CN2 1: Introduction

Paul Gribble

http://gribblelab.org

Sep 10, 2012

Administrivia

- Class meets Mondays, 2:00pm - 3:30pm and Thursdays, 11:30am - 1:00pm, in NSC 245A
- Contact me with any questions or to set up a meeting:
  - `paul@gribblelab.org`
  - Office: NSC 228 Lab: NSC 245G
- Materials will be posted on the course website: `http://www.gribblelab.org/compneuro/`

Grading

Component	Grade
Assignments	50%
Written Report	20%
Oral Presentation	20%
Participation	10%
Total	100%

Topics

- Modelling Dynamical Systems
- Modelling Action Potentials
- Musculoskeletal Models
- Perceptrons
- Multi-Layer Networks
- Recurrent Networks
- Stochastic Networks
- Unsupervised Learning
- Computational Motor Control

Useful Background Knowledge

- Calculus
- Linear Algebra
- Basic Probability Theory
- Differential Equations
- Computer programming
- **No Fear!**

Let me know if you need a refresher on any topics, I can provide resources to help you catch up.

Readings

- Selected readings from a number of books
- Selected original research articles

Software

We will be using **Python**: `http://www.python.org/`

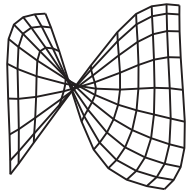
- also the add-on library **SciPy** `http://www.scipy.org/`
- free
- open-source
- available for all operating systems
- widely used

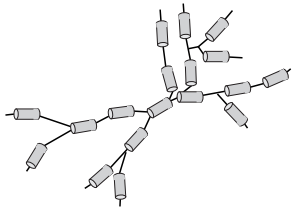
What is Computational Neuroscience?

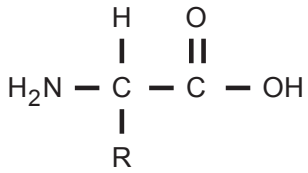
CN borrows methods and ways of thinking from

- mathematics
- physics
- computer science
- statistics
- machine learning
- psychology
- physiology
- neuroscience

