# Welcome to BIO-210

## Applied software engineering for life sciences

December 1st 2025 – Lecture 12

Prof. Alexander MATHIS

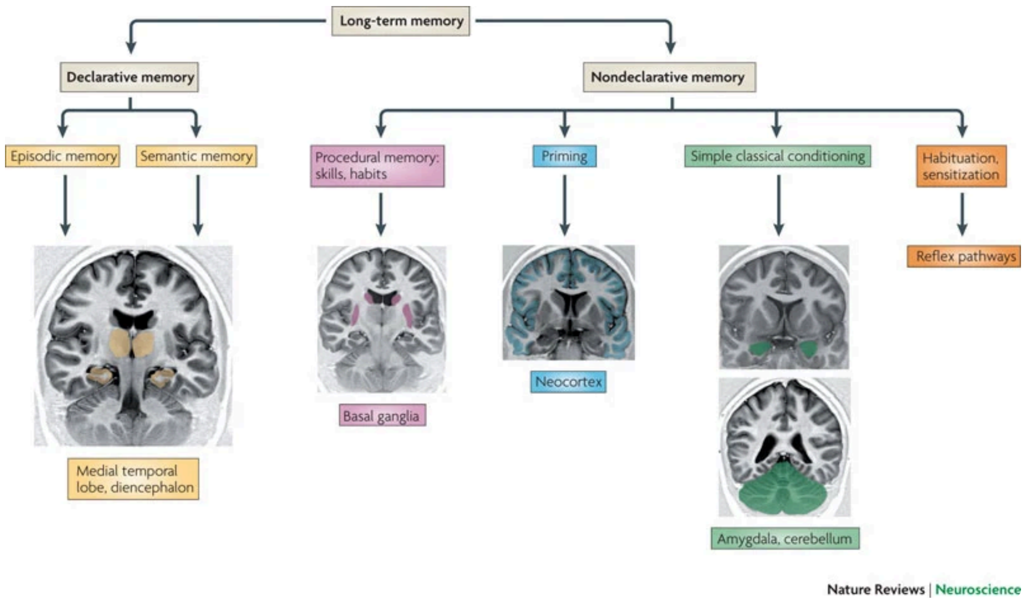
EPFL

| | Date | Topic | Software version | Software releases | Feedback |
|---|---|---|---|---|---|
| 0 | 08/09/2025 | Python introduction I | | | |
| 1 | 15/09/2025 | Python introduction II | | | |
| 2 | 22/09/2025 | Public holiday | | | |
| 3 | 29/09/2025 | Git & GitHub (+ installation) | | | |
| 4 | 06/10/2025 | Project introduction | v1 | | |
| 5 | 13/10/2025 | Functionify | v2 | v1 | |
| 6 | 20/10/2025 | EPFL fall break | | | |
| 7 | 27/10/2025 | Visualization & documentation | v3 | **v2** | code review (API) |
| 8 | 03/11/2025 | Unit-tests, functional tests | v4 | v3 | |
| 9 | 10/11/2025 | Code refactoring | v5 | v4 | graded (tests) |
| 10 | 17/11/2025 | Profiling & code optimization | v6 | v5 | code review |
| 11 | 24/11/2025 | Object oriented programming | v7 | v6 | graded (speed) |
| 12 | 01/12/2025 | Model analysis & project report | v8 | v7 | code review (OO) |
| 13 | 08/12/2025 | Work on project (no class) | | | |
| 14 | 15/12/2025 | Wrap up | | v8 | graded (project) |

# Model analysis and project reports

- Hopfield network
- Turing pattern formation

# Memory




Source: K Henke, Nature Reviews Neuroscience 2010

- memory serves different functions (see diagram)
- different types of memory are supported by different brain areas (how do we know this?)

