FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
NATURWISSENSCHAFTLICHE
FAKULTÄT

Abstract Learning Algorithms

Lecture “Mathematics of Learning” 2022/23

Wigand Rathmann
Friedrich-Alexander-Universität Erlangen-Nürnberg

WIRTSCHAFTS
MATHEMATIK

How does a learning algorithm work?

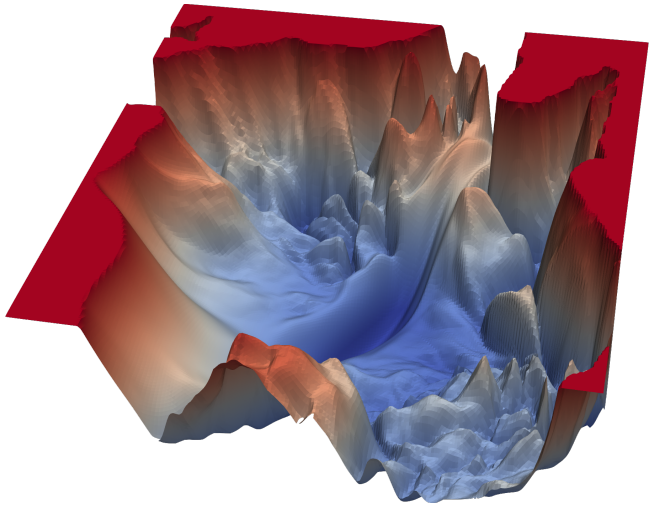
abstract approach

- Decide on a function or model with unknown parameters, e.g. a polynomial function of some order with unknown coefficients.
- Use a set of training data for determining best possible unknown parameters such that they minimize some loss function.
- Until now, we have focused on quadratic loss functions that measure the quadratic deviations between the observations and the predictions, summed up for all training data points.
- For linear regression models (see last lectures), least square method showed that the loss function could be minimized easily by determining critical points where the partial derivatives w.r.t. the parameters vanish. The latter can actually be determined by solving a linear equation system only.
- For more complex nonlinear models, however, minimizing the loss function can be more difficult, and we need an algorithm for this optimization problem.

Minima of the loss function

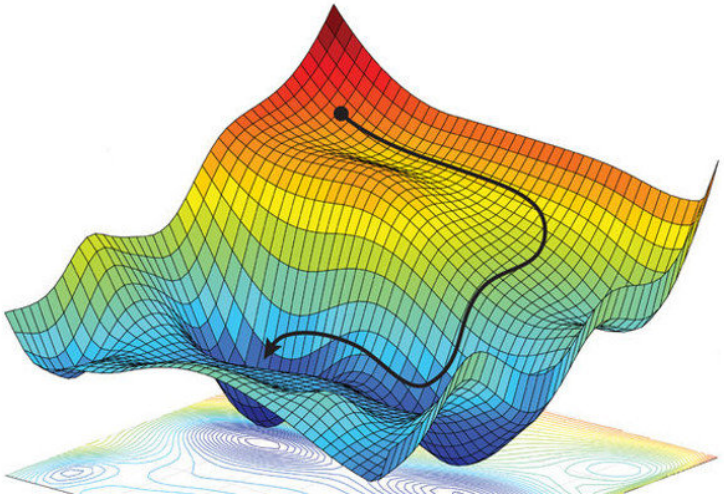
Question:
How does the loss function help us to find good parameters?
(Ambitiuous) Goal:
Determine the model parameters that attain the global minimum of the loss function

- computing the global minimum for a large system of nonlinear functions is challenging (at least!)
- loss function C is typically non-convex



Iterative minimization of loss function

(More realistic) Goal:
Decrease the loss function value iteratively.



Typically, this is done via gradient descent algorithms.

Gradient descent algorithm

Consider an unconstrained optimization problem. We assume a function $g : \mathbb{R}^n \to \mathbb{R}$ is bounded from below and continuously differentiable.
We also assume the gradient ∇g is Lipschitz continuous:

$$\text{There is } L > 0 \text{ with } ||\nabla g(x) - \nabla g(y)|| \leq L||x - y|| \text{ for all } x, y \in \mathbb{R}^n. \tag{1}$$

We consider the minimization problem

$$\begin{aligned} &\text{minimize } g(x) \\ &\text{s.t. } x \in \Omega := \mathbb{R}^n \end{aligned}$$

Gradient descent algorithm

Basic Descent Algorithm

1. Input: g, $x_0 \in \Omega$; choose $\epsilon > 0$.
2. Set $x_k \leftarrow x_0$.
3. While x_k does not satisfy an optimality condition (e.g. $|\nabla g(x_k)| > \epsilon$) do
   3.1 Calculate a descent direction d_k of g at x_k.
   3.2 Calculate a step size $t_k > 0$ such that

$$g(x_k + t_k d_k) < g(x_k)$$

   and $x_k + t_k d_k \in \Omega$.
   3.3 Set $x_k \leftarrow x_k + t_k d_k$.
4. Return $x_* \leftarrow x_k$.

Gradient Descent

- The step size is often called *learning rate* in data analysis settings.
- The direction of steepest descent is given by $-\nabla g$, i.e., by the negative gradient of g.

Convergence of Gradient Descent Method

For functions g that satisfy conditions from last slide and under additional conditions on chosen step size, the generated points x_k in the algorithm satisfy $\lim_{k\to\infty} \nabla g(x_k) = 0$. Every cumulation point $x_\star$ satisfies $\nabla g(x_\star) = 0$.

We will investigate the convergence of a stochastic variant of gradient descent more formally later and refrain from giving more details here.
We now have everything together for presenting a simple learning algorithm that minimizes a loss function via gradient descent.

A simple iterative Learning Algorithm

- define some loss function on training data
- determine model parameters iteratively via gradient descent algorithm
- use trained model for predicting results on new data

A simple iterative Learning Algorithm

- define some loss function on training data
- determine model parameters iteratively via gradient descent algorithm
- use trained model for predicting results on new data

Discussion

- We next make this abstract learning algorithm concrete both for the famous so-called *artificial neural networks*.
- Many other learning algorithms (e.g., regression, SVM, ...) are typically solved in practice by variants of gradient descent algorithms, within an iterative algorithm as outlined here.
- We will later see a computationally very efficient method that are very famous in data analysis, namely stochastic gradient algorithms.