3
1hr

Learning as optimization

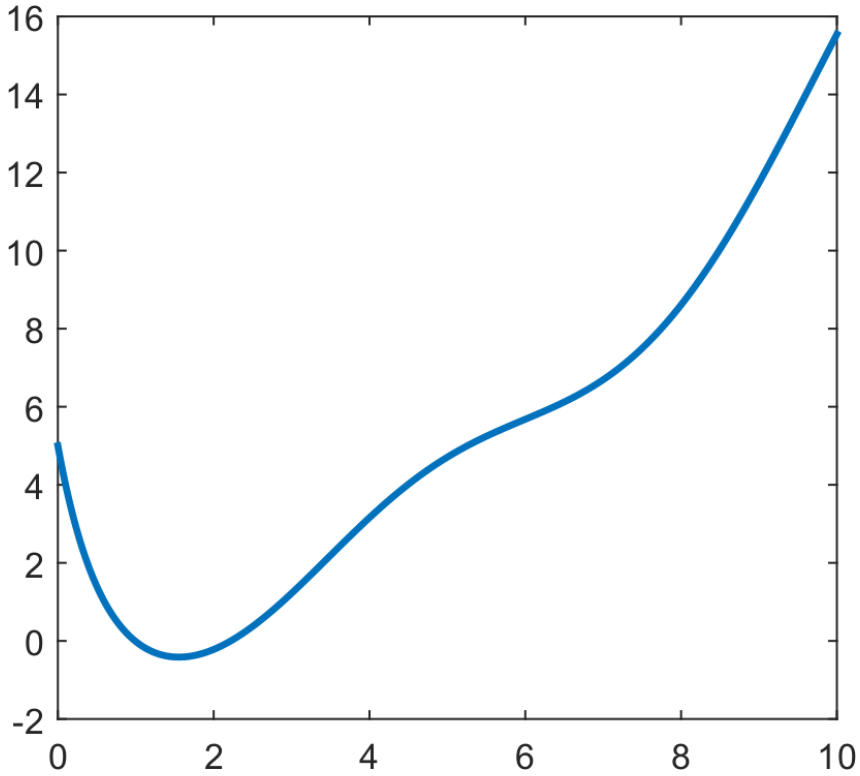
Many problems can be described by some quantity that we want to be as high or as low as possible

- Buying a car: min price, max quality
- Looking for the best shop to buy a given product: min price
- Cutting shapes from a sheet of steel: min discarded area

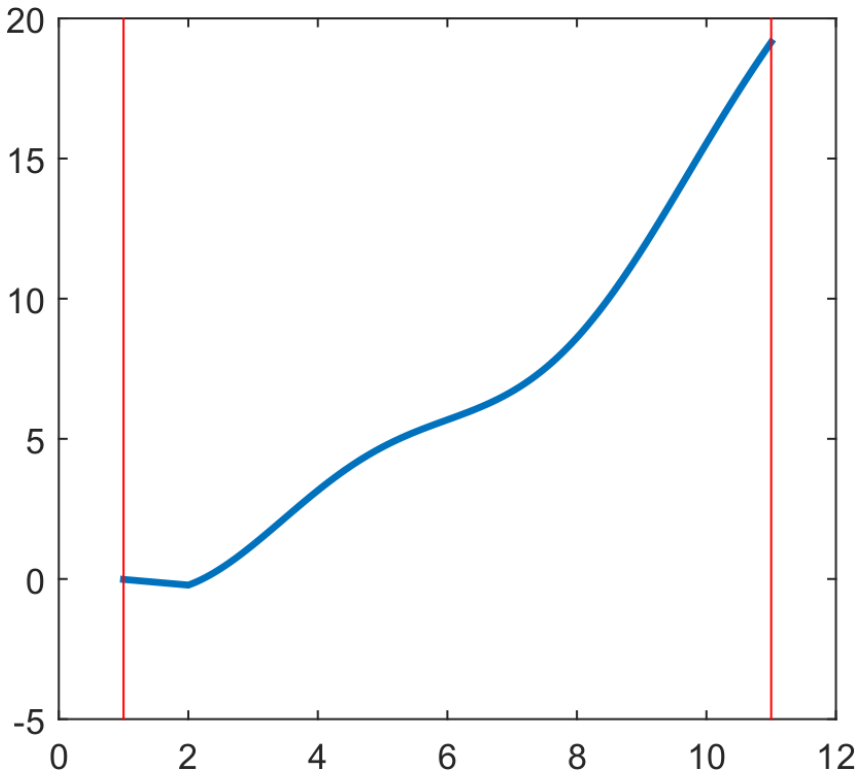
Our goal is to make the decision that results in the maximum or minimum value of the relevant quantity.

This is called **optimization**
and the measured quantity is the **objective function**

Two nice situations



Convex objective



Objective with constraints

When these occur, we are happier.

In general, the learning problem can be easily stated as an optimization problem:

- Minimize number of recognition errors
- Minimize difference of the output of a system w.r.t. a reference value
- Maximize number of tests passed
- Maximize the quality of an approximation
- ...

Optimization

TASK:
Finding extrema of an **objective function** $E = f(\mathbf{w})$, where
$f : S \subset \mathbb{R}^m \rightarrow \mathbb{R}$.

An **extremum** is a point $\mathbf{w}^* \in S$ that may be a **maximum** or **minimum**.

A point $\mathbf{w}^*$ is a minimum if there is a neighbourhood $R \subseteq S$, where the following holds:

$$f(\mathbf{w}) \geq f(\mathbf{w}^*) \quad \forall \mathbf{w} \in R \tag{1}$$

i.e.: A minimum is a point where f has a value smaller than in any other point in a given neighbourhood.

A point $\mathbf{w}^*$ is a maximum if it is a minimum of $-f$, or if $\leq$ is used.

Note: We will consider MINIMIZATION

A minimum is **relative** if $R \subset S$ strictly
i.e., there is some other point in S (outside R) where f has a smaller value than $f(\mathbf{w}^*)$.