# Project: Hopfield networks

*Proc. Natl. Acad. Sci. USA*
Vol. 79, pp. 2554–2558, April 1982
Biophysics

## Neural networks and physical systems with emergent collective computational abilities

(associative memory/parallel processing/categorization/content-addressable memory/fail-soft devices)

J. J. Hopfield

Division of Chemistry and Biology, California Institute of Technology, Pasadena, California 91125; and Bell Laboratories, Murray Hill, New Jersey 07974

*Contributed by John J. Hopfield, January 15, 1982*

**ABSTRACT** Computational properties of use to biological organisms or to the construction of computers can emerge as collective properties of systems having a large number of simple equivalent components (or neurons). The physical meaning of content-addressable memory is described by an appropriate phase space flow of the state of a system. A model of such a system is given, based on aspects of neurobiology but readily adapted to integrated circuits. The collective properties of this model produce a content-addressable memory which correctly yields an entire memory from any subpart of sufficient size. The algorithm for the time evolution of the state of the system is based on asynchronous parallel processing. Additional emergent collective properties include some capacity for generalization, familiarity recognition, categorization, error correction, and time sequence retention. The collective properties are only weakly sensitive to details of the modeling or the failure of individual devices.

calized content-addressable memory or categorizer using extensive asynchronous parallel processing.

**The general content-addressable memory of a physical system**

Suppose that an item stored in memory is "H. A. Kramers & G. H. Wannier *Phys. Rev.* **60,** 252 (1941)." A general content-addressable memory would be capable of retrieving this entire memory item on the basis of sufficient partial information. The input "& Wannier, (1941)" might suffice. An ideal memory could deal with errors and retrieve this reference even from the input "Vannier, (1941)". In computers, only relatively simple forms of content-addressable memory have been made in hardware (10, 11). Sophisticated ideas like error correction in accessing information are usually introduced as software (10).

There are classes of physical systems whose spontaneous behavior can be used as a form of general (and error-correcting)

Link to the original publication

# Neurons


Source: neuron drawing by Ramón y Cajal

# A simple, computational neuron model

Our computational neuron model is subject to the following simplifications:

- binary neurons, linear integration and thresholding ($\sigma$)

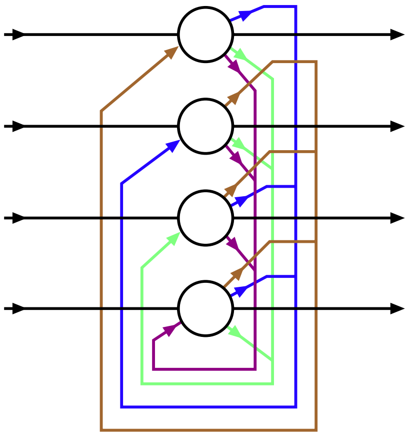
Specifically, the activity of neuron $j$ is given by

$$p_j = \sigma\left(\sum_i \mathbf{W}_{j,i} \cdot \mathbf{p}_i\right)$$

where $\mathbf{W}$ is the weight matrix and the nonlinearity $\sigma(x)$ is defined as

$$\sigma(x) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

# The Hopfield network is a recurrent neural network



At time $t+1$ the activity is given by the weight matrix and the activity at $t$.

$$\mathbf{p}^{(t+1)} = \sigma\left(\mathbf{W}\mathbf{p}^{(t)}\right)$$

# How are memories stored?

Consider $M$ memory patterns $\mathbf{p}^\mu \in \{-1, 1\}^N, \mu \in \{1, ..., M\}$.

The Hebbian learning rule is the simplest one to create a Hopfield network. It is based on the principle "Neurons that fire together, wire together. Neurons that fire out of sync, fail to link". This translates into the following network connectivity:

$$w_{ij} = \frac{1}{M} \sum_{\mu=1}^{M} p_i^\mu p_j^\mu \quad i \neq j$$

We can observe that the weight is the average of the contribution of each pattern to the synaptic weight.