Levels of Abstraction







1 m — CNS

10 cm — System

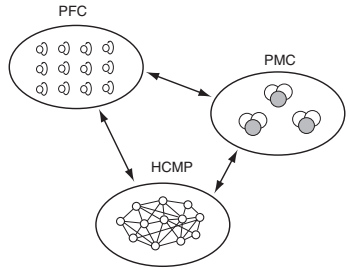
1 cm — Maps

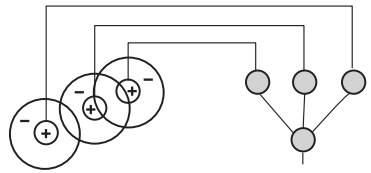
1 mm — Networks

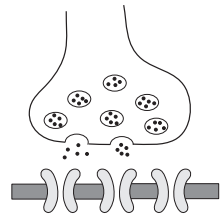
100 µm — Neurons

1 µm — Synapses

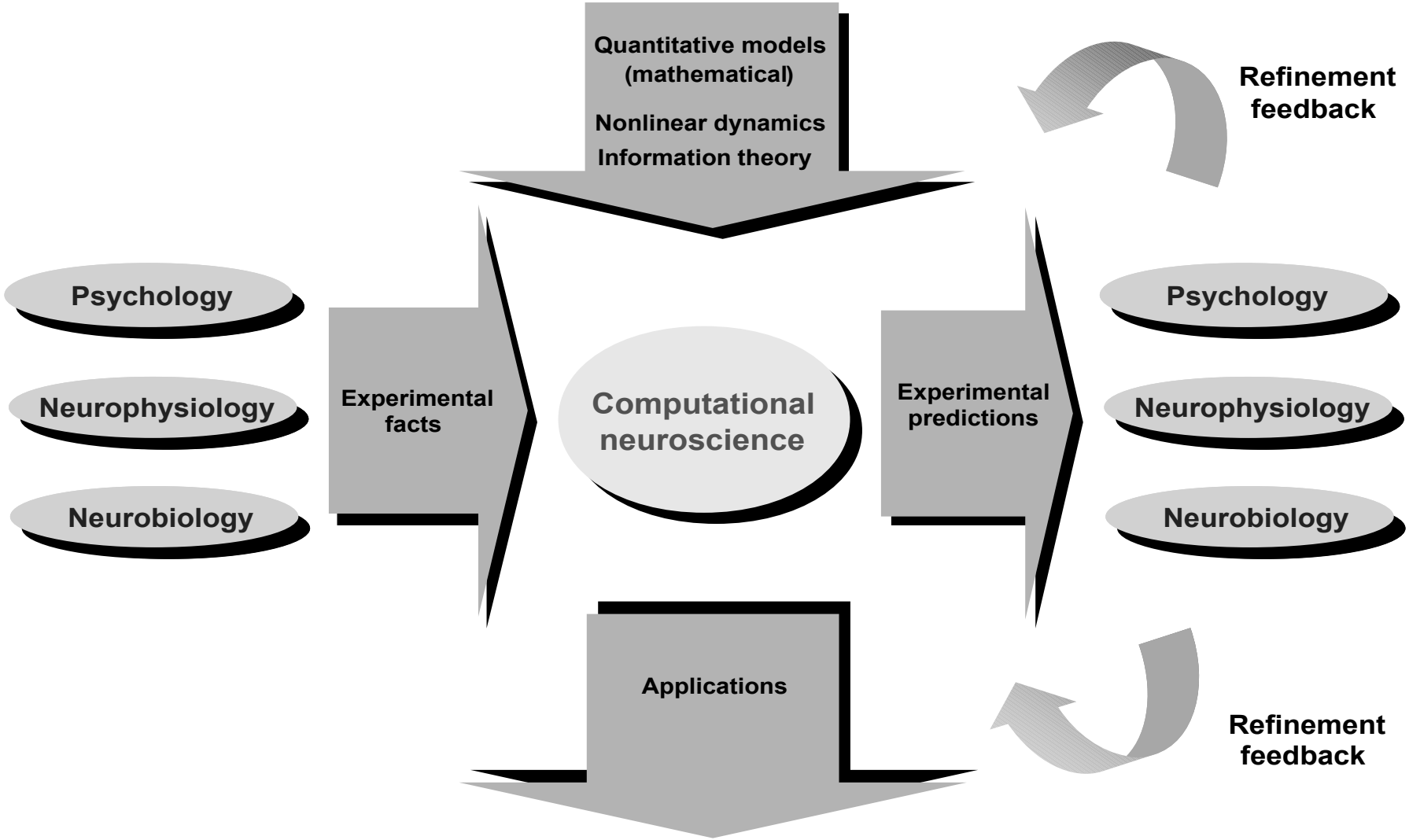
1 Å — Molecules







Integration



What is a Model?

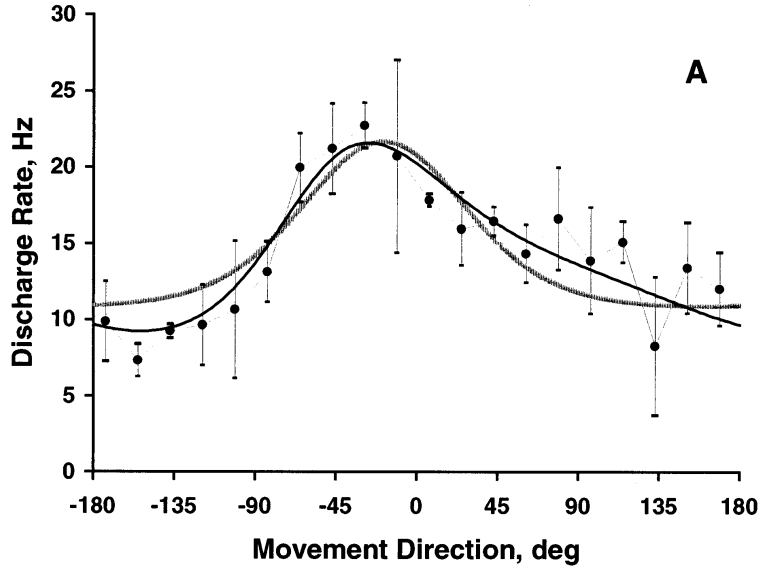
- abstraction
  - e.g. architectural model
- concrete implementation
  - express as equations
  - simulated on a computer
  - generate quantitative predictions
- test (not confirm) hypothesis

