

# **Introduction to Neuroinformatics**

Summary of the lectures 2019

Version 5.0

for internal use by students attending the lecture only

**Initial Version 2012**

Benjamin Ellenberger

**Revision 2013**

Joachim Ott

**Revision 2014**

Fabian Tschopp

Dora Sumislawska

**Revision 2018 and 2019**

Vanessa Leite

**Revision 2020**

Karla Burelo

Nicoletta Risi

Vanessa Leite

# Contents

<b>1 Neuroinformatics</b>	<b>6</b>
1.1 Introduction . . . . .	6
<b>2 Nervous System Organization</b>	<b>7</b>
2.1 Anatomy . . . . .	7
2.1.1 The human brain . . . . .	7
2.1.2 Gross anatomy of the brain . . . . .	8
2.1.3 Building elements of the brain . . . . .	8
2.1.4 The limbic system . . . . .	9
2.1.5 Hypothalamus and thalamus . . . . .	10
2.1.6 Basal ganglia . . . . .	10
2.1.7 Cerebellum . . . . .	10
2.1.8 Reticular Formation . . . . .	10
2.1.9 Cortical areas . . . . .	10
2.1.10 Connections . . . . .	10
2.1.11 Nervous system in numbers . . . . .	11
2.2 Basic structure of the neuron . . . . .	11
2.2.1 Types of neurons . . . . .	11
2.2.2 Components/Terminology . . . . .	12
2.2.3 Axon transport . . . . .	12
2.2.4 Synapse . . . . .	12
2.2.5 Post synaptic receptors . . . . .	12
2.2.6 Networks . . . . .	12
2.3 Muscle reflex and antagonists . . . . .	12
2.3.1 Reciprocal innervation of antagonistic muscles . . . . .	13
2.3.2 Elicitation of a stretch reflex . . . . .	13
<b>3 Membrane Potential</b>	<b>13</b>
3.1 Introduction . . . . .	13
3.2 Membrane structure . . . . .	14
3.3 Hyperpolarization . . . . .	14
3.4 Inputs to neurons, excitatory . . . . .	14
3.5 Chemical synapses . . . . .	15
3.6 Inhibitory post-synaptic potential . . . . .	15
3.7 Summation of Inputs . . . . .	15
3.8 Action potential . . . . .	15
3.9 Axon . . . . .	16
3.10 Ionic currents and equilibrium . . . . .	16
3.10.1 Receptors: . . . . .	16
3.10.2 Action potential: . . . . .	16
3.10.3 Ion equilibrium . . . . .	17
3.11 Permeability equations . . . . .	17
3.11.1 Acting forces . . . . .	17
3.11.2 Nernst equation . . . . .	17
3.11.3 Simplified equation . . . . .	18
3.11.4 Membrane capacitance . . . . .	18
3.11.5 Goldmann-Equation . . . . .	19

<b>4 Passive (Cable) Membrane Properties</b>	<b>19</b>
4.1 Biophysics of the membrane . . . . .	19
4.2 Basic electronics . . . . .	19
4.3 Ion channel replacement circuit . . . . .	19
4.4 Single-compartment model . . . . .	20
4.4.1 Steady-state . . . . .	21
4.4.2 Time-dependency . . . . .	21
4.4.3 Equivalent electrical circuit . . . . .	22
4.5 The cable equation . . . . .	22
4.5.1 Solution for simple cases . . . . .	24
4.6 Level of approximation . . . . .	26
<b>5 Action Potential (AP)</b>	<b>26</b>
5.1 Voltage clamp . . . . .	26
5.1.1 Voltage clamp experiment . . . . .	27
5.1.2 Identifying the nature of the current . . . . .	27
5.1.3 Voltage and time dependent conductances . . . . .	28
5.2 Patch clamp . . . . .	28
5.2.1 Persistent conductance . . . . .	29
5.2.2 Transient conductance . . . . .	30
5.3 Full Hodgkin-Huxley Model . . . . .	30
<b>6 Neural Coding</b>	<b>32</b>
6.1 Neuronal Rate Codes . . . . .	33
6.2 Neuronal Temporal Codes . . . . .	33
6.3 Sound localization by measuring the Interaural Time Difference (ITD) . . . . .	34
6.4 Peri-Stimulus Time Histogram (PSTH) . . . . .	34
6.5 Tuning Curves . . . . .	34
6.6 Orientation Maps . . . . .	34
6.7 Poisson Spike Trains . . . . .	34
6.8 What a single neuron can encode . . . . .	35
6.9 Population Rates . . . . .	35
6.10 Population Codes . . . . .	35
6.10.1 Population Vector Code . . . . .	35
6.10.2 Measuring Population Activity <i>in vivo</i> . . . . .	35
6.10.3 Taking multiple stimuli into account . . . . .	35
6.11 Neuronal Event Codes . . . . .	35
6.11.1 Time-to-first Spike Codes . . . . .	35
6.11.2 Burst- and Temporal Codes . . . . .	36
6.11.3 Oscillations and Phase Coding . . . . .	36
6.11.4 Coding by Synchrony . . . . .	36
6.12 Local Field Potential (LFP) . . . . .	36
6.13 fMRI . . . . .	37
6.14 Binding Problem . . . . .	37
6.15 Averages . . . . .	37
6.16 Stimulus Reconstruction . . . . .	37
<b>7 Synapse</b>	<b>37</b>
7.1 Introduction . . . . .	37
7.1.1 Soup vs Spark - Chemical or Electrical? . . . . .	38
7.2 Chemical transmission . . . . .	38
7.2.1 Steps of transmission . . . . .	39
7.2.2 Synapse properties . . . . .	39
7.2.3 Synaptic mechanism . . . . .	40

7.2.4	Receptor types . . . . .	40
7.2.5	Synapse types . . . . .	40
7.2.6	Receptor overview . . . . .	40
7.2.7	Glutamate receptors . . . . .	41
7.2.8	GABA . . . . .	42
7.2.9	Neuromodulators . . . . .	42
7.3	Electrical transmission . . . . .	42
<b>8</b>	<b>Modeling Synapses</b>	<b>42</b>
8.0.1	$E_{syn}$ and $E_{th}$ . . . . .	42
8.1	Alpha function . . . . .	43
8.1.1	Small synaptic input . . . . .	43
8.1.2	Large synaptic input . . . . .	43
8.2	Shunting inhibition . . . . .	44
8.3	Plasticity (Hebb's Law) - 1940's . . . . .	44
8.3.1	Spike-time dependent plasticity . . . . .	45
<b>9</b>	<b>Plasticity and Learning</b>	<b>45</b>
9.1	Learning and Memory . . . . .	45
9.1.1	Types of Memory . . . . .	45
9.1.2	Connections to Synapses . . . . .	46
9.2	Plasticity . . . . .	46
9.2.1	Networks and system plasticity . . . . .	46
9.2.2	Cellular plasticity - Perceptron . . . . .	46
9.3	Synapse plasticity . . . . .	47
9.3.1	Parameters that define synapse strengths . . . . .	47
9.3.2	Models of Synaptic Plasticity . . . . .	47
9.3.3	NMDA Synapse . . . . .	48
9.3.4	Potentiation with NMDA . . . . .	48
9.3.5	Short-term plasticity (STP) . . . . .	49
9.3.6	Long-term potentiation (LTP) . . . . .	49
9.3.7	Long-term depression (LTD) . . . . .	49
9.3.8	Spike-timing dependent Plasticity (STDP) . . . . .	50
9.3.9	Functional consequences of STDP . . . . .	50
9.3.10	Factors that influence plasticity . . . . .	51
9.4	Hebbian Learning . . . . .	51
9.5	Pavlovian Conditioning . . . . .	51
<b>10</b>	<b>Neuromorphic VLSI</b>	<b>52</b>
10.1	VLSI . . . . .	52
10.1.1	Neuromorphic . . . . .	52
10.2	Neuron circuits . . . . .	52
10.3	Circuits . . . . .	52
10.3.1	n-Fet subthreshold . . . . .	53
10.3.2	p-Fet subthreshold . . . . .	53
10.3.3	Differential-pair . . . . .	53
10.4	Remarkable circuits . . . . .	55
10.5	Spikes and Adress Event Representation . . . . .	56
10.6	Neuromorphic chips . . . . .	56
10.6.1	Dynap: DYNamic neuromorphic Asynchronous Processor . . . . .	56
10.6.2	ROLLS: a Reconfigurable On-Line Learning Spiking chip . . . . .	56
10.6.3	Neuromorphic Cognitive Systems . . . . .	56

<b>11 Introduction to Perceptron Learning Algorithm</b>	<b>56</b>
11.1 McCulloch-Pitts neurons vs biological neurons . . . . .	56
11.2 (Basic) Digital Logic . . . . .	57
11.3 Linear Threshold (LT) Unit/Gates (Perceptron) . . . . .	58
11.3.1 Threshold function . . . . .	59
11.3.2 XOR impossibility with LT/Perceptrons . . . . .	59
11.4 Perceptron Learning Algorithm . . . . .	59
11.4.1 Convergence . . . . .	60
11.5 Converting real weights to integer . . . . .	62
<b>12 Hopfield Networks</b>	<b>62</b>
12.1 Hopfield and Memory . . . . .	63
12.2 Updates and State Dynamics . . . . .	63
<b>13 Feed-Forward Networks</b>	<b>66</b>
13.1 Backpropagation and Error function . . . . .	67
13.2 Gradient Descent . . . . .	68
13.3 Boltzmann Machines . . . . .	68
<b>14 Interacting Neural Populations</b>	<b>69</b>
14.1 Neural Encoding of Information . . . . .	69
14.2 References . . . . .	72

# 1 Neuroinformatics

## 1.1 Introduction

- Human brain on average:  $1.5\text{kg}$  weight,  $1.2\text{l}$  volume.
- Human brains are large, but by far not the largest (for instance, elephants and whales have bigger brains).
- The brain mainly is there to receive stimuli from the environment, encode it, do sensory integration and finally decode to make movements, actions and decisions.
- The cells (neurons) that make up brains are very similar between species.
- Some neuron types occur in specific parts of the brain and it is consistent between species.
- A neuron is a processing unit that receives electrical input (dendrites) and generates electrical output (axon).
- Communication between neurons happens through synapses, in a directed way (pre  $\rightarrow$  post). This can cause a long effect called learning through spikes (or action potential).

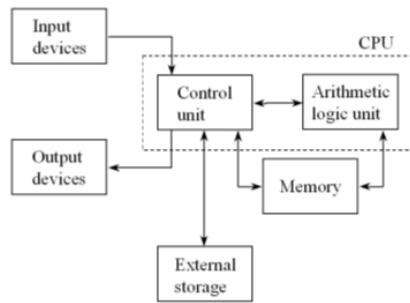


Figure 1: Turing Machine / CPU

**How information is processed in the brain?** We can consider many levels when we talk about brain processing information. From the highest to the lowest level: behavior  $\rightarrow$  system and pathways  $\rightarrow$  circuits  $\rightarrow$  neurons  $\rightarrow$  microcircuits  $\rightarrow$  synapses  $\rightarrow$  membrane potential (molecules and ions).

**Differences between a brain and a computer (what the brains have):**

- Massive parallelism
- Constantly adapting
- Chemical signaling
- Unreliable units (brain is noisy compared to a computer)
- Analog computation
- Robust to damage
- Very energy efficient
- Memory fixed in place (as part of each processing unit).

**Similarities between a brain and a computer:**

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

**Easy vs Difficult tasks** Things that we can do and computer can't, change over time. Things that are easy to humans can be hard to a computer, mostly because we don't know how to simulate a brain. And we don't know precisely how we do some things: we can do it without think consciously about it.

- It is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility.” – Hans Moravec
- We have the neuron diagram of the worm, but we still don’t know how it works.

## Neurons structure

- Membrane: Separates inside from outside.
- Soma: Cell body, contains nucleus and organelles.
- Dendrites: Connect to soma, provide inputs to soma.
- Axons: Connects to soma, conducts away from soma. Often myelinated and ends in synapses. Carries output.
- Synapse: Pre- and postsynaptic terminals, transmit information between neurons.
- In the order of magnitude, there are about 10'000 synapses per neuron.

## Other facts

- An 83000-Processor supercomputer can only match 1% of the human brain.
- C. Elegans (worm) has 302 nerve cells, a frog 16M, a cat 1B and a human 85B.
- Human brain project aims to simulate the entire human brain on computers.
- More efficient simulations of brain behavior by Neurogrid or IBM TrueNorth.
- Number of neurons in human brain:  $\sim 10^{11}$ , synapses:  $\sim 10^{15}$ .
- Number of genes in human genome:  $\sim 25000$ .
- The genome of amoeba dubia is around 200 times larger than human genome.
- WindowsXP contains more code (1.5Gb) for operating a personal computer than DNA ( $\sim 750\text{Mb}$ ) to generate life.
- Deep Neural Networks were inspired by the brain in the beginning but it is not like the brain (neurons) works.

# 2 Nervous System Organization

## 2.1 Anatomy

- Central nervous system (CNS): Brain and spinal cord.
- Peripheral nervous system (PNS): somatic and autonomic (sympathetic and parasympathetic) NS.
  - sympathetic NS: driven by adrenalin - fight/flight reactions
  - parasympathetic NS: rest and digest
- Cranial nerves: they are the gates between the sensory periphery and our brain.
- Brain cuts: Horizontal plane cut, coronal/frontal cut and sagittal cut (between eyes). Cross-section through spinal cord, for example.
- The skull protects, meninges envelope the CNS and has 3 layers, the dura mater, arachnoid mater and pia mater. Primary function is protection.
- The cortex is the layer directly under the surface of the brain.
- There are 4 lobes in each hemisphere. Lobes are separated by fissures in the cortex.

### 2.1.1 The human brain

- Volume of human brain started to increase 2 million years ago
- Most of the studies do not use human brain, they use mouse brain (70 million neurons) instead.

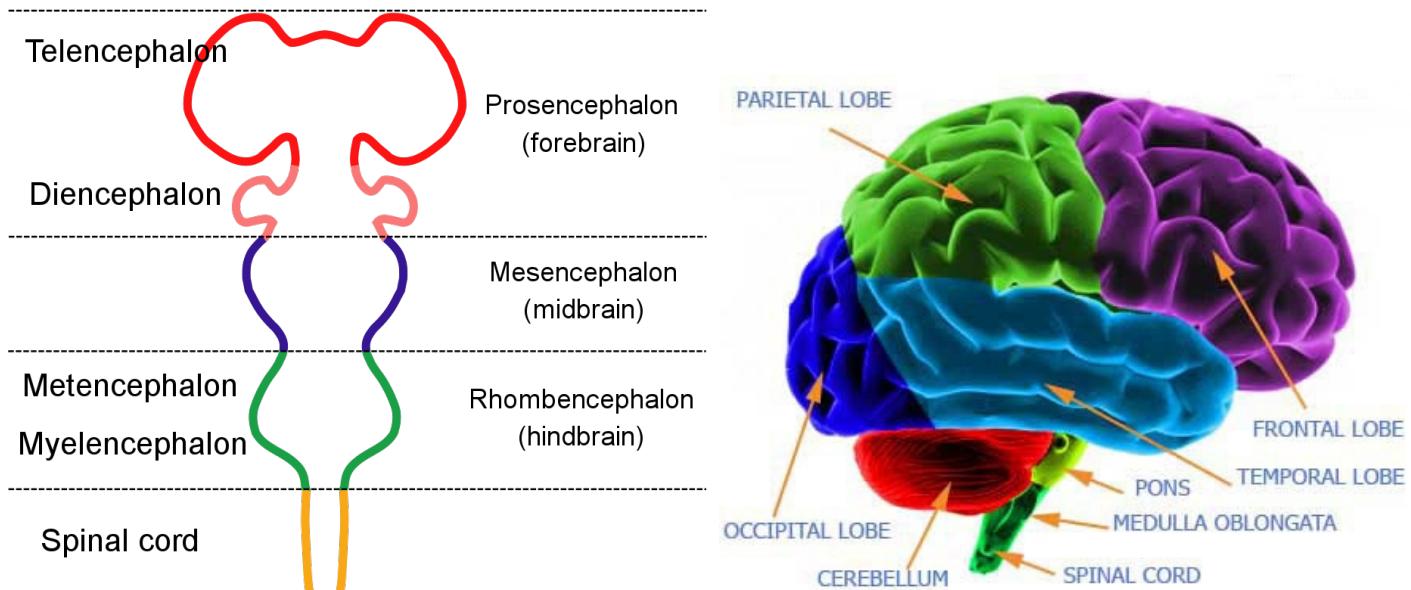
## 2.1.2 Gross anatomy of the brain

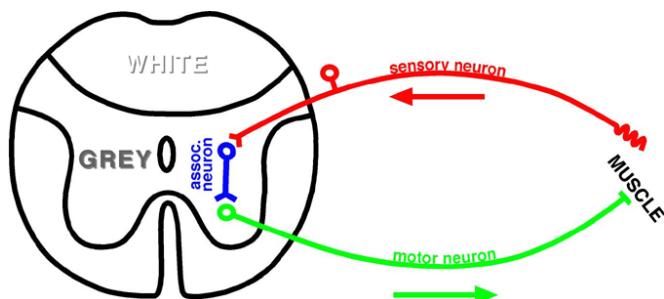
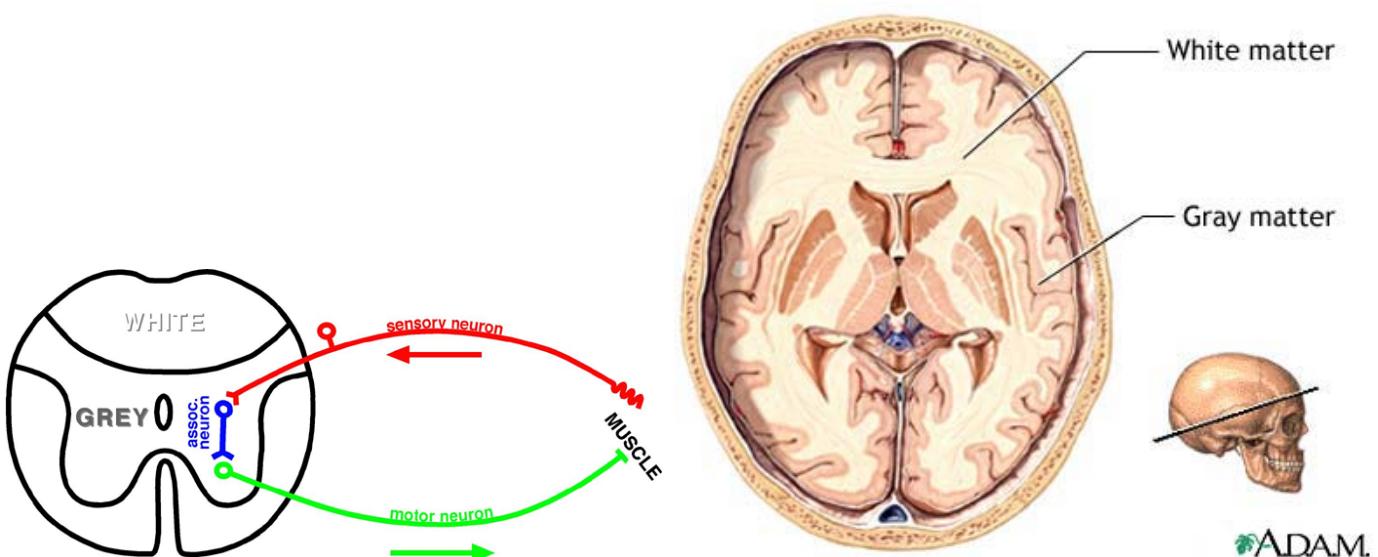
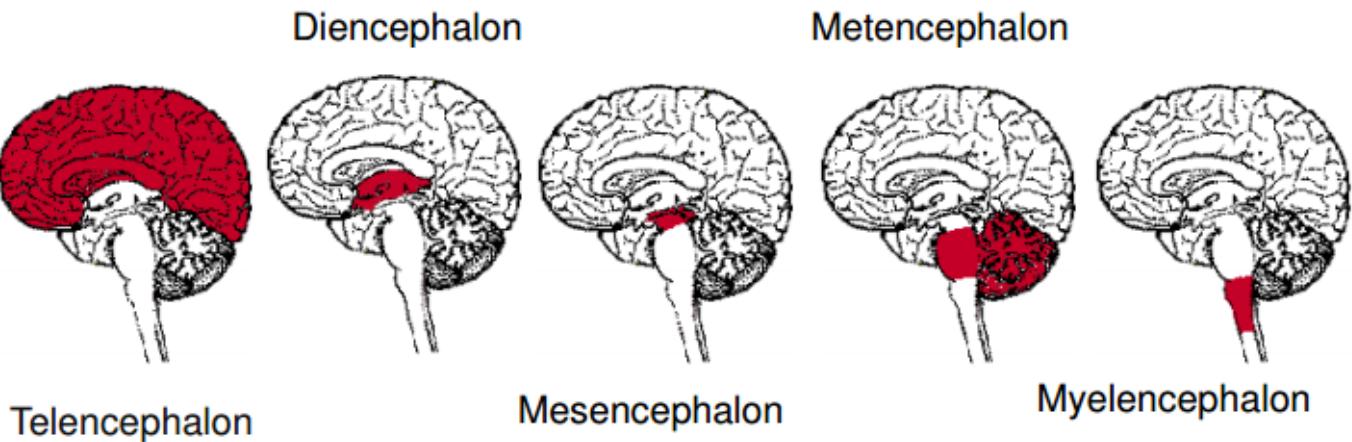
- Skull has 32 bones, its main function is to protect the brain. The hardest bone in the body is the skull.
- The skull provides fixed points for senses (eyes, ears, etc.)
- Below the skull we find the meninges, they protect the brain against infections
  - dura mater
  - arachnoid mater
  - pia mater: follows the brain surface/curvature
- The brain has 4 cavities filled with Cerebral Spinal Fluid (CSF). This is as a protection so the brain do not touch the bones and do not damage itself.
- To "navigate" the brain we use coronal, saggital and horizontal planes.
  - coronal plane: divides the brain in anterior (rostral) and posterior (caudal) parts.
  - saggital plane: divides the brain in left/right parts.
  - horizontal plane: divides the brain in superior/inferior parts.

## 2.1.3 Building elements of the brain

The brain develops from the neural plate.

- Forebrain (Prosencephalon): Cortex, thalamus, hippocampus, basal ganglia, corpus callosum.
- Midbrain (Mesencephalon): Tectum, tegmentum.
- Hindbrain (Rhombencephalon): Cerebellum, pons, medulla oblongata.
- White matter: Glia cells, myelinated axons.
- Grey matter: Neurons (soma).
- Neocortex: Six-layered cortex that forms the surface of most of the cerebral hemispheres.
- Corpus callosum: Midline fiber bundle, connects the two cerebral hemispheres.
- Gyrus: Ridges of the cortex, with valleys (sulci).



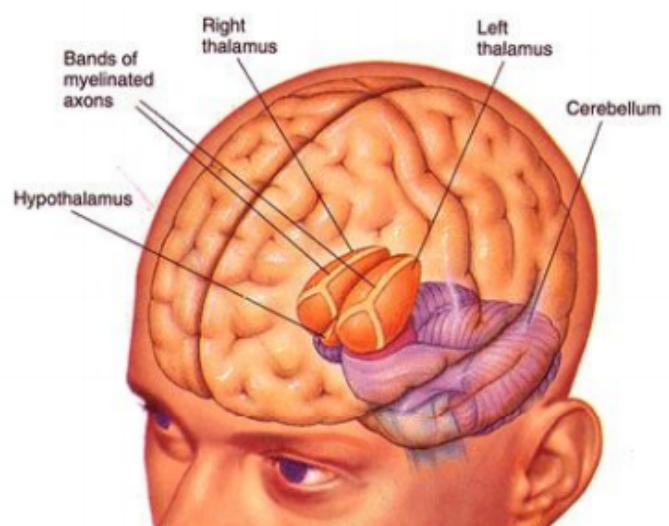
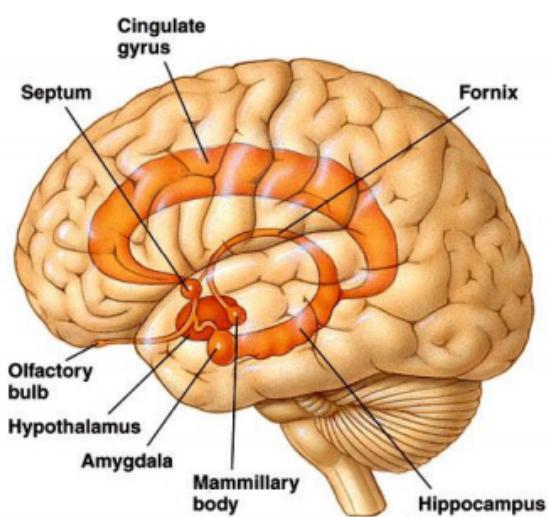


#### 2.1.4 The limbic system

It is related with our emotions.

- Structure: On medial and basal surfaces of cerebral hemispheres.
- Includes cingulate gyrus, parahippocampal gyrus, hippocampal formation, fornix, amygdala, septum, mamillary bodies
- Function: Emotional expression, memory acquisition, fear conditioning, violence and aggression.
- amygdala is really important for fear condition.

##### ► Location of Major Limbic System Structures



## 2.1.5 Hypothalamus and thalamus

Thalamus is the "gate to the cortex"

- The thalamus structure is relatively large with two symmetric large nuclei (receiving ascending and descending inputs) and many projections.
- Thalamus function: relay station, domain-specific information processing.
- The upper brain stem is the diencephalon.
- The hypothalamus is very small and controls autonomic mechanisms.
- Pituitary gland: controls hunger, body temperature, etc.

## 2.1.6 Basal ganglia

It is related with movement control. It triggers the movement, not the fine movement control. It is formed by caudate nucleus + putamen + globus pallidus + substantia nigra + subthalamic nucleus.

- Structure: Collection of nuclei embedded deep within the cortex.
- Partially surrounds the thalamus.
- Sensory projectons to the cerebrum
- Function: Regulate voluntary movement.
- Movement disorders like Parkinson's.

## 2.1.7 Cerebellum

- Structure: "little brain", has layered appearance and symmetry.
- Two hemispheres are connected by the vermis.
- Function: Coordinated motor behavior, posture adjustments and stores memories for simple learned motor responses.

## 2.1.8 Reticular Formation

- Structure: Diffuse arrangement of ascending and descending neurons.
- Function: Arousal, selective attention, respiration.

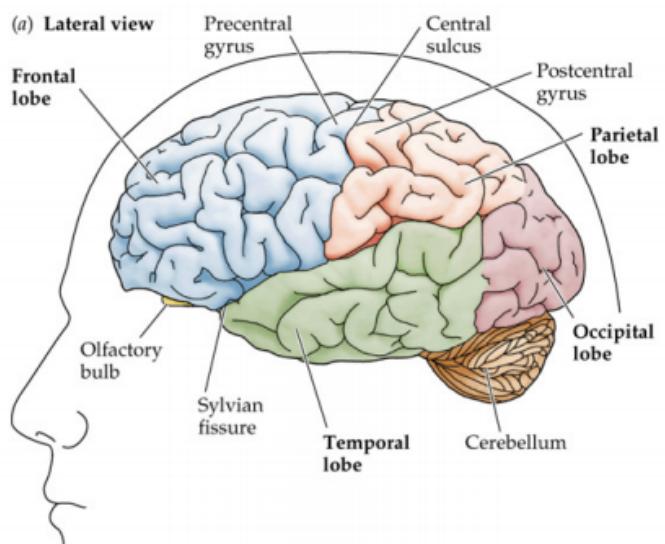
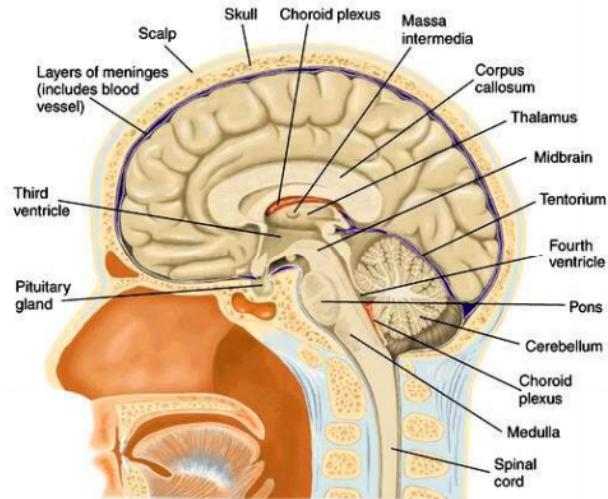
## 2.1.9 Cortical areas

- Phineas Gage: accident in 1848 destroyed left frontal lobe. "The equilibrium or balance, so to speak, between his intellectual faculties and animal propensities, seems to have been destroyed".