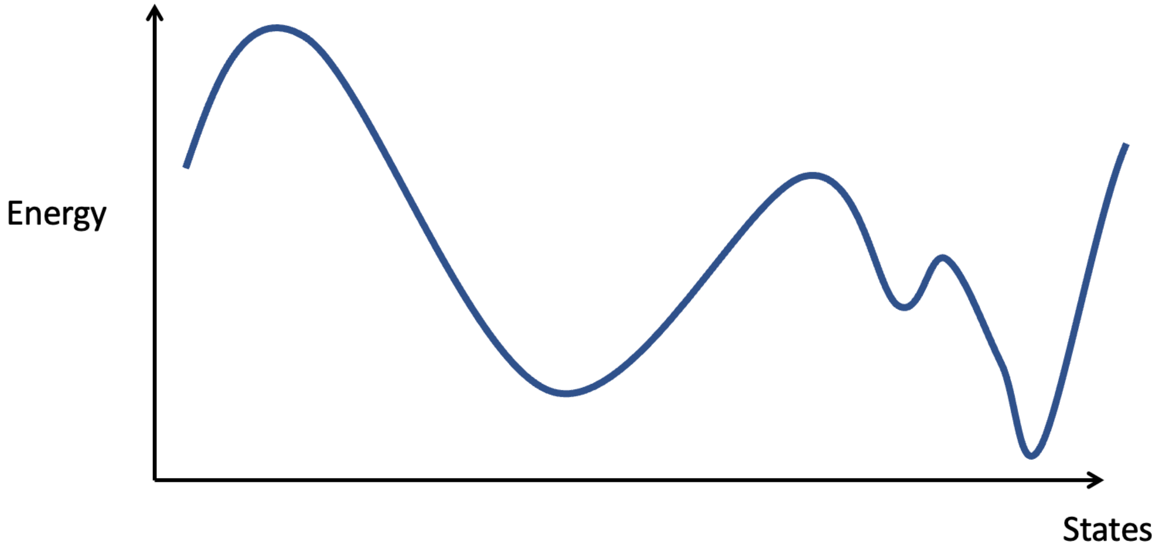
Each pattern contributes positively to a certain weight, if the state of the two connected neurons is the same, and negatively otherwise.

You also implemented a different learning rule, the Storkey learning rule. It is also a local, and incremental learning rule (with higher capacity, as you will show).

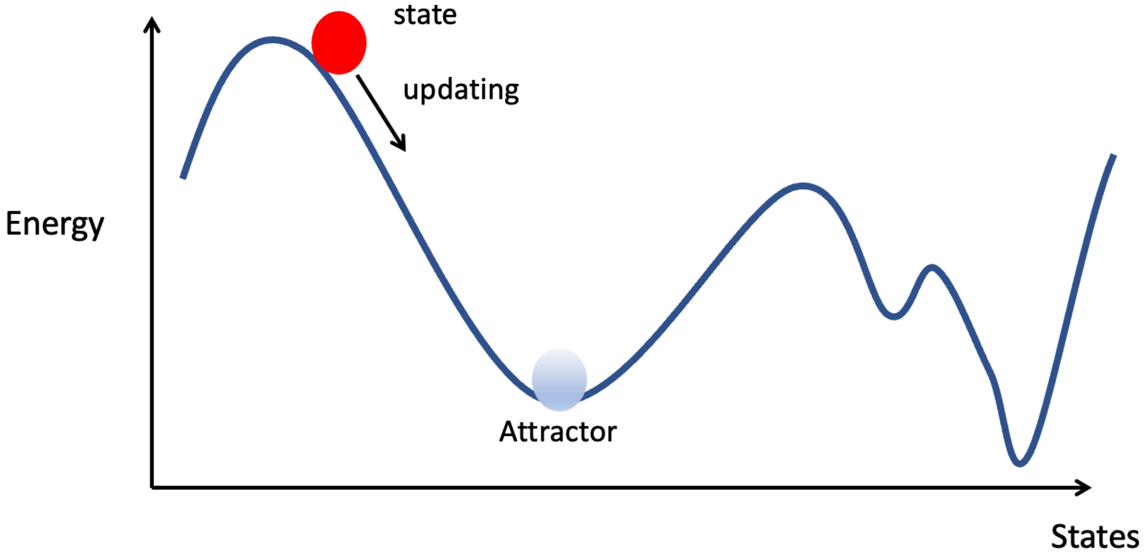
# Project goals

- implement Hopfield nets for a given set of memories (dynamics and learning)
- *understand the Hopfield network*
- *study dynamics, memory capacity (incl. visualization)*
- integrate tests, provide specified API
- refactor (functions, oop, speed-up)
- developed on GitHub with git

# Hopfield network



# Hopfield network




- stored patterns are local minima of the energy surface
- spurious minima also exist (e.g., $-\mu$ for pattern $\mu$)
- the type and *number* of patterns shape the energy landscape!