A minimum is **absolute** if $R = S$.

- An **optimal solution** is an extremum of f
- An **optimal value** is $f(\mathbf{w}^*)$
- A **suboptimal solution** is an approximation to an optimal solution (value $\approx f(\mathbf{w}^*)$)
- A **feasible solution** is any point $\mathbf{w}$ which satisfies all hypotheses of the optimization problem – it *might be* an optimal solution.

Convex sets and functions

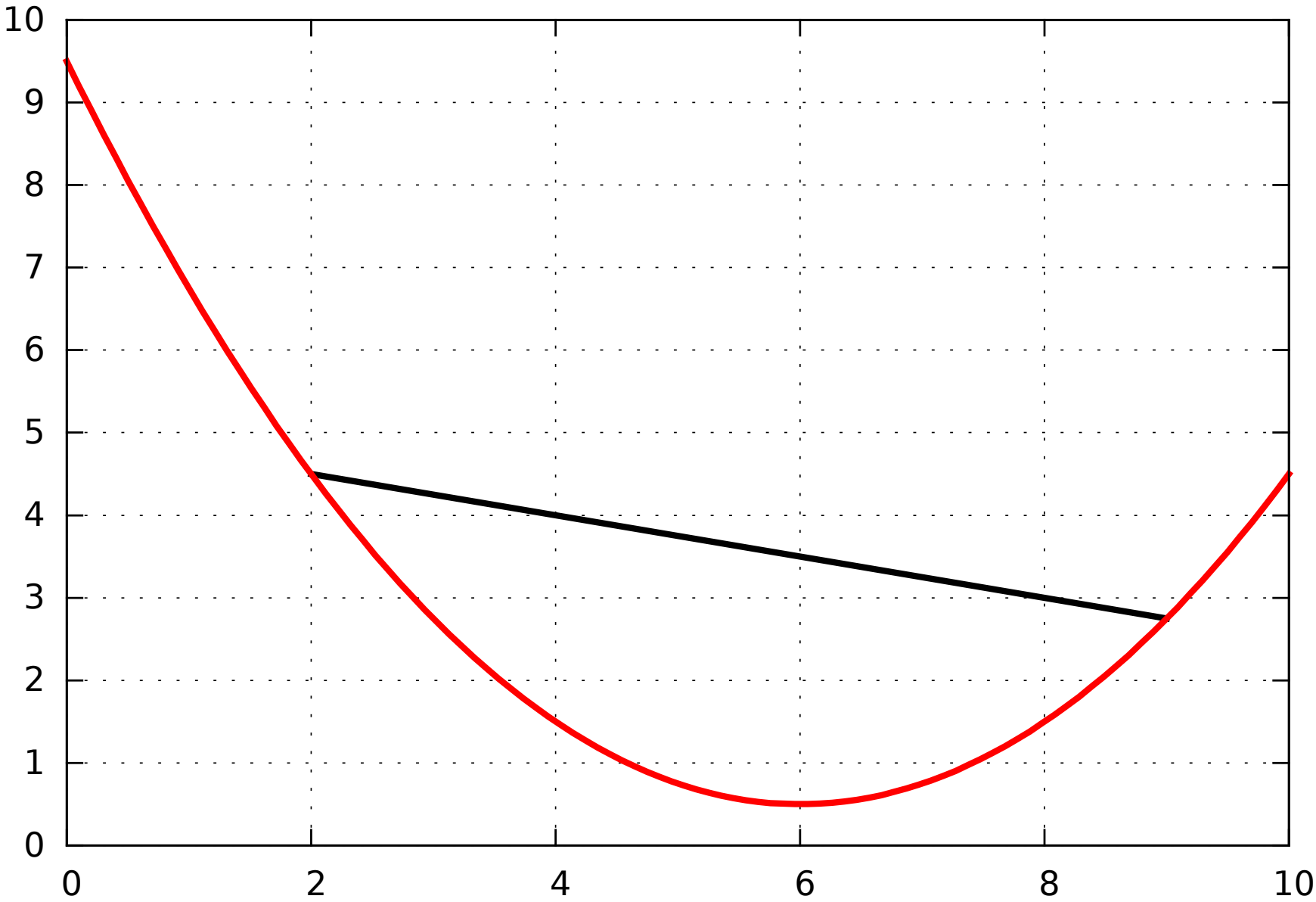
- **Convex sets:** A set $S \subset \mathbb{R}^m$ is convex if and only if, for any $\theta \in [0, 1]$,

