Karlsruhe Institute of Technology

Spiking Neural Networks
Gepulste Neuronale Netze

M. Sc. Juan Camilo Vasquez Tieck

M. Sc. Jacques Kaiser

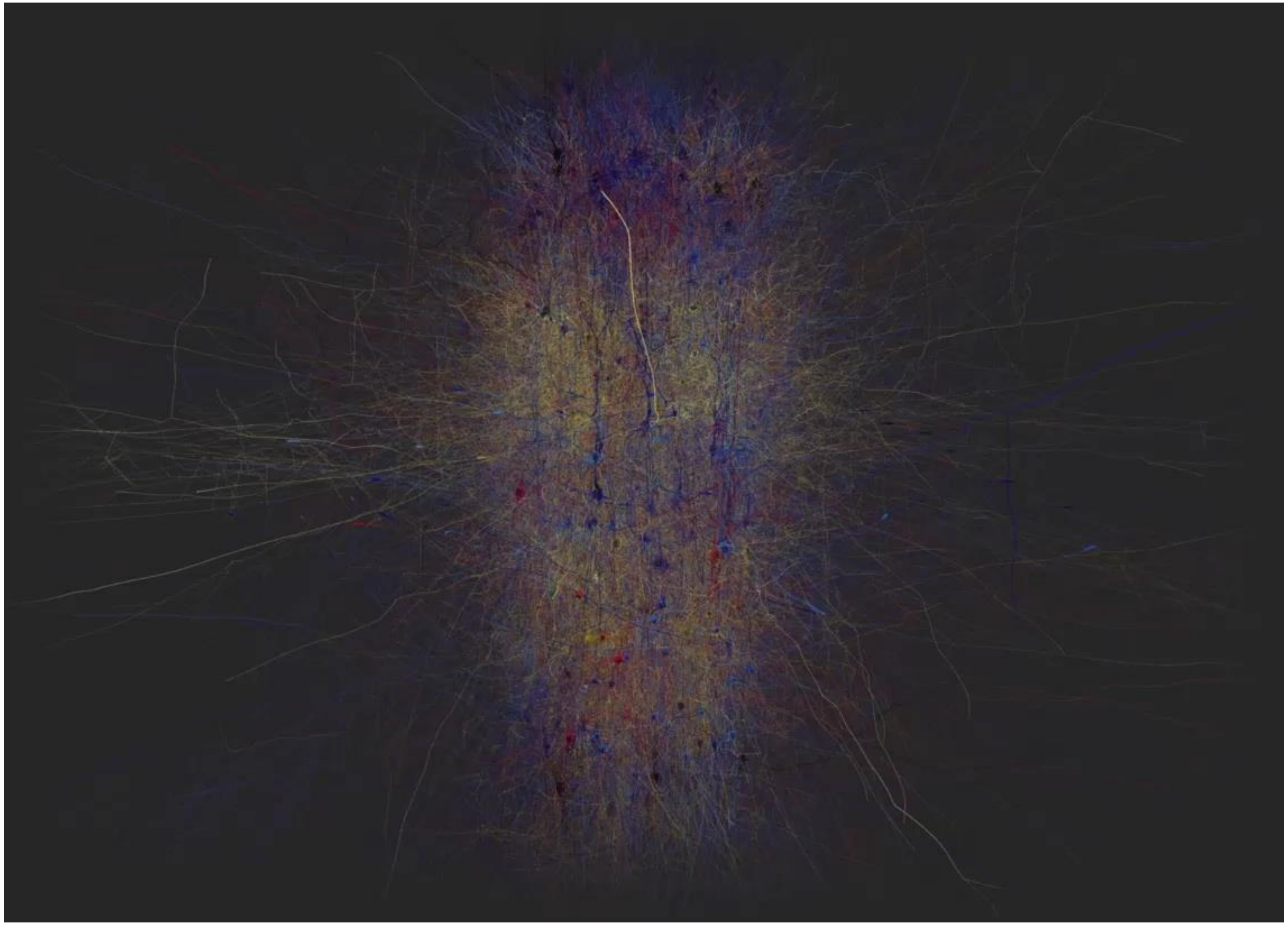
Dr.-Ing. Stefan Ulbrich

Prof. Dr.-Ing. J. Marius Zöllner

Prof. Dr. R. Dillmann

Motivation - what is intelligence?

Motivation – where it all happens

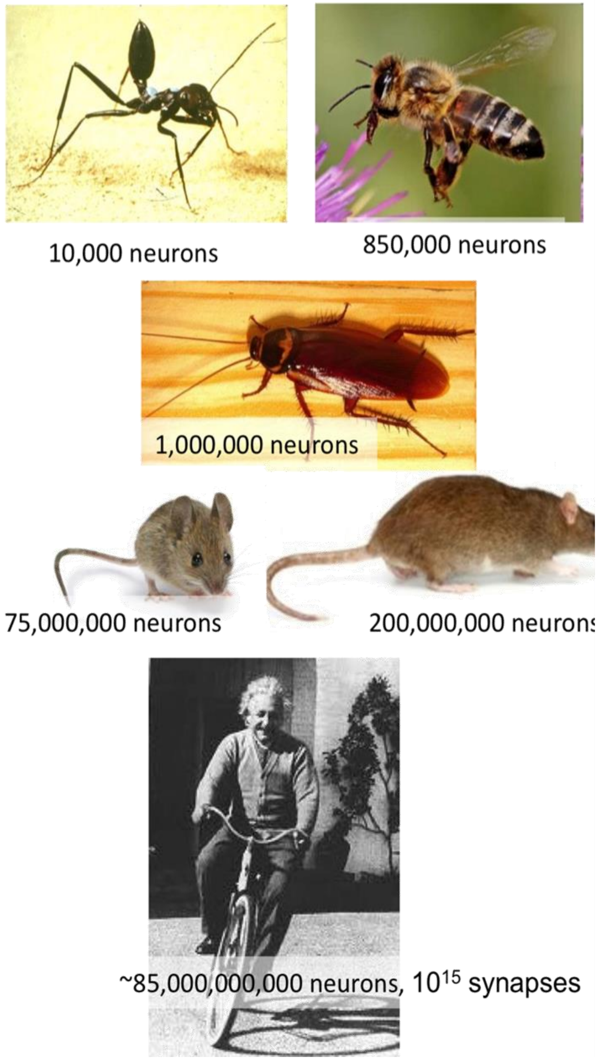


Human Brain Project

Blue Brain Project

EPFL-BBP. 2014. Digital Reconstruction of Neocortical Microcircuitry.

Neural control in biology



- Embodied algorithms
- Sensor processing and actuation in real-time
- Massively parallel computation
  - No deadlocks, non-determinism, race-conditions
- Self-organized
- Scalability
- Robustness (fault and noise tolerance)
- Extreme energy and space efficiency
  - („peta-flop computation with 20 Watt consumption“)

Robots could largely benefit from brains

A global initiative

Research projects

- **U.S.:** BRAIN Initiative, >$1 billion
- **Europe:** Human Brain Project, $1 billion
- **Japan:** Brain/MINDS, ~$300 million

10 years project

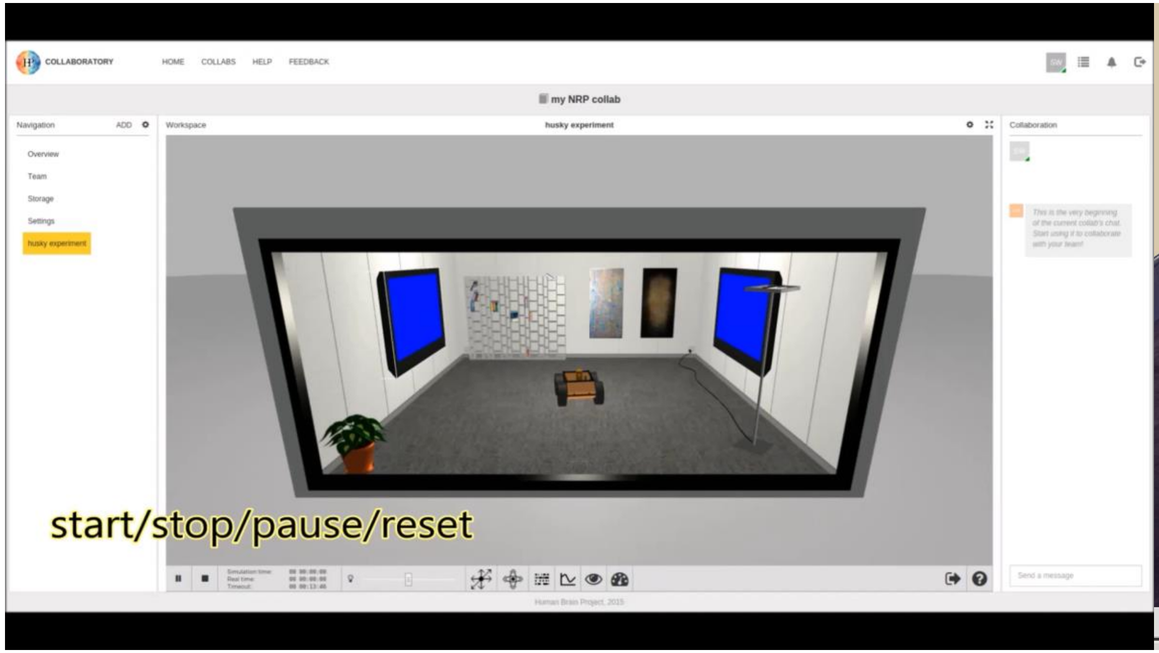
Ray Kurzweil (Google, Director of engineering) expects the singularity in 2045

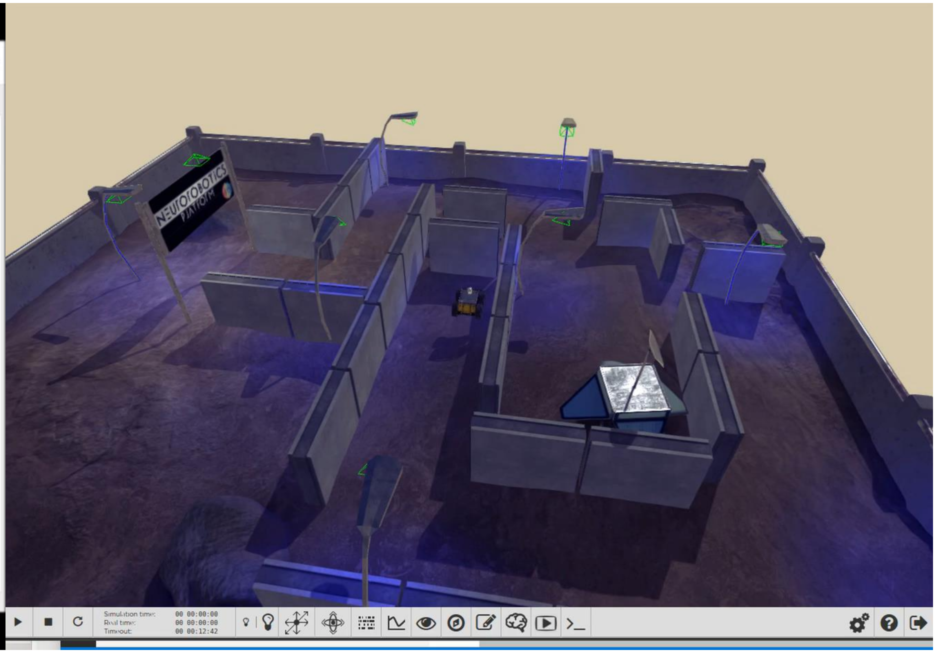
Companies

- OpenAI >$1 billion
- Deepmind 400 employees
- Google brain, IBM, NVIDIA, Numenta, many startups…

The Human Brain Project at FZI

Sub-project 10 - Neurorobotics





Possibilities of Hiwi, master thesis, **international** thesis, …

Institut für Anthropomatik und Robotik (IAR)
M. Sc. Jacques Kaiser
Maschinelles Lernen II

Fakultät für **Informatik**

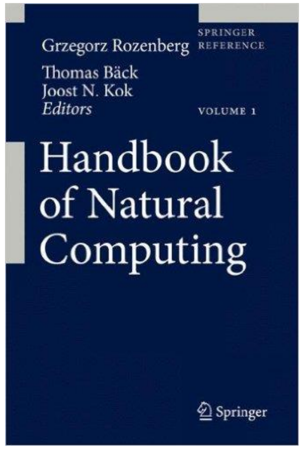
Selected literature

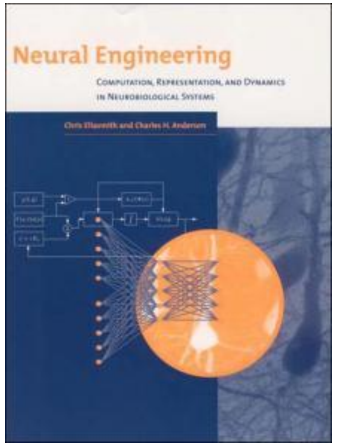
- H. Paugam-Moisy and S. Bohte:
  “*Computing with Spiking Neurons*“
  in Handbook of Neural Computing,
  Springer Verlag, 2012

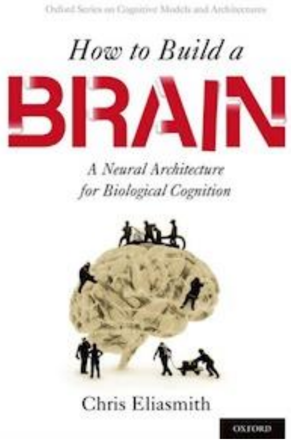
- C. Eliasmith and C. H. Anderson:
  “Neural Engineering“, MIT Press, 2003

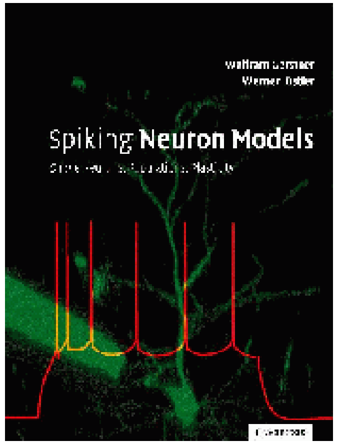
- C. Eliasmith: *“How to build a brain“*,
  Oxford University Press, 2013

- W. Gerstner and W. M. Kistler:
  “Spiking Neuron Models“,
  Cambridge University Press, 2002







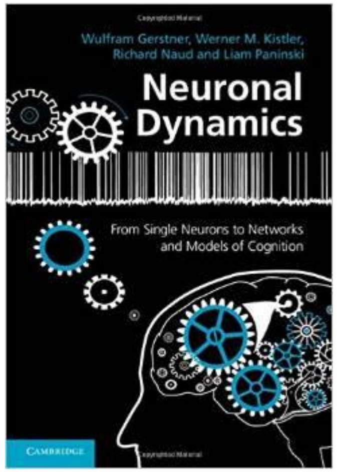


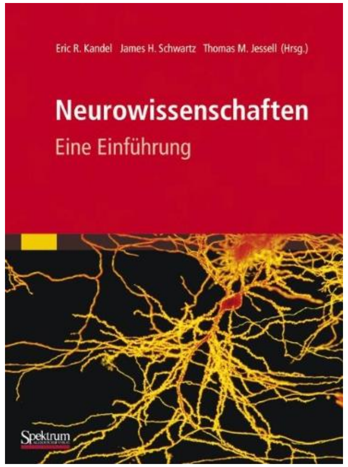
Selected literature

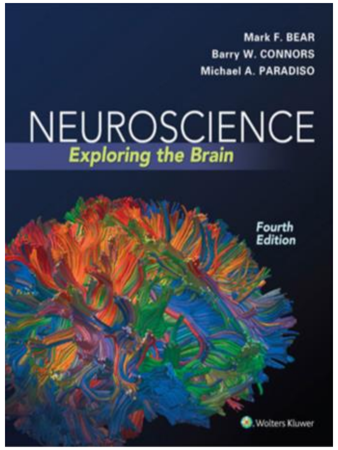
- W. Gerstner, W. M. Kistler, R. Naud and L. Paninski:
  “Neuronal Dynamics“,
  Cambridge University Press, 2014

- E. Kandel, J. Schwartz, T. Jessel:
  „Neurowissenschaften – Eine Einführung”,
  Spektrum Verlag, 1996 (2011)

- M. Bear, B. Connors, M. Paradiso: *„Neuroscience – Exploring the Brain“,*
  Wolters Kluwer, 4th Edition, 2015







Overview

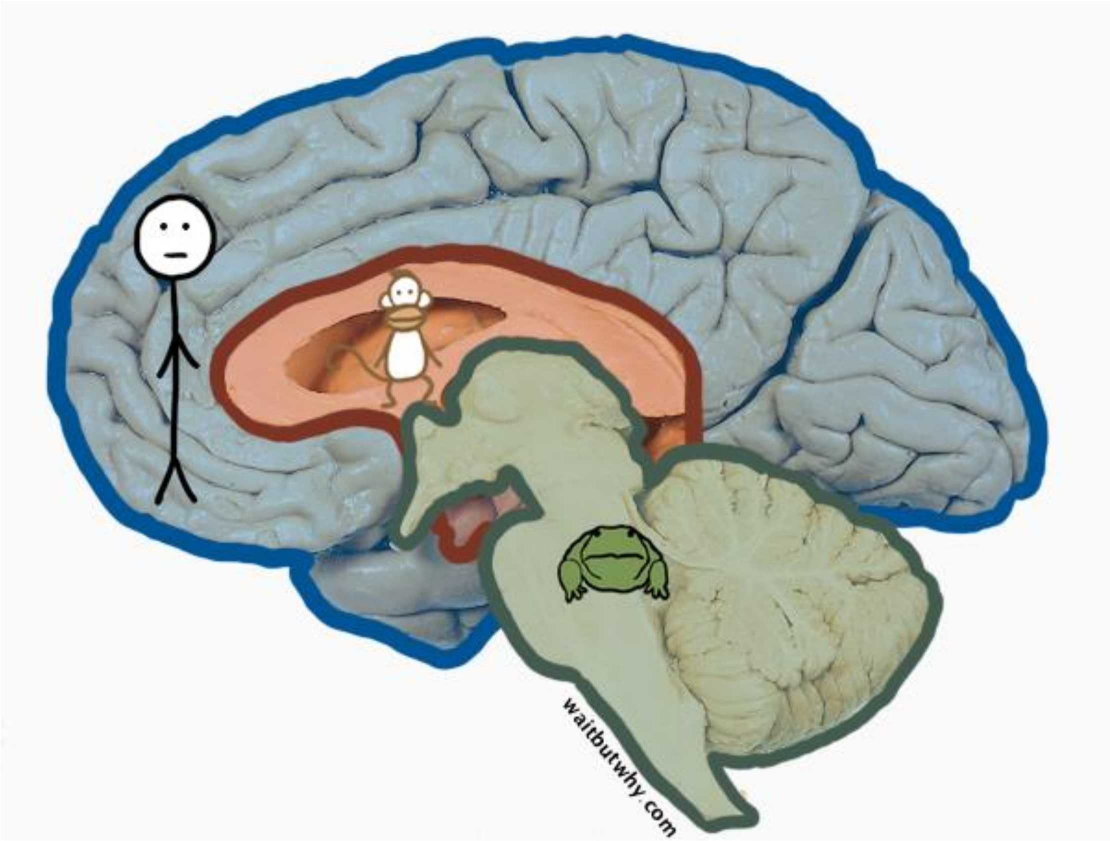
- Insights from biology
  - The Human Brain
  - The Cortex
  - Neurons and Synapses
- Modelling biology
  - Artificial Neural Networks
  - Spiking Neural Networks
  - Spiking Neurons
- Neural Coding
  - Common Coding Schemes
  - Input / Output Representation
- Learning with spiking networks - theory
  - Differences with analog networks
  - Formalization of plasticity
- Learning with spiking networks - practice
  - STDP in action
  - Spiking networks as kernel methods
- Materializing spiking networks
  - Neuromorphic hardware
  - Neurorobotics

Overview

- Insights from biology
  - The Human Brain
  - The Cortex
  - Neurons and Synapses
- Modelling biology
  - Artificial Neural Networks
  - Spiking Neural Networks
  - Spiking Neurons
- Neural Coding
  - Common Coding Schemes
  - Input / Output Representation
- Learning with spiking networks - theory
  - Differences with analog networks
  - Formalization of plasticity
- Learning with spiking networks - practice
  - STDP in action
  - Spiking networks as kernel methods
- Materializing spiking networks
  - Neuromorphic hardware
  - Neurorobotics

Back to biology – the brain

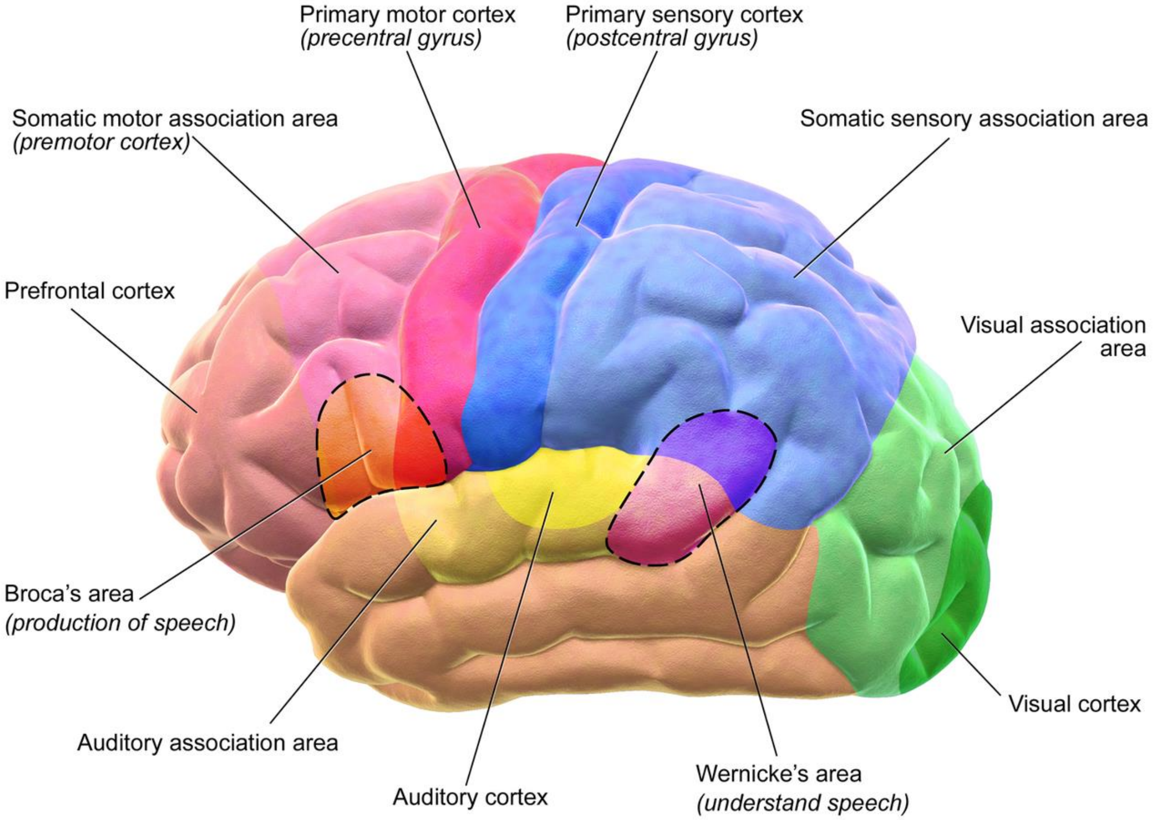
- The brain has 86 billion neurons
- Each neuron has between 5,000 and 10,000 synapses
- Knowledge is encoded in the synaptic weights
- Need for learning:
  - Synapses: 100 trillion
  - Genetic code: 100 billion bits
- 600 million years of evolution before the human brain

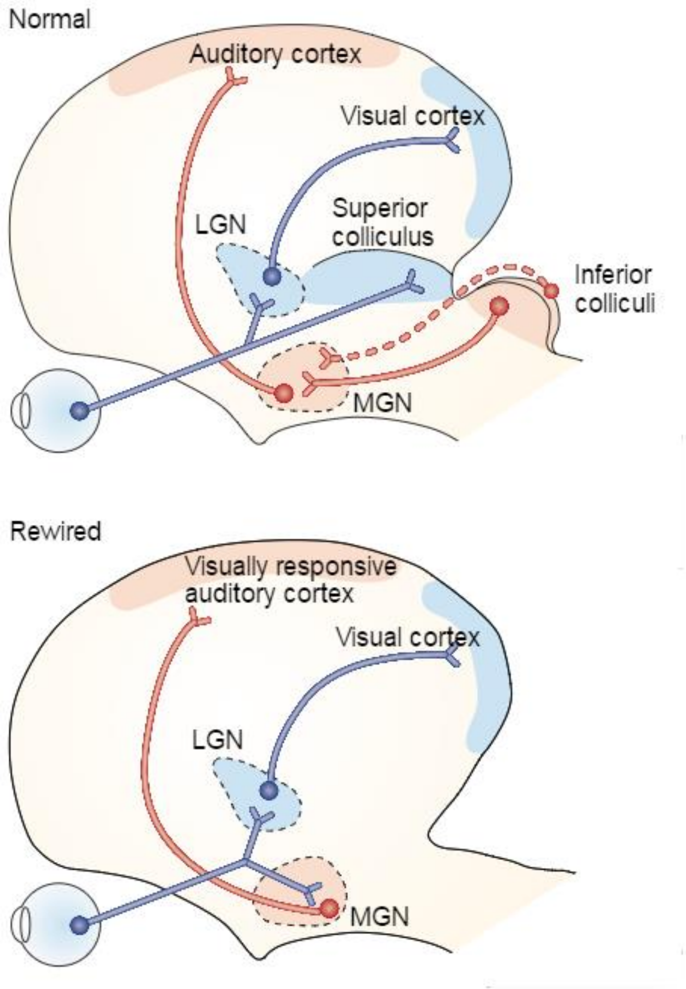


http://waitbutwhy.com/2017/04/neuralink.html

What makes us humans - the cortex

- Sub-regions of the cortex are specific to some functions
- Yet, any region can replace another one
- Support for „single learning algorithm“ theory

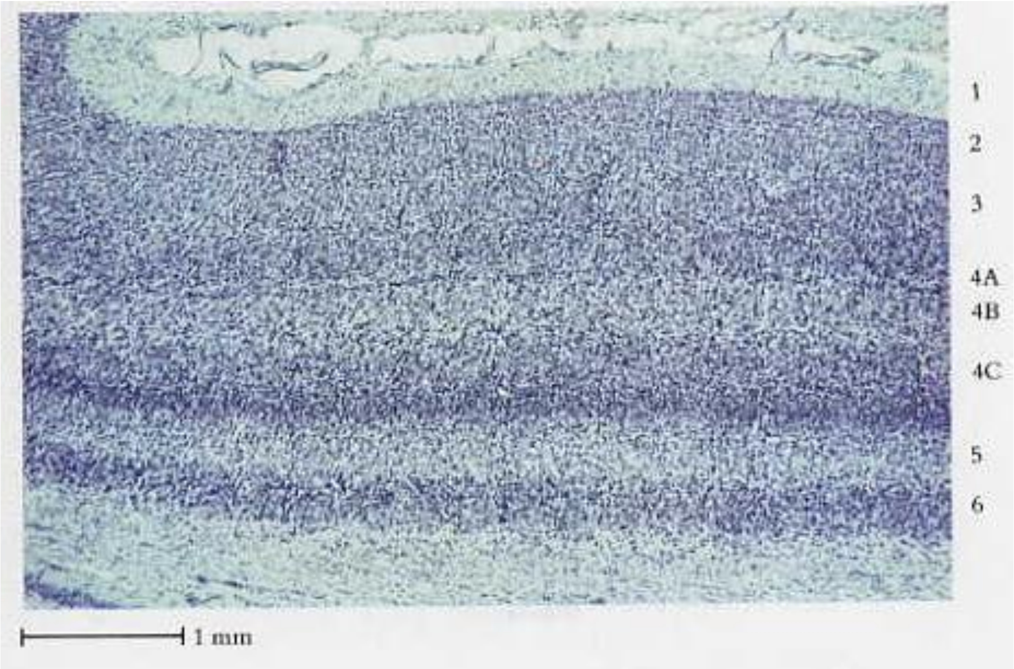


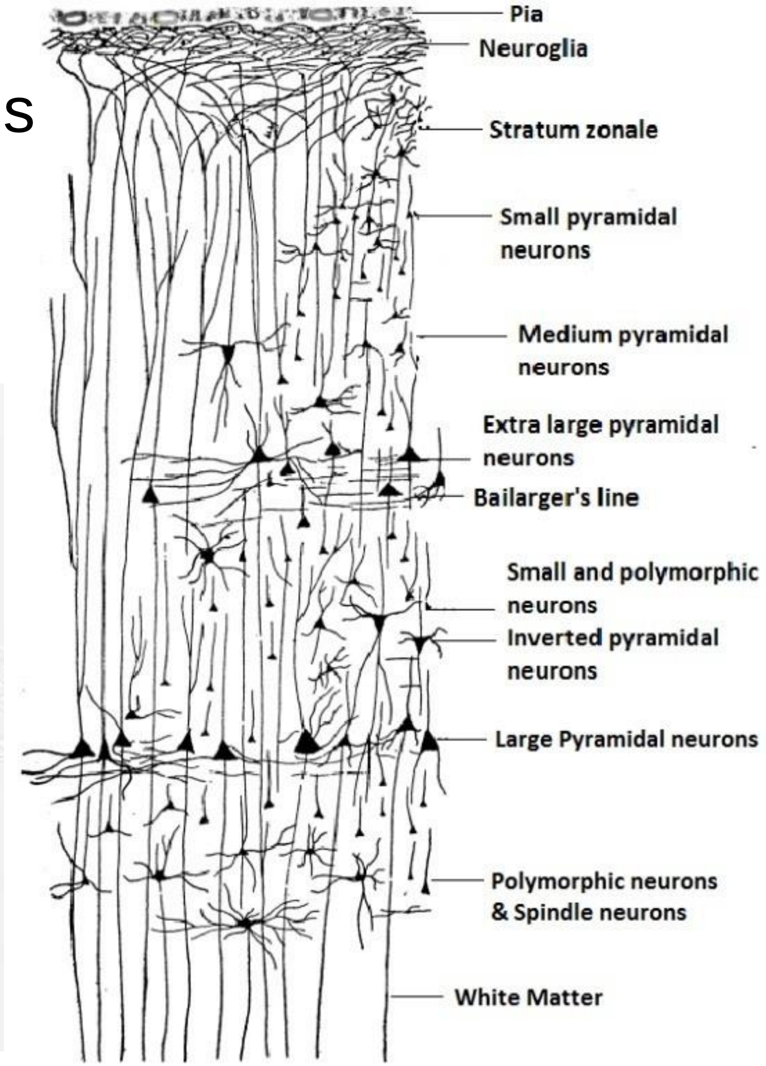


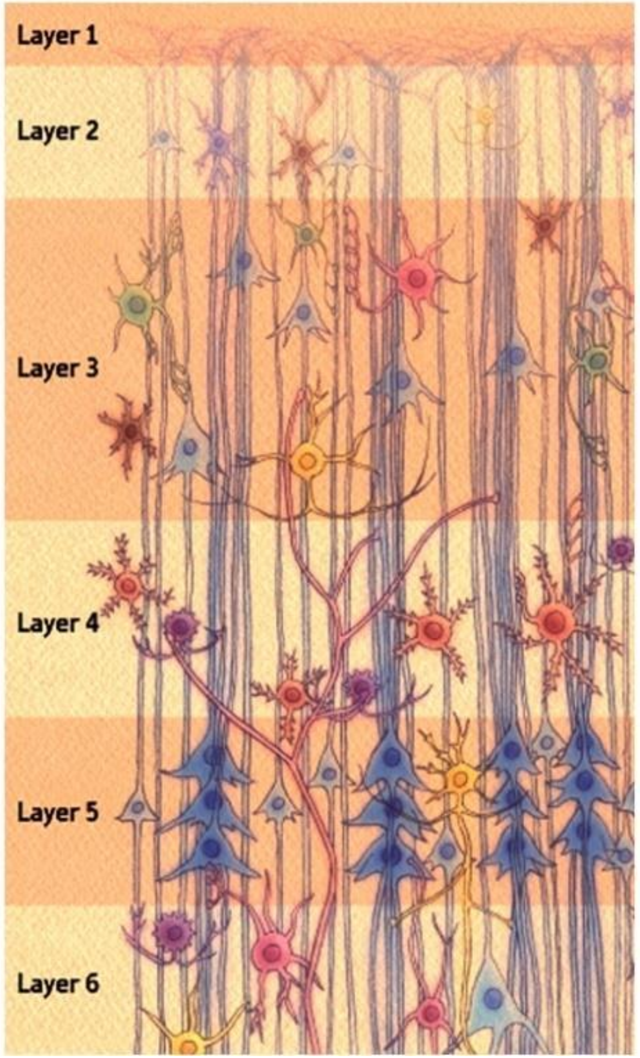
“Development and plasticity of cortical areas and networks” Sur et al., Nature 2001