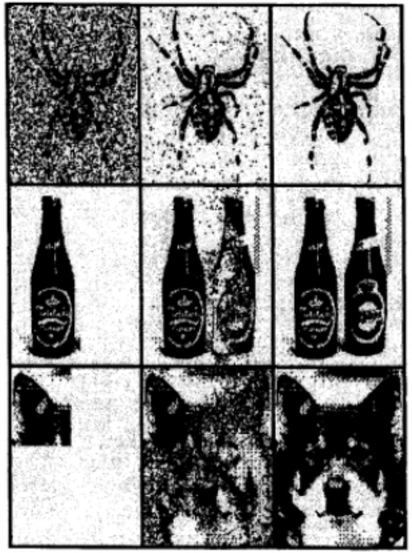
# Hopfield networks project

You will analyze the model and answer two key questions:

- How robust are stored memories?
- How many memories can be stored (incl. correlated memories)?

# Associative memory (for binary images of 130x180)

Left: initial state, middle: intermediate and right: final state for 3 examples.



**Try it out for your model!**

From Introduction To The Theory Of Neural Computation by J. Hertz

# Project: Morphogenesis



## THE CHEMICAL BASIS OF MORPHOGENESIS

By A. M. TURING, F.R.S. *University of Manchester*

(*Received* 9 *November* 1951—*Revised* 15 *March* 1952)

It is suggested that a system of chemical substances, called morphogens, reacting together and diffusing through a tissue, is adequate to account for the main phenomena of morphogenesis. Such a system, although it may originally be quite homogeneous, may later develop a pattern or structure due to an instability of the homogeneous equilibrium, which is triggered off by random disturbances. Such reaction-diffusion systems are considered in some detail in the case of an isolated ring of cells, a mathematically convenient, though biologically unusual system. The investigation is chiefly concerned with the onset of instability. It is found that there are six essentially different forms which this may take. In the most interesting form stationary waves appear on the ring. It is suggested that this might account, for instance, for the tentacle patterns on *Hydra* and for whorled leaves. A system of reactions and diffusion on a sphere is also considered. Such a system appears to account for gastrulation. Another reaction system in two dimensions gives rise to patterns reminiscent of dappling. It is also suggested that stationary waves in two dimensions could account for the phenomena of phyllotaxis.

The purpose of this paper is to discuss a possible mechanism by which the genes of a zygote may determine the anatomical structure of the resulting organism. The theory does not make any new hypotheses; it merely suggests that certain well-known physical laws are sufficient to account for many of the facts. The full understanding of the paper requires a good knowledge of mathematics, some biology, and some elementary chemistry. Since readers cannot be expected to be experts in all of these subjects, a number of elementary facts are explained, which can be found in text-books, but whose omission would make the paper difficult reading.

### 1. A model of the embryo. Morphogens

Link to the original publication

# Reaction-diffusion equations

The Turing pattern formation mechanism can be modeled with two time-dependent reaction-diffusion equations, describing the evolution and the interaction between an activator $u(x, y)$ and an inhibitor $v(x, y)$ morphogen.

$$\frac{\partial u}{\partial t} = \gamma f(u, v) + \nabla^2 u$$

$$\frac{\partial v}{\partial t} = \gamma g(u, v) + d\nabla^2 v$$

with diffusion terms $\nabla^2 u$ and $\nabla^2 v$ as well as reaction terms $\gamma f(u, v)$ and $\gamma g(u, v)$.

See problem set for more details.