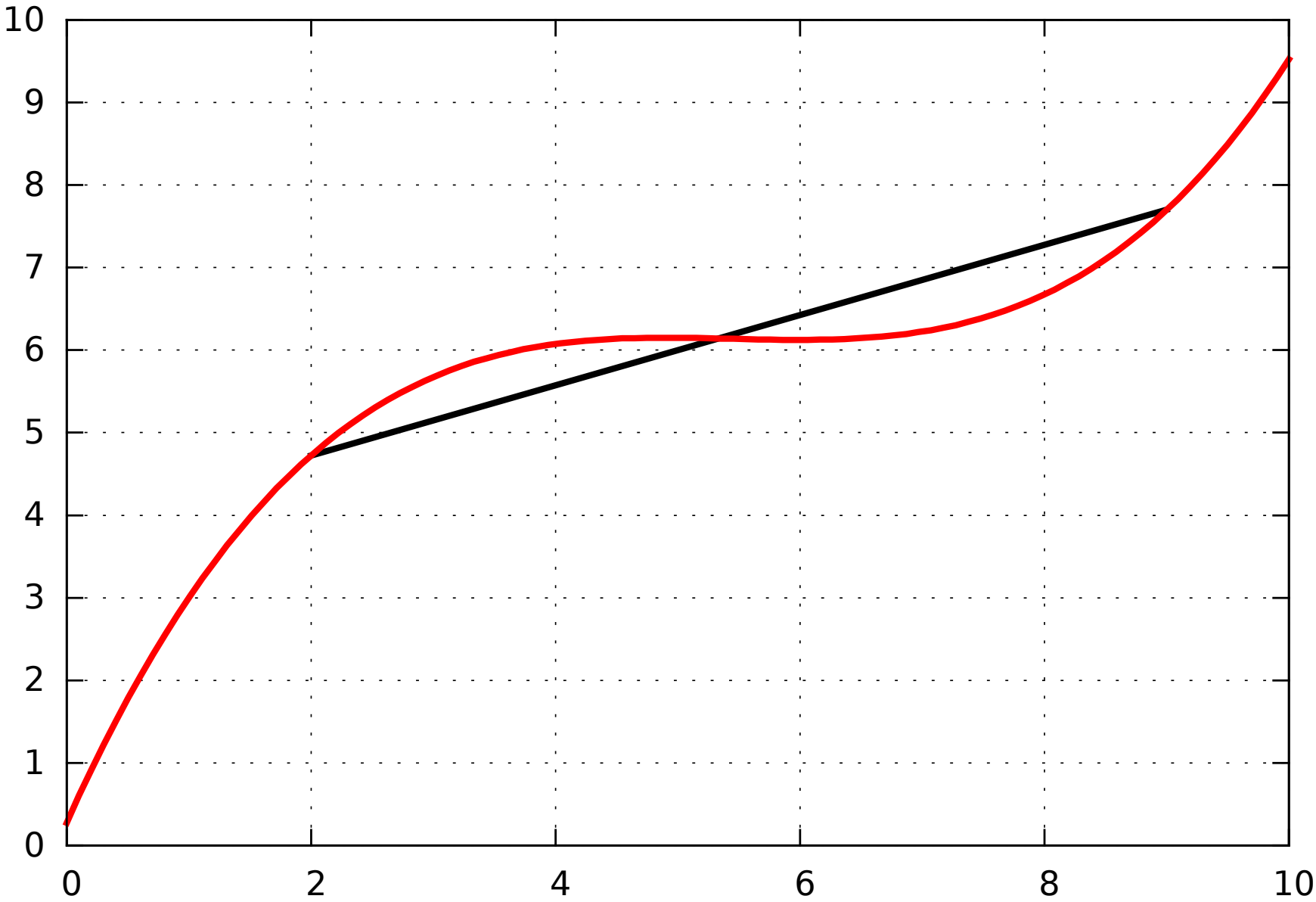
$$\forall x, y \in S \implies \theta x + (1 - \theta) y \in S \tag{2}$$

- **Convex functions:** A function $f : S \subseteq \mathbb{R}^m \to \mathbb{R}$ is convex if S is a convex set
  and if $\forall x, y \in S$, and with $0 \leq \theta \leq 1$:

$$f(\theta x + (1 - \theta) y) \leq f(x) + (1 - \theta) f(y) \tag{3}$$

- f is concave if $-f$ is convex


10
9
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10


10
9
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10

Multi-dimensional differentiation

The partial derivative of a function of m variables with respect to the j-th variable w_j is defined as

$$\frac{\partial f(\mathbf{w})}{\partial w_j} = \frac{\partial f(w_1, w_2, \ldots, w_n)}{\partial w_j} =$$

$$= \lim_{\Delta w_j \to 0} \frac{f(w_1, \ldots, w_j + \Delta w_j, \ldots, w_n) - f(w_1, \ldots, w_j, \ldots$$

Note: as many partial derivatives as variables (m).