## 2.1.10 Connections

- The Basal ganglia projects to the cerebral cortex (via thalamus).
- The cerebellum projects to the cerebral cortex (via thalamus).
- The cerebral cortex projects to basal ganglia, cerebellum and motor neurons (and interneurons) via pons.
- The cortex has six layers.
- From superficial layers to deep layers we say that we have a feedfoward connection. From deep to superficial layers we have a feedback connection.

#### ► Midsagittal View of the Brain and Part of the Spinal Cord



#### 2.1.11 Nervous system in numbers

- 1 mm<sup>3</sup> of white matter is 9 m of axons.
- 1 mm<sup>3</sup> of grey matter is 50'000 neurons.
- In 1 mm<sup>3</sup> about 100'000 cells.

#### 2.2 Basic structure of the neuron

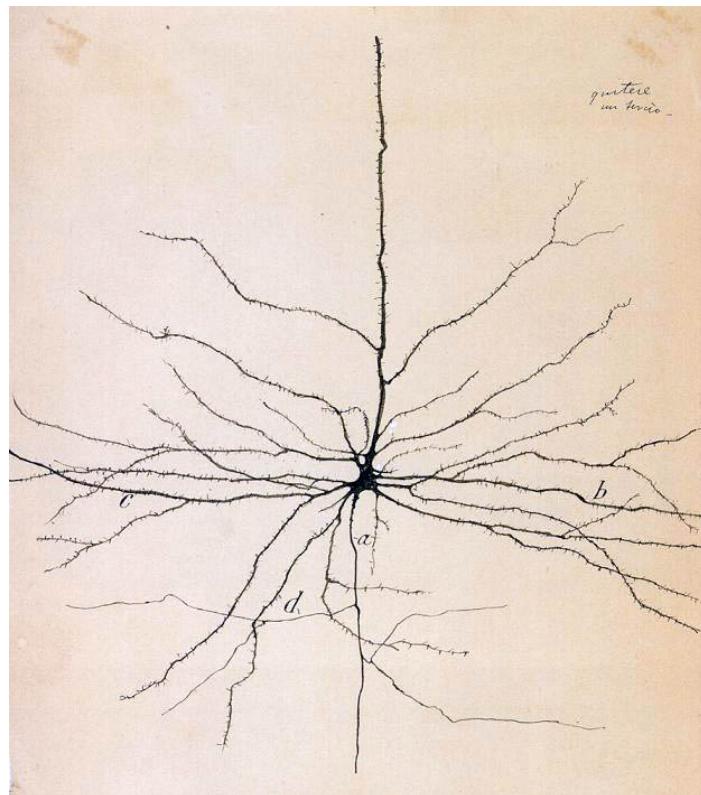


Figure 6: Pyramidal neuron drawn by Ramon y Cajal, founder of modern neuroscience

##### 2.2.1 Types of neurons

There are different types of neurons, for instance: bipolar cells, ganglion cells (retina), cortical pyramidal cells, cerebellar purkinje cells, etc. But all of them follow the same basic structure: cell body, axon and axon terminal.

## 2.2.2 Components/Terminology

- Cell body
- Nucleus
- Dendrite: input component.
- Axon: output component, makes contact to other neurons.
- Myelin: wraps around axons, makes white matter white.
- Boutons: at the ends of the axons, connects neurons.
- Soma: body of a cell without its extensions.
- Afferent: neurons that carry nerve impulses from receptors to the CNS.
- Efferent: neurons that carry information away from the CNS.
- Projection neuron: neuron with long axons that project to distant targets.

## 2.2.3 Axon transport

- Golgi apparatus sits at the cell body.
- Transport of vesicles to the axon terminal (anterograde).
- Transport of empty vesicles back to the cell body (retrograde).

## 2.2.4 Synapse

Conversion between electrical to chemical signals take place in the synapse. Basic sequence: neurotransmitter release → receptor binding → ion channels open or close → conductance change causes current flow → postsynaptic potential changes → postsynaptic cells excited or inhibited → summation determines whether or not an action potential occurs.

Lot of computation occurs in the synapse.

In myelinated axons we have "jumps" of action potential.

Flow of negative/positive charged ions into the neuron can lead to depolarization (excitation) or polarization (depression).

- Boutons: connection point.
- Cleft: little gap between presynaptic and postsynaptic neuron.
- Dendritic spines: dendritic part of the synapse.
- Transmitter: gets released by the presynaptic neuron, in vesicles.
- Vesicles: transport the transmitter inside the cell.
- Receptors: binding site for the transmitter.
- Postsynaptic membrane

## 2.2.5 Post synaptic receptors

- ligand-gated ion channels
- g-protein-coupled receptors

## 2.2.6 Networks

A neuron receives input from many different neurons and can send information to many different neurons as well (colateral on axon).

## 2.3 Muscle reflex and antagonists

1

---

<sup>1</sup>This was not covered in 2018

### 2.3.1 Reciprocal innervation of antagonistic muscles

1. A tact produces a burst of firing (sensory neurons, for example on the finger).
2. The burst excites excitatory spinal interneurons, which then excite the motor neurons of a muscle.
3. The burst also excites inhibitory spinal interneurons that inhibit antagonist muscle motor neurons.
4. One muscle gets contracted, the other relaxed, allowing for a rapid flexion. No brain involved (but gets informed).

### 2.3.2 Elicitation of a stretch reflex

- When hitting the knee tendon with a hammer, the spindles of the thigh muscle get stretched and this elicits a burst of firing in the spindle afferents.
- The burst triggers a burst of firing in the thigh muscle motor neurons, causing contraction.

## 3 Membrane Potential

### 3.1 Introduction

In the lowest level, the brain process information in each processing unit (neurons) through membrane potential, molecules and ions.

Experiments in visual area of monkeys and cats (V1 recordings) allowed us to know more about the visual system. We now know that the visual system has orientation selectivity and the MT area respond to motion/velocity.

Here<sup>2</sup> you can see an experiment and hear the neurons firing.

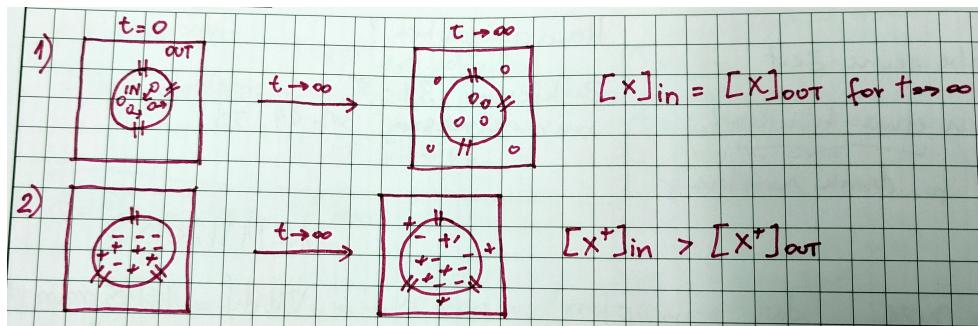


Figure 7: membrane potential experiments.

Figure 7 shows two experiments of membrane potential. In [1],  $[X]_{in} = [X]_{out}$  for  $t \rightarrow \infty$  because of diffusion. In a macroscopic point of view there is no change, microscopically, there is a constant change but on average we have a steady state. In [2], however, we see pairs of ions. Considering that the channels are selective for +, at  $t = 0$  we have no net charge ( $\sum_+ + \sum_- = 0$ ). After a while, + goes out and we have an excess charge on the surface and  $[X^+]_{in} > [X^+]_{out}$  because the inside becomes negatively charged and thus "attractive" for the + ( $V_{in} < 0$ ). Thus, we have a net charge  $V_{in} < V_{out}$ . Analogously, if the channels were selective for -,  $[X^-]_{in} > [X^-]_{out}$ ,  $V_{in} > V_{out}$ .

This example has all the ingredients of the membrane potential in neurons:

- A physical barrier (in vs out)
- $[X]_{in} \neq [X]_{out}$ : different concentrations in/out
- Selective channels

<sup>2</sup><https://www.youtube.com/watch?v=8VdFf3egwfg>

## 3.2 Membrane structure

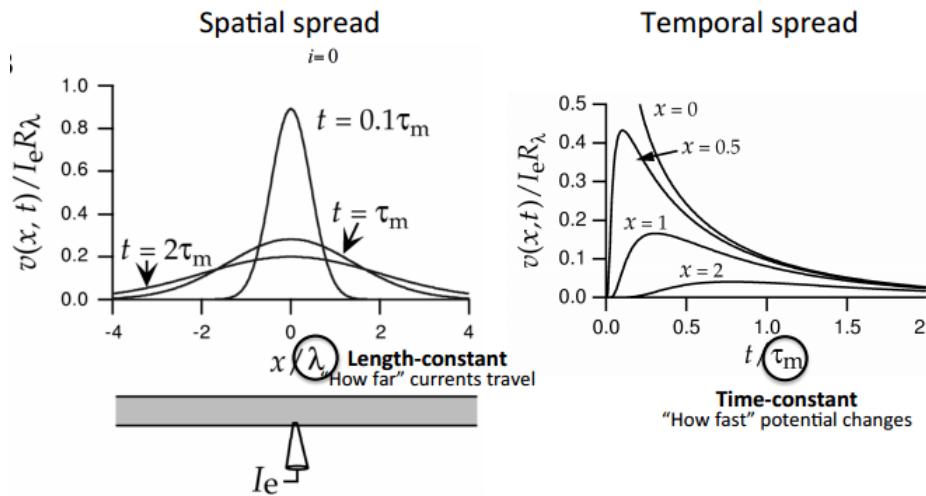
- The membrane is a phospholipid bilayer and creates an energy barrier.
- Ions can not just flow through. Channels and pumps are needed.
- ECS: Extra cellular solution.
- ICS: Intra cellular solution.
- Membrane is built of two types of molecules, charged hydrophilic dipole head-group (outside) and an uncharged, hydrophobic hydrocarbon tail.

## 3.3 Hyperpolarization

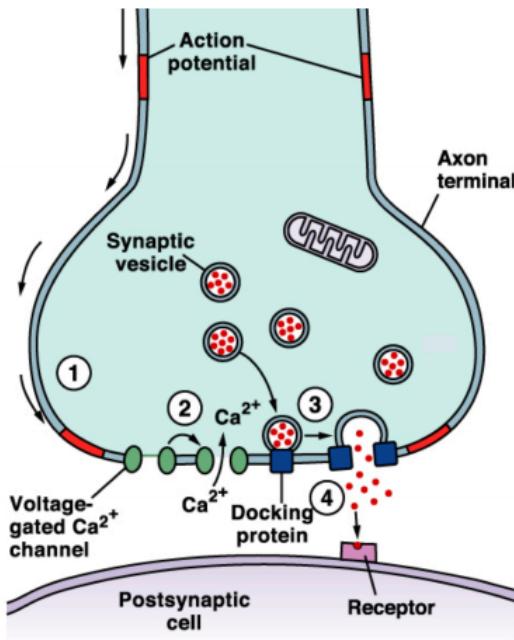
- The extracellular space has potential  $V = 0 \text{ mV}$ .
- The intracellular space has resting potential  $V = -70 \text{ mV}$ . Neurons at rest have their membrane potential mostly determined by  $K^+$ .

## 3.4 Inputs to neurons, excitatory

- Inputs to neurons over synapses.
- Depolarization, about  $V = 30 \text{ mV}$  presynaptic.
- Excitatory current is positive charge, which comes from the extra- to intra-cellular space.
- The signal is analog and graded.
- On the way to the soma, the intra-cellular current gets reduced by leak current (positive charge) that leaves the intra-cellular space.
- From the large depolarization ( $V = 0 \text{ mV}$ ), about  $V = -69.5 \text{ mV}$  is the value at the soma.
- EPSP (excitatory postsynaptic potential) 0.2 to  $0.4 \text{ mV}$ .
- Spatial and temporal spread of the signal.
- $\tau_m$  defines how fast potentials changes.
- $\lambda$  defines how far currents travel.



### 3.5 Chemical synapses



- Digital transmission, but can have failures, and graded release.
- There can even be synapses directly on the soma or axon.

### 3.6 Inhibitory post-synaptic potential

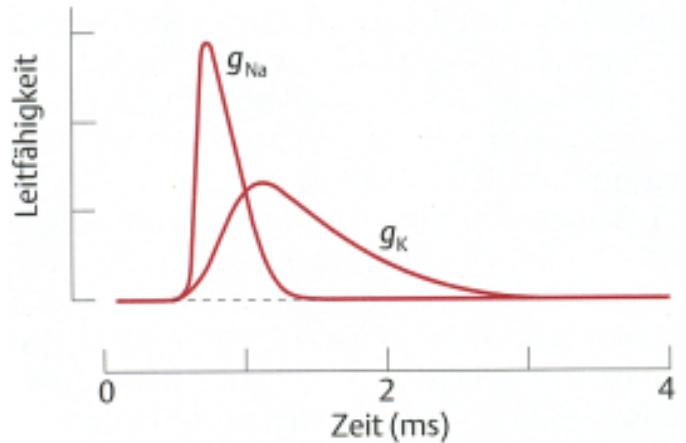
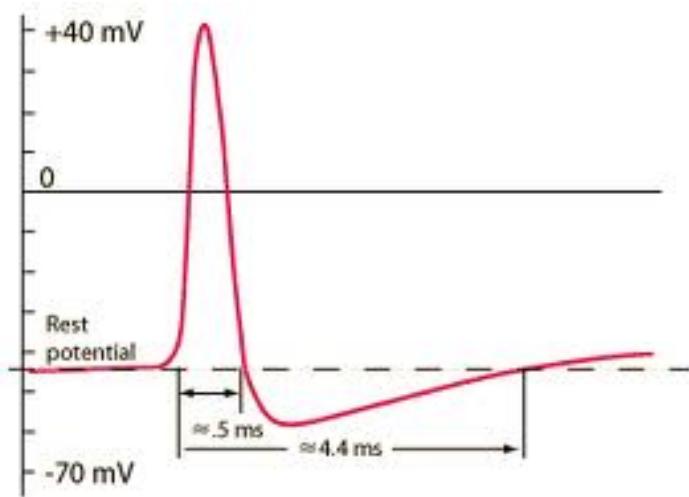
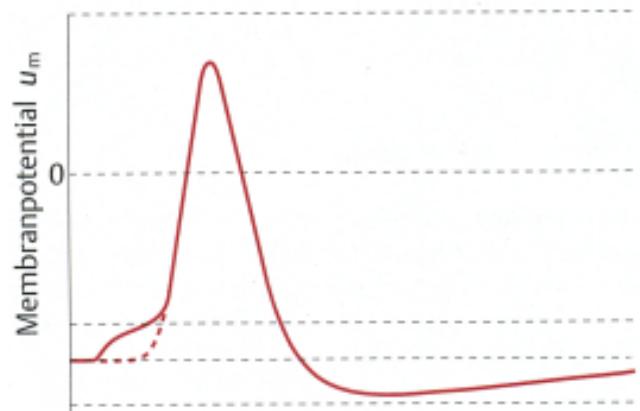
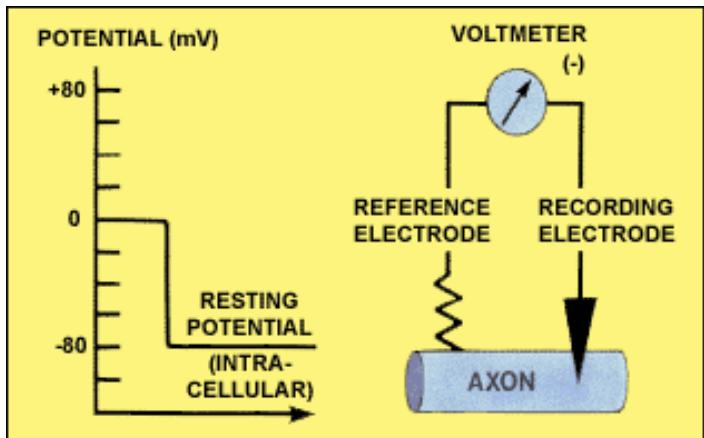
- Depolarization, about  $V = 30 \text{ mV}$  presynaptic.
- A large hyperpolarization,  $V = -90 \text{ mV}$  at postsynaptic dendrite.
- Small hyperpolarization at soma,  $V = -70.2 \text{ mV}$ .

### 3.7 Summation of Inputs

- Temporal summation, one input after another.
- Spatial summation, inputs from different dendrite branches.
- Summing up, the threshold is crossed.
- Typically 20 to 30 Inputs are needed to go above threshold.
- The action potential gets triggered at the beginning of the axon.
- The threshold is about  $-60 \text{ mV}$ .

### 3.8 Action potential

- Has an active regenerative process.
- The duration is about 1 to 2 ms.
- All-or-none (digital).
- Amplitude gets converted into rate.
- Components: Depolarization, overshoot ( $> 0 \text{ mV}$ ), repolarization/hyperpolarization and a refractory period (back to  $-70 \text{ mV}$ ).
- Peak about 0.5 ms long, 4.4 ms refractory period.



### 3.9 Axon

- Myelin sheet is often wrapped around the axon.
- This makes the white-matter white.
- Myelin is an electrical insulator which grants faster propagation.
- Less energy is needed with myelinated axons.
- The current goes through the node of ranvier (myelin sheath gaps).

### 3.10 Ionic currents and equilibrium

#### 3.10.1 Receptors:

- Excitatory: AMPA/NDMA, mixed cation,  $V(\text{drive}) = 0 \text{ mV}$
- Inhibitory: GABA A, chloride ( $\text{Cl}^-$ ),  $V(\text{drive}) = -65 \text{ mV}$
- Inhibitory: GABA B, potassium ( $\text{K}^+$ ),  $V(\text{drive}) = -90 \text{ mV}$

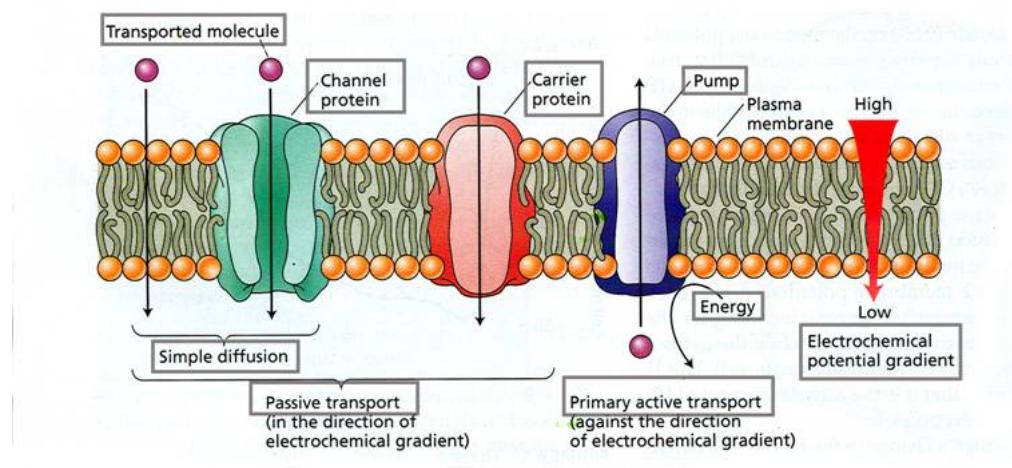
#### 3.10.2 Action potential:

- Sodium ( $\text{Na}^+$ ):  $V(\text{drive}) = 55 \text{ mV}$
- Potassium ( $\text{K}^+$ ):  $V(\text{drive}) = -90 \text{ mV}$

### 3.10.3 Ion equilibrium

Charge carrier (giant squid axon):

Ion type	Cytoplasm (mM)	Extracellular (mM)	Equilibrium potential (mV)
$K^+$	400	20	-75
$Na^+$	50	440	+55
$Cl^-$	52	560	-60
$Ca^{2+}$	0.0001	10	+140



## 3.11 Permeability equations

### 3.11.1 Acting forces

- Ion concentration gradient (diffusion).
- Electric potential (electrostatic force).
- Both forces are in equilibrium in resting/passive state.
- Equilibrium potential can be computed with the Nernst equation.
- Neurons have  $K^+$ ,  $Na^+$  and  $Cl^-$  channels.
- $K^+$  permeability is greater than  $Na^+$  permeability.
- Active  $Na^+-K^+$  pump creates an ion gradient. The exchange is 2  $K^+$  against 3  $Na^+$  ions.
- When a positively charged ion tries to go in, it is repelled by the potential in the membrane. So, the probability of the ion to get in depends of a constant times the amount of positive ions out.

### 3.11.2 Nernst equation

$$V_{eq} = \frac{K_B T}{qZ} \cdot \ln\left(\frac{[Ion]_{extracellular}}{[Ion]_{intracellular}}\right)$$

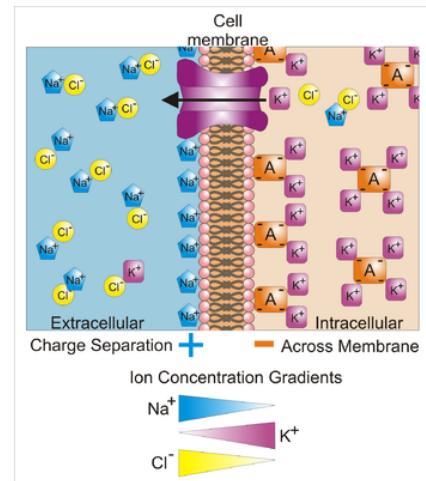
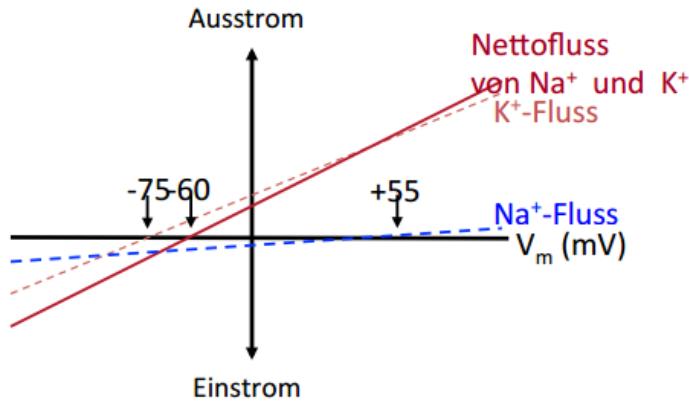
- $K_B$ : Boltzman constant.
- T: Absolute temperature (kelvin).
- qZ: charge of the ion
- $V_{eq}$ : Voltage on equilibrium
- One mole has  $6.022 \cdot 10^{23}$  ions, solution is one molar when its concentration is  $1 \frac{mol}{l}$ .
- This equation doesn't take into account conductance or number of channels. This is because we are looking to the equilibrium state, so it doesn't matter.
- if temperature increases, the potential increases: more thermal energy thus more ions can cross the potential barrier
- if  $|Z|$  increases, then potential decreases: (?) larger energy barrier ( $\Delta E = qZV$ ).
- change in charge sign (Z) leads to change in potential sign
- dependent of concentration:  $[X]_{out} > [X]_{in} \rightarrow V > 0$  ( $Z > 0$ ) and  $[X]_{out} < [X]_{in} \rightarrow V < 0$  ( $Z > 0$ )

### 3.11.3 Simplified equation

- Take the temperature as  $300K$ .
- $K_B = 1.38 \times 10^{-23} \frac{J}{K}$
- $q = 1.60 \times 10^{-19} C$
- Replace  $\ln$  with  $\log$ , gives a factor 2.3.

$$\frac{K_B T}{q} \approx 24 - 27 mV$$

for cold and warmed blood animals.



### 3.11.4 Membrane capacitance

Consider the capacitance of the membrane  $C_m = \frac{c_m}{A}$ , where  $c_m$  is the specific capacitance and  $A$  is the area. Consider  $c_m \approx 10 \frac{nF}{mm^2}$ ,  $A \approx 0.01 - 0.1 mm^2$ , thus  $C_m \approx 0.1 - 1 nF$ .

**How many ions are necessary to get  $V_m = -70 mV$ ?**  $C_m = \frac{Q_m}{V_m} \rightarrow Q_m = C_m \times V_m = 10^{-9} \times 70 \times 10^{-3} C = 7 \times 10^{-11} C$ . Number of ions:  $\frac{Q_m}{q} = \frac{7 \times 10^{-11}}{1.6 \times 10^{-19}} \approx 10^9$  ions.

**How many ions there are in a neuron?** Volume of a neuron is approximately  $10^6 \mu m^3$ , thus, contains  $10^{14}$  ions

Only  $1 : 10^5$  ions contribute to the membrane potential at rest. For each ion that moves, we need a lot that do not. But, what if  $C_m$  and  $Q_m \propto r^2$ , volume  $\propto r^3$ : ratio  $\propto \frac{1}{r}$ . What if potential  $V \neq V_{eq}$ ?

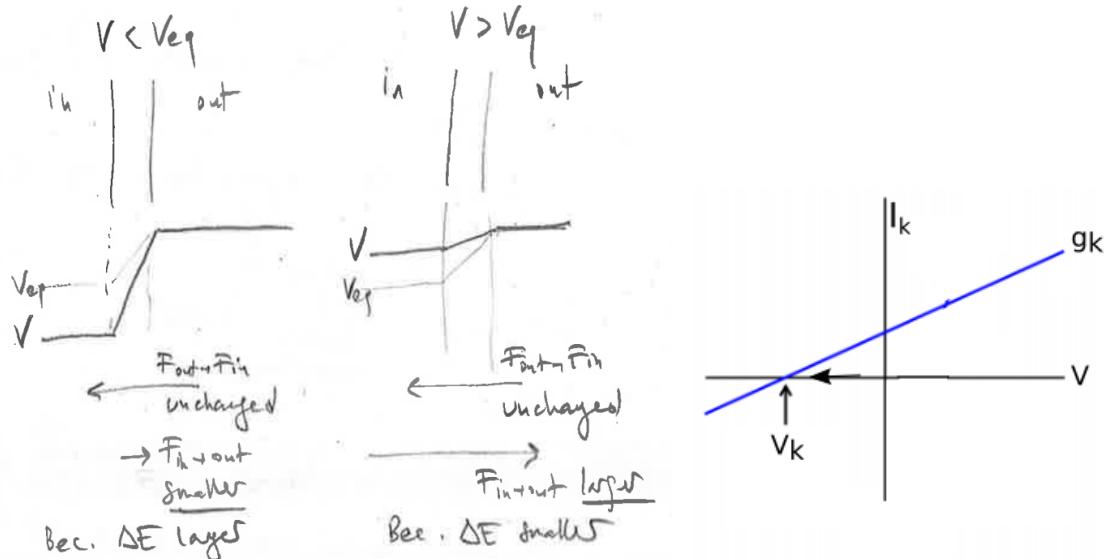


Figure 11: membrane flow

Figure 11 shows the behavior for the case of a different potential.  $I_k$  tries to pull  $V$  towards  $V_k$ . The slope difference of  $g_k$  tell us how much is necessary to pull until achieve the equilibrium.

