ENM 531: Data-driven Modeling and Probabilistic Scientific Computing

Lecture #3: Statistical estimation

Paris Perdikaris
January 25, 2022

Penn
University *of* Pennsylvania

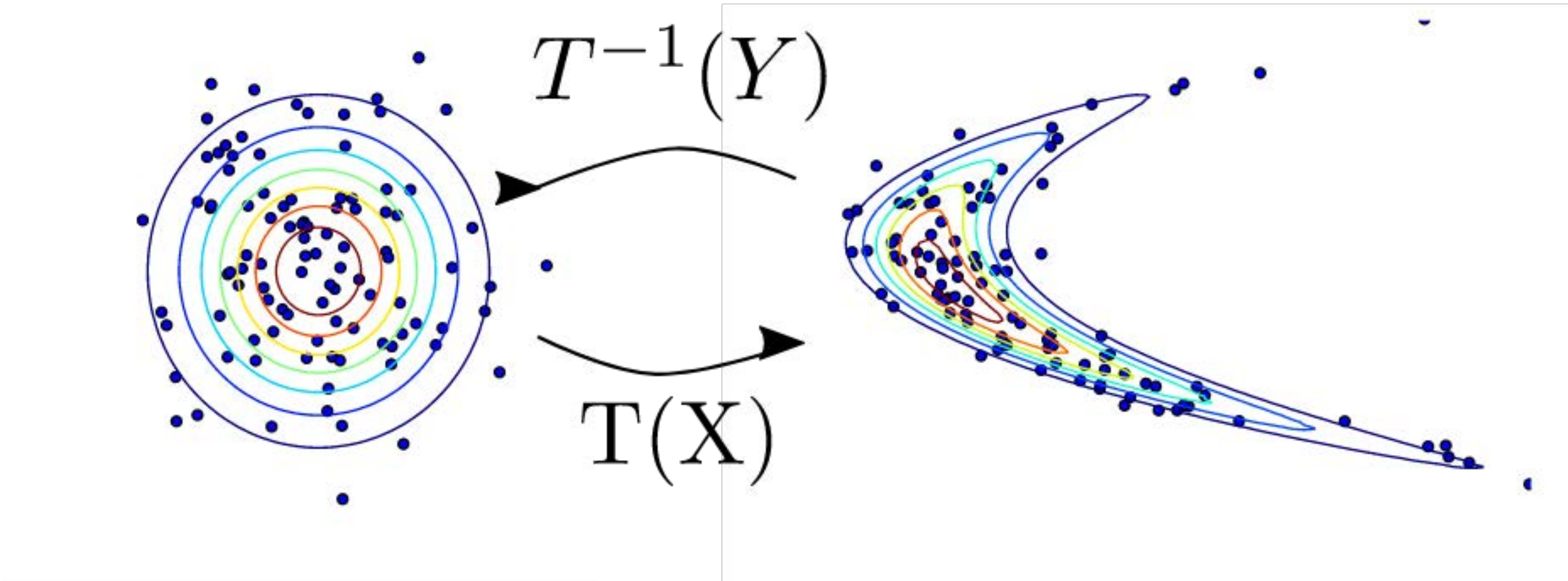
Maximum likelihood estimation

$$\theta_{\mathrm{MLE}} = \arg\max_{\theta \in \Theta} p(\mathcal{D}|\theta)$$

Maximum a-posteriori estimation

