DESIGN OF BRAIN-INSPIRED COMPUTATIONAL MODELS MODELS AND THEIR APPLICATIONS

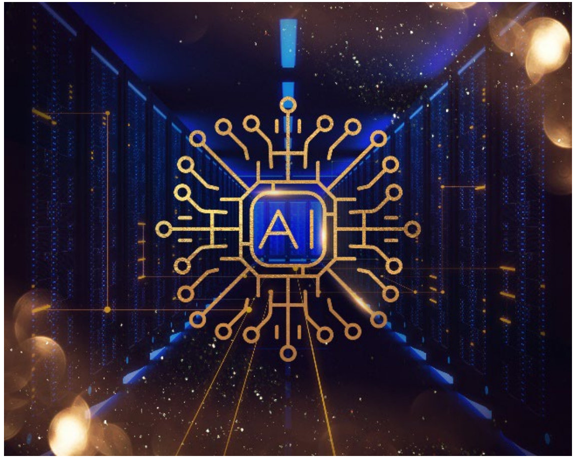
Petia Koprinkova-Hristova, Georgi Rusev, Svetlozar Yordanov

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences

Natural vs artificial intelligence

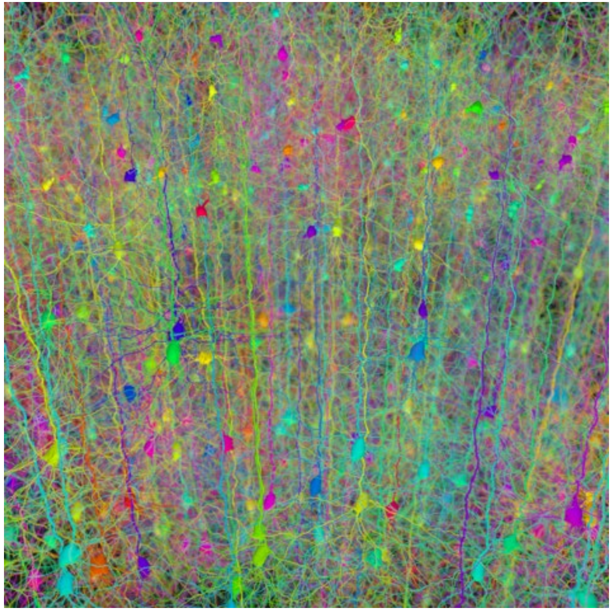
Intelligence (from Latin *intellectus* – knowledge, understanding) – the human ability to reason, analyze and synthesize information.

Artificial intelligence – the intelligence demonstrated by computers.

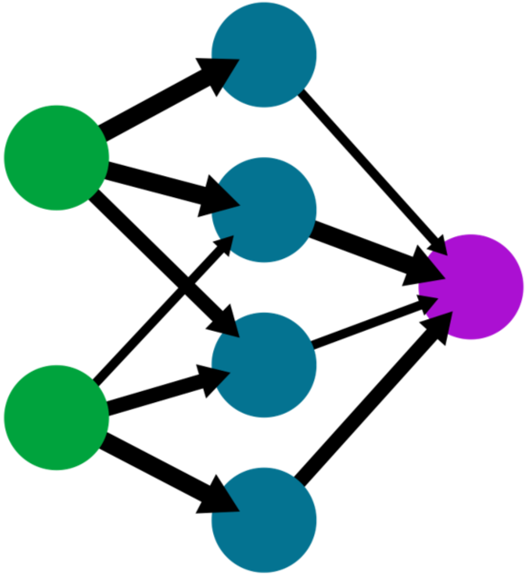


Neural networks

Natural NN

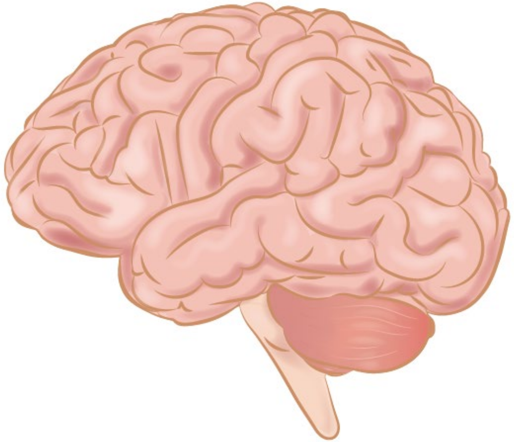


Artificial NN

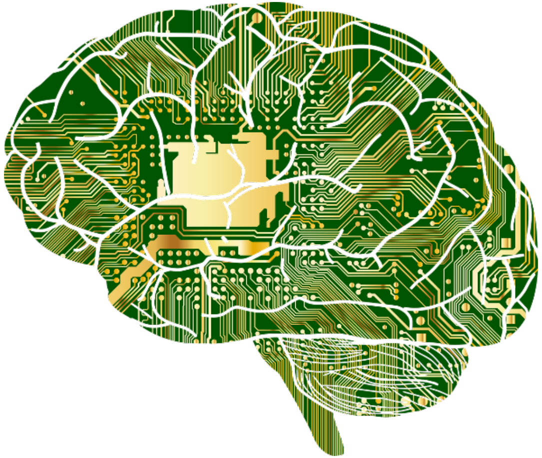


Material basis of natural intelligence - the brain

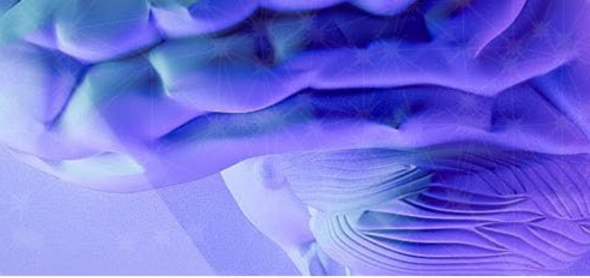
Natural intelligence includes all the biological systems controlling our behavior in interaction with the environment. **Neurobiology** studies the function of the brain as the material basis of natural intelligence.



One of the best approaches to studying **natural intelligence** is to try to replicate its behavior in simulation.



The brain as the governing organ of our body



It controls all the basic functions of our body

Interprets information coming from the outside world through our primary sensors: sight, hearing, smell, touch, taste and smell

Controls intelligence, creativity, emotions and memory

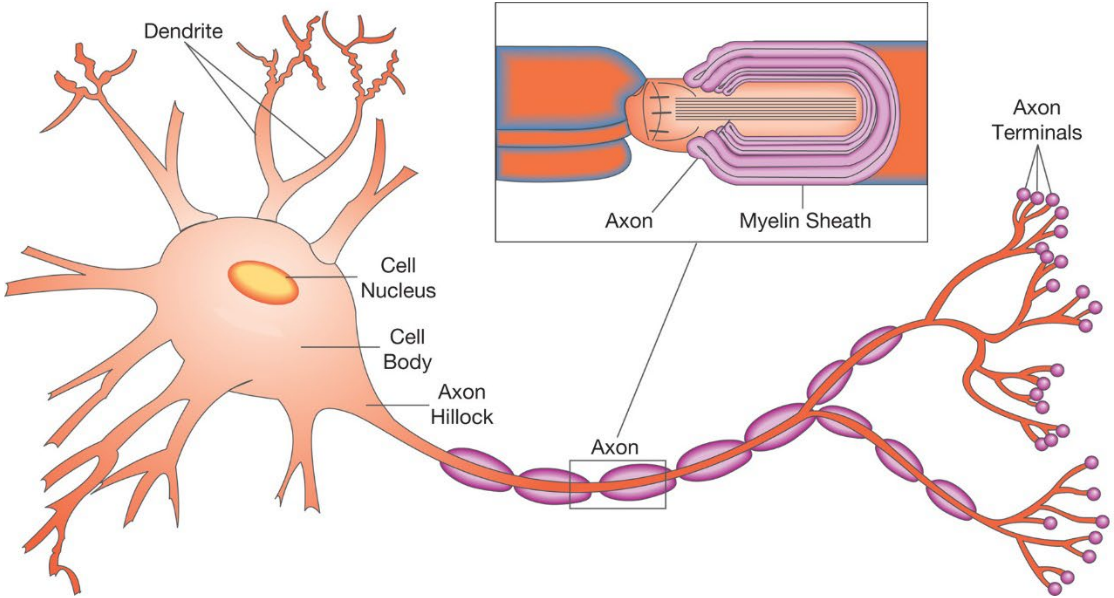
Neurobiology vs AI

1. The power of natural neurons:
Modern artificial neural networks are quite simplistic. Neurobiology's knowledge of how natural neurons work would allow a revolutionary expansion of the capabilities of artificial neural networks.

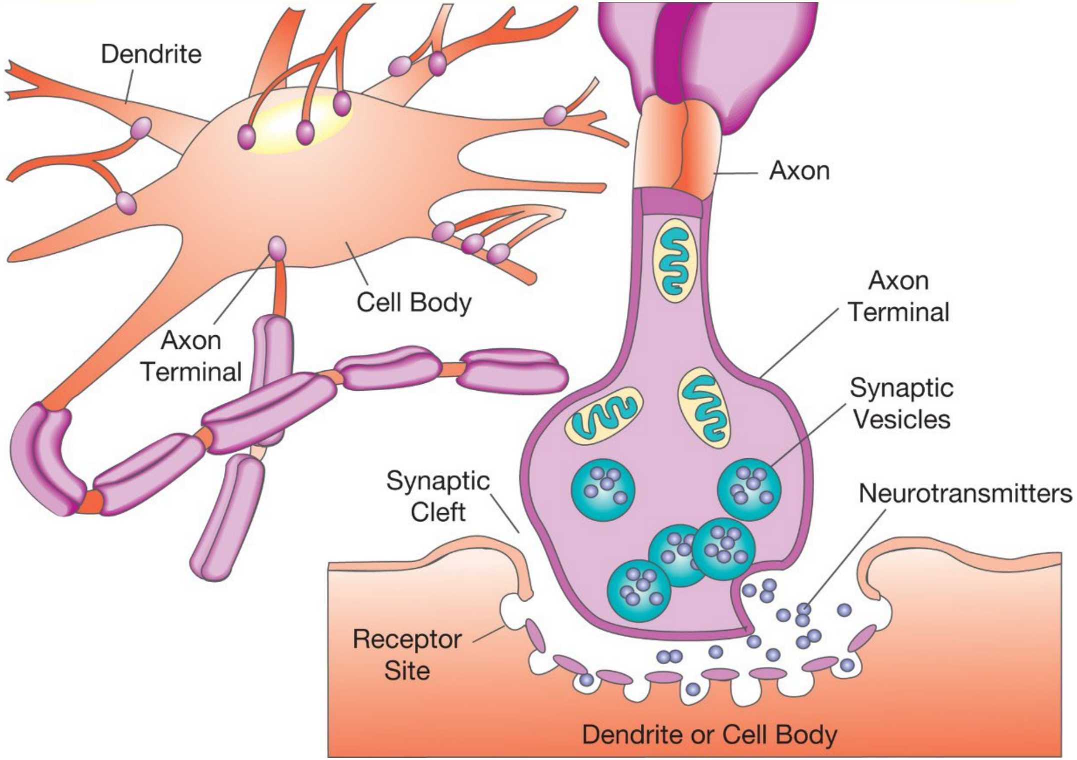
2. Structure of neural networks.
Accumulating knowledge about the complex deep structure of neural networks in the brain would allow the improvement of artificial neural networks.

