

Introduction to Neuroinformatics

Summary of the lectures 2013

Version 2.0

Initial Version 2012

Benjamin Ellenberger

Revision 2013

Joachim Ott

Dora Sumisławska

Contents

1 Lecture 1 - Neuroinformatics (Michael Pfeiffer)	1
1.1 Introduction	1
2 Lecture 2 - Nervous System Organization (Kevan Martin)	4
2.1 Building elements of the brain	4
2.2 Nervous System in numbers	5
3 Lecture 3 - Membrane Potential (Kevan Martin)	6
3.1 Membrane Potential - Structure and Function	6
3.2 Concentrations across the membrane	8
3.3 Nernst equation	8
3.4 Goldman-Hodgkin-Katz equation (GHK-Voltage equation) . .	9
4 Lecture 4 - Passive (Cable) Membrane Properties (Valerio Mante)	10
4.1 Summary of membrane properties:	10
4.2 Electrical nature of the membrane	10
4.3 The cable equation	11
4.4 The Hodgkin-Huxley Model	12
5 Lecture 5 - Action Potential (Valerio Mante)	13
6 Lecture 6 - Synapse 1 (Kevan Martin)	16
7 Lecture 7 - Synapse 2 (Valerio Mante)	17
7.1 Synaptic mechanism	17
7.2 Categorizing synapses according to:	17
7.2.1 way of activation – complexity	17
7.2.2 carrier used to transmit information	19
7.2.3 function	19
7.3 Glutamate receptors	20
7.4 Neuromodulators **	21
8 Lecture 8 - Plasticity/Learning (Michael Pfeiffer)	22
8.1 Learning & Memory	22
8.2 Plasticity	22
8.3 Models of plasticity	24
8.4 Pavlovian Learning	24

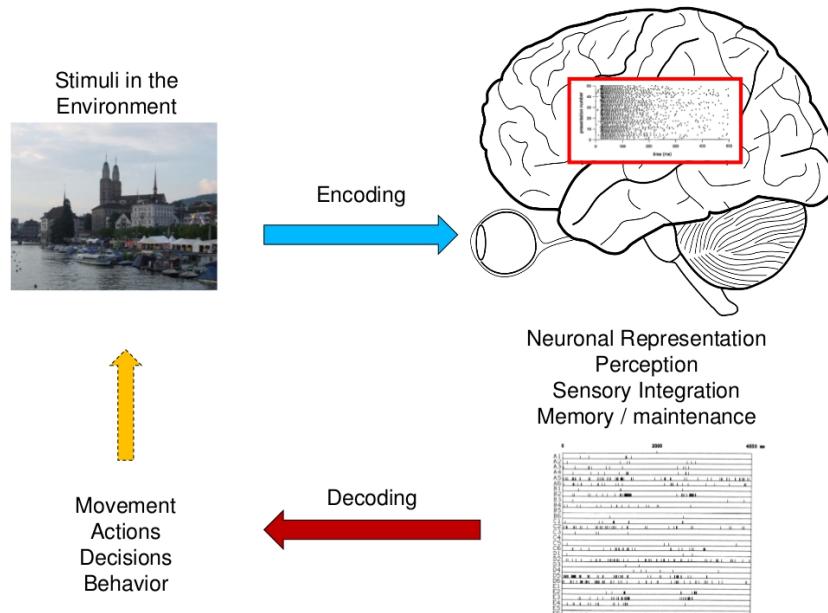
8.5	Hebbian Learning	24
8.6	NMDA synapse	25
8.7	Short term plasticity(STP)	25
8.8	Long term plasticity(LTP)	26
8.9	Spike-timing dependent plasticity (STDP)	27
8.10	Facts	28
8.11	Dopamine	28
9	Lecture 9 - Rate / Event Coding (Michael Pfeiffer)	29
9.1	Neuronal rate codes(average over time(single neuron))	30
9.2	Poisson Spike Trains	31
9.3	What can a single neuron encode?	31
9.4	Population rate	31
9.4.1	Population codes	31
9.4.2	Neuronal Event codes	32
9.5	Binding problem	34
9.6	Averages & Estimation	34
9.6.1	Spike Triggered Average	34
9.6.2	Issues to remember:	34
10	Lecture 10 - Neuromorphic VLSI(Giacomo Indiveri)	36
10.1	VLSI	36
10.2	Why VLSI for neural computation?	36
10.3	MOSFET Characteristics	36
10.4	Different remarkable circuits	37
11	Lecture 11 - Perceptron Learning Algorithm (Matthew Cook)	39
11.1	Logic Gates	39
11.2	Why XOR is Impossible with Perceptrons	42
11.3	Comparison Perceptrons vs. Real Neurons	43
11.4	Perceptron Learning Algorithm	43
12	Lecture 12 - Hopfield Networks(Matthew Cook)	46
12.1	Updates and State Dynamics	47
13	Lecture 13 - Feed-Forward Networks (Matthew Cook)	49
13.1	Error Function	49
13.2	Improve Error	50
13.3	Back Propagation Algorithm	51

14 Lecture 14 - Interacting Neural Populations(Matthew Cook)	53
14.1 Population codes	53

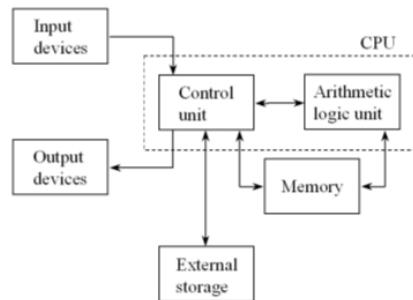
1 Lecture 1 - Neuroinformatics (Michael Pfeiffer)

1.1 Introduction

- Why do we have the brain?



- Alan Turing
John von Neumann



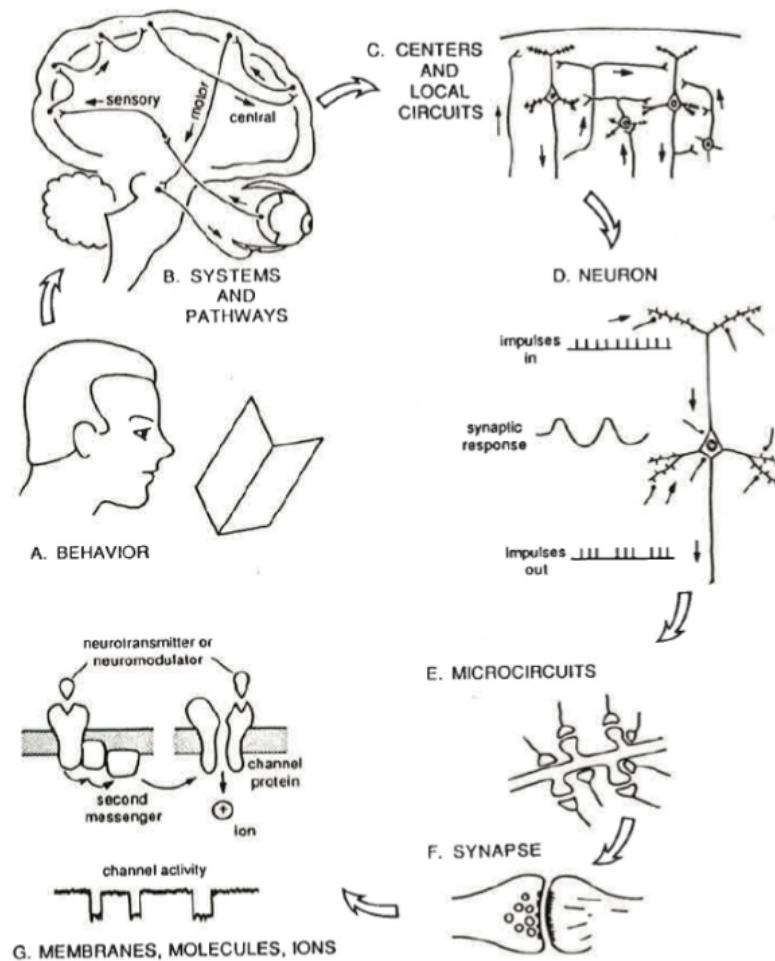
- How is a brain different/ similar to a computer?

Similar:

Process information, Logical operations, Memory, Use electrical (digital) signaling, Can learn from inputs, Consume energy, ...

Different:

Massive parallelism, Constantly adapting, Chemical signaling, Unreliable units, Analog computation, Robust to damage, Very energy efficient, ...



- What this course will be about?

Information processing in the brain: neurons, synapses, nervous system organization; Analytical descriptions of neural computations; Learning and plasticity; Encoding information in the brain; Theoretical neural network models; Engineering brain-like computers.

MIPS = million instructions per second

Tianhe world's fastest computer, match only 1% of your brain (≈ 85 Billion Nerve Cells)

Windows XP $\approx 1.5GB$ of code, human genome ($\approx 750MB$).

- The era of "Big Brain Projects" in ex. Human Brain Project.

Neurogrid – real-time emulation on 1 Mio. neurons

2 LECTURE 2 - NERVOUS SYSTEM ORGANIZATION (KEVAN MARTIN)

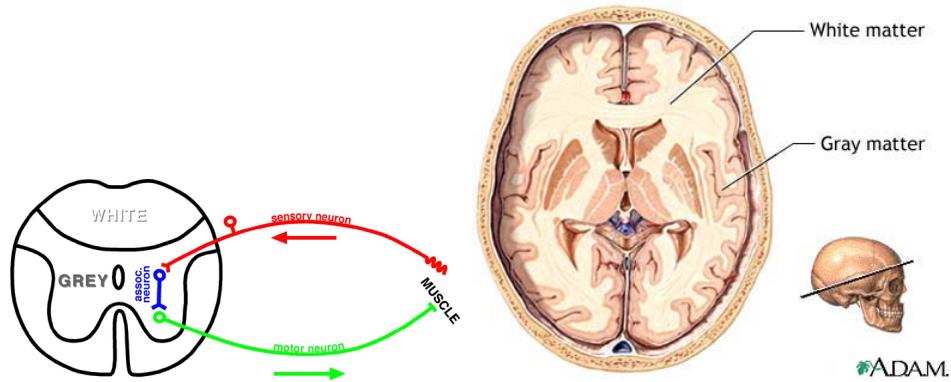
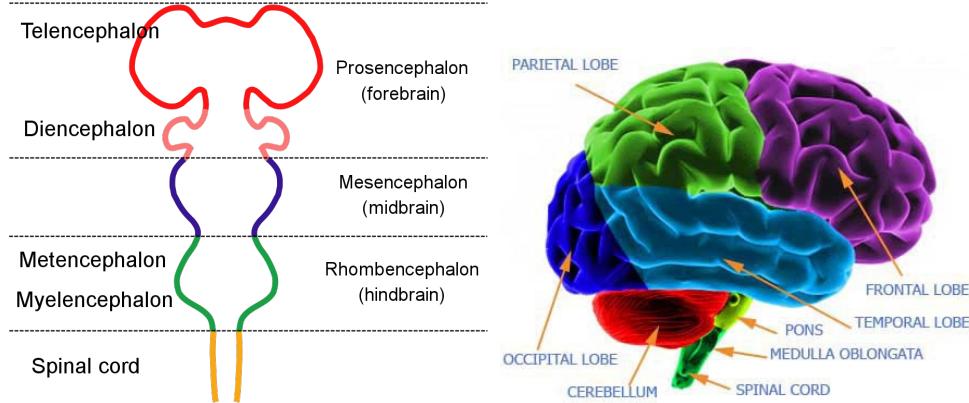
2 Lecture 2 - Nervous System Organization (Kevan Martin)

2.1 Building elements of the brain

Brain consists of:

- Forebrain (Prosencephalon)
 - Cortex
 - Thalamus ('couch')
 - Hippocampus
 - Basal Ganglia
 - Corpus Callosum
- Midbrain (Mesencephalon)
 - Tectum
 - Tegmentum
- Hindbrain (Rhombencephalon)
 - Cerebellum (Computationally mysterious)
 - Pons
 - Medulla Oblongata
- Whitematter
 - Glia cells, myelinated axons
- Greymatter
 - Neurons (Soma)

2 LECTURE 2 - NERVOUS SYSTEM ORGANIZATION (KEVAN MARTIN)



2.2 Nervous System in numbers

White matter $1mm^3 \rightarrow 9m$ of wires (axons).