Hierarchical organization of the cortex

- Composed of different neurons
- 6 layers
- Vertically - Cortical columns

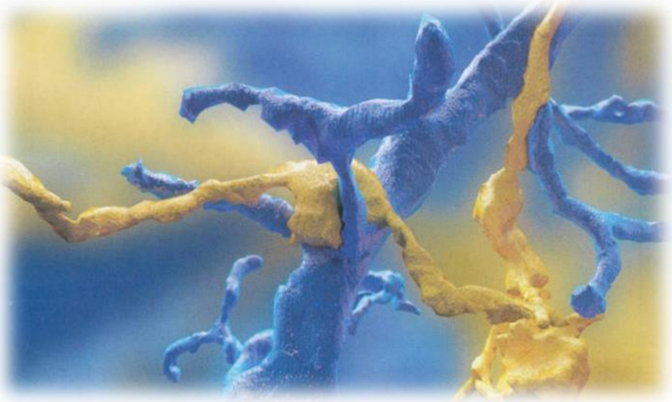


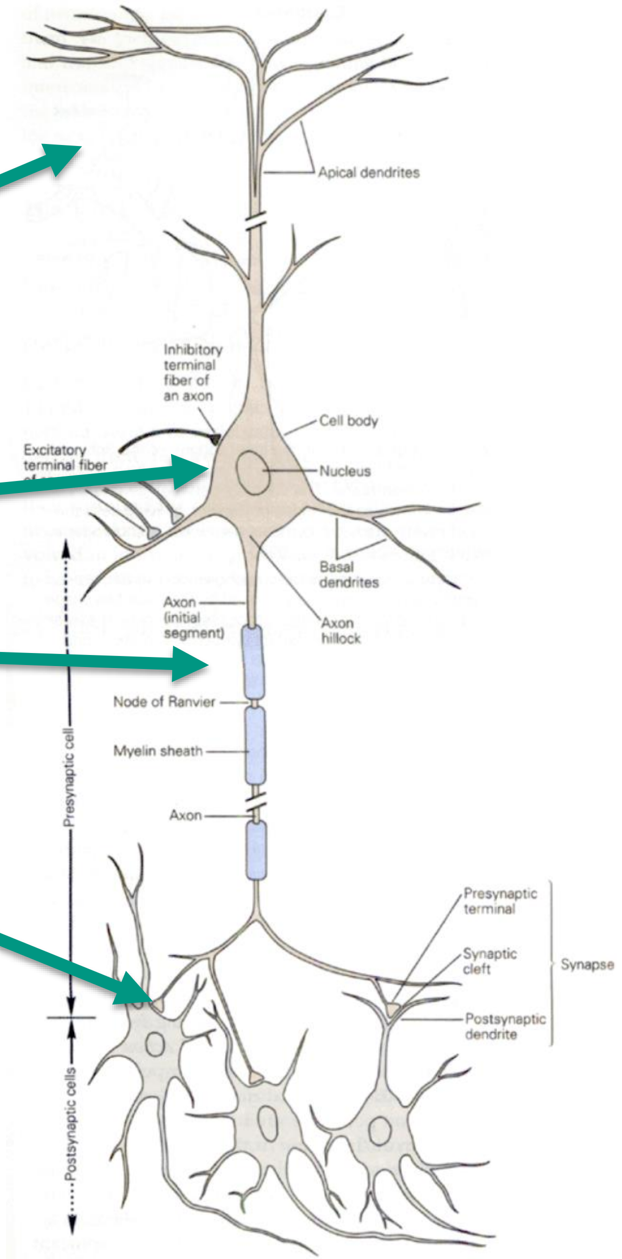




Cortical cells - neurons

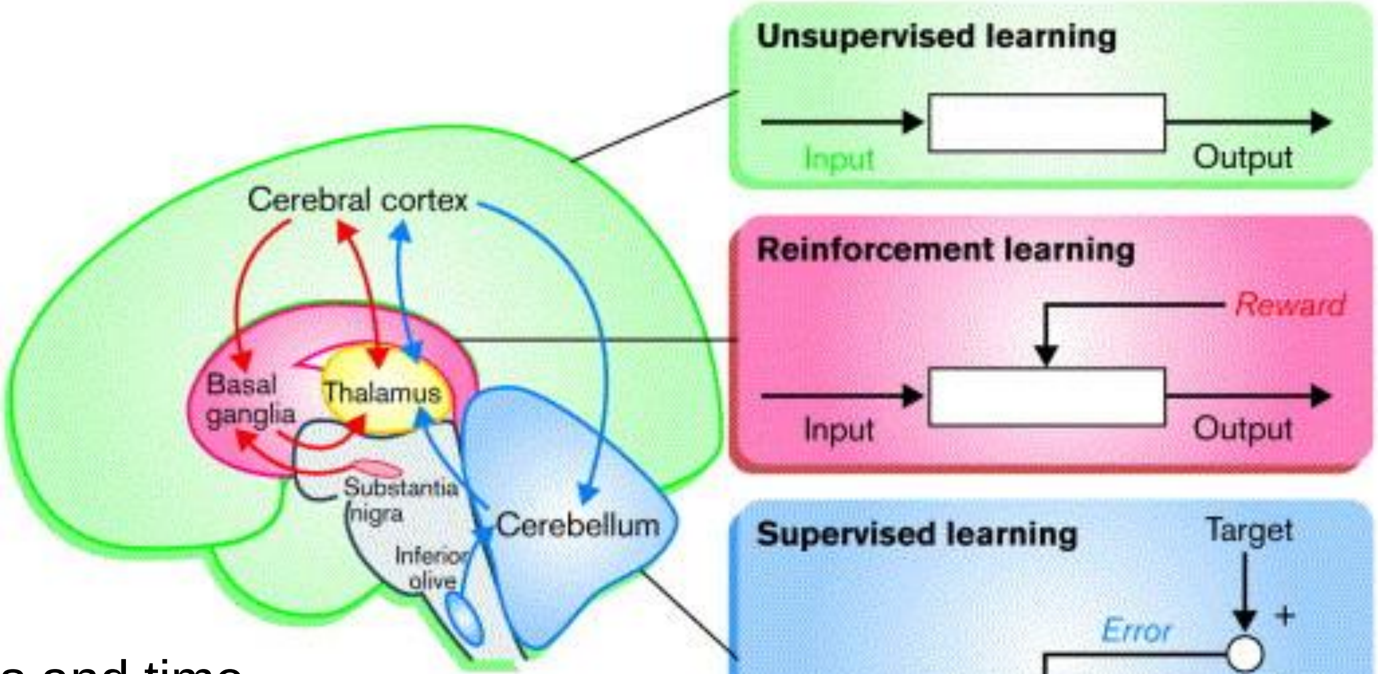
- Different types
- Structure of a neuron:
  - Dendrites – input
  - Soma - summation
  - Axon - output
  - Synapses - connection





How do we learn - a theory

- Cortex: unsupervised learning
- Basal ganglia: reinforcement learning
- Cerebellum: supervised learning



Hypotheses

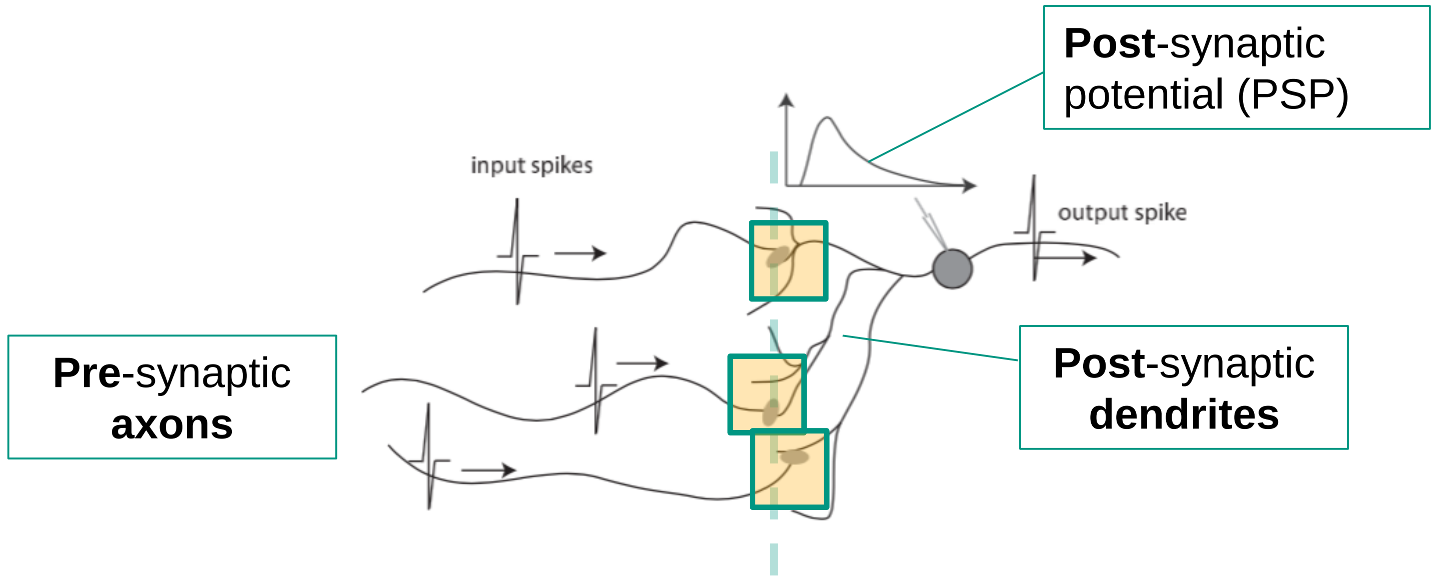
1. The brain optimizes cost functions
2. Cost functions are diverse across areas and time
3. Specialized circuits for key problems

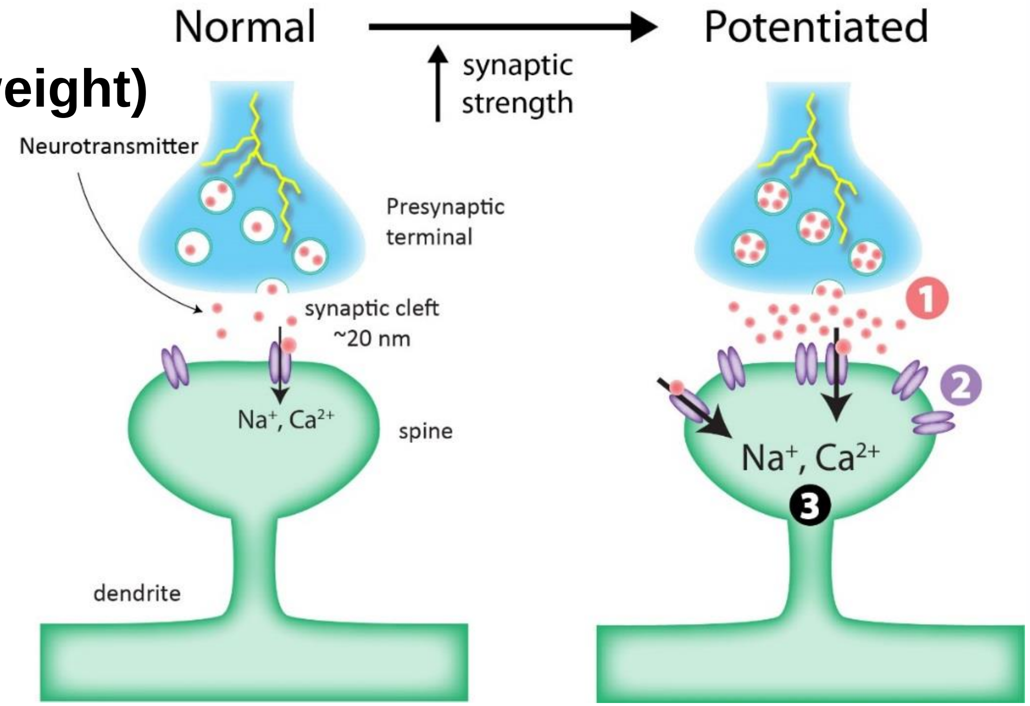
„Toward an Integration of Deep Learning and Neuroscience“,
Marblestone et al., Frontiers in computational neuroscience, 2016

„Complementary roles of basal ganglia and cerebellum in learning and motor control“,
Doya K, Current Opinion in Neurobiology, 2000

What happens in neurons and synapses?

- Strength of post-synaptic potential (PSP) depends on:
  - Amount of neurotransmitters in axon
  - Number of ion channels (receptors) in dendrites
  - In simulators, abstracted by **synaptic strength (weight)**
- Plasticity: change in one of these quantities



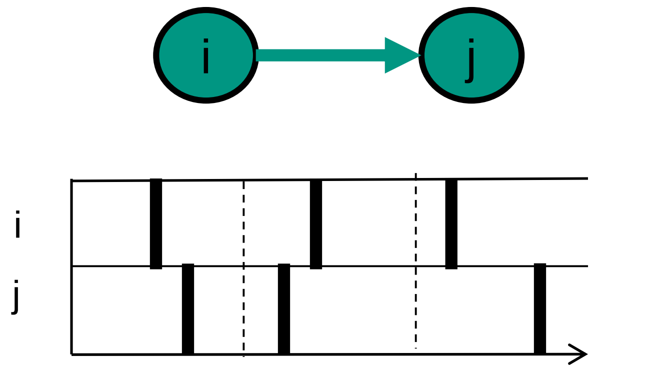


Institut für Anthropomatik und Robotik (IAR)
M. Sc. Jacques Kaiser
Maschinelles Lernen II

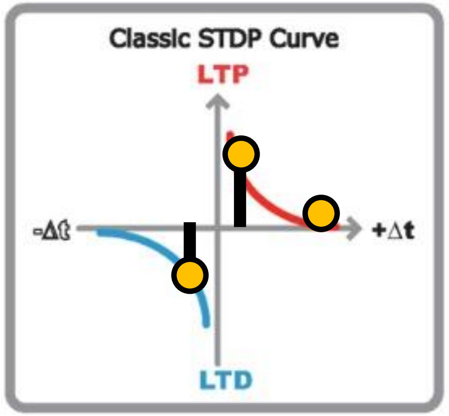
Fakultät für **Informatik**

Synaptic plasticity enables learning

- Plasiticy depends on precise timing of spikes
- LTP – Long Term Potentiation (+)
- LTD – Long Term Depression (-)
- Hebbian rule:
  - “Neurons who fire together wire together”
  - Learning is local and incremental



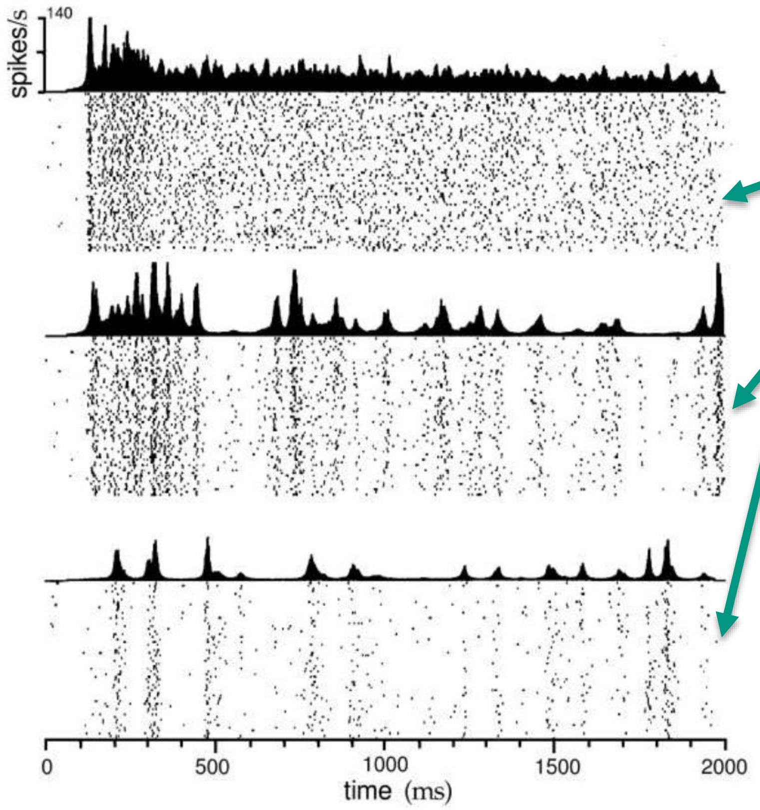
Spike train



“A history of spike-timing-dependent plasticity”, Henry Markram et. al

Reverse-engineering the brain – spike trains

Motion selective neurons



Constant velocity stimulus

Varying velocity stimulus

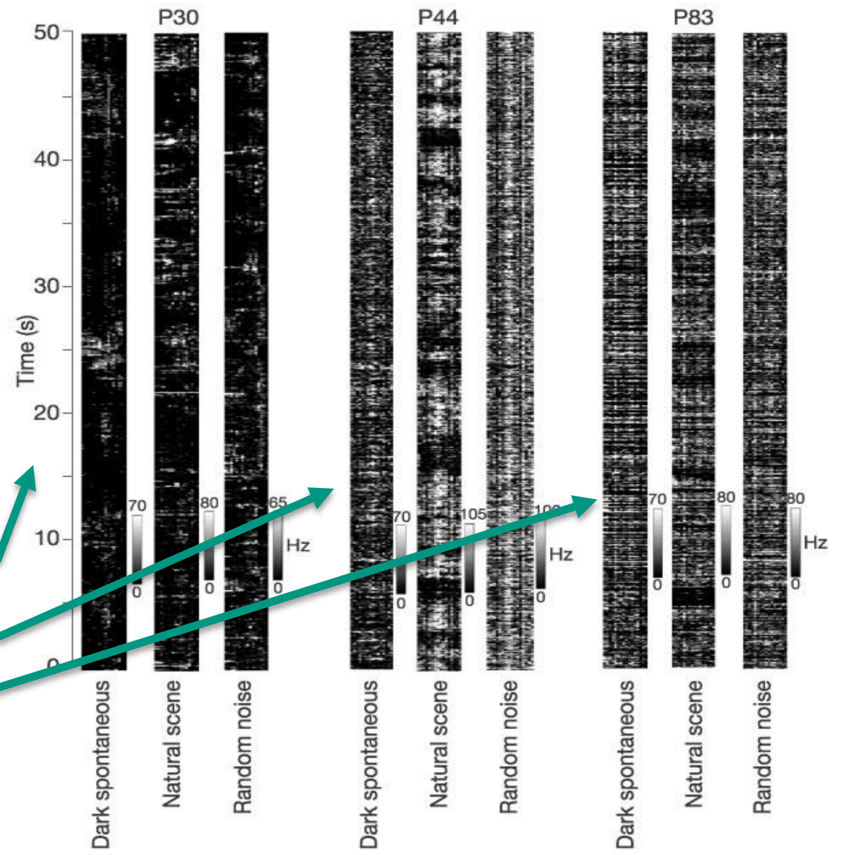
„Neural Computation.“, Mark van Rossum

Modulation of spontaneous activity

Post-natal

Mature

Old



„Small modulation of ongoing cortical dynamics by sensory input during natural vision.“, Nature 2004

Overview

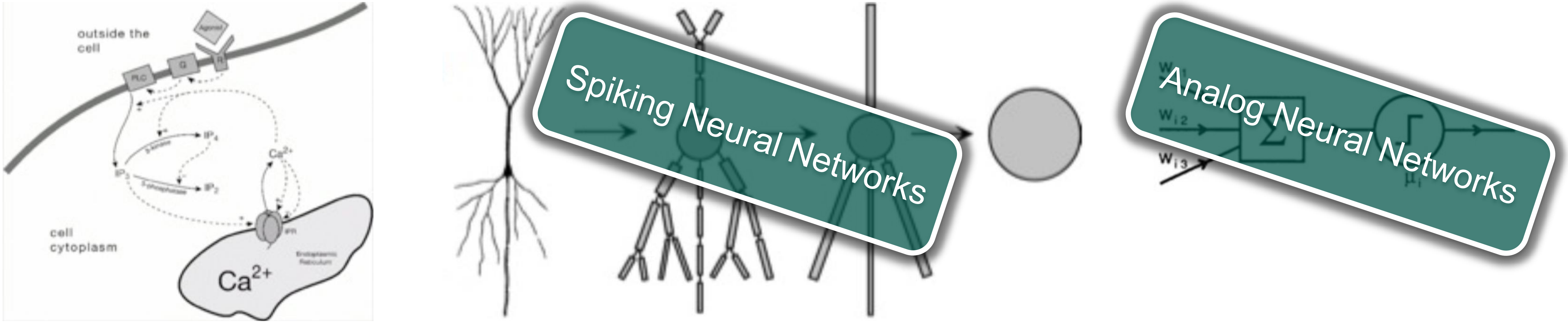
- Insights from biology
  - The Human Brain
  - The Cortex
  - Neurons and Synapses
- Modelling biology
  - Artificial Neural Networks
  - Spiking Neural Networks
  - Spiking Neurons
- Neural Coding
  - Common Coding Schemes
  - Input / Output Representation
- Learning with spiking networks - theory
  - Differences with analog networks
  - Formalization of plasticity
- Learning with spiking networks - practice
  - STDP in action
  - Spiking networks as kernel methods
- Materializing spiking networks
  - Neuromorphic hardware
  - Neurorobotics

History of artificial neural network models

- **-400: Aristotle** thinks the brain is a cooling system for the blood, while the heart is responsible for intelligence
- **1839**: **Theodor Schwann** proposes the **cell theory** - a basic functional unit of all living things
- **1888: Santiago Ramón y Cajal** discovers the neurons – the cells in the nervous system
- **1943**: **McCulloch und Pitts** describes the first neural network model
- **1949**: **Hebb** postulates that synaptic weight can be calculated as a product of the pre- and post-synaptic activity
- **1958**: **Frank Rosenblatt** presents the **Perceptron**
- **1997**: **Wolfgang Maass** describes **Spiking Neural Networks**
- **Today**:

  Active field of research in neuroscience, machine learning and robotics

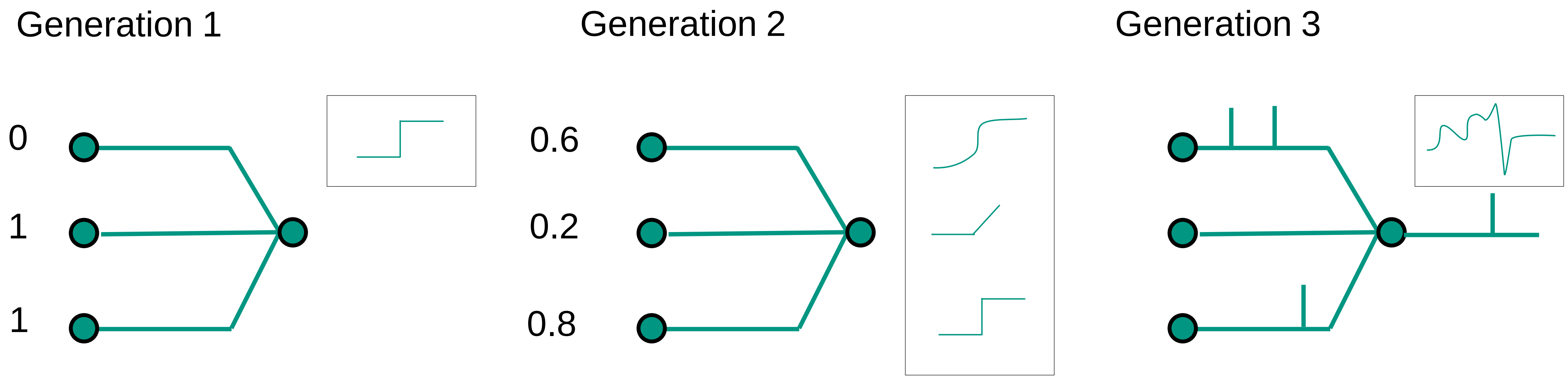
Abstraction level – what gives rise to the function?



- Biological processes (molecular)
- Electrical properties (morphology)
- Electrical properties (average)
- Computational properties (processing)

Computational neuroscience. Research group at the University of Tartu.

Different abstractions yield different models



- Neurons & synapses are modeled as dynamical systems

„Network of spiking neurons: the third generation of neural network models“, Maass W, Neural Networks, 1997

Why modelling at the spike level?

- Many functions are based on precise spike-time in humans, e.g.:
  - Locating a sound in 3D
  - Vision – your retina does not encode static images, see [1]
  - Computationally efficient, see [2]
- Many functions are also based on rate-coding, e.g.:
  - Muscle activation

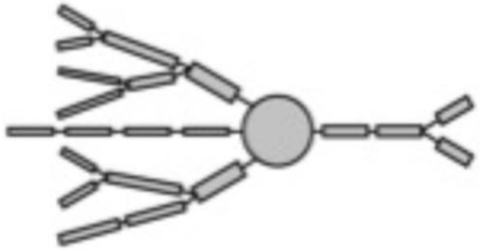
By modelling down to the spike level, we can do both… and more

- Computational benefits of distributed computing
- Neuromorphic hardware

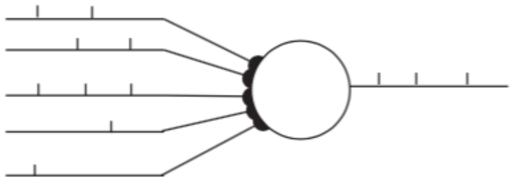
[1] „Rapid Neural Coding in the Retina with Relative Spike Latencies ", Gollisch et al., Science 2008
[2] „Computing with Spiking Neurons", Maass 1999

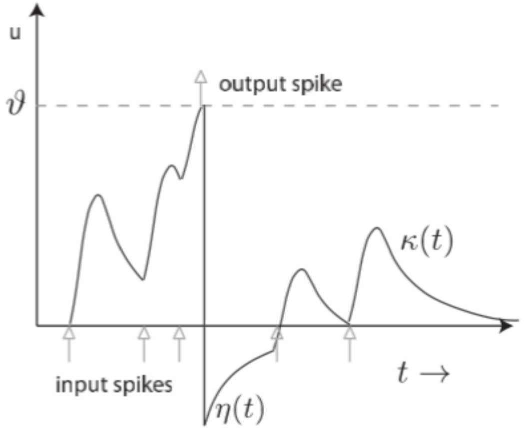
Modelling at the spike level

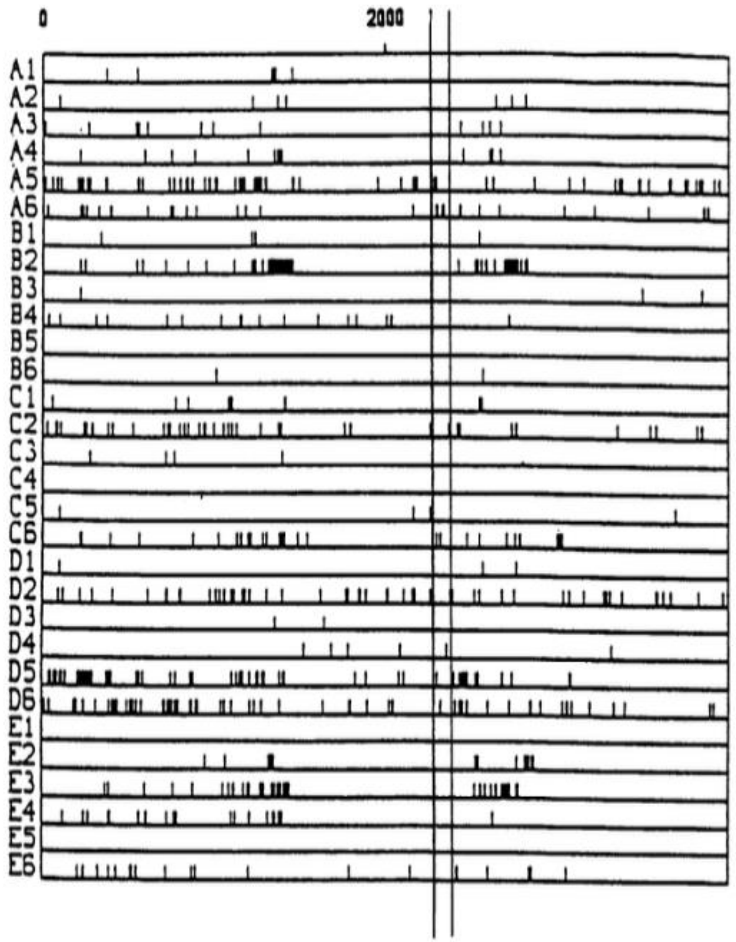
- Multi-compartment



- Point-neuron



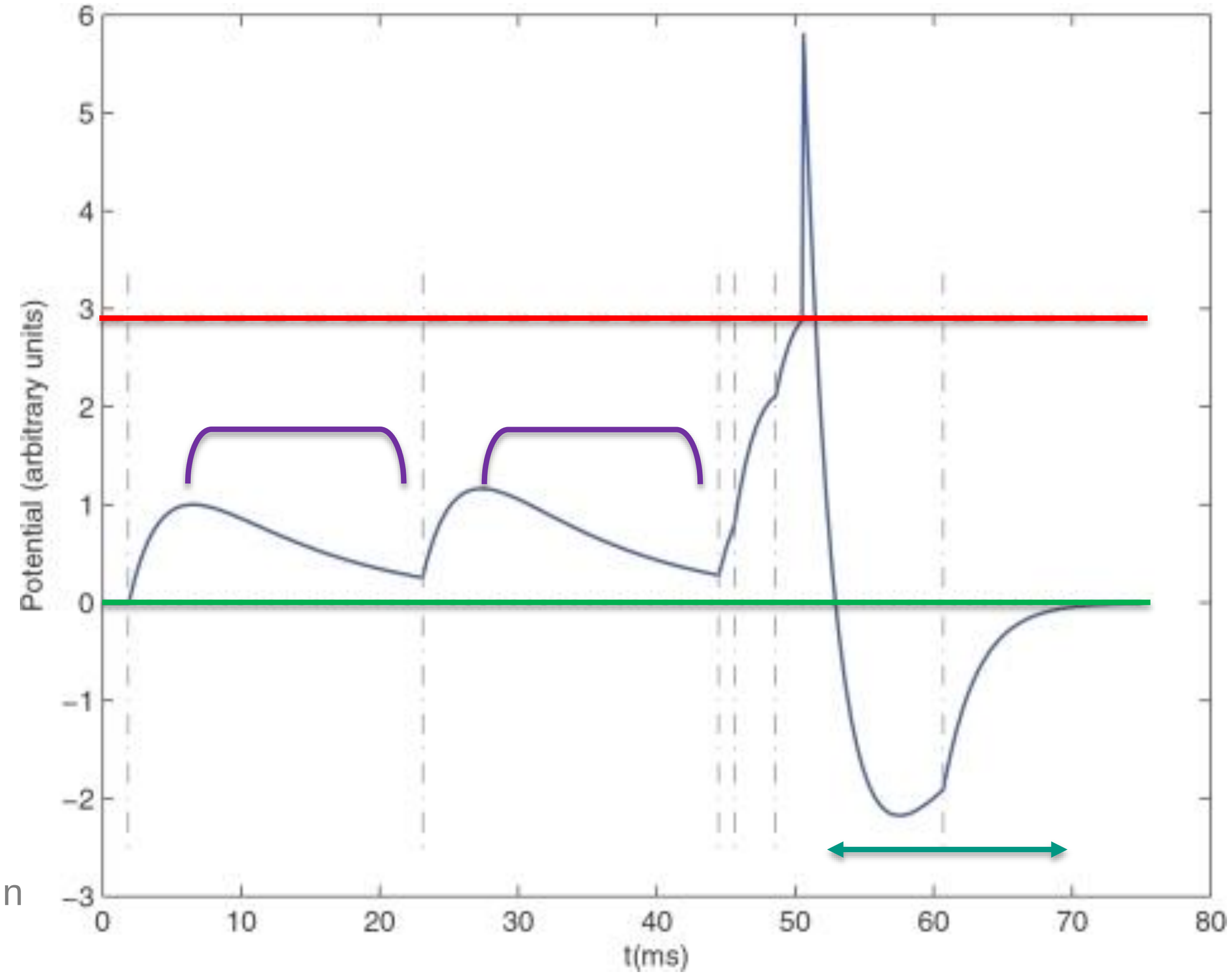




Grüning, Bohte , 2014, Spiking Neural Networks Principles and Challenges
Wolfgang Mass, 1997, Networks of spiking Neurons: The Third Generation of Neural Network Models

Classic spiking neuron model

- Differential equations with respect to time
- PSP shape (kernel)
- Input:
  - Current
- Output:
  - Spikes
- Variables:
  - Membrane potential: $V(t)$
- Parameters:
  - Threshold: V_{th} or ϑ
  - Resting potential: V_{rest}
  - Leak (membrane time constant): τ_m
  - Refractory period: τ_{ref}



"Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains.", Masquelier et al., PLoS ONE 2008

Popular spiking neuron models

- Integrate-and-fire model
  - Simple electrical circuit, focus on precise spike-time

  „Lapique‘s introduction of the integrate-and-fire model neuron“ by Abbott

- Hodgkin-Huxley model
  - Realistic model, implying extreme computing times

  „A quantitative description of membrane current and ist application to conduction and excitation in nerve“ by Hodgkin & Huxley

- Izhikevich’s neuron model
  - Compromise between biological plausibility and computing time

  „Simple model for Spiking Neurons“ by Izhikevich

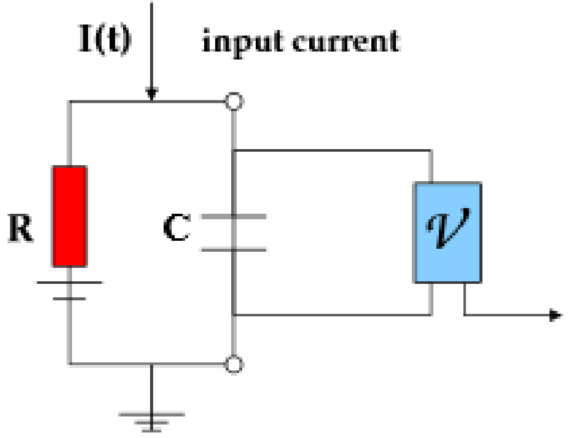
- Spike Response Model
  - A powerful, yet simple and general model, including refractoriness

