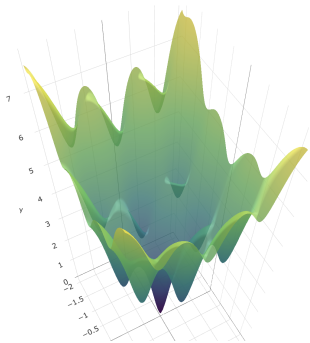
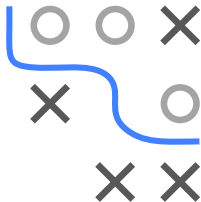


Optimization in Machine Learning

First order methods

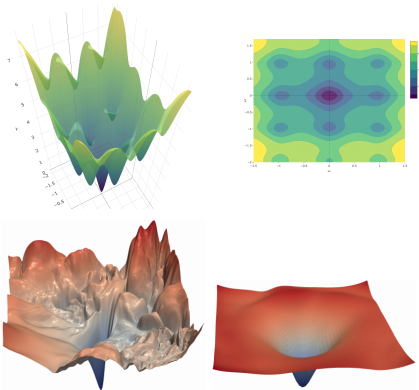
GD – Multimodality and Saddle points



Learning goals

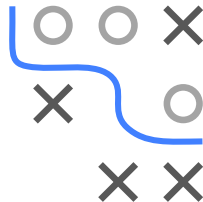
- Multimodality, GD result can be arbitrarily bad
- Saddle points, major problem in NN error landscapes, GD can get stuck or slow crawling

UNIMODAL VS. MULTIMODAL LOSS SURFACES



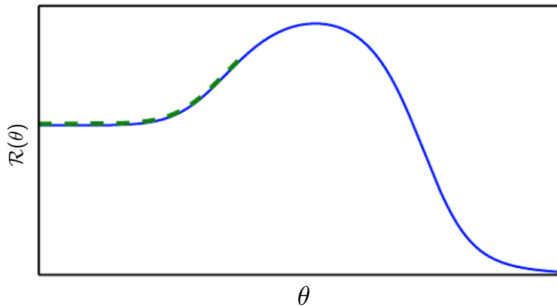
► [Click for source](#)

- Snippet of a loss surface with many local optima
- In deep learning, we often find multimodal loss surfaces.
- **Left:** Multimodal loss surface.
- **Right:** (Nearly) unimodal loss surface.



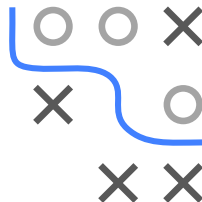
GD: ONLY LOCALLY OPTIMAL MOVES

- GD makes only **locally** optimal moves
- It may move away from the global optimum



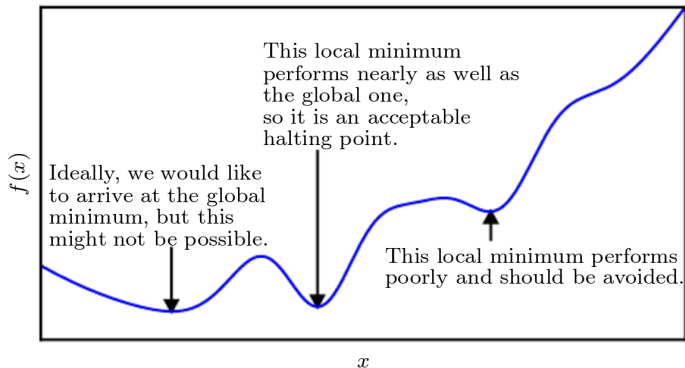
► [Click for source](#)

- Initialization on “wrong” side of the hill results in weak performance
- In higher dimensions, GD may move around the hill (potentially at the cost of longer trajectory and time to convergence)

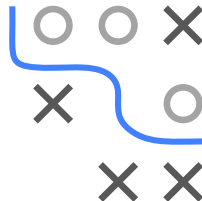


LOCAL MINIMA

- **In practice:** Only local minima with high value compared to global minimum are problematic.