3. Built-in cognitive abilities.
Knowing the human's innate ability to recognize and learn would allow systems to be enriched with artificial intelligence.

Neural cells - the building blocks of the brain

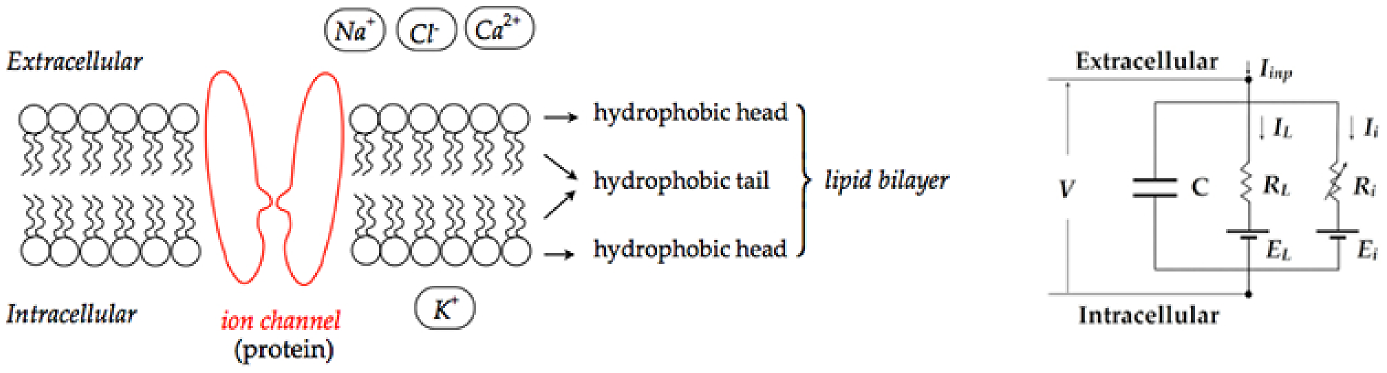


Synapses



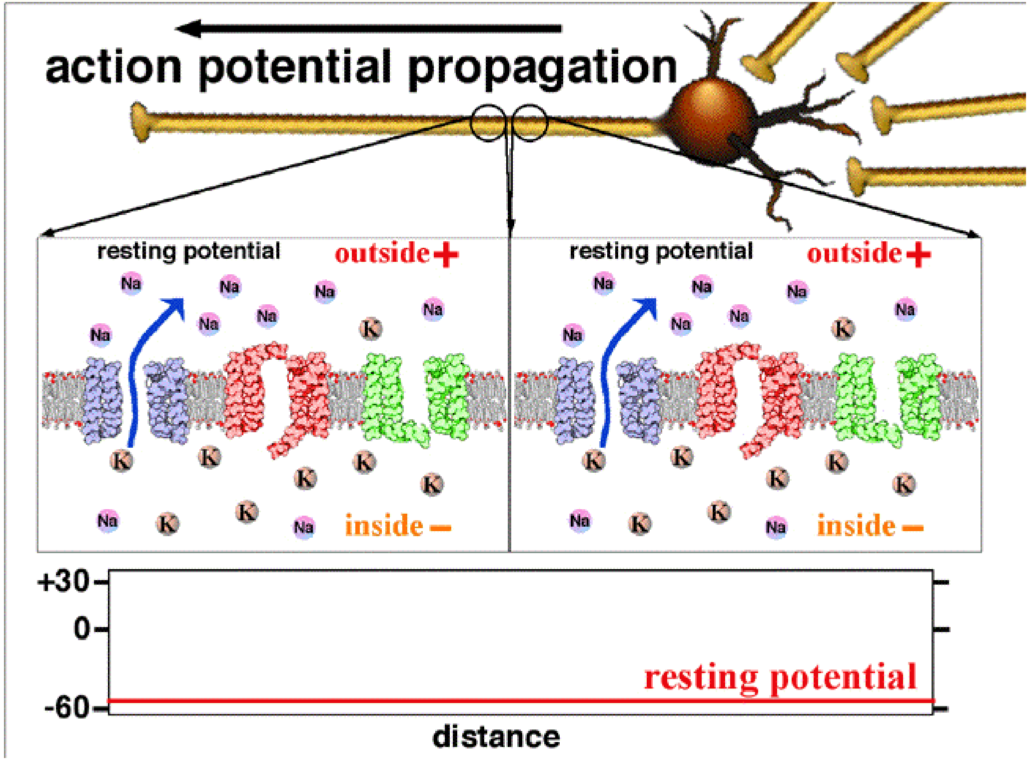
Dendrite
Axon
Cell Body
Axon Terminal
Axon Terminal
Synaptic Vesicles
Synaptic Cleft
Neurotransmitters
Receptor Site
Dendrite or Cell Body

Functioning of neurons

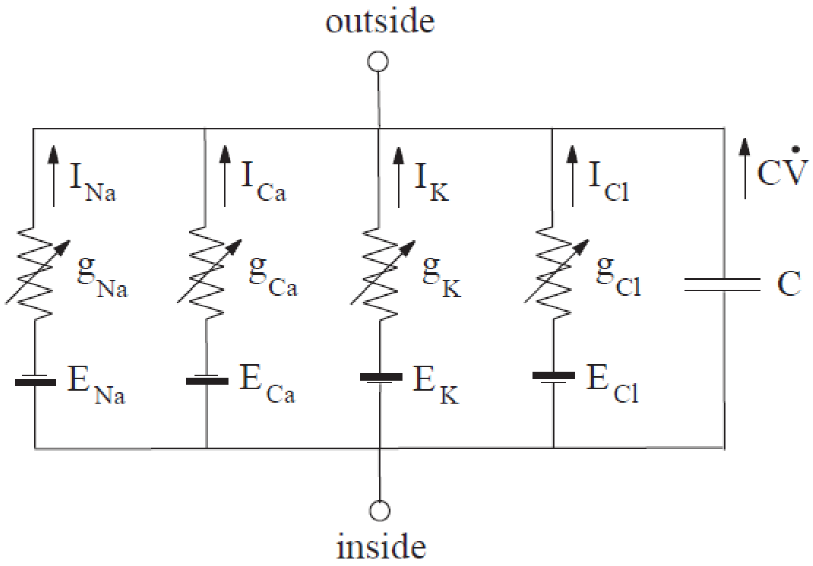


(a) (b)

Ion channels functioning



Ion channel models



$$I_{\mathrm{K}} = g_{\mathrm{K}}\,(V - E_{\mathrm{K}})$$

$$I_{\mathrm{Na}} = g_{\mathrm{Na}}\,(V - E_{\mathrm{Na}})$$

$$I_{\mathrm{Ca}} = g_{\mathrm{Ca}}\,(V - E_{\mathrm{Ca}})$$

$$I_{\mathrm{Cl}} = g_{\mathrm{Cl}}\,(V - E_{\mathrm{Cl}})$$

$$I = C\dot{V} + I_{\mathrm{Na}} + I_{\mathrm{Ca}} + I_{\mathrm{K}} + I_{\mathrm{Cl}}$$

Hodgkin-Huxley equations

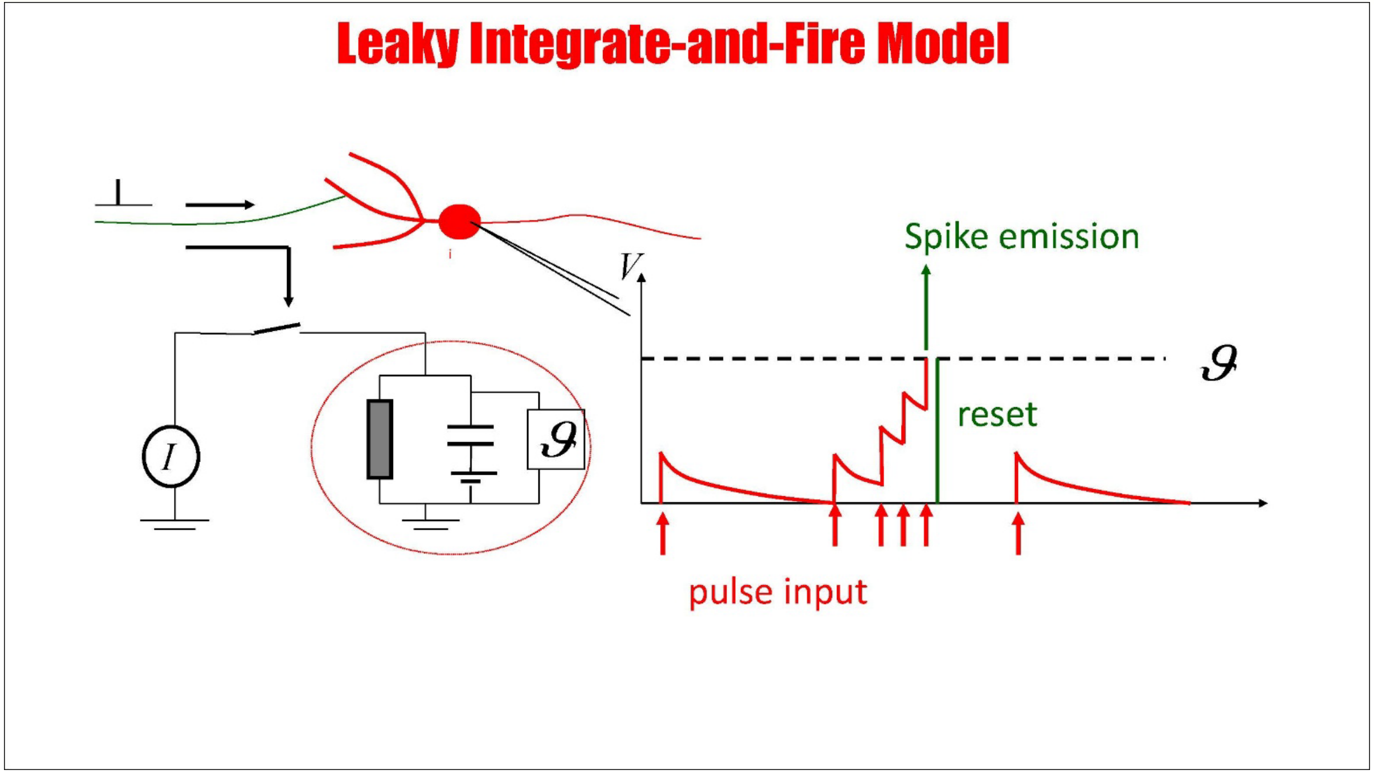
$$
\begin{aligned}
C\dot{V} &= I - \overbrace{\bar{g}_{\mathrm{K}} n^4 (V - E_{\mathrm{K}})}^{I_{\mathrm{K}}} - \overbrace{\bar{g}_{\mathrm{Na}} m^3 h (V - E_{\mathrm{Na}})}^{I_{\mathrm{Na}}} - \overbrace{g_{\mathrm{L}} (V - E_{\mathrm{L}})}^{\overset{\mathrm{Cl}^-}{I_{\mathrm{L}}}} \\
\dot{n} &= \alpha_n(V)(1-n) - \beta_n(V)n \\
\dot{m} &= \alpha_m(V)(1-m) - \beta_m(V)m \\
\dot{h} &= \alpha_h(V)(1-h) - \beta_h(V)h \ ,
\end{aligned}
$$

$$E_{\mathrm{K}} = -12 \text{ mV} \qquad E_{\mathrm{Na}} = 120 \text{ mV} \ , \qquad E_{\mathrm{L}} = 10.6 \text{ mV};$$