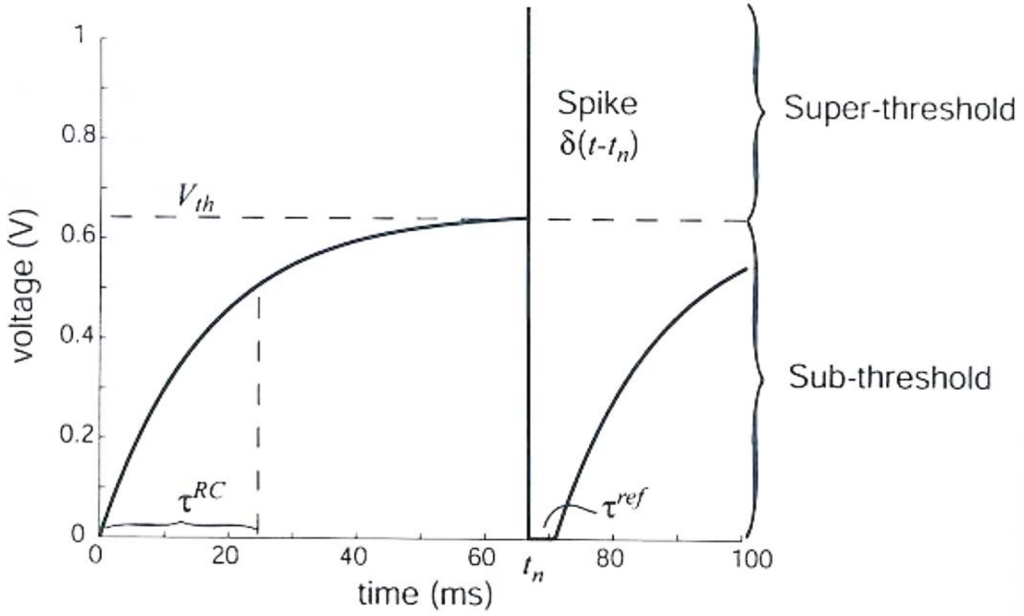
  „Time structure of the activity in neural network models“ by Gerstner

Focus on simplicity - Leaky integrate and fire (LIF)

- Most popular model for neuro-engineers
- Based on an electrical circuit
- Simple and fast
- Does not model the shape of action potentials
- **Membrane potential**: capacitor
- **Threshold**: gate
- **Resting potential**: battery
- **Leak**: resistance

$$\tau_m \cdot \frac{dV(t)}{dt} = V_{rest} - V(t) + R \cdot I(t)$$





http://neuronaldynamics.epfl.ch/online/Ch1.S3.html
„Lapique‘s introduction of the integrate-and-fire model neuron“ by Abbott

Leaky integrate and fire in action

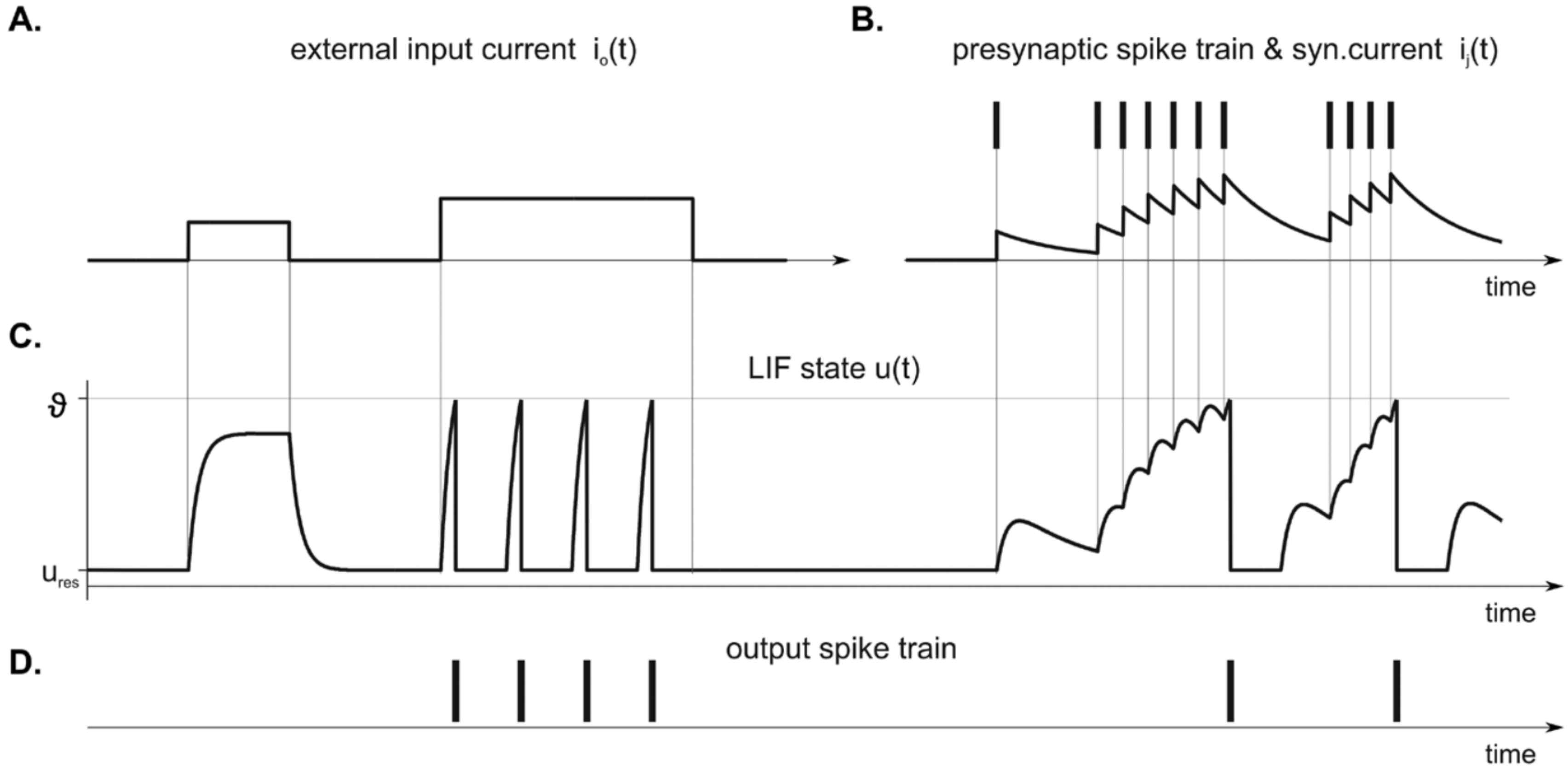
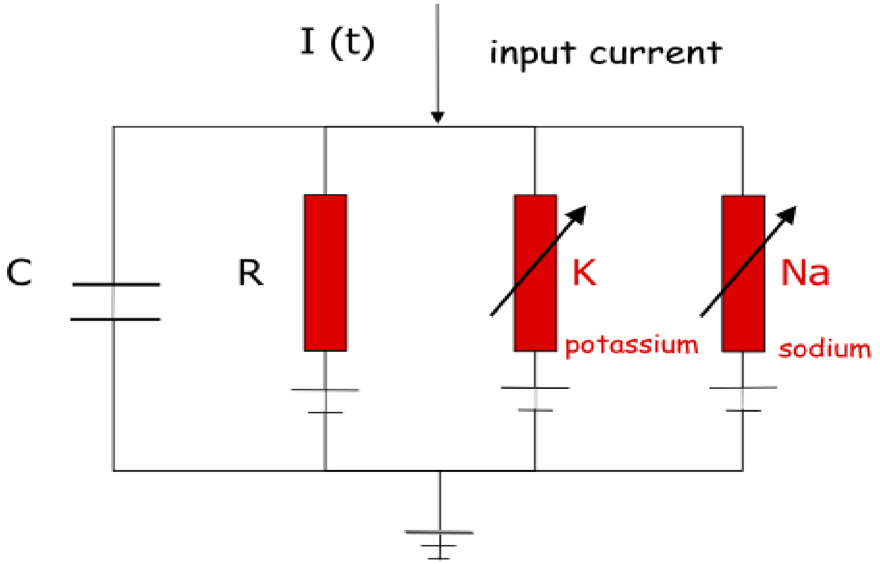


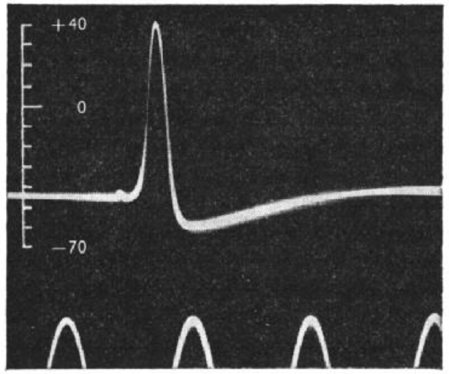
Fig. 1. Time course of the membrane potential u(t) of a leaky-integrate-and-fire neuron LIF (panel C) driven by the external input current $i_o(t)$ (shown in panel A) or by the synaptic current i_j (t) evoked by the sample presynaptic spike train (panel B). Initially, the state u(t) of the LIF neuron is at the resting value u_{res}. The currents $i_o(t)$ and $i_j(t)$ increase the membrane potential towards the firing threshold θ. Whenever the threshold is crossed the neuron emits a spike and the membrane voltage u(t) is reset to a new value - here assumed u_{res}. The firing times of the LIF neuron are shown as vertical bars in panel D.

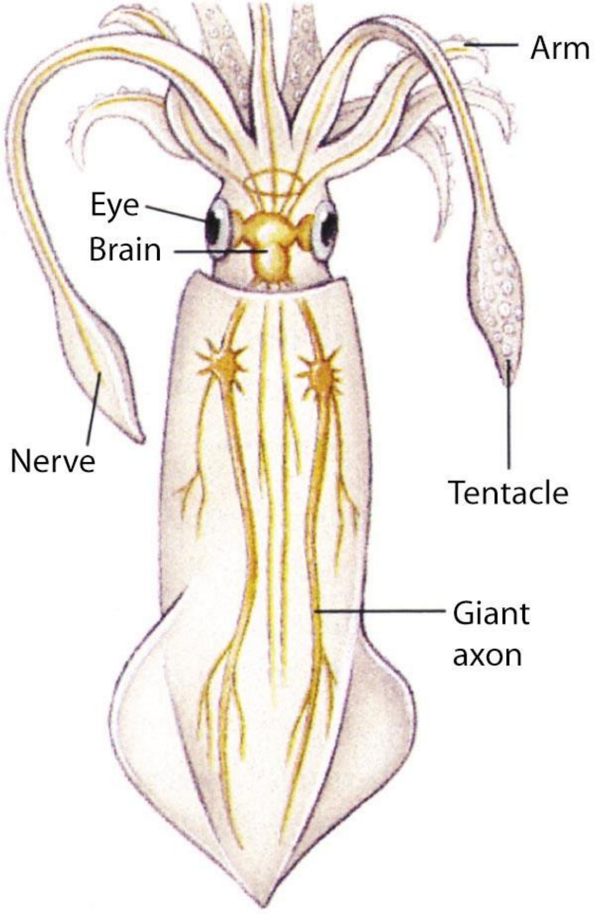
F. Ponulak, A. Kasinski. 2011. Introduction to spiking neural networks: information processing, learning and applications.

Focus on realism - Hodgkin-Huxley

- Formalized in 1952
- Models the membrane potential of an octopus neuron
- Also based on an electrical circuit





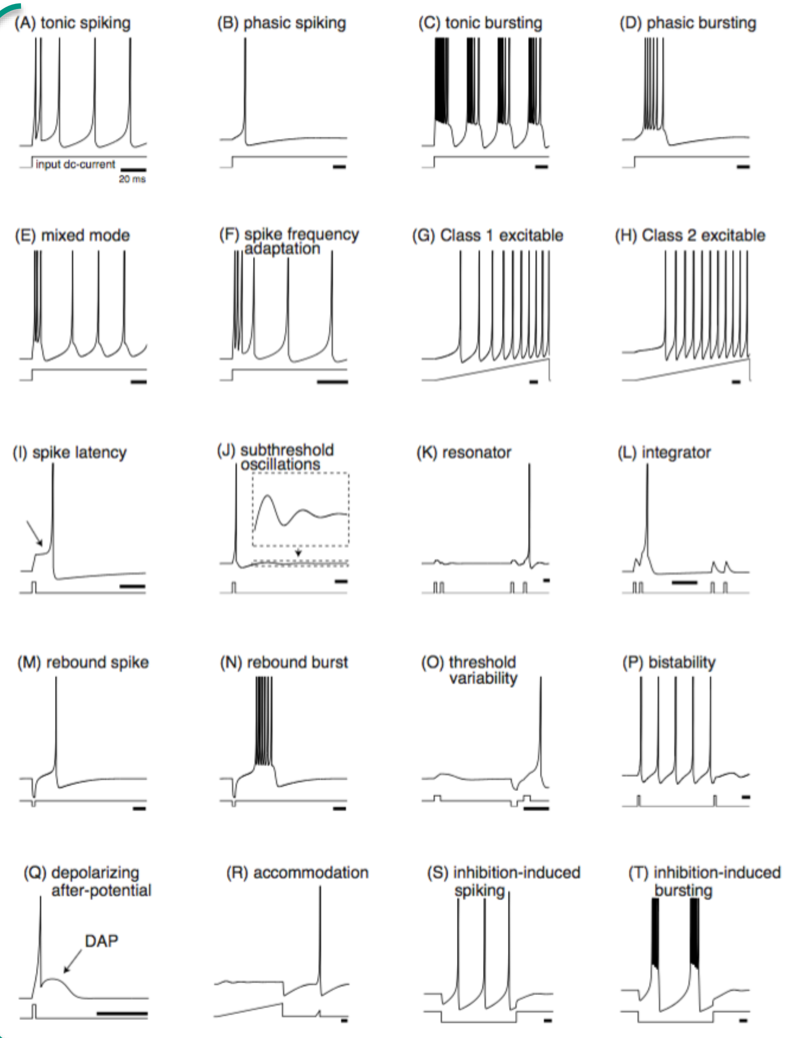


Focus on matching data - Izhikevich Model

- Differential equation of 2-Dimensions

$$\begin{cases} \dfrac{dV(t)}{dt} = 0.04 \cdot V(t)^2 + 5 \cdot V(t) + 140 \; - w(t) + I(t) \\ \dfrac{dw(t)}{dt} = a \cdot (b \cdot V(t) - w(t)) \end{cases}$$

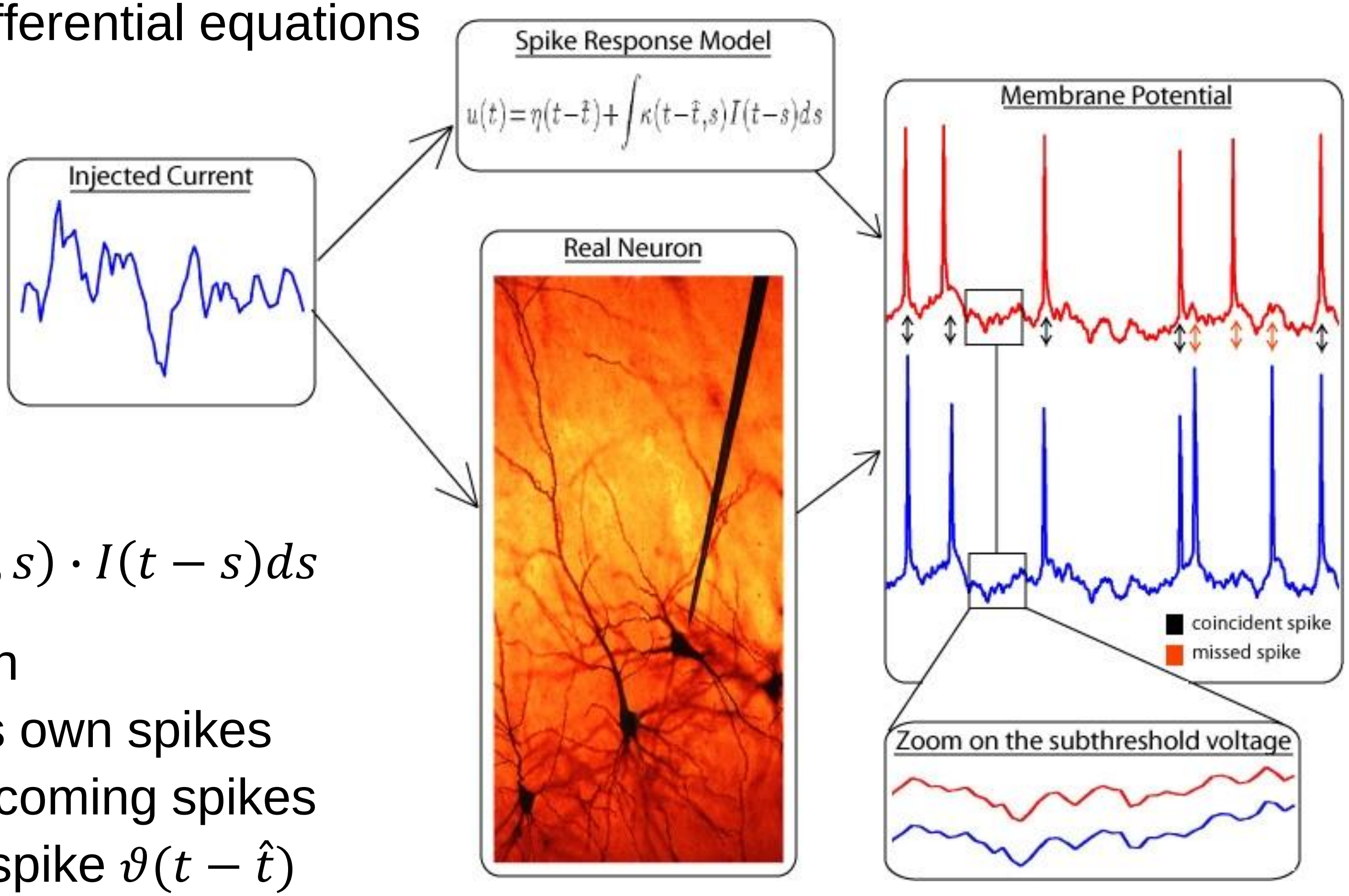
- Constants were fitted against biological data
- a and b can be tweaked for different dynamics
- Fast to simulate



Institut für Anthropomatik und Robotik (IAR)
M. Sc. Jacques Kaiser
Maschinelles Lernen II

A different formulation – The Spike Response Model

- Generalization of leaky integrate-and-fire model
- Formulated with filters instead of differential equations



$$V(t) = \eta(t-\hat{t}) + \int_0^\infty \kappa(t-\hat{t},s)\cdot I(t-s)ds$$

- $\hat{t}$ is the last spike time of the neuron
- η is the response of the model to its own spikes
- κ is the response of the model to incoming spikes
- The threshold depends on the last spike $\vartheta(t-\hat{t})$

Comparaison between models

Speed

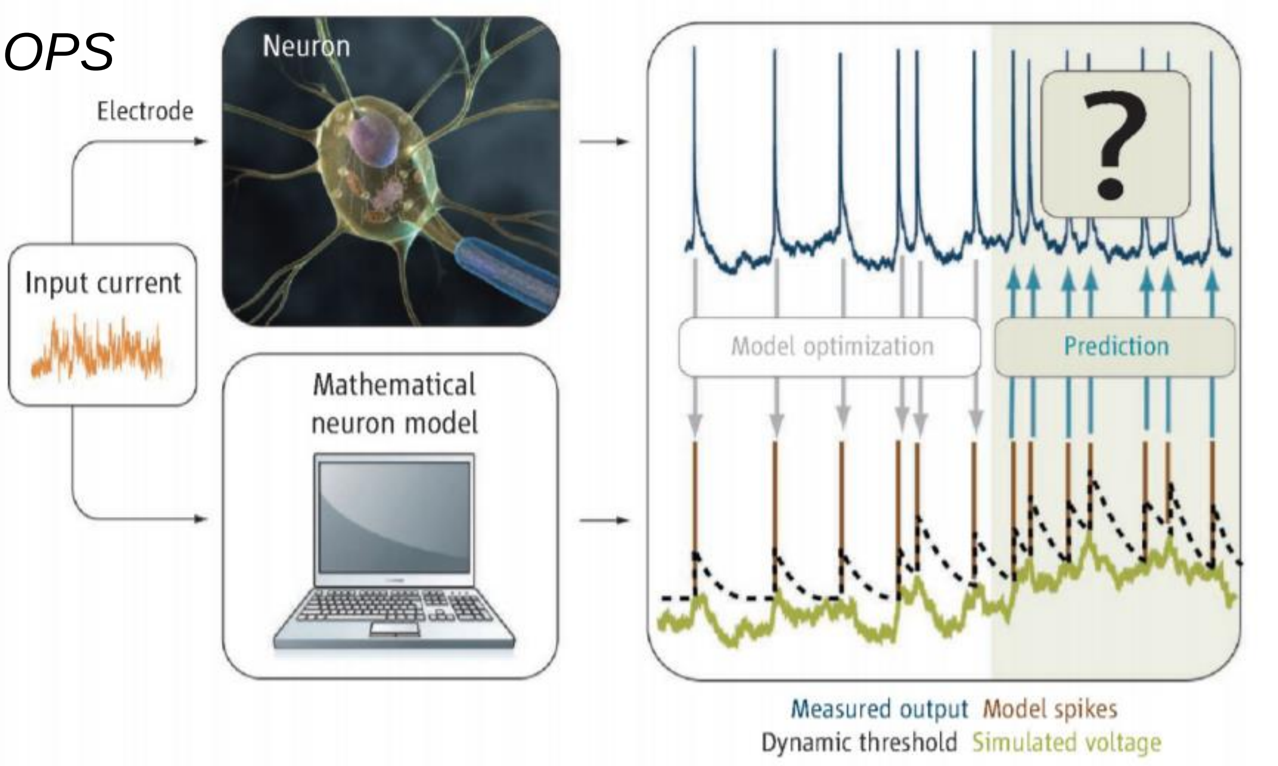
Hodgkin-Huxley	*1200 FLOPS*
Leaky Integrate and Fire	*5 FLOPS*
Izhikevich Model	*13 FLOPS*
Adaptive Leaky Integrate and Fire	*10 FLOPS*

Accuracy

Measured with spike prediction competitions

Spike Response Model	*0.69*
Adaptive Leaky Integrate and Fire	*0.82*

Problem: by injecting current in soma, dendritic non-linearities are not taken into account.



„A benchmark test for a quantitative assessment of simple neuron models “, Jolivet et al., Journal of Neuroscience, 2008

Overview

- Insights from biology
  - The Human Brain
  - The Cortex
  - Neurons and Synapses
- Modelling biology
  - Artificial Neural Networks
  - Spiking Neural Networks
  - Spiking Neurons
- Neural Coding
  - Common Coding Schemes
  - Input / Output Representation
- Learning with spiking networks - theory
  - Differences with analog networks
  - Formalization of plasticity
- Learning with spiking networks - practice
  - STDP in action
  - Spiking networks as kernel methods
- Materializing spiking networks
  - Neuromorphic hardware
  - Neurorobotics

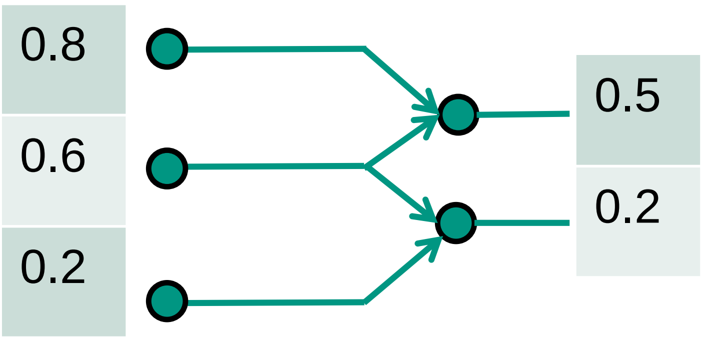
Information coding – a spiking network problem

Analog neural network

- Static
- Spatial
- No concept of time

Trivial definition:

$$h: input\ vector \rightarrow output\ vector$$



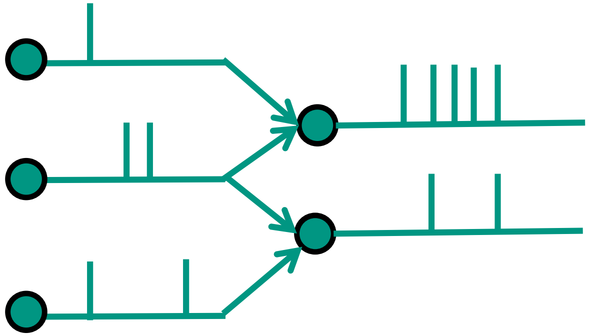
Spiking neural network

- Dynamic
- Spatio-temporal
- Defined with respect to time

How to define the processing?

How to **encode** the input?

How to **decode** the output?



Rate coding – from spiking to analog network

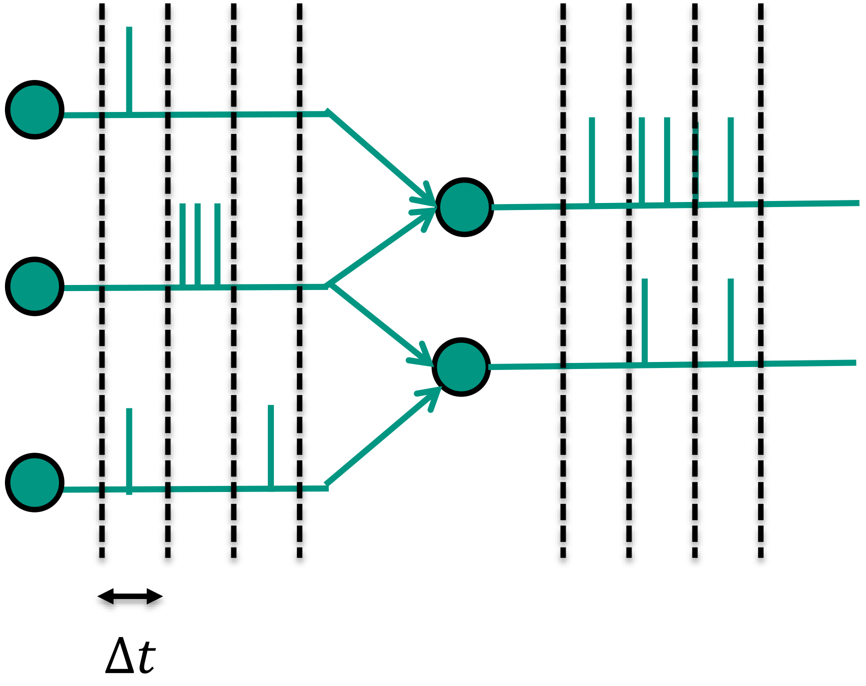
- Spiking rate is computed over discrete time intervals
- Input vectors map to output vectors
- Rate-based networks = Analog networks

Inconvenients:

- Computing spike rates is slow
- Inefficient

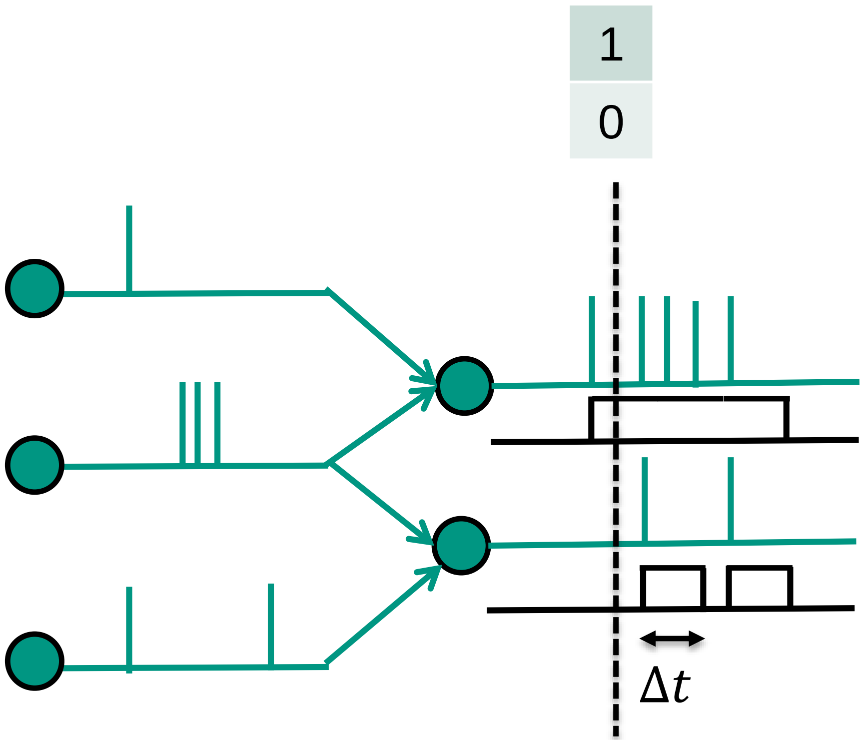
- **Common use**: cognition and images

$$\frac{1}{\Delta t} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad \frac{1}{\Delta t} \times \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$



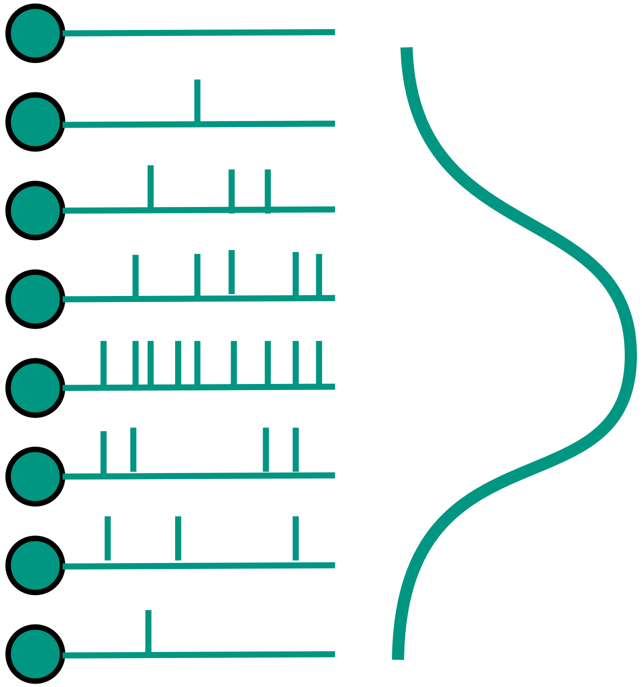
Binary coding – spike train sampling

- When a neuron fire, it is said „active“ for a given amount of time Δt
- We can sample the spike train at any time
- Same principles for values $\in \mathbb{R}$: exponential filter instead of binary (simulates PSPs)
- **Common use**: stochastic inference



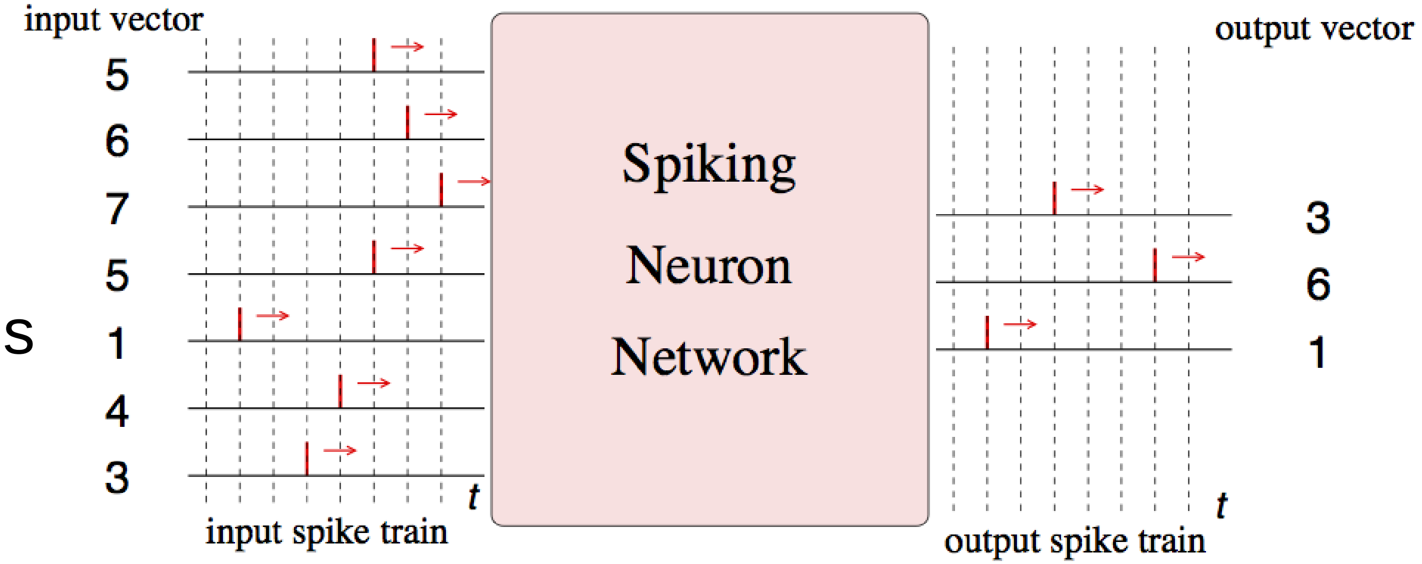
Gaussian coding – dealing with spatial stimuli

- Neurons have spatial positions
- We fit a Gaussian on the spiking rates
- **Common use**: proprioception in muscles



Synchronous coding schemes

- Need to define a reference time, e.g., a spike
- Information is encoded with respect to the reference
- Different schemes:
  - Time-to-first spike
  - Temporal coding
  - Rank order coding
  - Correlation coding
- Support complex computations with few neurons



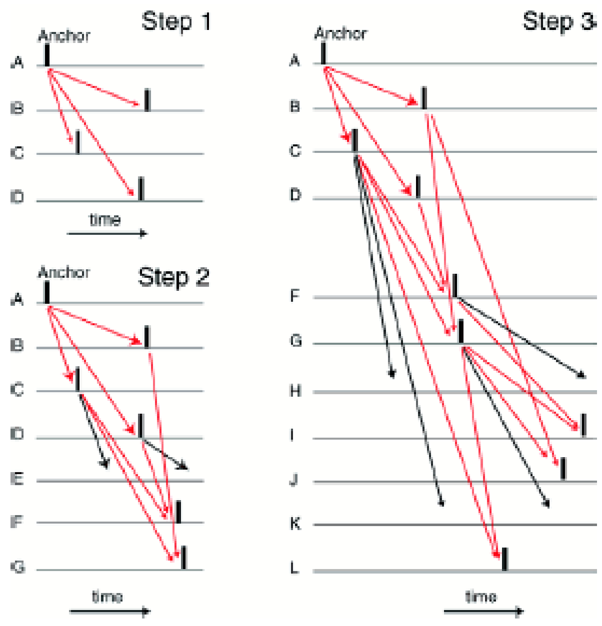
Paugam-Moisy, Bohte. 2012. Computing with Spiking Neuron Networks.