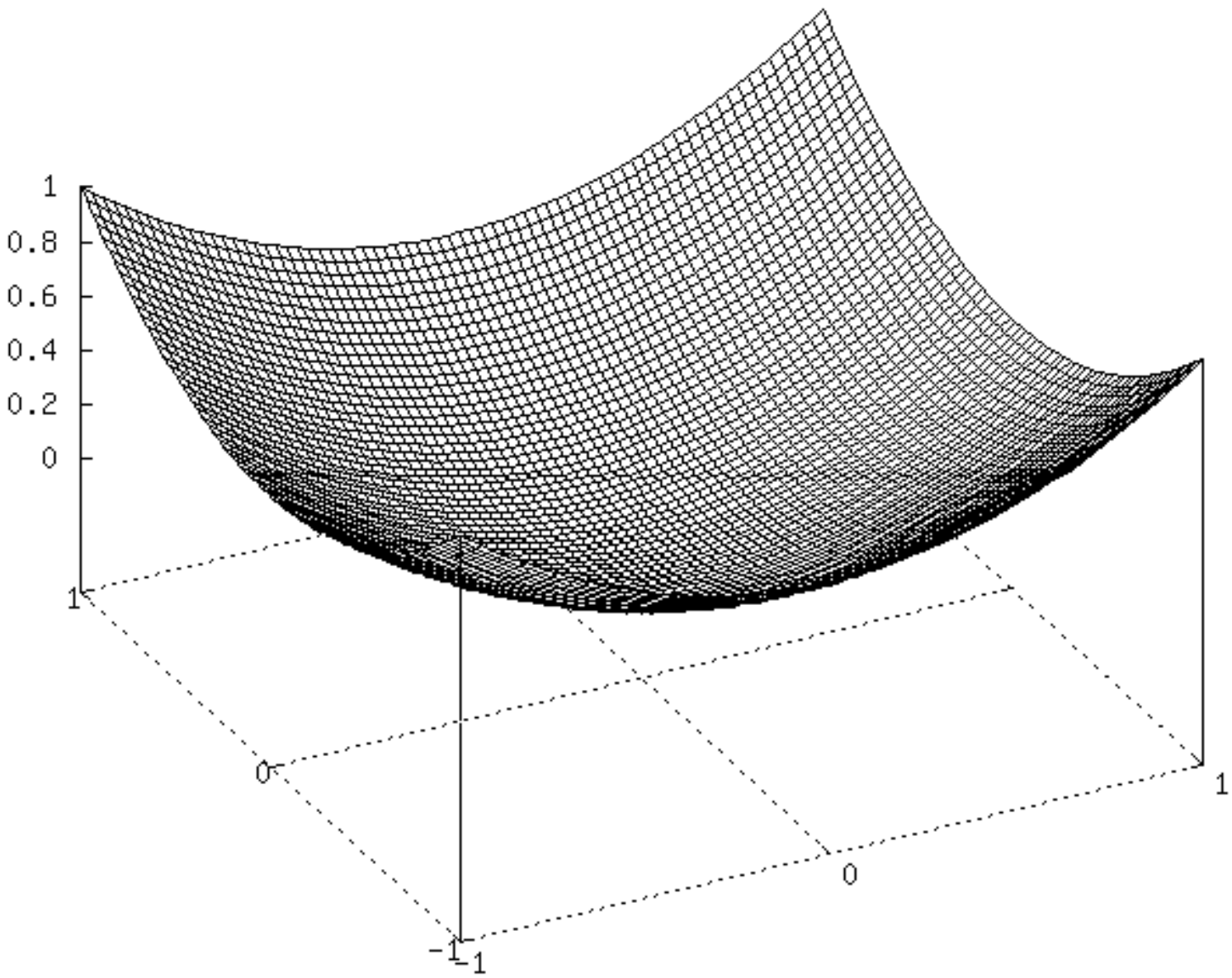
The gradient

$$\nabla f = \begin{bmatrix} \dfrac{\partial E}{\partial w_1} \\ \dfrac{\partial E}{\partial w_2} \\ \vdots \\ \dfrac{\partial E}{\partial w_m} \end{bmatrix}$$

The gradient is a vector field.

- Derivative $\rightarrow$ rate of growth of a function of a scalar variable
- Negative sign $\rightarrow$ decreasing

- Gradient *length (norm)* $\rightarrow$ rate of maximum growth
- *Direction* $\rightarrow$ *direction of maximum growth*


1
0.8
0.6
0.4
0.2
0
1
0
-1
-1
0
1

The gradient indicates the direction of maximum increase, and moving in the opposite direction $-\nabla f(\mathbf{w})$ we achieve the *maximum rate of decrease.*

This observation is very useful in optimization techniques.

Hessian matrix

or simply Hessian

$$H_f(\mathbf{w}) : \mathbb{R}^m \rightarrow \mathbb{R}^m \times \mathbb{R}^m \qquad \text{s.t.} \qquad h_{ij}(\mathbf{w}) = \frac{\partial^2 f(\mathbf{w})}{\partial w_i \partial w_j} \tag{4}$$

The Hessian matrix can be thought of as a list of m vectors

$$\mathbf{h}_i = \nabla \left(\nabla f(\mathbf{w}) \right)_i \tag{5}$$

Derivative is a linear operator and the order of differentiation does not matter:

$$\frac{\partial^2 f(\mathbf{w})}{\partial w_i \partial w_j} = \frac{\partial}{\partial w_i} \left(\frac{\partial f(\mathbf{w})}{\partial w_j} \right) = \frac{\partial}{\partial w_j} \left(\frac{\partial f(\mathbf{w})}{\partial w_i} \right) \tag{6}$$

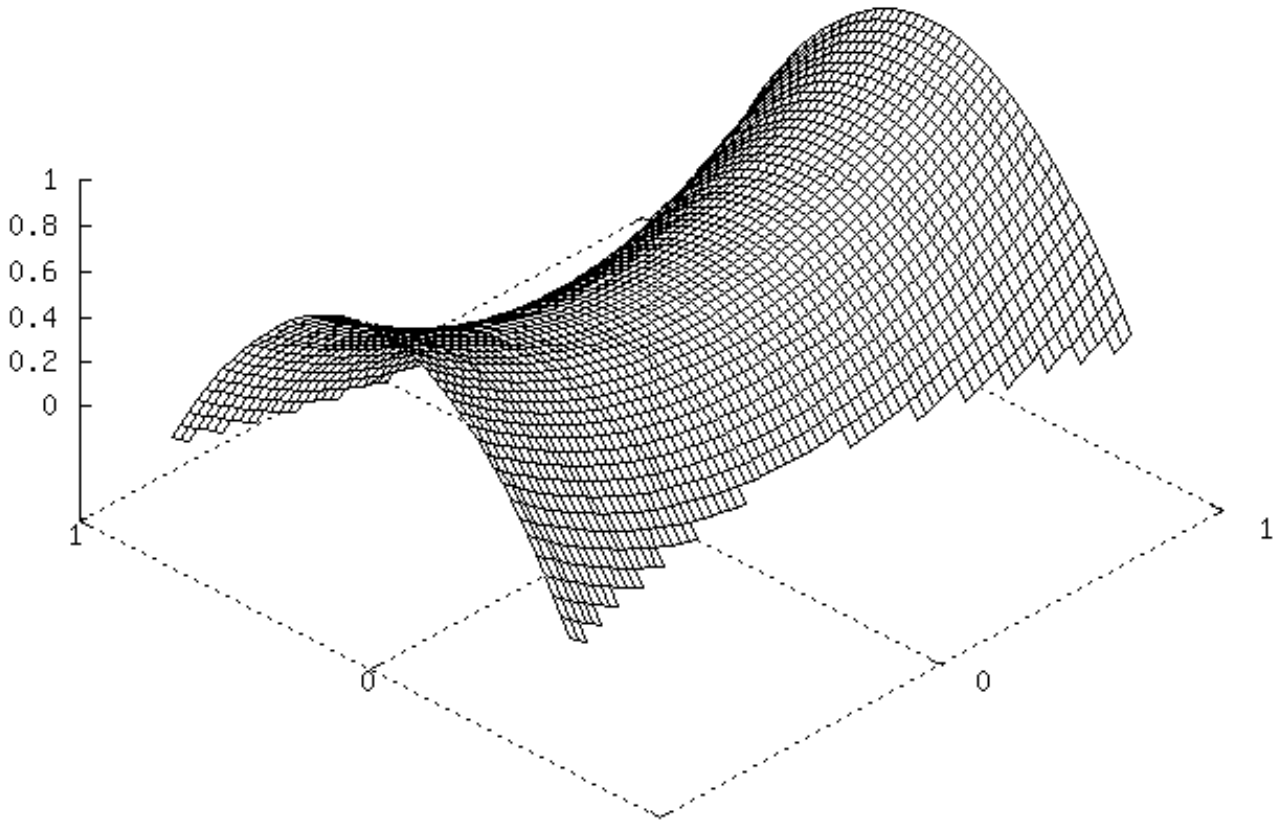
$\Rightarrow H$ is a symmetric matrix.