Types of Models

Descriptive Models

phenomenological functional description characterization of data curve-fitting

Cosing Tuning of M1 Neurons



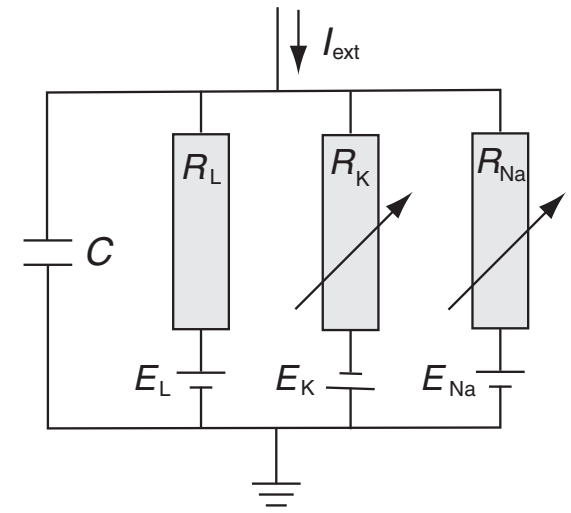
$$d(M) = b + kcos\theta_{CM}$$

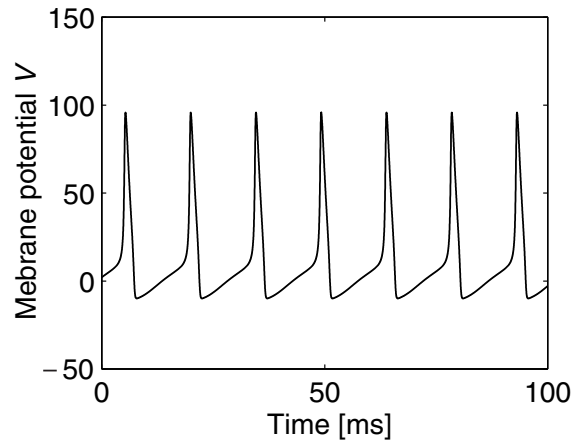
Types of Models

Explanatory Models

explain complex functions in terms of interactions of subordinate (simplified) components

Hodgkin-Huxley Model





Serial vs Parallel Processing

- **Computers**
  - small # of very powerful CPUs
  - CPUs execute complex binary instructions
  - Gflops (billions of operations per sec)
- **Nervous Systems**
  - Many simple processing units
  - millions (billions?) of simple integrate-and-fire units
  - out of rich connectivity emerges complex computation

Parallel Distributed Processing

- artificial neural networks
- connectionist models
- interactions between units enables processing abilities not present in single units
- emergent properties
- simple rules lead to complex behaviour
- “knowledge” emerges that is not explicitly specified in system
- networks learn from experience

Neural Networks: Properties

- Generalization
- Nonlinearity
- Parallelism
- Gradedness
- Contextual Processing
- Adaptivity
- Graceful Degradation