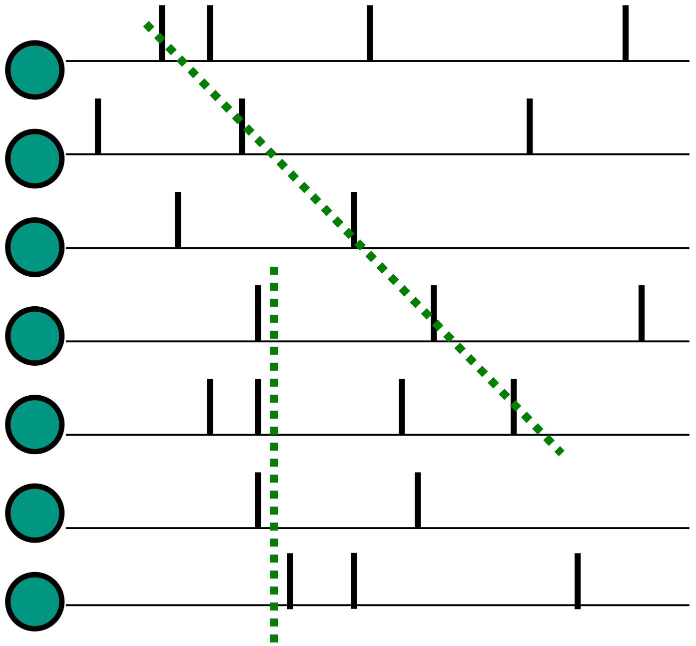
Correlations – dealing with spatio-temporal stimuli

- Repeating spatio-temporal spiking patterns
- Requires spike train analysis tools
- **Common use**: decoding stimuli in spike trains

elephant
ELECTROPHYSIOLOGY ANALYSIS TOOLKIT

http://neuralensemble.org/elephant/

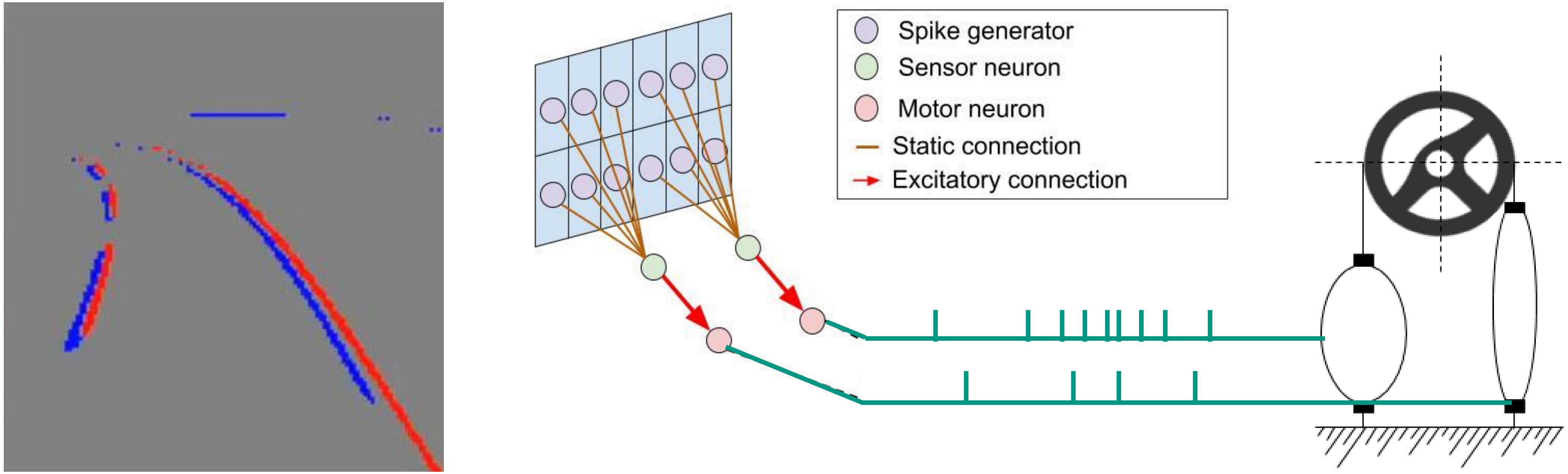




correlations

I/O - encoding and decoding

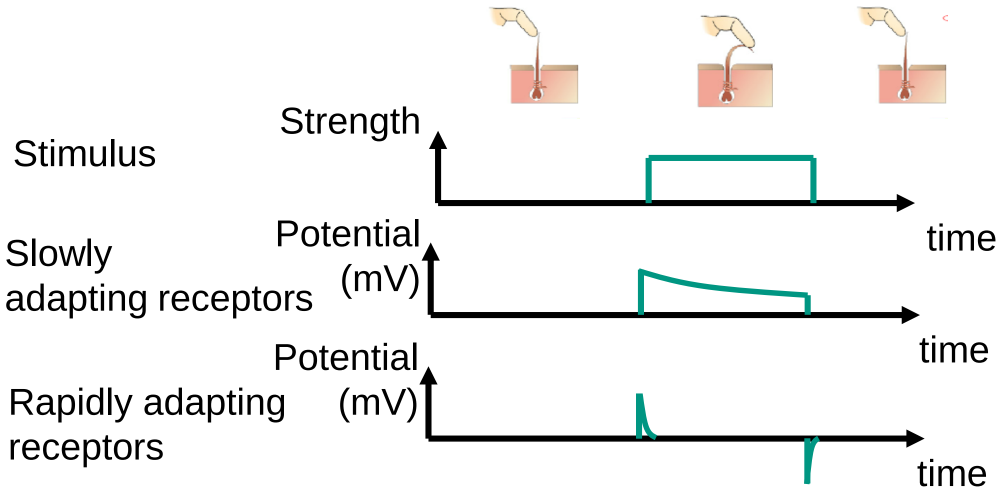
- It is up to you how do you encode inputs and decode outputs
- Different coding schemes can be used within the same network

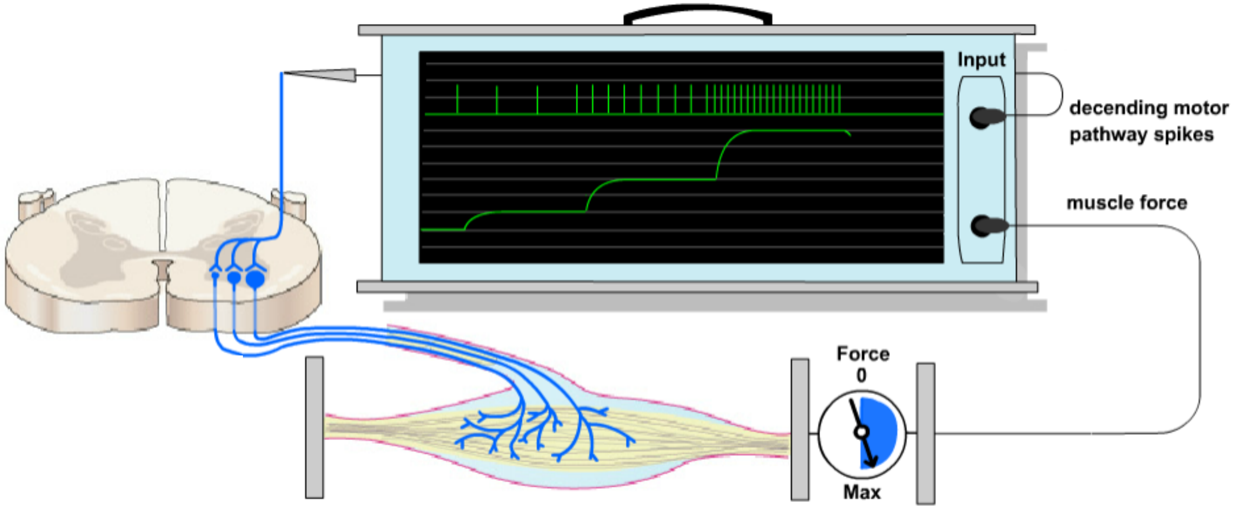


„Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks“, Kaiser, Tieck, et al., SIMPAR 2016

What coding does the brain use?

- We have many models for encoding (senses) and decoding (muscles)





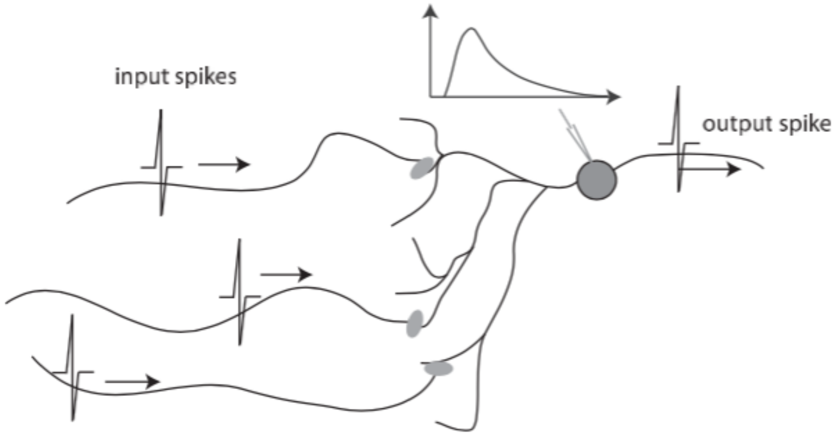
- For cognition, it is mostly unknown
- For vision, controversy between rate coding and temporal coding
- It is believed that different codings are used in different areas

„Philosophy of the Spike: Rate-Based vs. Spike-Based Theories of the Brain ", Brette et al., Frontiers in Systems Neuroscience, 2015

http://neuroscience.uth.tmc.edu/s3/chapter01.html

Conclusion

- The brain evolved within a body
- Neurons take current as input, output spikes, maintain a membrane potential
- Synapses transform spikes into current (Post-Synaptic Potential, PSP)
- Spiking neural networks model spikes, while analog networks model spike rates
- Spiking neural network are dynamical systems defined with respect to time
- There are many ways to encode and decode information with spikes



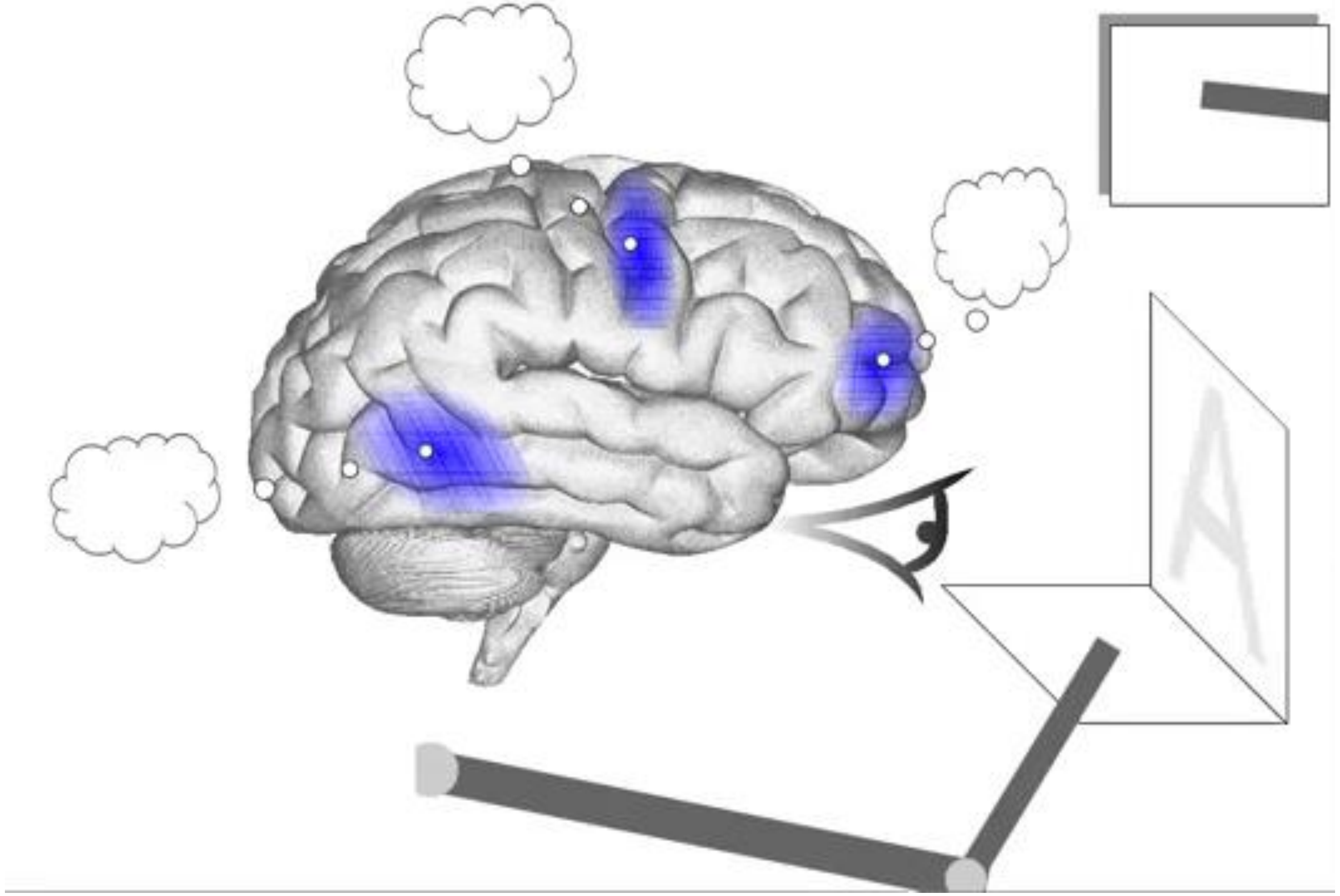
Danke!

Fragen???

Spiking Neural Networks
Gepulste Neuronale Netze

M. Sc. Juan Camilo Vasquez Tieck
M. Sc. Jacques Kaiser
Dr.-Ing. Stefan Ulbrich
Prof. Dr.-Ing. J. Marius Zöllner
Prof. Dr. R. Dillmann

Motivation – SPAUN, the most complex artificial brain

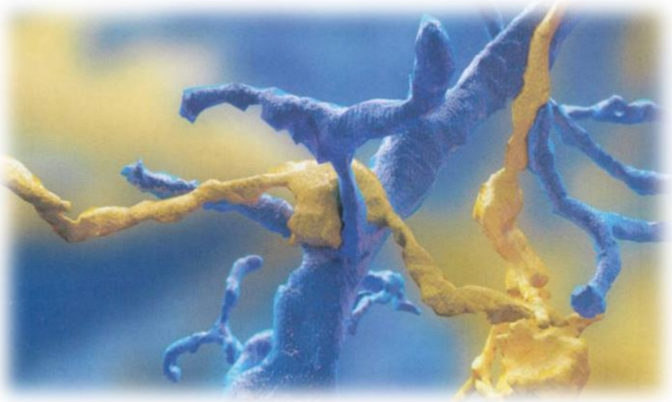


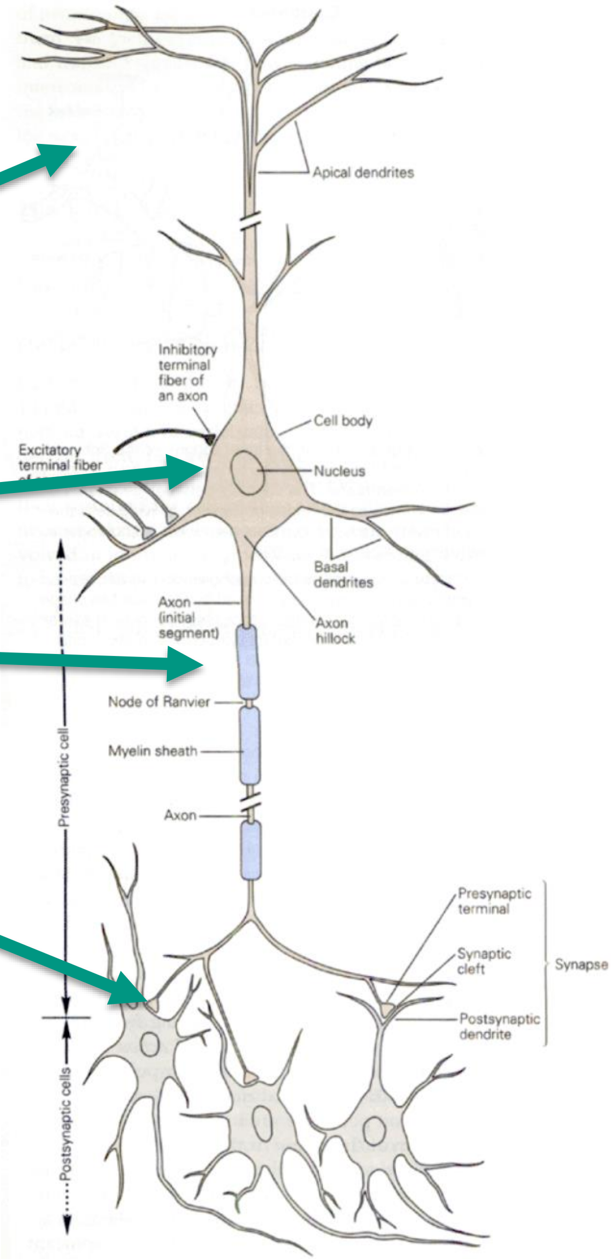
Overview

- Insights from biology
  - The Human Brain
  - The Cortex
  - Neurons and Synapses
- Modelling biology
  - Artificial Neural Networks
  - Spiking Neural Networks
  - Spiking Neurons
- Neural Coding
  - Common Coding Schemes
  - Input / Output Representation
- Learning with spiking networks - theory
  - Differences with analog networks
  - Formalization of plasticity
- Learning with spiking networks - practice
  - STDP in action
  - Spiking networks as kernel methods
- Materializing spiking networks
  - Neuromorphic hardware
  - Neurorobotics

Let’s recap – Neurons anatomy

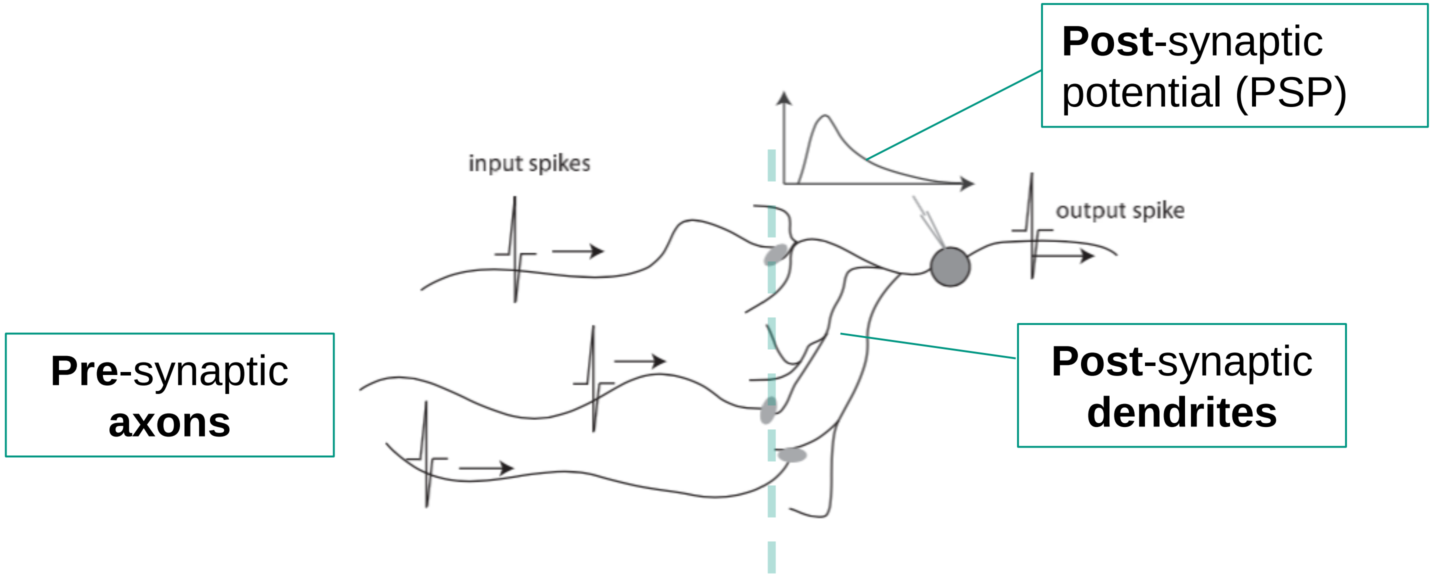
- Different types
- Structure of a neuron:
  - Dendrites – input
  - Soma - summation
  - Axon - output
  - Synapses - connection





Let‘s recap – Neurons in action

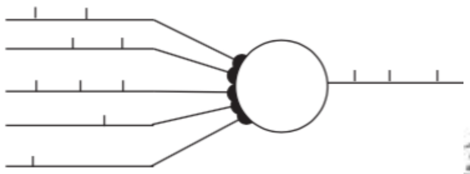
- Strength of post-synaptic potential (PSP) depends on:
  - Amount of neurotransmitters in axon
  - Number of ion channels (receptors) in dendrites
  - In simulators, abstracted by **synaptic strength (weight)**
- Plasticity: change in one of these quantities

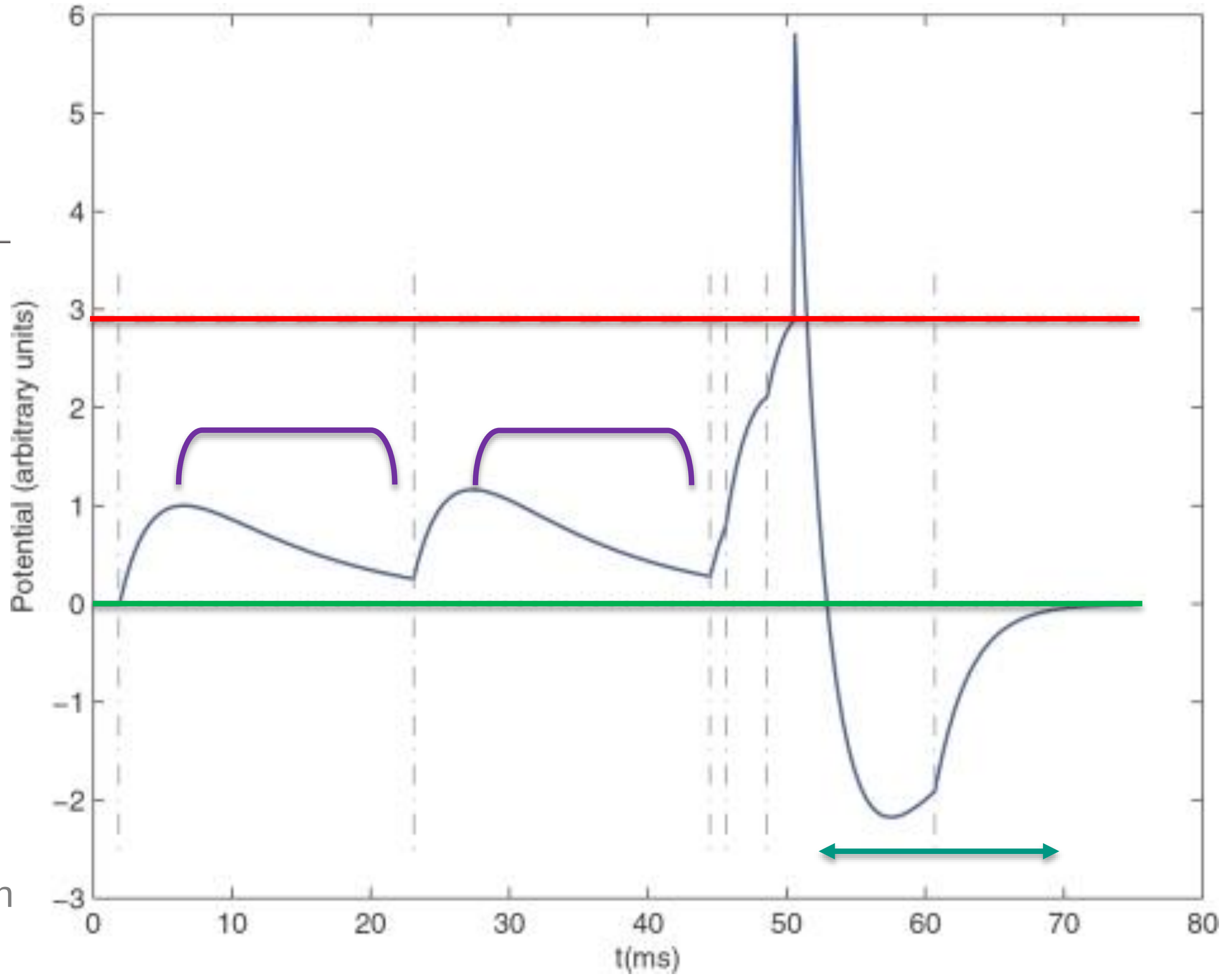


Let‘s recap - Modelling neurons

- Differential equations with respect to time
- PSP shape (kernel)
- Input:
  - Current
- Output:
  - Spikes
- Variables:
  - Membrane potential: $V(t)$
- Parameters:
  - Threshold: V_{th} or ϑ
  - Resting potential: V_{rest}
  - Leak (membrane time constant): τ_m
  - Refractory period: τ_{ref}

- Point-neuron

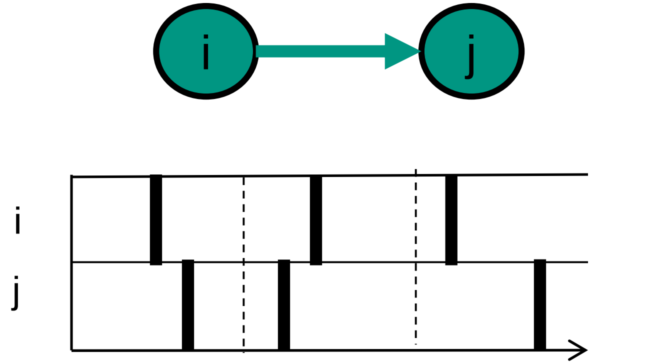




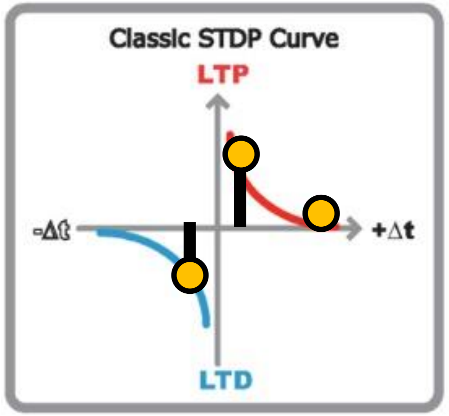
“Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains.”, Masquelier et al., PLoS ONE 2008

Let‘s recap - Synaptic plasticity

- Plasiticy depends on precise timing of spikes
- LTP – Long Term Potentiation (+)
- LTD – Long Term Depression (-)
- Hebbian rule:
  - “Neurons who fire together wire together”
  - Learning is local and incremental



Spike train



“A history of spike-timing-dependent plasticity”,
Henry Markram et. al

Institut für Anthropomatik und Robotik (IAR)
M. Sc. Jacques Kaiser
Maschinelles Lernen II

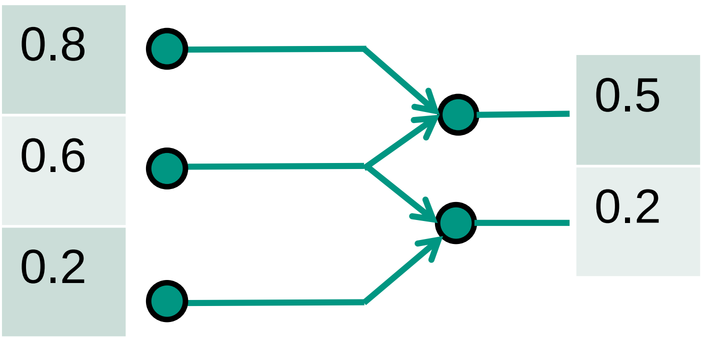
Let‘s recap – Coding of information

Analog neural network

- Static
- Spatial
- No concept of time

Trivial definition:

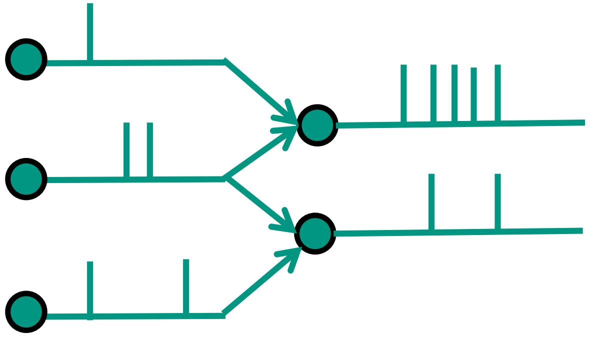
$$h\colon input\ vector \ \rightarrow output\ vector$$



Spiking neural network

- Dynamic
- Spatio-temporal
- Defined with respect to time

How to define the processing?
How to **encode** the input?
How to **decode** the output?



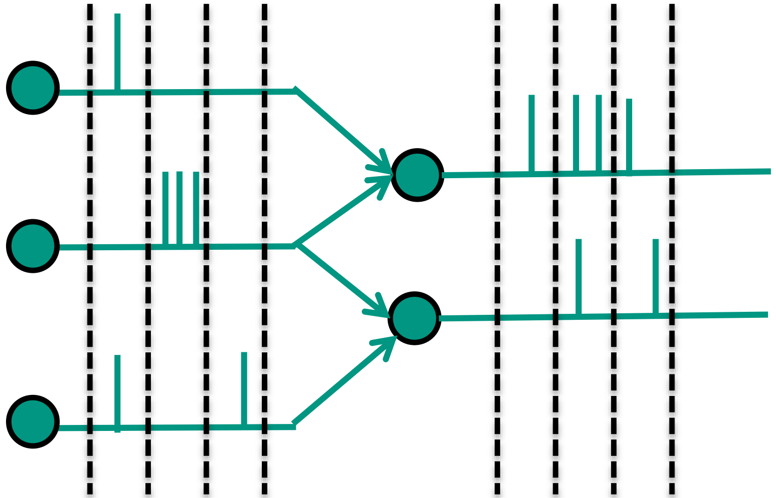
Overview

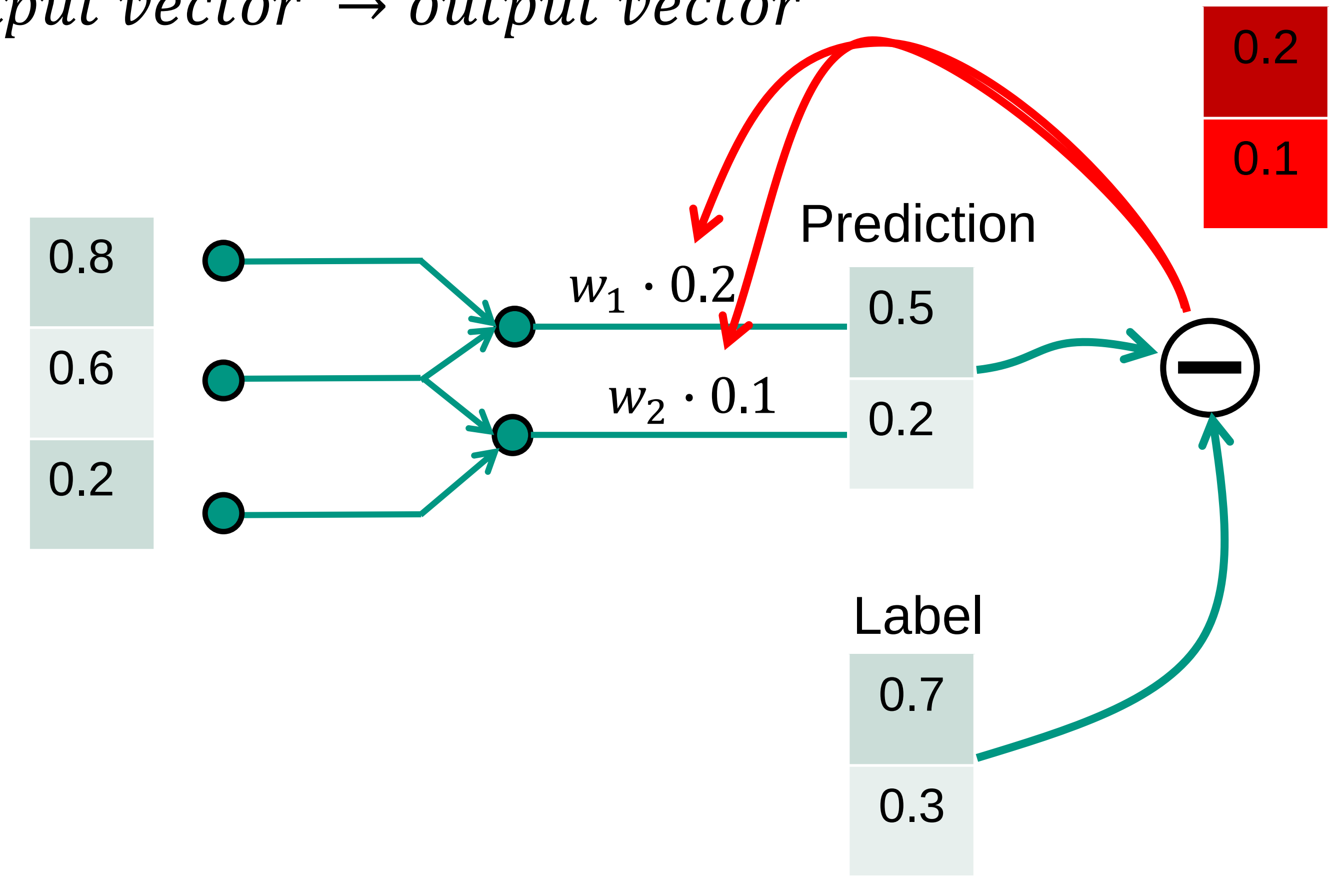
- Insights from biology
  - The Human Brain
  - The Cortex
  - Neurons and Synapses
- Modelling biology
  - Artificial Neural Networks
  - Spiking Neural Networks
  - Spiking Neurons
- Neural Coding
  - Common Coding Schemes
  - Input / Output Representation
- Learning with spiking networks - theory
  - Differences with analog networks
  - Formalization of plasticity
- Learning with spiking networks - practice
  - STDP in action
  - Spiking networks as kernel methods
- Materializing spiking networks
  - Neuromorphic hardware
  - Neurorobotics

Why not backpropagation in spiking networks?