Characterizing minima

Necessary first-order minimum condition:

$$\nabla E(\mathbf{w}^*) = 0 \tag{7}$$

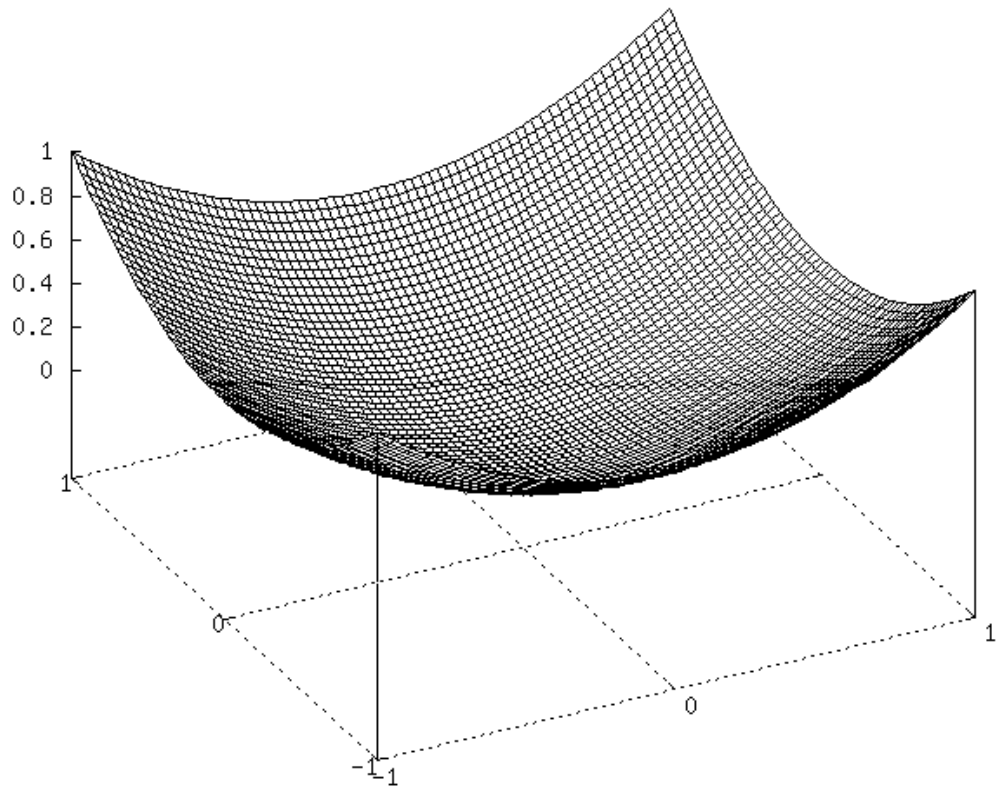
This condition characterizes all points which are local minima, but also local maxima or *saddle points* (points which are minima along one direction and maxima along another direction).



A saddle.

Convex cost function

The first-order condition is also **sufficient**.



A quadratic function or paraboloid is a convex function

Locally convex cost function

Convexity only in a neighbourhood of $\mathbf{w}^*$: intermediate situation.
The first-order condition is then a **necessary and sufficient condition of <u>local</u> minimum**.
Other local minima belong to different neighborhoods ("basins").

Taylor polynomials

The Taylor polynomial of degree 2 for a scalar function $f(w)$ centered around w_0:

$$f(w) \approx f(w_0) + f'(w) \mid_{w=w_0} (w - w_0) + \frac{1}{2} f''(w) \mid_{w=w_0} (w - w_0)^2 \quad (8)$$

Equivalent formula for a scalar field ($\mathbf{w} \in \mathbb{R}^m$)

$$f(\mathbf{w}) \approx f(\mathbf{w}_0) + \nabla f(\mathbf{w}) \mid_{\mathbf{w}=\mathbf{w}_0} (\mathbf{w}-\mathbf{w}_0) + \frac{1}{2}(\mathbf{w}-\mathbf{w}_0)^T H \mid_{\mathbf{w}=\mathbf{w}_0} (\mathbf{w}-\mathbf{w}_0) \tag{9}$$

Remark: Here we assume that $\mathbf{w}$ is a **column** vector.