$$\theta_{\mathrm{MAP}} = \arg\max_{\theta \in \Theta} p(\theta | \mathcal{D})$$

Transformations



Objectives

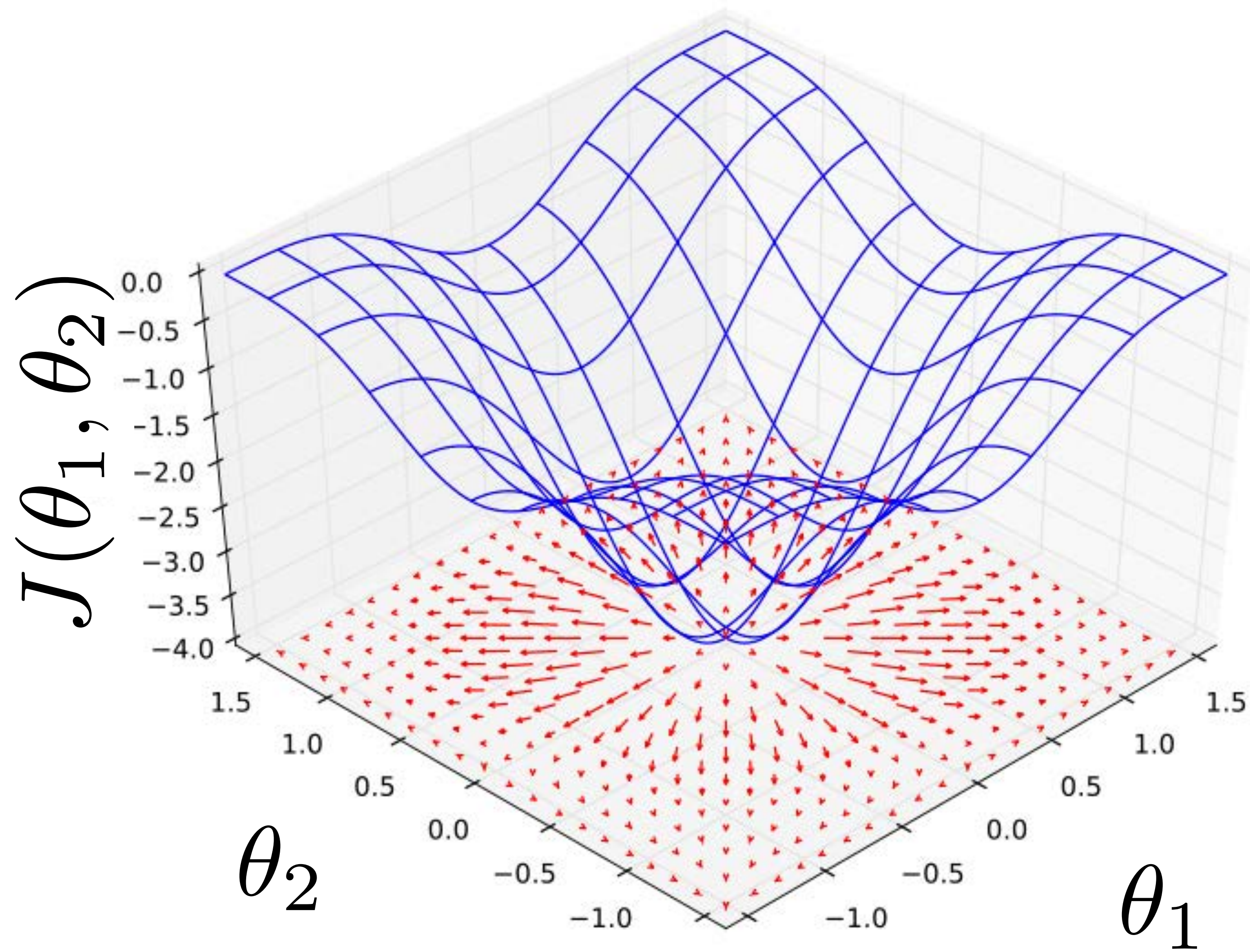
At its core, machine learning is all about integration (e.g., computing expectations, etc.) and ***optimization***. Today we'll revisit some basic concepts in optimization, and introduce them in the context of training machine learning algorithms.

Specifically, we'll cover:

