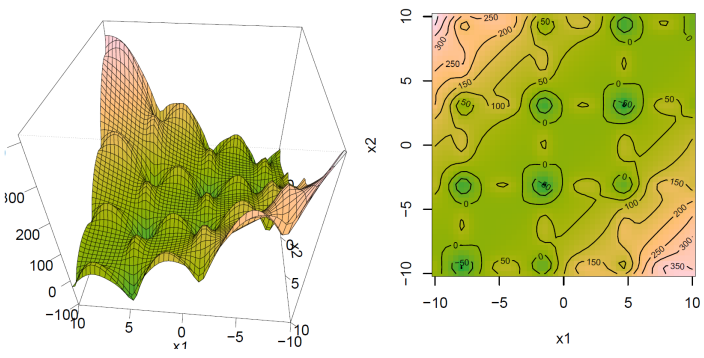
# Optimization in Machine Learning

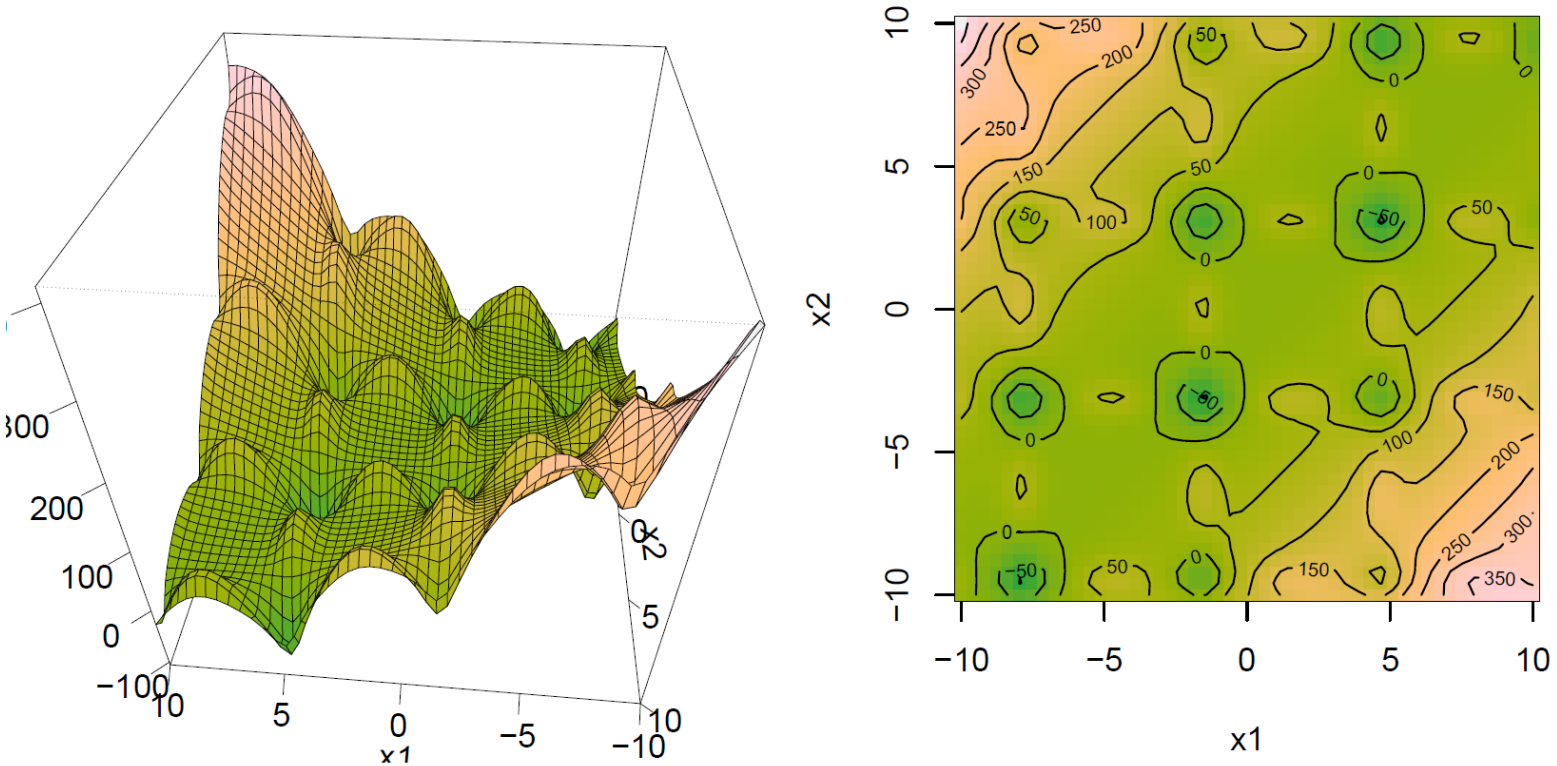
# First order methods:
# GD – Multimodality and Saddle points




