

Introduction

Computation is everywhere, but what is computation actually? In this lecture we will discuss the power and limitations of computation. Computational thinking is about understanding machine intelligence: What is computable, and how efficiently?

Understanding computation lies at the heart of many exciting scientific, social and even philosophical developments. Computational thinking is more than programming a computer, it means thinking in abstractions. Consequently, computational thinking has become a fundamental skill for everyone, not just computer scientists. For example, functions which can easily be computed but not inverted are at the heart of understanding data security and privacy. Machine learning on the other hand has given us fascinating new tools to teach machines how to learn function parameters. Thanks to clever heuristics, machines now appear to be capable of solving complex cognitive tasks. To give just one more example: How can we design the best electronic circuit for a given problem? In this class, we study various problems together with the fundamental theory of computation.

While computation is predominantly an engineering discipline, some of our insights will be highly surprising. One may claim that this class is going to discuss the three biggest computational surprises. Discussing these surprises will add a philosophical touch to the class.

The weekly lectures will be based on blackboard discussions and coding demos, supported by a script and coding examples. The course uses Python as a programming language. Python is popular and intuitive, a programming language that looks and feels like human instructions. The lecture will feature weekly exercises, on paper and in Python.

→ notebook

Have fun!