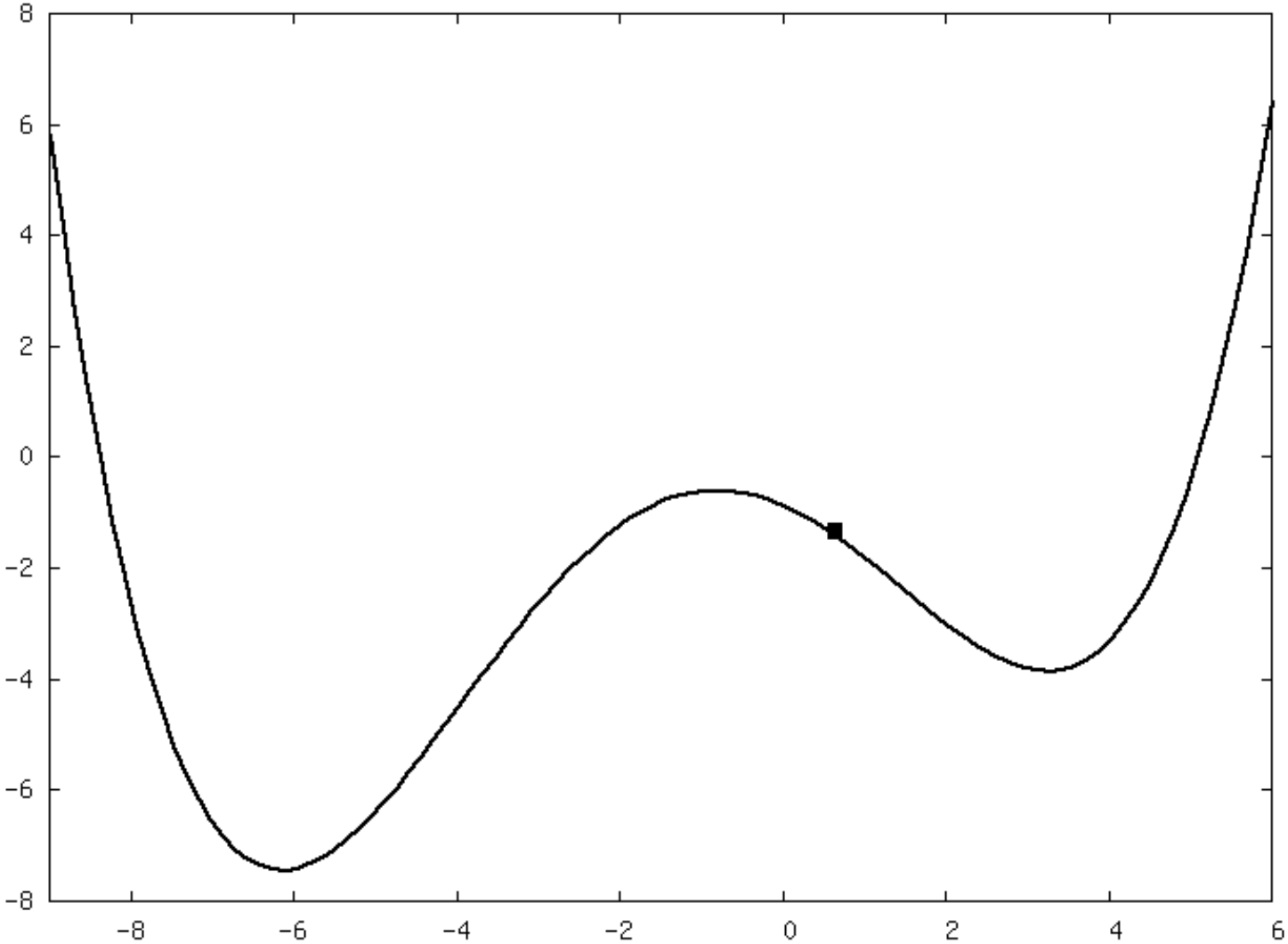
Descent techniques

Iteratively descend toward the minimum

$$\mathbf{w}(\tau + 1) = \mathbf{w}(\tau) + \Delta\mathbf{w}(\tau) \tag{10}$$

by taking steps in the direction of the reverse gradient, $-\nabla f(\mathbf{w})$:

$$\Delta\mathbf{w}(\tau) = -\eta f\left(\mathbf{w}(\tau)\right) \tag{11}$$


8
6
4
2
0
−2
−4
−6
−8
−8
−6
−4
−2
0
2
4
6

Gradient descent algorithm

1. Initialize: set $l = 0$; select $\mathbf{w}(l = 0) = \mathbf{w}_0$,
2. Compute the direction: $\Delta\mathbf{w}(l) \leftarrow \dfrac{-\nabla f(\,\mathbf{w}(l)\,)}{\|\,\nabla f(\,\mathbf{w}(l)\,)\,\|}$
3. Compute (by line search) the appropriate step size η
4. Scale $\Delta\mathbf{w}(l) \leftarrow \eta\Delta\mathbf{w}(l)$
5. Perform step $\mathbf{w}(l+1) \leftarrow \mathbf{w}(l) + \Delta\mathbf{w}(l)$
6. Compute convergence test. If necessary, iterate from step 2.

Line search

- We identify the direction of decrease by the **versor** (vector with unit length) $\mathbf{z} = -\nabla E(\mathbf{w}_\tau)/|\nabla E(\mathbf{w}_\tau)|$
- we perform a minimization of the function $y(t) = f(\mathbf{z}t)$, which is a function of one variable
- $\eta = \min_t y(t)$.

This may not be possible in distributed implementations, so in these cases the value of η must be guessed.
There are also indirect methods which modulate η adaptively, changing it according to the variations of the cost function.