### 3.11.5 Goldmann-Equation

- Nernst does not consider multiple ions and permeability.
- Goldmann describes membrane potential, with multiple ions and permeability.
- Assumes ion flux obeys Nernst/Planck equation.
- Assumes ions move across membrane independently, without interaction.
- Equilibrium:  $P_K : P_{Na} : P_{Cl}$  is  $1 : 0.04 : 0.45$ .
- Action potential:  $P_K : P_{Na} : P_{Cl}$  is  $1 : 20 : 0.45$ .
- $V_m$ : Membrane potential.
- $P$ : Membrane permeability.
- $[A^x]$ : Ion concentration.
- New  $V_{eq}$  depends on the conductance of the channels.
- $I_k$  is also called  $I_L$  (leak).
- Reverse potential that are bigger than  $I_L$  leads to excitation and smaller leads to inhibition.

$$V_m = \frac{K_B T}{q} \ln \left( \frac{P_K [K^+]_{out} + P_{Na} [Na^+]_{out} + P_{Cl} [Cl^-]_{in}}{P_K [K^+]_{in} + P_{Na} [Na^+]_{in} + P_{Cl} [Cl^-]_{out}} \right)$$

## 4 Passive (Cable) Membrane Properties

### 4.1 Biophysics of the membrane

- Membrane separates in and out. It is an insulator/capacitor.
- Ionic pumps:  $[ion]_{in} \neq [ion]_{out}$
- Selective ion channels

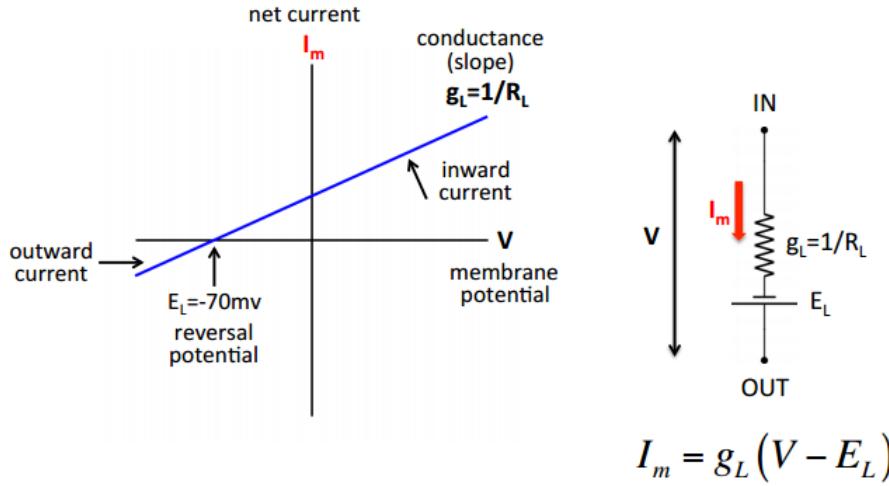
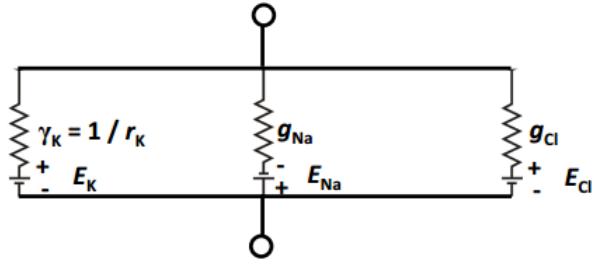
Those three factors give us the reversal (resting) potential:  $E_{ion}$ .

### 4.2 Basic electronics

- Ohm's Law:  $V = I \cdot R$
- Kirchoff's Current Law (KCL): The sum of all currents entering and leaving any node in a circuit is zero.
- Kirchoff's Voltage Law (KVL): The sum of all voltages around a closed loop is equal to zero.

### 4.3 Ion channel replacement circuit

- Ion channel is equal to resistance.
- Ion gradient is equal to battery.
- Cell membrane is equal to capacitor.
- Conductivity  $S = \frac{1}{R}$ .
- Ion resistance is  $R = V/I$ .
- Ion conductivity is  $\gamma = g_m = I/V$ .
- Outward current for an ion is  $I_m = g_m(V - E_m)$ .
- If the concentration of an ion on one side is raised (with a corresponding molecule of opposite charge) and the membrane is permeable for this ion, then the side which has a higher concentration gets more negative, because the ions go to the other side, leaving behind uncompensated negative charges.



We say conductance is ohmic when we have passive properties, thus,  $I = g \cdot V$ . Different channel types have different resting potential. For example, AMPA/NMDA (excitatory) have  $E_{rev} \approx 0mV$ . GABA A and GABA B (inhibitory) have  $E_{rev} \approx -65mV$  and  $-90mV$ , respectively.

Synapses are injecting (externeal) current. Assume dendrite are passive cables that just conduct current (this is a simplification, we are discarding, for instance, backprop action potential and dendrite computation.). This way, we have three models:

- Single compartment model: voltage has only temporal dependency  $V = V(t)$ .
- Cable equation: depends on time and location (analytical solutions)  $V = V(x, t)$ .
- Multicompartment model: (numerical solutions)  $V = V(x, t)$ .

Those approaches offer a trade-off between realism and complexity.

## 4.4 Single-compartment model

### Assumptions, configuration

- Assume isopotential  $V = V(t)$ : holds locally
- Spherical membrane.
- $I_e$  is an injected current.
- $I_C$  is the capacitive current, charges the membrane.
- There is also a leak current  $I_m = g_m(V - E_R)$ .
- $[ion]_{in} = [ion]_{out} \rightarrow V_{rest} = 0$ , equivalent to  $v(t) = V(t) - E_R$

**Membrane as electrical circuit** At  $t = 0$  an external current arrives, the membrane is charged and some current leak. At  $t > 0$ ,  $I_e > I_m$ , i.e., the injected current is bigger than the leak current (some current is charging the membrane). At  $t = \infty$ , we reach the equilibrium,  $I_e = I_m$  and  $I_C = 0$ , with the membrane completely charged, all the current that enters, leave.

- $I_e - I_m = I_C$  and  $I_m = g_m(V - E_R)$
- $C = \frac{Q}{V} \rightarrow CV = Q \rightarrow C \frac{dV}{dt} = \frac{dQ}{dt} = I_C$

- $I_e - gm \cdot v(t) = C_m \cdot \frac{dV(t)}{dt}$
- Membrane time-constant:  $\tau_m = R_m \cdot C_m$   
depends on the resistances and the capacitors. Usual range from 10-100 ms.
- Larger current due to spatial summation.
- Less depolarization with small resistance (larger area).

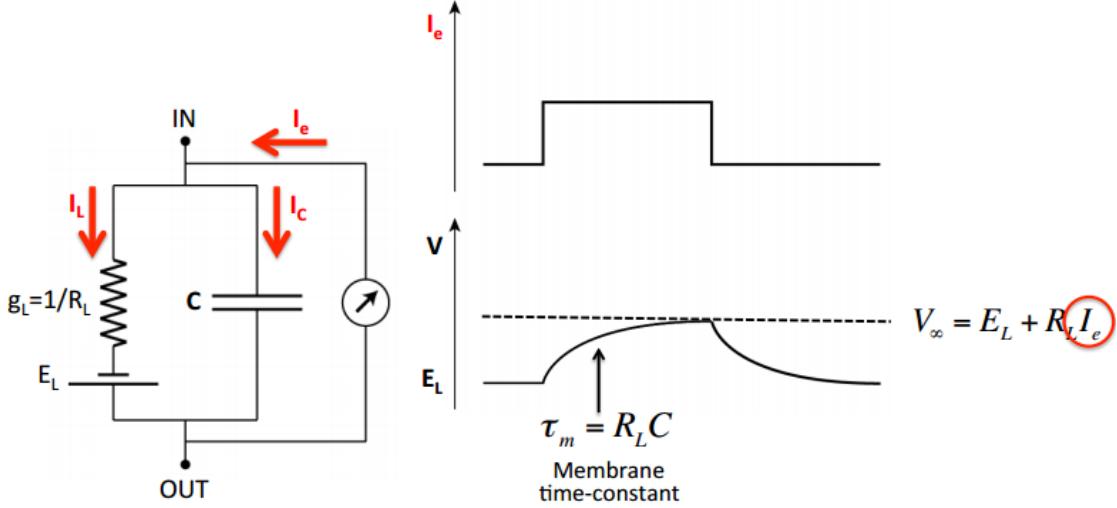


Figure 12:  $R_L = R_m$ ,  $E_L = E_m$ ,  $g_L = g_m$ ,  $I_L = I_m$

#### 4.4.1 Steady-state

$R_m$  as input resistance.

$V_\infty = R_m \cdot I_e + E_m$ : steady-state, i.e., voltage doesn't change anymore ( $\frac{dV}{dt} = 0$ ).

$V_\infty$  increases if  $R_m$  increases (less leak) or  $I_e$  increases (more input)

**Consequences** Two neurons with the same concentration of channels, differing only regarding the size: the smaller neuron will have more resistance, thus, for the same amount of current ( $I_e$ ) it will have smaller voltage change ( $R_{smallneuron} \rightarrow V_\infty^{small}$  and  $R_{largeneuron} \rightarrow V_\infty^{large}$ ). Thus,  $R_m = \frac{r_m}{area}$ .

Two neurons, one myelinated and another unmyelinated. The myelinated has less resistance thus it needs less current input ( $I_e$ ) to achieve the same voltage change.

#### 4.4.2 Time-dependency

$\tau_m$  is the memory of cell  $\approx 10$  to  $100\text{ms}$ . Neuron forgets after  $\tau$ . Longer memory requires few mechanisms, for example, charge in synapse or recurrency.

**several solutions**

$$V(t) = E_m + R_m \cdot I_e + (V(0) - E_m - R_m \cdot I_e) e^{-\frac{t}{\tau_m}}$$

**single solution**

$$v(t) = v_\infty + (v(0) - v_\infty) e^{-\frac{t}{\tau_m}}$$

for  $v = V - E_m$

**Consequences** Consider  $I_{e1} \rightarrow V_1(t)$  and  $I_{e2} \rightarrow V_2(t)$ , then,  $V_1(t) + V_2(t)$  is result of  $I_{e1} + I_{e2}$ .

**Spatial summation** Simultaneous inputs ( $\delta t = 0$ ) sum linearly. If  $I_e \rightarrow k \cdot I_e$  then  $V_\infty \rightarrow k \cdot V_\infty$ .

**Sequential inputs** Sequential inputs sum if  $\delta t < \tau_m$ . It is quite fuzzy, very different from computers.

**Constant inputs** When achieve the threshold, it generates an action potential and right after, reset it. This is known as Integrate and Fire neuron.  $r_{isi} = \frac{1}{t_{isi}} \approx \frac{E_m - V_{th} + R_m I_e}{\tau_m \cdot (V_{th} - V_{reset})}$

**idealized synapse** if  $I_e > 0$  then  $V > E_m$  leading to EPSP (depolarization). if  $I_e < 0$  then  $V < E_m$  leading to IPSP (hyperpolarization).

#### 4.4.3 Equivalent electrical circuit

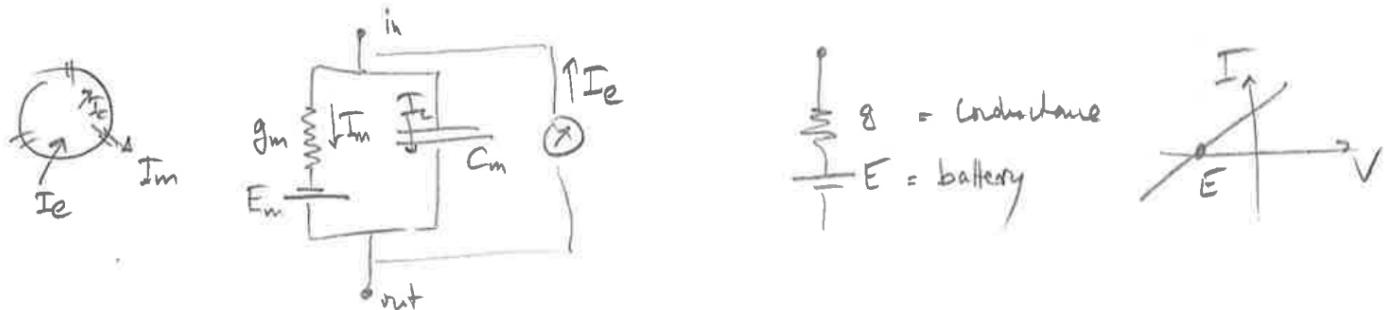


Figure 13: Equivalent electrical circuit for a neuron.

#### Neuron

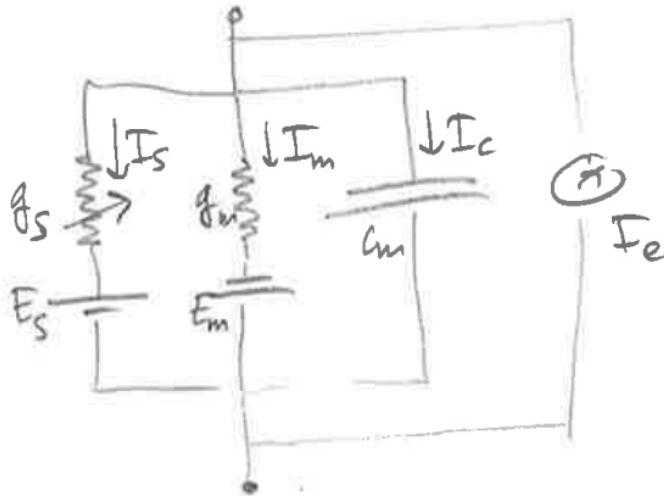


Figure 14: Equivalent electrical circuit for a neuron including synapse. The conductance of a synapse ( $g_s$ ) is variable and depends on presence of neurotransmitters.

**Adding synapse**  $I_e = I_s + I_m + I_c \rightarrow I_e = C_m \frac{dV}{dt} + g_m(V - E_m) + g_s(V - E_s)$  In equilibrium  $I_c = 0 \rightarrow I_e = g_s(V - E_s) + g_m(V - E_m) \rightarrow V_\infty = \frac{g_m \cdot E_m + g_s \cdot E_s + I_e}{g_m + g_s}$

- Equilibrium at  $I_m = I_s$
- $V_\infty = \frac{g_m \cdot E_m + g_s \cdot E_s + I_e}{g_m + g_s}$
- For  $g_s \gg g_m$  (synapse open):  $V_\infty = E_s + \frac{I_e}{g_s}$ , thus the effect of  $I_e$  is reduced: shunting inhibition ( $V_{eq} \approx E_s$ ).
- For  $g_s \ll g_m$  (synapse closed):  $V_{eq} \approx E_m$ .

#### 4.5 The cable equation

So far, ions flow (in → out) to achieve  $E_{rest}$ . But, what if  $V = V(t) \neq E_{rest}$  most of the time?

**leak current**  $\sum_{time} I_{leak} \approx 0$ : balance of exc. and inh, no action potential

**synaptic current**  $\sum_{time} |I_s| > 0$ : depletes concentration gradients  $\rightarrow$  needs pump (needs energy).

**Longitudinal current** In the cable equation we use the same variables as before but we need to express  $I_L$ : longitudinal current. Now we have a current that flows inside the membrane (for instance, left to right).

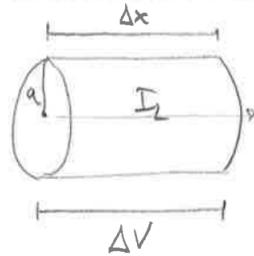
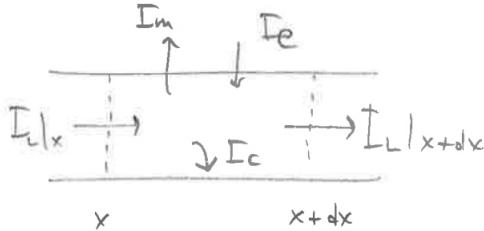


Figure 15: To start, let's consider a piece of cable of length  $\delta x$ . In this piece, we have a voltage drop  $\Delta V$ .

$$R_L = \frac{r_L}{\pi a^2} \cdot \Delta x = \frac{r_L}{Area} \cdot \Delta x.$$

- $r_L$  is a constant, a property of intracellular medium.
- $\frac{1}{Area}$ : resistances in parallel  $\rightarrow \frac{1}{R_{total}} = \sum \frac{1}{R} \rightarrow R_{total} = \frac{R}{N}$ .
- $\Delta x$ : in series  $\rightarrow R_{total} = N \cdot R$

By Ohm's law:  $\Delta V = R_L \cdot I_L = \frac{r_L}{\pi a^2} \Delta x I_L \rightarrow (\Delta x \rightarrow 0) \frac{dV}{dx} x = -\frac{r_L}{\pi a^2} \cdot I_L(x)$ . By definition, the sign is -:  $\Delta V < 0$  for pos. current from  $x \rightarrow dx + x$ .



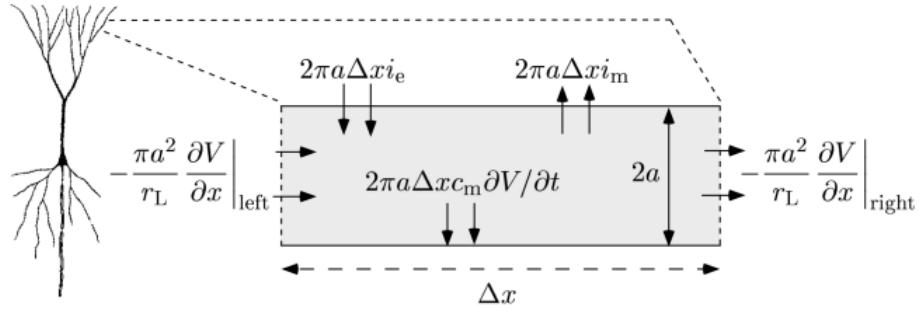
**Charge conservation** In essence,  $I_L(x + \Delta x) = I_L(x) - I_m + I_e - I_c$ , where:

- $I_c \approx \frac{\partial V}{\partial t}$
- $I_m \approx g m (V - E_m)$
- $I_L(x + \Delta x) \approx \frac{\partial V}{\partial x}(x + \Delta x)$

To derive cable equation:

- $I_m = 2\pi a \cdot \Delta x \cdot i_m$
- $I_e = 2\pi a \cdot \Delta x \cdot i_e$   
where  $i_m$  and  $i_e$  are current densities = current/Area
- $I_L = \frac{\pi a^2}{r_L} \frac{\partial V}{\partial x}$
- $I_c = 2\pi a \Delta x C_m \frac{\partial V}{\partial t}$

$$c_m \frac{\partial V}{\partial t} = \frac{1}{2ar_L} \frac{\partial}{\partial x} (a^2 \frac{\partial V}{\partial x}) - i_m + i_e$$



## Note

- $a$  need not to be continuous, i.e.,  $a = a(x)$ .
- $i_m$  could be very complicated
- in general, there is no analytical solution  
but we can consider simple cases where solution exists

Assume  $a$  is continuous,  $i_m = \frac{V-E_r}{r_m}$ , and define  $v = V - E_r$ :

$$c_m \frac{\partial V}{\partial t} = \frac{a}{2r_L} \frac{\partial^2 v}{\partial x^2} - \frac{v}{r_m} + i_e$$

**Define**  $\tau_m = r_m \cdot c_m$  (time constant) and  $\lambda = \sqrt{\frac{a r_m}{2r_L}}$  (length constant: "electrotonic length"), thus:

$$\tau_m \frac{\partial V}{\partial t} = \lambda^2 \frac{\partial^2 v}{\partial x^2} - v + r_m i_e$$

### 4.5.1 Solution for simple cases

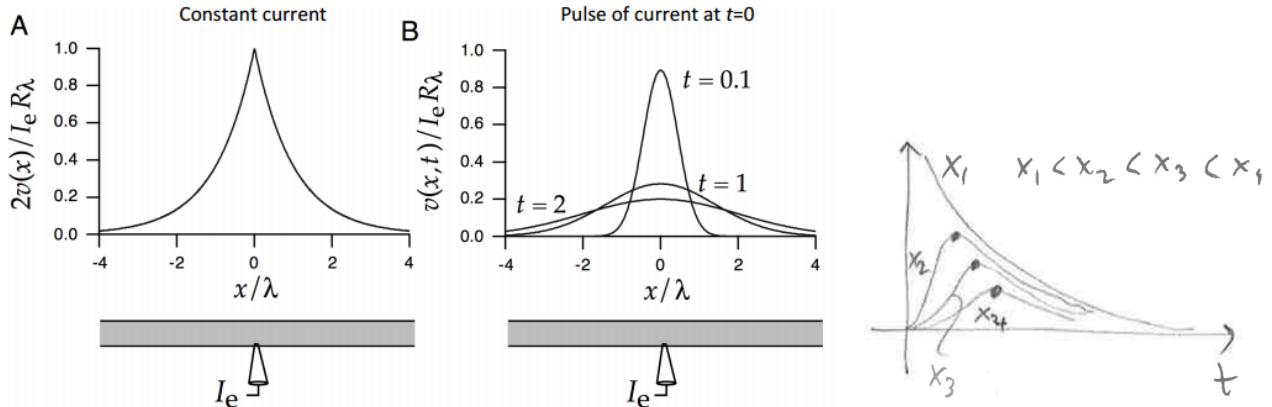


Figure 16: Left: constant current by location; center: pulse of current by location; right: pulse of current by time.

**Infinite cable, constant  $I_e$  (not time varying) and steady state solution  $V = V(x)$  for  $t \rightarrow \infty$**   
for  $\frac{\partial V}{\partial t} = 0 \rightarrow V_\infty(x) = \frac{I_e R_\lambda}{2} \exp(-\frac{|x|}{\lambda})$

## Consequences

- If  $I_e$  injected at  $x = 0$  in dendrite (e.g., through synapse)
  - $V(x)$  decays exponentially with distance
  - $V(x)$  reduced to  $\frac{1}{e}$  at  $x = \lambda$  compared to  $V(x = 0)$
- dendrites have effective length close to  $\lambda$  (or in that order of magnitude)
  - strong alternation of  $V(x)$  from distal dendrites to soma.
- increase in  $\lambda$ :
  - $r_m$  increases: less leak current  $I_m$
  - $r_L$  decreases: less intracellular resistance
  - $a$  increases: larger cable

**Finite cable - dendrites** length of the cable decreases  $\rightarrow$  closer to isopotential.

**Infinite cable, pulse of input current ( $I_e = \delta x \delta t$ )** At each time  $t$ ,  $V(x, t)$  is a gaussian in  $x$ . The width of the gaussian is  $r = \lambda \sqrt{\frac{t}{\tau_m}}$ , where  $\lambda$  is a spatial scale and  $\tau_m$  is a temporal scale. The area of the gaussian is given approximately by  $e^{-\frac{t}{\tau_m}}$ . The area decrease with time, some charge is lost through  $i_m$ .

Plotting  $V$  as a function of time (16) for different  $x$ :

- time of max  $V$  changes with  $x$
- width of peak changes with  $x$
- by looking at the peak and the time occurrence of the peak at the soma one can infer about how far from the dendrite the current was injected.
- "speed" of the bump:  $V_{bump} \approx 2 \cdot \frac{\lambda}{\tau_m}$ .
  - this approximation also holds for action potentials:  $V_{ap} \approx 0.25 - 100 mV$
  - if  $\lambda$  increases,  $V$  goes far, thus  $V_{bump}$  increases
  - if  $\tau_m$  increases,  $V$  changes slowly, thus  $V_{bump}$  decreases

How to increase  $V_{bump}$ ?

- by increasing  $a$ : giant axon
- myelin:  $r_m$  increases and  $c_m$  decreases.

$$V_{bump} = \sqrt{\frac{1}{2r_m r_L}} \frac{1}{c_m}$$

$$\bullet v(x) = \frac{I_e R_\lambda}{2} \exp\left(-\frac{|x|}{\lambda}\right)$$

$$\bullet R_\lambda = \frac{r_m}{2\pi a \lambda} = \frac{r_L \lambda}{\pi a^2}$$

•  $\lambda = \sqrt{\frac{a r_m}{2 r_L}}$  sets the scale for the spatial variation in the membrane potential.

•  $\lambda$  is the electronic length, or how far the signal travels.

•  $\tau_m = r_m c_m$  sets the scale for the temporal variation in the membrane potential.

$$\bullet v(x, t) = \frac{I_e R_\lambda}{\sqrt{4\pi \lambda^2 t / \tau_m}} \exp\left(-\frac{\tau_m x^2}{4\lambda^2 t}\right) \exp\left(-\frac{t}{\tau_m}\right)$$

•  $a$  is the radius of the axon, about  $2 \mu m$

•  $r_m$  is the specific membrane resistance, about  $1 M\Omega \cdot mm^2$

•  $v = V - V_{rest}$

•  $r_L$  is the longitudinal resistance, about  $1 k\Omega \cdot mm$

•  $I_e$  is the injected current.

• It follows, that increasing  $R_m$  also increases  $\lambda$ . With better isolation, signals travel further.

• Increasing the diameter also increases  $\lambda$ .

## 4.6 Level of approximation

- A neuron can be represented by a variable number of discrete compartments.
- Compartments represent a region, each with a single membrane potential.
- The connections between compartments have resistive couplings.
- The basic model is the single compartment (isopotential)
- There is also a model for selective ion channels: drive  $V$  towards reversal potential ( $E$ ) for that channel type

## 5 Action Potential (AP)

- Occurs in **axons**
- **All-or-none**, i.e., stereotyped (if  $I_e$  increases it will lead to an AP initiated sooner, but with the same shape. It is non linear, unlike passive membrane).
- It travels down the axon (if initiated close to the soma, i.e., in axon initial segment).
- Hodgkin-Huxley: Nobel Prize Medicine Physiology in 1963. Everything they did was before the existence of ion-channel was known.
- Refraction period ensures maximum frequency.
- Action potential is about 1 to 10 m/s fast.
- It would take 100 years to go through all axons of the human brain in a serial fashion.

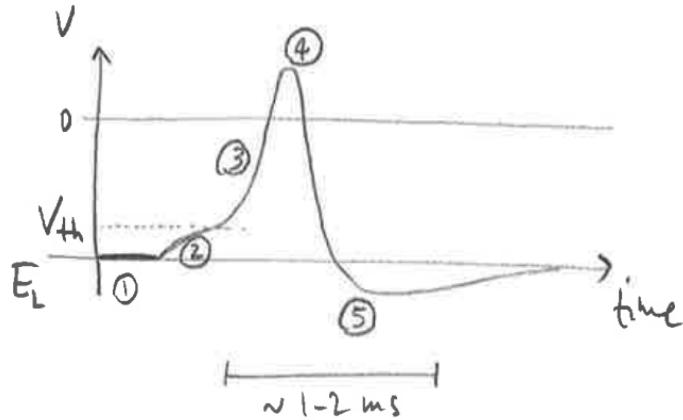


Figure 17: 1: at rest,  $V = E_L$  (resting potential); 2: current injected ( $I_e$ ) leads to an increase in  $V$  exponentially, for small  $I_e$  it doesn't reach  $V_{th}$ . Large  $I_e$  crosses  $V_{th}$ ; 3 If  $V > V_{th}$  a fast depolarization happens, i.e., an AP; 4 overshoot  $V > 0$  (around 30mV); 5: hyperpolarization and undershoot ( $V < E_L$ ).

During an AP we see channels opening and pulling  $V$  towards  $E$ . The hypotheses is that in the rising phase of AP the sodium and calcium conductances increase ( $g_{Na}$  and  $g_{Ca}$ ), and in the decaying phase of AP the sodium and calcium conductances decrease or potassium and chloride conductances increase. All as function of  $V$ . For testing this hypotheses, we need to measure  $g_{Na}, g_K$  etc. We can use the  $IV$ -relation: measuring  $I_{Na}, I_K$  etc for different  $V$  then infer  $g_{Na}, g_K$ , etc. To do this, we can use voltage clamp.

### 5.1 Voltage clamp

A new technique invented by Hodgkin and Huxley. Previously, current  $I_e$  was injected and voltage  $V$  was measured, now set  $V$  and measure  $I_e$  required to keep  $V_{measured} = V_{set}$ . It measures the current required to clamp the membrane voltage. Fast feedback system to fix  $V$  and measure  $I$ .