## Learning goals

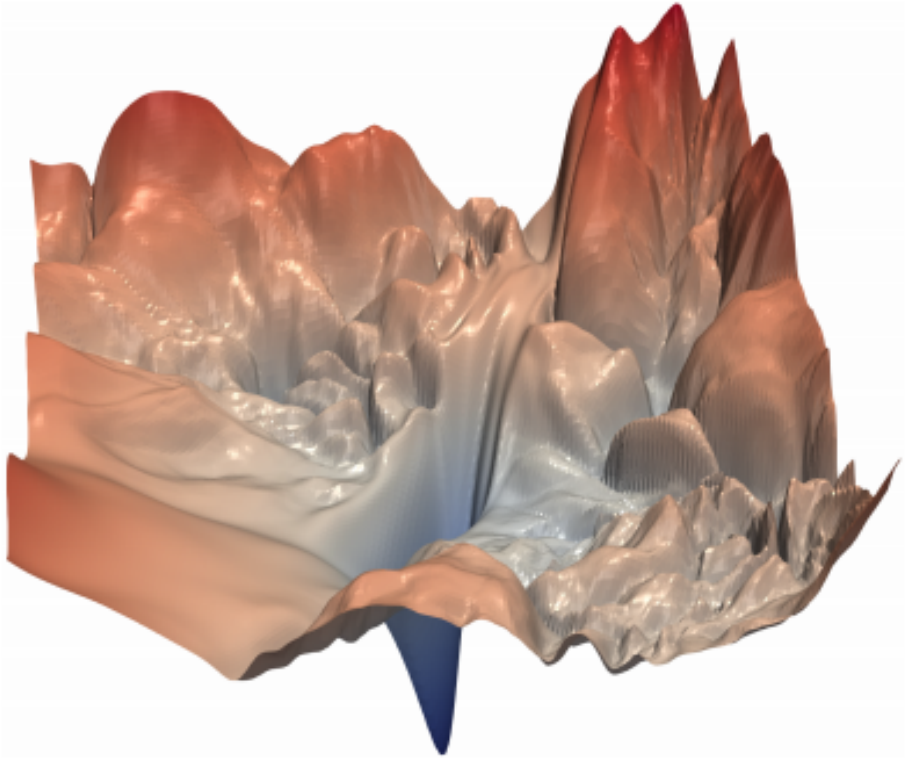
- Multimodality, GD result can be arbitrarily bad
- Saddle points, major problem in NN error landscapes, GD can get stuck or slow crawling