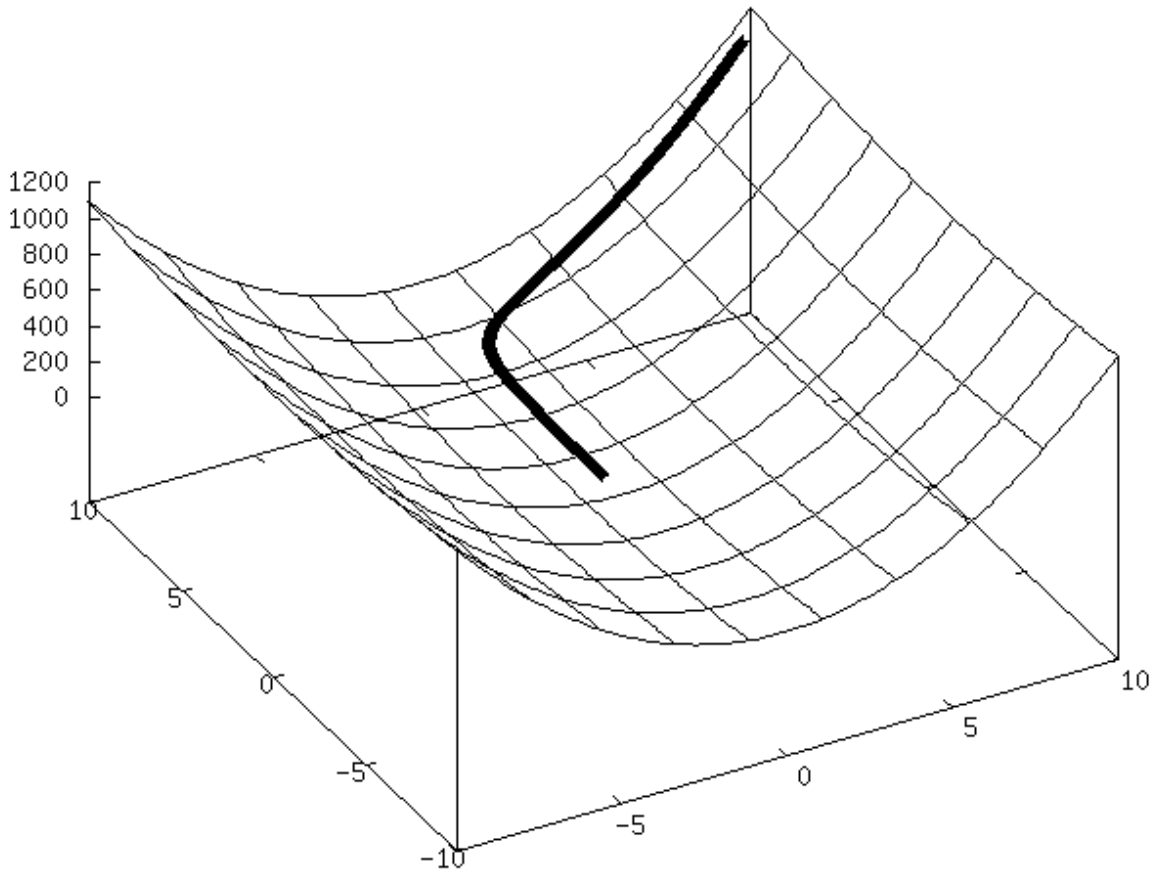
Features of gradient descent

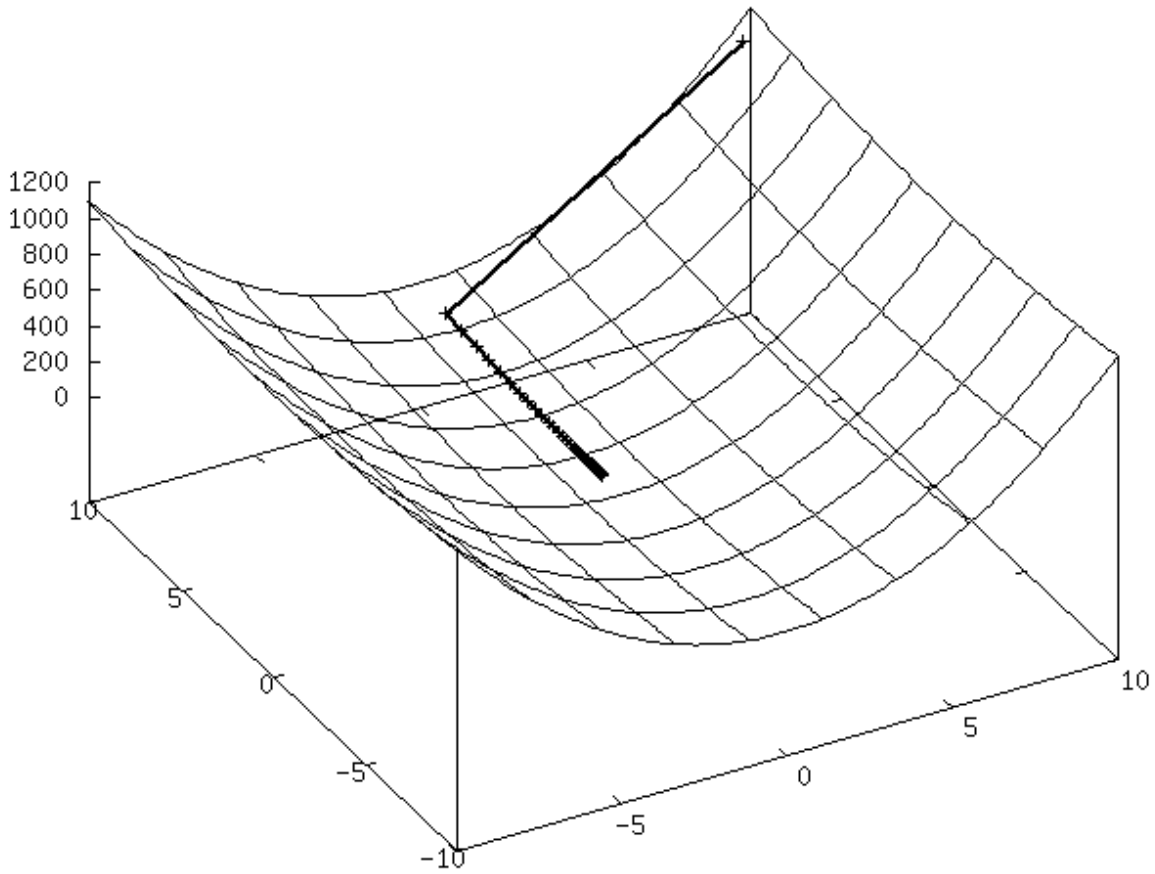
Pros

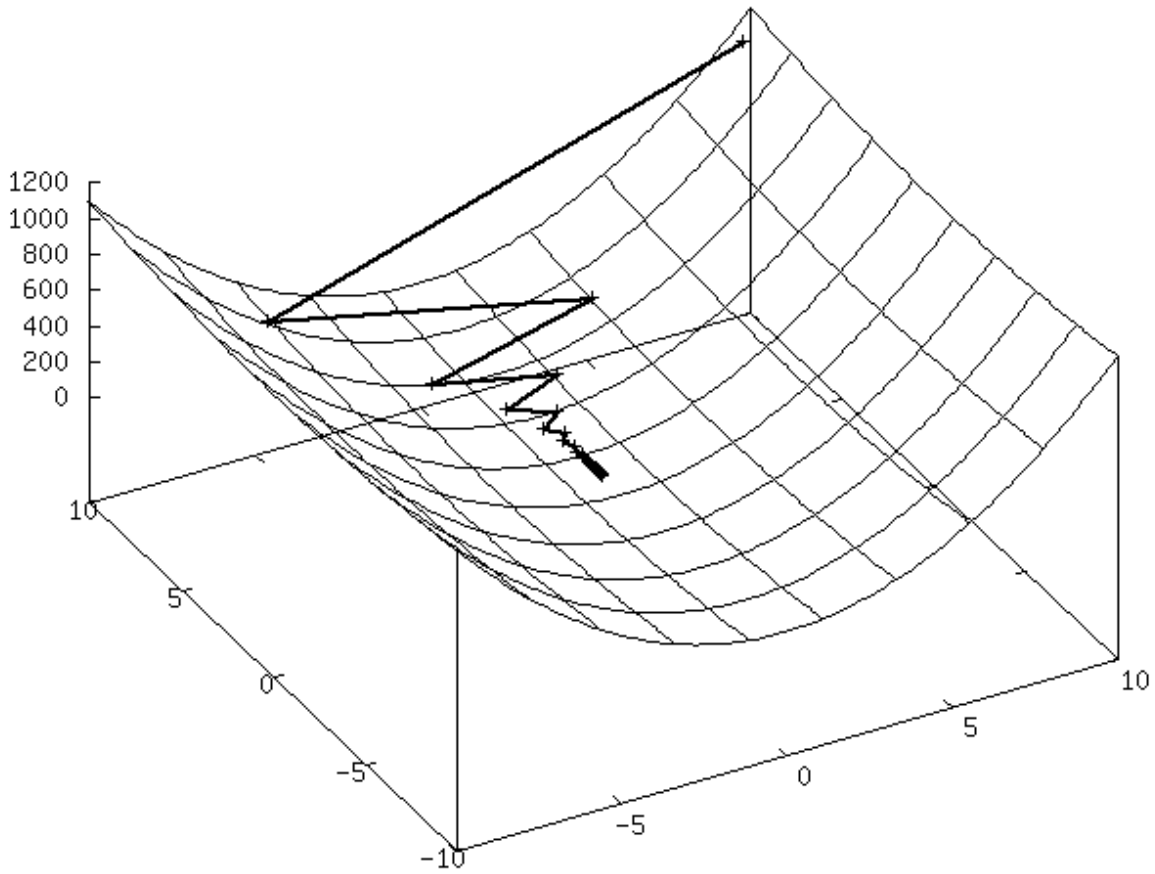
- Simple!

Cons

- Unnecessarily slow convergence (always directed exactly as the negative gradient)


1200
1000
800
600
400
200
0
10
5
0
-5
-10
-10
-5
0
5
10


1200
1000
800
600
400
200
0
10
5
0
-5
-10
-10
-5
0
5
10


1200
1000
800
600
400
200
0
10
5
0
-5
-10
-10
-5
0
5
10

Statistics

- Originally, the techniques for collecting data with the aim of governing a state.
- The science of using *empirical* data to create models based on probability.
- The science of induction of concepts from experiments.
- The science of discovery of the Laws of Nature / The science of modeling Nature.
- The science of scientific inquiry.

Plato: Myth of the cave. We perceive real models only through partial experience.

Aristotle, **St. Thomas of Aquino**: "Nihil est in intellectu quod non prius in sensu". Different perspective: no real models, only what we perceive contributes to our view of reality.

Empirists up to **Popper**: Experience is finite, therefore error=0 is impossible. Everything that is scientific must also be *falsifiable*, i.e., it must be accompained by an evaluation of its limitations (for instance, a probability of error).

Statistics

- is about collecting experimental observations
- then using them for estimating a model for the observed phenomenon
- usually this model is probabilistic
- The model is then evaluated w.r.t. its **probability of error**

Machine learning

- is about collecting experiences
- and then using them to learn some task (e.g., a classification rule)

A large part of Machine learning is the same as statistic inference. (Terms are sometimes different)

The training set

We assume that

- the training set was obtained by random sampling
- the observed phenomenon has a fixed statistical behaviour
- each experiment is unrelated to any other