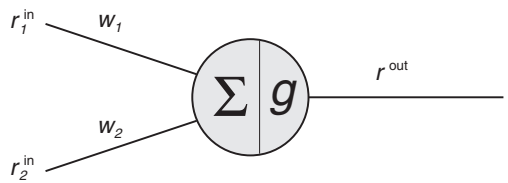
Neural Networks: Applications

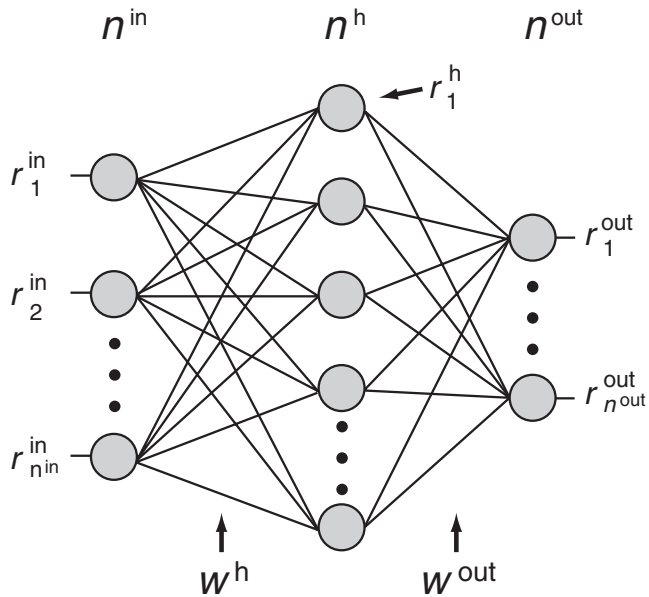
- **Aerospace**: aircraft autopilot systems
- **Banking**: OCR (e.g. cheques); credit evaluation; credit card fraud
- **Financial**: real estate appraisal; currency/equity price prediction
- **Military**: sonar object classification; image identification; who knows what else
- **Industrial**: process control; equipment failure prediction
- **Medical**: diagnosis; screening

Computing with Neurons

McCulloch & Pitts (1943)

- simple binary units
- proved: with sufficient # of small all-or-none units, and synaptic connections *set to appropriate weights*, a network can compute **any computable function**

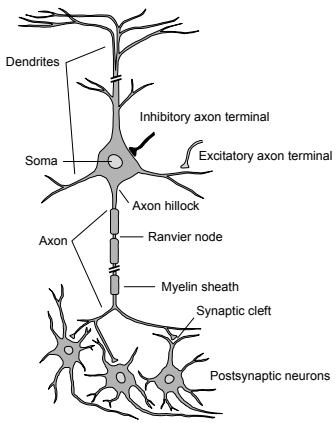


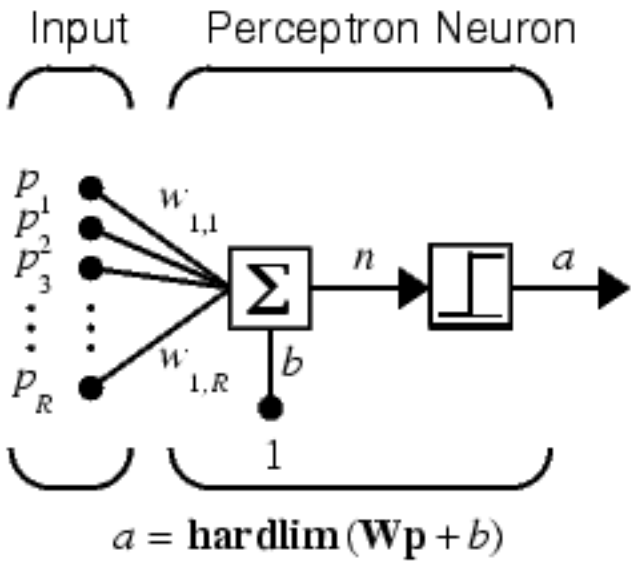


Perceptron

Rosenblatt (1960)

- single-layer networks, given a **simple local learning rule** are guaranteed to converge on appropriate weights
- perceptrons compute mappings from one space to another
- many to one mappings