Grey matter $1mm^3 \rightarrow 50'000$ neurons, $4km$ of wires. \rightarrow Long distance communication needs space.

Only 1m goes to spinal cord.

100k cells in $1mm^3$.

Auditory vs visual pathway from sensor 1.2 m to 15km.

3 Lecture 3 - Membrane Potential (Kevan Martin)

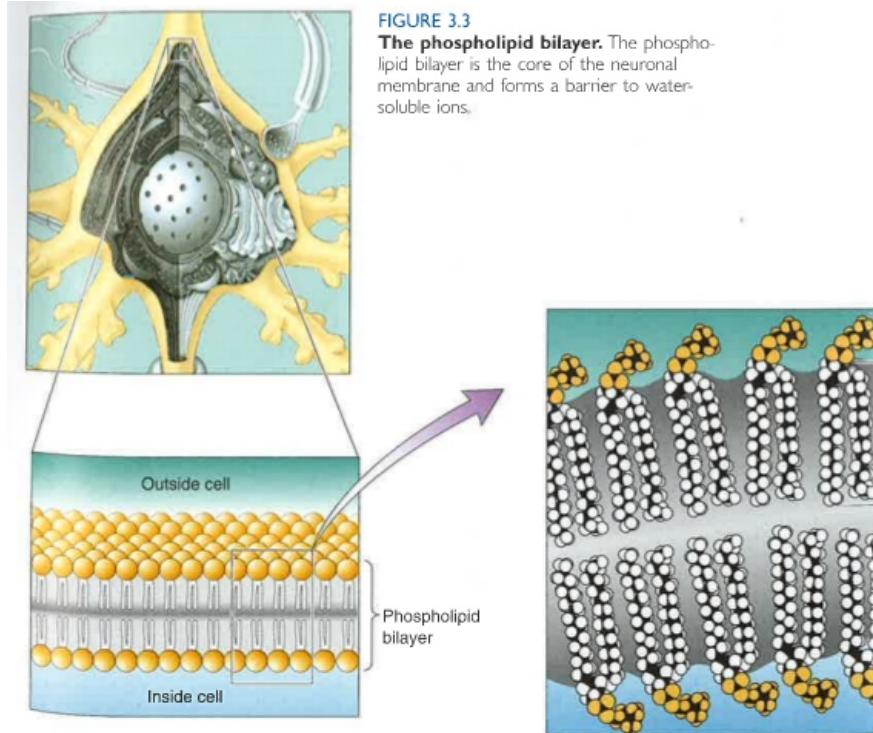
3.1 Membrane Potential - Structure and Function

The membrane bilayer creating the energy barrier (ion can not just flow through it without channels or pumps) between:

- ECS (Extra cellular solution)
- ICS (Intra cellular solution)

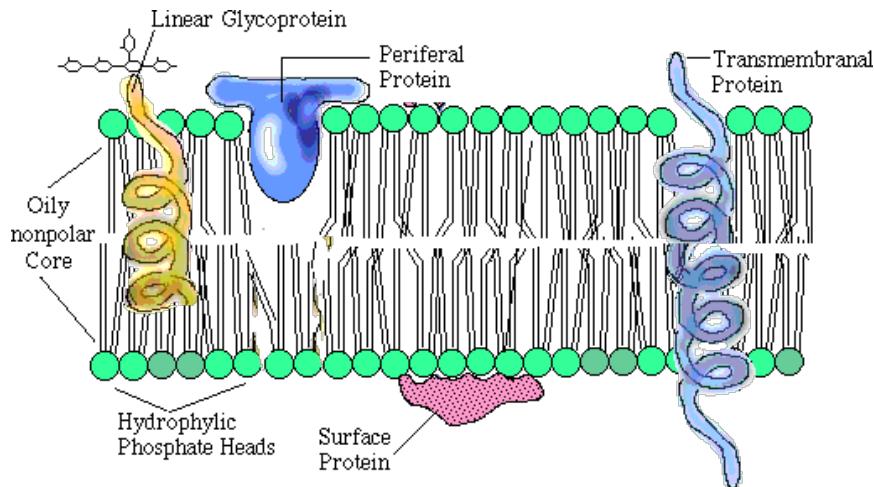
is built out of two types of molecules:

- the charged hydrophilic dipole head-group (hydrophile – funs of water – water molecules are also polarised),
- uncharged, hydrophobic (with phobia of water) hydrocarbon tails turning inward, avoiding contact with water.

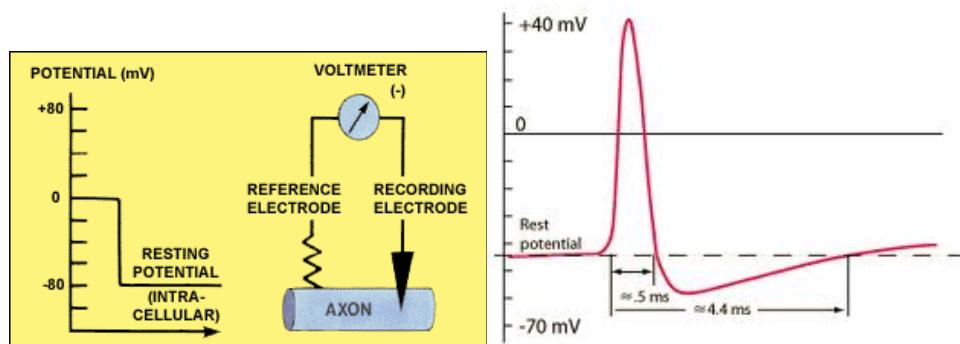


3 LECTURE 3 - MEMBRANE POTENTIAL (KEVAN MARTIN)

Transmembranal proteines play an important role as they have charged ends(phosphate heads) and uncharged bodies and therefore fit inside of the membrane as it aligns with the uncharged inner part of the membrane and with the outer charged parts.



The spike is the action potential caused by sodium (Na^{2+}) and changes in the membrane conductance. This shows that there is a voltage source in the membrane.



There are 2 basic mechanisms for the membrane potential leading to equilibrium (resting potential):

- Active mechanisms (Na^{+}/K^{+} -Pump from ECF to ICF).

3 LECTURE 3 - MEMBRANE POTENTIAL (KEVAN MARTIN)

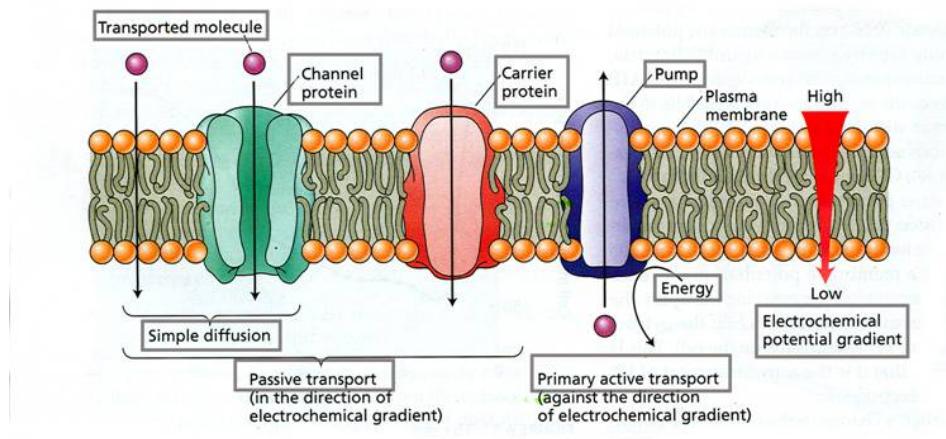
- Electrical force – in order to keep the difference in ion concentrations in ECF, ICF (asymmetry across membrane) there is a negative potential with reference to extracellular potential. Because in the rest there is only potassium ion flows we concentrate on them. Negative potential in ICF attracts positive charged K^+ ions into the cell while due to diffusion they are likely to leave the cell – inside greater concentration. In this way both forces are balanced.

3.2 Concentrations across the membrane

ION	ICF	ECF
Na^+ (Sodium)	-	+
K^+ (Potassium)	+	-
Cl^- (Chlorine)	-	+
Ca^{2+} (Calcium)	x	(+)
Large Anions(not passing membrane)	+	(-)

There is a concentration gradient top-down from high concentration to low concentration. The pump uses ATP to build up the gradient if it is released.

There are different structures enabling proteins to cross the membrane.



3.3 Nernst equation

$$E_{ion} = \frac{RT}{zF} \cdot \ln\left(\frac{[Ion]_{extracellular}}{[Ion]_{intracellular}}\right) \quad (1)$$

- R Universal gas constant ($8.3144 \frac{J}{mol \cdot K}$)

3 LECTURE 3 - MEMBRANE POTENTIAL (KEVAN MARTIN)

- F Faraday constant ($96500 \frac{C}{mol}$)
- z number of electrons involved in the reaction
- one mole = 6.02×10^{23} , solution is one molar when its concentration is $1 \frac{mole}{liter}$

At $37^\circ C$:

$$E_{ion} \approx 60mV \cdot \ln\left(\frac{[Ion]_{extracellular}}{[Ion]_{intracellular}}\right) \quad (2)$$

3.4 Goldman-Hodgkin-Katz equation (GHK-Voltage equation)

$$\Delta V = \frac{RT}{zF} \ln\left(\frac{P_K[K^+]_{out} + P_{Na}[Na^+]_{out} + P_{Cl}[Cl^-]_{in}}{P_K[K^+]_{in} + P_{Na}[Na^+]_{in} + P_{Cl}[Cl^-]_{out}}\right) \quad (3)$$

This is only for static situations but gives us a sense of how it is actually working. Basic assumptions:

- Ion flux obeys Nernst/Planck equation
- Ions move across membrane independently (no interaction)
- Electric field in the membrane is constant $E = -\frac{\delta V}{\delta x} = -\frac{\Delta V}{l}$

4 Lecture 4 - Passive (Cable) Membrane Properties (Valerio Mante)

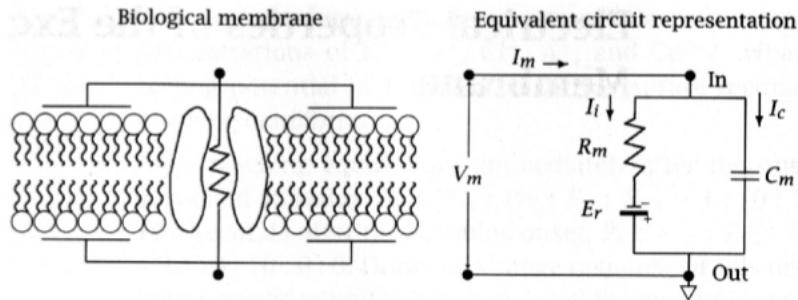
4.1 Summary of membrane properties:

- J_{diff} and J_{drift} are in an equilibrium (V_m = resting potential),
- Resting potential is due to K^+ asymmetric concentration,
- GHK-equation enables to calculate ΔV = Difference in Potential across the membrane,
- if for all ions crossing the membrane $P(Permeability) = 1 \rightarrow Nernst-equation$,
- In great approximation for passive membrane (only potassium and a little bit of sodium channel can cross the membrane) the resting potential is The function of conduction is approximately a weighted mean of equilibrium potentials for ions that can cross it. Membrane is not only permeable to potassium ions. There is also steady leak of sodium ions.

$$E_m = \frac{g_K E_K + g_{Na} E_{Na}}{g_K + g_{Na}} \approx -65mV \quad (4)$$

4.2 Electrical nature of the membrane

Ohmic model



$$C_m \frac{dV_m}{dt} + \frac{V_m - E_r}{R_m} = I_m$$

- The conductance of the membrane indicates its temporally dynamic character.