- We need a formalization like $h\!: input\ vector \rightarrow output\ vector$
- Spikes are not differentiable
- Only viable with rate-coding

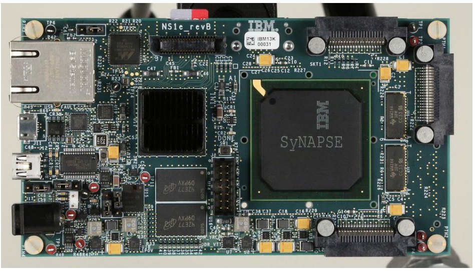
$$\frac{1}{\Delta t} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad \frac{1}{\Delta t} \times \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$





Is backprop biologically plausible?

- There are many „conversion“ papers, which train in analog networks with backprop and then convert to spiking. E.g, IBM TrueNorth:



Core problems with biological plausability:

- The computation would need to be precisely clocked to alternate between feedforward and backward phases
- Information (the error) does not travel backward in synapses
  - If the error travels in recurrent connections, then these connections need to know about the feedforward weights

“Towards Biologically Plausible Deep Learning”, Yoshua Bengio et al., arXiv, 2015

Synaptic plasticity as learning

- Hebb‘s postulate: learning is local and cooperative
- Local → The weights are adjusted with respect to local information
- Cooperative → The weights are adjusted on simultaneous activation

Short-term	Long-term
• Order of millisecond • Network stabilization	• Order of minutes, days • Learning, memory

- We will only consider long-term plasticity

http://www.scholarpedia.org/article/Short-term_synaptic_plasticity

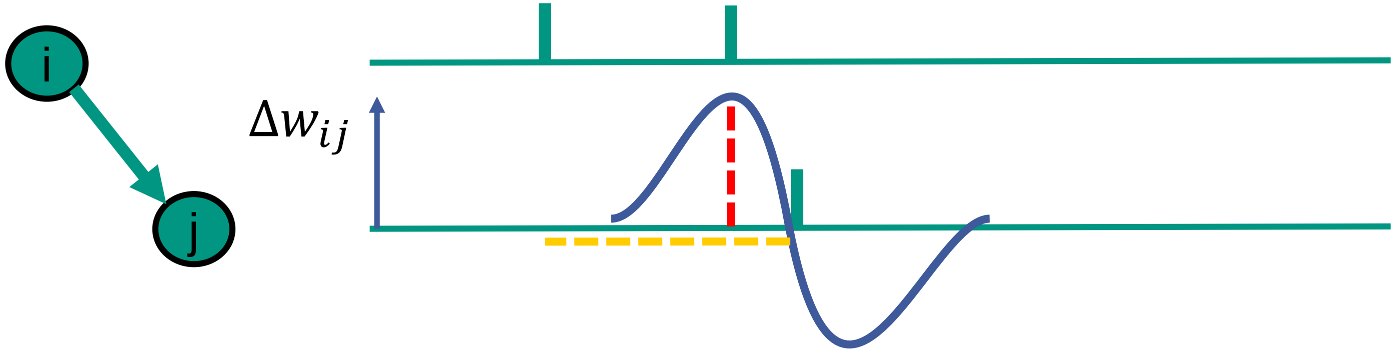
Different types of long-term synaptic plasticity

- Spike-timing-dependent plasticity
  - Depends on relative spike-timing of post synaptic neurons
- Rate based plasticity
  - Depends on rate (frequency) of pre- and postsynaptic firing
- Reward based plasticity
  - Plasticity controlled by global a reward signal (neuromodulator dopamine)
- Structural plasticity
  - Learning by rewiring connections instead of just changing the synaptic weights

Formalization of precise spiketime plasticity rules

- Weight updates depend on the spike times of the pre- and post-synaptic neurons

$$\Delta w_{ij} = \sum_{t_i^{pre}, t_j^{post}} W\left(t_j^{post} - t_i^{pre}\right)$$



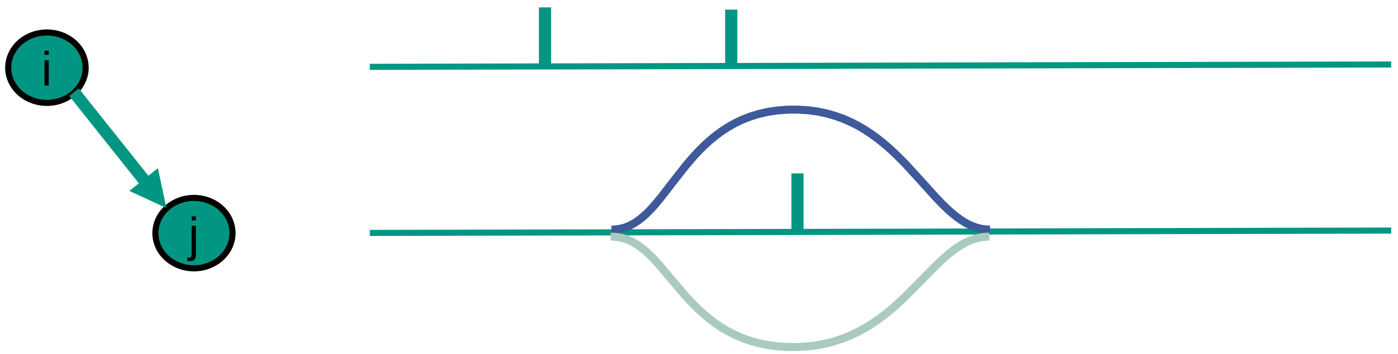
- A common choice is $W(x) = \begin{cases} A_+ \exp\left(\frac{-|x|}{\tau_+}\right) & if\ x > 0 \\ A_- \exp\left(\frac{-|x|}{\tau_-}\right) & if\ x < 0 \end{cases}$

ipython notebook exercise

Example: Hebbian and anti-hebbian learning rules

- Weight updates depend on the spike times of the pre- and post-synaptic neurons

$$\Delta w_{ij} = \sum_{t_i^{pre}\, t_j^{post}} W\left(t_j^{post} - t_i^{pre}\right)$$



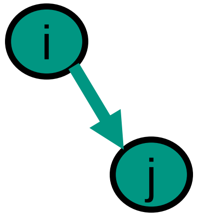
$$W(x) = \begin{cases} A_+ \exp\left(\frac{-|x|}{\tau_+}\right) & if\ x > 0 \\ A_- \exp\left(\frac{-|x|}{\tau_-}\right) & if\ x < 0 \end{cases}$$

- Hebbian learning: $W(x) = A \cdot \exp\left(\frac{-|x|}{\tau}\right) \quad with\ A > 0$
- Anti-hebbian learning: $W(x) = A \cdot \exp\left(\frac{-|x|}{\tau}\right) \quad with\ A < 0$

Formalization of rate-based plasticity rules

- Weight updates depend on rates of the pre- and post-synaptic neurons

$$\frac{d}{dt} w_{ij} = F(w_{ij}, v_i^{pre}, v_j^{post})$$



- With v_i, v_j the activity of neurons i, j
- Common choices for F:

v_i	v_j	$v_i v_j$	$v_i v_j - c_0$	$v_i(v_j - v_\theta)$	$v_j(v_i - v_\theta)$	$(v_i - \langle v_i \rangle)(v_j - \langle v_j \rangle)$
ON	ON	+	+	+	+	+
ON	OFF	0	-	-	0	-
OFF	ON	0	-	0	-	-
OFF	OFF	0	-	0	0	+

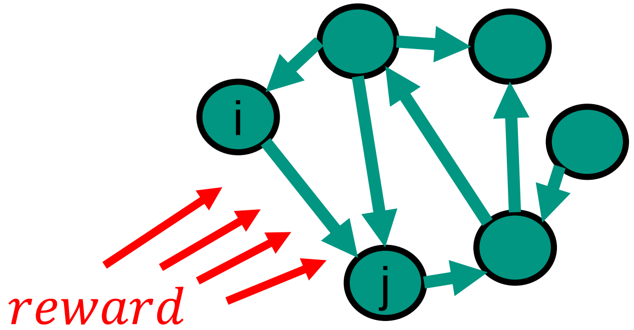
http://icwww.epfl.ch/~gerstner/SPNM/node72.html

Formalization of reward-based plasticity rules

- Weight updates depend (additionally) on a global reward signal
- In the brain, the reward is the neuromodulator dopamine (DA)
- For spike-time dependent plasticity:

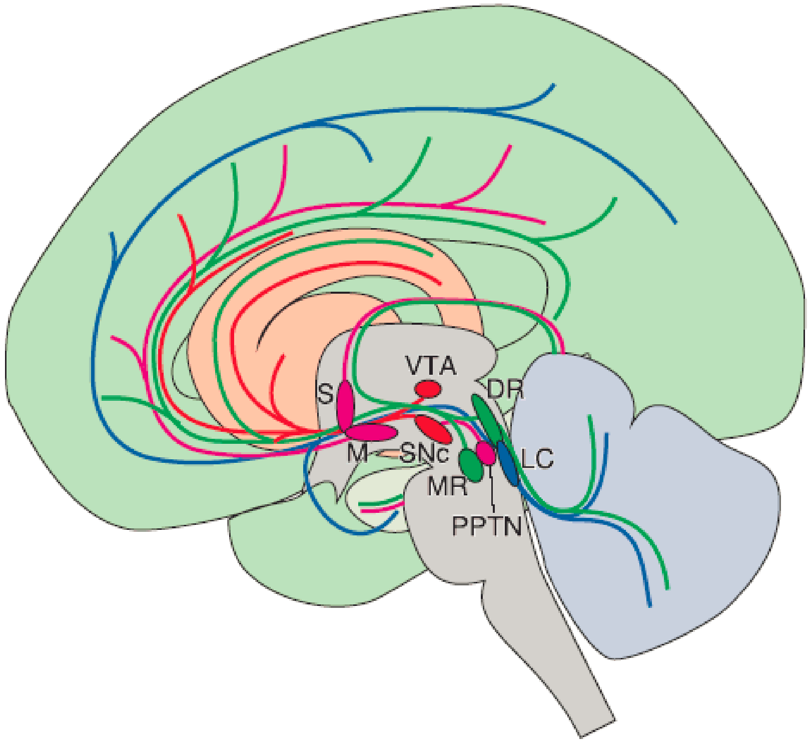
$$\Delta w_{ij} = \sum_{t_i^{pre} t_j^{post}} W\left(t_j^{post} - t_i^{pre}, \color{red}{reward}\right)$$

- **Credit Assignment Problem**: which ones of the 10^{15} synapses in the cortex contributed to the currently successful (and internally rewarded) behavior?



Other neuromodulators

- Dopamine is not the only top-down signal that affects learning
- There are other neuromodulators: Serotonin, Acetylcholine, Norepinephrine, …
- They provide information about punishments, curiosity, alertness, anxiety, …



http://www.igi.tugraz.at/lehre/PrinciplesOfBrainComputation/SS16/

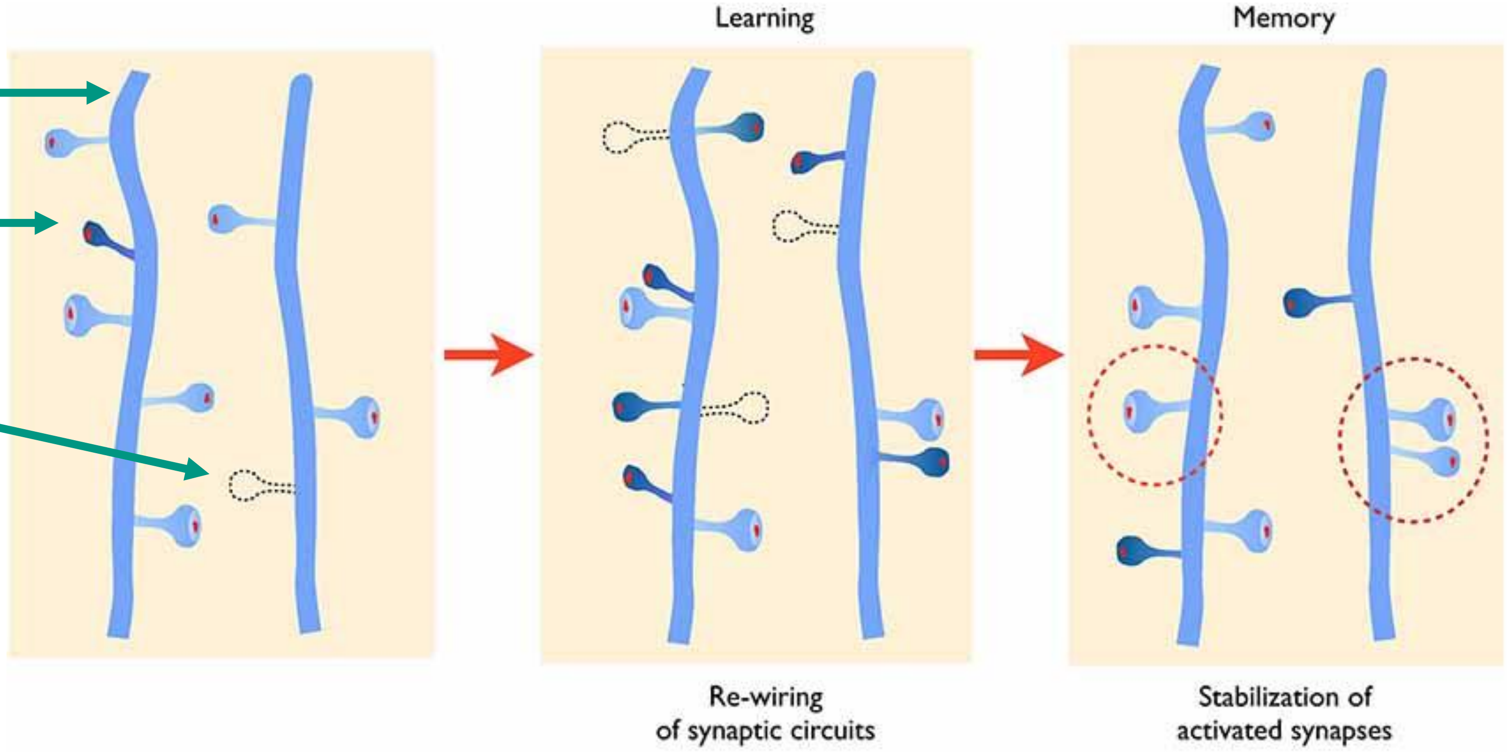
Structural plasticity – rewiring synapses

- So far, we treated plasticity as change of synaptic weights
- Structural plasticity: creation and destruction of synapses
- Timescale of hours or days

Dendrite

New spine

Destructed spine



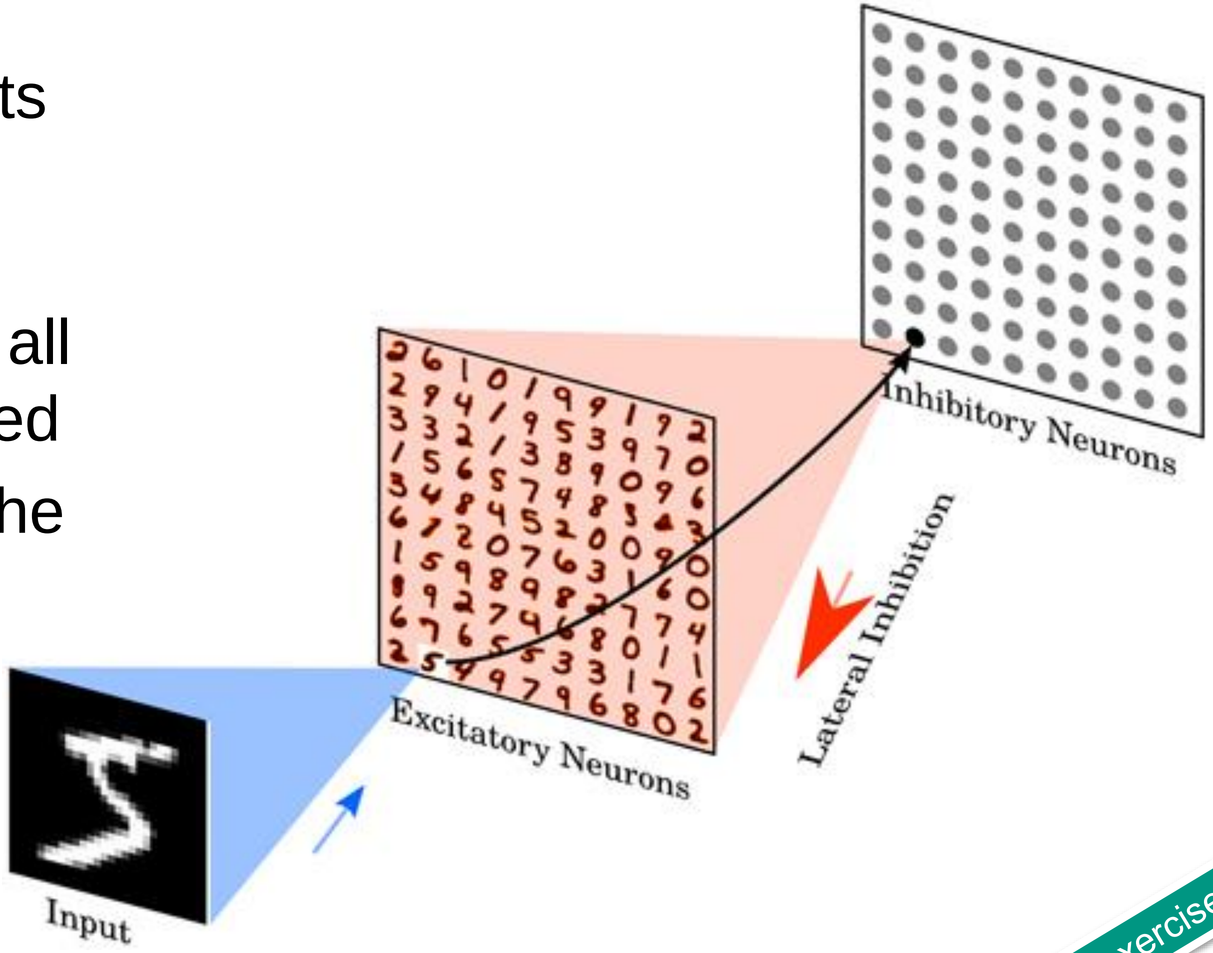
“Structural plasticity: mechanisms and contribution to developmental psychiatric disorders”, Bernadinelli et al., Frontiers in Neuroanatomy 2014

Overview

- Insights from biology
  - The Human Brain
  - The Cortex
  - Neurons and Synapses
- Modelling biology
  - Artificial Neural Networks
  - Spiking Neural Networks
  - Spiking Neurons
- Neural Coding
  - Common Coding Schemes
  - Input / Output Representation
- Learning with spiking networks - theory
  - Differences with analog networks
  - Formalization of plasticity
- Learning with spiking networks - practice
  - STDP in action
  - Spiking networks as kernel methods
- Materializing spiking networks
  - Neuromorphic hardware
  - Neurorobotics

Unsupervised learning with STDP

- All-to-all connections between inputs and excitatory neurons, with randomized weights
- When an excitatory neuron spikes, all other excitatory neurons are inhibited
- A neuron is assigned the label for the class it is most active
- 95% accuracy on MNIST

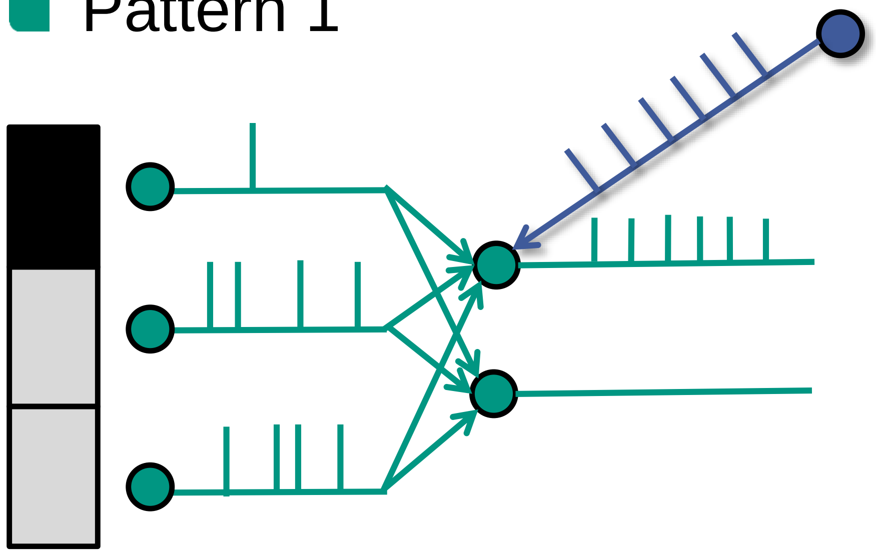


„Unsupervised learning of digit recognition using spike-timing-dependent plasticity",
Peter U. Diehl and Matthew Cook,
IEEE Transactions in Neural Networks and Learning Systems, 2014

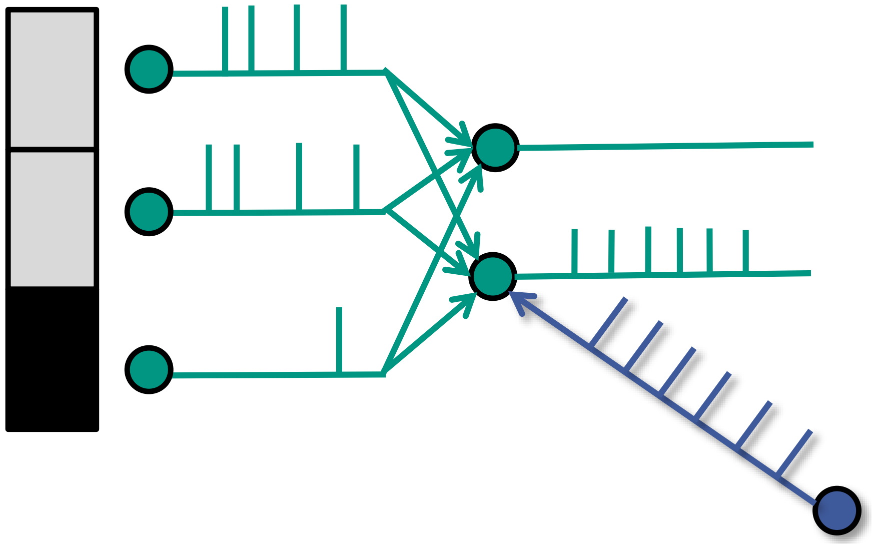
ipython notebook exercise

Supervised learning with STDP – Associative learning

- Pattern 1



- Pattern 2



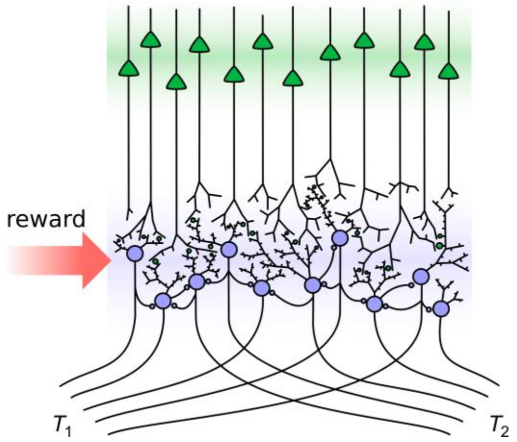
- Initialize with small weights
- Force spiking output with a teaching signal
  - STDP will strengthen correlated connections, and weaken the others
- Evaluate by removing teaching signal

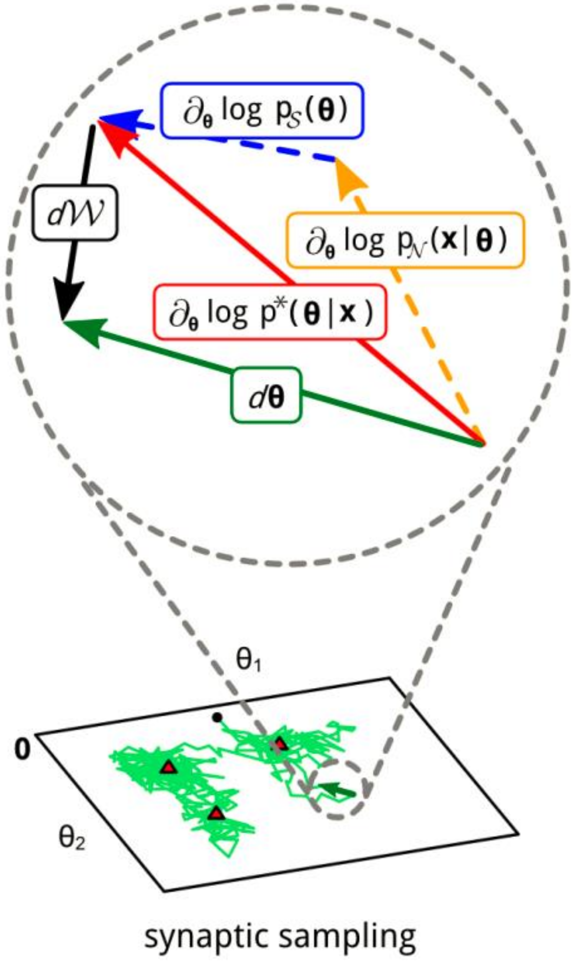
SPORE –
Synaptic Plasticity with Online Reinforcement learning

- Agnostic of the network topology
- Reward-based structural plasticity
- A temperature T term controlling the exploration

$$d\theta_i = b\left(\frac{1}{\sigma^2}(\theta_i - \mu_i) + \frac{\delta}{\delta\theta_i} \cdot log\, p_n(R = 1|\theta)\right) + \sqrt{2bT}\, dw_i$$

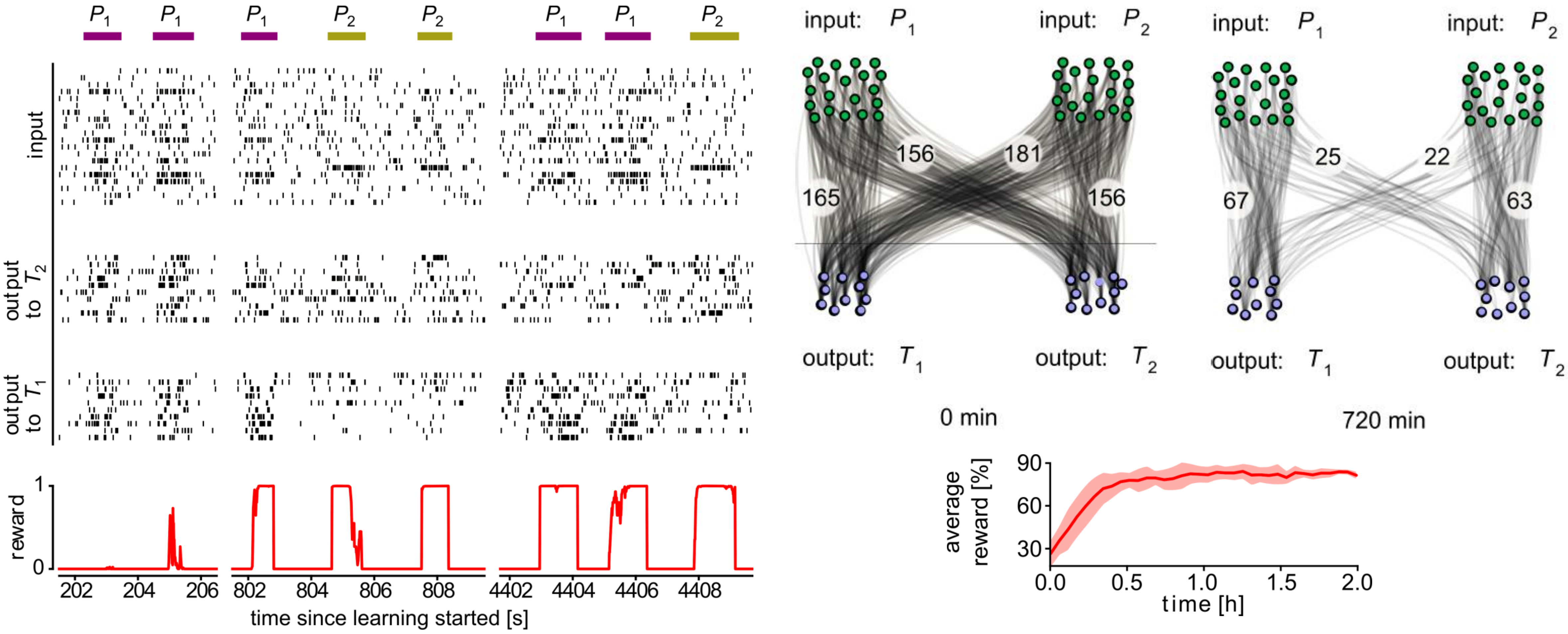
drift reward explore





„Reward-based stochastic self-configuration of neural circuits",
Kappel et al., arxiv 2017

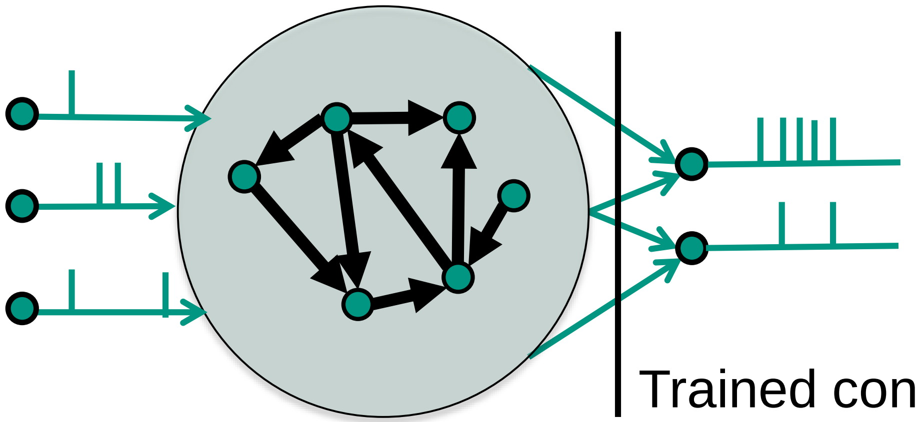
SPORE in action – Binary classification



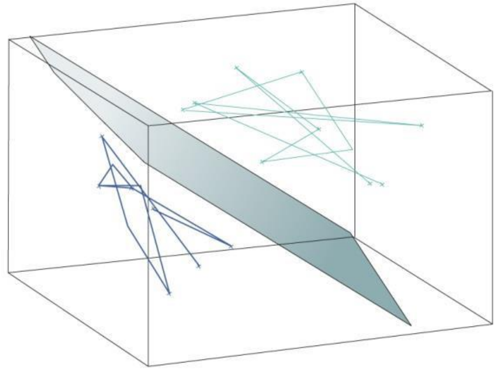
Try it yourself: https://github.com/IGITUGraz/spore-nest-module