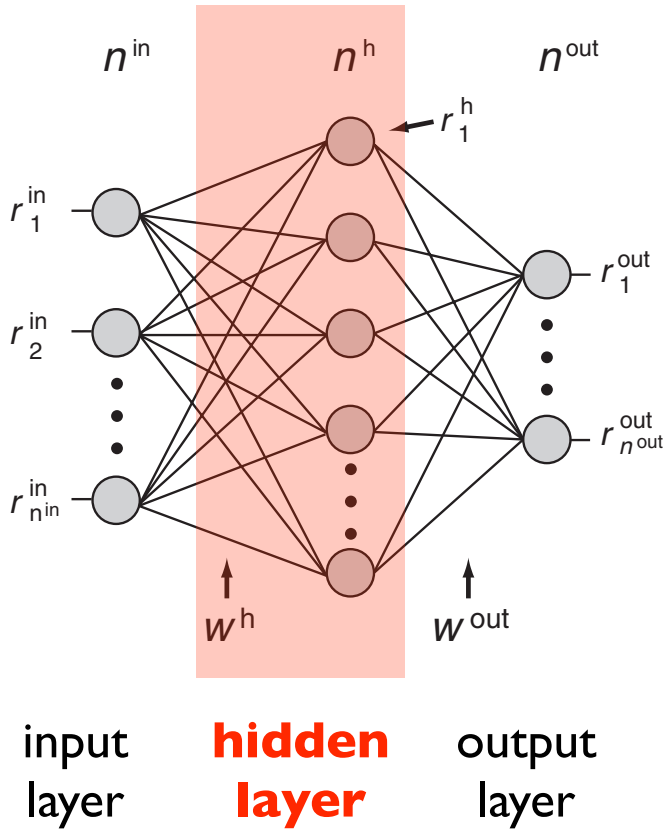
$$a = \mathbf{hardlim}(\mathbf{Wp} + b)$$

Where

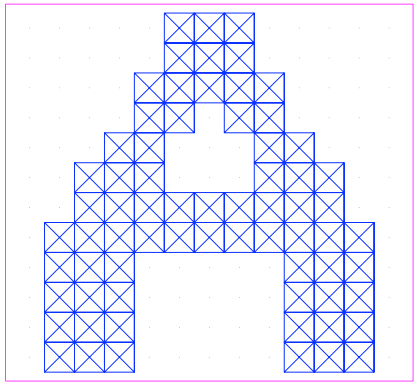
R = number of elements in input vector

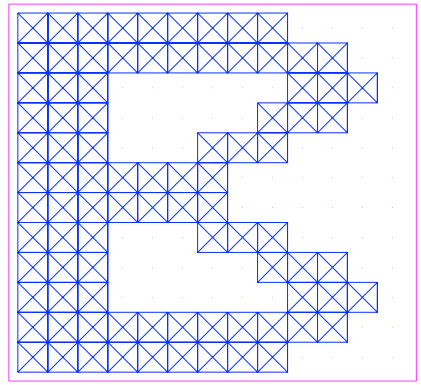
Multi-Layer Networks

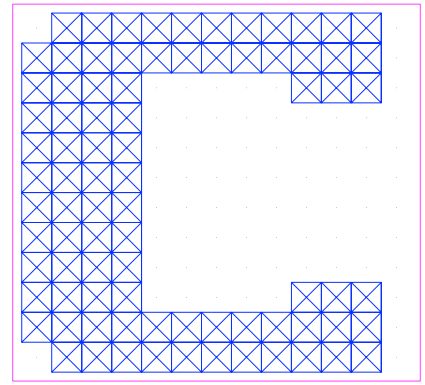
- single-layer perceptrons are limited to problems that are *linearly separable*
- multi-layer networks given enough units and appropriate weights can compute any mapping
- learning rule called *backpropagation* allows weights to adapt