$$g_m = \frac{\text{Current}}{\text{Voltage}} = \frac{1}{\text{Resistance}} = \frac{I}{U} = \frac{1}{R_m} \quad (5)$$

- $I_c + I_o = I_m$
- If $\frac{\delta V_m}{\delta t} = 0 \rightarrow$ membrane is at rest.
- Changing the permeability of the membrane results in changeable conductance $g(t)$.
- Temporal summation of the membrane potential.

$$V(t) = V(\infty) + (V_{(0)} - V_{(\infty)})e^{-\frac{t}{RC}} \quad (6)$$

4.3 The cable equation

For a passive membrane, the membrane potential $V(x, t)$ is determined by solving the following partial differential equation:

$$\tau_m \left(\frac{\delta v}{\delta t} \right) = \lambda^2 \left(\frac{\delta^2 v}{\delta x^2} \right) - v + r_m i_e \quad (7)$$

where:

$\tau_m = (r_m c_m)$ sets the scale for the temporal variation in the membrane potential

a = radius of the axon ($= 2\mu m$)

$v = V - V_{rest}$

r_m = specific membrane resistance ($= 1 M\Omega \cdot mm^2$)

r_L = longitudinal resistance ($= 1 k\Omega \cdot mm$)

i_e = the current injected into a cell

$\lambda = \sqrt{\left(\frac{a r_m}{2 r_L}\right)}$ sets the scale for the spatial variation in the membrane potential (λ is called the *electronic length*)

$\lambda = 0.6 mm$ means no signal left after $0.6 mm$

\rightarrow Increasing R_m of the cable increases λ

\rightarrow Increasing diameter of the cable increases λ

For the steady-state solution ($\frac{\delta v}{\delta t} = 0$) of the equation then is:

$$v(x) = \left(\frac{i_e R_\lambda}{2} \right) e^{-|x|/\lambda}, \quad \text{where } R_\lambda = \frac{r_L \lambda}{\pi a^2} \quad (8)$$

4.4 The Hodgkin-Huxley Model

This is a model that describes how action potentials in neurons are initiated and propagated. It is a set of differential equations that approximates the electrical characteristics of excitable cells. The Hodgkin-Huxley model for generation of an action potential:

$$I_{ionic} = g_L(V_m - E_L) + \overline{g_K} \cdot n^4 \cdot (V_m - E_K) + \overline{g_{Na}} \cdot m^3 h \cdot (V_m - E_{Na}) \quad (9)$$

the membrane potential is:

$$I_{ion} = g_{ion}(V_m - E_{ion}) \quad (10)$$

for several ions, the conductance g_{ion} is not constant, and can thus be written in terms of a maximal conductance $\overline{g_{ion}}$ multiplied by a gate variable:

$$g_{ion} = \overline{g_{ion}} \cdot n_{ion} \quad (11)$$

where:

n = the probability of a single gate to be open

m =

h = the probability that an open channel is not blocked

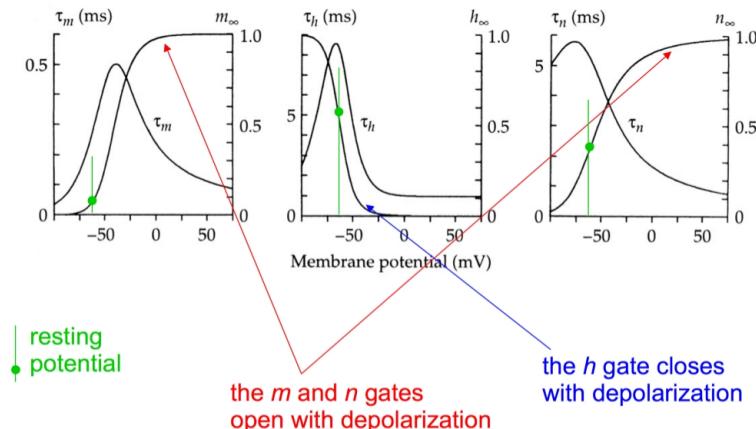
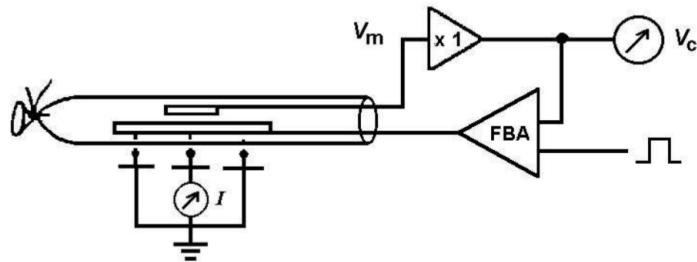


Figure 3: Gating Variables (from HH Model): voltage - dependence.

5 Lecture 5 - Action Potential (Valerio Mante)

- Voltage Clamp experiment



Here: Set **voltage V**, measure **current I**

Before: Set **current I**, measure **voltage V**

- Command voltage is set by the experimenter, the feedback circuit holds the voltage constant.
- The voltage clamp allows the membrane voltage to be manipulated independently of ionic currents, allowing the current-voltage relationships of membrane channels to be studied.

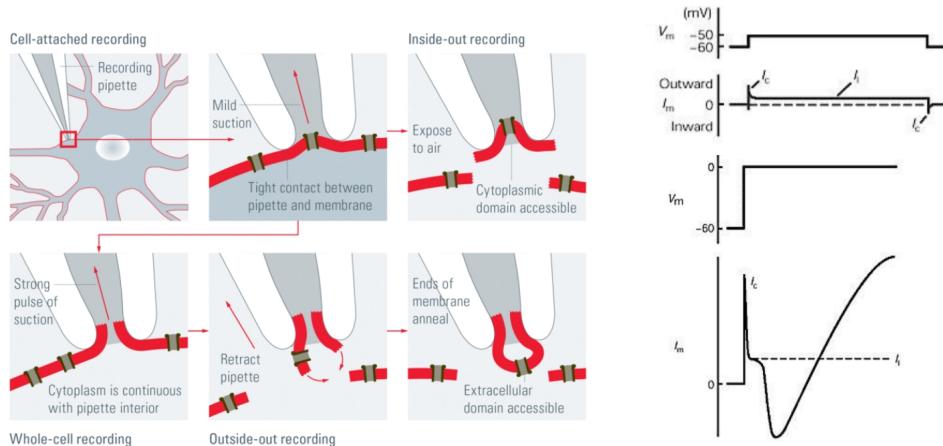


Figure 4: Patch Clamp

5 LECTURE 5 - ACTION POTENTIAL (VALERIO MANTE)

- With negative feedback circuit – The Na^+ current is autocatalytic. An increase in V increases g , which increases the Na^+ current, which increases V , etc.

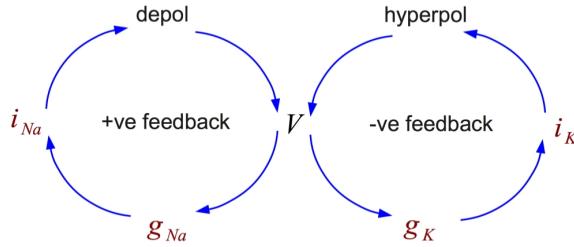


Figure 5: Feedback mechanism underlie the AP

- Voltage and time dependent conductances for g_{Na}, g_K :

g_{Na} increases quickly, but then inactivation kicks in and it decreases again.

g_K increases more slowly, and only decreases once the voltage has decreased.

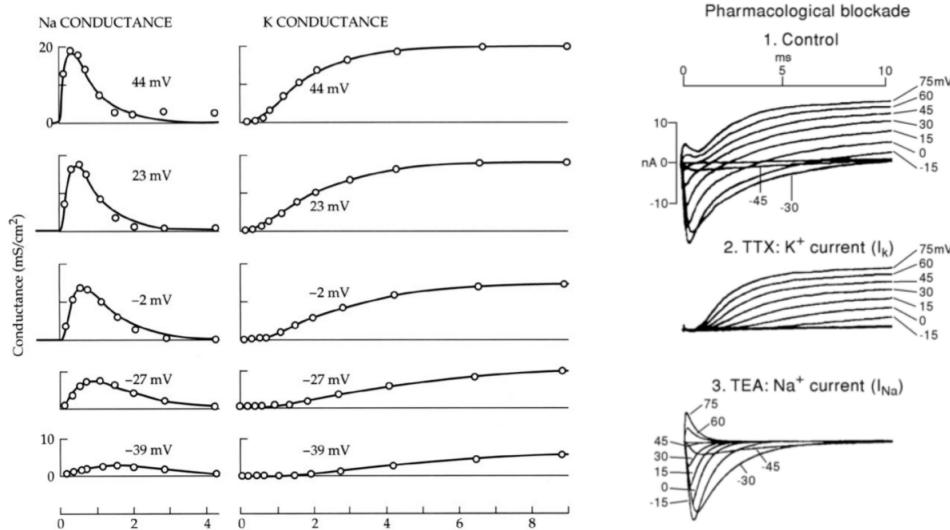


Figure 6: From current to conductance

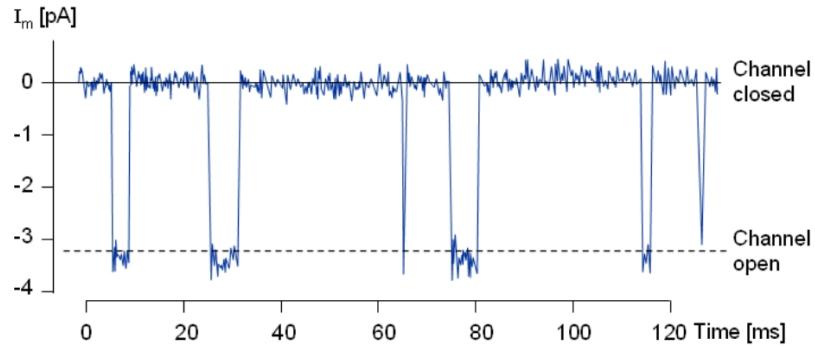
- The threshold for action potential initiation is where the inward Na^+

5 LECTURE 5 - ACTION POTENTIAL (VALERIO MANTE)

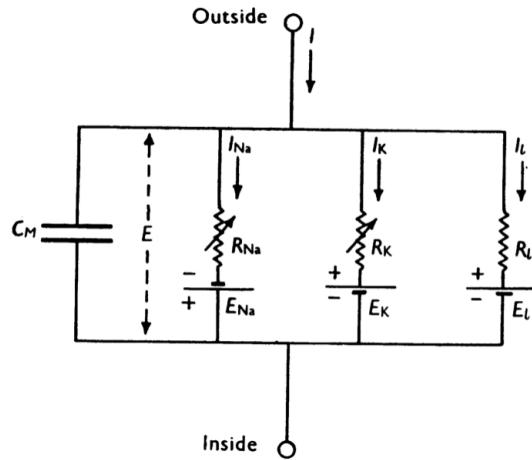
current exactly balances the outward K^+ current.

$\rightarrow g_{Na} > g_K$ (leads to the spike[depolarisation]) $\rightarrow g_K$ increases $b \rightarrow g_K > g_{Na}$ (hyperpolarisation)

- Single Channel current



- The circuit for action potential generation



6 Lecture 6 - Synapse 1 (Kevan Martin)

- Sherrington (1873)

First research on synapses. Word "synapse" & "neuron".

- Otto Loewi & Vagus nerve

Stimulating the vagus nerve slows down the heart beat → inhibitory function. (Injection of the solution from 1st heart to 2nd heart).

Electrical signal → chemical signal.

- Synapse

Only vesicles which are already on the presynaptic membrane will be released after the AP (not all vesicles are released after an AP)

One single synapse produces only a small potential. it needs many synapses to create an actual AP. → Release of neurotransmitters is Ca dependent.

- Probabilistic release of neurotransmitter

in CNS most of the time only one vesicle is released with probability 0.2-0.4

Amplitude histogram → Poisson distribution → probability of firing (all synapses have probabilistic release).