# You’ll implement discretized equations

Reaction-diffusion equations:

$$u_{ij}^{(t+1)} = u_{ij}^{(t)} + \Delta t \left[ \gamma f(u_{ij}^{(t)}, v_{ij}^{(t)}) + \frac{u_{i-1,j}^{(t)} + u_{i+1,j}^{(t)} + u_{i,j-1}^{(t)} + u_{i,j+1}^{(t)} - 4u_{ij}^{(t)}}{\Delta x^2} \right]$$

and

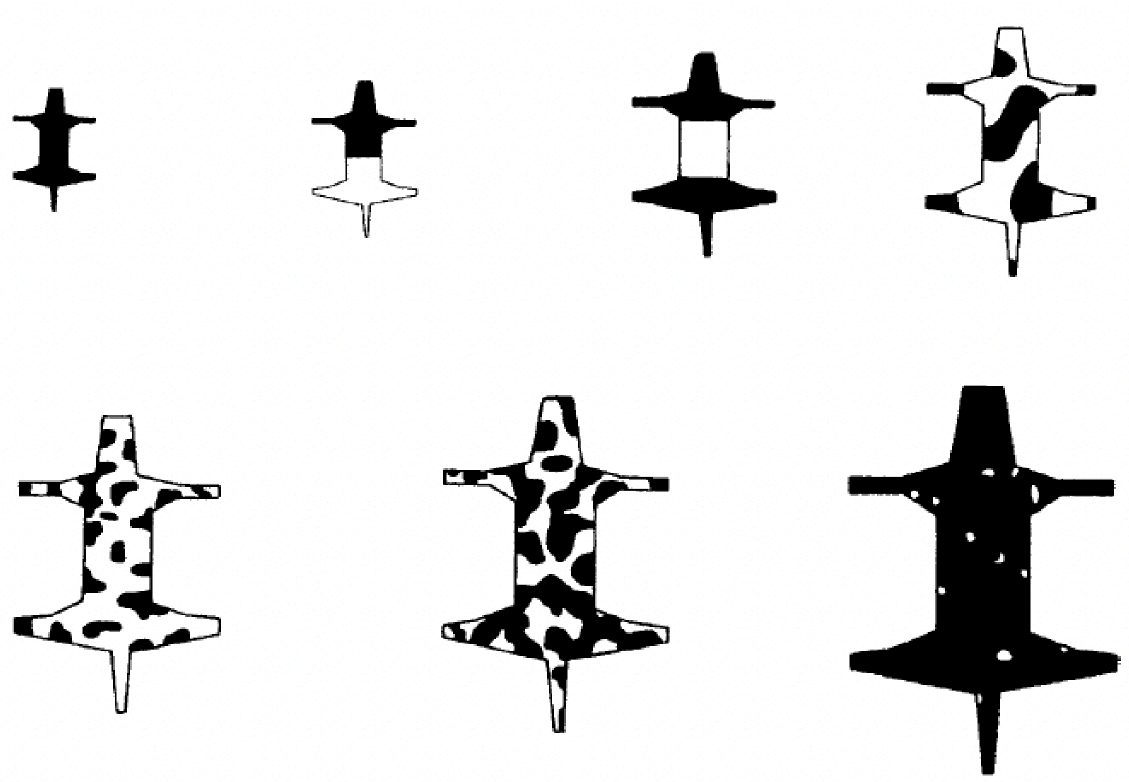
$$v_{ij}^{(t+1)} = v_{ij}^{(t)} + \Delta t \left[ \gamma g(u_{ij}^{(t)}, v_{ij}^{(t)}) + d \frac{v_{i-1,j}^{(t)} + v_{i+1,j}^{(t)} + v_{i,j-1}^{(t)} + v_{i,j+1}^{(t)} - 4v_{ij}^{(t)}}{\Delta x^2} \right]$$

with parameters defined on the problem set.