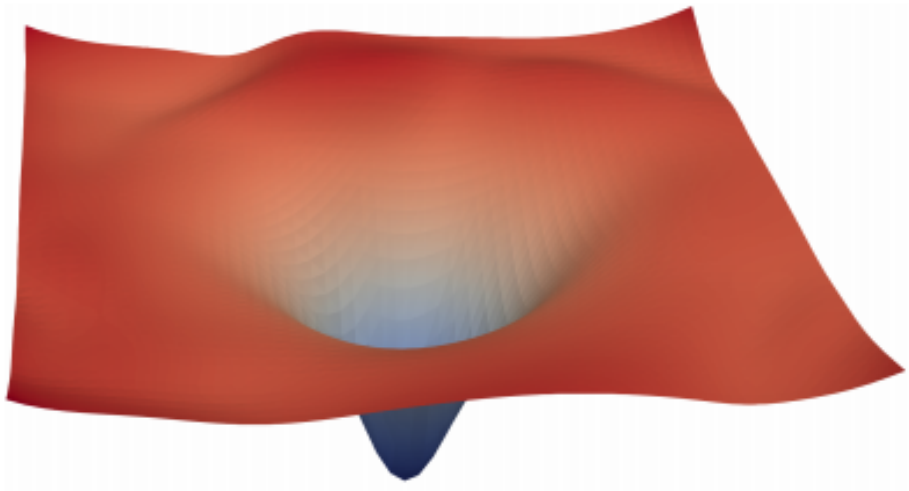
# UNIMODAL VS. MULTIMODAL LOSS SURFACES




Potential snippet from a loss surface with many local minima

# UNIMODAL VS. MULTIMODAL LOSS SURFACES

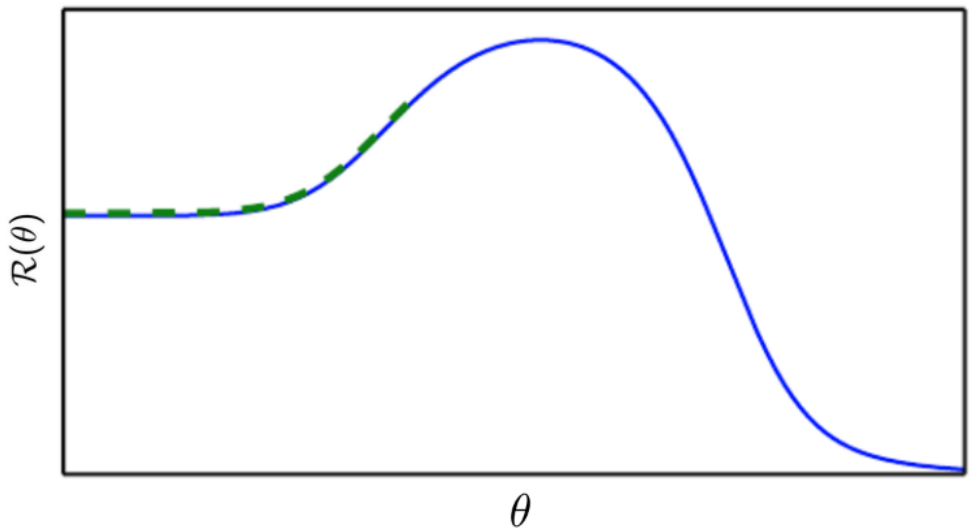




In deep learning we often find multimodal loss surfaces. Left: Multimodal loss surface; Right: (Nearly) unimodal loss surface. Source: Hao Li et al. (2017).

# GD: ONLY LOCALLY OPTIMAL MOVES

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

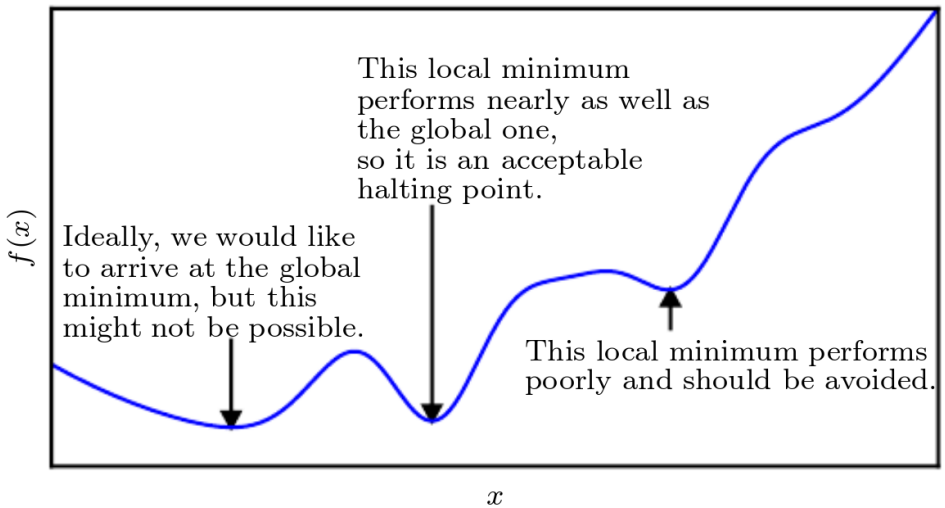



Source: Goodfellow, Ch. 8

- Figure above: initializion on “wrong” side of the hill resulting in suboptimal performance
- In higher dimensions, GD may move around the hill (potentially at the cost of longer trajectory and time to convergence)