Action potential (probability of firing of synapses, probability of postsynaptic receptors to bind neurotransmitter) = *Plasticity* (the overall probability of passing action potential to postsynaptic neurite changes).

7 Lecture 7 - Synapse 2 (Valerio Mante)

7.1 Synaptic mechanism

1. Synthesis: Building blocks of transmitter substance are imported into the terminal where the neurotransmitter is synthesized and packaged into vesicles.
2. Release: In response to an AP, the transmitter is released across the membrane by exocytosis.
3. Receptor activation: The transmitter crosses the synaptic cleft and binds to a receptor.
4. Inactivation: The transmitter is either taken back into the terminal or inactivated in the synaptic cleft.

7.2 Categorizing synapses according to:

7.2.1 way of activation – complexity

ionotropic receptor	metabotropic receptor
binding site + channel combined	binding site not associated with channel
second messenger-independent	G-protein or 2 nd messenger involvement
short latency action	longer latency
rapid responses (10-50 ms)	slow responses
postsynaptic, in general	pre- and postsynaptic

Ionotropic receptors take part in neurotransmission that involves exclusively ligand-gated ion channels which is much faster (without 2nd messen-

ger).

Ligand – ion or molecule that binds to a central metal atom.

7.2.2 carrier used to transmit information

electrical synapse	chemical synapse
simple primitive system	highly developed structure
often symmetrical, bidirectional	polarized, structurally and functionally
gap junction (connexins)	pre: active zone post: postsynaptic density
very fast, no synaptic delay	slower, synaptic delay (~ 0.5 ms)
Ca^{2+} -independent	transmitter release requires Ca^{2+} influx
temperature-insensitive	temperature-sensitive
large synapse	thousands of small synapses
limited functions, usually excitatory	versatile: excitatory and inhibitory
synchronized activity	specificity: point to point communication

electrical synapse	chemical synapse
simple primitive system	highly developed structure
often symmetrical, bidirectional	polarized, structurally and functionally
gap junction (connexins)	pre: active zone post: postsynaptic density
very fast, no synaptic delay	slower, synaptic delay (~ 0.5 ms)
Ca^{2+} -independent	transmitter release requires Ca^{2+} influx
temperature-insensitive	temperature-sensitive
large synapse	thousands of small synapses
limited functions, usually excitatory	versatile: excitatory and inhibitory
synchronized activity	specificity: point to point communication

7.2.3 function

Excitatory	Inhibitory
$E_S > E_{Th}$	$E_S < E_{Th}$
Glutamate	GABA
Ach nic.	ACh musc.

Neurotransmitters:

- ACh

nicotinic — $E_s \approx 0\text{mV}$, reason why nicotine is addictive → fake excitation

muscarinic — metabotropic – requires 2nd messenger

7 LECTURE 7 - SYNAPSE 2 (VALERIO MANTE)

- Glutamine

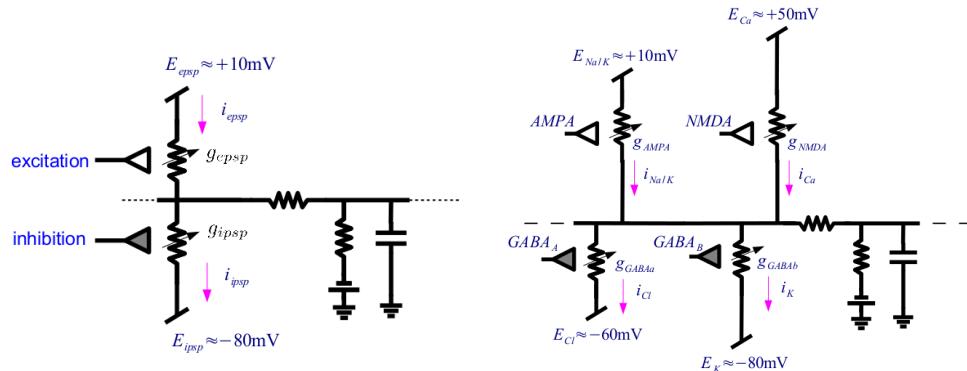
AMPA (AMPA receptors bind Glu & AMPA neurotr.) — contain Mg^{2+} blocking throughput → released by exceeding $30mV$ potential, very fast, $E_s \approx 0mV$

NMDA (NMDA receptors bind Glu & NMDA neurotr.) — ionic transport of Na^+, K^+ ; Ca^{2+} = 2nd messenger; $E_s \approx 0mV$

- GABA

$GABA_A$ — Cl^- , ionotropic (fast), $E_s = \approx -65mV$

$GABA_B$ — K^+ , metabotropic (slow), $E_s = \approx -90mV$



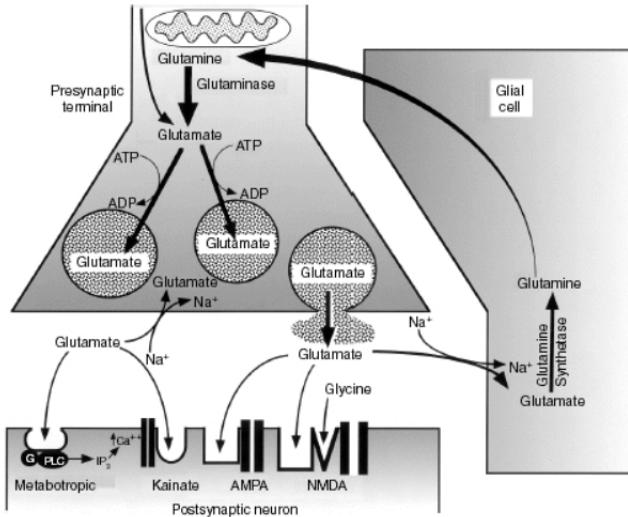
Receptor	Transmitter	ion	Approx E_{rev}	Agonist
AMPA	glutamate	Na, K, Ca	+0mV	AMPA
NMDA	glutamate	Ca, Na, K	+0mV	NMDA(glycine)
mGLU	glutamate	G-coupled		
$GABA_A$	gaba	Cl^-	-65mV	muscimol
$GABA_B$	gaba	K	-90mV	

7.3 Glutamate receptors

- AMPA-receptor: Calcium activates NMDA-receptor. Second messenger activates. Weak stimulation by Glutamate only AMPA receptor is bound to Glutamate.
 $\searrow Na^+, K^+, Ca^{2+}$

7 LECTURE 7 - SYNAPSE 2 (VALERIO MANTE)

- NMDA-receptor: Depolarization $\rightarrow Mg^{2+}$ is removed. Activation by Glutamate and co-agonist $\rightarrow Ca^{2+}$ can float in
 $\searrow Ca^{2+}, Na^+, K^+$
- Important feature of NMDA-receptor
 - Activation of glutamate requires co-agonists glycine or serine.
 - Effect requires coincidence of depolarization of post-synaptic membrane to dislodge Mg^{2+} and binding of agonist.
 - Relatively slow post-synaptic EPSP.
 - 10x more permeable to Ca^{2+} than to Na^+ or K^+



7.4 Neuromodulators **

Neurotransmitter that is not reabsorbed by the pre-synaptic neuron or broken down into a metabolite. Such neuromodulators end up spending a significant amount of time in the cerebrospinal fluid (CSF), influencing (or "modulating") the activity of several other neurons in the brain, as some neurotransmitters also do (serotonin and acetylcholine).

- Norepinephrine
- Dopamine
- Serotonin

8 Lecture 8 - Plasticity/Learning (Michael Pfeiffer)

8.1 Learning & Memory

- Learning is the acquisition of new information or knowledge
- Memory is the retention of learned information
- Types of memory
 - Declarative Memory (Facts,Events)
 - Non-Declarative Memory
 - Procedural Memory (Skills, Habits)
 - Emotional responses
- Facts about Synapses
 - Neurons communicate via AP and are interconnected via synapses
 - Information is represented by distributed activity
 - Learning and memory is based on changes in synaptic connections (Formation & retraction of synapses (development), Changes in synaptics efficacies (plasticity))

8.2 Plasticity

What is plasticity?

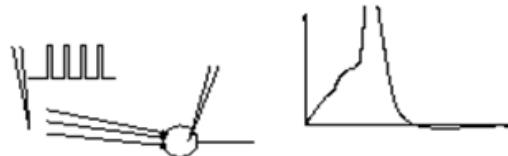
Axiomatic rule: Everything is somehow encoded in synapses.

No information in the shape of the spike, but in the frequency and synchronicity.

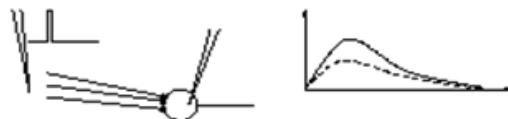
A



B



C

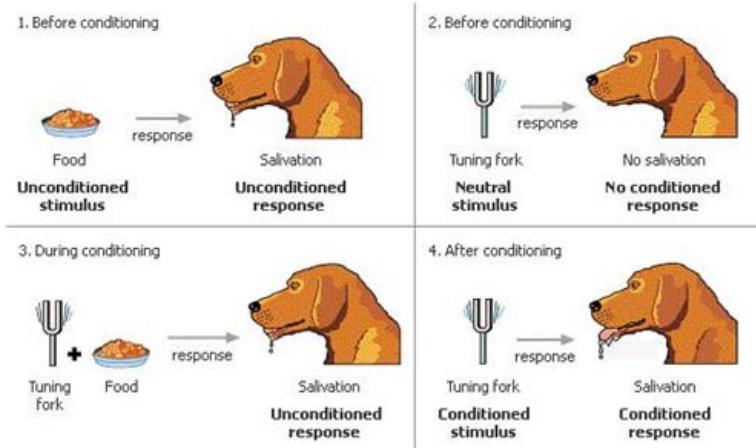


- Modification of postsynaptic potentials (PSP) evoked by presynaptic spikes
 - A. Postsynaptic response triggered by a weak test pulse (left).
 - B. Strong stimulation sequence (left) triggers postsynaptic firing.
 - C. A later test pulse evokes a larger postsynaptic response than initially.
- Parameters that define synapse strengths
 - Neurotransmitter and receptor type
 - Position of synapse
 - Availability of vesicles
 - Re-uptake
 - Neuromodulators(Dopamin etc.)
 - „Non-synaptic“ plasticity(excitability of neurons, dendritic branch strength)
 - Postsynaptic cellular processes
 - Pre-/postsynaptic firing
 - etc.
- it is unlikely that there is only one single model that explains all plasticity effects found in biology.
- Diseases affect plasticity: Alzheimer, Parkinson

8.3 Models of plasticity

- Non-synaptic plasticity (Excitability of neurons, dendritic branch strength)
- Synaptic plasticity
 1. Phenomenological models (High-level, relationships between activity & plasticity etc. exp: Pavlov Classical conditioning)
 2. Biophysical models (Low-level, cellular processes etc. exp: Hebbian learning)

8.4 Pavlovian Learning



8.5 Hebbian Learning

- "Fire together, wire together"
- Learning based on correlations between pre- and postsynaptic firing
- Uses only variables locally available at the synapse
- Rate-based model: $\Delta_{synaptic\ efficacy}_{neuronA,neuronB} \propto firing-rate_{neuronA} \cdot firing-rate_{neuronB}$
- Only weight increase modelled/No depression
→ Can lead to instability (positive feedback loops)
- Other rules: BCM rule, Oja's rule

- Implications:

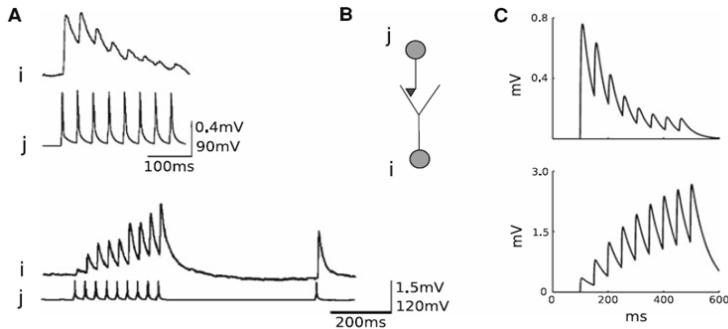
Global effects arise from local learning

Variables(pre- & postsynaptic action potential, efficacy(weight),local concentration)

8.6 NMDA synapse

- Can act as coincidence detector for pre- and postsynaptic firing
- Backpropagation action potentials
- Depolarization from other synapses
- Calcium influx crucial for plasticity
- Strong NMDA activation → potentiation
- Weak NMDA activation → depression

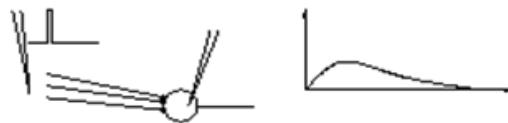
8.7 Short term plasticity(STP)



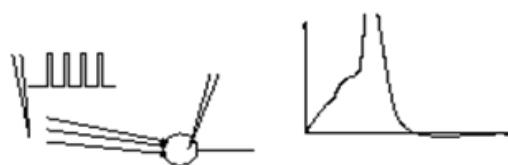
- A neuron j fires several times, neuron i fires as well and the spike size is increased(the higher the spike, the more efficient the neuron), but decreases after a short time.(Caused by loss of vesicles)
- Effect goes away in order of seconds.