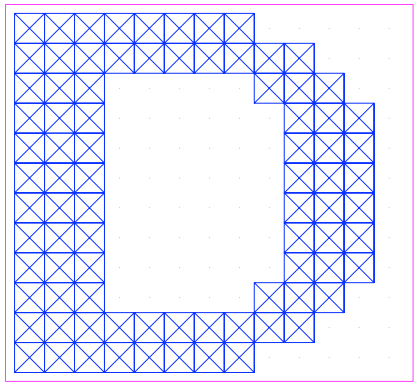


Multi-Layer Networks

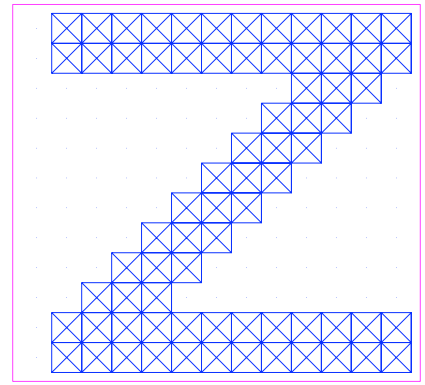




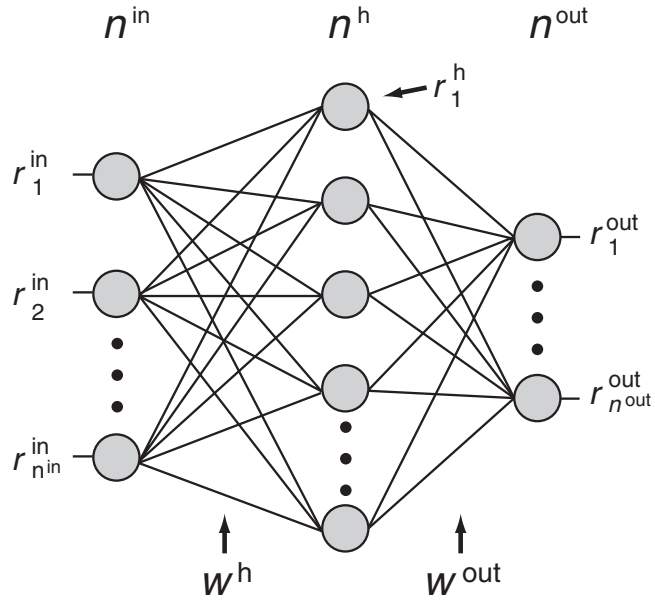




each character
is a
12 x 13 matrix



Multi-Layer Networks



12 x 13 = 156 input neurons

26 neurons intermediate layer

26 neurons output layer

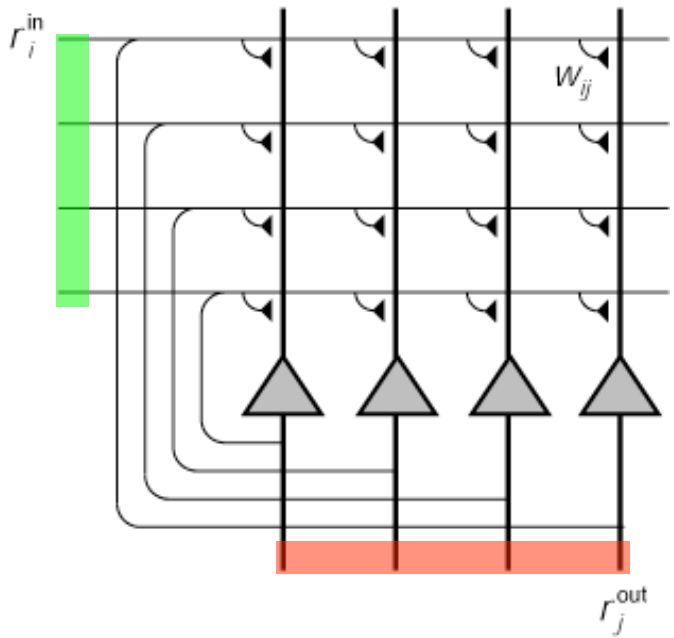
total # synaptic weights
= (156*26)+26 + (26*26)+26
= 4,784

Unsupervised Learning

- **Supervised Learning** requires a training set of input-output examples and a detailed *teaching signal*
- **Unsupervised Learning** approaches allow for learning about properties of the world by exposure only to inputs (exemplars)