$I_e$  has opposite sign, i.e., is positive if from outside to inside. But to keep the  $\Delta V$  constant, it is necessary to inject a current opposite to the ionic current. In the end, the current injected can be read as the ionic current (in the ionic current convention).

**Space clamp** Makes the axon isopotential, do not have an AP but it is the same mechanism.

The giant axon in squid has approximately 1mm of diameter and it is like a long wire, making the axon isopotential.

### 5.1.1 Voltage clamp experiment

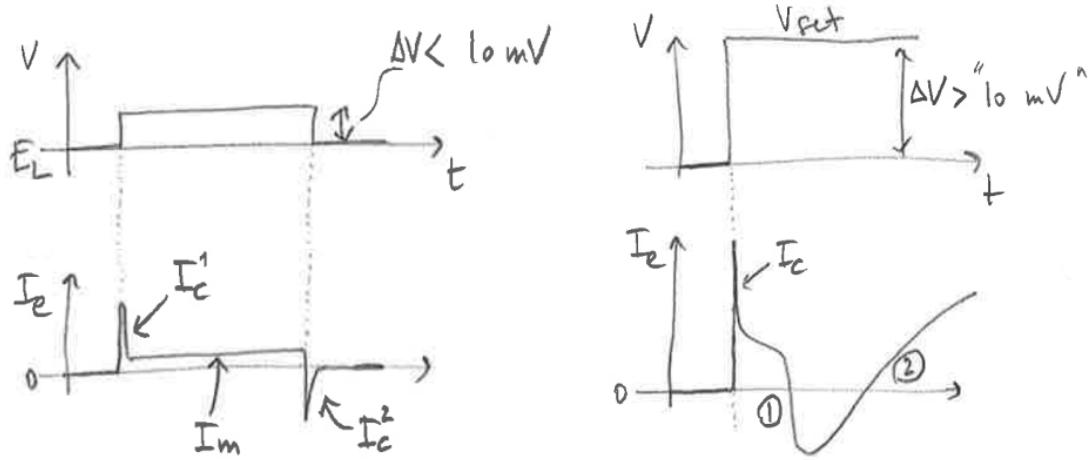
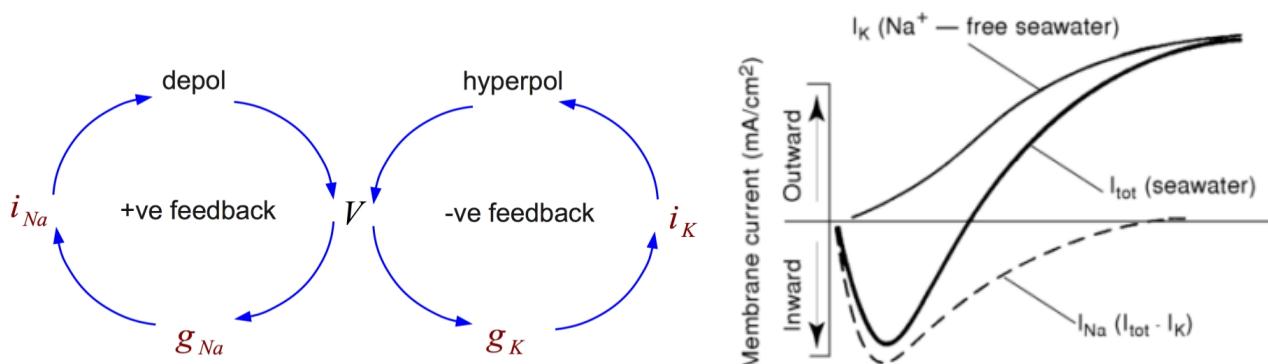


Figure 18: On left, small voltage is applied: passive/linear membrane.  $I_c^1$  depolarizes membrane (add + charge to inside).  $I_m$  compensates for leak current  $I_L$  ( $V \neq E_L$ : cell wants to go bad to  $E_L$ ).  $I_c^2$  repolarize the membrane, i.e., remove charged added in  $I_c^1$ ; On right, large  $V$  is applied ( $V > 10 \text{ mV}$ , i.e.,  $V > V_{th}$ ): active, non linear membrane. In 1 cell wants to depolarize, there is a need to inject negative  $I_e$  to keep  $V = V_{set}$ .  $I_{Na}, E_{Na} \approx 50 \text{ mV}$ ; In 2 cell wants to hyperpolarize, there is a need to inject positive  $I_e$  to keep  $V = V_{set}$ .  $I_K, E_K \approx -80 \text{ mV}$ .

- 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.
- With negative feedback circuit, the  $\text{Na}^+$  current is auto-catalytic. An increase in the voltage increases conductance, which increases the  $\text{Na}^+$  current, which increases the voltage again.
- The threshold for action potential initiation is where the inward  $\text{Na}^+$  current exactly balances the outward  $\text{K}^+$  current.



### 5.1.2 Identifying the nature of the current

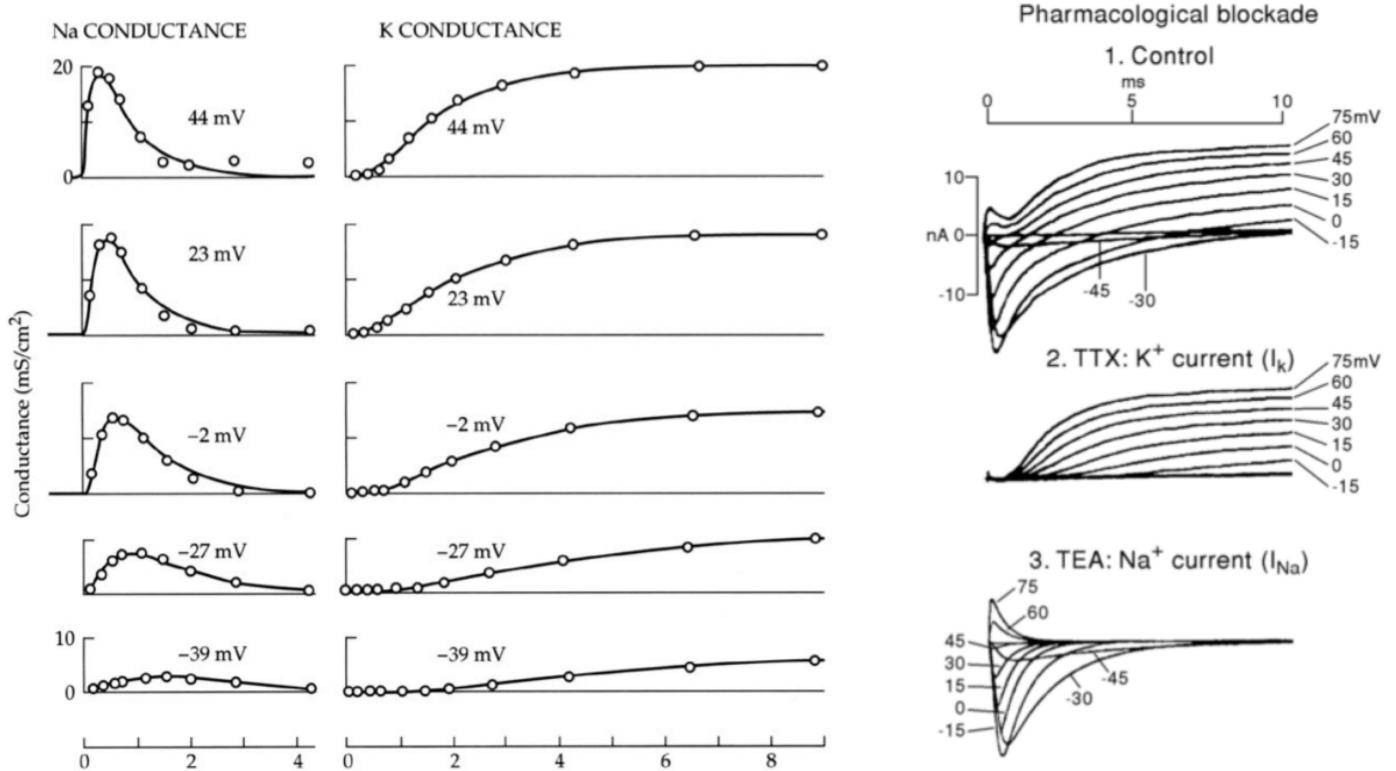
**H and H: remove  $\text{Na}$  gradients** They replaced extracellular medium with a solution that has 10%  $\text{Na}^+$  compared to seawater. This eliminates  $I_{Na}$  for  $\Delta V \approx 50 \text{ mV}$  ( $[Na]_{in} = [Na]_{ou}$ ). By repeating the experiment for many  $V_{set}$  (i.e.,  $\Delta V$ ) they get  $I_{Na}(V)$  and  $I_K(V)$ .  $I_{Na} = I_{seawater} - I_K$ . get  $g_{Na}(V)$  and  $g_K(V)$  with  $I_{Na} = g_{Na}(V - E_{Na})$  and  $I_K = g_K(V - E_K)$ .

**Later: removed intracellular  $K^+$**  It confirmed  $I_K$  and H and H predictions.

**Pharmacology** Uses pharmacological blockages.

**TTX** poison in pufferfish, it eliminates  $I_{Na}$ .

**TEA** eliminates  $I_K$ .



Voltage and time dependent conductances for  $g_{Na}, g_K$ :  $g_{Na}$  increases quickly (fast activation), but then inactivation kicks in and it decreases again (fast inactivation).  $g_K$  increases more slowly (slow activation), and only decreases once the voltage has decreased (no inactivation).

### 5.1.3 Voltage and time dependent conductances

Two possibilities that cause conductance to be voltage and time dependent:

1. Single channels have variable permeability (analog)
2. Single channels are either opened or closed (digital)  
today we know, this is the correct option

## 5.2 Patch clamp

- Technique that allows recording of current  $I$  through a single channel
- Neher and Sakmann - Nobel prize in Medicine and Physiology in 1991
- H and H inferred it from their Voltage clamp data
- Individual channels are probabilistic devices that are opened or closed. The conductance is measured by the average of all channels.

There are two types of voltage-dependent conductances:

**Persistent conductance type** It has two stages: deactivated (closed) and activated (opened). The channel opens and stay opened when the cell is depolarized. For example,  $g_K$  in AP.

**Transient conductance type** It has three stages: deactivated, activated and inactivated. Here we have two gating variables that describe the opening and closing of the channel. Activation and Inactivation are two process that work in opposite directions. The channel opens but then it closes while the cell is still depolarized. For example,  $g_{Na}$  in AP.

**H and H formalism** It is used for active conductances in general.

$$g_i = \bar{g}_i \cdot P_i$$

where  $g_i$  is the overall conductance of channels of type  $i$ ;  $\bar{g}_i$  is the maximal conductance (if all the channels were open); and  $P_i$  is the probability of the channel be open (or the fraction of channels that are open).

### 5.2.1 Persistent conductance

Assume that  $k$  events (independents and identical) are necessary to open a single channel, then  $P = n^k$ .  $n$  is a gating/activation variable: the probability of a subunit gate being open, and it is voltage and time dependent.  $k$  is the number of subunits necessary to open each channel.

**H and H**  $g_K = \bar{g}_K \cdot P_K = \bar{g}_K \cdot n^4 = n \cdot n \cdot n \cdot n$  (it is necessary 4 subunits to open the channel). When  $k$  was fitted to data it leaded to corrected predictions for  $K^+$  channels.

**Time dependence** Assume closed  $\rightarrow$  opened and opened  $\rightarrow$  closed transitions such that:

$$\underbrace{\Delta n}_{\text{change in the number of open channels in time } t} = \underbrace{\alpha_n(V)}_{\text{rate of opening}} \cdot \underbrace{(1 - n)}_{\text{fraction of closed}} \cdot \underbrace{\Delta t}_{\text{time}} - \underbrace{\beta_n(V)}_{\text{rate of closing}} \cdot \underbrace{n}_{\text{fraction of opened}} \cdot \Delta t$$

If channels closed  $\rightarrow$  open, then  $\Delta n > 0$ , open  $\rightarrow$  closed, then  $\Delta n < 0$ .

At  $\Delta t = 0$ ,  $\frac{dn}{dt} = \alpha_n(V)(1 - n(t)) - \beta_n(V)n(t)$ ,  $\alpha_n$  and  $\beta_n$  are not time dependent.

$\rightarrow \tau_n(V) \cdot \frac{dn}{dt} = n_\infty(V) - n$ , where  $\tau_n(V) = \frac{1}{\alpha_n(V) + \beta_n(V)}$  and  $n_\infty(V) = \frac{\alpha_n(V)}{\alpha_n(V) + \beta_n(V)}$ .  $0 \leq n_\infty(V) \geq 1$ . For  $\tau_n(V)$  we have that faster transitions leads to faster change in  $n$ .

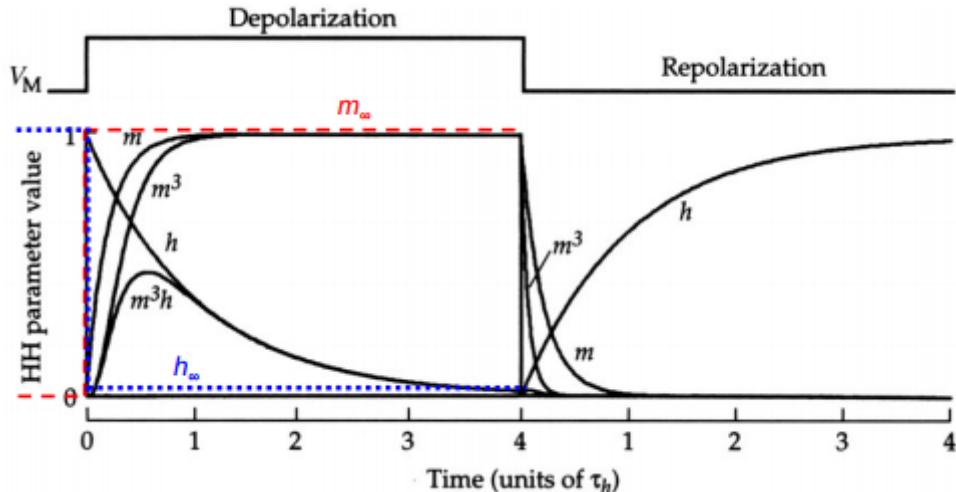


Figure 21: Gating variables: time dependence

**Voltage dependence** How to determine the values of  $\alpha_n(V)$  and  $\beta_n(V)$ ? In practice:  $\alpha_n(V)$  and  $\beta_n(V)$  are fitted to data (i.e.,  $g_K(V, t)$ ).  $\alpha_n(V)$  and  $\beta_n(V)$  are approximately exponential because the energy barriers that need to be overcome by the gating change.

In this case,  $\alpha_n(V) \approx A_\alpha \cdot e^{-\frac{E_\alpha}{k_B T}}$  and  $\beta_n(V) \approx A_\beta \cdot e^{-\frac{E_\beta}{k_B T}}$

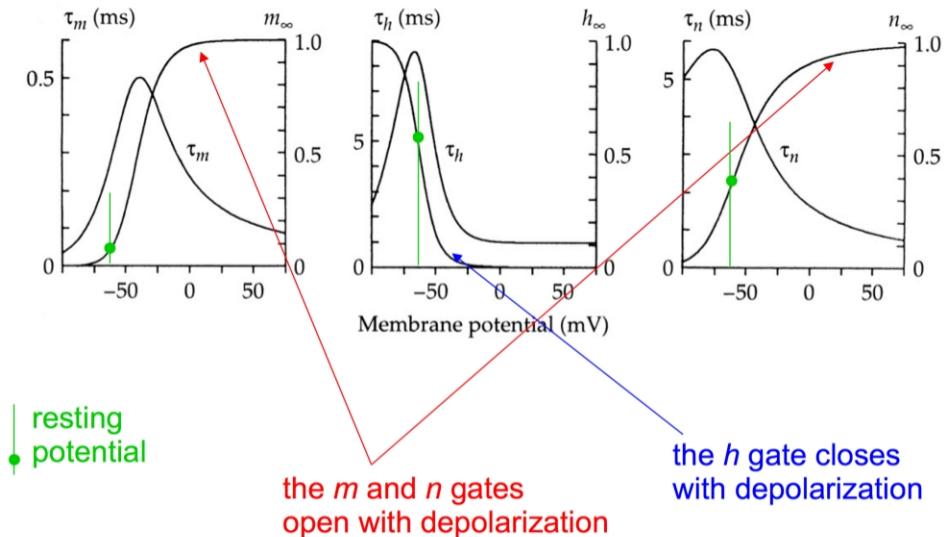


Figure 22: Gating variable: voltage dependence

### 5.2.2 Transient conductance

Similar, but includes inactivation:  $P_{Na} = \underbrace{m^3}_{\text{activation variable}} \cdot \underbrace{h}_{\text{inactivation variable}}$ .  $h$  is the probability that channel is not blocked by inactivation gate.

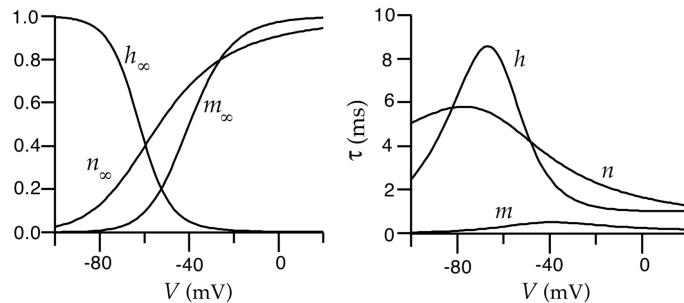
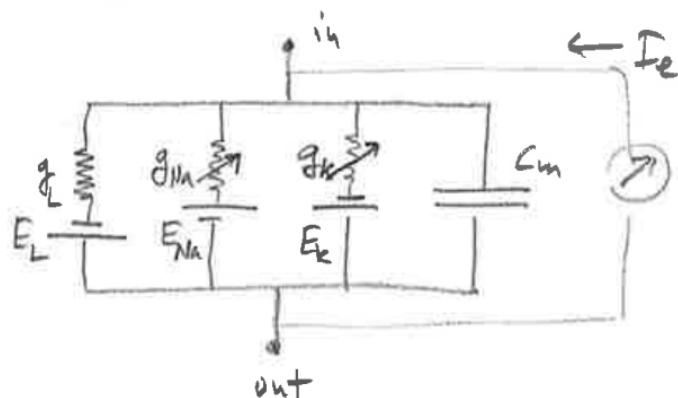


Figure 23:  $m$  is fast,  $h$  and  $n$  are slow.  $m$  and  $h$  have opposite  $V$ -dependence.

## 5.3 Full Hodgkin-Huxley Model



- Model that describes how action potentials in neurons are initiated and propagated. model parameters are fit to  $g_{Na}(V, t)$  and  $g_K(V, t)$  from voltage clamp.
- $n$  and  $m$  are probabilities for a gate to be open.

- $h$  is the probability that an open channel is not blocked.
- The gating variables have a voltage dependence.
- $\bar{g}$  values are the maximum conductance possible.
- There is no inactivation for potassium, only for sodium.
- The membrane does not get locked at positive values.
- $\bar{g}_L$  stands for some generic leak.
- The functions  $n_\infty(V)$ ,  $m_\infty(V)$  and  $h_\infty(V)$  determine whether gates serve to activate channels (with depolarization) or inactivate the channel (close with depolarization).  $\tau_m$ ,  $\tau_h$  and  $\tau_n$  are time constants.

$$C \frac{dV}{dt} + \bar{g}_K n^4 (V - V_K) + \bar{g}_{Na} m^3 h (V - V_{Na}) + \bar{g}_L (V - V_L) + I_{inj} = 0$$

**Some model parameters**  $\bar{g}_L = 0.003 \text{mS/mm}^2$ ,  $E_L = -54 \text{mV}$

$\bar{g}_K = 0.036 \text{mS/mm}^2$ ,  $E_K = -77 \text{mV}$

$\bar{g}_{Na} = 1.2 \text{mS/mm}^2$ ,  $E_{Na} = +50 \text{mV}$

here,  $S$  stands for Siemens =  $\frac{1}{\text{ohm}}$

## Model predictions

### AP-shape

**AP-threshold**  $n_\infty$ ,  $m_\infty$  and  $h_\infty$  are all  $> 0$  for  $V = E_L (= V_{rest})$ . That means that  $g_{Na}$  is already open at rest. Why we have no AP at rest?  $I_L$  and  $I_K$  are larger than  $I_{Na}$  for  $V < V_{th}$ . At  $V_{th} : |I_L + I_K| = |I_{Na}|$ .

**Refractory period** It is harder (requires larger current injection) to generate AP immediately after an AP. The reason is that  $g_K$  is still activated and  $g_{Na}$  is still inactivated.

### AP propagation in unmyelinated axon

**AP propagation in myelinated axon** In the myelinated part of the axon we have passive AP propagation (small capacitance and large resistance), but in the nodes of ranvier, we have active AP regeneration. Compared to unmyelinated:

- faster AP propagation
- smaller current
- faster  $V_{AP}$  increase with axon radius

**AP propagates in one direction along axon** Reason: refractory period,  $g_{Na}$  still inactivated in the wake of AP. Either direction is possible in principle. From soma to axon terminal: orthodromic. In the opposite direction: antidromic. In the brain we do not have usually antidromic AP. Antidromic AP can be generated artificially. With a collision experiment, i.e., both antidromic and orthodromic AP initiated, none achieve the other end, they annihilate each other.

**AP not reflected at axon terminal** At the end of the cable, there is no AP reflected because refractory period.

**AP does not usually propagate in dendrites** Because  $g_{Na}$  is missing. However, in a few cell types  $g_{Na}$  is present also in dendrites. It is not sufficient to generate an AP, but can propagate AP from soma into dendrite to some extent: axon backpropagation.

**Number of subunits in  $K$  channel** It was verified much later with structural studies.

## 6 Neural Coding

The problem we are trying to elucidate with neural coding is “the representation and transformation of information in the nervous system”. To better understand this, we need to work with four concepts:

**Correspondence** a code is the correspondence between two domains. This mean that one domain (for instance, visual signal) can be specified (encoded) by another (for instance, spike trains), this way one can, theoretically, reconstruct the original message from the encoded message with some accuracy (decoding).

**Representation** not all cases of correlation are considered as a instance of coding.

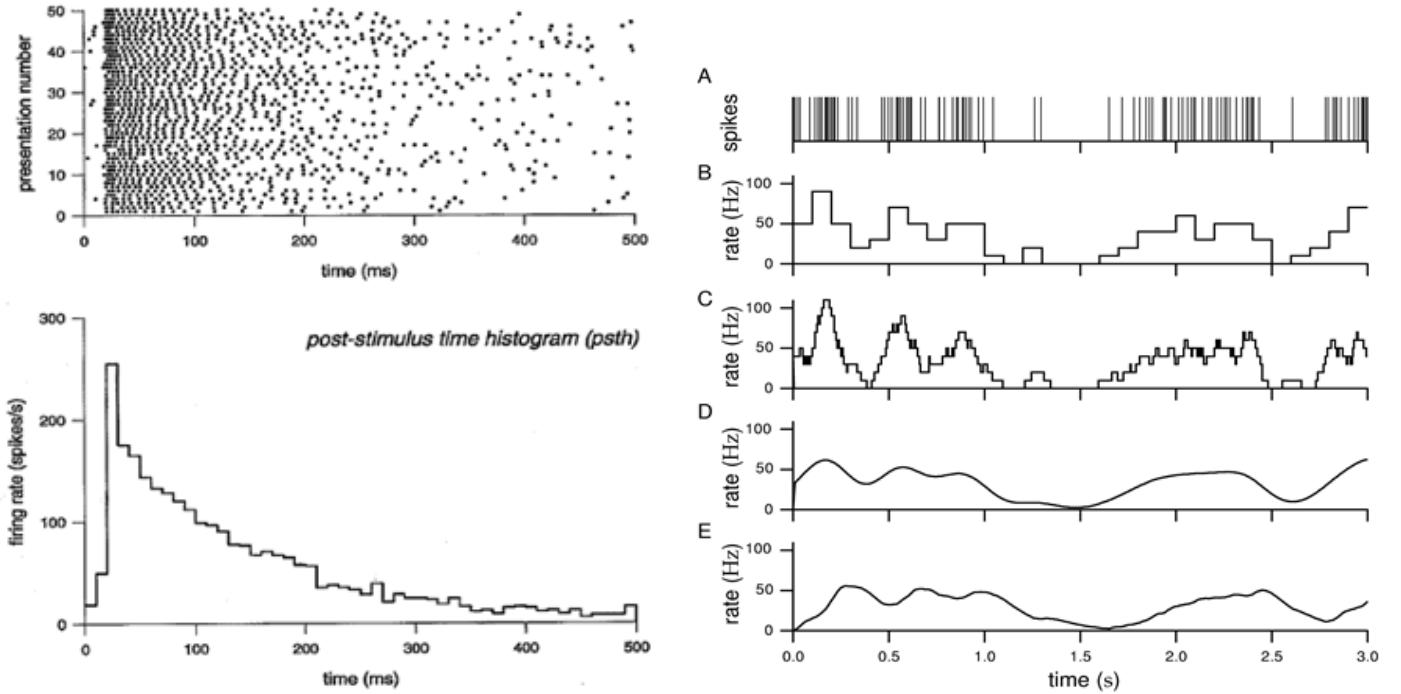
**Causality** Although spike trains encode visual signal, we wouldn't say that visual signals encode spike trains. This is because we assume a causal relation: visual signals generate (cause) spike trains.

**Encoding** How does a stimulus cause a pattern of responses? Building an approximate mechanistic model of the world.

**Decoding** What do the responses tell us about the stimulus? How can we reconstruct the stimulus? By recording neuronal responses in the cat visual cortex, we identify that the visual cortex encodes the orientation of a moving stimulus and has a orientation specific organization. In rats, spatial information is coded via hippocampal place cells.

### General

- Information is encoded by firing of single neurons and firing of populations of neurons.
- A neuron encodes information, fires to stimuli.
- Firing rate and spike timing encodes information.
- Spatial/temporal resolution of different measurement techniques tell us about the neural code.
- It is an issue to record from many neurons simultaneously.
- There is not much information in the slope of a spike.
- By recording neuronal responses from a stimuli, we can “see” how the brain encodes the stimuli.



- Raster plot with spikes and histogram. Stimulus takes place at  $t = 0$ .
- Use bins (about 100 ms in time) with a sliding window and a gauss filter and causal filter.
- Problematic are intermediate stages, probability of firing, background activity and varying membrane potentials.
- Causal filter is  $w(\tau) = [\alpha^2 \tau \exp(-\alpha\tau)]$
- Typically neurons fire between 1 Hz and 200 Hz - often around 40 Hz.

## 6.1 Neuronal Rate Codes

Rate coding refers to information being carried by the firing rate. It is often argued, or assumed, that firing rate captures essentially all relevant information.

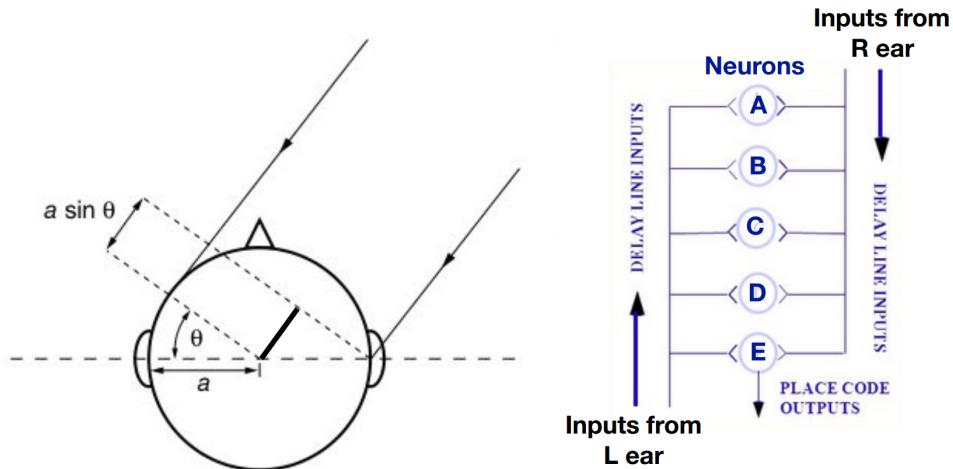
- Rate = average over time, single neuron, single run.
- $v = \frac{n_{sp}}{T}$
- Definition of the mean firing rate via a temporal average.
- Neuronal gain function (curve). The output spike rate is given as a function of the total somatic input current  $I_0$ .
- Easy to understand, but no timing effects and misleading as more than one stimulus might be encoded.
- It takes time to compute a temporal average and behavioral response time is shorter than integration time.