8.8 Long term plasticity(LTP)

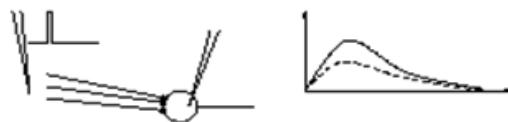
A



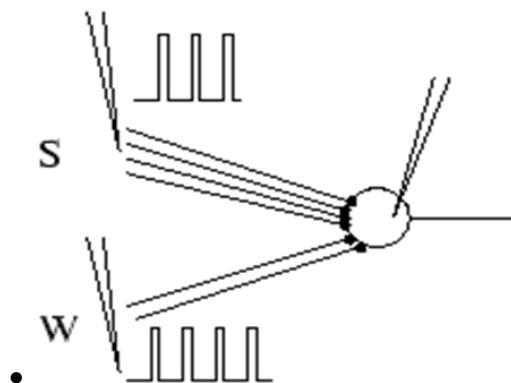
B



C



- Schematic drawing of a paradigm of LTP induction.
 - A. A weak test pulse (left) evokes the postsynaptic response sketched on the right-hand side of the figure.
 - B. A strong stimulation sequence (left) triggers postsynaptic firing (right, the peak of the action potential is out of bounds).
 - C. A test pulse applied some time later evokes a larger postsynaptic response (right; solid line) than the initial response. The dashed line is a copy of the initial response in A. (schematic figure).
- LTP occurs if a synapse and the post-synaptic neuron are simultaneously depolarized beyond a threshold. This can occur in cooperation (weak and weak or weak and strong signals)



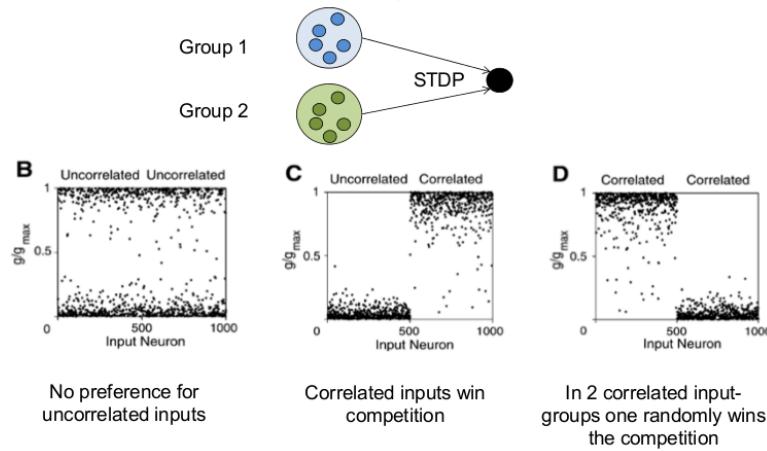
Cooperativity in the induction of LTP. Weaker synapse W is strengthened if the postsynaptic neuron is active and both presynaptic sites are firing.

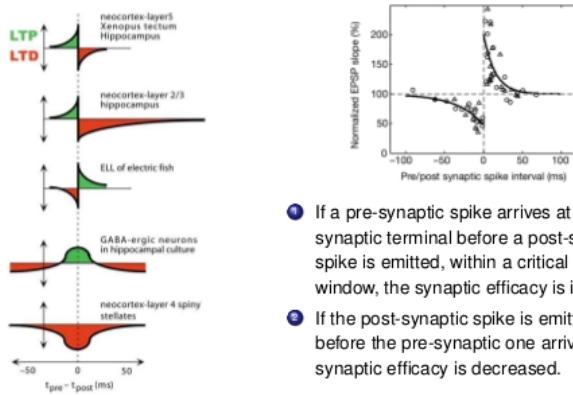
- LTP has a transient early phase (lasting 1-3h) and later phase (24h) demanding new protein and RNA synthesis resulting in the construction of new presynaptic zones and postsynaptic receptors.

8.9 Spike-timing dependent plasticity (STDP)

- Not only correlation, but also timing of the spikes determines plasticity.
- Sign of plasticity is determined by local calcium concentration
- Postsynaptic spike travels back to the dendritic tree and activates voltage-dependent Ca channels
- Presynaptic activity can allow Ca influx through NMDA channels (if postsynaptic part is sufficiently depolarized)
- If pre-spike is soon afterwards followed by post-spike, NMDA-R activity is supralinearly enhanced by depolarization due to backpropagating spike $\rightarrow Ca^{2+}$ determines the strength of plasticity
- Functional consequence:

Correlated firing groups win the battle against uncorrelated groups (depression). Battle of two correlated groups have a random winner.





- ➊ If a pre-synaptic spike arrives at the synaptic terminal before a post-synaptic spike is emitted, within a critical time window, the synaptic efficacy is increased.
- ➋ If the post-synaptic spike is emitted soon before the pre-synaptic one arrives, the synaptic efficacy is decreased.

8.10 Facts

- Different plasticity in different brain areas, diversity in STDP occurrences
- Diversity of neuron and synapse types
- Large number of control parameters for plasticity experiments (frequency, timing, postsynaptic voltage, position on the dendrite, ...)
- Influence of neuromodulators, calcium, drugs, and various proteins
- Long-term vs. short-term effects
- It is unlikely that there is one single model that explains all plasticity effects found in biology

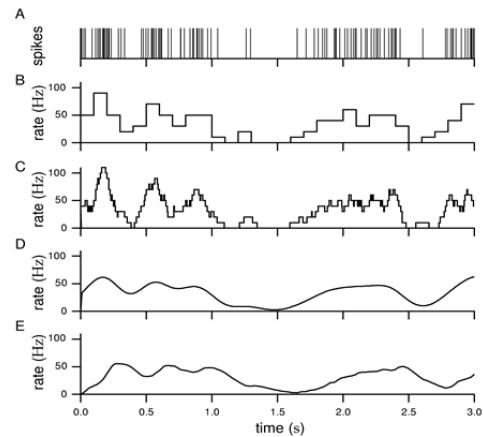
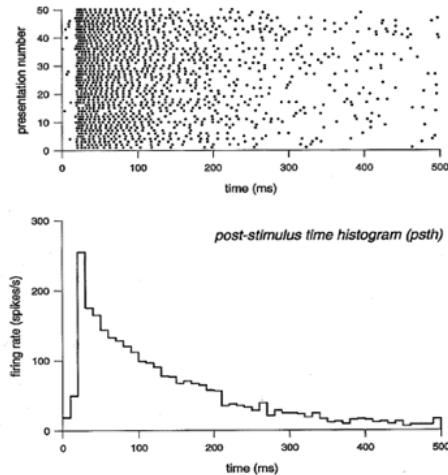
8.11 Dopamine

- Neurotransmitter and neuromodulator, Reinforcement learning
- Significant for motor processes (Parkinson), pleasure and reward, motivation, attention, emotions
- DA activation is a relatively homogeneous, global population signal
- DA activation is related to rewarding stimuli or reward prediction errors
- DA can extend the timing window for LTP, can convert LTD into LTP

9 Lecture 9 - Rate / Event Coding (Michael Pfeiffer)

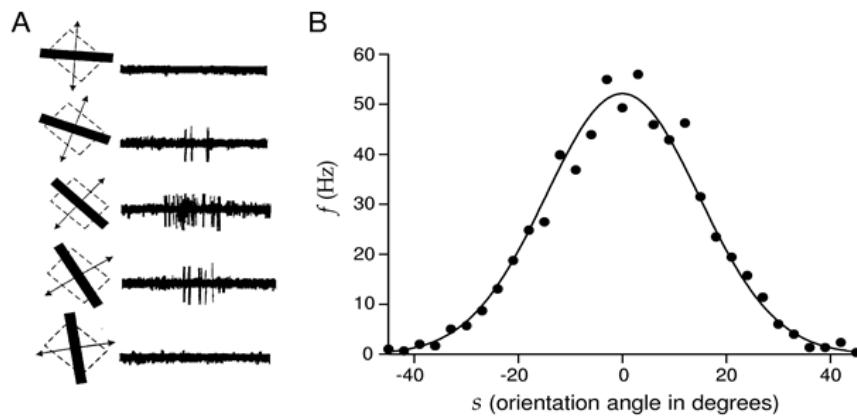
What is neural coding?

- How is information encoded?
- Single neuron firing \leftrightarrow Population firing
- How does a neuron encode information?
- Firing rate, Timing of spikes
- What do different measurement techniques tell us about the neural code?
- Spatial/temporal resolution
- What is a useful visualization of firings for interpretation?

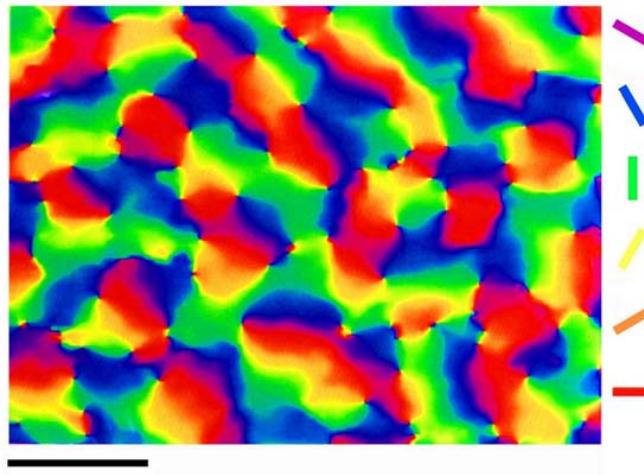


9.1 Neuronal rate codes(average over time(single neuron))

Tuning curves



Shown line and its response primary visual cortex. Tuning curve shows average firing rate of varying stimulus parameters. Tuning curves characterize a single cell.



Nearby neurons have similar preferred orientations(colors)

Rate codes

- + easy to understand

- No timing effects
 - Might be misleading
- More than one stimulus might be encoded

9.2 Poisson Spike Trains

Mathematical model to describe and generate spike trains (point process)
 Poisson distribution for the number of spikes in interval T with firing rate r :

$$P_T(n) = \frac{(rT)^n}{n!} \exp(-rT) \quad (12)$$

Homogeneous: constant rate r

Inhomogeneous: variable rate r

Approximation: probability of a spike occurring in short interval of length Δt :

$$r(t) \cdot \Delta t \quad (13)$$

9.3 What can a single neuron encode?

- Places (on entering a particular region)
- Grids (regularly arranged triangular grid of locations)
- Head-direction
- Single cell responds to one single human face ("Grandmother cell")

9.4 Population rate

9.4.1 Population codes

Different cells encode different range of the stimulus → allows accurate reconstruction of the signal (sparse coding, exp. 3 types of color cones in retina)

- Population vector code
 Populations of neurons stand for vector directions, encoded direction is vectorial addition weighted by firing rate.