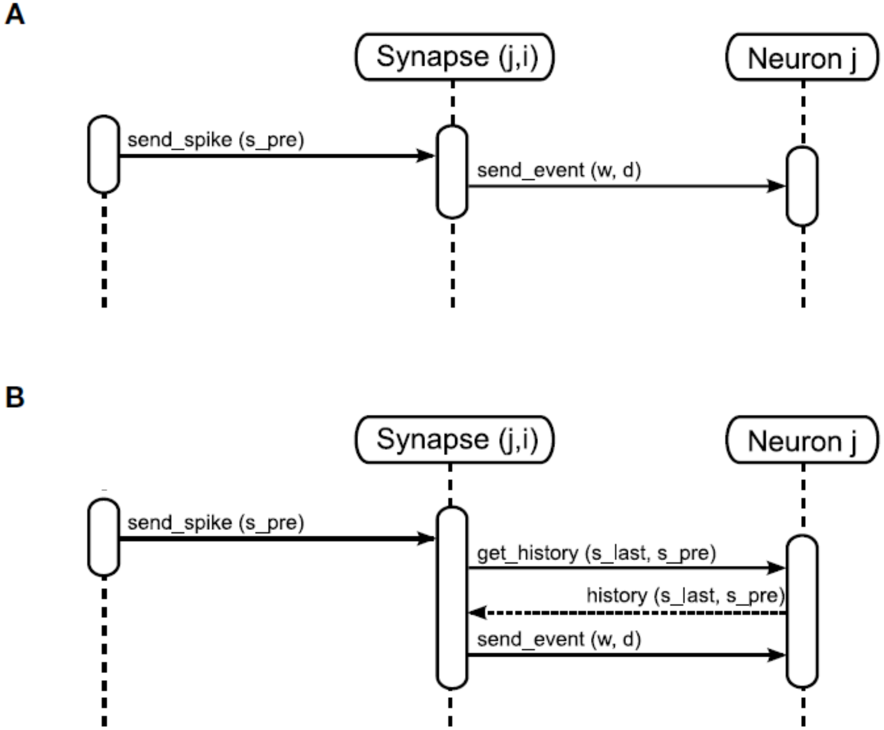
$$\bar{g}_{\mathrm{K}} = 36 \text{ mS/cm}^2 \qquad \bar{g}_{\mathrm{Na}} = 120 \text{ mS/cm}^2 \ , \qquad g_{\mathrm{L}} = 0.3 \text{ mS/cm}^2.$$

$$
\begin{aligned}
\alpha_n(V) &= 0.01 \frac{10 - V}{\exp(\frac{10-V}{10}) - 1} \\
\beta_n(V) &= 0.125 \exp\left(\frac{-V}{80}\right) \\
\alpha_m(V) &= 0.1 \frac{25 - V}{\exp(\frac{25-V}{10}) - 1} \\
\beta_m(V) &= 4 \exp\left(\frac{-V}{18}\right) \\
\alpha_h(V) &= 0.07 \exp\left(\frac{-V}{20}\right) \\
\beta_h(V) &= \frac{1}{\exp(\frac{30-V}{10}) + 1} \ .
\end{aligned}
$$

Simplest IAF neuron model



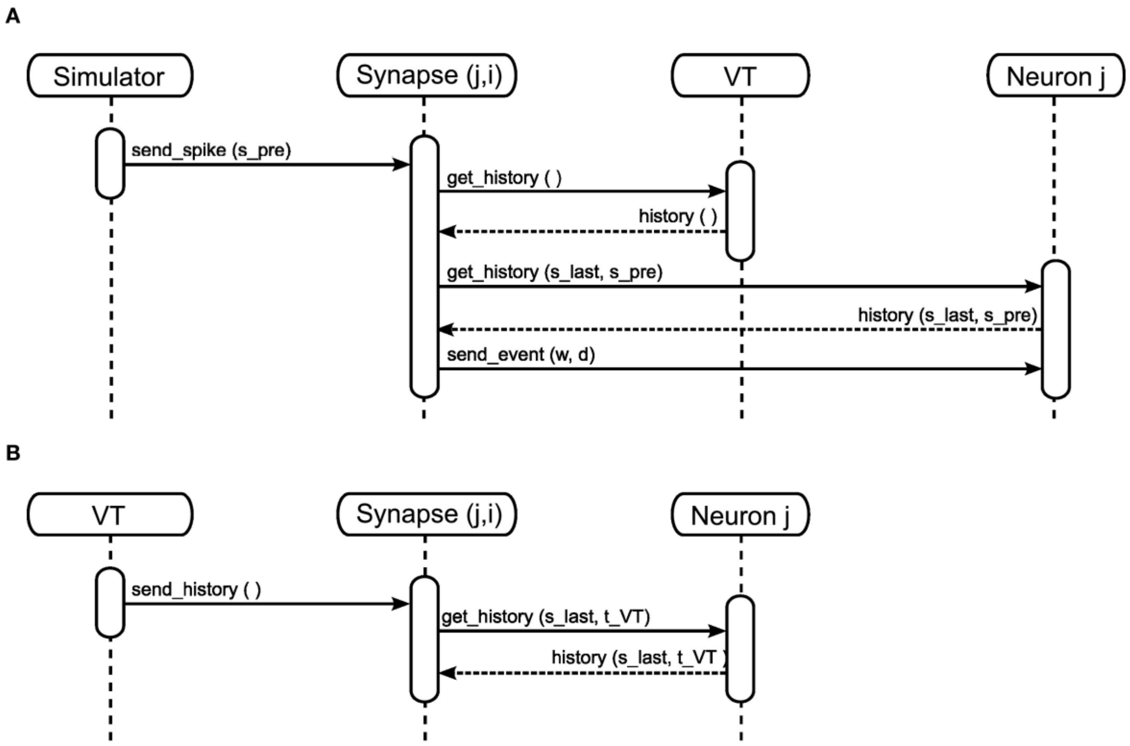
Synapses adaptation



A. Static synapse

B. Dynamic synapse

Synapses adaptation



A. Event-driven adaptation

B. Neuromodulator-driven adaptation

Event-driven adaptation

$$\Delta w = \begin{cases} -\lambda f_{-}(w) \times K(\Delta t), & \Delta t \le 0 \\ \lambda f_{+}(w) \times K(\Delta t), & \Delta t > 0 \end{cases}$$

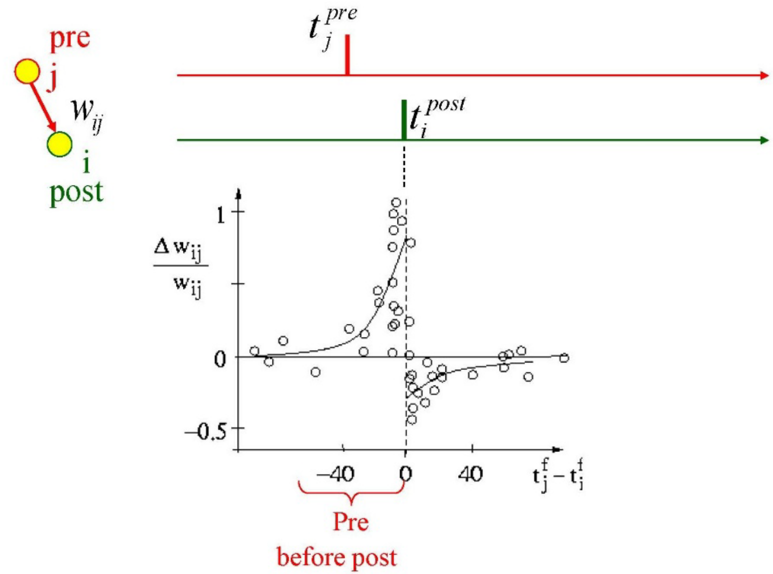
$$\Delta t = t_{post} - t_{pre}$$

$$K(\Delta t) = e^{-|\Delta t|/\tau}$$

$$0 \le \lambda << 1$$

$$f_{+}(w) = (1 - w)^{\mu}$$

$$f_{-}(w) = \alpha w^{\mu}$$



Neuromodulator-driven adaptation

$$\dot{w} = c(n - b)$$

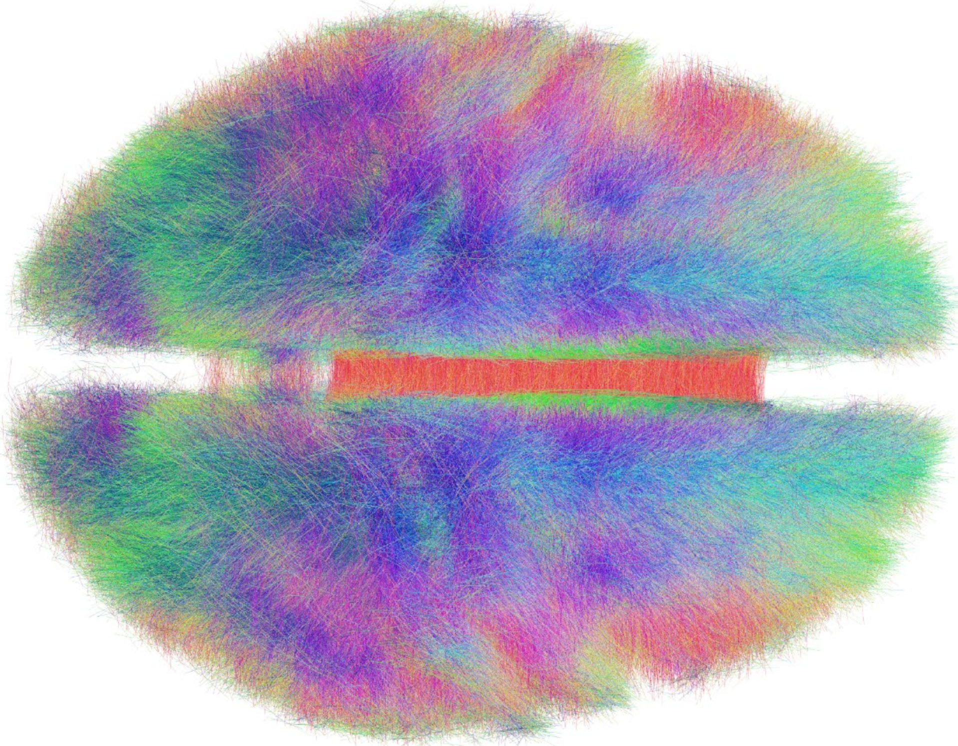
$$\dot{c} = -\frac{c}{\tau_c} + STDP(\Delta t)\delta(t - t_{pre/post})C_1$$

$$\dot{n} = -\frac{n}{\tau_n} + \frac{\delta(t - t_n)}{\tau_n}C_2$$

$$STDP(\Delta t) = \begin{cases} A_+ e^{-|\Delta t|/\tau_+}, & \Delta t > 0 \\ A_- e^{-|\Delta t|/\tau_-}, & \Delta t \le 0 \end{cases}$$