## 6.2 Neuronal Temporal Codes

Temporal coding may refer to several quite different ideas:

- Much of the information may be transmitted by a neuron during certain small intervals of time.
- synchronous, or what I would call quasi-synchronous, firing of neurons within and across ensembles may carry important information.
- the precise timing, or pattern, of spikes may carry information.

## 6.3 Sound localization by measuring the Interaural Time Difference (ITD)

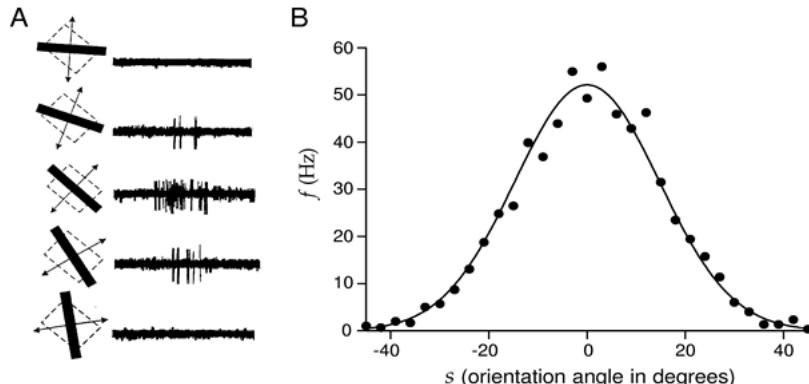


## 6.4 Peri-Stimulus Time Histogram (PSTH)

- $\rho = \frac{1}{\Delta t K} n_K(t; t + \Delta t)$  where  $K$  is the number of trials.
- Spike density is an average over several runs of the experiment.

## 6.5 Tuning Curves

- Tuning curves show average firing rate response to varying stimulus parameters.



## 6.6 Orientation Maps

- Nearby neurons have similar preferred orientations.
- Orientation-selective neurons (in the primary visual cortex of cats).
- Orientation column and pinwheels.

## 6.7 Poisson Spike Trains

- Mathematical model to describe and generate spike trains.
- Poisson distribution for the number of spikes in interval  $T$  with firing rate  $r$ .
- $P_T(n) = \frac{(rT)^n}{n!} \exp(-rT)$
- Homogeneous: Constant rate.
- Inhomogeneous: Variable rate.
- Approximation: Probability of a spike occurring in short interval of length  $\Delta t$ :  $r(t) \cdot \Delta t$

## 6.8 What a single neuron can encode

- Places (on entering a particular region).
- Grids (regularly arranged triangular grid of locations).
- Head-direction, compass-like.
- Single cells that respond to only one person.

## 6.9 Population Rates

- Rate = average over pool of equivalent neurons (several neurons, single run).
- Activity  $A = \frac{1}{\Delta t} \frac{n_{act}(t:t+\Delta t)}{N}$
- A postsynaptic neuron receives spike input from a population  $m$  with activity  $A_m$ . The population activity is defined as the fraction of neurons that are active in a short interval  $[t, t + \Delta t]$  divided by  $\Delta t$ .

## 6.10 Population Codes

- Different cells encode different ranges of the stimulus.
- Averaging over a population often meaningless.
- Allows accurate reconstruction of the signal, also interpolated between peaks.
- Sparse coding: Only few cells are activated.
- Retina as an example: Different cells for different light wavelengths.
- a neuron encodes a stimulus, a neuronal population encodes behavior.

### 6.10.1 Population Vector Code

- Population of neurons with different preferred arm movement directions.
- Encoded direction (arrow) corresponds to vectorial addition, weighted by firing rate.
- Interesting for brain-computer interfaces.

### 6.10.2 Measuring Population Activity in vivo

- calcium imaging
- fmri

### 6.10.3 Taking multiple stimuli into account

Repeatedly sample responses to a variety of stimuli so that we can characterize what feature combination triggers a spike or a behavior.

$$P(\text{response}|\text{stimulus}) = P(\text{response}|s_1, s_2, s_3, \dots, s_n)$$

. After collecting data, if we don't have any labels for the stimuli, we use an unsupervised/clustering approach, otherwise a supervised approach to identify the characteristics that trigger a behavior.

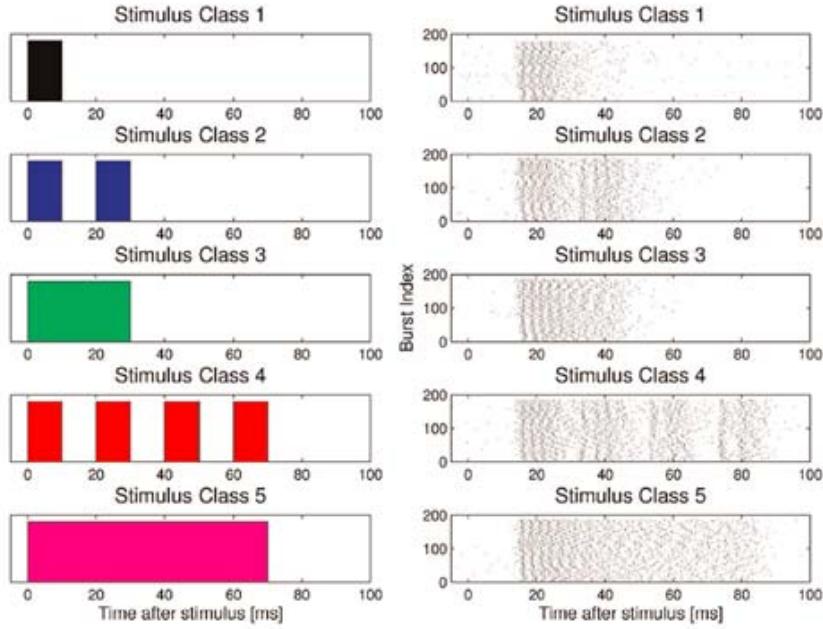
## 6.11 Neuronal Event Codes

### 6.11.1 Time-to-first Spike Codes

- High rate implies fast firing.
- Can implement competition among different input cells.
- Can be extended to rank-order codes (firing sequence of different neurons).
- Is very fast and efficient, has evidence in auditory, visual and somatosensory systems.
- Susceptible to noise, requires a reference signal.

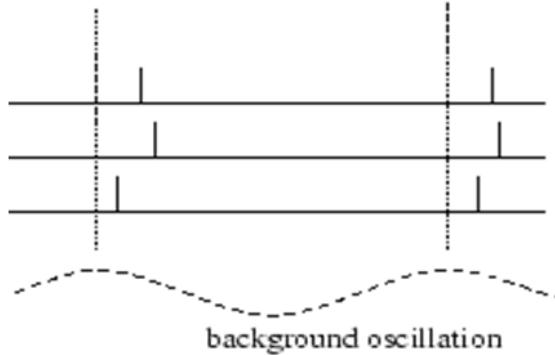
### 6.11.2 Burst- and Temporal Codes

- Bush-cricket auditory neurons in natural environment.
- Preserve very high coding precision in extreme noise.



### 6.11.3 Oscillations and Phase Coding

- Phase: The neurons fire at different phases with respect to the background oscillation.
- Phase could code relevant information.



### 6.11.4 Coding by Synchrony

- Synchrony can encode information.
- Neurons can fire (nearly) synchronous.

## 6.12 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.
- Spike sorting: Assigning spikes to different neurons from extracellular signal (spike shapes are unique for each neuron).

## 6.13 fMRI

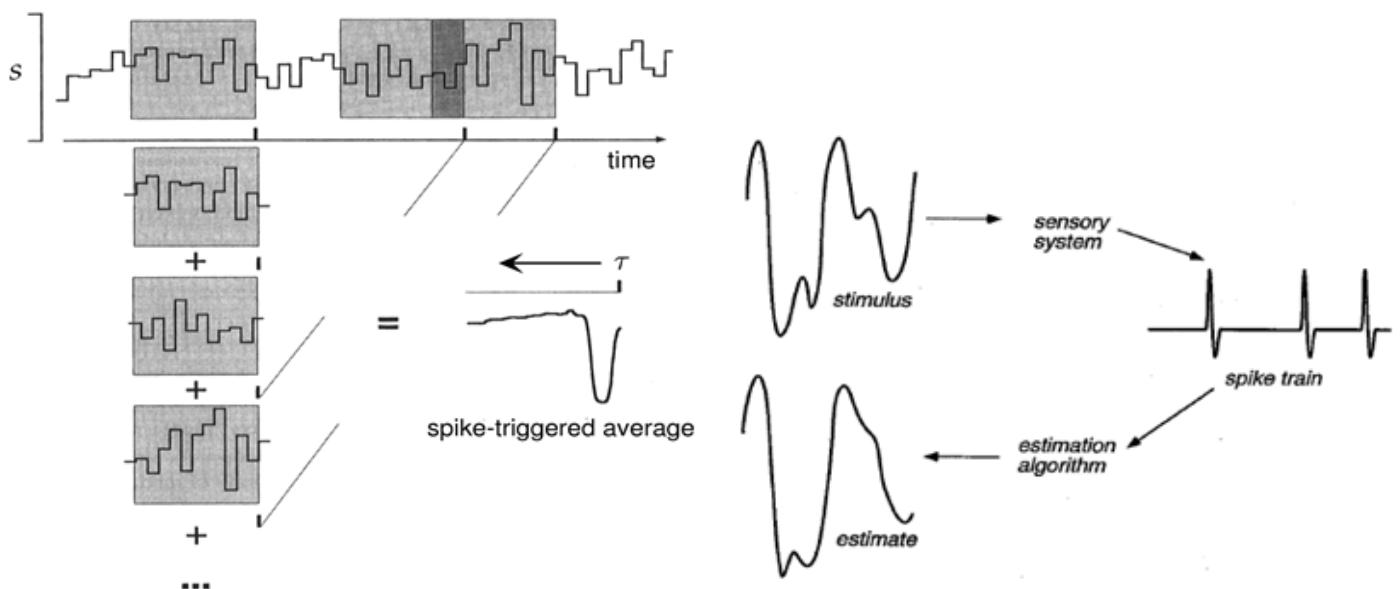
- Functional magnetoc resonance imaging.
- Non-invasive technique for monitoring brain function.
- Based on BOLD (blood oxygenation level dependent signal change). Haemodynamic response function (HRF).
- Slow temporal resolution.

## 6.14 Binding Problem

- Occurs frequently: Visual processing (what, where).
- Potential mechanism: Temporal synchrony, hierarchical coding, population coding.

## 6.15 Averages

- Spike triggered average: Average over stimulus in short time window before spike.
- Sensory neurons typically respond stronger to rapid changes in stimulus properties.



## 6.16 Stimulus Reconstruction

- Allows an observer to reconstruct the stimulus from spike trains.
- Probability and information theory is the mathematical background.
- Whole stimulus reconstruction may not be relevant.
- Evolution may have shaped us to encode particular features better than others, for example 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.

# 7 Synapse

## 7.1 Introduction

- First mentioned by Sherrington (1873).
- Cajal's golgi staining methods suggested the presence of contacts between cells that were used for communication 1900's.
- Otto Loewi experimented with the vagus nerve 26.  
Stimulating the vagus nerve slows down the heart beat, has an inhibitory function.

Ringers solution is a mixture of chemicals in which the heart can continue beating.

When switching the solution with one that has been used with an activated vagus nerve, the heart will slow down.

- It was found that the “Vagusstoff” is acetylcholine (ACh).
- The synapses are receptive for nicotine, muscarine and acetylcholine, because of ACh-receptors. This makes certain substances very addictive.
- Residual acetylcholine has to be cleared and removed immediately. This happens with acetylcholine esterase enzymes.

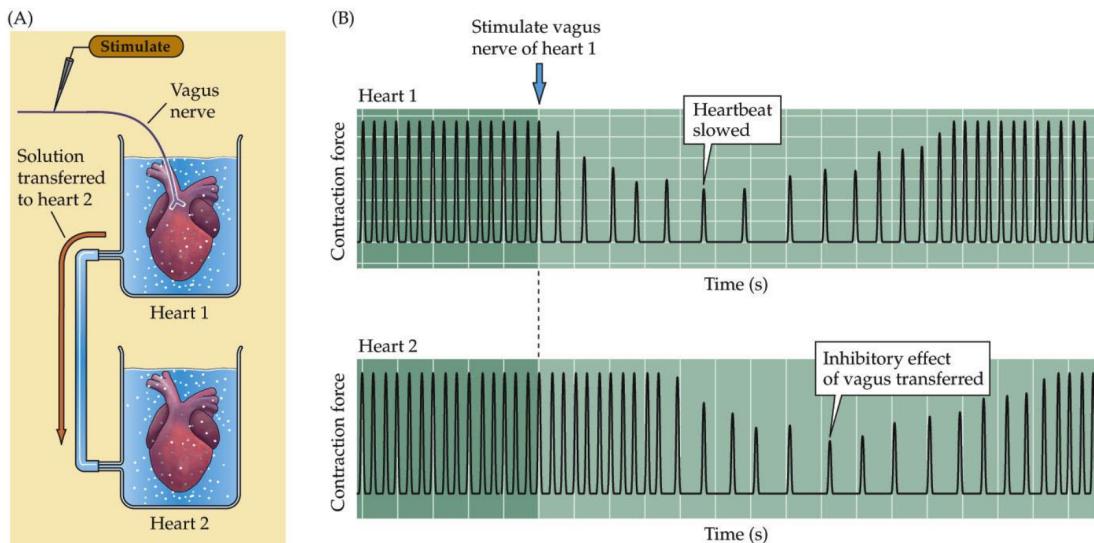


Figure 26: Otto Loewi vagus experiment. Two solutions connected, one heart in each. After stimulation of one of them, the other one, after a short time, has the same effect.

### 7.1.1 Soup vs Spark - Chemical or Electrical?

Evidence for chemical transmission at the neuro muscular junction was widely accepted by neuropharmacologists. However, some physiologists thought that certain aspects were too fast to be mediated chemically. Chemical synapses is the predominant way of communication between neurons, but there are some electrical synapses.

## 7.2 Chemical transmission

Communication between cells which involves the rapid release and diffusion of a substance to another cell where it binds to a receptor (at a localized site) resulting in a change in the postsynaptic cells properties.

- synaptic cleft: 20 to 40nm.
- vesicles in presynaptic terminal
- neurotransmitters ( $> 1000$  per vesicle)
- neurotransmitters released by depolarization,  $Ca^{2+}$  dependent
- vesicles are released by exocytose in the active zone (specialized site in presynaptic neuron for release).
- diffusion
- binding to receptors
- channel opening

- amplification
- Multiple steps are required to release transmitter chemicals and for them to act on postsynaptic receptors, resulting in a time delay (can be as short as  $0.2ms$ , from  $Ca^{2+}$  entry to secretion).
- Directional, select localization of release machinery to presynaptic terminals and receptors to postsynaptic specializations.
- Can change sign by release of inhibitory transmitter.

### 7.2.1 Steps of transmission

- Action potential generated and reach the axon of presynaptic cell
- Opening of voltage gate  $Ca^{2+}$  channels
- Diffusion and action of  $Ca^{2+}$  at release machinery
- Exocitosys and diffusion of transmitter in cleft
- Activation of post-synaptic cell

**Model of synaptic transmission** Neurotransmitter is released in discrete packages, or quanta.  
 Quantal size: size of individual quanta. Quantal content: number of quanta released.

- One package of neurotransmitter = one quantum
- AP transiently increases the probability of releasing  $NX$  quanta.
- Several quanta are available to be released
- Each quantum give approximately the same postsynaptic response: quantal amplitude.
- the average number of quanta released:  $m = np$   
 $n$  is the number of quanta available for release  
 $p$  their average release probability
- Probability that  $x$  units successfully contribute is given by:  $P(success = x) = \binom{n}{x} p^x (1-p)^{n-x}$

### 7.2.2 Synapse properties

- Only vesicles which are already on the presynaptic membrane (docked) will be released after the AP, not all of them.
- One single synapse produces only a small potential. More are needed for an actual action potential.
- Release of neurotransmitters is calcium dependent.
- Probabilistic release of neurotransmitter.  
 In the CNS, most of the time only one vesicle is released with probability 0.2 to 0.4.  
 An amplitude histogram shows poisson distribution, which gives the probability of firing.  
 The action potential (probability of firing of synapses, probability of postsynaptic receptors to bind neurotransmitter) give the plasticity (overall probability of passing action potential to postsynaptic neuritic changes).
- Single activated synapse is usually not enough. EPSP is about  $0.1\text{ mV}$ .
- The current-voltage lines have bio-measured sigmoid-curves, because channels open with a probability.
- Four types of synapse: axodendritic, axosomatic, axoaxonic and dendrodendritic.

### 7.2.3 Synaptic mechanism

1. Synthesis: Building blocks of transmitter substance gets to the terminal where neurotransmitter is synthesized and packed into vesicles.
2. Release: In response to an AP, the transmitter is released, across the membrane, by exocytosis. The presynapse has voltage-gated  $Ca^{2+}$  channels, which will cause an inflow and trigger vesicle fusion (exocytosis).
3. Receptor activation: The transmitter crosses the synaptic cleft and binds to a receptor.
4. Inactivation: The transmitter is taken back or inactivated in the synaptic cleft.

### 7.2.4 Receptor types

Neurotransmitters cross synaptic-cleft and can bind to two types of receptors: ionotropic (ligand-gated ion channels) or metabotropic (g-protein coupled receptors).

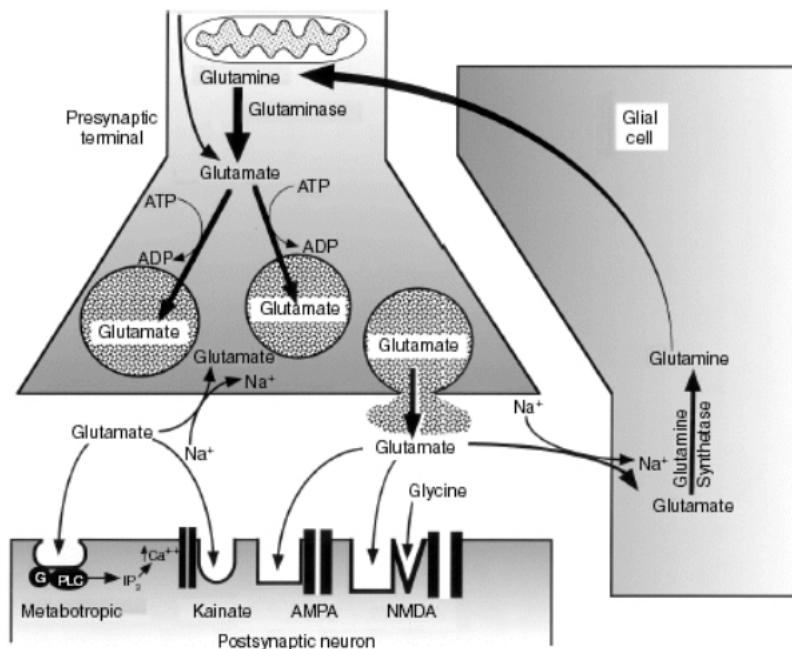
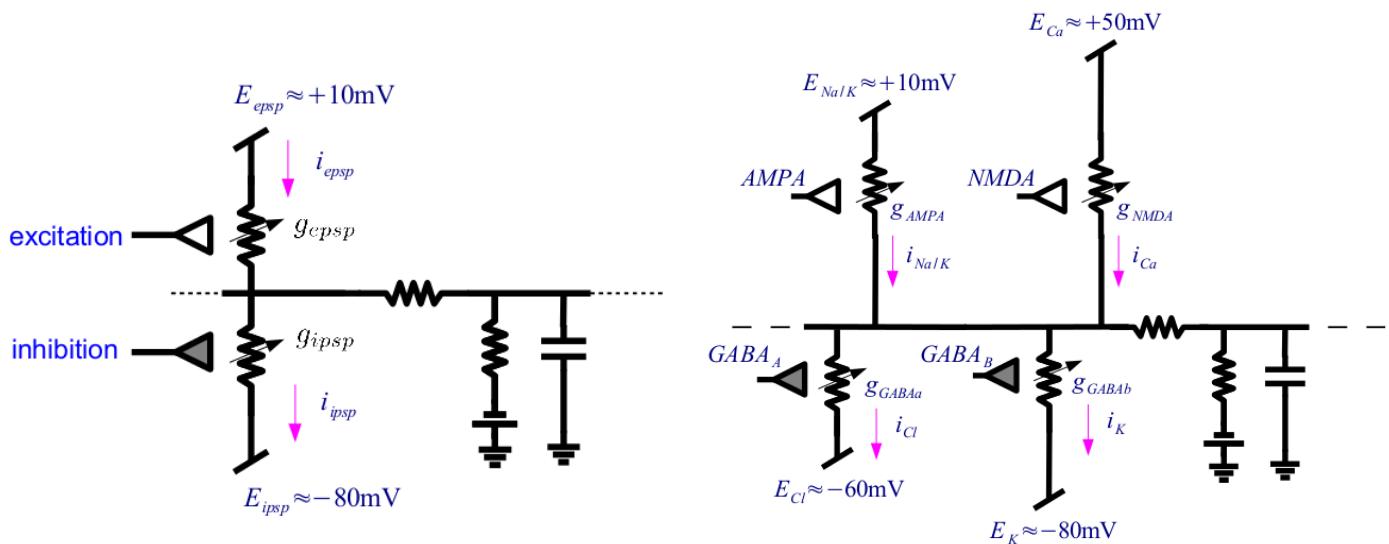
<b>Ionotropic receptor</b>	<b>Metabotropic receptor</b>
Binding site + channel combined	Binding site not associated with channel
Second messenger independent	G-protein or second messenger involved
Short latency action	Longer latency
Rapid response (10 to 50 ms)	Slow responses
Postsynaptic, in general	Pre- and postsynaptic

### 7.2.5 Synapse types

<b>Electrical synapse</b>	<b>Chemical synapse</b>
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
Large synapse	Thousands of small synapses
Limited functions, usually excitatory	Versatile: Excitatory and inhibitory
Synchronized activity	Specific: point to point communication

### 7.2.6 Receptor overview

<b>Receptor</b>	<b>Transmitter</b>	<b>Ions</b>	<b>Approx. <math>E_{rev}</math></b>	<b>Agonist</b>
AMPA	glutamate	Na, K, Ca	+0 mV	AMPA
NMDA	glutamate	Na, K, Ca	+0 mV	NDMA
mGLU	glutamate	G-coupled		
$GABA_A$	gaba	Cl	-65 mV	muscimol
$GABA_B$	gaba	K	-90 mV	



### 7.2.7 Glutamate receptors

- Glutamate enables both synapses, but NMDA is voltage dependent, while AMPA is not.
- The receptors end up being conductive for  $\text{Na}^+$  and  $\text{K}^+$ , as well as  $\text{Ca}^{2+}$ . 10 times more for  $\text{Ca}^{2+}$  than the others.
- $E_S = 0\text{ mV}$
- $\text{Ca}^{2+}$  inflow causes a calcium cascade: Phosphorylation ( $\text{PO}_4$ ) of the channel proteins opens channels even more.

### NMDA

- Voltage dependent.
- The channel is blocked by  $\text{Mg}^+$  below voltages of  $-40\text{ mV}$ .
- The block gets pushed out with more positive voltage.

### AMPA

- Voltage independent.

### 7.2.8 GABA

- GABA-A is for  $Cl^-$ , ionotropic (fast),  $-65\text{ mV}$
- GABA-B is for  $K^+$ , metabotropic (slow),  $-90\text{ mV}$

### 7.2.9 Neuromodulators

- Neurotransmitter that is not reabsorbed by the pre-synaptic neuron or broken down into a metabolite.
- These end up a longer time in the cerebrospinal fluid, modulating the activity of several other neurons in the brain.
- For example norepinephrine, dopamine, serotonin.

## 7.3 Electrical transmission

Electrical synapses are built for speed. Electrical coupling is a way to synchronize neurons with one another (for instance, in heart muscles). In electrical synapses, the postsynaptic neuron starts to change its membrane potential almost instantaneously with the presynaptic neuron.

Rod photoreceptors are connected by electrical synapses.

- simple primitive system, keep firing without refractory period
- often symmetrical, bidirectional
- dendrite gap junctions (connexins)
- very fast, no synaptic delays
- $Ca^{2+}$  independent
- temperature insensitive
- large synapse
- no amplification of the signal
- limited functions, usually excitatory
- synchronized activity

## 8 Modeling Synapses

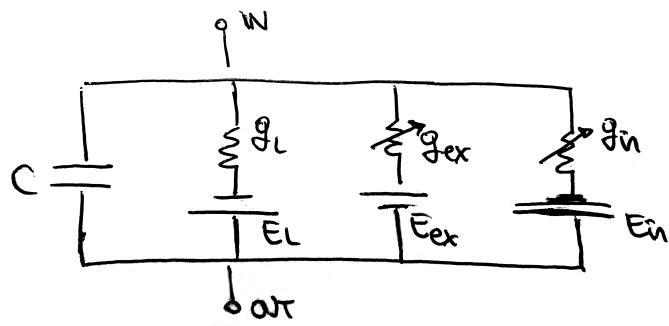


Figure 28: Synapse model

### 8.0.1 $E_{syn}$ and $E_{th}$

- $E_{syn} > E_{th}$ : excitatory

- $E_{syn} < E_L$ : inhibitory
- $E_{syn} \approx E_L$ : shunting inhibition

Voltage Clamp experiments (excitatory synapse): synapse input is well captured by Ohm Law's. Modified membrane patch equation with a synapse:

$$C \frac{dV_m}{dt} + g_{syn}(t)(V_m - E_{syn}) + \frac{V_m - V_{rest}}{R} = 0$$

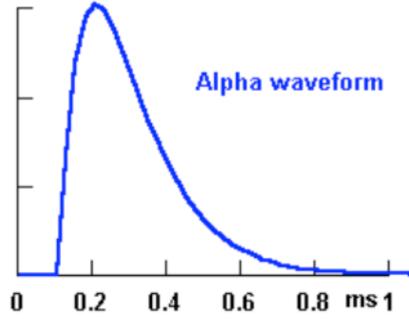
Rewriting:

$$\tau \frac{dV_m}{dt} = -(1 + Rg_{syn}(t))V_m + Rg_{syn}(t)E_{syn} + V_{rest}$$

where  $\tau = RC$ .

## 8.1 Alpha function

Synapse input is usually approximated by an “alpha function”:  $g_{syn} = g_{peak} t \exp\left(\frac{-t}{t_{peak}}\right)$ .



To model many synaptic input, we add them in parallel to a RC circuit. This way:

$$C \frac{dV_m}{dt} = \sum_{i=0}^n g_{syn,i}(t)(E_{syn,i} - V_m) + \frac{V_{rest} - V_m}{R}$$

Considering the synaptic input varying slowly:  $g_{syn}(t) \approx g_{syn}$ . Also, if  $V_m \ll E_{syn}$ , the synaptic input can be approximated as a const current source:  $g_{syn} \cdot E_{syn}$ . Then:

$$\tau' \frac{dV_m}{dt} = -V + \frac{g_{syn} E_{syn}}{G_{in}}$$

where,  $\tau' = \frac{C}{G_{in}}$  and  $G_{in} = g_{syn} + \frac{1}{R}$ .