# LOCAL MINIMA

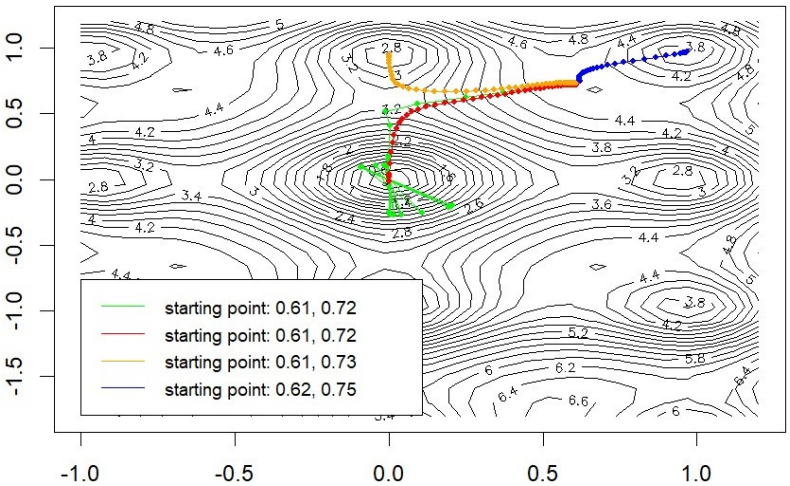
- In practice: Only local minima with high value compared to global minimium are problematic.

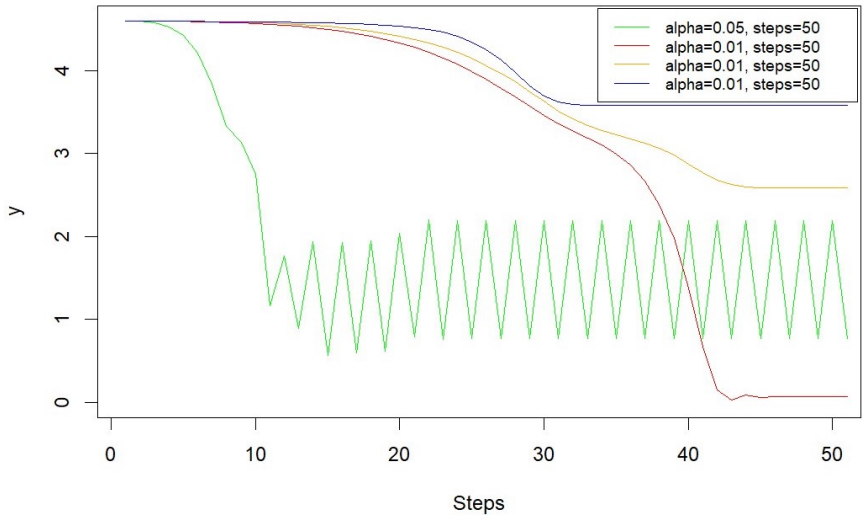



Source: Goodfellow, Ch. 4

# LOCAL MINIMA

(Non-)converging Gradient Descent for ackley function



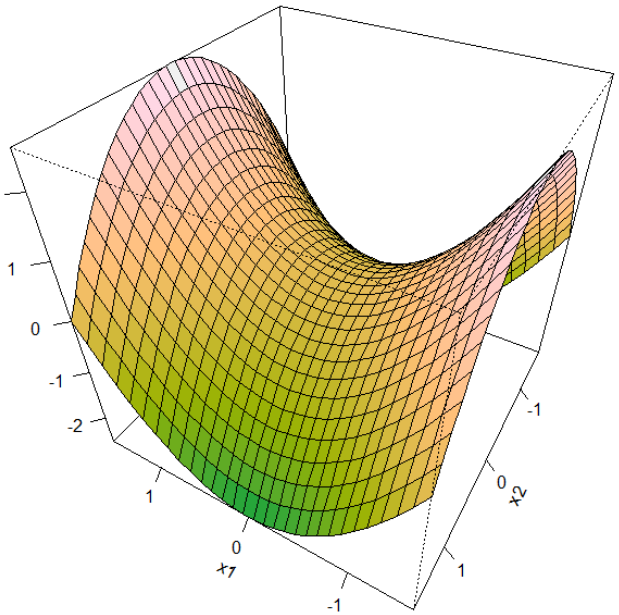




Small differences in starting value or step size can lead to huge differences in the reached minimum or even to non-convergence