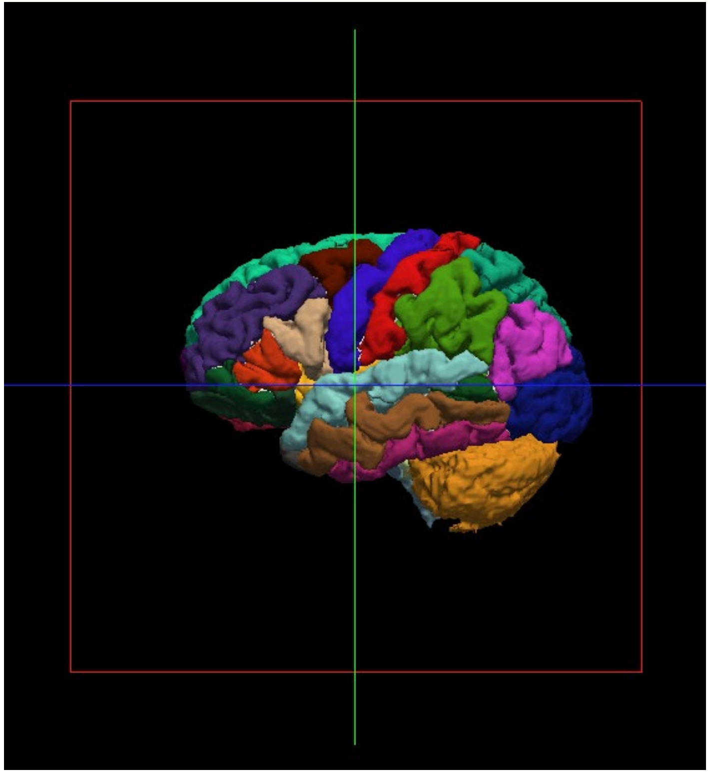
$$\Delta t = t_{post} - t_{pre}$$

Brain atlases



Brain atlases

Desikan-Killiany



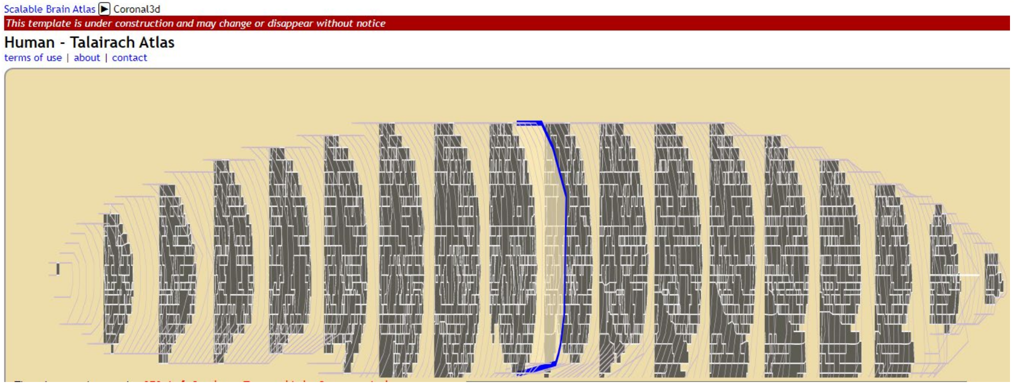
Desikan-Killiany atlas region list

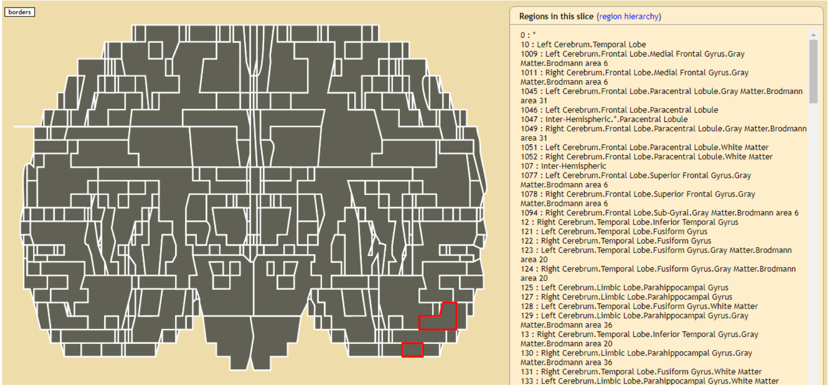
Left Hemisphere			Right Hemisphere		
ID	Abbreviation	DK Region Name	ID	Abbreviation	DK Region Name
1	L.BSTS	ctx-lh-bankssts	43	R.TH	Right-Thalamus-Proper
2	L.CACG	ctx-lh-	44	R.CA	Right-Caudate
3	L.CMFG	ctx-lh-caudalmiddlefrontal	45	R.PU	Right-Putamen
3	L.CU	ctx-lh-cuneus	46	R.PA	Right-Pallidum
4	L.EC	ctx-lh-entorhinal	47	R.HI	Right-Hippocampus
5	L.FG	ctx-lh-fusiform	48	R.AM	Right-Amygdala
6	L.IPG	ctx-lh-inferiorparietal	49	R.AC	Right-Accumbens-area
8	L.ITG	ctx-lh-inferiortemporal	50	R.BSTS	ctx-rh-bankssts
9	L.ICG	ctx-lh-isthmuscingulate	51	R.CACG	ctx-rh-
10	L.LOG	ctx-lh-lateraloccipital	52	R.CMFG	ctx-rh-caudalmiddlefrontal
11	L.LOFG	ctx-lh-lateralorbitofrontal	53	R.CU	ctx-rh-cuneus
12	L.LG	ctx-lh-lingual	54	R.EC	ctx-rh-entorhinal
13	L.MOFG	ctx-lh-medialorbitofrontal	55	R.FG	ctx-rh-fusiform
14	L.MTG	ctx-lh-middletemporal	56	R.IPG	ctx-rh-inferiorparietal
15	L.PHIG	ctx-lh-parahippocampal	57	R.ITG	ctx-rh-inferiortemporal
16	L.PaCG	ctx-lh-paracentral	58	R.ICG	ctx-rh-isthmuscingulate
17	L.POP	ctx-lh-parsopercularis	59	R.LOG	ctx-rh-lateraloccipital
18	L.POR	ctx-lh-parsorbitalis	60	R.LOFG	ctx-rh-lateralorbitofrontal
19	L.PTR	ctx-lh-parstriangularis	61	R.LG	ctx-rh-lingual
20	L.PCAL	ctx-lh-pericalcarine	62	R.MOFG	ctx-rh-medialorbitofrontal
21	L.PoCG	ctx-lh-postcentral	63	R.MTG	ctx-rh-middletemporal
22	L.PCG	ctx-lh-posteriorcingulate	64	R.PHIG	ctx-rh-parahippocampal
23	L.PrCG	ctx-lh-precentral	65	R.PaCG	ctx-rh-paracentral
24	L.PCU	ctx-lh-precuneus	66	R.POP	ctx-rh-parsopercularis
25	L.RACG	ctx-lh-	67	R.POR	ctx-rh-parsorbitalis
26	L.RMFG	ctx-lh-rostralmiddlefrontal	68	R.PTR	ctx-rh-parstriangularis
27	L.SFG	ctx-lh-superiorfrontal	69	R.PCAL	ctx-rh-pericalcarine
28	L.SPG	ctx-lh-superiorparietal	70	R.PoCG	ctx-rh-postcentral
29	L.STG	ctx-lh-superiortemporal	71	R.PCG	ctx-rh-posteriorcingulate
30	L.SMG	ctx-lh-supramarginal	72	R.PrCG	ctx-rh-precentral
31	L.FP	ctx-lh-frontalpole	73	R.PCU	ctx-rh-precuneus
32	L.TP	ctx-lh-temporalpole	74	R.RACG	ctx-rh-
33	L.TTG	ctx-lh-transversetemporal	75	R.RMFG	ctx-rh-rostralmiddlefrontal
34	L.IN	ctx-lh-insula	76	R.SFG	ctx-rh-superiorfrontal
35	L.CER	Left-Cerebellum-Cortex	77	R.SPG	ctx-rh-superiorparietal
36	L.TH	Left-Thalamus-Proper	78	R.STG	ctx-rh-superiortemporal
37	L.CA	Left-Caudate	79	R.SMG	ctx-rh-supramarginal
38	L.PU	Left-Putamen	80	R.FP	ctx-rh-frontalpole
39	L.PA	Left-Pallidum	81	R.TP	ctx-rh-temporalpole
40	L.HI	Left-Hippocampus	82	R.TTG	ctx-rh-transversetemporal
41	L.AM	Left-Amygdala	83	R.IN	ctx-rh-insula
42	L.AC	Left-Accumbens-area	84	R.CER	Right-Cerebellum-Cortex

DK = Desikan-Killiany atlas, ctx = cortex, lh = left hemisphere, rh = right hemisphere

Brain atlases

Thalairach





Examples

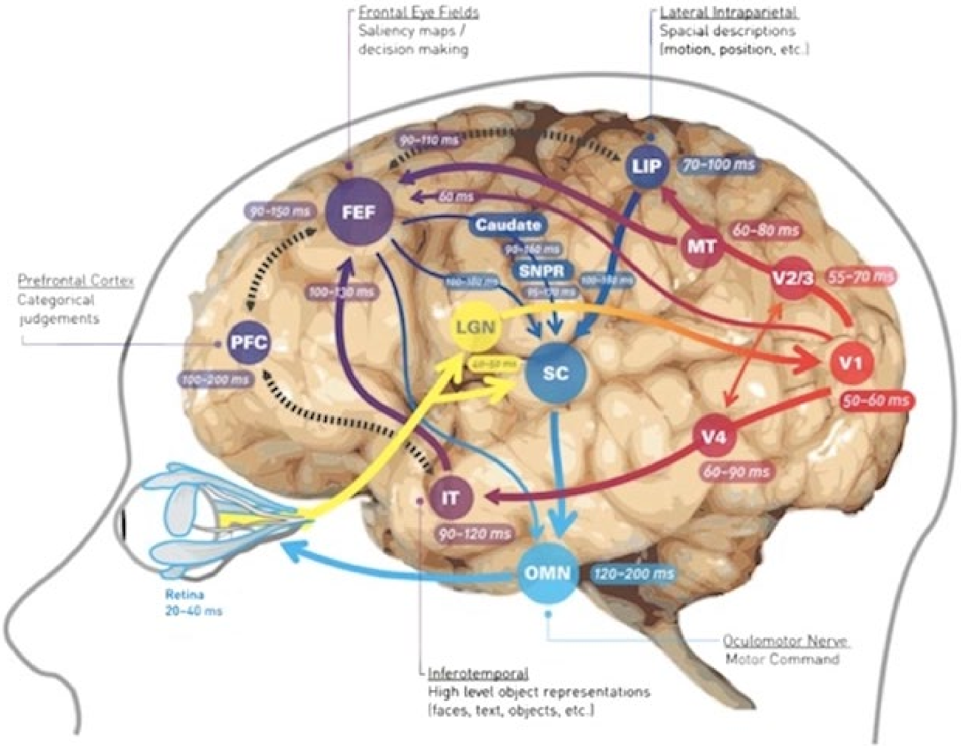
- SNN model of visual information processing and decision making
- SNN for brain signals decoding