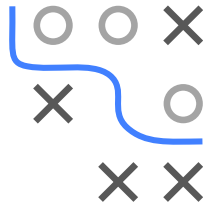
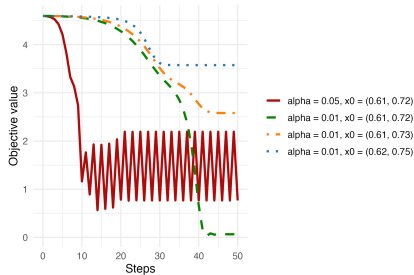
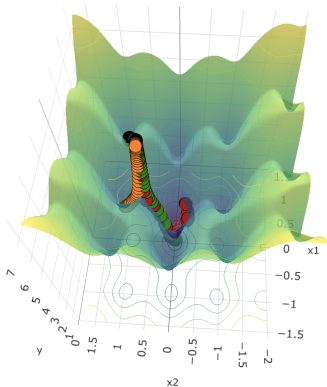


► [Click for source](#)



LOCAL MINIMA: SENSITIVITY

- Small differences in starting point or step size can lead to huge differences in the reached minimum or even to non-convergence



- (Non-)Converging gradient descent for Ackley function

GD AT SADDLE POINTS

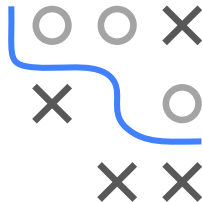
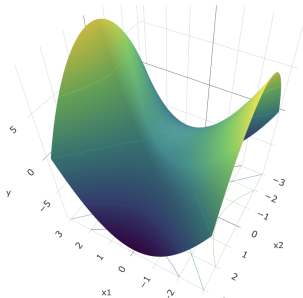
Example:

$$f(x_1, x_2) = x_1^2 - x_2^2$$

$$\nabla f(x_1, x_2) = (2x_1, -2x_2)^T$$

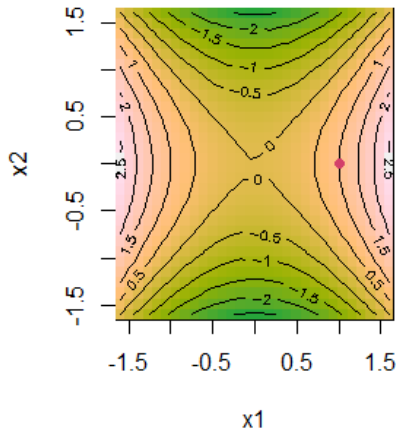
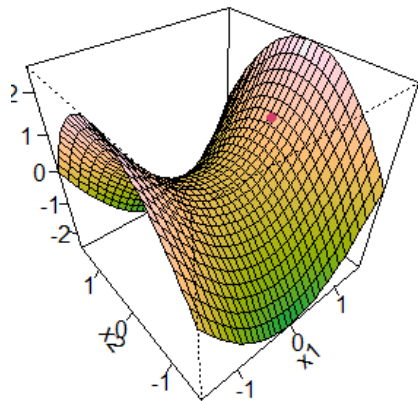
$$H = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$$

- Along x_1 , curvature is positive ($\lambda_1 = 2 > 0$).
- Along x_2 , curvature is negative ($\lambda_2 = -2 < 0$).

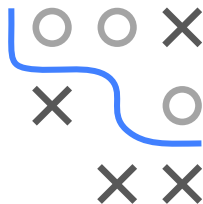


EXAMPLE: SADDLE POINT WITH GD

- How do saddle points impair optimization?
- Gradient-based algorithms **might** get stuck in saddle points

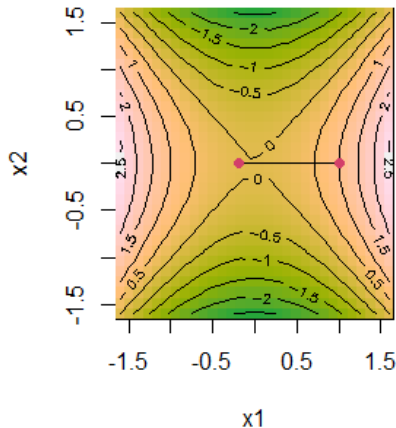
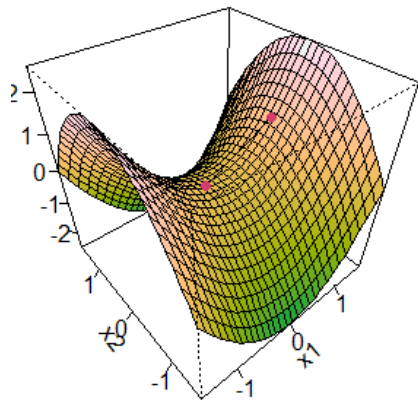


Red dot: Starting location

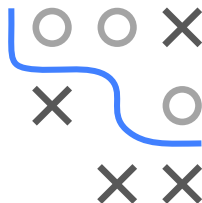


EXAMPLE: SADDLE POINT WITH GD

- How do saddle points impair optimization?
- Gradient-based algorithms **might** get stuck in saddle points