Hopfield Networks

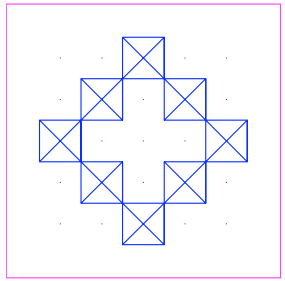
- outputs fully connected to inputs
- auto-associative memory
- network updates dynamically over time, converges on equilibrium states

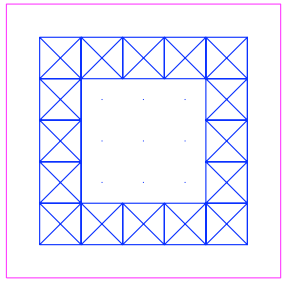


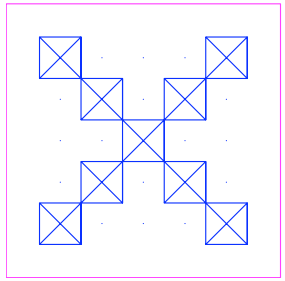
train a network on four 5x5 patterns

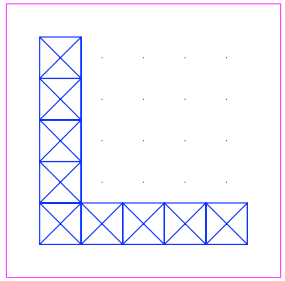
each 5x5 pattern is represented by a vector of length 25