Example 1

SNN model of visual information processing and decision making

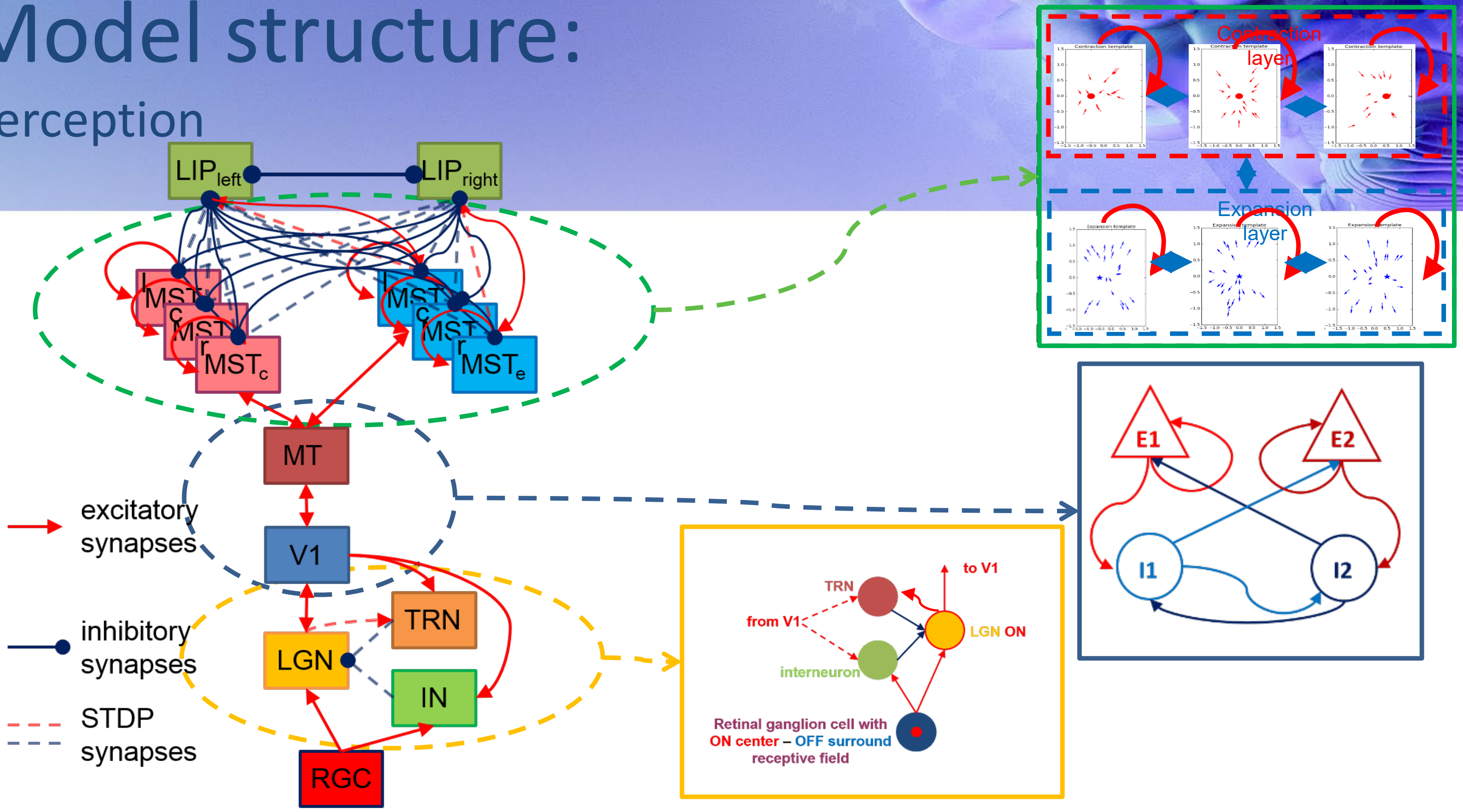
Brain models

Visual information processing

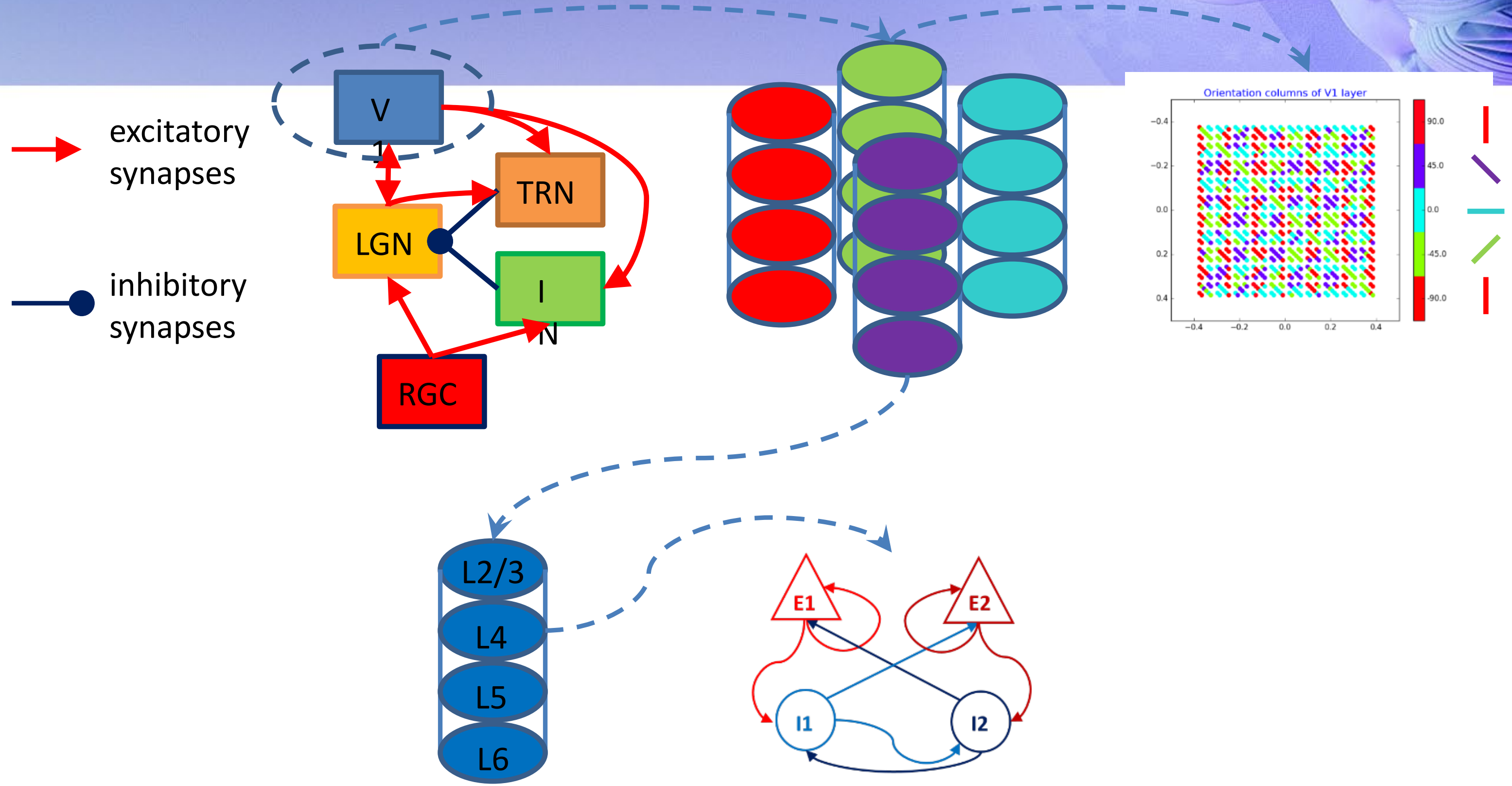


Model structure:

perception

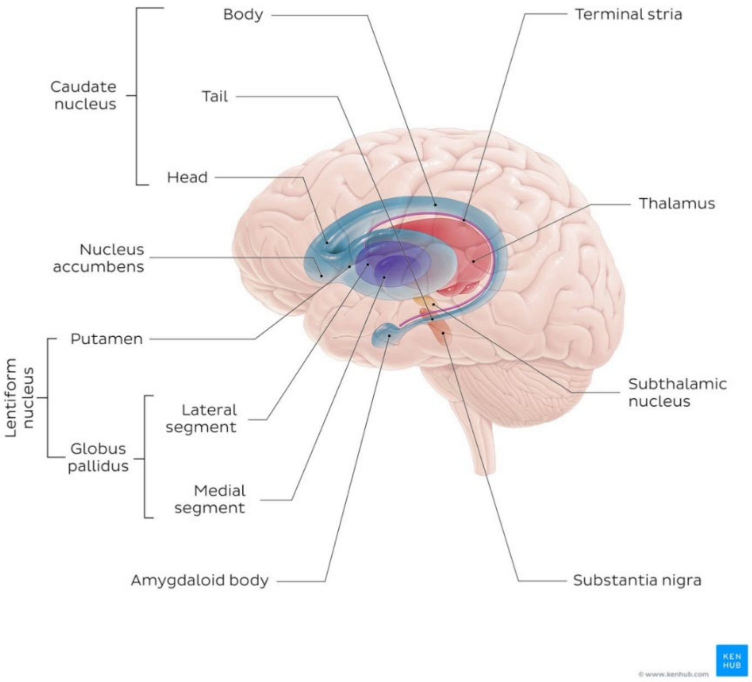


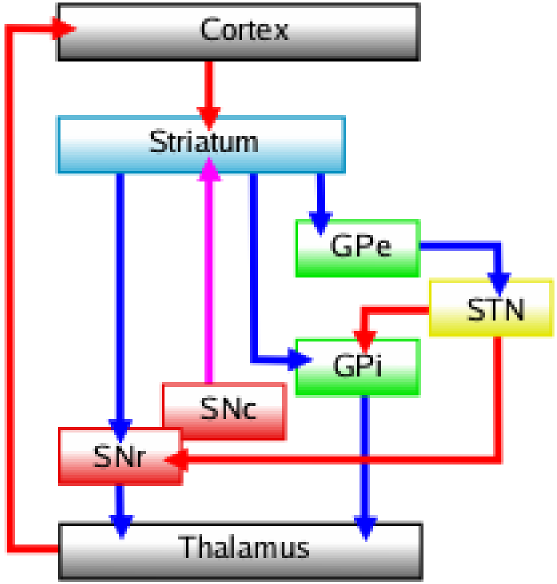
Visual cortex model



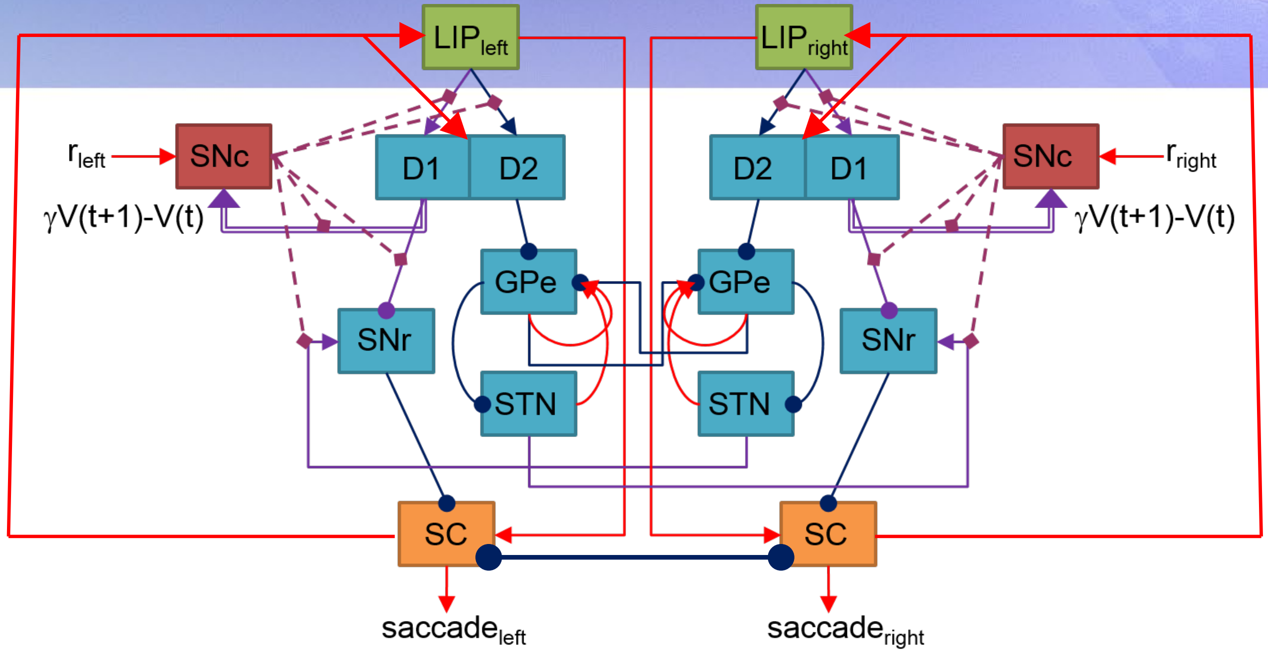
Brain models

Decision making and basal ganglia





Model structure



excitatory synapses

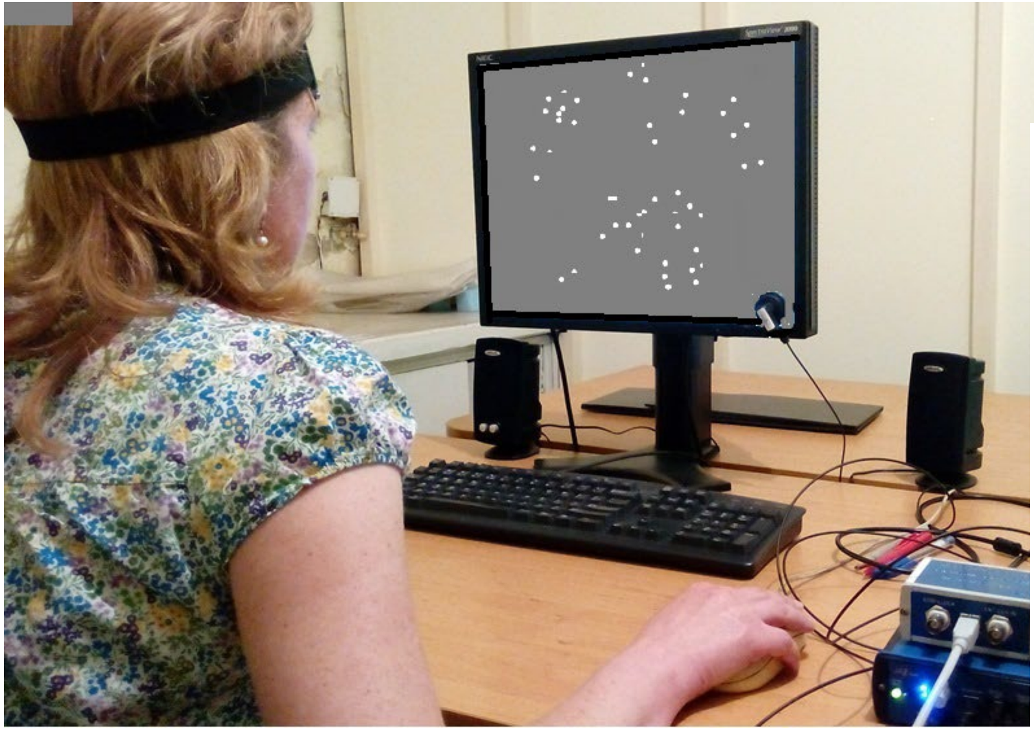
inhibitory synapses

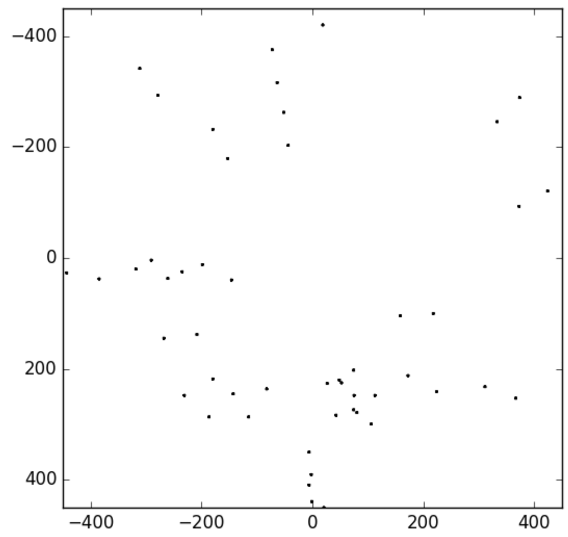
dopamine synapses

anti-dopamine synapses

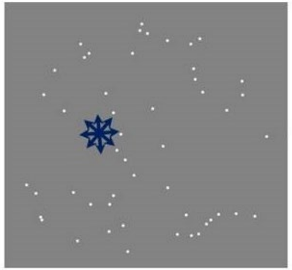
$$\delta(t) = SNc = F(r(t) + \gamma D1(t+1) - D1(t))$$

Experimental set-up

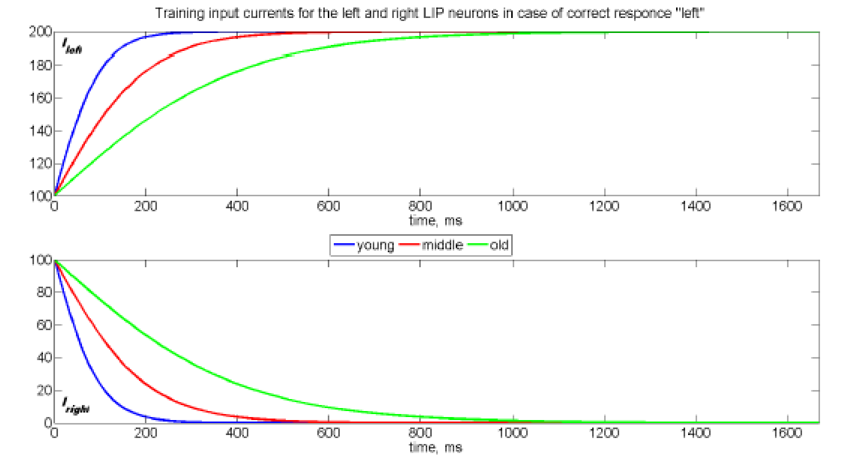




Training approach: perception and reaction time



left



TRAINING SIGNAL PARAMETERS

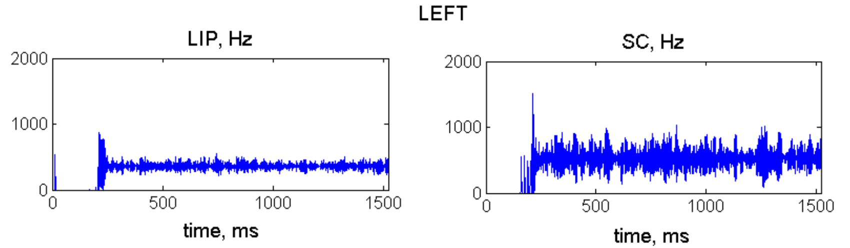
I	Age Group		
	Young	*Middle*	*Old*
k_{left}	-0.02	-0.01	-0.005
k_{right}	0.02	0.01	0.005

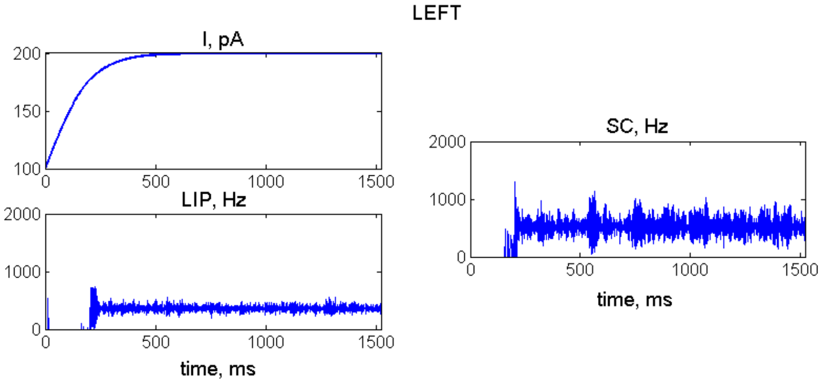
$$I_{left/right} = A_{left/right}/(1 + \exp(k_{left/right}t))$$

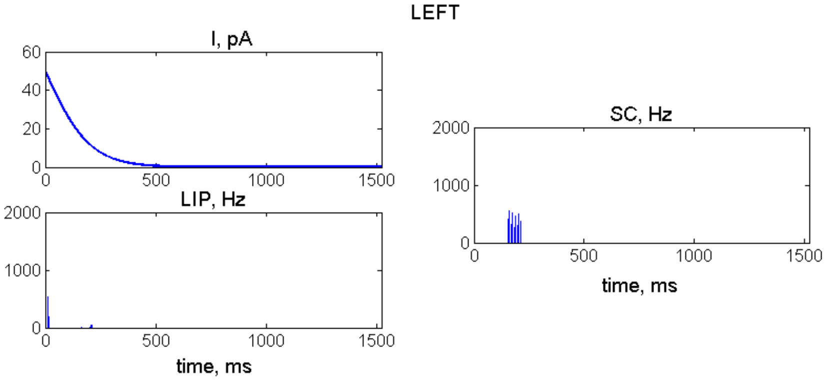
Koprinkova-Hristova, P. et al., AIAI (2020) and IJCNN (2020)