That leads to:

$$V_\infty = \frac{Rg_{syn}E_{syn}}{1 + Rg_{syn}}$$

### 8.1.1 Small synaptic input

Voltage scales linearly with synaptic input.  $Rg_{syn} \ll 1 \rightarrow V_\infty = Rg_{syn}E_{syn}$

### 8.1.2 Large synaptic input

Voltage saturates at synaptic reversal potential.  $Rg_{syn} \gg 1 \rightarrow V_\infty = E_{syn}$

## 8.2 Shunting inhibition

Special case when the synapse reversal potential is equal to the resting membrane potential.

Consider the scheme in Figure 28. We have an excitatory and an inhibitory synapse.

$$C \frac{dV}{dt} = g_{ex}(E_{ex} - V) - g_{in}V - \frac{V}{R}$$

Rewriting,

$$\tau' \frac{dV}{dt} = -V + \frac{g_{ex}E_{syn}}{G_{in}}$$

where,  $\tau' = \frac{C}{G_{in}}$  and  $G_{in} = g_{ex} + g_{in} + \frac{1}{R}$ .

$$V(t) = \frac{g_{ex}E_{ex}}{G_{in}}(1 - e^{-\frac{t}{\tau'}})$$

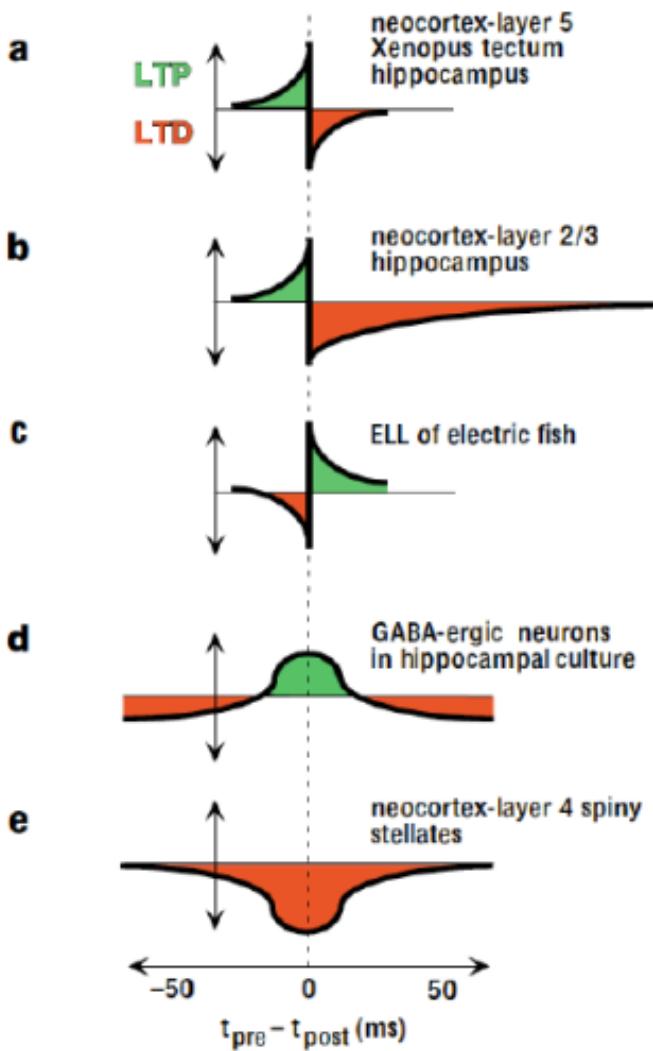
$$V_\infty = \frac{g_{ex}E_{ex}}{g_{ex} + \frac{1}{R} + g_{in}}$$

Notice,  $g_{in}$  only appears in the denominator. Shunting inhibition is also called “divisive inhibition”. The stronger the inhibition, smaller is the voltage.

## 8.3 Plasticity (Hebb's Law) - 1940's

Neurons that fire together, wire together.

### 8.3.1 Spike-time dependent plasticity



## 9 Plasticity and Learning

We need plasticity/learning because intelligent behavior emerge with learning.

### 9.1 Learning and Memory

- Learning is the acquisition/storage of new information or knowledge (or formation of a memory through experience).
- Memory is the stored information that can be recalled at a later stage.
- Learning results in memory and change in future behavior.
- Learning does not imply a conscious attempt to learn (simple observation can lead to creation of a memory).
- Plasticity is the biological implementation of learning: allow us to form memories.
- The brain has a more complex configuration than given by genes.
- There is lots of room for learning (and also need).
- Only brain area responsibility and cortex thickness (6 layers) are genetically fixed. But even that can sometimes change later on.

#### 9.1.1 Types of Memory

Fundamentally, we are derived from "learning" experiences that have been stored in our nervous system.

- Declarative/Explicit memory (facts, events): medial temporal lobe
- Non-declarative/Implicit memory:
  - priming: neocortex
  - procedural memory (skill, habits): striatum
  - associative emotional responses: amygdala
  - associative skeletal musculature: cerebellum
  - nonassociative (habituation, sensitization): reflex pathways

**Short- and Long-term Memory** Short-term memory (STM) lasts seconds to hours, long-term memory (LTM) lasts hours to months and long-lasting memory from months to lifetime.

### 9.1.2 Connections to 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.
- Synapses get formed, retracted.
- Synapse efficacies/plasticity can change.

## 9.2 Plasticity

Plasticity can occur in different levels: brain area, cellular level and synapse level.

### 9.2.1 Networks and system plasticity

Hippocampus as a model system to study learning and memory

**Patient HM** Amygdala, hippocampal gyrus, and anterior two thirds of the hippocampus were removed. Patient had severe anterograde amnesia. Normal STM, normal LTM (events before surgery). Couldnt transfer memory from STM to LTM and couldnt consolidate new declarative memory. Conclusion: Hippocampus is not a permanent area of storage, it is involved in consolidation but not in retrieval.

### 9.2.2 Cellular plasticity - Perceptron

<sup>3</sup> Occurs in the local brain circuitry and neuron. At this level, the plasticity it is like training the weights of a neural network.

**McCulloch and Pitts neurons, a.k.a, Perceptron** McCulloch and Pitts neurons implement a linear decision boundary, that is defined by the weights and bias. They can implement many logical operations as long as they are linearly separable. They can be trained on labeled datasets.

**Hippocampus and spatial memory** In the hippocampus we can find place cells, grid cells, etc.

**Morris water maze** In a control mouse, after training the mouse can find the platform in the maze without looking for it in the wrong places. However, if we block NMDA receptors in the hippocampus, the mouse doesnt recognize the right place to search for the platform.

**How can we measure neuronal plasticity in the Hippocampus?** Most studied synapses:  $CA3 \rightarrow CA1$  in hippocampus. We stimulate neurons at  $CA3$  and record the effects at  $CA1$  neurons.

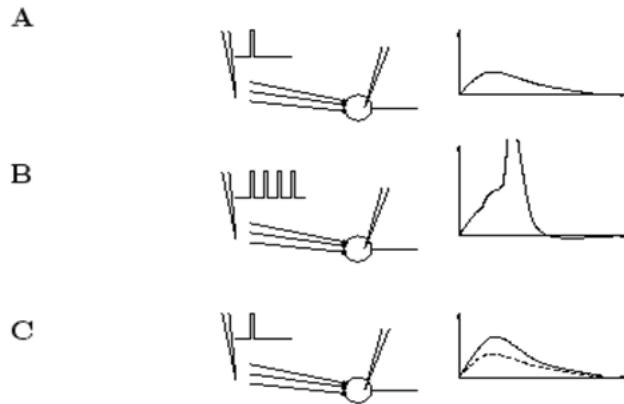
---

<sup>3</sup>There is a chapter only about Perceptron

## 9.3 Synapse plasticity

Synapse size changes, AMPA/NMDA ratio changes (more vesicles), number of spines changes.

- Modification of postsynaptic potentials (PSPs) evoked by presynaptic spikes.
- A: Postsynaptic response triggered by a weak test pulse.
- B: Strong stimulation triggers postsynaptic firing.
- C: A later test pulse evokes a larger postsynaptic response than initially.



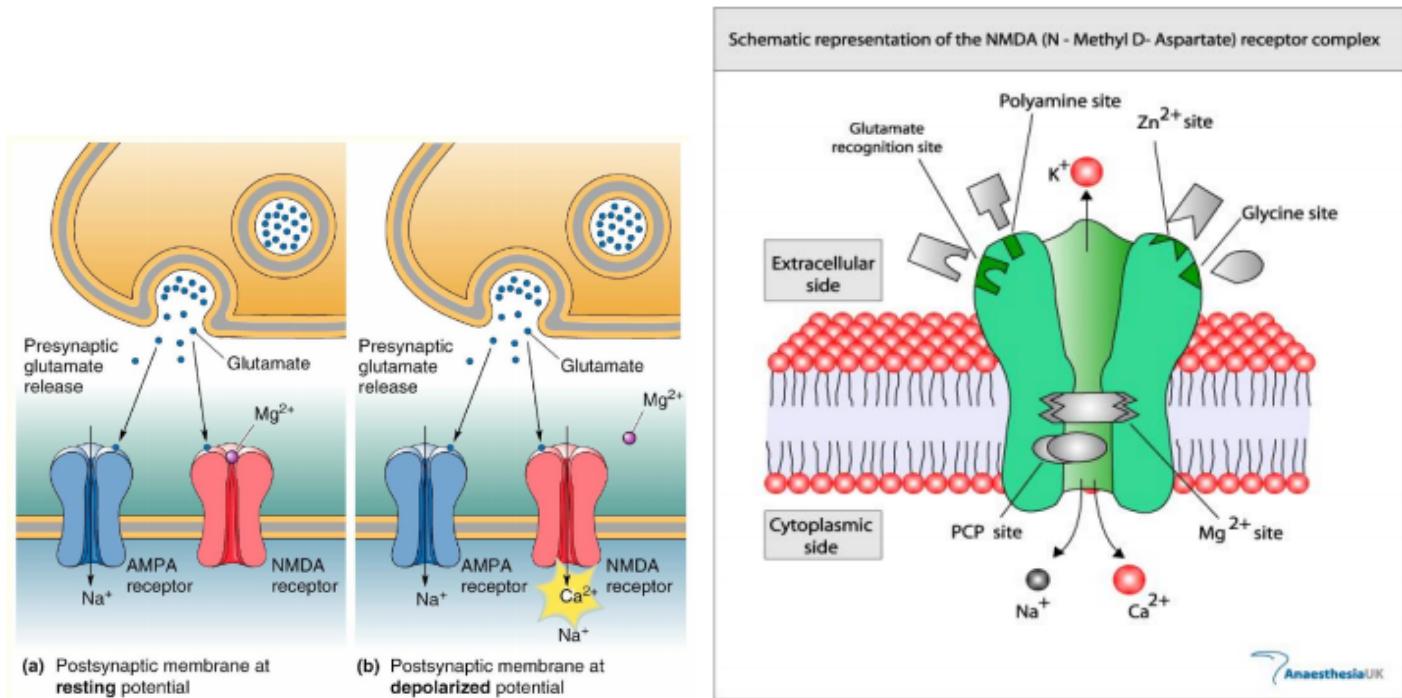
### 9.3.1 Parameters that define synapse strengths

- Neurotransmitter and receptor type
- Position of synapse
- Availability of vesicles
- Re-uptake of transmitters
- Neuromodulators, such as dopamine
- Postsynaptic cellular processes (such as more or less receptors)
- Pre/postsynaptic firing

### 9.3.2 Models of Synaptic Plasticity

- There is also non synaptic plasticity, such as dendrite strength, excitability of neurons, isolation of axons.
- Phenomenological models show input-output relationships between activity and plasticity.
- Biophysical models tell what processes are involved.
- Hebb's Postulate: "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"

### 9.3.3 NMDA Synapse

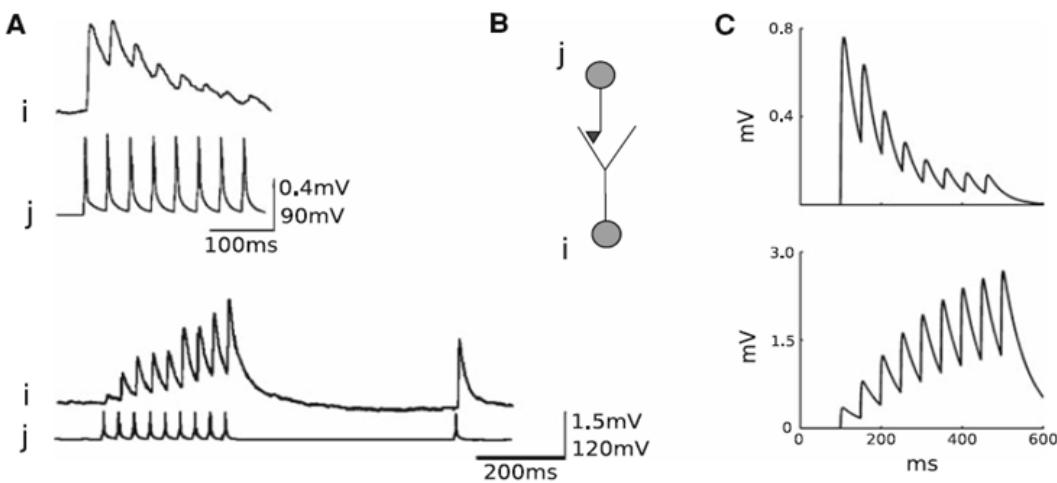


- Can act as a coincidence detector for pre- and postsynaptic firing.  
Backpropagating action potentials.  
Depolarization from other synapses.
- Calcium influx is crucial for plasticity (ions flow through the activated NMDA receptor)
- Level and timing of  $\text{Ca}^{2+}$  rise in spine determines LTD or LTP.
- Low frequency synaptic firing ( 5 Hz) produces LTD, high frequency synaptic firing ( 50 to 100 Hz) produces LTP.
- Strong NMDA receptor activation gives potentiation.
- Weak NMDA receptor activation gives depression.

### 9.3.4 Potentiation with NMDA

- Phosphorylation of AMPA receptors makes the synapse stronger.
- Synthesis of new AMPA (but not NMDA) receptors.
- Transport of AMPA receptors to membrane.
- Release probability or quantity of the presynapse can be improved.

### 9.3.5 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 milliseconds to seconds.
- Short term depression is a safety mechanism.

### 9.3.6 Long-term potentiation (LTP)

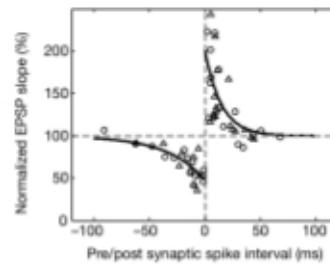
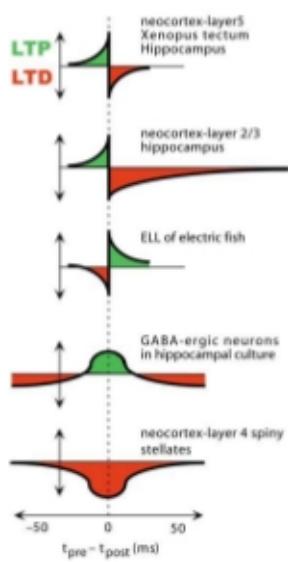
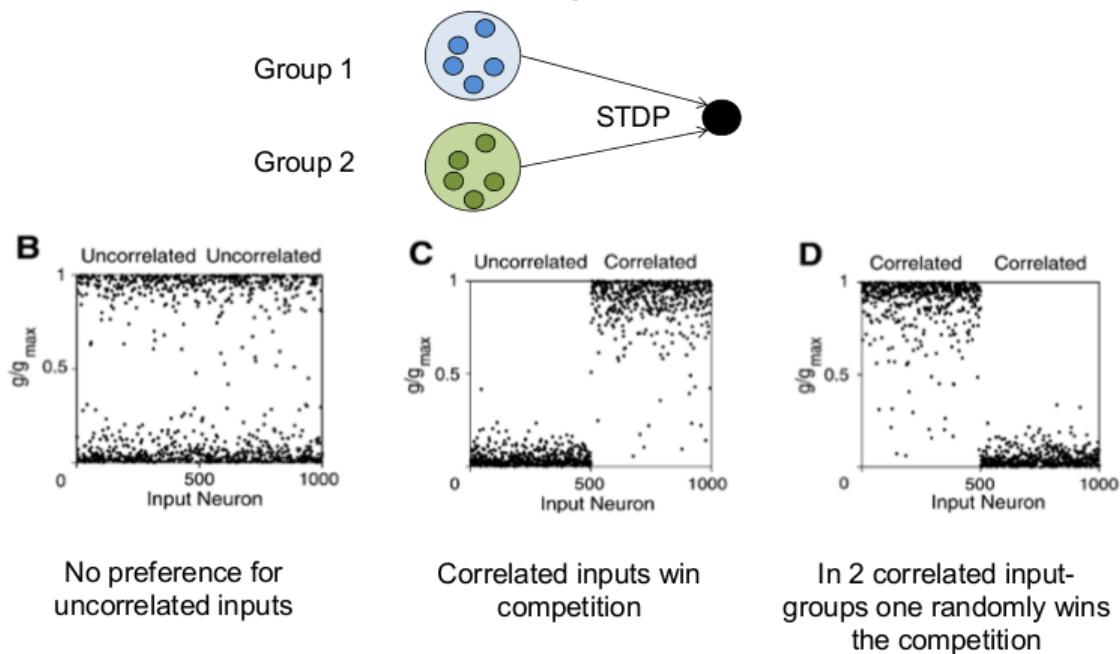
- It results in strengthening of the synapse for varying periods of time following tetanic stimulation.
- The mechanisms for LTP lasting 30 min to a few hours do not require new protein synthesis. Longer than a few hours do require protein synthesis.
- The hippocampus is involved in transferring from short to long term memory.
- Tetanus stimulus, strong, high frequent stimulation are required.
- A pre- and postsynaptic depolarization at the same time is needed.
- Voltage clamp during tetanus prevents LTP from happening.
- LTP is cooperative, many weak synapse stimulations give also some effect.
- LTP needs a simultaneous depolarization beyond a threshold.
- LTP is input specific and can enhance the synaptic effectiveness of a synapse without affecting other synapses in the same cell. This increases the storage capacity of individual neurons.
- LTP is associative, weak stimulation in pathways coupled with strong simulation in other pathways can induce LTP.
- Stimuli must be delivered at high frequency, because the post-synaptic cell must be depolarized past a certain threshold for LTP.
- LTP has a transient early phase (1 – 3 hours), followed by a consolidate later phase ( $\geq 24$  hours). The early phase doesn't need new protein synthesis. The later phase needs protein and RNA synthesis for new presynaptic active zones and postsynaptic receptors.

### 9.3.7 Long-term depression (LTD)

- Weakening of the synapse.
- NMDA are required

### 9.3.8 Spike-timing dependent Plasticity (STDP)

- Not only correlation but also timing of spikes is important.
- NMDA receptors and backpropagating action potential creates this timing-dependence of 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 the 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.



- 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.
- If the post-synaptic spike is emitted soon before the pre-synaptic one arrives, the synaptic efficacy is decreased.

### 9.3.9 Functional consequences of STDP

- Rate normalization, temporal coding, reduced latency, prediction and conditioning.
- Only one direction can get stronger, no positive feedback can occur.
- STDP can become a simple temporal pattern detector and already fire on the beginning of such a pattern.

- Stimulation frequency has an effect on STDP.
- Dopamine can extend the LTP timing window or even convert LTD to LTP. Dopamine floats around the cells.
- LTP is blocked by AP5. Behavioral success and LTP are correlated.

### 9.3.10 Factors that influence plasticity

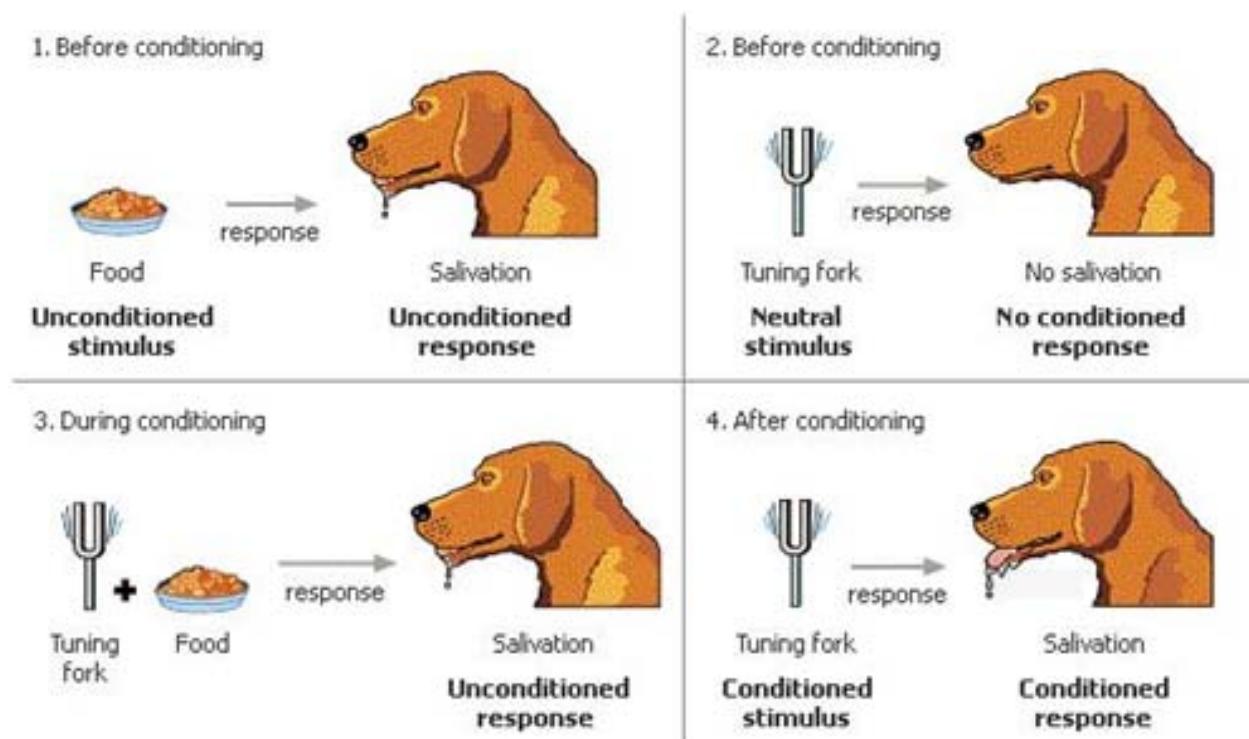
- Different plasticity in different brain areas.
- Diversity of neuron and synapse types.
- Large number of control parameters for plasticity experiments.
- Influence of neuromodulators, calcium, drugs, proteins.
- Long term vs. short term effects.
- Unlikely that a single model explains all plasticity effects found in biology.

## 9.4 Hebbian Learning

When an axon of cell A repeatedly or persistently takes part in firing cell B, then A's efficiency as one of the cells firing B is increased. "neurons that fire together, wire together", i.e., Spike Time Dependent Plasticity.

- Learning based on correlations between pre- and postsynaptic firing.
  - Uses only variables locally available at the synapse.
  - Expressed in a rate-based model:  $\Delta w_{ij} \propto v_i \cdot v_j$ .
  - Only weight increases, potentiation modeled.
  - It will lead to instability because of positive feedback loops.
  - Other rules can be added for weight reduction (depression) and normalization (Oja, BCM).
  - Hebb's postulate implies constraints for synaptic learning:
    - Direction of information flow (forward).
    - Global effects arise from local learning.
- Variables are action potentials, synapse weights (efficacy) and neuromodulator/calcium concentration (locally).

## 9.5 Pavlovian Conditioning



# 10 Neuromorphic VLSI

## 10.1 VLSI

- Very large scale integration technology allows to fabricate chips and memory.
- VLSI are usually digital, high power, not fault tolerant or robust, and clocked (synchronous), not massively parallel.

The failure of one transistor is the failure of the computer

The computer hardware had a radical paradigm shift when we looked to real brains. For instance, a bee brain is much smaller and consume much less power (using neurons in a slow way), and offer real time interaction with the environment and complex behavior.

### 10.1.1 Neuromorphic

The term appeared in the 80's to describe 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.

#### Goals

- Neuromorphic VLSI contain analog/digital circuits that exploit the physics of silicon to reproduce the bio-physics of neural circuits.
- The goals are to understand biological neural systems using standard CMOS VLSI technology as a tool.
- Known properties of biological systems can be exploited to design devices for engineering applications.

New hardware different of conventional computers: radically different from von Neumann architectures. Now, there are parallel elements with memory and computation co-localized, with continuous streaming data driven computation, no clock. The co-localization of memory and computation allows to have no I/O bottleneck and no memory bottleneck.

**Neuromorphic computing vs engineering** Neuromorphic computing uses a dedicated VLSI hardware, high performance computing, it is application driven and uses conservative approaches. Neuromorphic engineering is a fundamental research, deeply rooted in biology. It emulates neural function in a subthreshold analog and asynchronous digital.

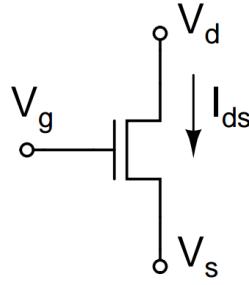
## 10.2 Neuron circuits

- Reproduce physics of neural computation using subthreshold analog circuits and asynchronous digital circuits.
- Build autonomous learning behaving systems that can interact with the environment in real-time.
- Best exploit for current and future VLSI technologies.
- Suited for nano and emerging technologies.
- Ideal tools for real- and accelerated-time modeling of neural systems.
- Compact, low-power sensory processing devices.
- Can interface directly with living systems.

## 10.3 Circuits

Digital transistors operating only in the minimum and the maximum. Analog transistor use also intermediate amounts, thus transistors can emulate physical proteic channels. In biology, at high voltages, the fraction of the channels that are open approaches unity, causing a saturation. The same

can be seen in a subthreshold regime. In subthreshold, the current is smaller than 1V, it increases exponentially, and after threshold currents changes quadratically. It changes from pico to nano amps.



In Complementary Metal-Oxide Semiconductor (CMOS) technology, there are two types of MOS-FETs: n-FETs and p-FETs. There is no current going to transistors.

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 connected to the power supply rail ( $V_{dd}$ ).