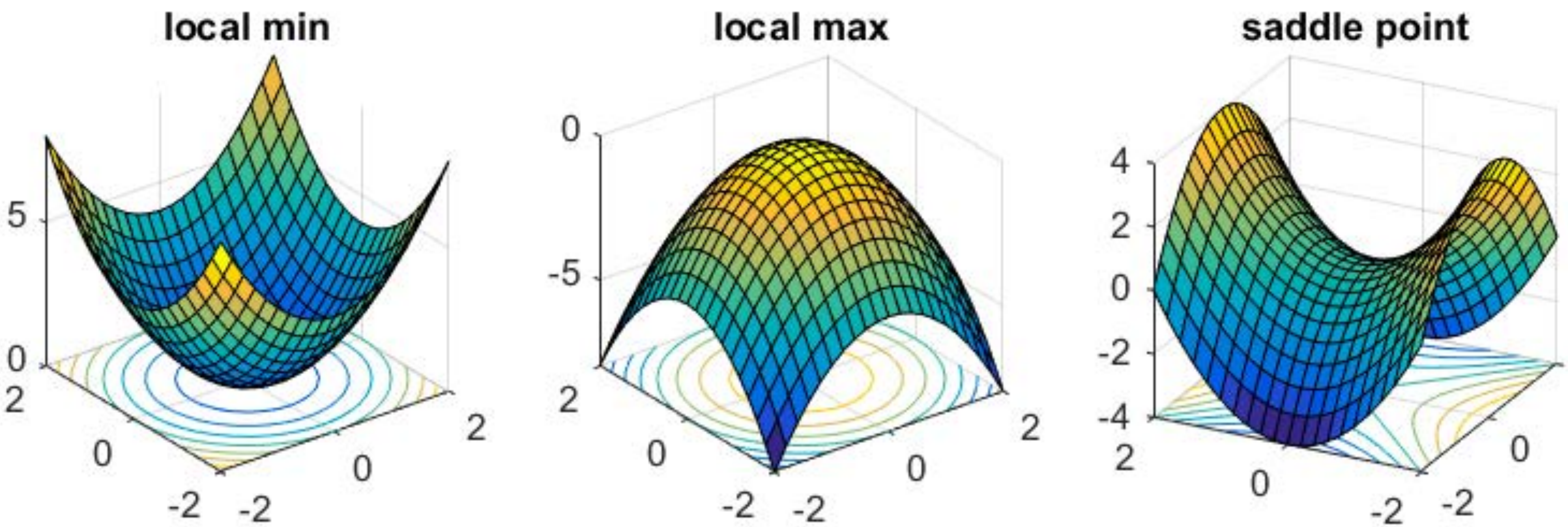
- The definition of gradients and Hessians.
- The gradient descent algorithm.
- Newton's algorithm.
- Applications to linear regression.
- Stochastic gradient descent.
- Modern variants of stochastic gradient descent.

Gradients

$$\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_1} \\ \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_2} \\ \vdots \\ \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_n} \end{bmatrix}$$