# GD AT SADDLE POINTS

**Example:**

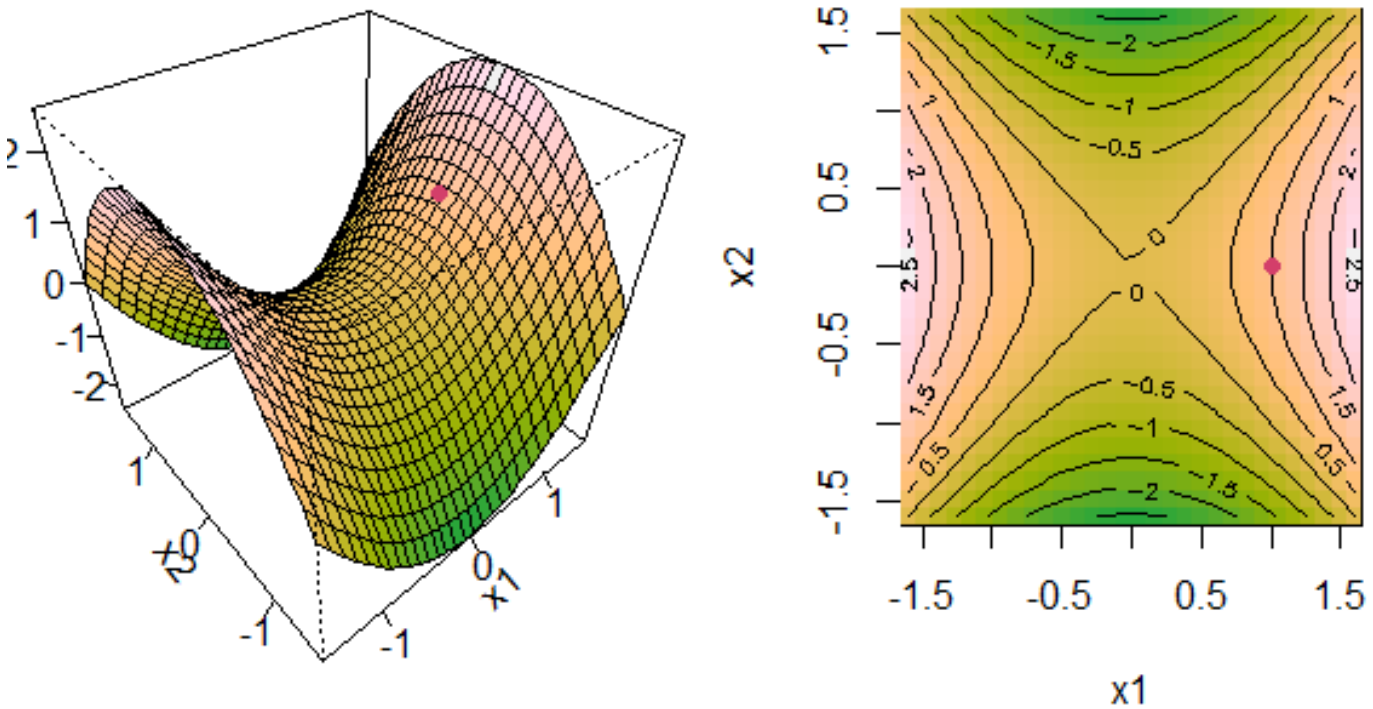
$$
\begin{aligned}
f(x_1, x_2) &= x_1^2 - x_2^2 \\
\nabla f(\mathbf{x}) &= (2 \cdot x_1, -2 \cdot x_2)^\top \\
\boldsymbol{H} &= \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}
\end{aligned}
$$




Along $x_1$, the function curves upwards (pos. eigenvalue belonging to eigenvector $(1, 0)$ of $\boldsymbol{H}$). Along $x_2$, the function curves downwards.