### 10.3.1 n-Fet subthreshold

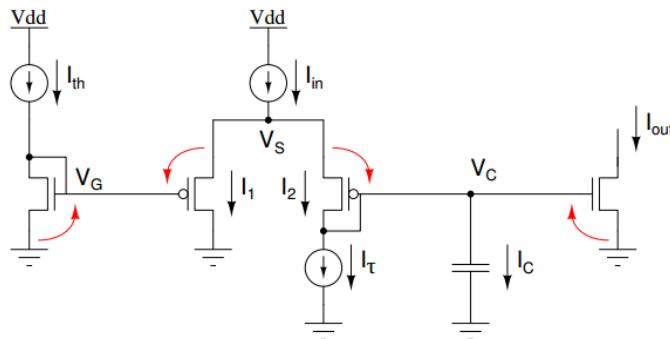
- $I_0$  current-scaling parameter.
- $\kappa_n$  subthreshold slope factor.
- $U_T$  the thermal voltage.
- $V_g$  the gate voltage,  $V_s$  the source voltage,  $V_d$  the drain voltage.
- The current is defined to be positive if it flows from the drain to the source.
- $I_{ds} = I_0 e^{\kappa_n \frac{V_g}{U_T}} (e^{-\frac{V_s}{U_T}} - e^{-\frac{V_d}{U_T}})$   
 $I_{ds} = I_0 e^{\kappa \frac{V_g}{U_T} - \frac{V_s}{U_T}} - I_0 e^{\kappa \frac{V_g}{U_T} - \frac{V_d}{U_T}} \rightarrow I_{ds} = I_f - I_r$
- If  $V_{ds} > 4U_T$  it becomes  $I_{ds} = I_0 e^{\kappa_n \frac{V_g}{U_T} - \frac{V_s}{U_T}}$

### 10.3.2 p-Fet subthreshold

The corresponding (complementary) equation for the p-FET:

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

### 10.3.3 Differential-pair



- $I_{in} = I_1 + I_2$
- $I_2 = I_\tau + I_C$
- $I_{out} I_0 e^{\frac{\kappa \alpha V_C}{U_T}}$

- $I_C = C \frac{d}{dt} V_C$
- $I_C = C \frac{U_T}{\kappa I_{out}} \frac{d}{dt} I_{out}$
- $\tau = \frac{C U_T}{\kappa I_\tau}$
- $I_{th} \cdot I_1 = I_2 \cdot O_{out}$
- $I_{th} \cdot (I_{in} - I_\tau - I_C) = (I_\tau - I_C) \cdot I_{out}$

$$\tau(1 + \frac{I_{th}}{I_{out}}) \frac{d}{dt} I_{out} + I_{out} = \frac{I_{th} I_{in}}{I_\tau} - I_{th}$$

If  $I_{in} \gg I_\tau$ :

$$\tau \frac{d}{dt} I_{out} + I_{out} = \frac{I_{th}}{I_\tau} I_{in}$$

For synapses:

$$\tau \frac{d}{dt} I_{syn} + I_{syn} = \frac{I_{th}}{I_\tau} I_w$$

## Differential pair circuit

- $I_b = I_1 + I_2 = I_0 e^{\frac{V_b}{U_T}}$
- $I_1 = I_b \frac{e^{\frac{\kappa V_1}{U_T}}}{e^{\frac{\kappa V_1}{U_T}} + e^{\frac{\kappa V_2}{U_T}}}$
- $I_2 = I_b \frac{e^{\frac{\kappa V_2}{U_T}}}{e^{\frac{\kappa V_1}{U_T}} + e^{\frac{\kappa V_2}{U_T}}}$

The output currents of the diff-pair can be rewritten in the canonical sigmoid form:

$$I_1 = I_b \frac{1}{1 + e^{\frac{\kappa}{U_T}(V_2 - V_1)}}$$

$$I_2 = I_b \frac{1}{1 + e^{\frac{\kappa}{U_T}(V_1 - V_2)}}$$

Difference of diff-pair currents:

$$I_1 - I_2 = I_b \frac{e^{\frac{\kappa V_1}{U_T}} - e^{\frac{\kappa V_2}{U_T}}}{e^{\frac{\kappa V_1}{U_T}} + e^{\frac{\kappa V_2}{U_T}}} = I_b \tanh\left(\frac{\kappa}{2U_T}(V_1 - V_2)\right)$$

Two ways to implement the difference of currents: using a current mirror or a transconductance amplifier.

## Current Mirror

- Two MOSFETs of the same size.
- $I_{out} = e^{(V_{s1} - V_{s2})/U_T} I_{in}$

## Transconductance Amplifier

$$I_{out} = I_b \tanh\left(\frac{\kappa}{2U_T}(V_1 - V_2)\right)$$

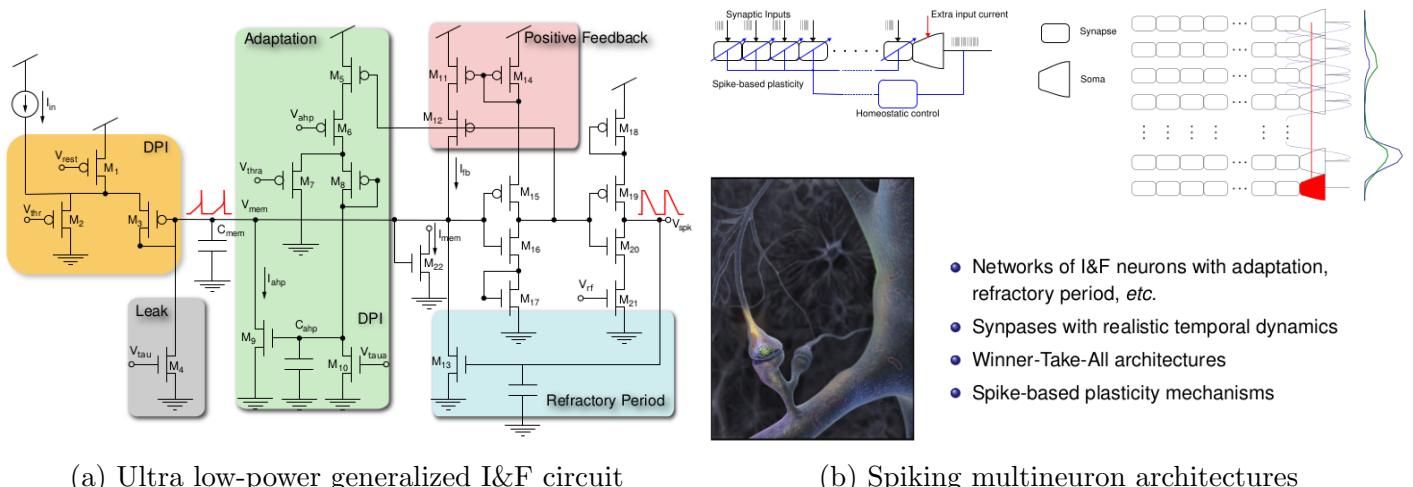
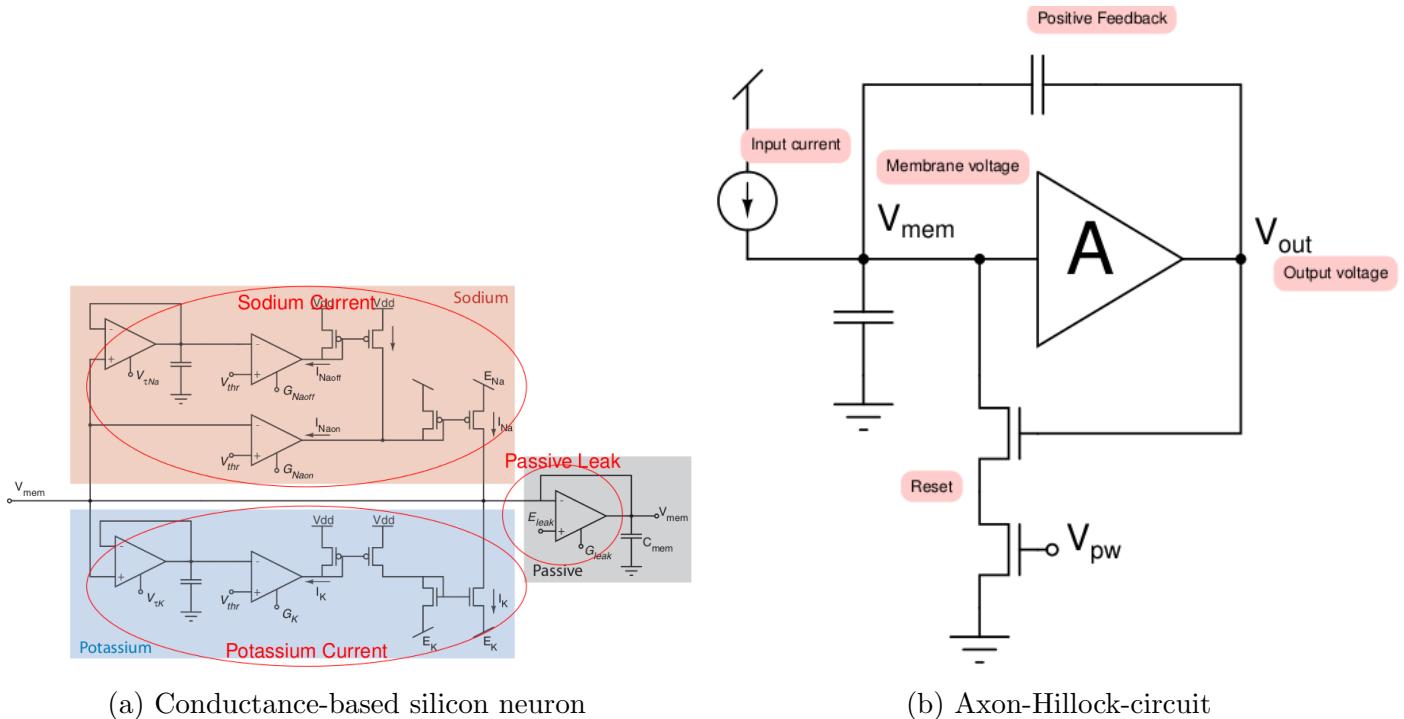
For small differential voltages (e.g.  $|V_1 - V_2| < 200mV$ ), the  $\tanh(\cdot)$  relationship is approximately linear and:

$$I_{out} \approx g_m(V_1 - V_2)$$

where  $g_m = \frac{I_b \kappa}{2U_T}$ .

## 10.4 Remarkable circuits

- McCulloch and Pitts artificial neuron model (first).
- Mahowald and Douglas conductance-based silicon neuron, similar to real cortical neurons, Figure a in ??.
- properties remarkably similar to those of real cortical neurons
- Two main classes of neuron models: Conductance based, and Integrate and fire model (I&F).
- Recently, Generalized Integrate and Fire models bridge the gap between the two.



## 10.5 Spikes and Adress Event Representation

Adress Event Representation (AER) represents when a spike (action potential) arrives. At the time of a spike, the adress where the spike occurred is broadcast.

## 10.6 Neuromorphic chips

### 10.6.1 Dynap: DYNamic neuromorphic Asynchronous Processor

- analog and digital co-design
- Distributed SRAM and TCAM memory cells
- On-chip inference and learning

### 10.6.2 ROLLS: a Reconfigurable On-Line Learning Spiking chip

- short-term plasticity
- long-term plasticity
- homeostatic plasticity
- configurable recurrent connectivity

Many more...

### 10.6.3 Neuromorphic Cognitive Systems

- Working memory and decision making in autonomous real-time systems
- Context dependent embedded systems and emerging technologies
- Brain machine interfaces and prosthetics

# 11 Introduction to Perceptron Learning Algorithm

When we look inside the brain, we see a bunch of neurons and a mess of connectivities. It's really hard to identify the connections among neurons, and with this information to understand how the brain works. We don't learn how the brain works by studying neurons, the same way that just by studying transistors we do not know how a computer works. We know the brain does processing but we don't know how it works. The bottleneck to understand brain is probably that we do not have the right abstractions to understand it.

McCulloch and Pitts developed a computational model of a biological neuron in 1943. The McCulloch-Pitts neural model is also known as linear threshold unit/gate. It models a neuron with a set of inputs and one output.

## 11.1 McCulloch-Pitts neurons vs biological neurons

### similarities:

- Both can be active or inactive.
- The input/output is directed.
- The activation of a neuron is dependent on a weighted function of other neurons.

### differences:

- Real neurons exist in continuous time, whereas McCulloch-Pitts neurons operate in discrete time.
- Real neurons have degrees of activation, not just on/off.
- The activation as a function of the inputs of real neuron is typically not linear or threshold linear.

## 11.2 (Basic) Digital Logic

Gates are processing units. They are functions that evaluate the inputs. Each input has a 0/1 value that can be seen as a false/true in digital logic. There are different types of gates, each type is represented by a shape, but we can also just use their names to refer to them. Figure 32 shows basic types of gates and their shapes: and, or and not. The behavior of a gate does not depend on the number of inputs. For instance, an AND gate returns 1 (or true) only if all the inputs are also 1 (or true), independent of the number of inputs. The same is valid for the OR gate, that only returns 0 (or false) when all the inputs are 0 (or false).

We use gates to help us to understand the computation that might happen in the brain. Neurons and axons do not behave as wires, but this is the tool we have available. And while neurons do not have 0s and 1s, they can be active/inactive, what gives us a good approximation.

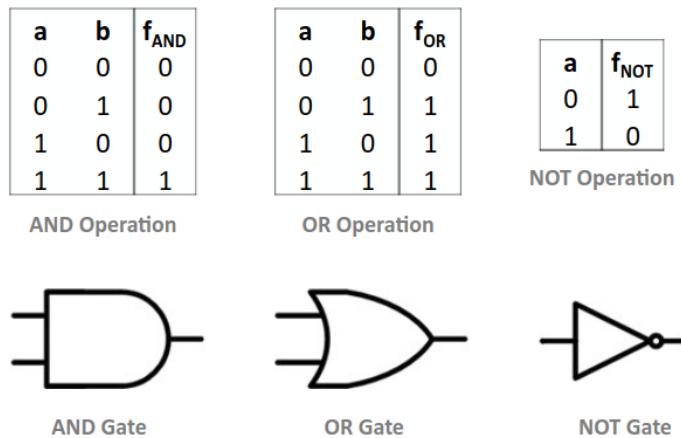


Figure 32: Basic gates.

Circuits are a combination of gates.

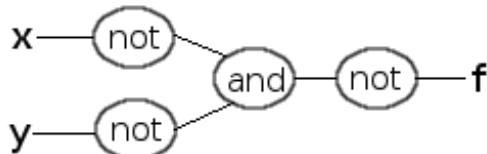


Figure 33: A basic circuit implementing an OR gate.

The circuit in Figure 33 produces the same output as an OR gate. With NOT and AND gates we can build an OR gate, but with AND and OR we can't build a NOT gate.

**XOR gates = exclusive OR** They are exclusive in the case of two inputs. For more than two, XOR counts the number of "active" (1's) inputs and returns 0 for even and 1 for odd.

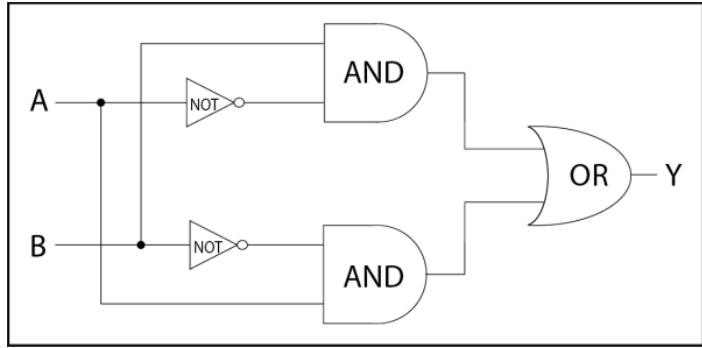


Figure 34: XOR gate built out other gates. Image from <https://blog.digilentinc.com/building-logic-gates-with-transistors/>

A table with  $N$  inputs has  $2^N$  rows.

**Other gates** There are more gates in digital logic than AND, OR and NOT gates. The Table 1 shows a few more types of gates with its truth table.

Table 1: Truth table of logic gates.

Input		Output						
A	B	AND	OR	NOT	XOR	NAND	NOR	XNOR
0	0	0	0	1	0	1	1	1
0	1	0	1	1	1	1	0	0
1	0	0	1	0	1	1	0	0
1	1	1	1	0	0	0	0	1

### 11.3 Linear Threshold (LT) Unit/Gates (Perceptron)

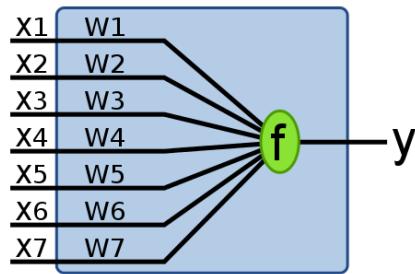


Figure 35: Perceptron

The model in Figure 35 represents a neuron with inputs  $x$ 's and one output  $y$ . Weights  $w$  determine the influence of the inputs.  $f$  is a function determining the output: if the influence of all the inputs combined cross a threshold, then the neuron become active. Active state:  $\sum_i(w_i \cdot x_i) \geq \theta$ . Otherwise, the neuron is inactive.

We add a bias input as  $-\theta$  so that  $w_0 + \sum_i(w_i \cdot x_i) \geq 0$  activates the neuron. As a digital abstraction, we consider that the activity of a neuron can be described as 0 or 1. Where 0 is inactive and 1 is active.

Neurons can learn by adjusting their weights, they can have thousand of inputs. For a desired behavior there are many possible weight vectors that can work (if any can work).

This model can create AND/OR/NOT-Gates.

### 11.3.1 Threshold function

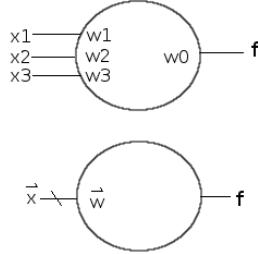


Figure 36: Same unit in two representations

Both units in Figure 36 are the same. The one in the bottom is a simplified version, where the inputs  $x$  and the weights  $w$  are represented as vectors. The bias term is added to the weight vector and a new input  $x_0$  is added with a fixed value of 1.

$$f(x_1, x_2, x_3) = \theta(w_1 \times x_1 + w_2 \times x_2 + w_3 \times x_3 + w_0)$$

$$f(\vec{x}) = \theta(\vec{w} \times \vec{x} + w_0)$$

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

### 11.3.2 XOR impossibility with LT/Perceptrons

To compute XOR (please check the truth table for XOR in the Table 1) with LT, it is required that:

- $w_0 < 0$
- $x_2 \times w_2 + w_0 \geq 0$
- $x_1 \times w_1 + w_0 \geq 0$
- $x_1 \times w_1 + x_2 \times w_2 + w_0 < 0$ .

This causes a contradiction because  $x_1 \times w_1 + x_2 \times w_2 + 2 \times w_0 < 0$  and  $x_1 \times w_1 + x_2 \times w_2 + 2 \times w_0 \geq 0$  when adding up the constraints.

XOR is not a linear combination of the inputs. And the perceptron/LT uses a line to separate classes, i.e., all the points in the same side of the line will have the same output. Although these units can't model the XOR, they are still powerful. You can't calculate it with a single unit, but with a combination of them, it is possible. The same way, one cannot compute the XOR function using one single OR, NOT or AND gate. In fact, we need three gates<sup>4</sup> to compute the XOR function, see Figure 34.

In the end, these model of neuron units are more efficient than digital logic gates, however, these new analog units run into precision problems. For implementing the XOR function, McCulloch and Pitts units allow exponentially smaller circuits.

## 11.4 Perceptron Learning Algorithm

This algorithm is called a learning algorithm because we use it to define the weights of the neurons. This learning is just a definition of parameters, and can be seen as an optimization, where we have some guess and we want to improve it.

---

<sup>4</sup>when counting gates, we do not count NOT gates

x1	x2	f
0	0	1
0	1	0
1	0	1
1	1	1

Table 2: Two inputs  $x_1$  and  $x_2$  resulting in  $f$

**Supervised error-correcting rules** We start with an initial guess of weights, we compare the output in response to the input with the desired output and then we change the weights to improve the performance.

Consider the mapping on the Table 2. Can we set  $w_0, w_1, w_2$  (bias term, and the weights of each input) so that this unit computes the  $f$  from the table?

Learning happens by changing synaptic weights. How can we change the synaptic weights of a unit to make it behave as desired?

Let's start with random weights, let's say all zero. With this weights, doesn't matter the values of  $x_1$  and  $x_2$ , the result will always be 1. And for the second case ( $x_1 = 0, x_2 = 1, f = 0$ ) we produce a wrong output. So, we reduce  $w_0$  and  $w_2$  because they contribute to the sum ( $0 \times w_1 + 1 \times w_2 + w_0$ ). We reduce the weights if the sum should go down and we increase the weights if the sum should go up. We iterate this step until convergence.

We can consider the bias as a weight with input value always one ( $x_0 = 1$ ), thus, we can write:

$$f(\vec{x}) = \theta(\vec{w} \times \vec{x} + w_0) = \theta(\vec{w} \times \vec{x})$$

**Recall**  $\vec{w} \times \vec{x} = |\vec{w}| \times |\vec{x}| \times \cos \alpha$ . So, if  $\alpha < 90 \text{ deg}$ ,  $\vec{w} \times \vec{x}$  is greater than zero and if  $\alpha > 90 \text{ deg}$ ,  $\vec{w} \times \vec{x} < 0$ . This way, we compute a similarity between the weight vector and the input vector.

#### 11.4.1 Convergence

These single units can separate two classes if they are linearly separable (i.e., if a solution exists). The Perceptron Learning Algorithm must converge, i.e., it will update the weights a finite number of times.

**Proof 1** Suppose there is a solution  $\vec{w}^*$ , i.e., the data is linearly separable. The weights of the perceptron units can be seen as the components of a normal vector to the hyperplane that separates the classes (1 and 0). In fact, the length of the normal vector doesn't matter, we are looking for its direction. Pick any solution  $\vec{w}$ , for instance, starting with all weights as zero, if this solution already satisfies our conditions, we are done. Otherwise, we pick an arbitrary misclassified point and update  $\vec{w}$ . Each step makes progress in the  $\vec{w}^*$  direction, because additions to  $\vec{w}$  are always  $\leq 90 \text{ deg}$ . The magnitude of  $\vec{w}^* \times \vec{w}$  increases linearly. Since  $\vec{w}^*$  doesn't change, there is a maximum growth from  $\vec{w}$  to achieve the solution. By contradiction in the limit of infinite steps, we can say that the algorithm converges, i.e., if there is a solution and it takes infinite steps to achieve it, this is a contradiction.

**Proof 2** In homogeneous coordinates  $\hat{\mathcal{X}}'_n \equiv \{\hat{x}'_i = l_i \hat{x}_i | (x_i, l_i) \in \hat{\mathcal{X}}_n\}$ .

- $\hat{w}^*$  (solution)  $\in \mathbb{R}^{n+1}$  is the homogeneous weight vector:  $\hat{w}^T \hat{x}'_i > 0, \forall i = 1, \dots, n$ .
- $\hat{w}(k) \in \mathbb{R}^{n+1}$  is the homogeneous weight vector of the  $k$ -th update.
- $\hat{x}'(k) \in \hat{\mathcal{X}}'_n$  is the normalized homogeneous feature vector that triggered the update.

Then,  $\hat{w}(1) = \hat{w}(0) + \eta \hat{x}'(1)$ ,  $\hat{w}(2) = \hat{w}(1) + \eta \hat{x}'(2)$ ,  $\hat{w}(3) = \hat{w}(2) + \eta \hat{x}'(3)$ ,  $\dots$ ,  $\hat{w}(k) = \hat{w}(k-1) + \eta \hat{x}'(k)$ . Can we find two constants  $A$  and  $B$  such that

$$Ak^2 \leq \|\hat{w}(k) - \hat{w}(0)\| \leq Bk \quad (1)$$

?

If Equation 1 holds then the growth of  $\hat{w}(k)$  is limited. As  $k$  increases,  $Bk$  will be smaller than  $Ak^2$ . For 1 to hold,  $\|\hat{w}(k) - \hat{w}(0)\|$  cannot increase after  $k = \frac{B}{A}$  updates.

We show that the vector that is the difference between the original vector and the  $k$ -th one is limited: the algorithm must converge.

**Lower bound A:**

$$\hat{w}(k) = \hat{w}(0) + \eta(\hat{x}'(1) + \hat{x}'(2) + \dots + \hat{x}'(k))$$

$$\hat{w}^{*T}(\hat{w}(k) - \hat{w}(0)) = \hat{w}^{*T}\eta(\hat{x}'(1) + \hat{x}'(2) + \dots + \hat{x}'(k))$$

$$\text{Given } a = \min_{\hat{x} \in \hat{\mathcal{X}}_m} \hat{w}^{*T} \hat{x} > 0 \text{ then } \hat{w}^{*T}(\hat{w}(k) - \hat{w}(0)) \geq \underbrace{\eta a k}_{\text{summing } k \text{ times the minimum term}} > 0$$

From the Cauchy-Schwartz:  $\|a\|^2 \|b\|^2 > |a^T b|^2$

$$\|\hat{w}^{*T}\|^2 \|\hat{w}(k) - \hat{w}(0)\|^2 > |\hat{w}^{*T}(\hat{w}(k) - \hat{w}(0))|^2 \geq (\eta a k)^2$$

$$\rightarrow \|\hat{w}(k) - \hat{w}(0)\|^2 \geq \underbrace{\left( \frac{\eta a}{\|\hat{w}^*\|} \right)^2}_{A} k^2.$$

**Upper bound B:**

$$\hat{w}(1) - \hat{w}(0) = \eta \hat{x}'(1)$$

$$\hat{w}(k) - \hat{w}(0) = (\hat{w}(k-1) - \hat{w}(0)) + \eta \hat{x}'(k)$$

$$\|\hat{w}(1) - \hat{w}(0)\|^2 = \eta \|\hat{x}'(1)\|^2$$

$$\|\hat{w}(k) - \hat{w}(0)\|^2 = \|\hat{w}(k-1) - \hat{w}(0)\|^2 + 2\eta(\|\hat{w}(k-1) - \hat{w}(0)\|^2)^T \hat{x}'(k) + \|\hat{x}'(k)\|^2 \eta^2$$

Since  $\hat{x}'(1)$  triggers the first update, then  $\hat{w}(0)^T \hat{x}'(1) < 0 \rightarrow \hat{w}(j-1)^T \hat{x}'(j) < 0 \forall j = 1, \dots, k$

Summing up the above:  $\|\hat{w}(k) - \hat{w}(0)\|^2 \leq \|\hat{w}(k-1) - \hat{w}(0)\|^2 + B^2$

By induction:  $\|\hat{w}(k) - \hat{w}(0)\|^2 \leq kB^2$

**Combining the bounds:**

$$k^2 A \leq \|\hat{w}(k) - \hat{w}(0)\|^2 \leq kB^2$$

from which follows that  $k \leq \frac{B^2}{A}$ . So, the number of errors the PLA makes is finite, as shown in Figure 37.

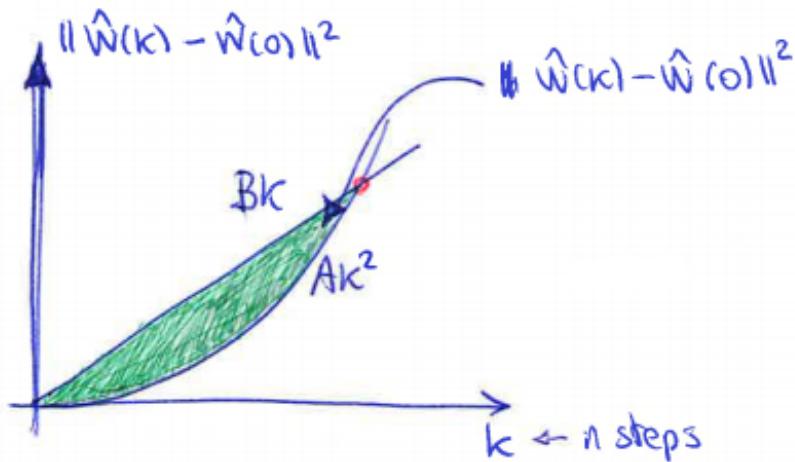


Figure 37: The space of searching for the solution has an upper and a lower bound.

## Algorithm

- Choose random initial weights.

- Calculate output for given input.
- If the output is not the expected value, then  $e = d - c$ , where  $d$  is the desired output and  $c$  the current output.
- Change the weight of inputs and bias by  $\Delta w_i = e \cdot \alpha \cdot x_i$ . For the bias, always use  $x = 1$ .

## 11.5 Converting real weights to integer

Any unit can be converted into one with integer weights.

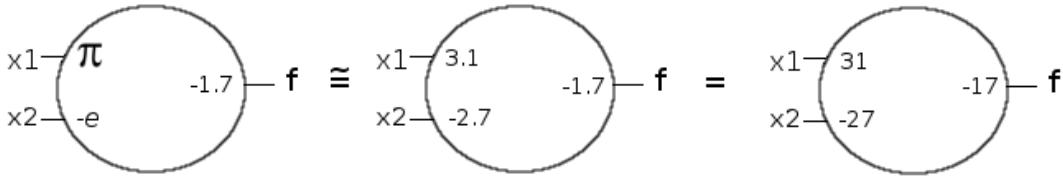


Figure 38: Converting real weights to integer

Figure 38 shows the approach to convert real weights to integer. The approximation (images left and center) can be problematic if  $x$  in  $\theta(x)$  is equal to zero (in the real setup), because it can be shifted in the approximation. In the second step (images center and right) there is no problem. By multiplying the weights by a factor of 10, we do not change the value of  $x$  in  $\theta(x)$ .