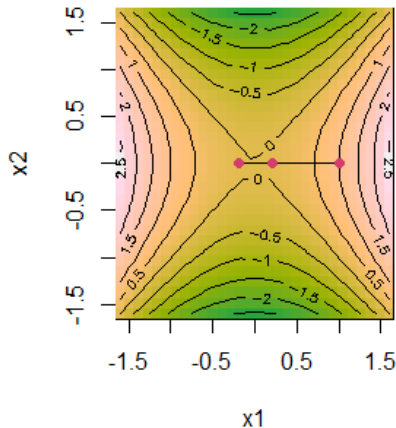
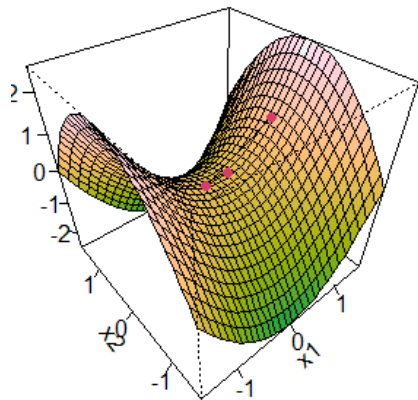


Step 1 ...

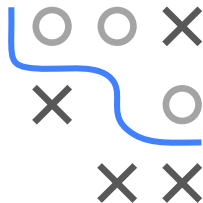


EXAMPLE: SADDLE POINT WITH GD

- How do saddle points impair optimization?
- Gradient-based algorithms **might** get stuck in saddle points

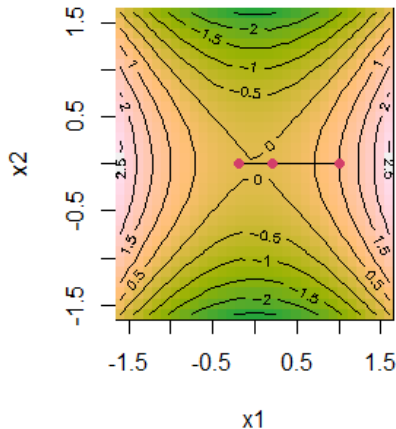
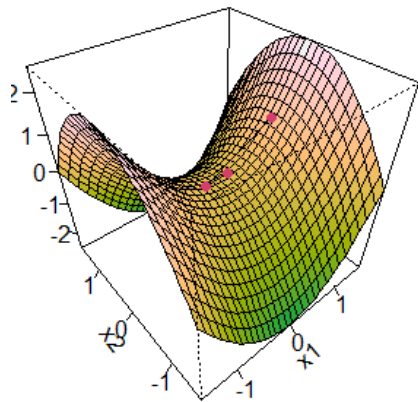


... Step 2 ...

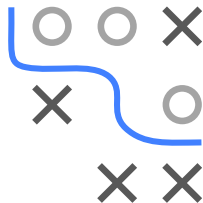


EXAMPLE: SADDLE POINT WITH GD

- How do saddle points impair optimization?
- Gradient-based algorithms **might** get stuck in saddle points

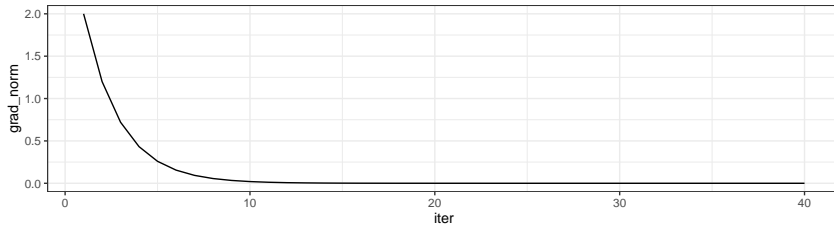


... Step 10 ... got stuck and cannot escape saddle point

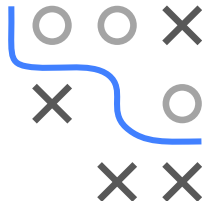


EXAMPLE: SADDLE POINT WITH GD

- How do saddle points impair optimization?
- Gradient-based algorithms **might** get stuck in saddle points



... Step 10 ... got stuck and cannot escape saddle point



SADDLE POINTS IN NEURAL NETWORKS

- For the empirical risk $\mathcal{R} : \mathbb{R}^d \rightarrow \mathbb{R}$ of a neural network, the expected ratio of the number of saddle points to local minima typically grows exponentially with d
- In other words: Networks with more parameters (deeper networks or larger layers) exhibit a lot more saddle points than local minima
- **Reason:** Hessian at local minimum has only positive eigenvalues. Hessian at saddle point has positive and negative eigenvalues.