The individual patterns are **independent** (the probability of any pattern does not depend on the probability of any other patterns)
They are also **identically distributed**: they come from a single data distribution, not different ones, nor one that varies in time.

independent, **identically distributed** or **i.i.d.** sample

Two goals of statistics

1. **Compute something** (a "statistic") based on the data
2. **Evaluate the probability that the above computation is wrong** (confidence)

High confidence = good generalization

Parameter estimation

problem setting: There is one (or more) quantity that we want to measure. This quantity is not directly accessible, for two possible reasons:

1. It is not a physically measurable quantity
2. The measurements are affected by random variations or **noise**.

Estimation requires the following ingredients:

- A training set;
- A **model**
- A set of parameters to adapt the model to the actual data;
- A model for the **uncertainty** of the measurements (a noise model).

Estimation process

1. Select an **estimator** suited to the problem
2. Select the value of parameters that optimizes the estimator

Remember: The estimator is a statistic = a function of the training set.

Error evaluation

Estimating the probability that an estimate is wrong

In the general case, a parameter w is estimated by a statistic $\hat{w}$.

There is a random error in this estimate: $|w - \hat{w}|$

In machine learning, **estimating generalization** means evaluating this error.

Often done by **generalization bounds**

$$P(\|w - \hat{w}| > \epsilon) < \delta$$

Summary

1. Learning is statistical estimation
2. The training set is the sample
3. The result of learning is a classifier with some measured performance
4. Performance of learning depends on the sample: it is a **statistic**, a random variable itself
5. It is necessary to evaluate the probability of correctness of this statistic
6. Good generalization = correct statistic