9.4.2 Neuronal Event codes

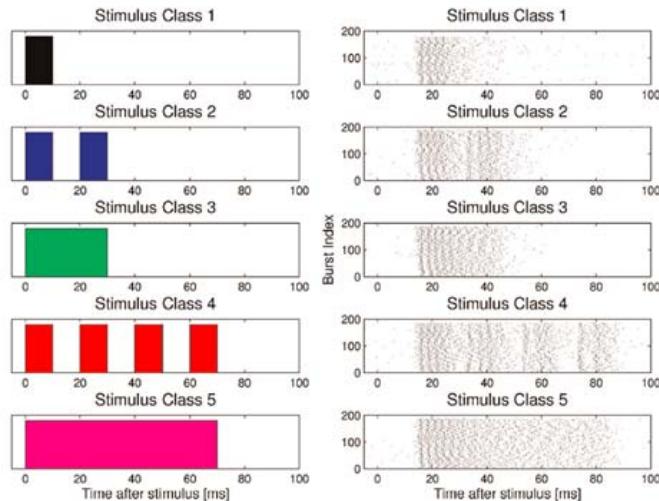
- Time-to-first spike codes

Can implement competition among different cells

Can be rank-order code(sequence matters)

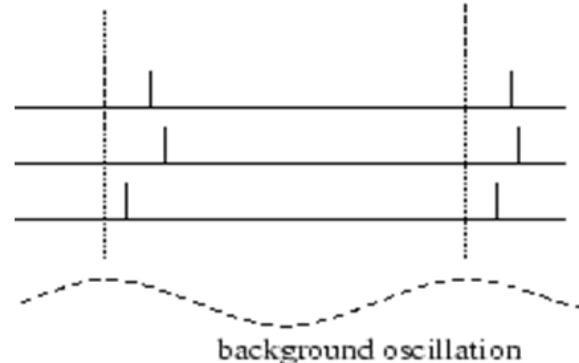
- Burst- and Temporal Codes

Bushcricket auditory neurons in natural environment preserve very high coding precision in extreme noise



- Oscillations and Phase Coding

The neurons fire at different phases with respect to the background oscillation



- Local Field Potential(LFP)

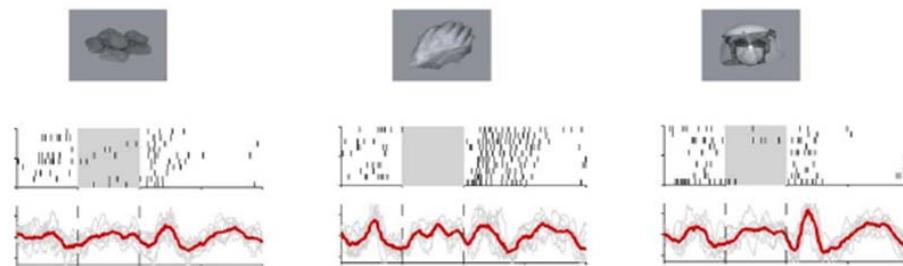
Low-pass filtered extracellular recording

Reflects the integration of membrane currents in a local region

Dominated by dendritic synaptic activity

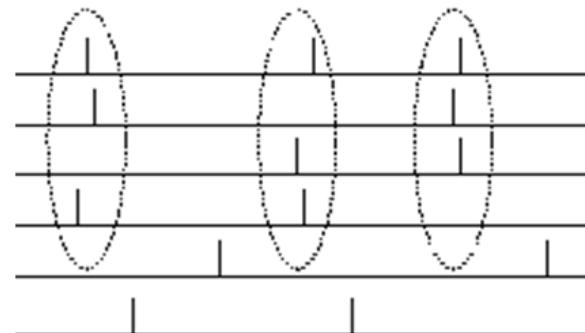
Might encode different properties of the stimulus than single cell firing

Where does it come from / what does it show?

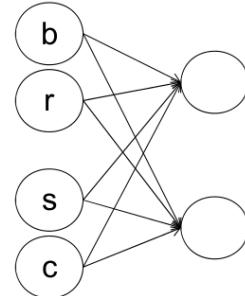
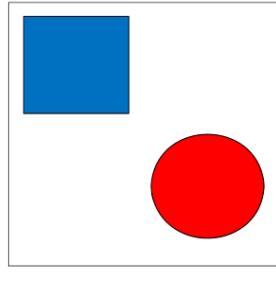


MUA(upper signal) & LFP (lower signal)

- fMRI (functional magnetic resonance imaging)
based on blood oxygenation level
- Synchrony coding



9.5 Binding problem



Occurs frequently:

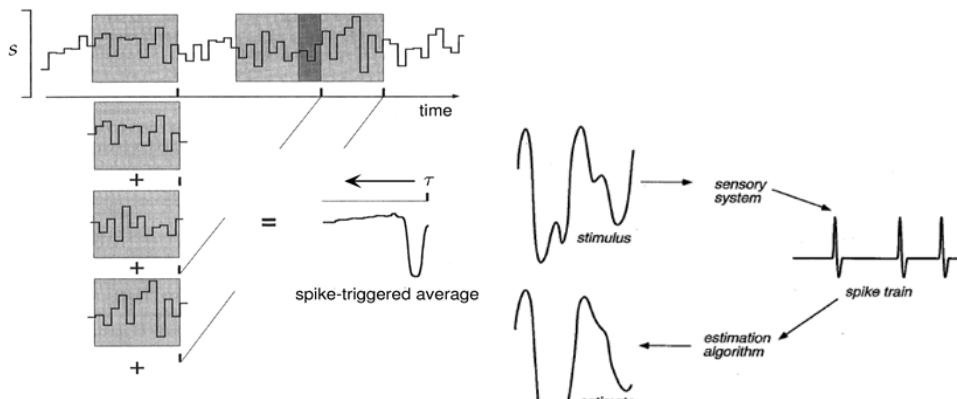
- Visual processing (what? where?)
- Memory
- etc.

Potential mechanisms:

- Temporal synchrony
- Hierarchical coding
- Population coding

9.6 Averages & Estimation

9.6.1 Spike Triggered Average



(a) Average over stimulus in short time window before spike

(b) Stimulus estimation

9.6.2 Issues to remember:

- Whole stimulus reconstruction may not be relevant

9 LECTURE 9 - RATE / EVENT CODING (MICHAEL PFEIFFER)

- Evolution may have shaped us to encode particular features better than others (e.g. faces)
- Cells may respond to only particular aspects of stimulus
- Cells may respond to multiple aspects of stimulus
- Artificial stimuli used for studies may be predictable

10 Lecture 10 - Neuromorphic VLSI(Giacomo Indiveri)

10.1 VLSI

- Very Large Scale Integration Technology allows us to fabricate chips and memory. Digital VLSI(Today's computers) not analog, not low power, not fault tolerant, not robust to inhomogeneities, not asynchronous (clocked), not massively parallel
- Neuromorphic = VLSI systems containing electronic analog/digital circuits that exploit the physics of silicon to reproduce the bio-physics of neural circuits present in the nervous system. Two main goals:

To understand the computational properties of biological neural systems using standard CMOS VLSI technology as a tool.

To exploit the known properties of biological systems to design and implement efficient devices for engineering applications.

- Neuromorphic VLSI neuron circuits

To reproduce the physics of neural computation using subthreshold analog circuits and asynchronous digital circuits.

To build autonomous learning behaving systems that can interact with the environment in real-time

10.2 Why VLSI for neural computation?

- Best exploit current and future VLSI technologies
- Optimally suited for nano- and future emerging technologies
- Ideal tools for real- and accelerated-time modeling of neural systems
- Compact, low-power sensory processing devices for autonomous/flying robots, embedded systems, etc.
- Direct interface to living systems

10.3 MOSFET Characteristics

n-FET subthreshold transfer function

I_0 denotes the nFET current-scaling parameter,

κ_n denotes the nFET subthreshold slope factor,
 U_T the thermal voltage,
 V_g the gate voltage, V_s the source voltage, and V_d the drain voltage.

$$I_{ds} = I_0 e^{\frac{\kappa_n V_g}{U_T}} \left(e^{-\frac{V_s}{U_T}} - e^{-\frac{V_d}{U_T}} \right) \quad (14)$$

Which is equivalent to

$$I_{ds} = I_0 e^{\frac{\kappa_n V_g}{U_T} - \frac{V_s}{U_T}} - I_0 e^{\frac{\kappa_n V_g}{U_T} - \frac{V_d}{U_T}} \quad (15)$$

The first term describes the forward current I_f , the second the reverse current I_r

$$I_{ds} = I_f - I_r \quad (16)$$

If $V_{ds} > 4U_T$ the I_r term becomes negligible, and the transistor is said to operate in the saturation regime:

$$I_{ds} = I_0 e^{\kappa_n \frac{V_g}{U_T} - \frac{V_s}{U_T}} \quad (17)$$

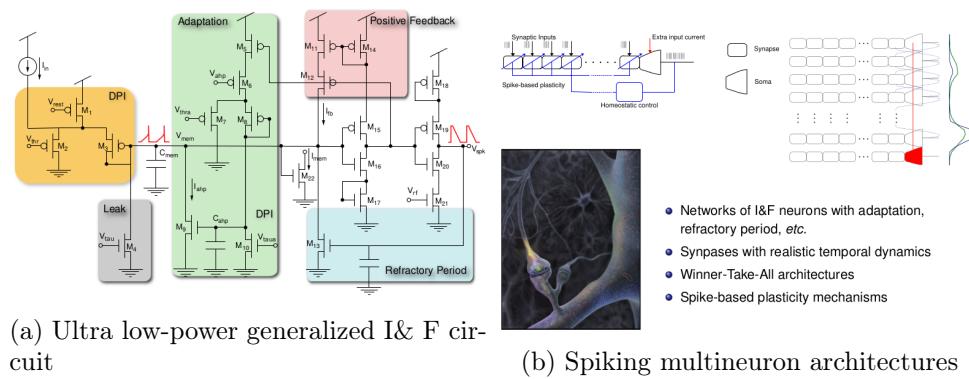
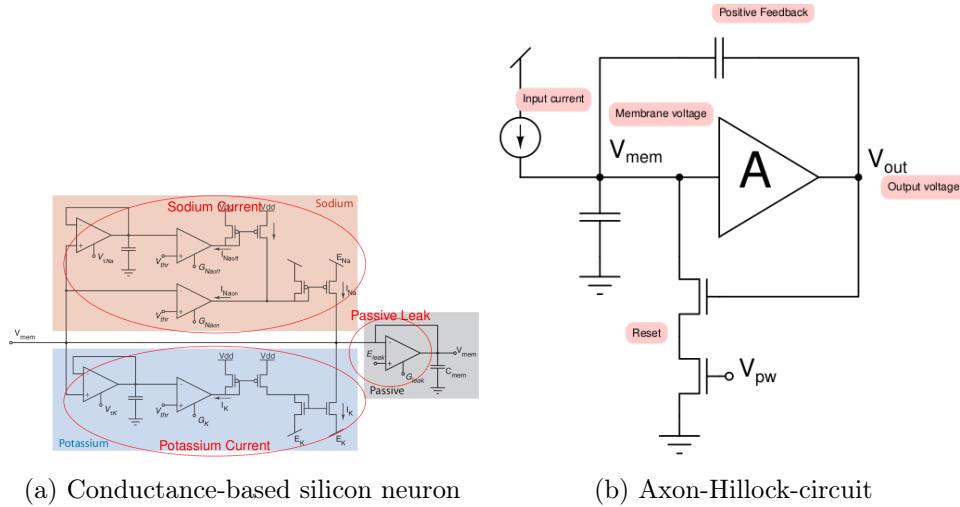
p-FET subthreshold transfer function

In traditional CMOS circuits, all n-FETs have the common bulk potential (V_b) connected to Ground (Gnd), and all p-FETs have a common bulk potential (typically) connected to the power supply rail (V_{dd}). The corresponding (complementary) equation for the p-FET is

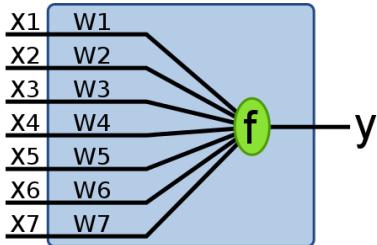
$$I_{ds} = I_0 e^{\frac{\kappa_p (V_{dd} - V_g)}{U_T}} \left(e^{-\frac{V_{dd} - V_s}{U_T}} - e^{-\frac{V_{dd} - V_d}{U_T}} \right) \quad (18)$$

10.4 Different remarkable circuits

- Artificial neuron model by McCulloch & Pitts
- Integrate & fire model (I&F)



11 Lecture 11 - Perceptron Learning Algorithm (Matthew Cook)



Perceptron, McCulloch-Pitts Neuron, Linear Threshold Unit

- Model represents a neuron as a number of inputs X (dendrites) and one output Y (axon). The weights W determine the influence of a dendrite input, which is either excitatory or inhibitory. f is a function that defines how to combine the weights x and w (normally $\sum(w_i \cdot x_i)$) and θ (not shown in picture but usually at the place between y and f) is normally a bias that is added to the sum to compare the result to 0.
- Each neuron has two states: active(1) & inactive(0)
- Summing up the products + the bias $\sum x_i \cdot w_i + bias$, then compared to 0 gives us either an output of 1 for sums ≥ 0 or a 0 if < 0 .
- Using this model, we can create conventional electronic gates such as AND-Gate, OR-Gate and NOT-Gate.
- However the XOR-Gate/NXOR-Gate is not possible in this model. Both functions are not linear.

