i$$

$$\mathbf{s}^{(i+1)} = \mathbf{s}^{(i)} + \mathbf{g}_i \odot \mathbf{g}_i$$

RMSProp

$$\mathbf{s}^{(i+1)} = \gamma \mathbf{s}^{(i)} + \mathbf{g}_i \odot \mathbf{g}_i$$

**hyper-gradient**

$$\frac{\partial f(\mathbf{x}^{(i)})}{\partial \epsilon} = \mathbf{g}_i^\top \frac{\partial}{\partial \epsilon} \left( \mathbf{x}^{(i-1)} - \epsilon \mathbf{g}_{i-1} \right)$$

$$= -\mathbf{g}_i^\top \mathbf{g}_{i-1}$$

**g: gradient**

improved learning rate

$$\epsilon_{i+1} = \epsilon_i - \mu \frac{\partial f(\mathbf{x}^{(i)})}{\partial \epsilon} = \epsilon_{i+1} = \epsilon_i + \mu \mathbf{g}_i^\top \mathbf{g}_{i-1}$$

Adadelta, Adam, Nadam, AMSGrad, …