Numerik des Maschinellen Lernens

Jan Heiland

TU Ilmenau – Sommersemester 2024

Contents

Vorwort

Das ist ein Aufschrieb der parallel zur Vorlesung erweitert wird.

Korrekturen und Wünsche immer gerne als *issues* oder *pull requests* ans github-repo.

Chapter 1

Introduction

What is *Numerical Methods for Machine Learning*? (ML)

In short, for the training of an ML model, a computer steps through millions of instructions that are formulated in terms of mathematical expressions. Same holds for the evaluation of such a model.

In order to describe what is happening and for the analysis later, we introduce the general concepts

- algorithm
- methods
- accuracy
- stability
- exception

some of which are classical *numerical analysis*.

Curiously, the term *algorithm* is similarly intuitive and abstract. It took great efforts to come up with a general and concise definition that would meet requirements and limitations of all fields (ranging from, say, *cooking recipes* to the analysis of of *formal languages*).

Definition 1.1 (Algorithm)**.** A problem solution procedure is called an *algorithm* if, and only if, there exists a *Turing machine* that is equivalent to the procedure and that, for every input for which a solution exists, *stops*.

This definition is not too helpful in its generality – we haven't even defined what is a Turing machine.

> A *Turing machine* can be described as a machine that reads a strip of instructions and that can write onto this strip. Depending on what it reads it may move forward, move backward, or stop (when

the strip has reached a predefined state). The beauty is that this setup can be put into an entirely mathematical framework.

It is more helpful and more common, to look at the implications of this definition to check if a procedure has at least the necessary conditions for being an algorithm

- The algorithm is described by finitely many instructions (finiteness).
- Every step is *feasible*.
- The algorithm requires a finite amount of memory.
- It will finish after finitely many steps.
- At every step, the next step is uniquely defined (*deterministic*).
- For the same initial state, it will stop at the same final state (*determined*).

Thus, an informal good-practice definition of an algorithm could be

Definition 1.2 (Algorithm – informally)**.** An procedure of finitely many instructions that deterministically defines a determined solution to a problem – if it exsists – in finitely many steps is called an algorithm.

Note how some properties (like finitely many instructions) are assumed a-priori.

Further reading:

- wikipedia:Algorithmus

Referenzen