Machine Learning: Bra Tittel

Vetle Nevland, Vetle Vikenes & Sigurd Sørlie Rustad

FYS-STK4155 – Applied Data Analysis and Machine Learning
Autumn 2021
Department of Physics
University of Oslo

November 22, 2021

ABSTRACT: Coming soon!

Contents

1 Introduction

We will in no way answer all questions linked to the aforementioned methods. So that anyone can reproduce or continue our studies, we list all the code, results and instructions on running the code in our GitHub repository[1].

2 Theory

In the theory-section we aim to give a brief explanation of the main concepts and terminology used in this report. For a more in-depth explanation we recommend reading the appropriate sections in [2], which has been of great inspiration and help for us throughout the project.

The diffusion equation

The full diffusion equation reads

$$\frac{\partial u(\mathbf{r},t)}{\partial t} = \nabla \cdot \left[D(u,\mathbf{r})\nabla u(\mathbf{r},t)\right], \tag{2.1}$$

where $\mathbf{r}$ is a positional vector and $D(u,r)$ the collective diffusion coefficient. If $D(u,\mathbf{r}) = 1$ the equation simplifies to a linear differential equation

$$\frac{\partial u}{\partial t} = \nabla^2 u(\mathbf{r},t), \tag{2.2}$$

or

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) u(x,y,z,t) = \frac{\partial u(x,y,z,t)}{\partial t} \tag{2.3}$$

in cartesian coordinates. In this report we are going to study a one dimensional rod of length $L = 1$. I.e. we need the one dimensional diffusion equation

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{\partial u(x,t)}{\partial t}, \tag{2.4}$$

with boundary conditions

$$u(x,0) = \sin(\pi x) \quad 0 \leq x \leq L, \tag{2.5}$$

$$u(0,t) = 0 \quad t \geq 0 \text{ and} \tag{2.6}$$

$$u(L,t) = 0 \quad t \geq 0. \tag{2.7}$$

Explicit forward Euler

In this section we want to cover the explicit forward Euler. By explicit we mean that the value at the next grid point is determined entirely by known or previously calculated values.

[1] https://github.com/sigurdru/FYS-STK4155/tree/main/project3

The one-dimensional diffusion equation (2.4) reads

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{\partial u(x,t)}{\partial t} \quad \text{or} \quad u_{xx} = u_t. \tag{2.8}$$

In this report we are going to study a one dimensional rod of length $L = 1$, with boundary conditions

$$u(x,0) = \sin(\pi x) \quad 0 \leq x \leq L, \tag{2.9}$$

$$u(0,t) = 0 \quad t \geq 0 \text{ and} \tag{2.10}$$

$$u(L,t) = 0 \quad t \geq 0. \tag{2.11}$$

To approximate the solution, we have to discretize the position and time coordinates. We can choose $\Delta x = L/N$ and Δt as small steps in x-direction and time, where N are the number of discretized points in x-direction. Then we can define the value domain of t and x,

$$t_j = j\Delta t, \quad j \in \mathbb{N}_0 \quad \wedge \quad x_i = i\Delta x, \quad \{i \in \mathbb{N}_0 | i \leq N\}.$$

The algorithm for explicit forward Euler in one dimension (from [1] chapter 10.2.1) reads

$$u_{i,j+1} = \alpha u_{i-1,j} + (1 - 2\alpha)u_{i,j} + \alpha u_{i+1,j} \tag{2.12}$$

where

$$\alpha = \frac{\Delta t}{\Delta x^2}.$$

This has a local approximate error of $O(\Delta t)$ and $O(\Delta x^2)$. Note that the expression on the right hand side, used to calculate the value at a time $t_j + \Delta t$, only contains the state of the system at time t_j. This can be written as a matrix equation (see [1] chapter 10.2.1)

3 Method

4 Results

5 Discussion

6 Conclusion

References

[1] Morten Hjorth-Jensen. Computational physics, lecture notes fall 2015. *Department of Physics, University of Oslo*, August 2015. https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf.

[2] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, 810:1–124, May 2019.