Spiking networks as kernel methods

- Three components: input, hidden, output
- The **hidden** component projects the input to a high dimensional space
- Only the connections from **hidden to output** are trained with linear supervised learning
- Usually, weights are computed in a closed-form, not learned



Trained connections
(linear decoders)

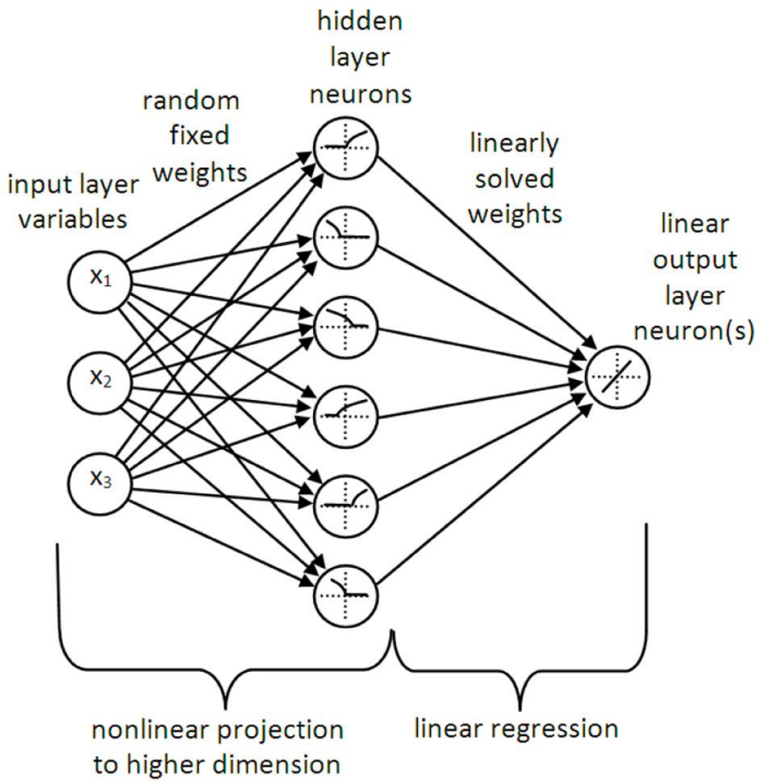


„Synthesis of neural networks for spatio-temporal spike pattern recognition and processing", Tapson et al., Frontiers in Neuroscience, 2013

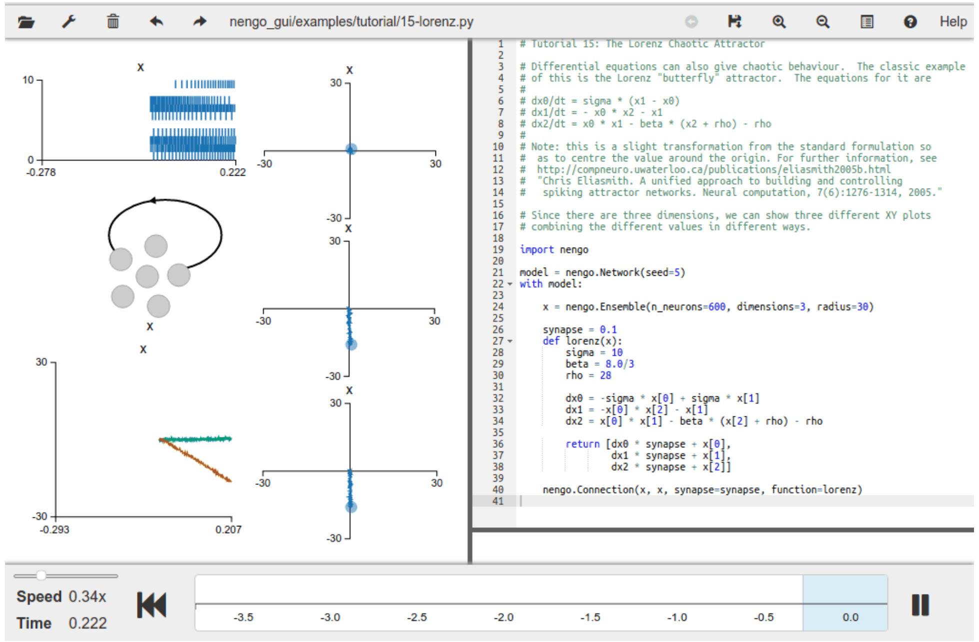
- Examples: Neural Engineering Framework (NEF), Liquid State Machines (LSM)

Linear regression on rates – Neural Engineering Framework

- The output layer (decoder) is trained with respect to **spiking rates**
- Hidden component is one-layer feedforward



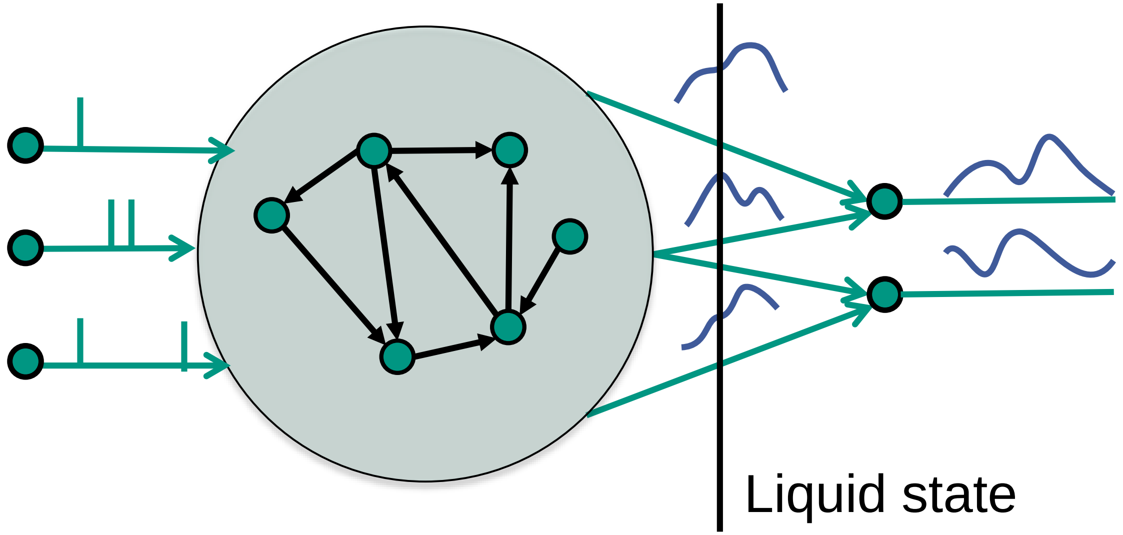
Nengo GUI



„A Technical Overview of the Neural Engineering Framework", Terrence Stewart, 2012

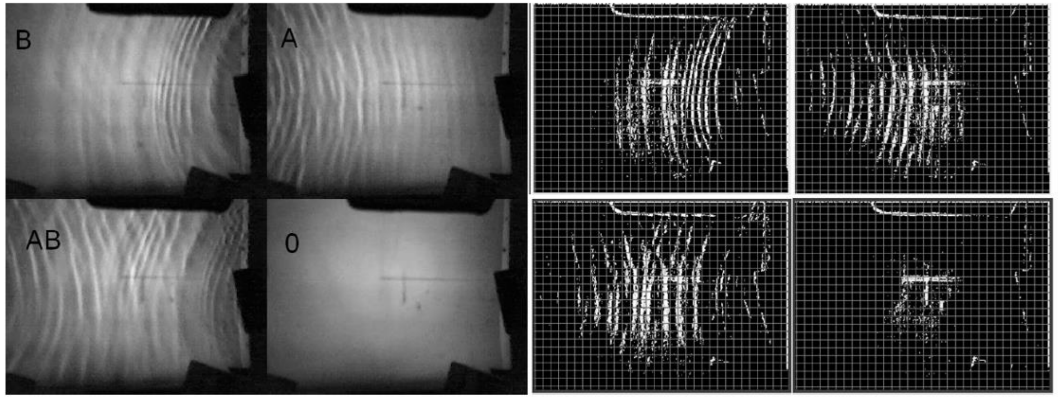
Regression on Post-Synaptic Potentials – Liquid State Machines

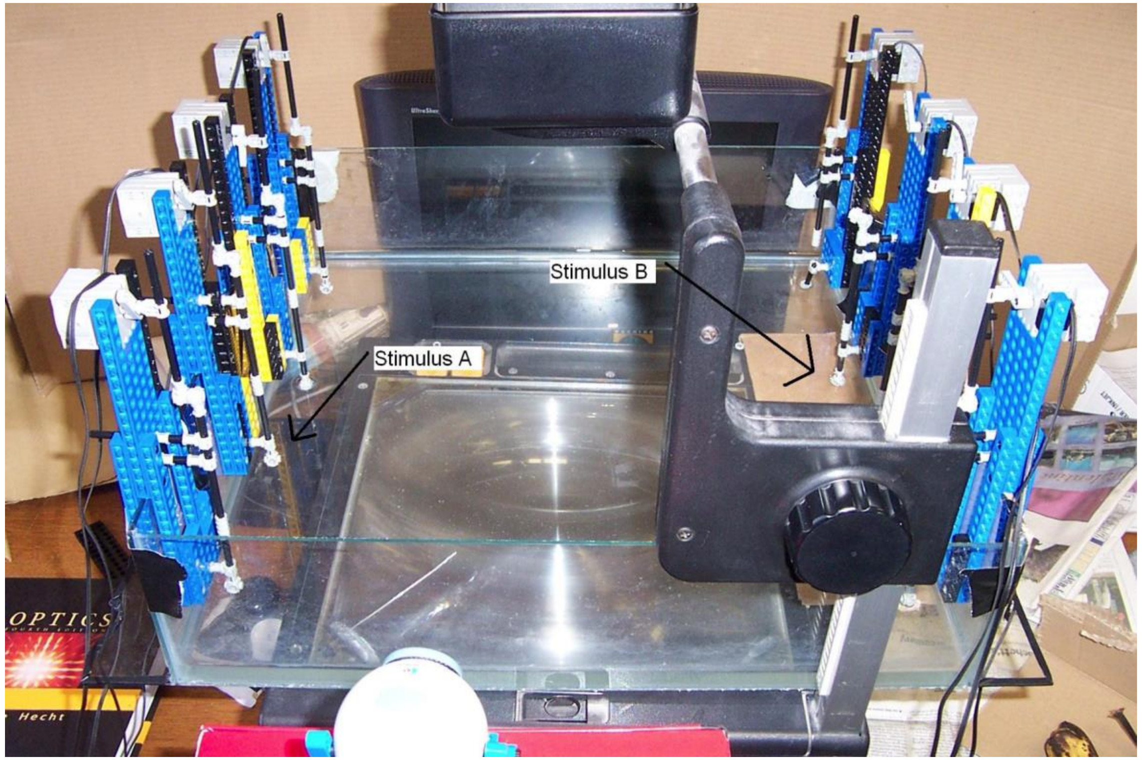
- The output layer (decoder) is trained with respect to **post-synaptic potentials**
- Hidden component is a recurrent reservoir (liquid)
- Arbitrary number of readouts
- Anytime computing: we receive the output before the input is completely streamed



Liquid State Machines in real life

- **Inputs**: left and right motors
- **Liquid**: actual water instead of neurons
- **Liquid state**: image observed by the camera (surface of the water)
- Can solve XOR and categorize speech data with linear regression





„Pattern Recognition in a Bucket", Chrisantha Fernando and Sampsa Sojakka, Advances in artificial life, 2013

Overview

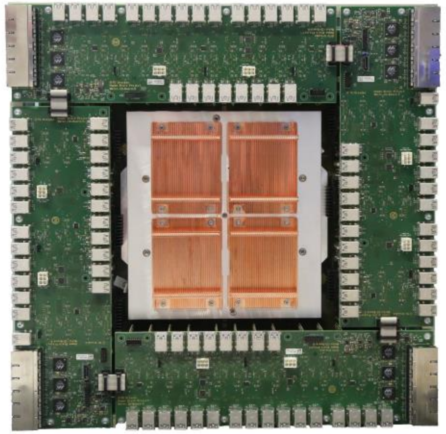
- Insights from biology
  - The Human Brain
  - The Cortex
  - Neurons and Synapses
- Modelling biology
  - Artificial Neural Networks
  - Spiking Neural Networks
  - Spiking Neurons
- Neural Coding
  - Common Coding Schemes
  - Input / Output Representation
- Learning with spiking networks - theory
  - Differences with analog networks
  - Formalization of plasticity
- Learning with spiking networks - practice
  - STDP in action
  - Spiking networks as kernel methods
- Materializing spiking networks
  - Neuromorphic hardware
  - Neurorobotics

Neuromorphic chips – spiking network

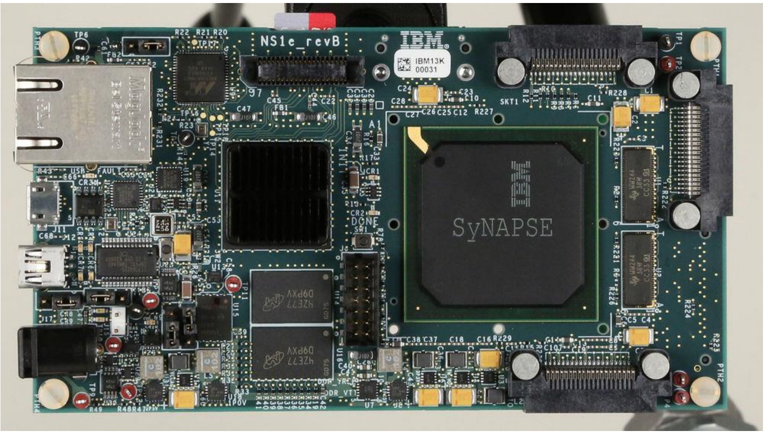
Most advantages of using spiking networks are only valid on special hardware:

- Computationally powerfull
- Energy-efficient
- Fault tolerant

- SpiNNaker



- BrainScales



- TrueNorth

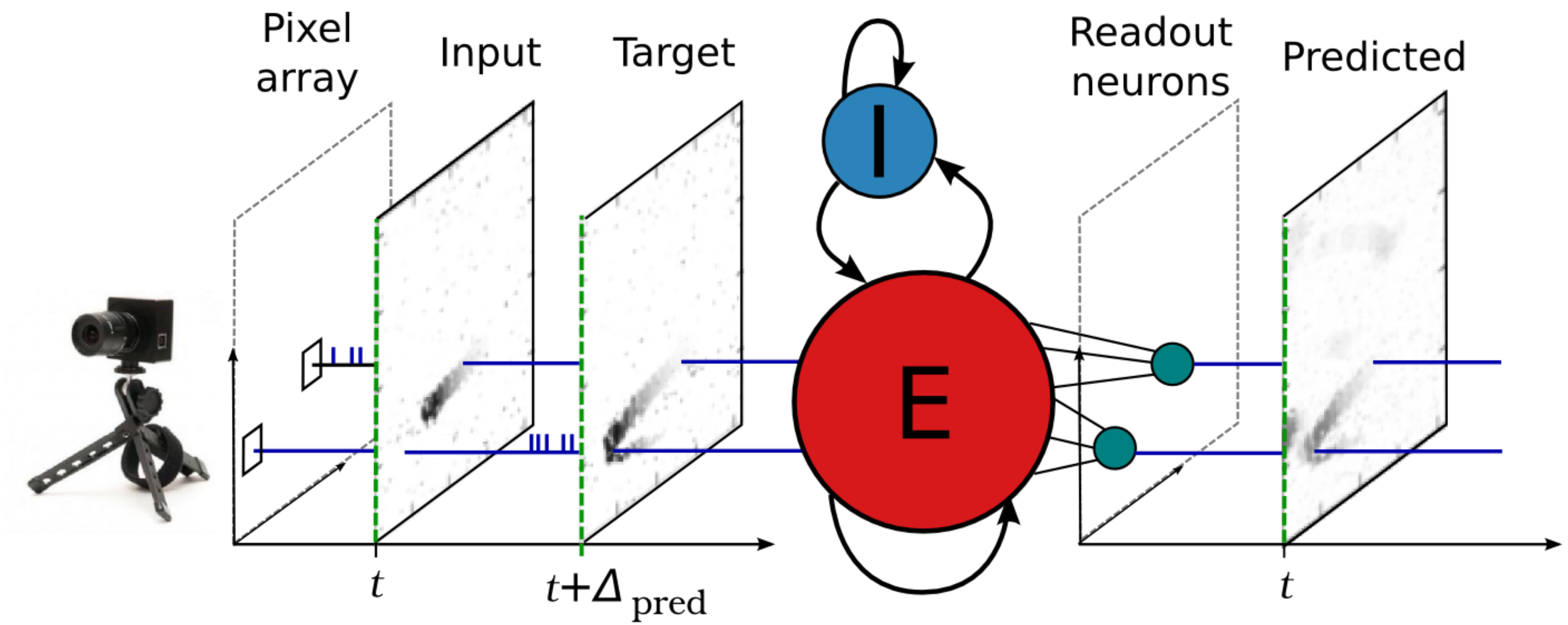
GPUs are good for parallel computation, not distributed (communication accross nodes is slow)

Neuromorphic sensors – Silicon retina (DVS)

- Biologically inspired silicon retina
- Asynchronous: individual pixel events are sent when a local change in light intensity is detected
- Frameless – no discrete timestep
  - Hard to interact with generation 2 neural networks
  - Easy to interact with spiking neural networks

Short-term visual prediction of address events

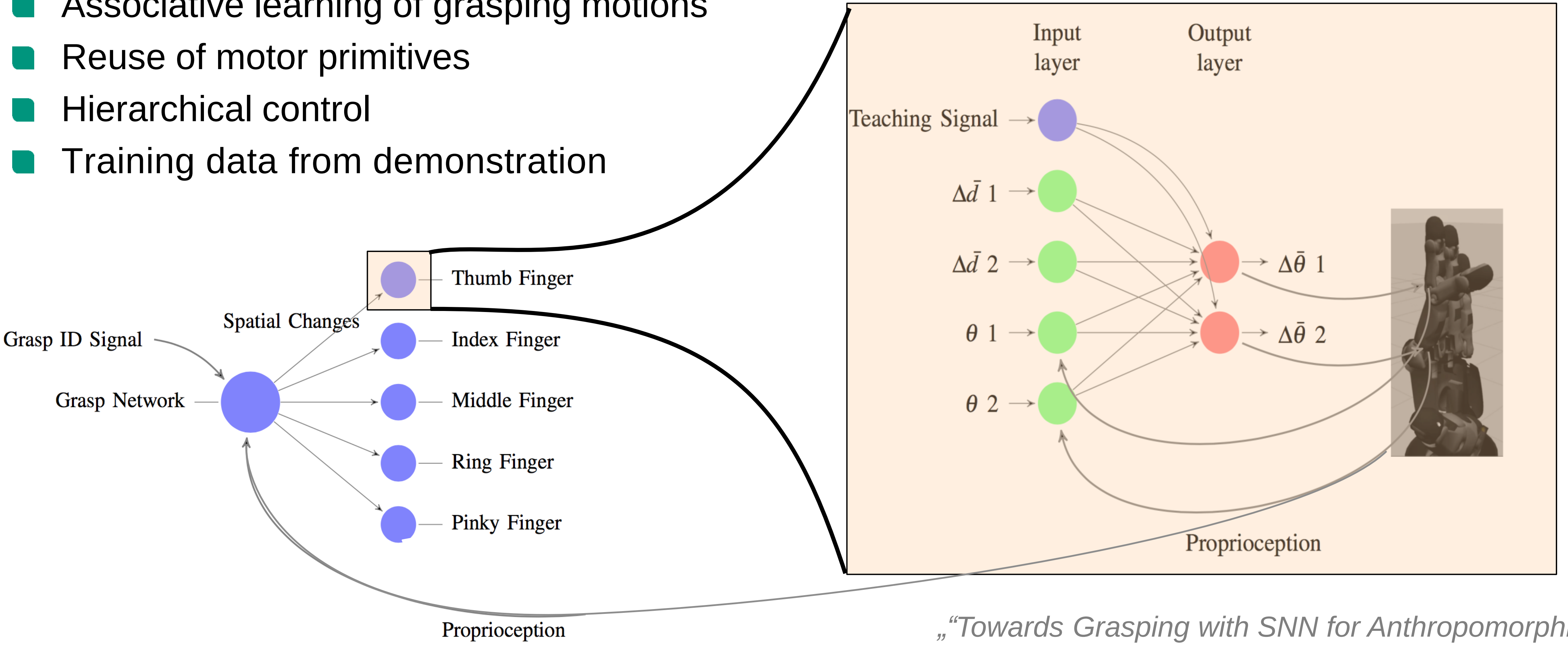
- Liquid state machine trained to predict addres events



„Scaling up liquid state machines to predict over address events from dynamic vision sensors", Kaiser et al, Bioinspiration & Biomimetics 2017

Connecting robots to spiking networks

- Associative learning of grasping motions
- Reuse of motor primitives
- Hierarchical control
- Training data from demonstration

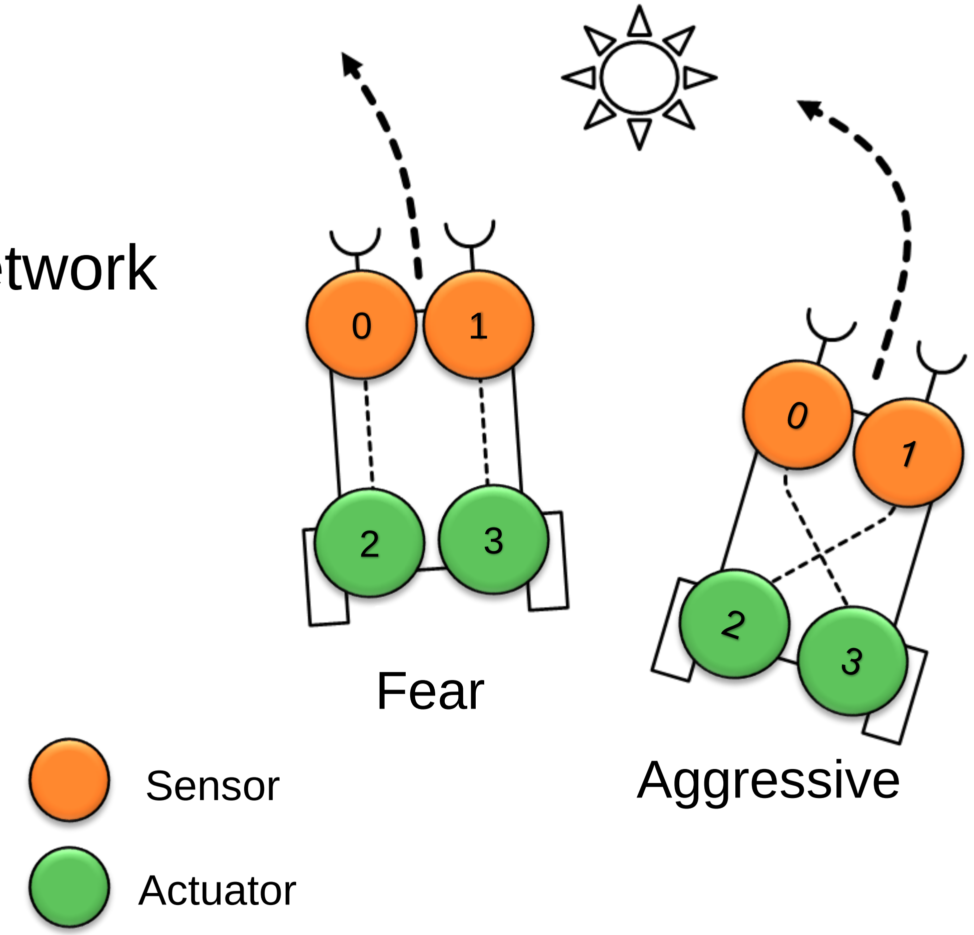


„"Towards Grasping with SNN for Anthropomorphic Robot Hands", Tieck et al., ICANN 2017

Grasping motions with spiking networks

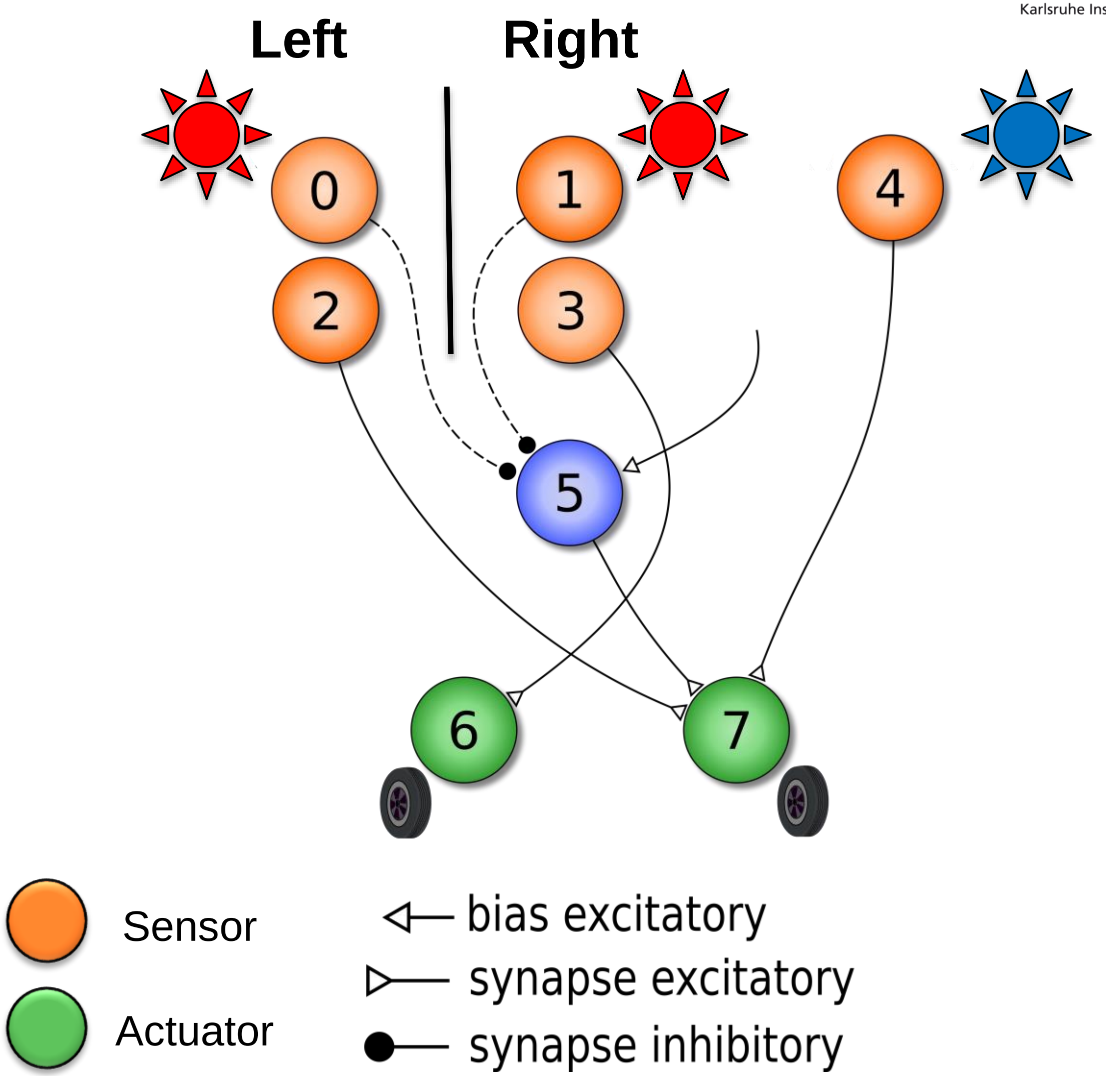
End-to-end robot control - Braitenberg vehicles

- Mobile agent
  - Two sensors – left and right
  - Two actuators – left and right
- „Intelligent“ behavior with a simple network
- Hardcoded weights



Building Braitenberg vehicles

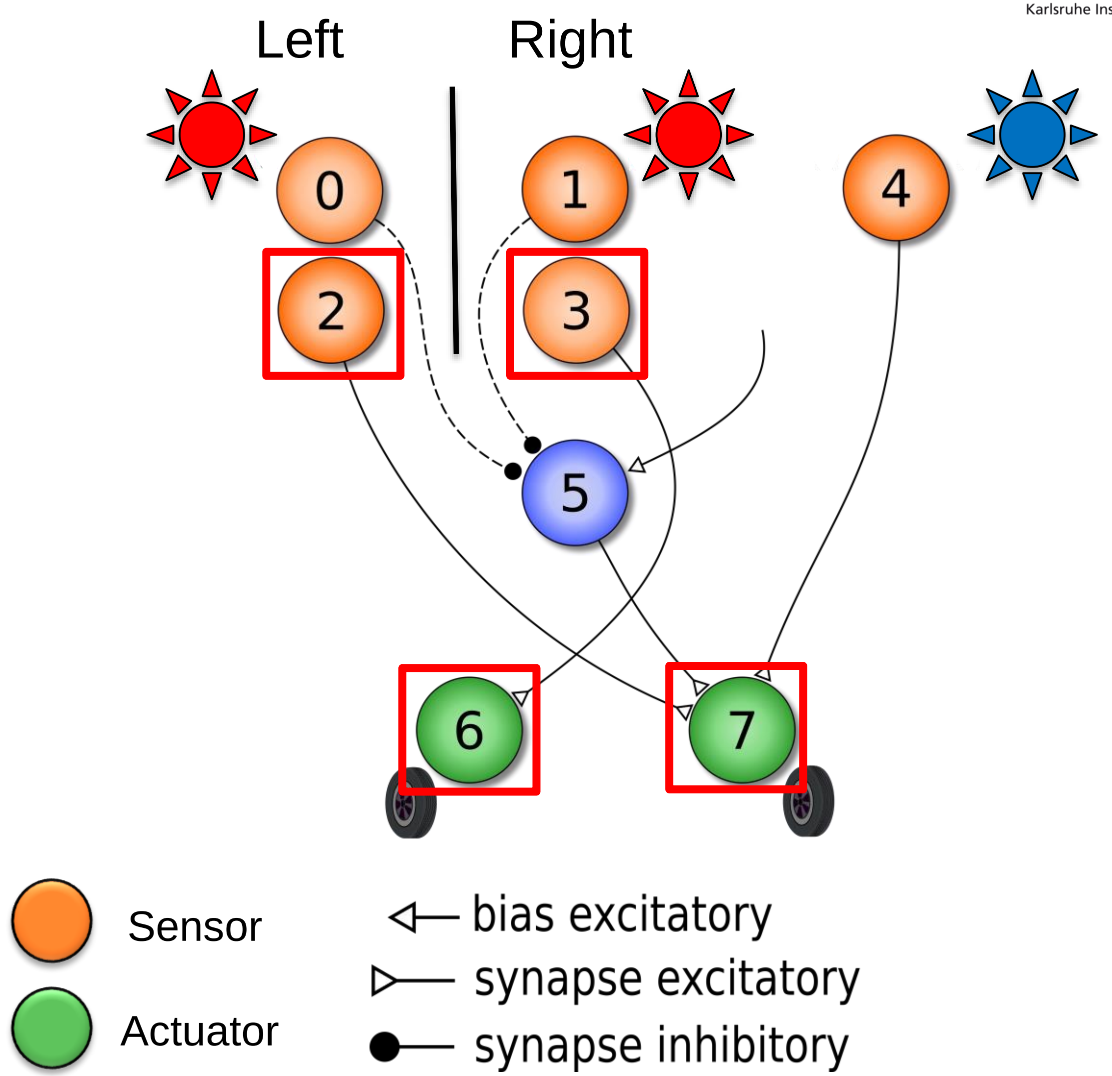
- What does this network do?



Beispiel: Braitenberg Vehicles

- What does this network do?