Training results - left

zero reinforcement

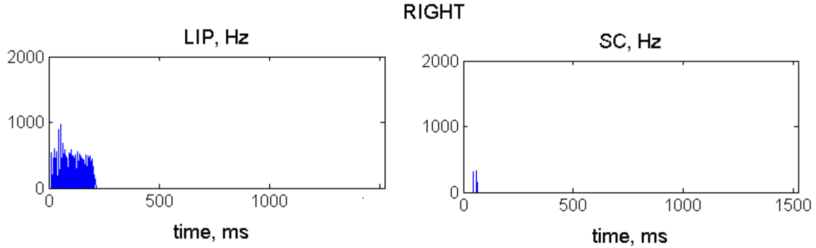


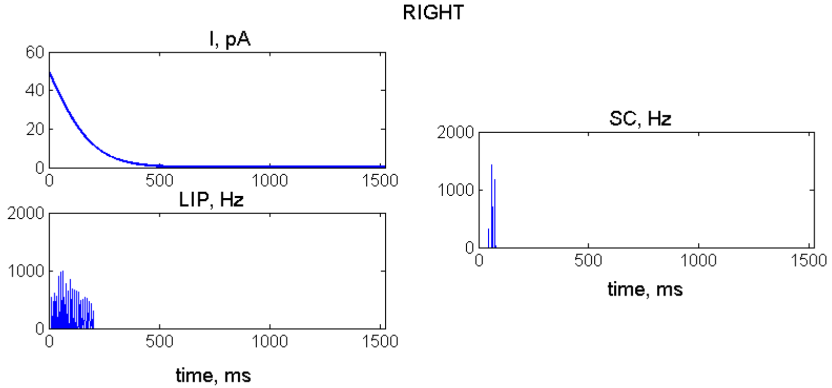


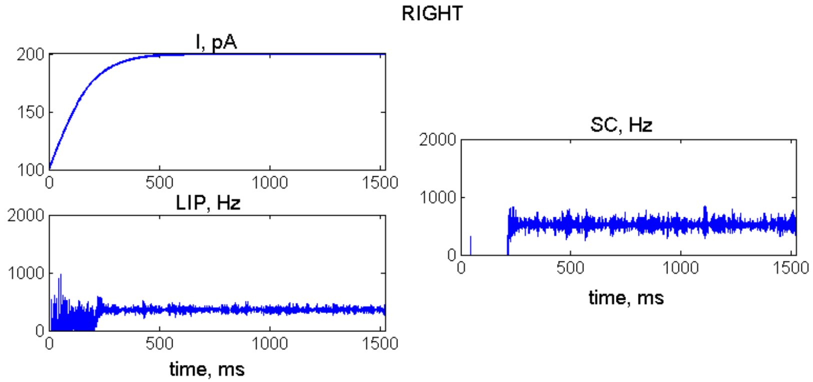


Training results - right

zero reinforcement





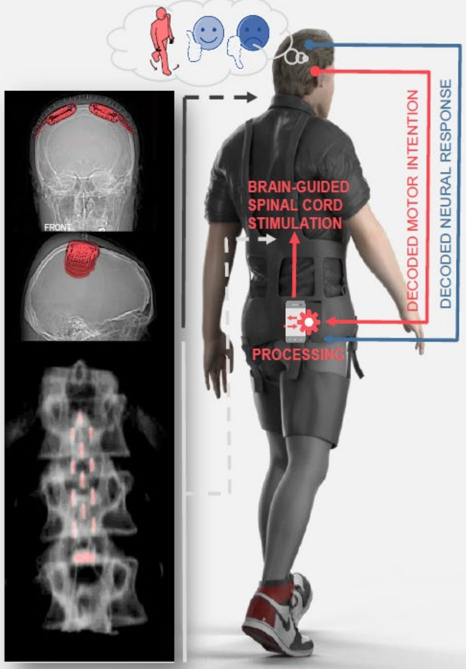


Example 2

SNN for brain signals decoding

NEMO-BMI project

FULLY EMBEDDED AUTO-ADAPTIVE BRAIN MACHINE INTERFACE



IMPLANTABLE MEASURE – STIMULATION TECHNOLOGY

- CHRONIC WIRELESS BRAIN RECORDING WIMAGINE IMPLANT
- SPINAL CORD STIMULATION ONWARD IMPLANT
- 2 CLINICAL TRIALS ONGOING: BRAIN MACHINE INTERFACE PROOF OF CONCEPT

AUTO-ADAPTIVE MOTOR BMI DECODING

- NATURAL CONTROL BASED ON PATIENT'S INTENTION
- MULTIPLE DEGREES OF FREEDOM CONTROL
- DECODING OF NEURAL RESPONSE LINKED TO INTENTION/ACTION COHERENCE
- REAL-TIME AUTO-ADAPTIVE DECODER
- ASSISTANCE FREE
- NEUROMORPHIC DECODING ALGORITHMS

BRAIN-GUIDED SPINAL CORD STIMULATION

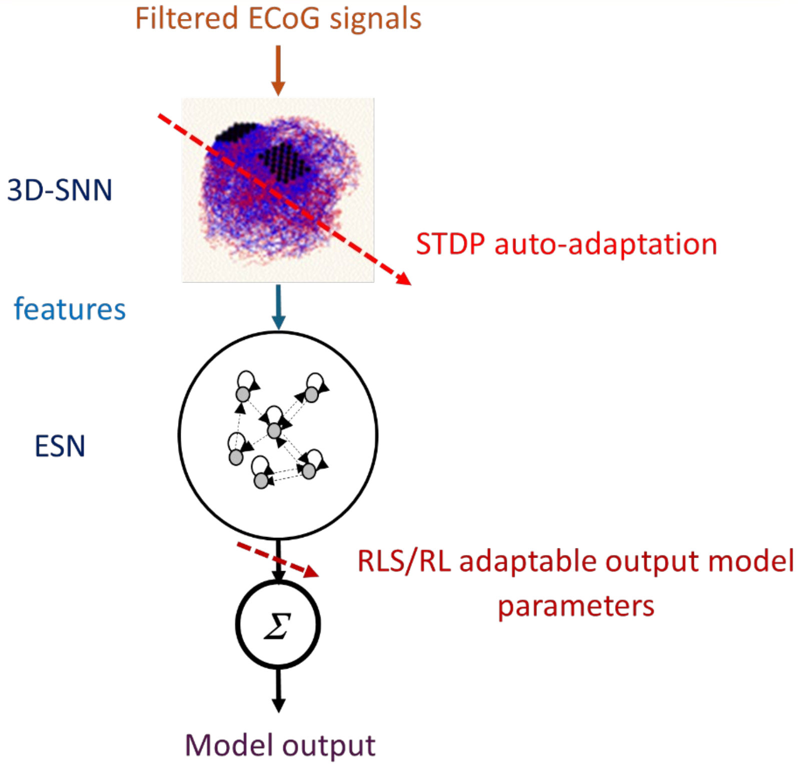
- EPIDURAL ELECTRICAL TARGETED DYNAMIC STIMULATION
- AUTO-ADAPTATIVE STIMULATION PATTERNS

MINIATURIZATION OF BMI TECHNOLOGY

- LOW POWER INTEGRATED CIRCUIT FOR ACCELERATING THE DECODING ALGORITHMS
- HIGH SYSTEM LEVEL INTEGRATION
- PORTABLE BATTERY-POWERED SOLUTION

Experimental set-up

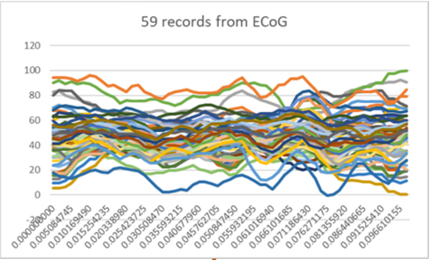
SNN for ECoG data decoding



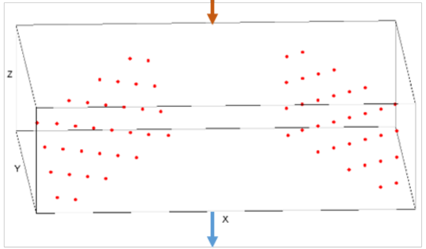
European Innovation Council
Funded by the European Union

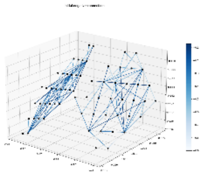
Features extraction process

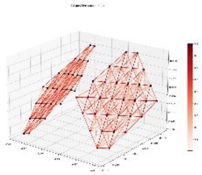
input currents



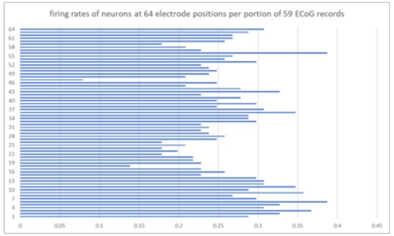
3D-SNN







features



Software and hardware tools

NEST simulator

A tool for biologically plausible AI development

The Neural Simulation Technology Initiative

nest::

ABOUT NEST DOWNLOAD FEATURES DOCUMENTATION PUBLICATIONS COMMUNITY

1. NEST provides over 50 neuron models many of which have been published. Choose from simple integrate-and-fire neurons with current or conductance based synapses, over the Izhikevich or AdEx models, to Hodgkin-Huxley models.
2. NEST provides over 10 synapse models, including short-term plasticity (Tsodyks & Markram) and different variants of spike-timing dependent plasticity (STDP).
3. NEST provides many examples that help you getting started with your own simulation project.
4. NEST offers convenient and efficient commands to define and connect large networks, ranging from algorithmically determined connections to data-driven connectivity.
5. NEST lets you inspect and modify the state of each neuron and each connection at any time during a simulation.
6. NEST is fast and memory efficient. It makes best use of your multi-core computer and compute clusters with minimal user intervention.
7. NEST runs on a wide range of UNIX-like systems, from MacBooks to BlueGene supercomputers.
8. NEST has minimal dependencies. All it really needs is a C++ compiler. Everything else is optional.
9. NEST developers are using agile continuous integration-based workflows in order to maintain high code quality standards for correct and reproducible simulations.
10. NEST has one of the largest and most experienced developer communities of all neural simulators. NEST was first released in 1994 under the name SYNOD and has been extended and improved ever since.
11. NEST is open source software and is licensed under the GNU General Public License v2 or later.

Neuromorphic architectures

Sandamirskaya *et al.*, *Sci. Robot.* **7**, eabl8419 (2022)

Table 1. Overview of some of the neuromorphic chips available today. 1 K = 1056; 1 M = 1 million; Y, yes; N, no; SNN, spiking neural network; HPC, high performance computing.

Company/Lab	Chip type	#Neurons/ synapses	On-chip learning	Power	Software	Applications
ROLLS (*16*)	Mixed-signal	256/64 K	Y	~5 mW	Custom python	Research
DYNAP-SE (*15*)	Mixed-signal	4 K/4 M	N	~5 mW	Custom python	Research
NeuroGrid (BrainDrop)/ Stanford (*29*)	Mixed-signal	1 M/billions	N	~3 W	NEF	Real-time SNN emulation
Innatera	Mixed-signal	256/64 K	N	~1 mW	PyTorch	Smart sensing
BrainScaleS 1/ Universität Heidelberg (*17*)	Mixed-signal	~180,000/40 M (in 352 chips)	N	~300 W	BrainScaleS OS	Accelerated SNN emulation; HPC
BrainScaleS 2/ Universität Heidelberg (*30, 31*)	Mixed-signal	512/~130,000	Y	~1 W	BrainScaleS OS	Edge processing, robotics
TrueNorth/IBM (*9*)	Digital	1 M/256 M (in 4 K cores)	N	~0.3 W	Custom	DNN acceleration
SpiNNaker/University of Manchester (*13*)	Digital	1B/10 kilobytes (in 64 K x 18 ARM cores)	Y	~kW	PyNN, NEST	Real-time simulation of SNN; HPC
Loihi/Intel Labs (*12*)	Digital	~128,000/128 M per chip (scalable)	Y	~1 W	Lava	Research chip
Dynap-CNN/ SynSense	Digital	~327,000/278,000	N	~5 mW	Rockpool, PyTorch	Smart sensing
BrainChip/Akida	Digital	Configurable, 8-Mb SRAM	Y	~30 mW	TensorFlow, CNN → SNN	Smart sensing, one-shot learning
Tianjic/Tsinghua University (*34*)	Digital	40,000/10 M (on 156 cores)	N	~1 W	Custom	ANN/SNN acceleration

Loihi vs supercomputer simulation

Based on: Dey S and Dimitrov A (2022) Mapping and Validating a Point Neuron Model on Intel's Neuromorphic Hardware Loihi. Front. Neurosci. 16:883360. doi: 10.3389/fnins.2022.883360

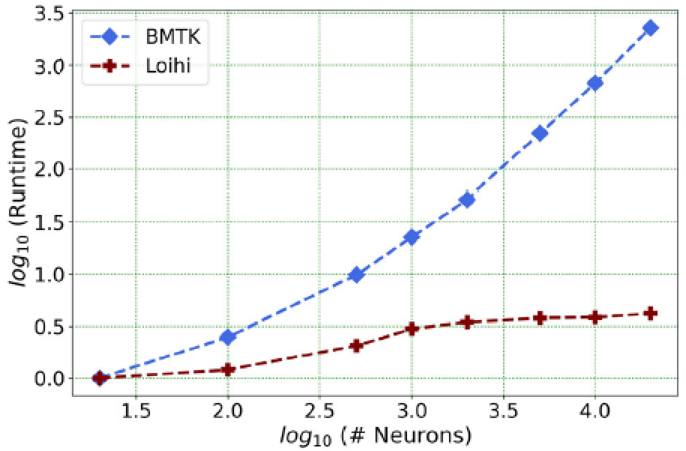


FIGURE 1 | As the network size increases, Loihi outperforms consistently in terms of time. The figure shows runtime comparison of 500 *ms* of dynamics for up to 20,000 neurons for Loihi and BMTK, with the values scaled by the respective smallest runtime. Loihi has a maximum runtime of up to 12 ms, whereas BMTK runtime goes up to 273 s (See **Table 3** for the explicit runtime values and Section 4.4 for further details about the network.).

