Computational Modeling of Neuronal Plasticity : Online Course

Florence I. Kleberg and Jochen Triesch, FIAS 2017 - 2020

Introduction

(Video Introduction) Welcome to the Online Course "Computational Modeling of Neuronal Plasticity". In this course you will learn how computational neuroscientists use mathematical models and computer simulations to study different plasticity phenomena in the brain. During the course, you will program from scratch your own neuron model, a so-called leaky-integrate-and-fire (LIF) neuron model, and simulate it with a computer. Then you will add various neuronal properties and plasticity mechanisms to the model and study how they operate. This course will deepen your understanding of neural plasticity and prepare you for studying plasticity and learning in larger models such as neural networks.

General advice for the course

- It is advised to be familiar with the basics of the Python programming language before commencing the course. A good place to start is for instance: `http://www.tutorialspoint.com/python/`.
- The book "Theoretical Neuroscience" by Dayan and Abbott is an advised read [**?**]. It will help you to get acquainted with different concepts in simulating neurons and plasticity mechanisms. This course is in part based on the book.
- With mathematical equations, there is often more than one way to express the same thing. Equations shown in this course may look different from those in other resources, even if they convey the same meaning.
- The main idea is to model a neuron and its plasticity mechanisms from scratch, without the use of specialised neuronal modelling software such as Brian, NEURON or NEST. Of course, the use of python packages such as numpy, scipy, ipython and so on is allowed.
- Good programming practices: It is strongly advised to use good programming habits, such as object-oriented programming, a separate parameter file, and commenting in the code. An excellent guide for scientific programming can be found here:
`http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000589`
- Good programming practices save a lot of time later on, including other people reading your code.
- Some students also use version control, to keep track of what changes have been made when. A popular tool is Git (`https://git-scm.com/`) which is often used with hosting services such as GitHub (`https://github.com/`) to maintain online repositories.
- When plotting or saving figures, it is important to record information about how the figure was generated, as working on a neuron model involves adjusting parameters over time, whose values are easily forgotten. For example, save a text version of the parameter file under the same name as the figure.

- Every now and then it may be good to update the structure of the code, or do a code cleanup if necessary. Do not be afraid to allocate time for improving code, it always ends up paying off in time.

Good luck! From Florence Kleberg and Prof. Jochen Triesch