Classic aggressive

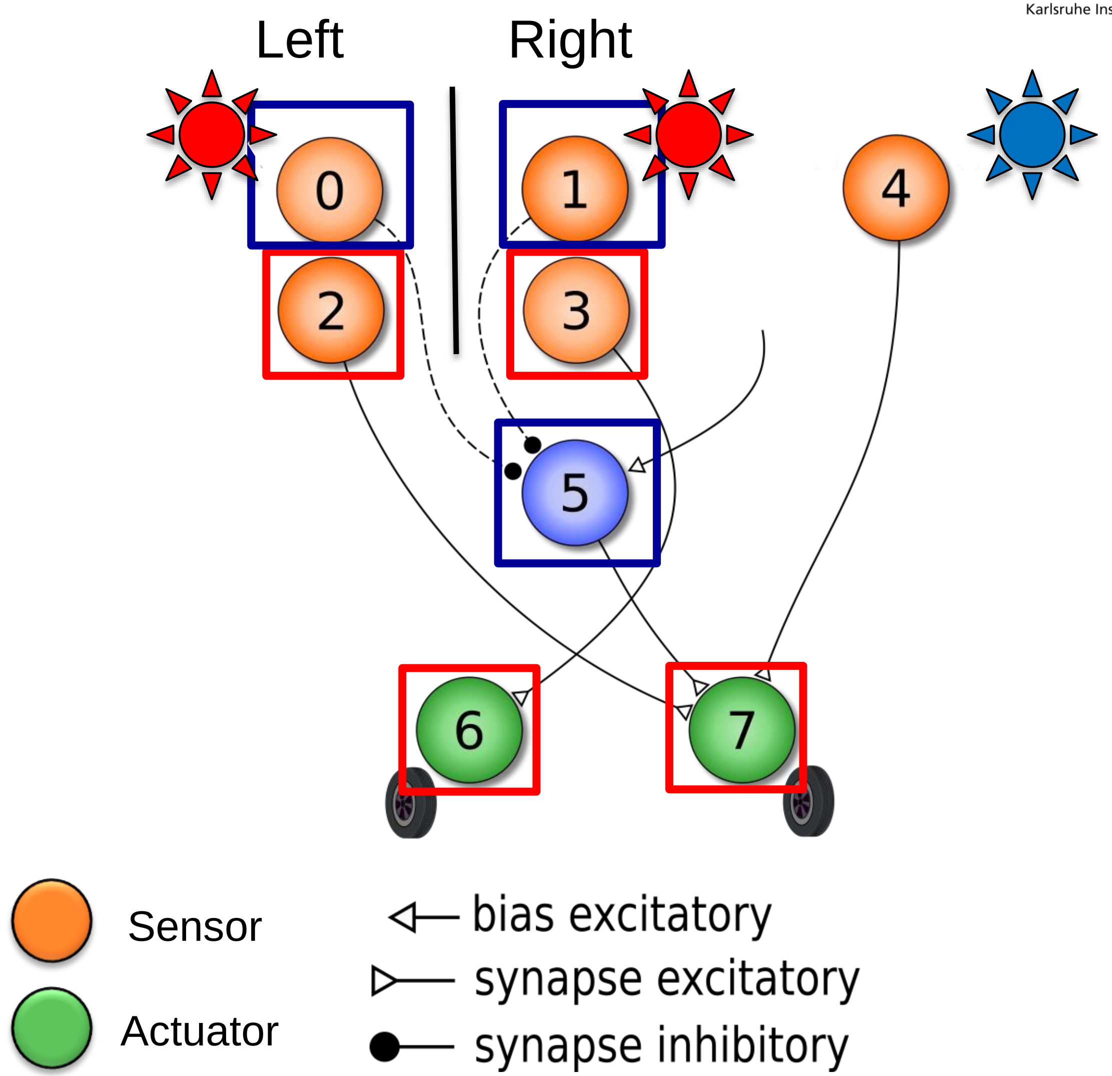


Beispiel: Braitenberg Vehicles

- What does this network do?

Classic aggressive

Bias, inhibited when red light



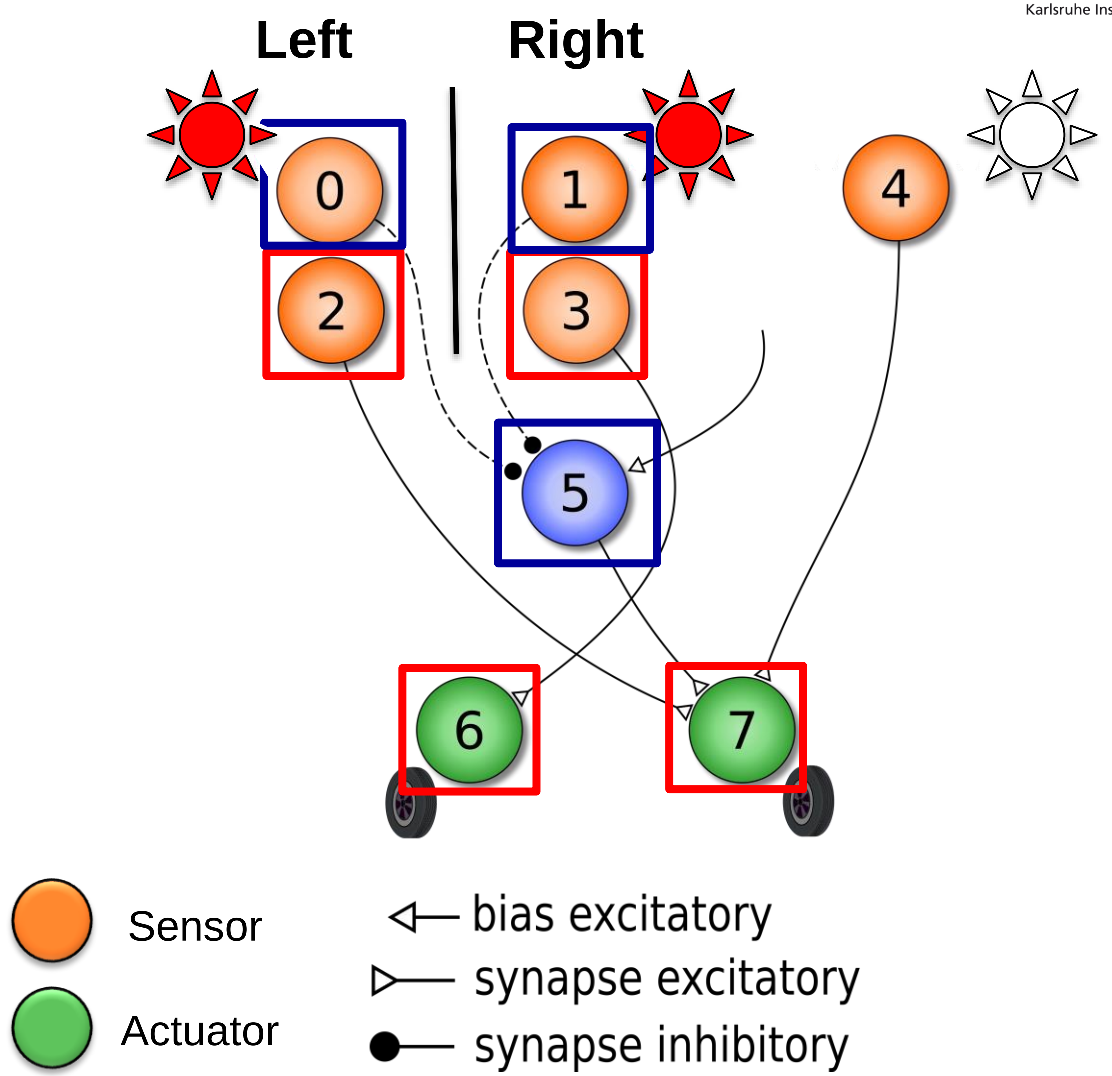
Building Braitenberg vehicles

- What does this network do?

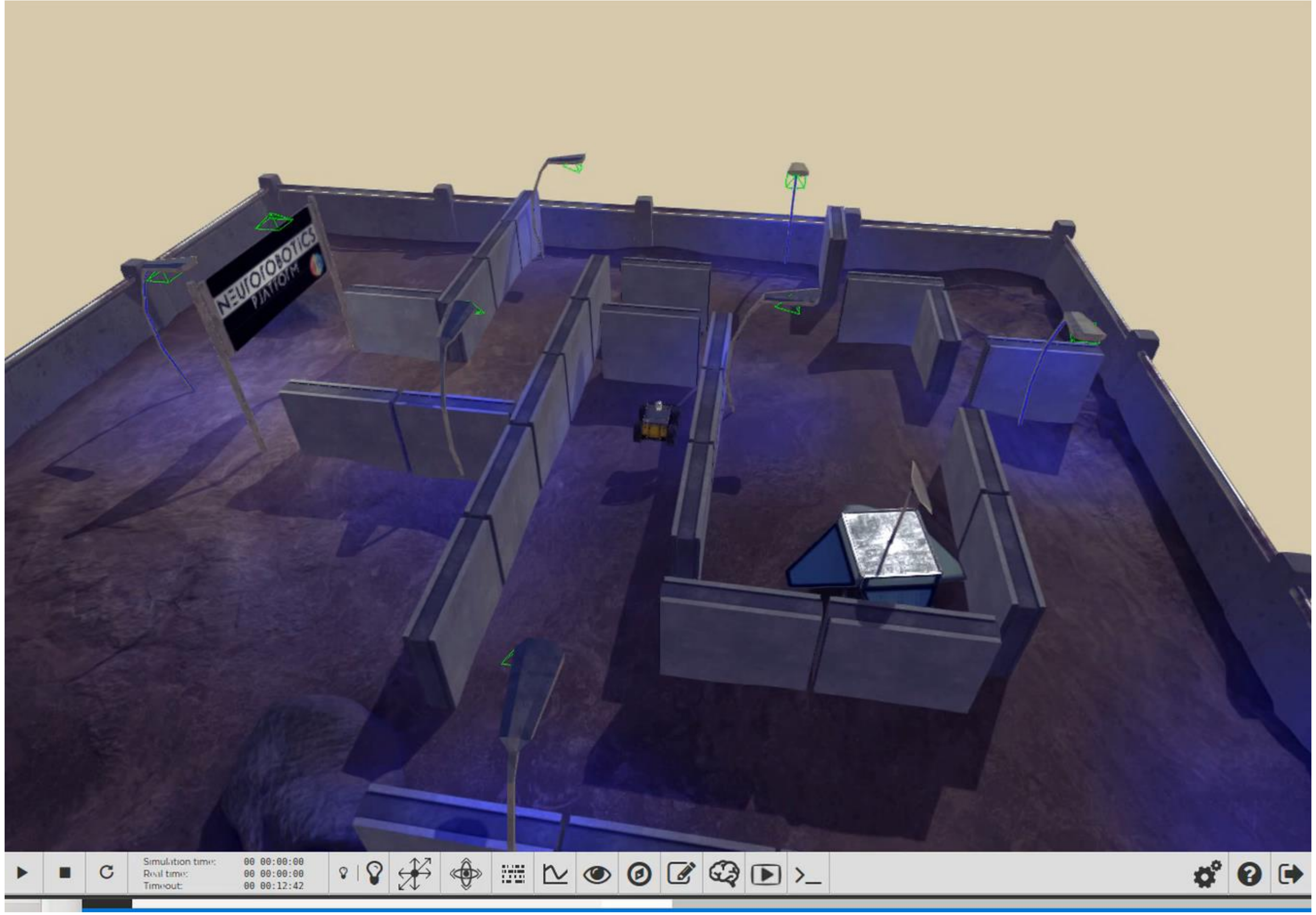
Classic aggressive

Inhibit bias when red light

- When no red light, turn left
- When red light, aggressive

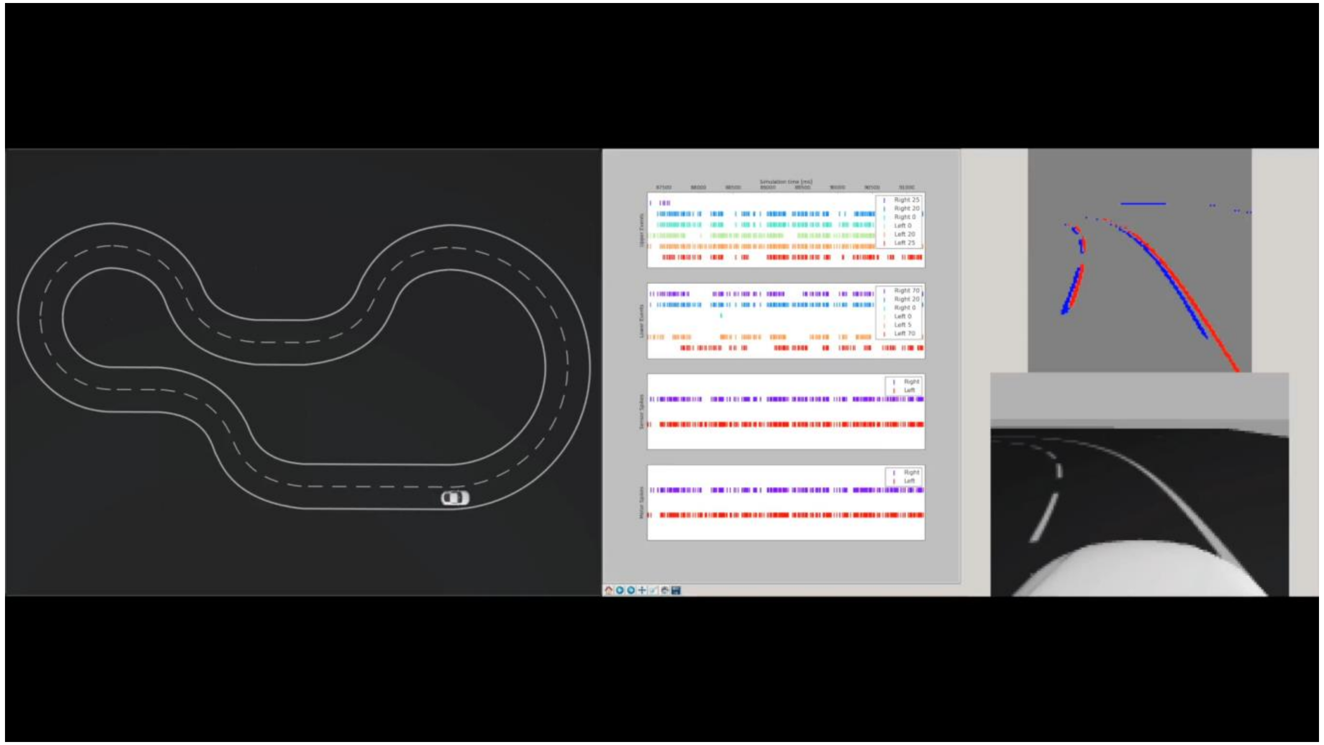


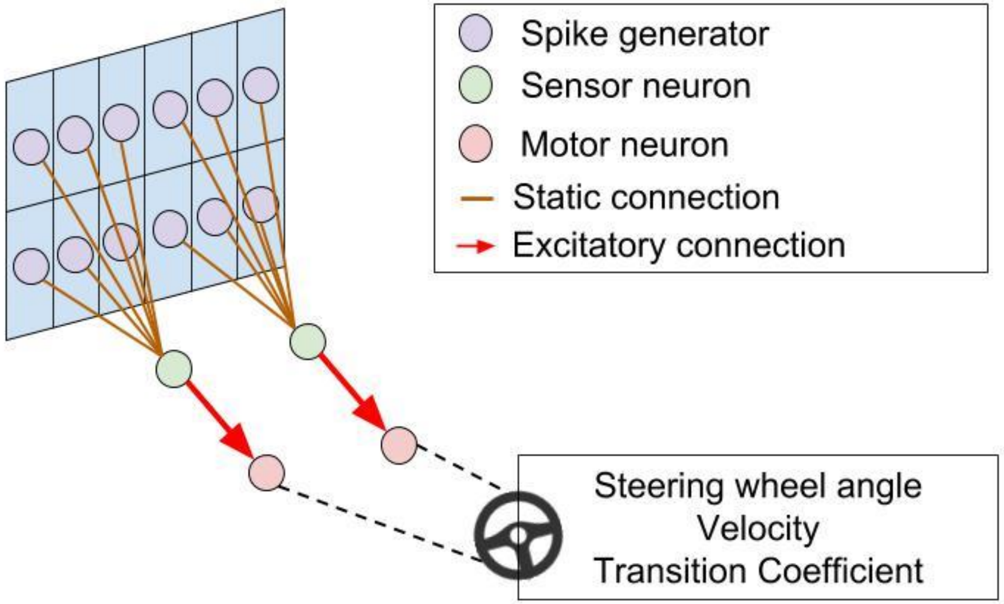
Building Braitenberg vehicles in simulation



„“Connecting artificial brains to robots in a comprehensive simulation framework: the neurorobotics platform”, Falotico et al., Frontiers in Neurorobotics 2016

Braitenberg on the road





„Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks", Kaiser, Tieck, et al., SIMPAR 2016

Where to go next?

M. Sc. Juan Camilo Vasquez Tieck (tieck@fzi.de)

M. Sc. Jacques Kaiser (jkaiser@fzi.de)

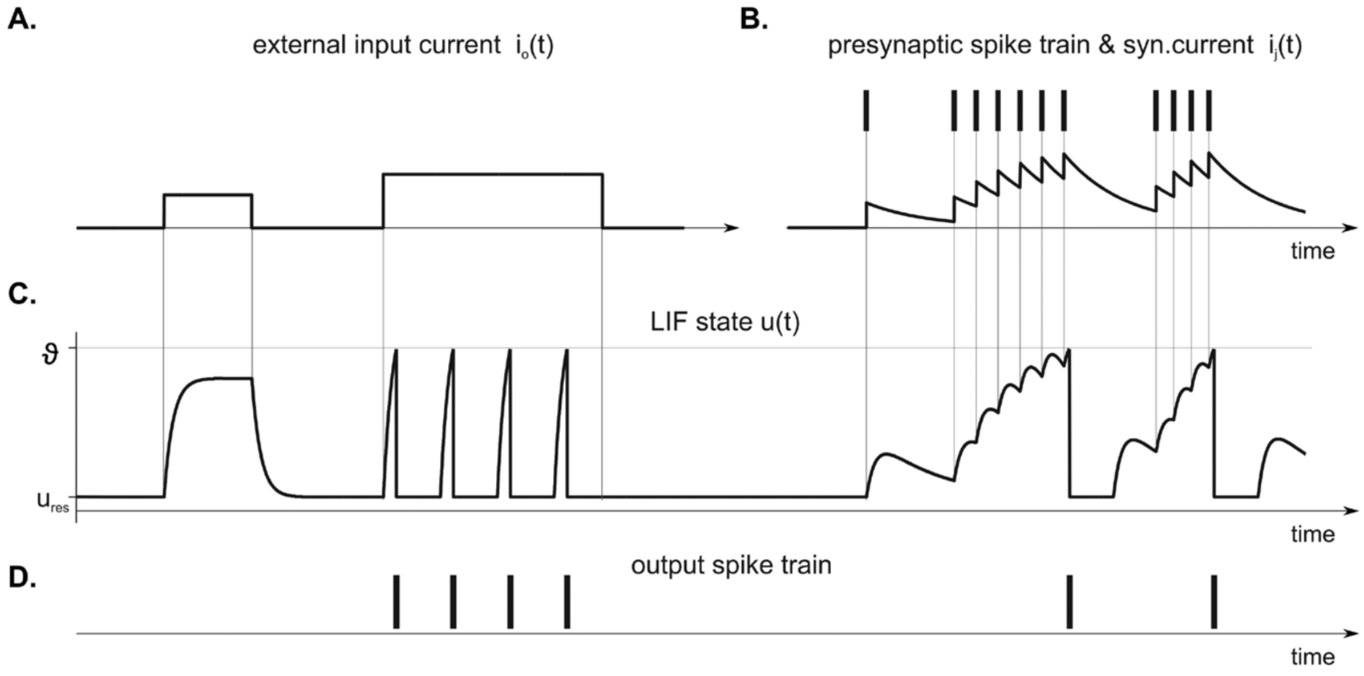
Danke!

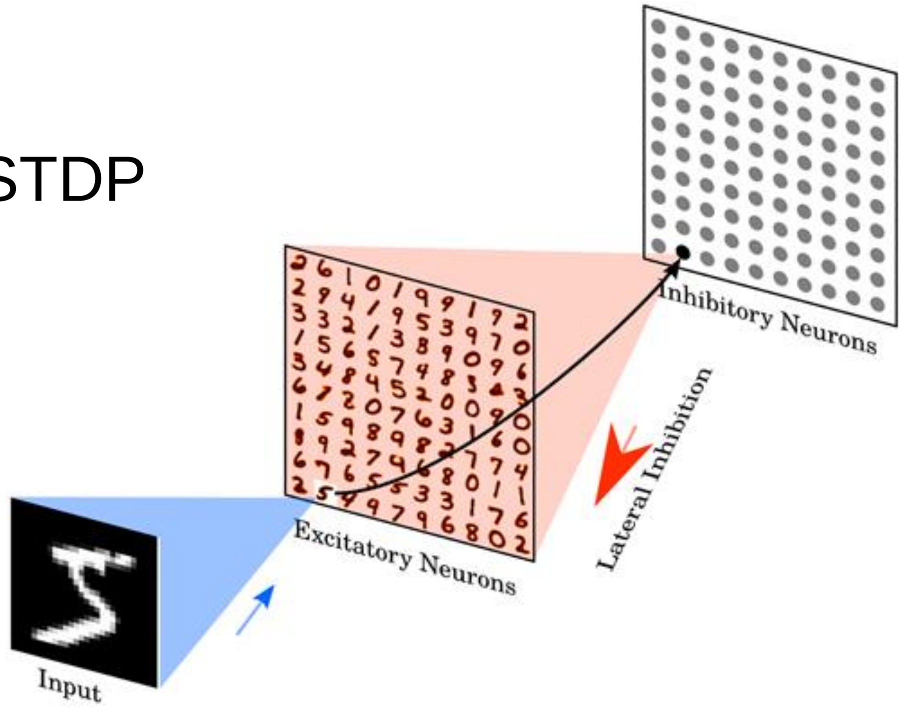
Fragen???

Please fill the feedback form

The pyNN / NEST exercise

- Learning MNIST with spiking networks using STDP





„Unsupervised learning of digit recognition using spike-timing-dependent plasticity", Peter U. Diehl and Matthew Cook, IEEE Transactions in Neural Networks and Learning Systems, 2014

- Correction on Wednesday 12th at 14:00

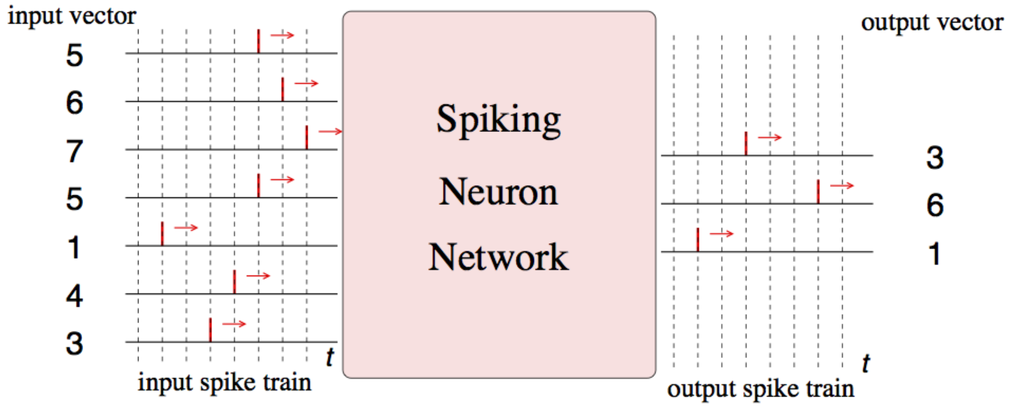
ipython notebook exercise

Auswertung neuronaler Aktivität

- Pulsraten Kodierung
  - Ähnlich zum Kontinuierlichen Ausgangssignal klassischer KNN
  - *Rate* und *Population rate*
    - Mittlere Pulsrate einzelner Neuronen oder Populationen
  - *Population Temporal*
  - Plausibel für langsam ändernde Signale

- Spike-timing Kodierung
  - Raten Kodierung zu langsam für schnelle Signalverabeitung
  - Temporale Kodierung
    - Je stärker ein Input, je früher ein Aktionspotential
    - Ermöglicht die Emulation klassischer KNN
  - Time-to-first-spike
  - Phasenkodierung
    - Kodierung in periodischen Signalen
  - Korrelation und Synchronität
  - …

- Übliche Darstellung von Spike Trains:
  Spike raster plot



Temporale Kodierung

Paugam-Moisy, Bohte. 2012. Computing with Spiking Neuron Networks.

Die Poissonverteilung

- Einfaches Modell „ohne Gedächtnis” (keine Refraktionszeit)
- $f(x,\lambda) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$

Lernen mit SNN

- Wie lernt man mit gepulsten neuronalen Netzen?

- Konzepte traditioneller KNN
  - können nicht angewandt werden
  - zu unterschiedliche Funktionsweise

- Neuronale Plastizität
  - Parameter der Neuronen sind fix
  - Anpassung der Synaptischen Gewichte
  - Hebbsches Lernen (1949)

    “When an axon of cell A is near enough to excite cell B or repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased”

    “What fires together wires together”

  - *Longterm/Shortterm Potentiation/Depression (LTP/STP/LTD/STD)*

Synaptic plasticity

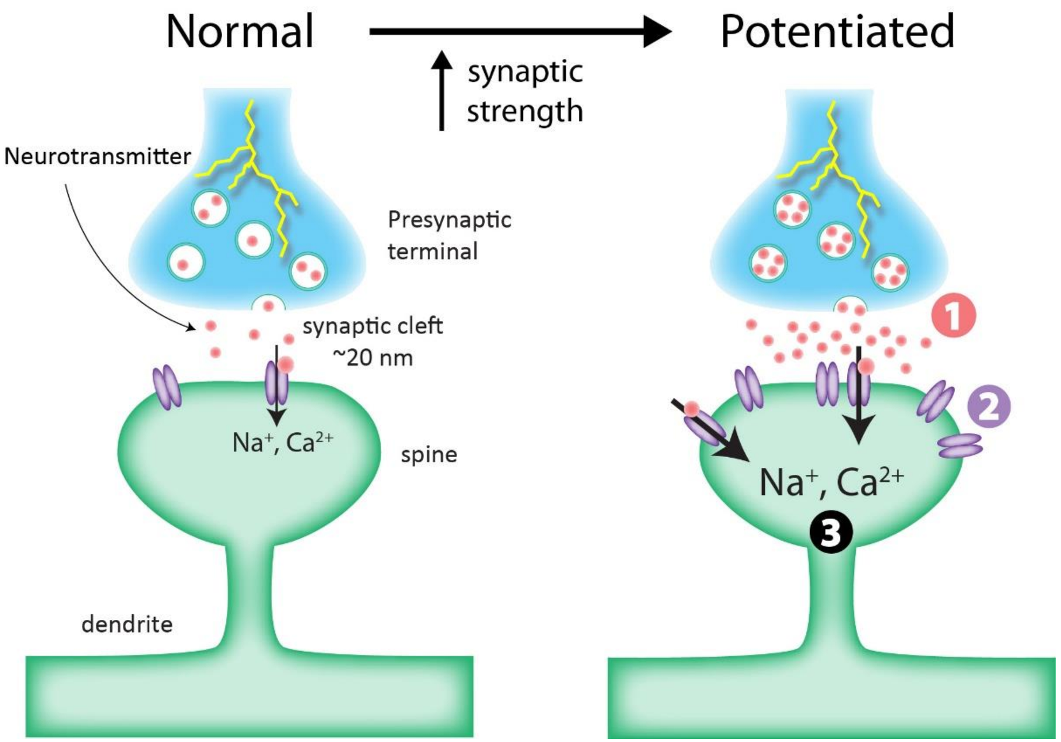


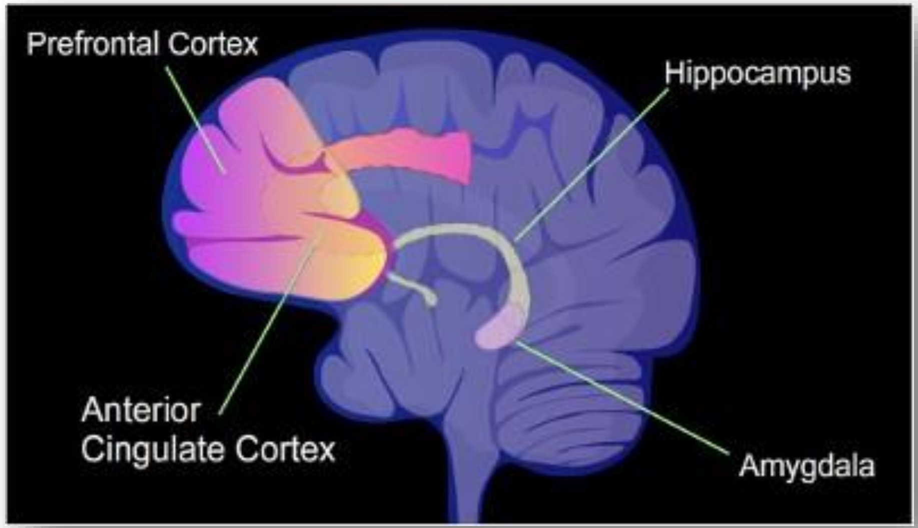
Image taken from: http://qbi.uq.edu.au/the-brain/physiology/long-term-synaptic-plasticity

- Strength of spike depends on number of:
  - Neurotransmitters in axonal ending of presynaptic neuron
  - Ion channels (receptors) in postsynaptic neuron's membrane
- Plasticity is change in any (or both) of these two
- Functionally this increases (potentiation) or decreases (depression) of spike efficacy

Short term vs. long term memory

- Long term storage -> synaptic connectivity
  - Takes long time to form (1h – 10h)
  - Hippocampus has a major role in memory formation
  - Requires creation of new proteins in synapses

- Short term storage -> current activity of population
  - Working memory performs fast manipulation on short term memory buffers

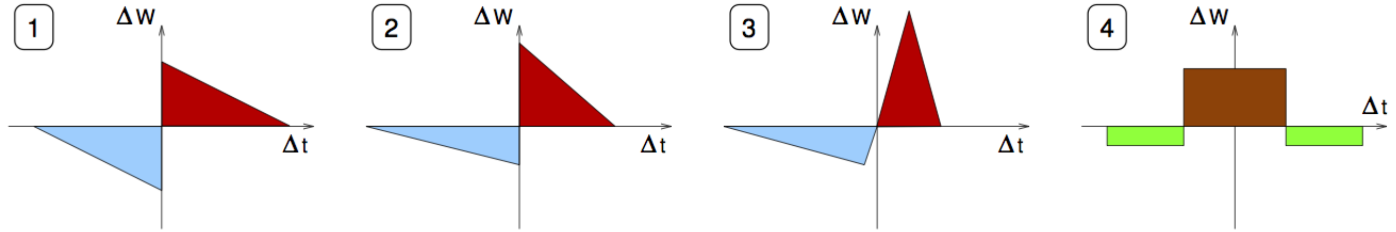


“Rescuing cocaine-induced prefrontal cortex hypoactivity prevents compulsive cocaine seeking” by Billy T. Chen, et. Al, *Nature,* April 3 2013, doi:10.1038/nature12024

Spiking Neural Networks: Lernen

- Spike Timing Dependent Plasticity (STDP)
  - Anpassung der synaptischen Gewichte je nachdem, wann der postsynaptischen Neuronen feuern.
- Rate Based Plasticity
  - Anpassung wird vor allem durch die Rate der prä- und postsynaptischen Feuerrate beeinflusst.
  - Problem: unkontrollierte Gewichtszunahme und unselektive rezeptiven Felder.
- Reward based plasticity
  - Plastizität wird durch Reinforcement Learning gesteuert.