there are 25x25 = 625 weights





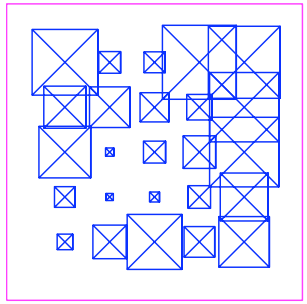




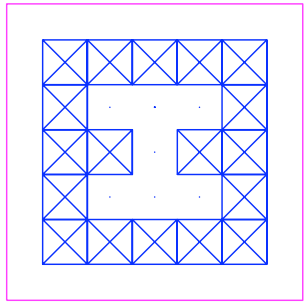
Hopfield Network

noisy input

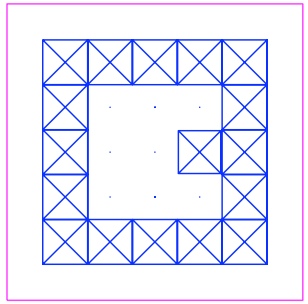
ITERATION 1



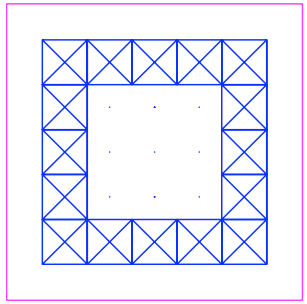
ITERATION 2



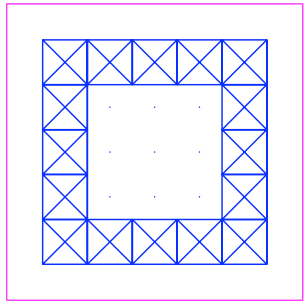
ITERATION 3



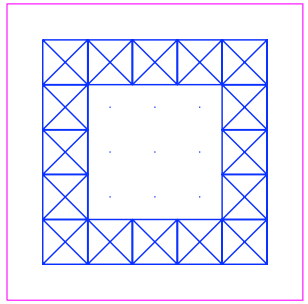
ITERATION 4



ITERATION 5



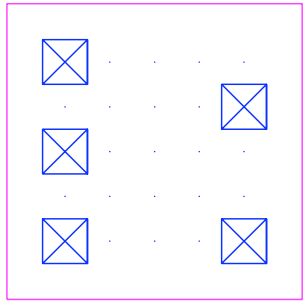
ITERATION 6



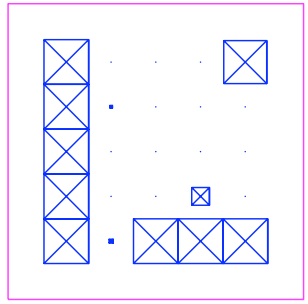
Hopfield Network

partial input

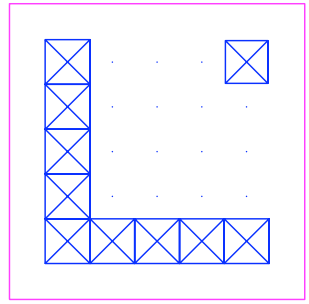
ITERATION 1



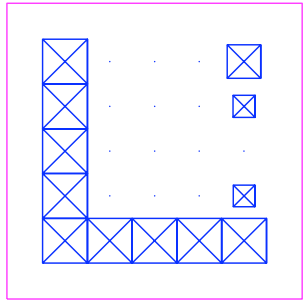
ITERATION 2



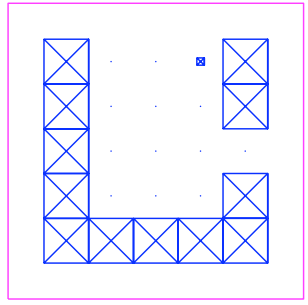
ITERATION 3



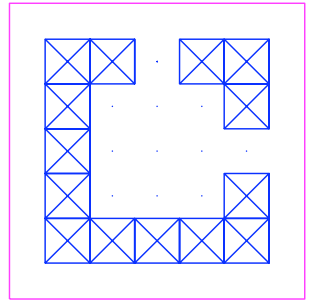
ITERATION 4



ITERATION 5



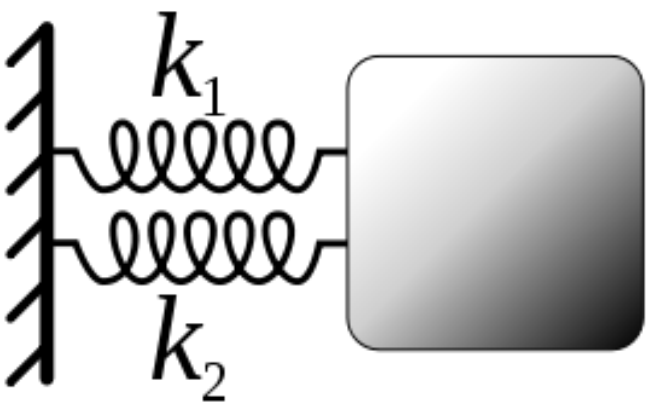
ITERATION 6

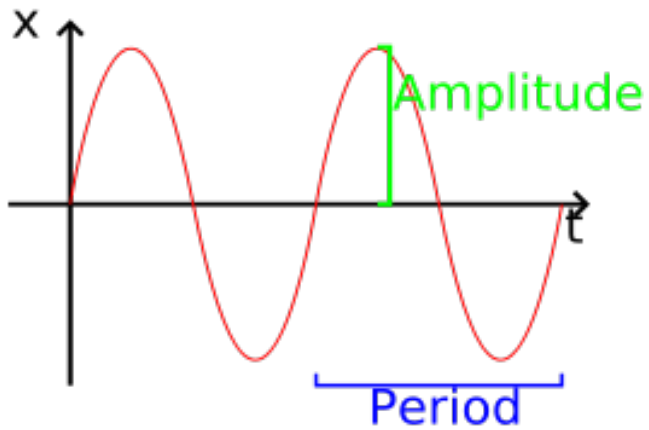


Musculoskeletal & Neuromuscular Models

Sensory-Motor Control

- in order to understand what neural control signals to muscles look like, **it's necessary to have a detailed account of the neuromuscular plant**
- imagine you were studying the control inputs to a mechanical system containing springs, but you didn't know that springs exert force in response to mechanical deformation
- what would you conclude about the control signals?





Musculoskeletal & Neuromuscular Models