11.1 Logic Gates

AND



INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

OR



INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

NOT



INPUT		OUTPUT
A		NOT A
0		1
1		0

XOR



INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

NAND



INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

NOR



INPUT		OUTPUT
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

XNOR



INPUT		OUTPUT
A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

11.2 Why XOR is Impossible with Perceptrons

Consider a perceptron with two inputs, x_1 and x_2 , and an output y . x_1 has weight w_1 and x_2 has weight w_2 , and of course we have a threshold t

XOR-Table		
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

$$\begin{aligned}
 0 \cdot w_1 + 0 \cdot w_2 &< t \rightarrow y = 0 \\
 0 \cdot w_1 + 1 \cdot w_2 &\geq t \rightarrow y = 1 \\
 1 \cdot w_1 + 0 \cdot w_2 &\geq t \rightarrow y = 1 \\
 1 \cdot w_1 + 1 \cdot w_2 &< t \rightarrow y = 0
 \end{aligned}$$

This causes a contradiction:

$$\begin{aligned}
 w_1 &\geq t \\
 w_2 &\geq t \\
 0 &< t \\
 w_1 + w_2 &< t
 \end{aligned}$$

11.3 Comparison Perceptrons vs. Real Neurons

- Similarities to biological neurons
 - Active or inactive state
 - Directionality(input/output)
 - Activity dependent of weighted functions of other neurons
- Differences to biological neurons
 - Continous time vs. discrete time
 - Degrees of activation
 - Activation as a function of the inputs of a real neuron is not linear.

11.4 Perceptron Learning Algorithm

Given a set of (input vector x_i , desired output d_i) pairs, finds weights that produce the desired output d_i . (If such weights exist)

- Choose random weights
- Calculate actual output.
- $y_i = f[w(t) \cdot x_j] = \sum w_i \cdot x_i + \text{bias}$
- If wrong output \rightarrow change weights
- For every weight $w_i(t+1) = w_i(t) + \alpha(d_j - y_j(t))x_i$
- This is repeated until the iteration error $\frac{1}{s} \sum_j^2 [d_j - y_j(t)]$ is less than a user-specified error threshold γ or until we completed a predefined numer of iteration.
- Algorithm creates a sufficient result if a solution exists.

Example

(copied from <http://en.wikipedia.org/wiki/Perceptron>)

A perceptron learns to perform a binary NAND function on inputs x_1 and x_2 .

x_0 held constant at 1

Threshold (t): 0.5

Bias (b): 0

11 LECTURE 11 - PERCEPTRON LEARNING ALGORITHM (MATTHEW COOK)

Learning rate (r): 0.1

Training set, consisting of four samples: $\{((0, 0), 1), ((0, 1), 1), ((1, 0), 1), ((1, 1), 0)\}$

In the following, the final weights of one iteration become the initial weights of the next. Each cycle over all the samples in the training set is demarcated with heavy lines.

11 LECTURE 11 - PERCEPTRON LEARNING ALGORITHM (MATTHEW COOK)

Input				Initial weights			Output				Error	Correction	Final weights			
Sensor values		Desired output		Per sensor			Sum	Network					w_0	w_1	w_2	
x_0	x_1	x_2	z	w_0	w_1	w_2	c_0	c_1	c_2	s	n	e	d	$\Delta(x_0 * d)$	$\Delta(x_1 * d)$	$\Delta(x_2 * d)$
				$x_0 * w_0$	$x_1 * w_1$	$x_2 * w_2$	$c_0 + c_1 + c_2$	if $s > t$ then 1, else 0		$z - n$	$r * e$	$\Delta(x_0 * d)$	$\Delta(x_1 * d)$	$\Delta(x_2 * d)$		
1	0	0	1	0	0	0	0	0	0	0	1	+0.1	0.1	0	0	0
1	0	1	1	0.1	0	0	0.1	0	0	0.1	1	+0.1	0.2	0	0.1	0.1
1	1	0	1	0.2	0	0.1	0.2	0	0	0.2	0	+0.1	0.3	0.1	0.1	0.1
1	1	1	0	0.3	0.1	0.1	0.3	0.1	0.1	0.5	0	0	0.3	0.1	0.1	0.1
1	0	0	1	0.3	0.1	0.1	0.3	0	0	0.3	0	+0.1	0.4	0.1	0.1	0.1
1	0	1	1	0.4	0.1	0.1	0.4	0	0.1	0.5	0	+0.1	0.5	0.1	0.2	0.2
1	1	0	1	0.5	0.1	0.2	0.5	0.1	0	0.6	1	0	0.5	0.1	0.2	0.2
1	1	1	0	0.5	0.1	0.2	0.5	0.1	0.2	0.8	1	-1	-0.1	0.4	0	0.1
1	0	0	1	0.4	0	0.1	0.4	0	0	0.4	0	+0.1	0.5	0	0.1	0.1
1	0	1	1	0.5	0	0.1	0.5	0	0.1	0.6	1	0	0.5	0	0.1	0.1
1	1	0	1	0.5	0	0.1	0.5	0	0	0.5	0	+0.1	0.6	0.1	0.1	0.1
1	1	1	0	0.6	0.1	0.1	0.6	0.1	0.1	0.8	1	-1	-0.1	0.5	0	0
1	0	0	1	0.5	0	0	0.5	0	0	0.5	0	+0.1	0.6	0	0.1	0.1
1	0	1	1	0.6	0	0	0.6	0	0	0.6	1	0	0	0.6	0	0
1	1	0	1	0.6	0	0	0.6	0	0	0.6	1	0	0	0.6	0	0
1	1	1	0	0.6	0	0	0.6	0	0	0.6	1	-1	-0.1	0.5	-0.1	-0.1
1	0	0	1	0.5	-0.1	-0.1	0.5	0	0	0.5	0	+0.1	0.6	-0.1	-0.1	-0.1
1	0	1	1	0.6	-0.1	-0.1	0.6	0	-0.1	0.5	0	+0.1	0.7	-0.1	0	0
1	1	0	1	0.7	-0.1	0	0.7	-0.1	0	0.6	1	0	0	0.7	-0.1	0
1	1	1	0	0.7	-0.1	0	0.7	-0.1	0	0.6	1	-1	-0.1	0.6	-0.2	-0.1
1	0	0	1	0.6	-0.2	-0.1	0.6	0	0	0.6	1	0	0	0.6	-0.2	-0.1
1	0	1	1	0.6	-0.2	-0.1	0.6	0	-0.1	0.5	0	+0.1	0.7	-0.2	0	0
1	1	0	1	0.7	-0.2	0	0.7	-0.2	0	0.5	0	+0.1	0.8	-0.1	0	0
1	1	1	0	0.8	-0.1	0	0.8	-0.1	0	0.7	1	-1	-0.1	0.7	-0.2	-0.1
1	0	0	1	0.7	-0.2	-0.1	0.7	0	0	0.7	1	0	0	0.7	-0.2	-0.1
1	0	1	1	0.7	-0.2	-0.1	0.7	0	-0.1	0.6	1	0	0	0.7	-0.2	-0.1
1	1	0	1	0.8	-0.2	-0.1	0.8	-0.2	0	0.6	1	0	0	0.8	-0.2	-0.1
1	1	1	0	0.8	-0.2	-0.1	0.8	-0.2	-0.1	0.5	0	0	0	0.8	-0.2	-0.1
1	0	0	1	0.8	-0.2	-0.1	0.8	0	0	0.8	1	0	0	0.8	-0.2	-0.1
1	0	1	1	0.8	-0.2	-0.1	0.8	0	-0.1	0.7	1	0	0	0.8	-0.2	-0.1

12 Lecture 12 - Hopfield Networks(Matthew Cook)

Key properties of associative memory

- Room for error, but still recognizable

If some units are retrievable & all others set randomly, the correct units will eventually set wrong units right

how actual memory works (reliable but boundaries/limits not well defined)

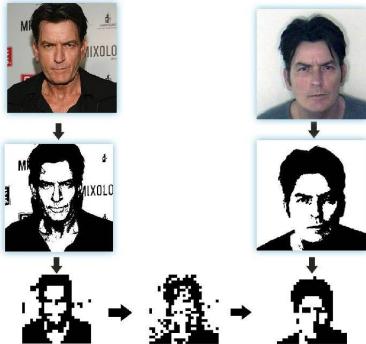
- not stored in list
- Converges to nearby stable states

Only helpful if reliable input

In a Hopfield network (a recurrent network), every node is connected to every other node but not to itself. The nodes therefore form a single layer. The entire network is in some state at any time. The important readout is therefore the state (set of active units) of the entire network rather than the value of some output nodes in other network types. Some of the states are stable and some are not. While the network is not in a stable state, updating the network leads to a state change that ultimately converges to a certain stable state (local minimum).

Features:

A hopfield network is an associative type of memory just as the human memory and is able to remember states it learned before. A Hopfield network of 100 nodes can store about approximately 15 pictures(stored as local minima in the network or stable states). Important is that the pictures are distinct, otherwise it can be that the network gets new stable states and the pictures only get "half remembered".