Spike-timing dependent plasticity (STDP)



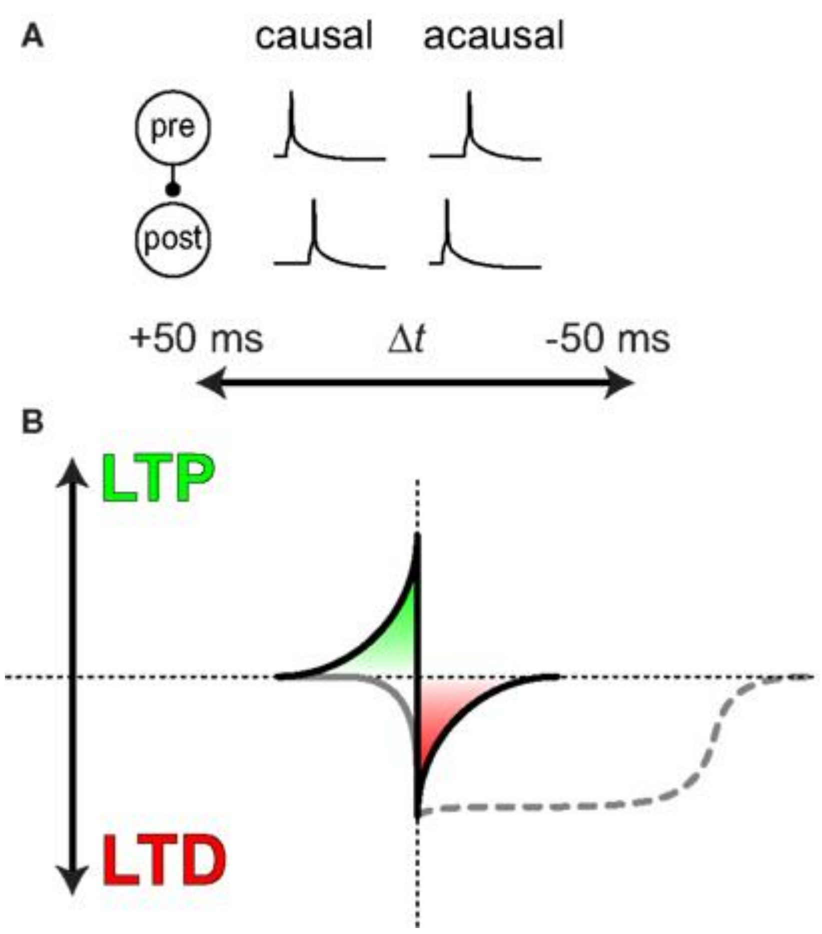
- *Spike-timing dependent plasticity (STDP)*
  - Temporales Hebbsches Lernen
  - Synaptische Plastizität empfindlich auf präzises Timing von Pulsen
    $\Delta t = t_{post} - t_{prä},\ w \leftarrow w + \Delta w$ oder $w \leftarrow w(1 + \Delta w)$
  - Funktionsweise für exzitatorische Synapsen (1-3)
    - Präsynaptischer Puls kurz vor dem Aktionspotential verstärkt
    - Präsynaptischer Puls kurz nach dem Aktionspotential schwächt
    - Grosse zeitliche Differenz hat keine Auswirkung
  - Für inhibitorische Synapsen
    - Standard Hebbsches Lernen (4)

Spike Timing Dependent Plasticity

- Neurons store causes of their activity in synaptic weight

$$STDP(t) = \begin{cases} A^{+}e^{-\Delta t/\tau^{+}} & \text{if } \Delta t \geq 0 \\ -A^{-}e^{\Delta t/\tau^{-}} & \text{if } \Delta t < 0 \end{cases}$$

- LTP – Long Term Potentiation (+)
- LTD – Long Term Depression (-)
- Hebbian rule:
  - “Neurons who fire together – wire together.”
  - Learning is local and incremental

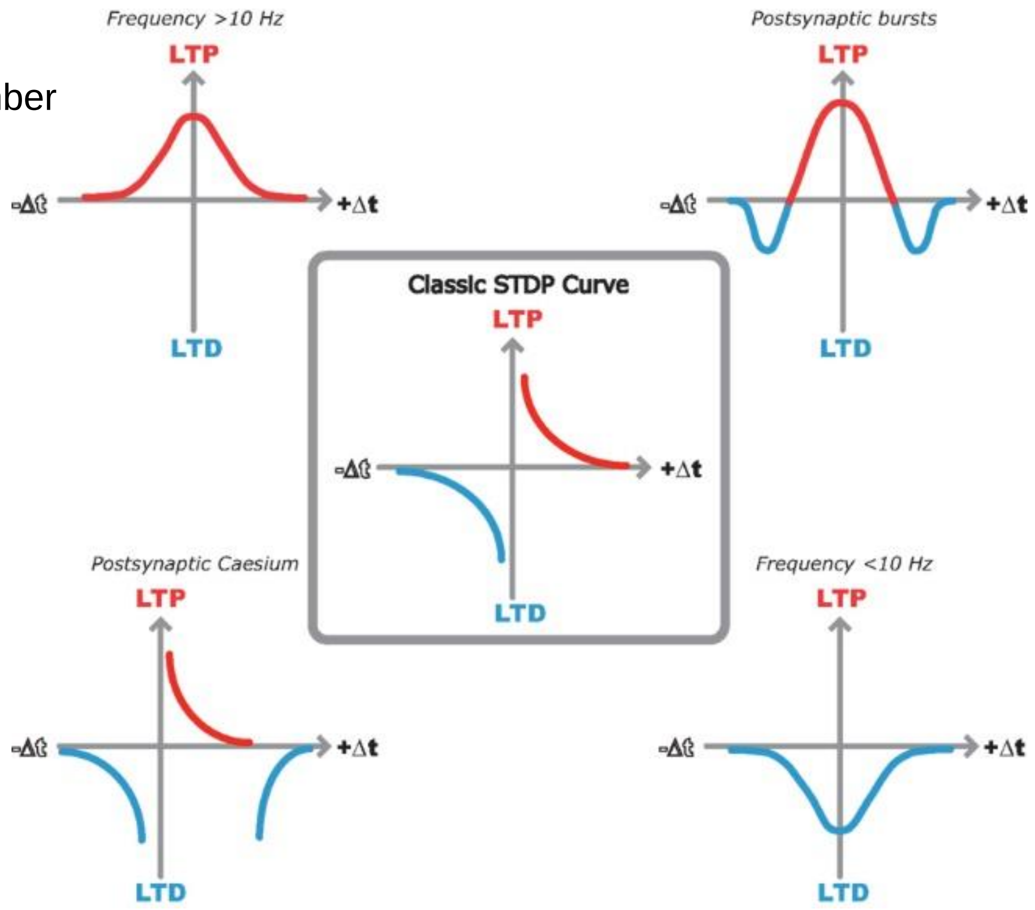


“A history of spike-timing-dependent plasticity”, Henry Markram et. al

Different variants of STDP

- Symmetric
- Strictly potentiating
- Results in high number of false positives

- Symmetric
- Coincidence detection
- Tunes well only to correlated input

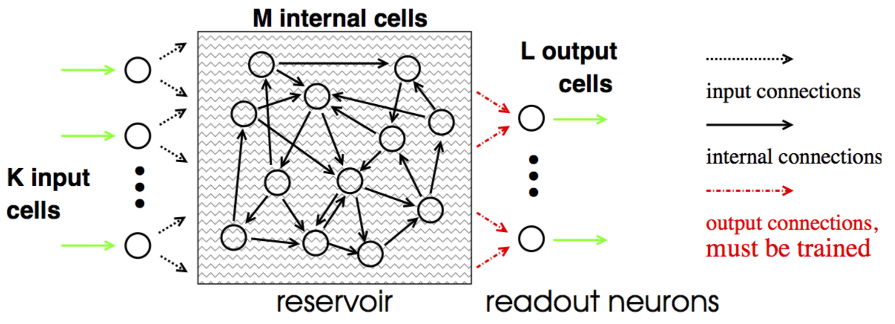


- Asymmetric
- Strictly causal coincidence detection

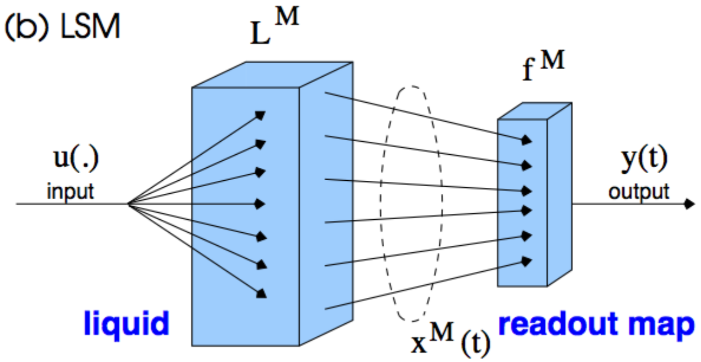
- Symmetric
- Anti-Hebbian learning

Liquid State Machines

- Reservoir computing
  - Biologische Netze sind meist unregelmäßig und dünn vernetzt
  - Überwachtes Lernen in rekurrenten Netzen schwierig
  - Struktur
    - Eine Eingabeschicht mit K Neuronen
    - Rekurrentes SNN mit M Neuronen (Reservoir) wird nicht trainiert
    - Ausgabeschicht mit L Neuronen. Nur diese wird trainiert
  - Ausnutzen der Mächtigkeit und Selbstorganisation zufällig vernetzter Rekurrenter Netzwerke
  - Für klassische und gepulste Neuronale Netze
  - SNN können mit STDP zusätzlich im Reservoir lernen



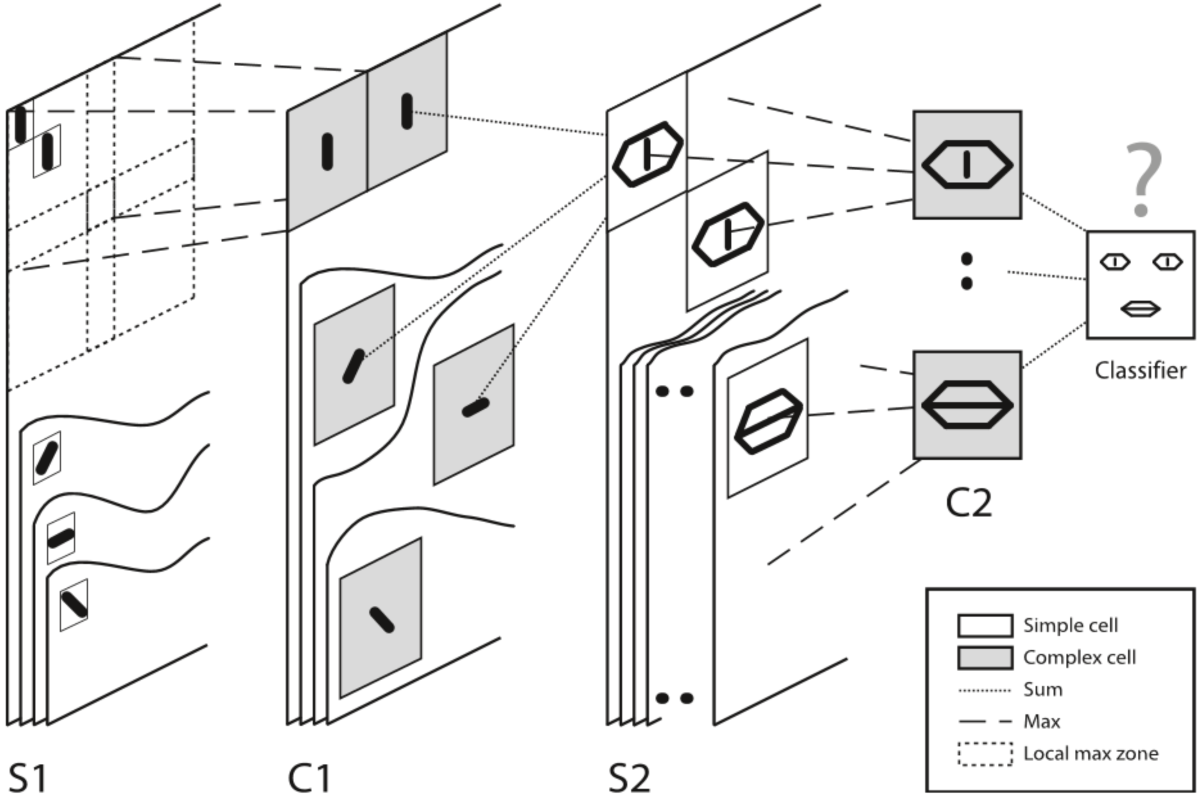
Lernen mit SNN



- Liquid State Machine (Maas et al. 2002)
  - Echtzeitverarbeitung kontinuierlicher Daten $u(t)$ mit rekurrenten LIF Netzen
  - Idee
    - Liquid Filter (Das Reservoir) verhält sich wie Wellen im Wasser (hochdimensional)
    - Angeregt durch Motoren (niedrigdimensional)
  - Ausgabeschicht
    - Einzelnes LIF Neuron oder Netz
    - Auffällige Informationen aus dem Liquid Filter werden erlernt
    - Training durch einfache Lernregeln (lineare Regression oder p-delta Regel)

Unsupervised learning with STDP

- All-to-all connections between inputs and excitatory neurons, at randomized weights
- When an excitatory neuron spikes, all other excitatory neurons are inhibited
- A neuron is assigned the label for the class it is most active



„Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity", Timothee Masquelier and Simon J. Thorpe, PLoS Computational Biology, 2007

Implementierungen

- Simulatoren
- NRP
- Nengo
- Neuron
- Brian
- PyNN
- Neuromorphic Hardware
  - SpiNNaker
  - heidelberg
- $\sum_i$

Population Temporal Kodierung

- Basierend auf dem Neuro-Engineering Framework (NEF), (Eliasmith & Anderson 2003)
- Erlaubt das interpretieren und „programmieren“ eines SNN
- Drei Prinzipien
  - Nichtlineare Kodierung / Lineare Dekodierung von Variablen
  - Transformationen bestimmt alternative lineare Dekodierung bestimmt
  - Neuronale Dynamik dargestellt durch neuronal kodierte Zustandsvariablen

- Nichtlineare Kodierung:
  - Aktivität einer Population $a_i(\boldsymbol{x})$ für einen Vektor $\boldsymbol{x}$

$$a_i(\boldsymbol{x}) = G_i\big(J_i(\boldsymbol{x})\big)$$

  - $G(\cdot)$ beschreibt die Aktivität eines Neurons
  - Eingangsstrom $J_i(\cdot)$ mit Kodierungsvektor $\boldsymbol{e}_i$, Verstärkungsfaktor α_i und Hintergrundaktivität J^{bias}

$$J_i(\boldsymbol{x}) = \alpha_i\langle \boldsymbol{x}, \boldsymbol{e_i} \rangle + J_i^{bias}$$

  - Im Fall des LIF gilt

$$a_i(\boldsymbol{x}) = G_i\big(\alpha_i\langle \boldsymbol{x}, \boldsymbol{e_i} \rangle + J_i^{bias}\big) = \delta(t - t_m)$$

Population Temporal Kodierung

- Lineare Dekodierung:
  - Simples Modell des Postsynaptischen Stroms (PSC): $h(t) = e^{-t/\tau_{PSC}}$ (Verfallszeit τ_{PSC} vom Neurotransmitter abhängig)
  - Aktivität des Neurons (Faltung des PSC mit der Pulsfolge):

$$a_i(\boldsymbol{x}) = \sum_n h_i(t) * \delta_i(t - t_n) = \sum_m h(t - t_m)$$

  - Lineares Optimierungsproblem (mit Rauschen)

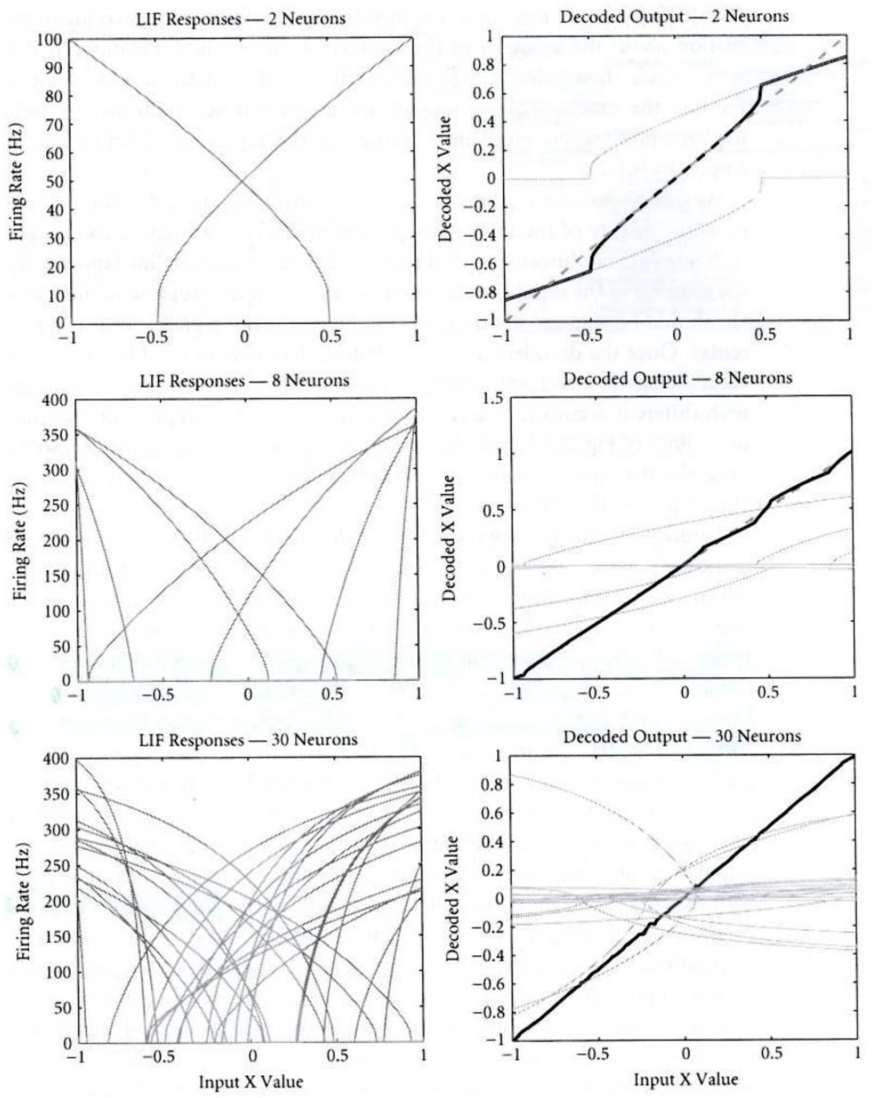
$$E = \int_R \left(\boldsymbol{x} - \sum_i^N (a_i(\boldsymbol{x}) + \eta_i)\boldsymbol{d}_i \right)^2 d\boldsymbol{x} d\eta$$

  - Daraus ergibt sich die lineare Dekodierung:

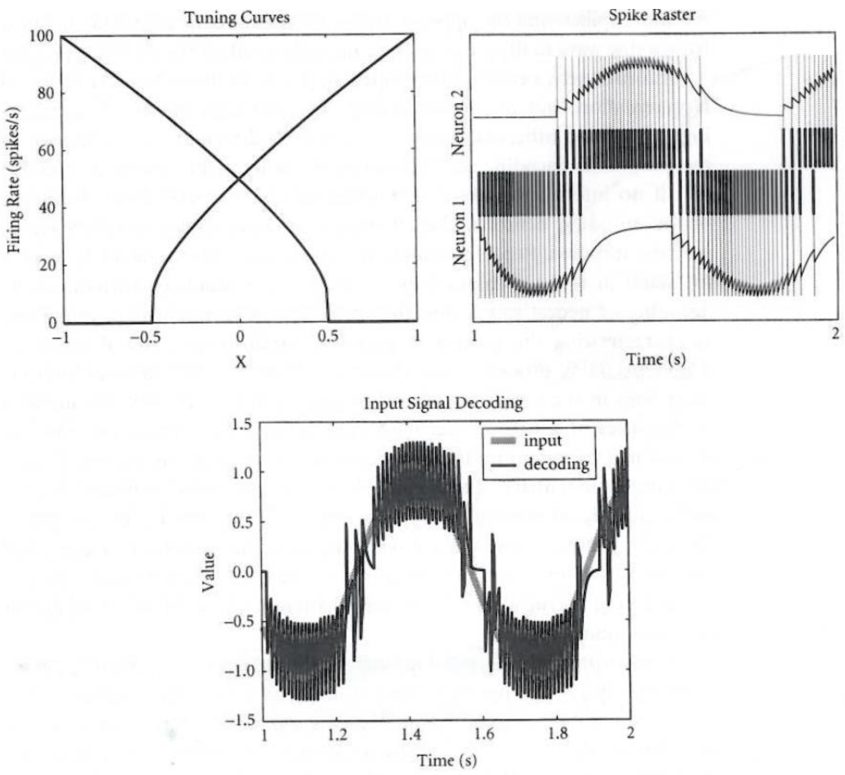
$$\hat{\boldsymbol{x}} = \sum_{i,m}^{N,M} h_i(t - t_m) \cdot \boldsymbol{d}_i$$

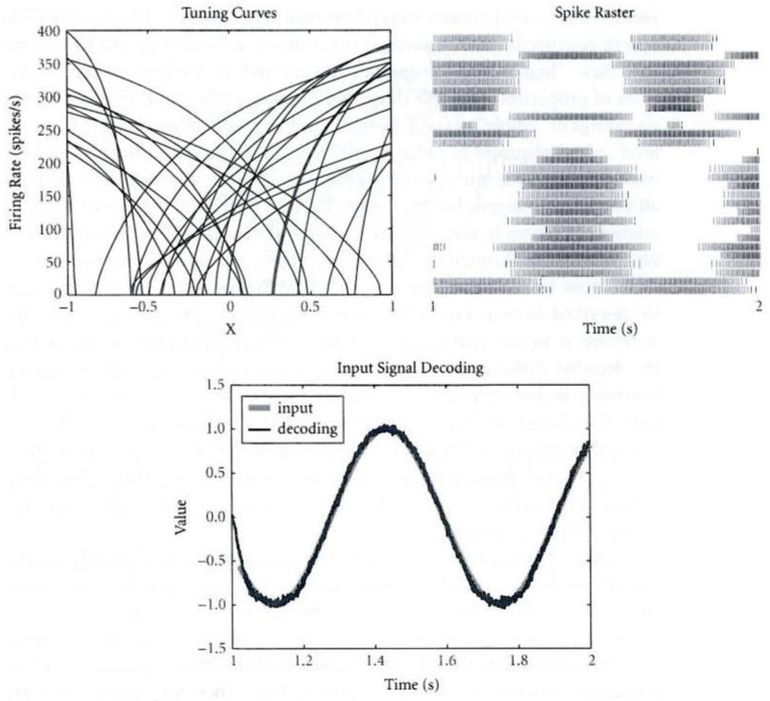
- Dekodierung von Transformationen analog

Population Temporal Kodierung

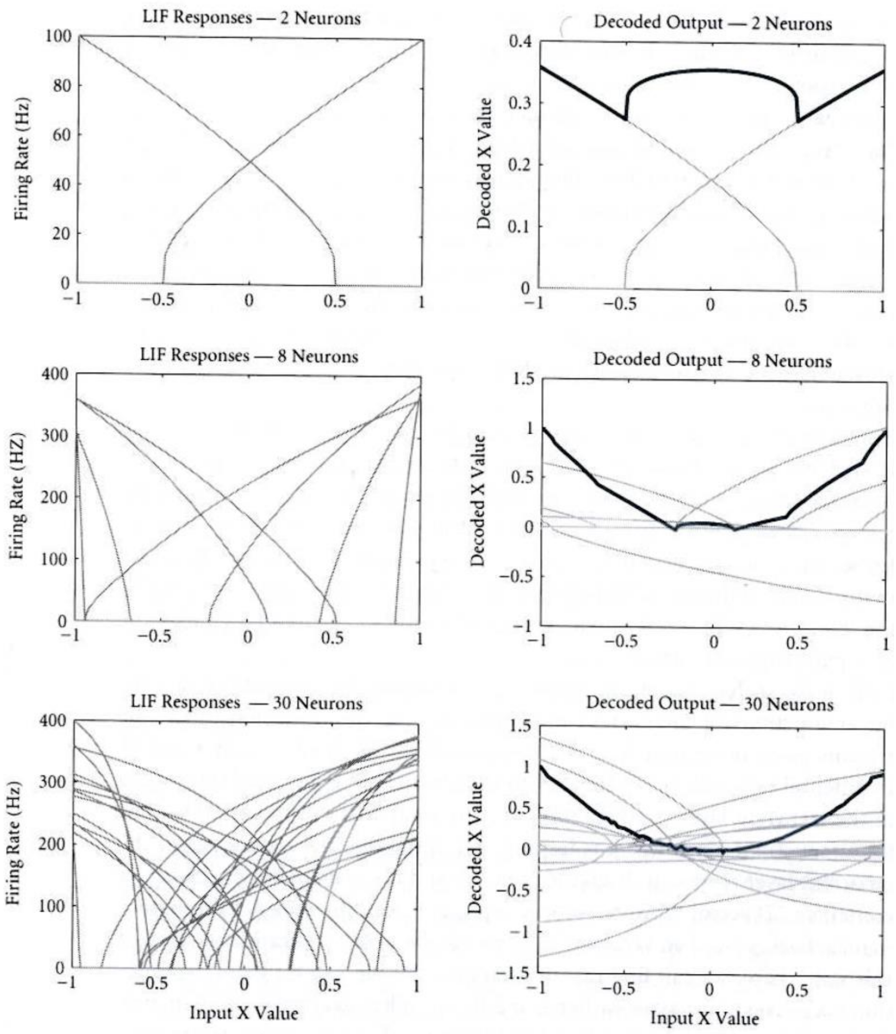


Population Temporal Kodierung





Population Temporal Kodierung

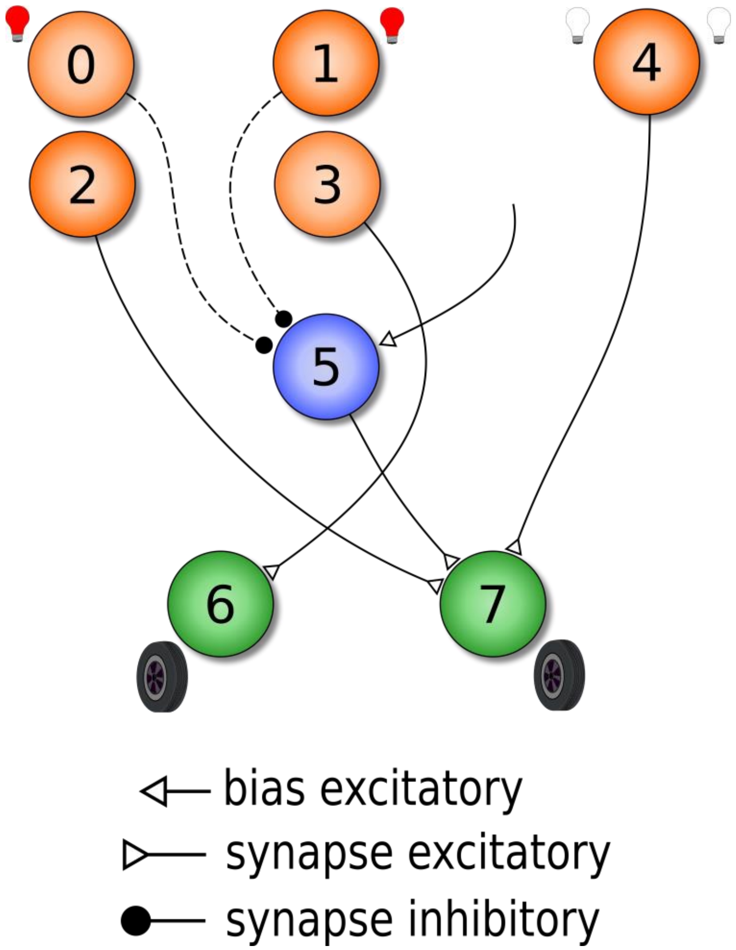


Übersicht

- Motivation
  - Das menschliche Gehirn
  - Das Human Brain Projekt
- Neuronenmodelle
  - Geschichte der Künstlichen Neuronalen Netze
  - Natürliche Neuronen
  - Gepulste Neuronenmodelle
- Neuronale Kodierung
  - Population temporal Kodierung
  - Mächtigkeit gepulster Neuronaler Netze
- Lernen mit gepulsten Neuronalen Netzen
  - Synaptic time-dependent Plasticity
  - Reservoir computing
- Beispiele
  - Braitenberg vehicle
  - Beinsteuerung

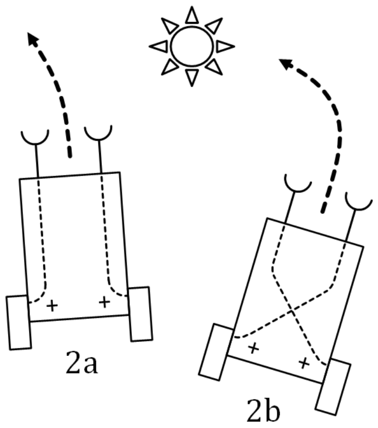
Beispiel: Braitenberg Vehicles

- Gepulstes Neuronales Netz (8 Neuronen)
  - Eingangsneuronen
    - Werden als *Poissongeneratoren* modelliert
    - Intensität des Stimulus bestimmt die Rate
    - 0,2 verbunden mit linkem Sensor
    - 1,3 verbunden mit rechtem Sensor
    - 4 aktiv ohne stimuli
  - Ausgangsneuronen
    - Werden als *Leaky Integrators* modelliert
    - Membranspannung als Steuersignale
    - 6 linker Aktuator, 7 rechter Aktuator
- Einfaches Verhalten
  - Ohne Stimulus
    - Drehen gegen Uhrzeigersinn
  - Mit Stimulus
    - Drehung und Bewegung zum Stimulus



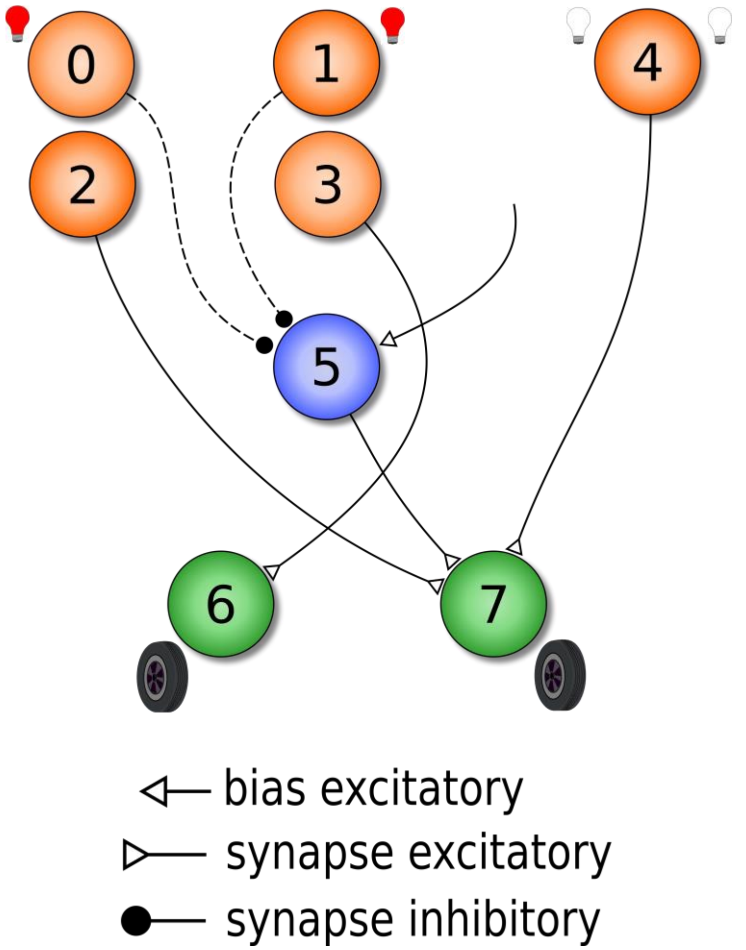
Beispiel: Braitenberg Vehicles

- Einfaches Kognitives Experiment
- Nach Braitenberg 1984
- Mobiler Agent
  - Zwei Seitliche Sensoren
  - Zwei Aktuatoren
- Durch einfaches (neuronales) Verschalten kann vielfältiges Verhalten erzeugt werden

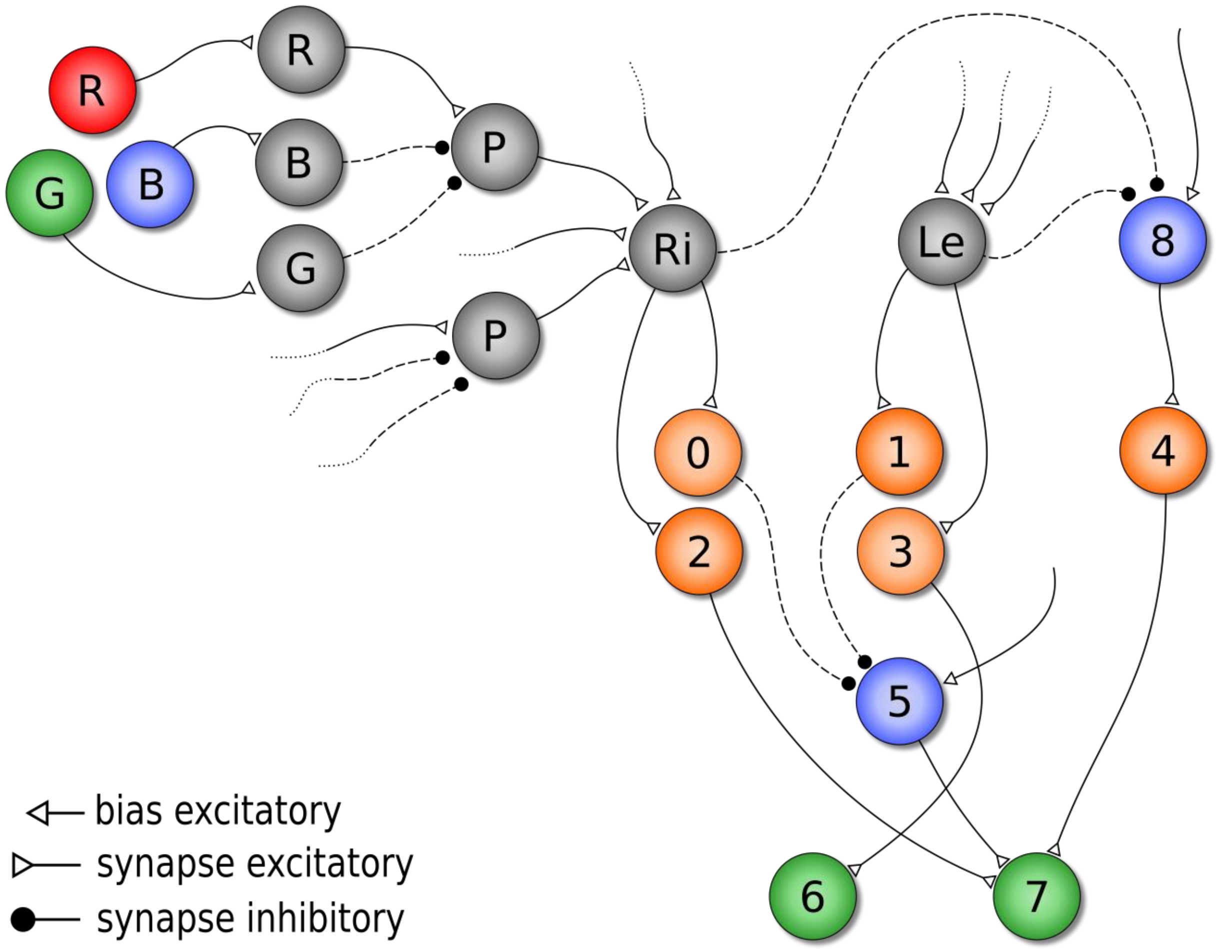


Beispiel: Braitenberg Vehicles

- Gepulstes Neuronales Netz (8 Neuronen)
  - Eingangsneuronen
    - Werden als *Poissongeneratoren* modelliert
    - Intensität des Stimulus bestimmt die Rate
    - 0,2 verbunden mit linkem Sensor
    - 1,3 verbunden mit rechtem Sensor
    - 4 aktiv ohne stimuli
  - Ausgangsneuronen
    - Werden als *Leaky Integrators* modelliert
    - Membranspannung als Steuersignale
    - 6 linker Aktuator, 7 rechter Aktuator
- Einfaches Verhalten
  - Ohne Stimulus
    - Drehen gegen Uhrzeigersinn
  - Mit Stimulus
    - Drehung und Bewegung zum Stimulus



Beispiel: Braitenberg Vehicles



- Erweiterung zur Auswertung von Kamerabildern

Beispiel: Braitenberg Vehicles