# EXAMPLE: SADDLE POINT WITH GD

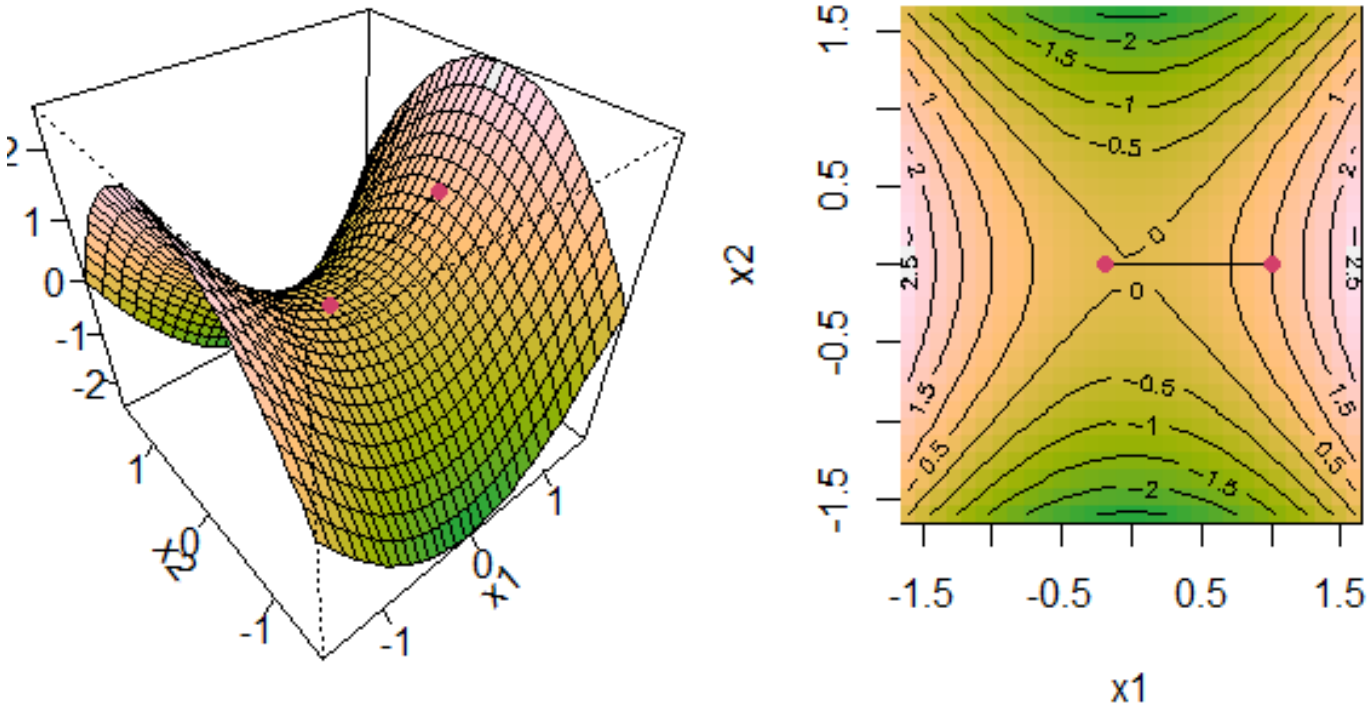
- So how do saddle points impair optimization?
- First-order algorithms that use only gradient information **might** get stuck in saddle points