# Morphogenesis project

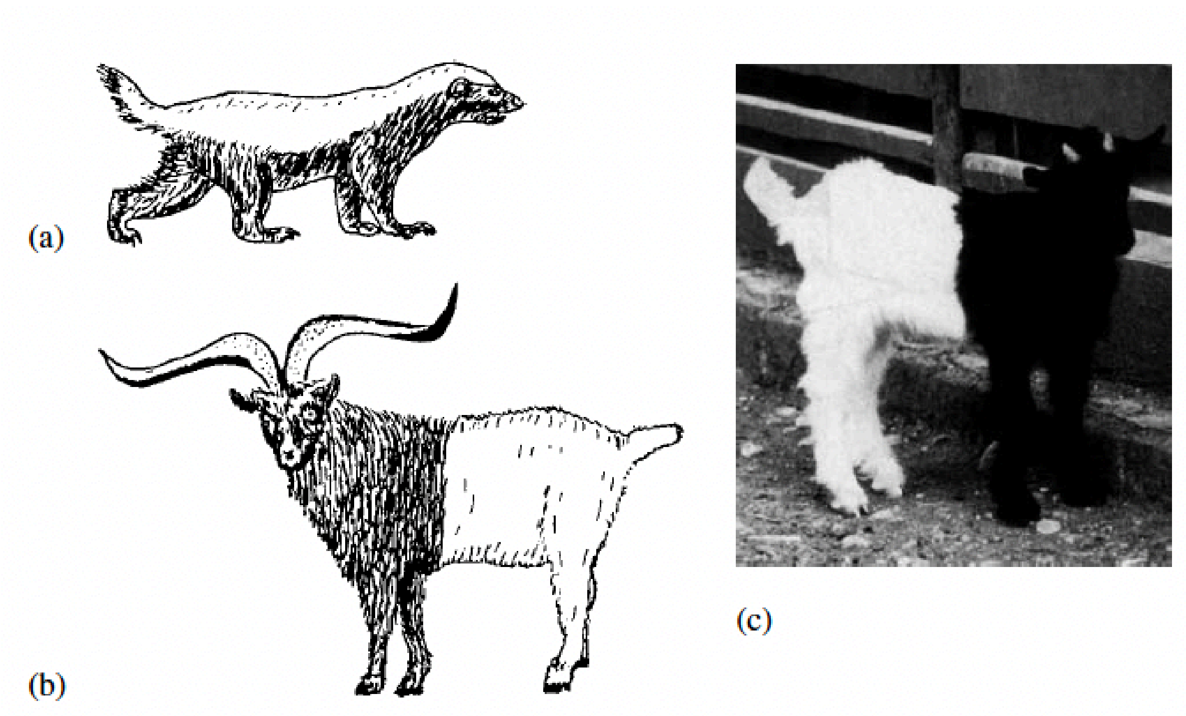
- implement a model of morphogenesis (discretized partial differential equation)
- integrate tests, provide specified API
- refactor (functions, oop, speed-up)
- developed on GitHub with git
- *understand why patterns can emerge with this mechanism*
- *study what patterns can emerge (incl. visualization)*

# Reaction-diffusion model



Adapted from Mathematical Biology II by Murray

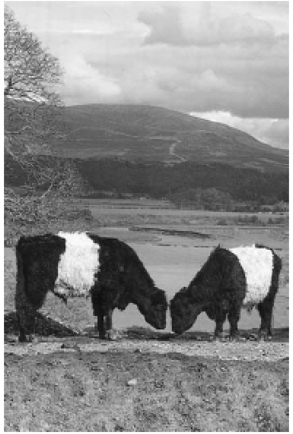
# The simplest, non-trivial coat patterns




Honey badger & valais goat; adapted from Mathematical Biology II by Murray

# The next bifurcation …

(d)



Anteater and galloway cows; adapted from Mathematical Biology II by Murray

# Reaction-diffusion model

You will analyze the model and answer two key questions:

- When is the solution numerically stable?
- What type of coat patterns can emerge? You will also use Fourier analysis to analyze how the patterns depend on the parameters.

# How to document your work?

- write the report in markdown (on GitHub)
- store results as pandas files
- embed visualizations and results
- make your final release before Dec 23 at 10am

The final problem set gives all details.

Note: we will not grade speed/coverage in this release. This was already done. Of course, put forward your best code version with all features working. Use the fastest code, so that all simulations are achieved quicker. All the best!

# Questions?

# Upcoming schedule:

Today:

- Monday 13 - 15: exercises working on your project (final version)
- no office hours this and next week!

Next weeks:

- Monday, Dec 8: *NO* class (so you can work on your project)
- Monday, Dec 8, 13 - 15: exercises working on your project
- Monday, Dec 15: final class (incl. award ceremony)
- Monday, Dec 15, 13 - 15: exercises working on your project

Release your final version on 23.12. at 10am!