Important properties of Hopfield networks

- Each node has a value of -1/1 or 0/1. It is therefore active or inactive.
- Weight of a connection is correlated to frequency of firing together (Hebbian learning)
- Symmetric, connection weight is the same regardless of direction
 - If Node A is connected to Node B via connection C_{AB} and Node B is connected to Node A via connection C_{BA} , then weight of C_{AB} equals weight of C_{BA}
- Feedback loops, but no connection to itself (cannot feedback on itself)
- No quantitative description (formula)
- Nodes can be updated synchronously or asynchronously

12.1 Updates and State Dynamics

State: Set of units that are active

Dynamics: Units Update their activity level

When we update a node, we consider weights to that node from all other active nodes (nodes with value 1). If the sum of all weights of the active nodes is higher than zero then this neuron is active too. Updates: can be synchronous (all together) or asynchronous (one at a time)

Asynchronous updates converges to a stable state

Asynchronous updates are done with a greedy algorithm

Activities in $\{0, 1\} \rightarrow$ Network (approx weighted graph w. weighted edges (maximized)) operation is greedy max-clique

Activities in $\{-1, 1\} \rightarrow$ Network operation is greedy min-cut

Synchronous update converges to a pair of patterns (flipping) or converge to stable state

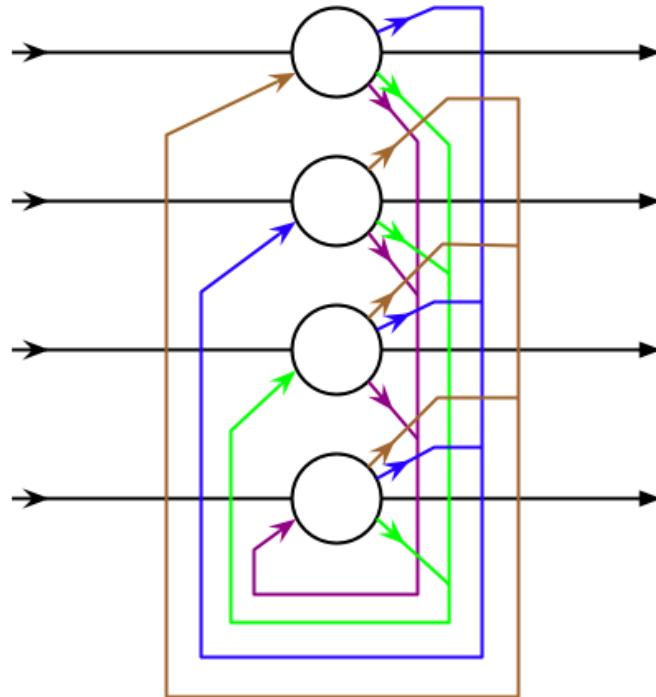
Activities in $\{0, 1\} \rightarrow$ Network (approx weighted graph w. weighted edges (maximized)), operation is greedy max-clique

Activities in $\{-1, 1\} \rightarrow$ Network operation is greedy min-cut

Update function:

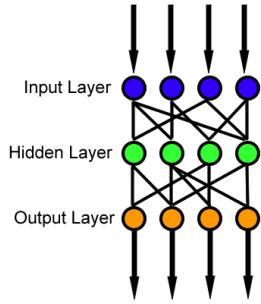
$a_{i,t}$ = output of unit at time t

$$a_{i,t+1} = \begin{cases} 1 & \sum_{j=1}^N a_{j,t} \cdot w_{ij} > \theta_i \text{ (often 0)} \\ -1 & \text{otherwise} \end{cases}$$



13 Lecture 13 - Feed-Forward Networks (Matthew Cook)

Feed-Forward Network with back-propagation is a network that is single-directed(multiple layers of nodes) and has a certain number of inputs x and a certain number of outputs f . Every layer of nodes "feeds" the next layer with information.



FFN are helpful if known input & output pairs exists, we therefore only have to adjust the weights. This procedure is called training or learning. In many applications the nodes use a sigmoid function as an activation function.

13.1 Error Function

To evaluate the performance of the newtork we use an error function:

$$E(\vec{w}) = \sum_{(input, output) pairs} \text{error in output} \quad (19)$$

where $A_1 \dots A_n$ are the values of the output nodes
 \vec{w} = weights of function

$$E(\vec{w}) = [(network\ output) - (desired\ output)]^2 \quad (20)$$

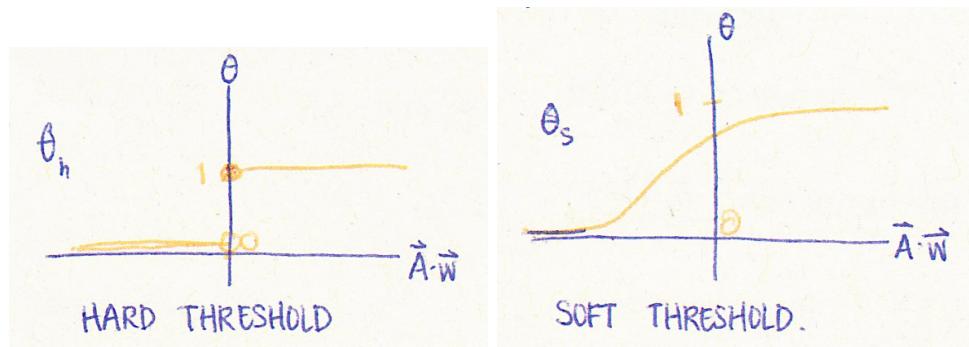
$$E(\vec{w}) = [(A_1, A_2, \dots, A_n) - (D_1, D_2, \dots, D_n)]^2 \quad (21)$$

$$E(\vec{w}) = (A_1 - D_1)^2 + (A_2 - D_2)^2 + \dots + (A_n - D_n)^2 \quad (22)$$

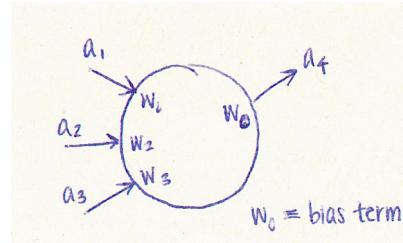
13.2 Improve Error

To improve the error, we need to adjust \vec{w} , ie: calculate $\frac{\delta E}{\delta w_i}$ for each i. This is, however, not particularly helpful for reconstruction $f(x)$ because $\frac{\delta f^2}{\delta} = 0$ everywhere.

We have to use units with continuous range of outputs:



$$\theta_n(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \quad \theta_s(x) = \frac{e^x}{1+e^x} \quad (23)$$



$$a_4 = \theta_s(a_1 w_1 + a_2 w_2 + a_3 w_3) \quad (24)$$

$$\text{where: } \frac{\delta a_4}{\delta a_1} = \theta'_s(\vec{A} \cdot \vec{w}) \cdot w_1 \quad \text{and} \quad \frac{\delta a_4}{\delta w_1} = \theta'_s(\vec{A} \cdot \vec{w}) \cdot a_1 \quad (25)$$

$$\frac{\delta a_4}{\delta w_0} = \theta'_s(\vec{A} \cdot \vec{w}). \quad (26)$$

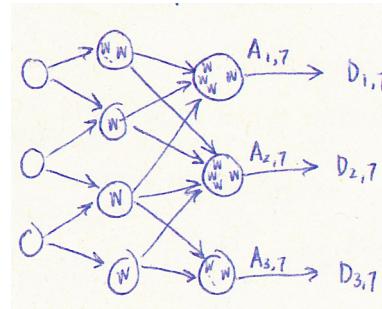
Error of network:

Knowing desired output $D_{p,i}$ can adjust weights in previous activity layer
 → use adjusted activity to adjust weights in previous layer until $A_{i,p} = D_{i,p}$
 \implies Back-Propagation Algorithm
 → useful because we do not need to know network specifics → network can learn on its own.

$$\frac{\delta E}{\delta w_1} = \sum_{p,i} 2(A_{i,p} - D_{i,p}) \frac{\delta A_{i,p}}{\delta w_1} \quad (27)$$

$$A_4 \rightarrow \frac{\delta E}{\delta A_4} \quad (28)$$

$$\frac{\delta E}{\delta w_1} = \sum_{\text{target units } i} \frac{\delta E}{\delta A_i} \cdot \frac{\delta A_i}{\delta A_4} \quad (29)$$



13.3 Back Propagation Algorithm

Back-propagation is a way method that lets information flow backwards inside of the single directed network. It improves the weight via gradient descent. The back flow of information seems not to be the way biology is doing it. It is a learning technique that compares the output values f_i to

the desired values g_i to compute corrective values with a predefined error function.

$$E = \sum (f_i - g_i)^2$$

The error is then propagated back through the network to adjust the weights of each connection to reduce the deviation from the desired values. The network will converge to either no or just a small error.

To adjust weights properly, one optimizes by a method called gradient descent:

$$\frac{\delta E}{\delta w_k} = \sum 2(f_i - g_i) \cdot \frac{\delta(f_i - g_i)}{\delta w_k} = \sum 2(f_i - g_i) \frac{\delta f_i}{\delta w_k}$$

Feed inputs to network one at a time. Compare output to desired output

$$f_i - g_i$$

For each weight, compute the sensitivity of output

$$\frac{\delta f_i}{\delta w_k}$$

Adjust weights

$$w_k \text{ by } \epsilon \cdot (f_i - g_i) \frac{\delta f_i}{\delta w_k}$$

14 Lecture 14 - Interacting Neural Populations(Matthew Cook)

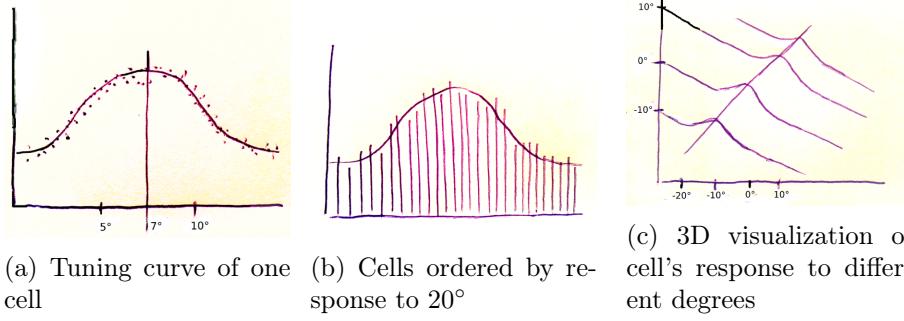
Neurons represent information in many ways, and many of them are unknown. One way of representation is through population code.

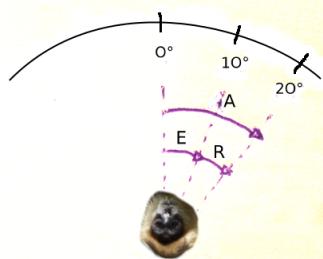
14.1 Population codes

Idea: Information is represented by the pattern of activity in a neural population. Specifically, neurons are "tuned" to preferred stimuli.

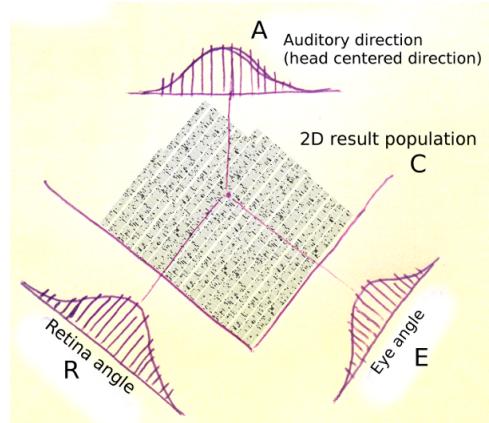
The population represents a value (or several), e.g. the orientation of a line. Here, each neuron has a preferred orientation that it responds to. We say it is "tuned" to this value. Neurons of similar tuning are grouped together in some species (e.g. in cats), in others they seem to be distributed randomly (e.g. in squirrels).

We train a monkey to hold its gaze fixed while we move a visual stimulus around. The monkey holds its gaze fixed at $E = -15^\circ / 0^\circ / 15^\circ$ (eye angle) while we move the stimulus from $R = -40^\circ \dots 40^\circ$ (retinal angle).

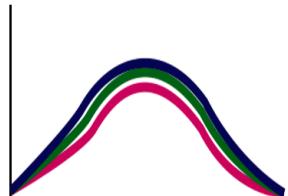




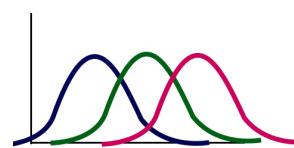
(a) Monkey holding gaze fixed on point 10° and light falling in from 20°



(b) Visualization of Retina angle ordering cell set R, Eye angle ordering set E, 2D result population C and auditory direction set A.



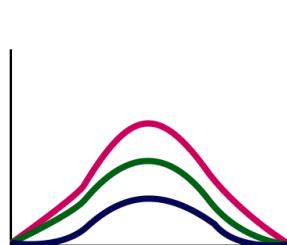
(a) Cell A \in R



(b) Cell B \in A



(c) Cell C \in E



(a) Cell D \in C



(b) Cell E (Not every neuron shows clear tuning curves)