Red dot: Starting location

# EXAMPLE: SADDLE POINT WITH GD

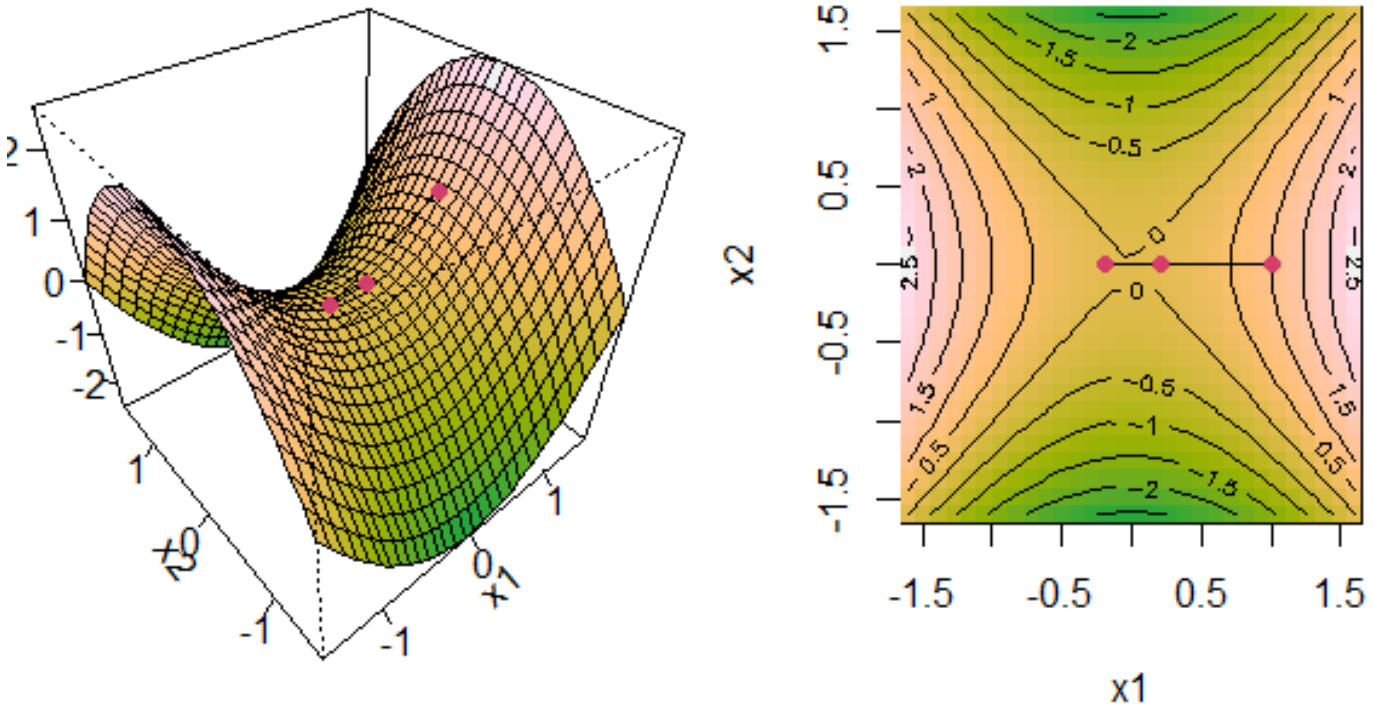
- So how do saddle points impair optimization?
- First-order algorithms that use only gradient information **might** get stuck in saddle points




First step...

# EXAMPLE: SADDLE POINT WITH GD

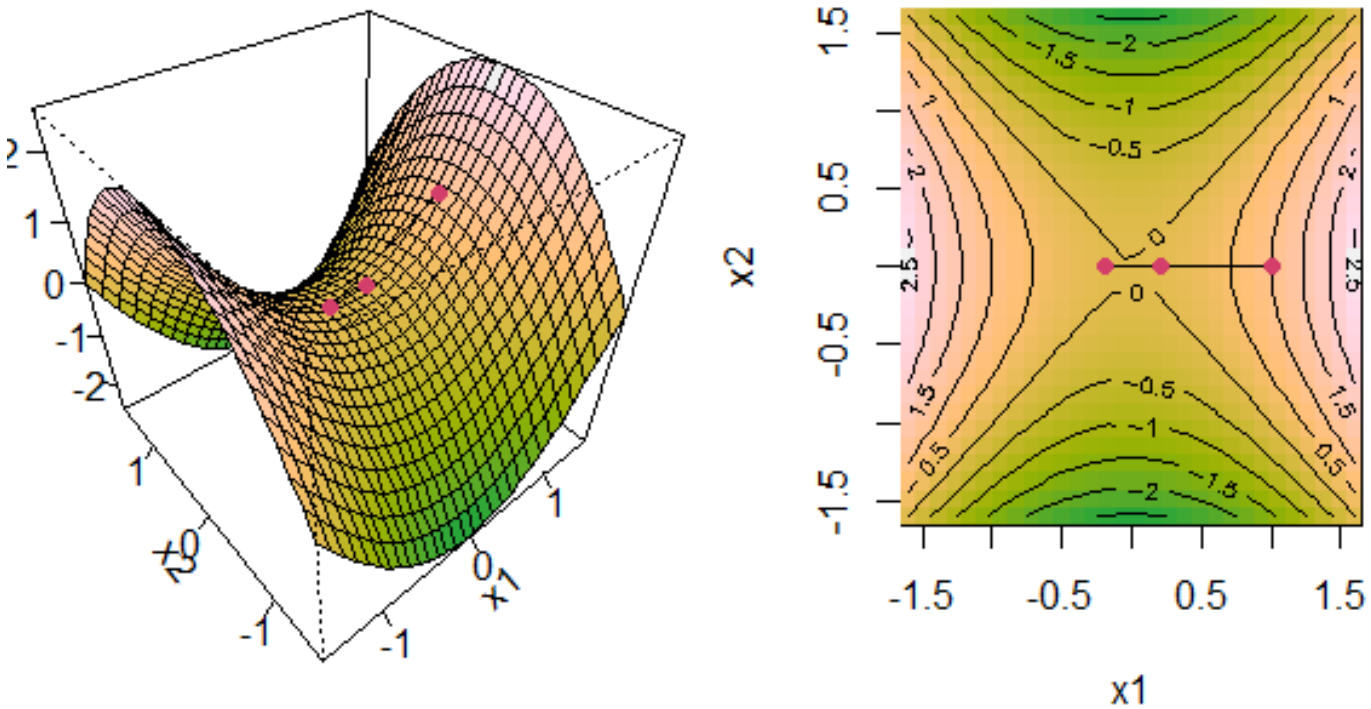
- So how do saddle points impair optimization?
- First-order algorithms that use only gradient information **might** get stuck in saddle points