TABLE 5 | Simulation runtime in Loihi and BMTK.

Network size	Loihi time (ms)	BMTK time (s)
20	2.52	0.12
100	3.03	0.3
500	5.21	1.13
1,000	7.56	2.72
5,000	9.57	26.47
10,000	9.73	80.45

The classical BMTK simulation are instantiated and run on a single node of Kamiak, a high performance computing cluster. A typical Kamiak node contains 2 Intel Xeon E5-2660 v3 CPUs at 2.60 GHz, with 20 cores and 128–256 GB RAM.

Recreating the behavior of a SNN modeled in the NEST simulator using the Lava framework

Why this matters:

- NEST is widely used in computational neuroscience.
- Lava targets neuromorphic hardware like Intel's Loihi.
- Bridging them allows for efficient deployment of neuroscience models on edge devices.

Challenges in Mapping

Input mechanism mismatch:

- NEST allows continuous input current.
- In Lava, we had to simulate this by manually adjusting the input current inside a loop.

Simulation:

- Lava requires manual state resets and run conditions after each manual change of input current

Model parameters:

- Proper scaling is required to map exactly NEST model to LAVA

Core Mapping Strategy

Feature	NEST	Lava
Neurons	iaf_psc_alpha	LIF process
Input Injection	step_current_generator	Manual update of u each step
Connectivity	Connect() with weight matrix	Dense(weights=...)
Time Step Control	Internal in Simulate()	Explicit with RunSteps() loop
Spiking Monitoring	spike_recorder	Monitor().probe(lif1.s_out, ...)

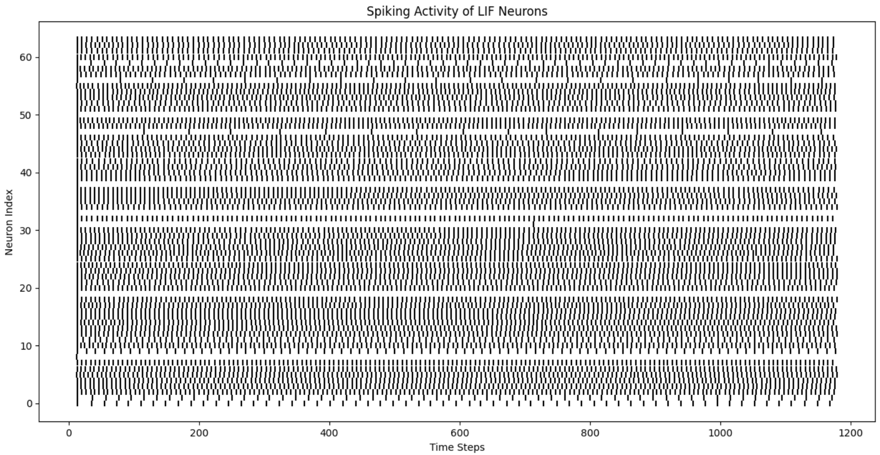
Key components in our models:

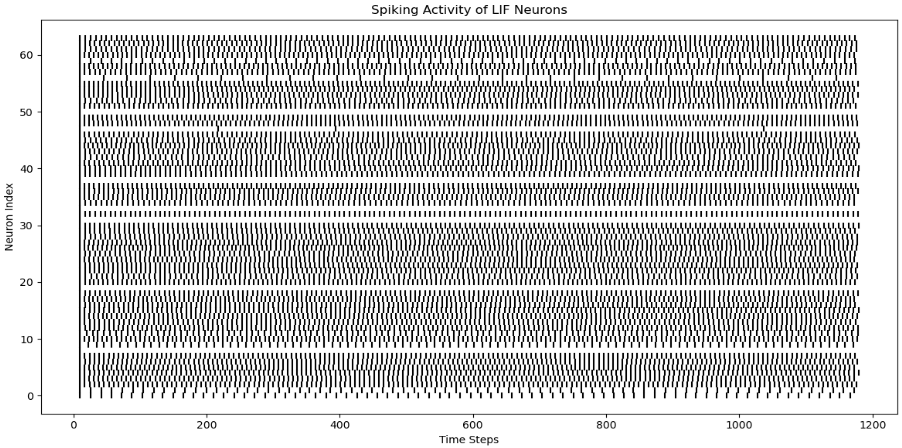
64 neurons with injected currents from real ECoG data.

Our custom connectivity.

Simulation time for 20 portions of 59 ECoG records, i.e. totally 1180 time steps.

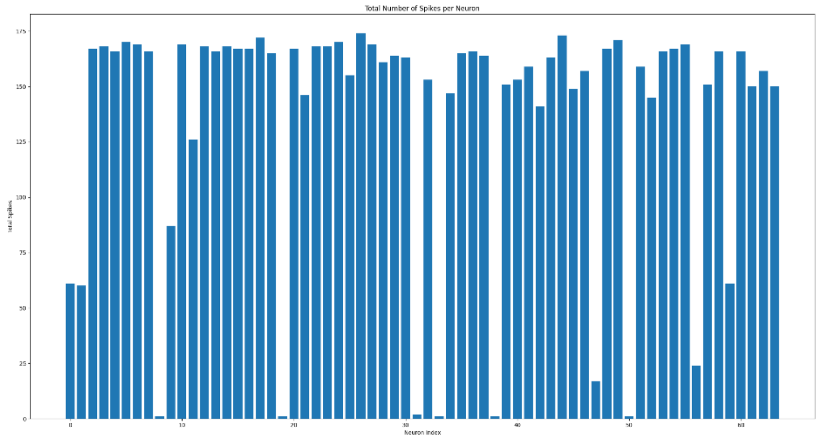
Results Comparison

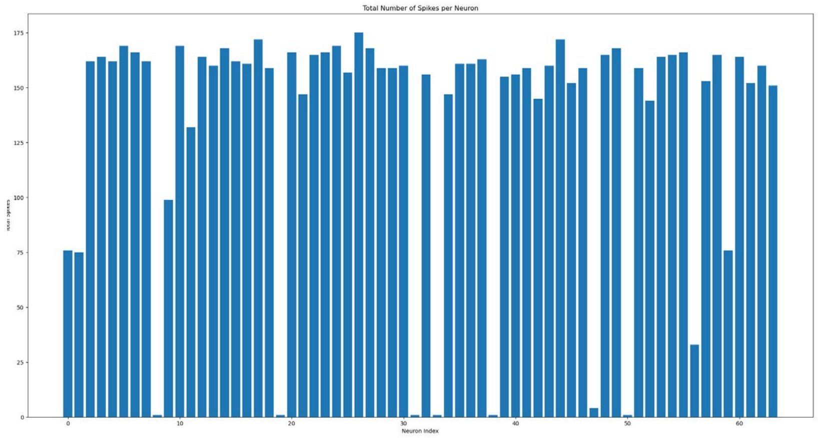




Spiking activity of whole SNN structure (left-LAVA, right-NEST)

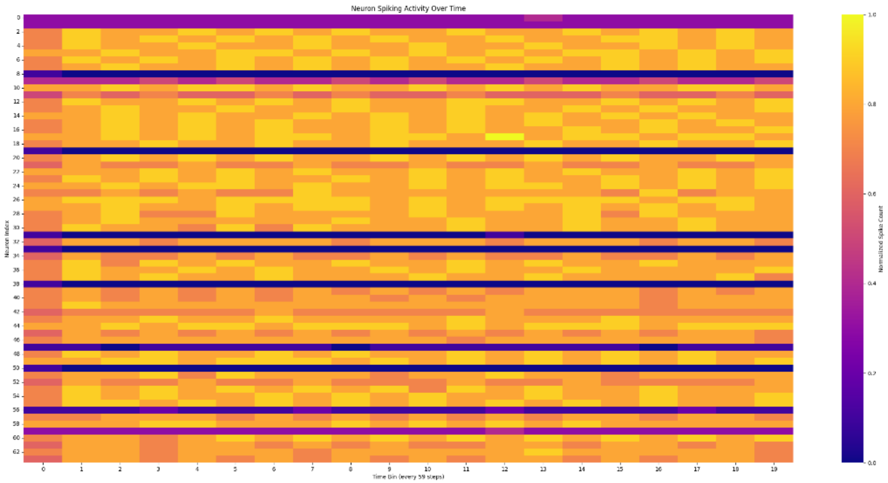
Results Comparison

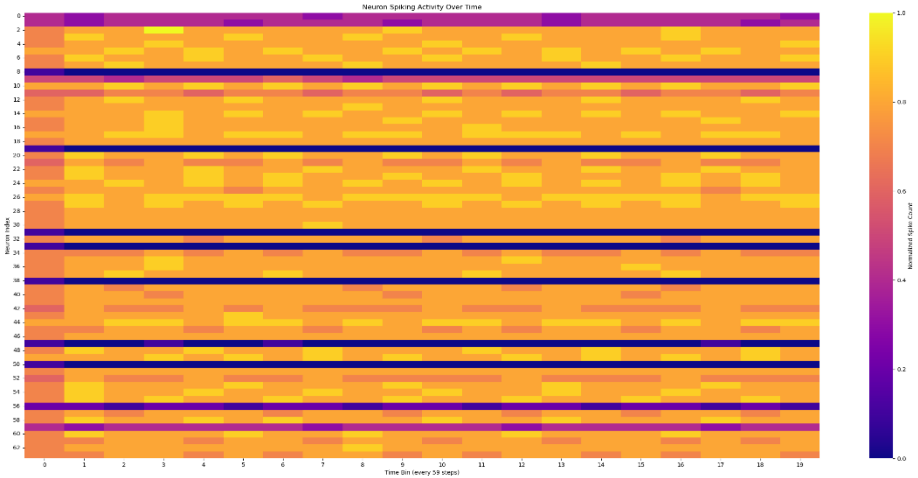




Total number of spikes per neuron for whole simulation time (left-LAVA, right-NEST)

Results Comparison





Normalized number of spikes per neuron for 20 bins of 59 steps each (left-LAVA, right-NEST)

Thank you for your attention!

Questions?