One way to solve the above problem (approximation) is to shift the bias term a bit.

1. Find the largest possible negative sum (closest to zero)
2. Adjust bias (increase) if necessary, so no sum is zero
3. Replace weights with rational approximations more precise than sum closest to zero
4. Scale up weights to be integers

In the first two steps, we avoid having a sum of zero. On the last two, we transform the weights to integer.

So far, the difference of this unit and a real neuron is that neurons can adjust their own weight (learning) and these units are behaving as gates.

## 12 Hopfield Networks

The Hopfield Network, or Hopfield Model, was proposed by John Hopfield in the early 80s. He was trying to understand what neural networks do. When we look into the brain and how neurons are connected to each other, we do not see always a clear pathway, i.e., it is hard to think about the connections in the brain in terms of input-output. He thought about connecting units to each other in an all-to-all pattern and see what would happen.

### Characteristics of Hopfield Networks

- Every node is connected to every other node but not to itself.
- Connection weights are symmetric (bidirectional).
- $\sum x_i w_i < 0$  represents an inactive unit,  $\sum x_i w_i \geq 0$  is active, when using a zero threshold.
- The entire network is in some state at any time. Set of active units of the entire network is important.
- Some states are stable and some are not. While in an unstable state, updating the network leads to a state change.
- Stable state is a local minimum.
- Bias is a unit that is always on.

- Weight symmetry of a connection is correlated to the frequency of neurons firing together (following Hebbian learning).

## 12.1 Hopfield and Memory

We are used to think about computation as a process that receives an input, does something and generates an output. However, this is not what we “see” in the memory process, for instance. It seems our memory works with Pattern Completion, also known as Content Addressable Memory or Associative Memory. This memory has no input-output relation: given any piece of it, we can recover the rest. An example is when certain smell can bring us a memory, even though the smell is only part of the memory. Hopfield Networks can be used to give us insights about how the memory works by having a highly connected network that computes with no input-output, but with states. The network has some dynamics that “walk” through a state space, i.e., the space of all the possible states. The Hopfield Network is dynamic and moves from one state to another until it arrives to a stable state (also known as an attractor). Having stable states help the network to recover information giving partial inputs.

- A Hopfield Network is an associative type of memory. Information is stored in the stable states as local minima.
- It is important that information is distinct.
- Associative memory has room for error but is still recognizable. Convergence to nearby stable states.
- If some units are retrievable and all others are set randomly, the correct units will eventually set the wrong units right.

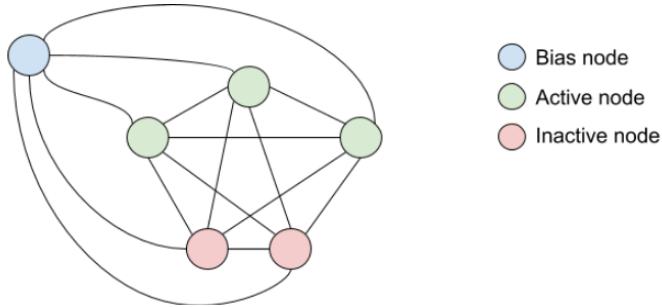


Figure 39: Model of a Hopfield Network. Red units are inactive, green units are active, blue unit is the bias node that is always active.

## 12.2 Updates and State Dynamics

Hopfield found that these networks always converge to a stable state. Idea: Consider the sum of weights between active units ( $Q$ ). The update rule is equivalent to always increase  $Q$ , i.e., we turn an unit active if  $\sum x_i w_i \geq \theta$ , where  $x_i$  is the value of the state of each unit,  $w_i$  is the weight between the units and  $\theta$  is the threshold, thus maximizing  $Q$  (weights between active units) by updating the output of one unit at time (see Figure 40). In this setup, the update algorithm of Hopfield Networks can be seen as a greedy algorithm to find the MAX-CLIQUE. We update all the units but the bias node (bias node are always active and do not change their state). Remember, inactive neurons with zero threshold don't send inhibitory signal, instead they do not take part in the activation of other neurons.

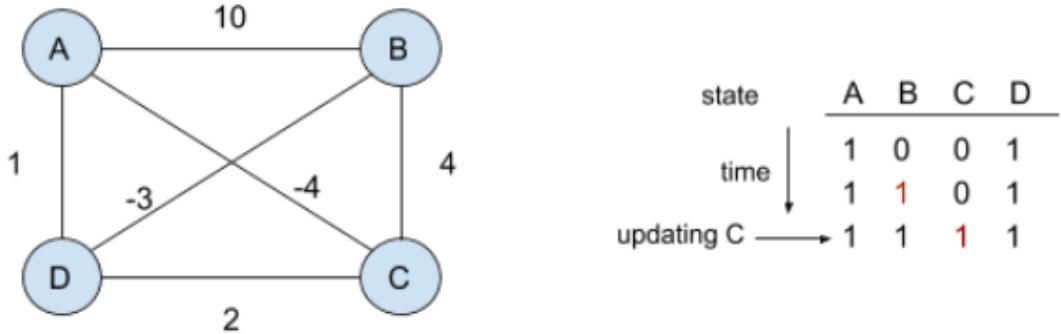


Figure 40: Hopfield network model, asynchronous updates and greedy algorithm to find MAX-CLIQUE

Consider the network presented in Figure 40: states are in the table on the right. In the beginning A and D are active, the sum of the active weights ( $Q$ ) is 1. Now, we look at B and we sum the weights of the active units linked to B (D and A).  $Q$  is  $(10 - 3 = 7 \geq 0)$ . If  $Q \geq 0$  we put the unit on the top part (unit active), otherwise we put the unit on the bottom (unit inactive), see Figure 41. If B becomes active,  $Q$  increases (sum = 1 + 7). If B becomes inactive,  $Q$  does not change (sum = 1). After updating C, the network achieves a stable state, i.e., updating any unit does not change its output.

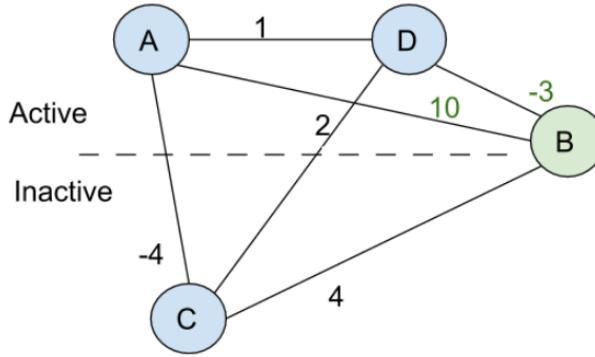


Figure 41: Evaluation of node B

When we update a unit and change its value (active or inactive), then  $Q$  increases or stays the same (if we are making units active). When we make units inactive,  $Q$  doesn't change. We can also consider active units as having a value of "+1" and inactive units as a "-1". In this new representation, the Hopfield Network dynamic is equivalent to a graph min-cut, i.e., we want to minimize the sum of weights that link active and inactive units.

**Asynchronous updates** When we update one unit at time, we are using an asynchronous method. Asynchronous Hopfield Networks always converge to a stable state, regardless of the update order, however, to which stable state is dependent on the update order, see Figure 42.

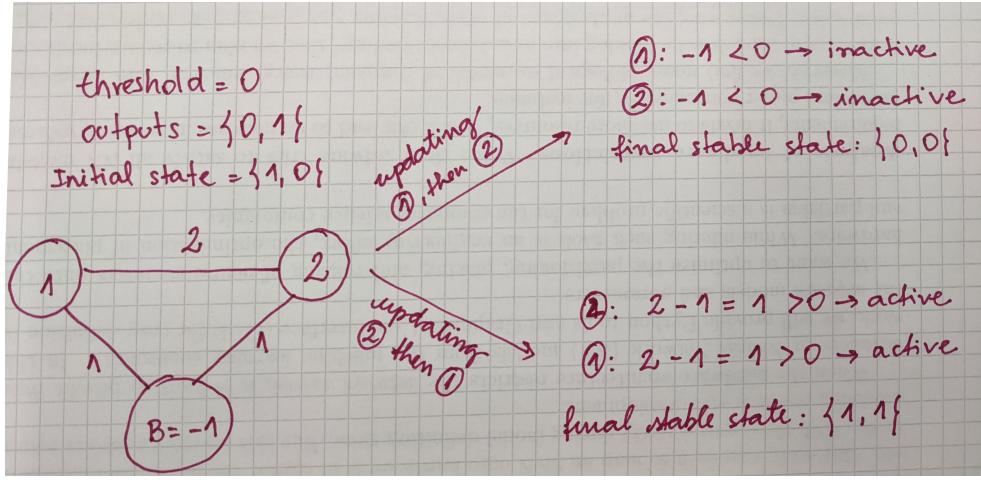


Figure 42: Considering a Hopfield network with two units connected with weight = 2, threshold zero, and a bias unit of  $-1$ . Starting in a state with 1 active and 2 inactive, we can see how the order of update can lead to different stable states. By updating first unit 1 then unit 2, the stable state is with both units inactive. However, updating unit 2 then unit 1, we arrive to a stable state where both units are active.

**Synchronous updates** Let's now consider a synchronous case, i.e., when we update all units at the same time: for this, active units for time  $t + 1$  are computed based on active units at time  $t$ . In this case, the system becomes a deterministic system because it doesn't depend on the order of updates anymore. Consider the network on Figure 43, starting with units 1 and 2 active, in the next step of update (after all units), unit 3 and 4 should be active. It may be confusing to see this by updating one unit per time, let's say we update units in the order 1, 2, 3, 4: after updating unit 3 it will become active but, when updating unit 4, we shouldn't consider unit 3, yet.

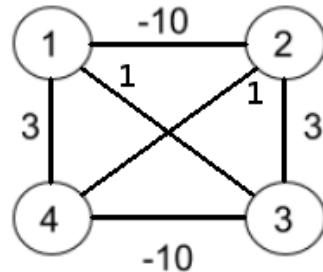


Figure 43: Starting with 1 and 2 actives, in the next step 3 and 4 will be active. This network doesn't have a unique stable state but converges to a cycle between two states.

Trick for analysis: make a larger asynchronous network based on the network we want to analyse. Duplicate the units in two columns and only use non-zero weights between them.

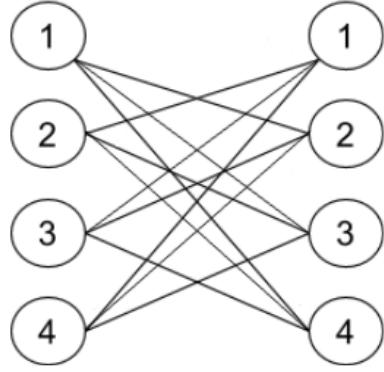


Figure 44: Starting with 1 and 2 actives, we will end up with 3 and 4 active in the second column. First column represents  $t = 0$ , second  $t = 1$ . With synchronous updates, a Hopfield Network converges to a cycle of length 2 or to a stable state.

### Summary

- Nodes can be updated synchronously or asynchronously.
- State: Set of units that are active, it can be seen as the activity vector.
- Dynamics: Units update their activity level.
- When a node is updated, weights are considered from all other active nodes, like with a perceptron.
- Asynchronous updates (greedy algorithm) converge to a stable state (sequential), but the converged state can depend on update order.
- Asynchronous is either in max-clique state if activity is in  $\{0, 1\}$  or min-cut if activities are in  $\{-1, 1\}$ .
- Synchronous, parallel updates either also go to a stable state, just like asynchronous, or can get stuck in a pair of patterns (flipping or cyclic).

## 13 Feed-Forward Networks

Feed-forward networks (FFN) are not like the networks in the brain. In the feed-forward networks, the information moves in only one direction (forward) from the input nodes, through the hidden nodes (if they exist), and to the output nodes. **There are no cycles in this network.** Usually, people are referring to feed-forward networks when they talk about artificial neural networks (ANN).

General structure:

- Multiple layers of neurons with a certain number of inputs and outputs.
- Every layer of nodes feeds the next layer with inputs.
- Each input in one layer is connected to all the inputs in the next layer.
- There is an input and an output layer with hidden layers in between.

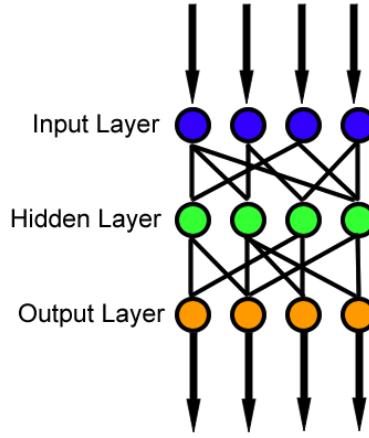


Figure 45: Example of a feed-forward network with an input layer, a hidden layer and an output layer.

A single unit, like a perceptron, can be seen as a feed-forward network.

We can write down the connections of a FFN as a matrix of weights, so  $w_{ij}$  is the weight from  $i$  to  $j$ .

Why FFN are nice? Because we can think about functions that receive inputs and generate outputs. When we use FFN we know what we want to compute. We need to set the weights of the network in order to compute the function we want. The process of defining the weights is called learning or training.

In feed-forward networks it is easy to evaluate each unit. The outputs are continuous functions of the input, which facilitates the optimization in case of wrong outputs. The training can be done using "training data": input/output pairs  $(x_i, d_i)$ . Where  $x_i$  is the input value and  $d_i$  is the desired output. Then we can define the error  $E = \sum_k (f(x_k) - d_k)^2$ , where  $f(x_k)$  is the output of the network. Differently of hopfield networks, we don't need continuous updates and we do not reevaluate units. FFN have the idea of a pipeline (not like the brain).

If a node on layer  $n$  in a FFN is connected to layer  $n + i$  with  $i > 1$ , this is still a FFN, however the most common structure is to connect nodes on layer  $n$  to nodes on layer  $n + 1$ .

How can we change the weights to reduce the error? We can use gradient descent. We calculate all the  $\frac{\partial E}{\partial w_i}$  in the network, easily, by starting at the end and then walking backwards. This is known as Backpropagation.

### Training process

1. Give to the network an input.
2. Calculate all the values in the network to produce the output.
3. Compare output with the desired output. This gives us the "error" for this input.
4. Update the weights. For example, use gradient descent to optimize the parameters to minimize the error.

## 13.1 Backpropagation and Error function

Backpropagation is the process of calculating the derivatives, using the chain rule, from the last layer (directly connected with the output, thus, with the loss function) to the first layer (connected with the inputs). This process can be seen as walking through the network in a backwards manner.

- The inputs and desired outputs are given as  $S = \{(x, d)^1, \dots, (x, d)^l\}$ .
- The error function is given as  $E(S) = \sum_i \frac{1}{2} \|y(x^i) - d^i\|^2$ .
- The output is a non-linear transformation  $y = f(a)$ .
- $f(a)$  is the activation function, which is usually a sigmoid function.
- The error function for a single training sample is  $E(S) = \frac{1}{2}(f(x_1w_1 + x_2w_2 + w_0) - d)^2$ .

- The output of a simple network is for example  $y(x_1, x_2, x_3) = f(x_1w_{21} + f(x_2w_{11} + x_3w_{12} + w_{10})w_{22} + w_{20})$ .
- $\frac{\partial E(w_1, w_2, w_0)}{\partial w_1} = (f(x_1w_1 + x_2w_2 + w_0) - d) \cdot f'(x_1w_1 + x_2w_2 + w_0) \cdot x_1$ .
- $\frac{\partial E(w_1, w_2, w_0)}{\partial w_2} = (f(x_1w_1 + x_2w_2 + w_0) - d) \cdot f'(x_1w_1 + x_2w_2 + w_0) \cdot x_2$ .
- $\frac{\partial E(w_1, w_2, w_0)}{\partial w_0} = (f(x_1w_1 + x_2w_2 + w_0) - d) \cdot f'(x_1w_1 + x_2w_2 + w_0)$ .
- The error terms travel backwards through the network and get multiplied with the derivative of the activation function of that input. Multiple error terms can just be added up.
- The partial derivative of the error  $E$  term in relation to the weight  $w$  to be adjusted can be added to the weight in order to learn. An additional weighting factor can be added.

## 13.2 Gradient Descent

Consider  $E = \sum(f_w(x) - y_i)^2$ , we want to adjust  $\vec{w}$  (weights of the network) to minimize  $E$ . Gradient descent is the process of descending through the gradients (using the derivatives calculated with back-propagation), in this algorithm we try to reach the minimum of the loss function. This is an iterative process.

Generally, the iterative process is given by  $\theta_{i_{new}} = \theta_i + \Delta\theta_i$ , where  $\Delta\theta_i = -\alpha \frac{\partial J(\theta)}{\partial \theta_i}$ .

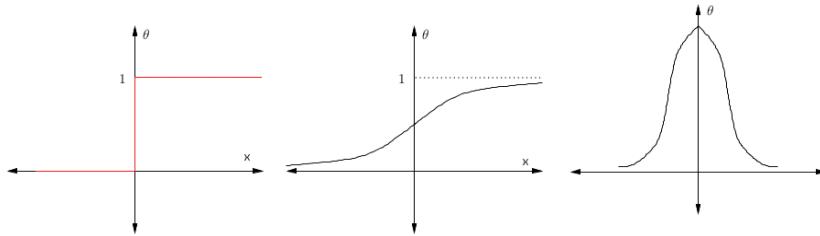


Figure 46: left-most figure:  $f_{w_0}(x) = \theta(wf + w\dots)$ ,  $\frac{df}{dw} = 0 \rightarrow \theta$  is not a good threshold function. To know in what direction we should move to find our minima, we need to use a threshold function that is continuous and differentiable, like the one in the center figure. Right figure:  $\frac{df}{dw} = \theta'$ .

We haven't found biological mechanism that would be similar to gradient descent in the brain.

## 13.3 Boltzmann Machines

Boltzmann Machines were invented in 1985 but not by Boltzmann. The name is given because these units use a Boltzmann distribution in their sampling function.

These units are similar to Hopfield Networks, however, they have a probability of being active. When updating a unit, we set its value to zero or one probabilistically, following a sampling function, see Figure 47. Boltzmann machines do not converge, they do not reach a stable state. It can be seen as a system for sampling. It's a way to do a random walking in the state space.

**Sampling** There are many types of distributions. When we want to get examples of these distributions, we need to sample from it. And giving some samples we can recover the distribution of the data.

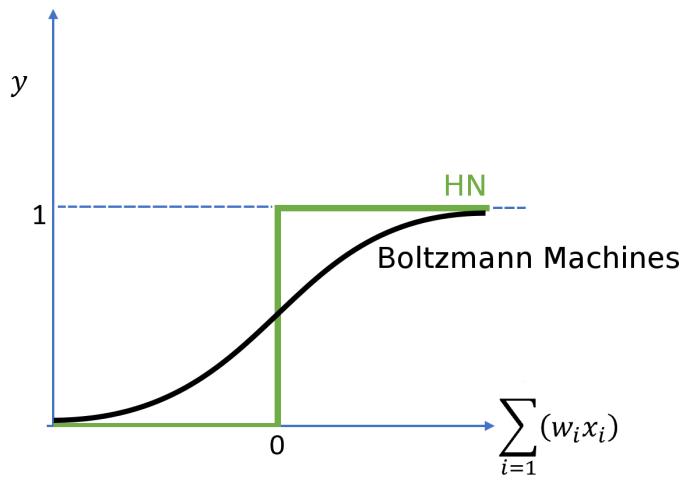


Figure 47: Representations of activation function for Hopfield Networks (HN) and Boltzmann Machines

We want the units to forget previous states so the sampling is not biased, i.e., not similar to previous ones, thus really "random". While this sampling is nice, it is not useful as a memory, i.e., we don't want to sample things randomly from our memory.

We can define a network where the units have a real-valued activity level ( $a_i \in [0, 1]$ ), and also we can make time continuous, so  $\frac{\partial a_i}{\partial t} = \theta(\Sigma) - a_i$ .

Using units like this, we can make a feed-forward network.

## 14 Interacting Neural Populations

We know how to think about units and their computation, but we don't exactly know how to think about how the information is processed . To try to understand it, we use engineering tools, but we don't know how the brain is doing it.

### 14.1 Neural Encoding of Information

How the neurons encoding/process information? We would like to understand it in a mechanistic level. To understand the brain, we need to understand its structure and functionality (processes). Different areas in the brain are highly connected, practically from any area you can reach, virtually, any other area.

Humans are capable to learn things that we were not evolved for, for instance, to fly a drone. This is an inspiration for us. It seems to exist a general solution for solving problems: the brain.

Neurons respond to **combinations** of properties, features, aspects of the situation, etc. Neurons are tuned to a set of values for the parameters they care about. For example, a neuron can be "tuned" to a moving bar at a certain angle in its perceptive field. It also responds to a certain velocity, position ( $x, y$ ), bar width, etc.

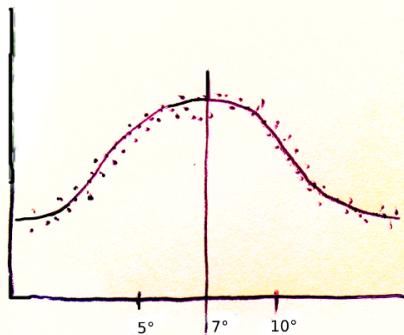
Neurons response is called firing rates. Although neurons are tuned, they aren't super picky about the exact values. And, experimentally, neurons do not code a single attribute but a combination of them.

**Population code** Information is encoded by a group (population) of neurons. A group includes all neurons in that area. The values are encoded by the pattern of activity. This code of information by population is our understanding of how neurons represent information, but this is not how we do it, typically, in an engineering process.

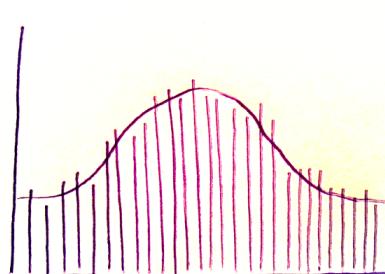
Given a parameter, and looking at the neurons tuned to some value of this parameter, we can define tuning curves. In Figure 48a, we see in the  $x$  axis the parameter we are evaluating, and in the  $y$  axis, the activity of one single neuron (firing response). Doing this process to many neurons, we can sort

them regarding to the parameter we are looking for. In Figure 48b all neurons were ordered accordingly to the response to a specific parameters.

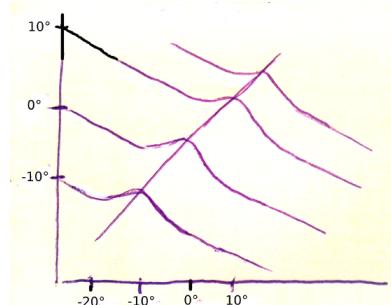
The read out of information given a population code is an easy way to figure out what the population is doing, it is robust to noise, however, requires a lot of units.



(a) Tuning curve of one cell



(b) Cells ordered by response to 20°



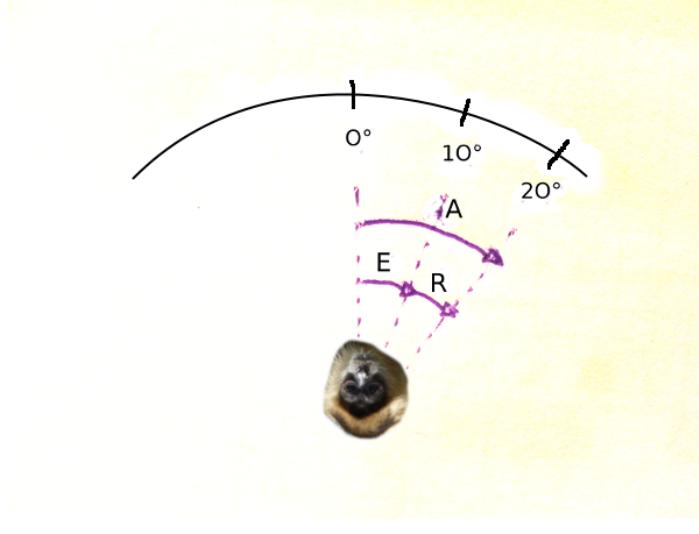
(c) 3D visualization of cell's response to different degrees

Consider now an example where we want to correlate the angle of the eye (E), the position of an object on retina (R) and the head-centered direction (A), as showed in Figure 49a. This is an example of processing that is not feed-forward.

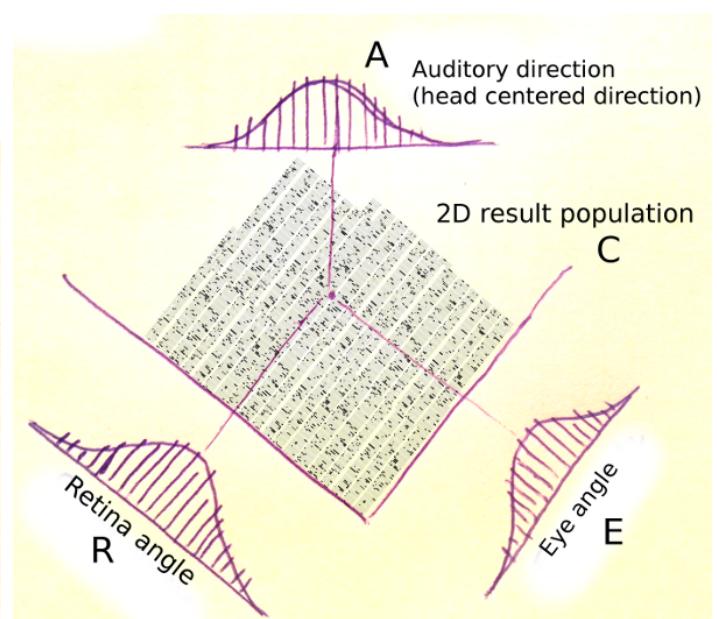
How can a relation like  $A = E + R$  be represented? Say  $R$ ,  $A$  and  $E$  are encoded by population codes, i.e., by population of units, each tuned to a particular value of that variable. Considering these three variables, we can use units that are tuned to combinations of  $A$ ,  $E$  and  $R$ . The set of  $(A, E, R)$  triples that satisfy the relation will be active. This can be seen in Figure 49b: when certain neuron fires for a parameter in  $R$ , another one can fire for a parameter in  $E$  and this can result in the firing of neurons in  $A$ , after learning. Given any two information in this relation, the third one can be recovered. There is no obligatory order.

$R$ ,  $A$  and  $E$  have the following relations:

- $R = A - E$
- $A = E + R$
- $E = A - R$



(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  $A$ .

## Summary

- Neurons can represent information through population codes.
- Neurons are tuned to preferred stimuli.
- Information is represented by the pattern of activity in a neural population.
- Each neuron has a preferred input, for example orientation, that it responds to. The neuron is tuned to that value.
- Not every neuron shows clear tuning curves.
- Neurons usually do not only respond to their preferred stimulus, but also with decaying strength to close ones.
- Population encoding has an easy read out value.
- Population encoding is robust to noise.
- Population encoding requires a lot of units.

## 14.2 References

The pictures used in this summary are from the following books and slide sets and belong to their respective owners. In the context of the summary they are used for educational purposes only.

- D. Purves et al.: Neuroscience. Fifth edition. Sinauer
- Squire, Berg, Bloom, du Lac, Ghosh & Spitzer: Fundamental Neuroscience. Elsevier
- Eric Kandel et al.: Principles of Neural Science. Fifth edition. McGraw-Hill
- Peter Dayan, L. F. Abbott: Theoretical Neuroscience, The MIT Press
- Christof Koch: Biophysics of Computation. Oxford University Press
- John Nicholls et al.: From Neuron to Brain. Palgrave Macmillan
- W. Gerstner et al.: Neuronal Dynamics. From Single Neurons to Networks and Models of Cognition. Cambridge University Press (2014).
- Hertz, Krogh, Palmer: Introduction to the Theory of Neural Computation. Westview Press
- Bishop: Pattern Recognition and Machine Learning. Springer
- The lecture slides of the Introduction to Neuroinformatics course of the years 2012, 2013, 2014, 2018 and 2019.