...second step...

# EXAMPLE: SADDLE POINT WITH GD

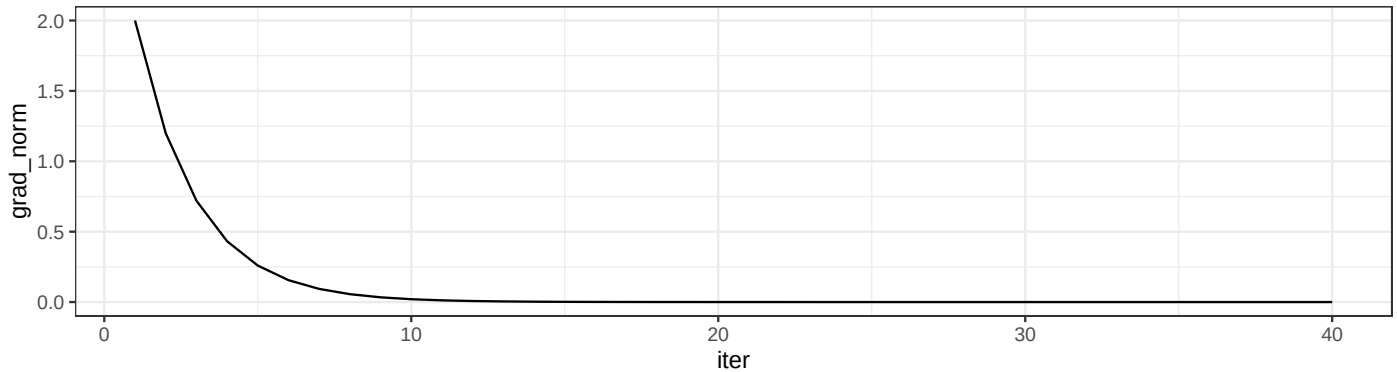
- So how do saddle points impair optimization?
- First-order algorithms that use only gradient information **might** get stuck in saddle points




...tenth step got stuck and cannot escape the saddle point!

# EXAMPLE: SADDLE POINT WITH GD

- So how do saddle points impair optimization?
- First-order algorithms that use only gradient information **might** get stuck in saddle points




...tenth step got stuck and cannot escape the saddle point!

# SADDLE POINTS

- In optimization we look for areas with zero gradient.
- A variant of zero gradient areas are saddle points.
- For the empirical risk $\mathcal{R}$ of a neural network, the expected ratio of the number of saddle points to local minima typically grows exponentially with $m$

$$\mathcal{R} : \mathbb{R}^m \to \mathbb{R}$$

  In other words: Networks with more parameters (deeper networks or larger layers) exhibit a lot more saddle points than local minima.
- Why is that?
- The Hessian at a local minimum has only positive eigenvalues. At a saddle point it is a mixture of positive and negative eigenvalues.

# SADDLE POINTS

- Imagine the sign of each eigenvalue is generated by coin flipping:
  - In a single dimension, it is easy to obtain a local minimum (e.g. “head” means positive eigenvalue).
  - In an $m$-dimensional space, it is exponentially unlikely that all $m$ coin tosses will be head.
- A property of many random functions is that eigenvalues of the Hessian become more likely to be positive in regions of lower cost.
- For the coin flipping example, this means we are more likely to have heads $m$ times if we are at a critical point with low cost.
- That means in particular that local minima are much more likely to have low cost than high cost and critical points with high cost are far more likely to be saddle points.
- “Saddle points are surrounded by high error plateaus that can dramatically slow down learning, and give the illusory impression of the existence of a local minimum” (Dauphin et al. (2014)).