

Organisation of exercises

NX 465

Computational Neuroscience:
Neuronal Dynamics



Kasper Smeets



Shuqi Wang



Louis Pezon

▪ Exercises

- Pen and paper exercises
- Python exercises

▪ Mini-project

- In groups of 2
- Handed out around week 7
- 2 topics to choose from
- Coding + written report (30% of final grade)
- Mandatory!

Pen and paper exercises

- On the content of the lecture
- Partly covered together during class
- Released weekly
- Answers released later in the week
- Exam material!

Exercise 1: Inhibitory rebound

Consider the following two-dimensional Fitzhugh-Nagumo model:

$$\begin{cases} \frac{du}{dt} = u(1-u^2) - w + I \equiv F(u, w) \\ \frac{dw}{dt} = \varepsilon(u - 0.5w + 1) \equiv \varepsilon G(u, w), \end{cases} \quad (1)$$

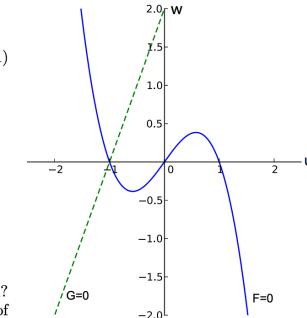
where $\varepsilon \ll 1$.

1.1 Suppose that an inhibitory current step is applied,

$$I(t) = \begin{cases} -I_0 & t \leq 0 \\ 0 & t > 0 \end{cases}$$

How does the fixed point move?

1.2 What happens after the driving current is removed? Sketch the form of the trajectories for increasing values of I_0 . What happens for large I_0 ?



Exercise 2: Phase Plane Analysis

In this exercise, we use the phase plane to study the dynamics of a two dimensional, nonlinear neuron model. The system is described by:

$$\begin{cases} \frac{du}{dt} = F(u, w) \\ \frac{dw}{dt} = G(u, w) \end{cases} \quad (2)$$

where $F(u, w) = f(u) - w + I(t)$ and $G(u, w) = \varepsilon(g(u) - w)$ with $\varepsilon = 0.1$. $I(t)$ is an external current.

- Complementary to the written exercises
- Using pre-implemented functions
- Easy network simulations with Brian2

- We can help with the installation of the neuronal dynamics package

Neuronal Dynamics: Python Exercises

This documentation is automatically generated documentation from the corresponding code repository hosted at [Github](#). The repository contains python exercises accompanying the book [Neuronal Dynamics](#) by Wulfram Gerstner, Werner M. Kistler, Richard Naud and Liam Paninski.

Contents

- [Introduction](#)
 - [Quickstart](#)
 - [Requirements](#)
 - [Disclaimer](#)
 - [Bug reports](#)
- [Exercises](#)
 - [1. Setting up Python and Brian](#)
 - [2. Leaky-integrate-and-fire model](#)
 - [3. The Exponential Integrate-and-Fire model](#)
 - [4. AdEx: the Adaptive Exponential Integrate-and-Fire model](#)
 - [5. Dendrites and the \(passive\) cable equation](#)
 - [6. Numerical integration of the HH model of the squid axon](#)
 - [7. FitzHugh-Nagumo: Phase plane and bifurcation analysis](#)
 - [8. Hopfield Network model of associative memory](#)
 - [9. Type I and type II neuron models](#)
 - [10. Oja's hebbian learning rule](#)
 - [11. Network of LIF neurons \(Brunel\)](#)
 - [12. Spatial Working Memory \(Compte et. al.\)](#)
 - [13. Perceptual Decision Making \(Wong & Wang\)](#)
- [Python exercise modules](#)
 - [neurodynex package](#)
- [License](#)

- Course overview
- Course material:
 - Video lectures
 - Slides
 - Exercises
 - Solutions
 - Announcements

Week 1

LECTURE 1. Introduction: brain vs. computer and a first simple neuron model

- First introduction and overview of the course
- Coding by spikes (action potentials)
- Model of a passive membrane
- Leaky integrate-and-fire model
- Nonlinear integrate-and-fire model
- Quality of integrate-and-fire models: comparison with experiments



Slides Lecture 1 - Integrate-and-Fire neurons



Question Set 1



Python Exercise: LIF



Suggested Reading

- Q&A and discussion forum
- Ask questions and answer other students
- Moderated by TAs



Ed Discussion NX-465

Link on Moodle!

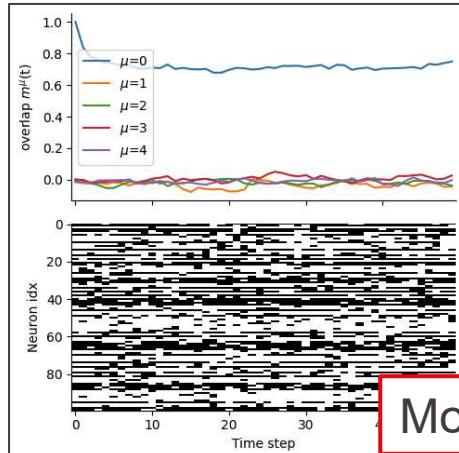
Use the Ed forum for questions and discussions. Updates will be posted here

Mini-projects

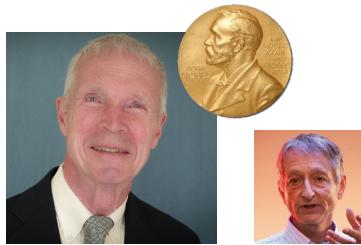
- In pairs!
- Choose from two projects:

Bioplausible Hopfield model

Model of memory storage and recall with biological constraints



More details will follow!



Dynamic Hopfield model

Model of information propagation through cyclical dynamics