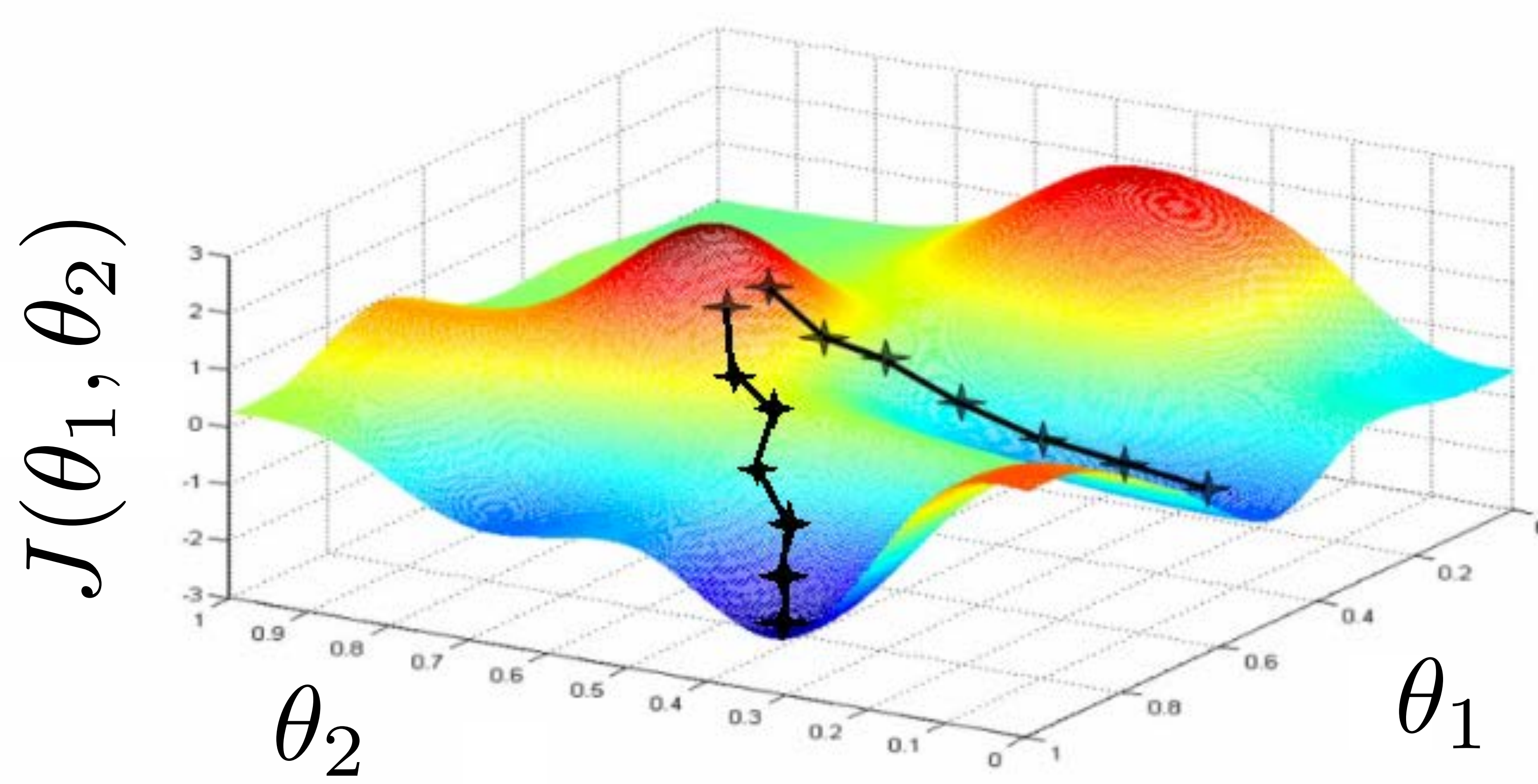
Minima, maxima, and saddle points



Gradient descent

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} J(\boldsymbol{\theta})$$

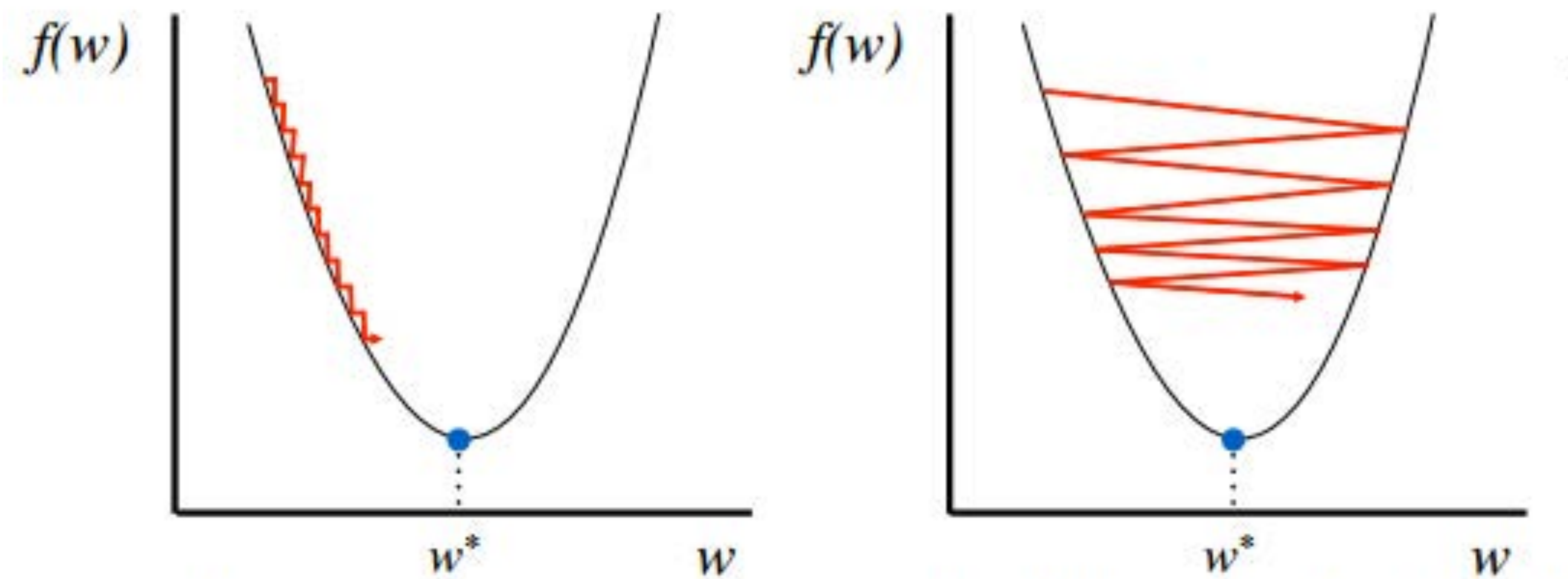
$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \eta\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$



Gradient descent

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \eta \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Effect of the learning rate



Too small: converge very slowly

Too big: overshoot and even diverge

Hessian

$$\nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \theta_1^2} & \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_n} \\ \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \theta_2^2} & \cdots & \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \theta_d \partial \theta_1} & \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \theta_d \partial \theta_2} & \cdots & \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \theta_d^2} \end{bmatrix}$$

Gradient descent vs Newton's method