

**KNOWLEDGE ENGINEERING & DISCOVERY
RESEARCH INSTITUTE**
AN INSTITUTE OF AUT UNIVERSITY



**The International
Neural Network
Society (INNS)**

ICANN 2013
Sofia, 10-13.09.2013
Keynote Speech

Contemporary Developments in Neural Networks

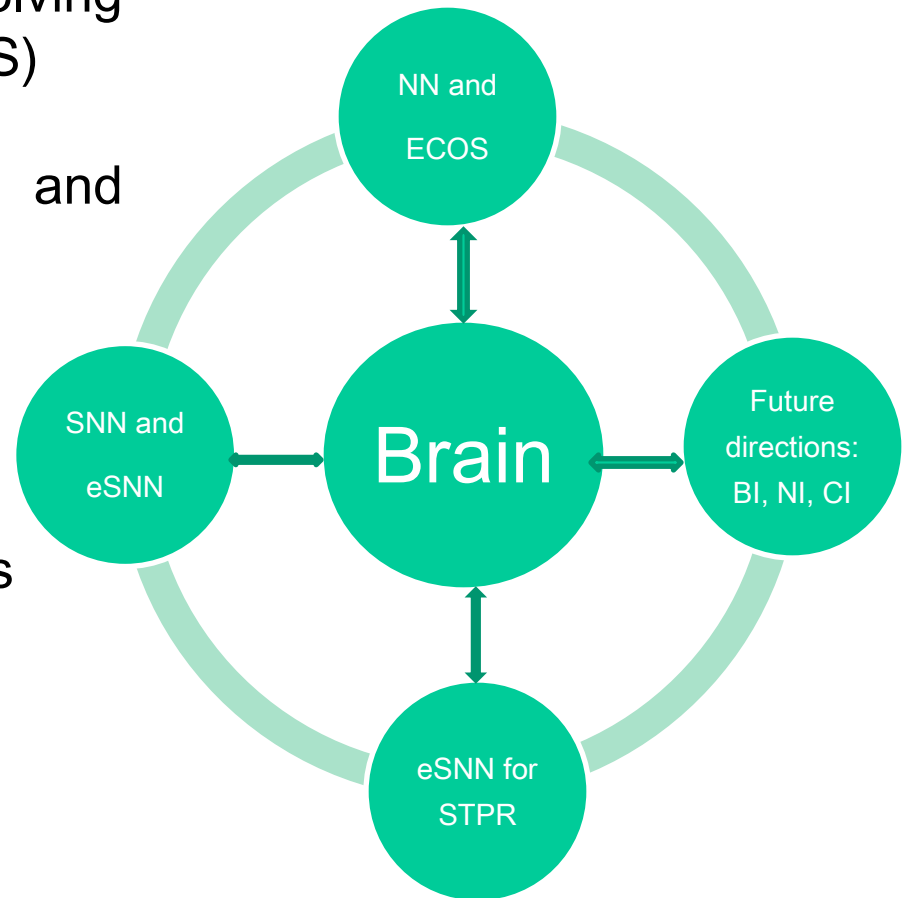
Spiking Neural Networks for Adaptive Spatio-/Spectro Temporal Pattern Recognition

Nikola Kasabov, FIEEE, FRSNZ,
Royal Academy of Engineering Distinguished Visiting Fellow

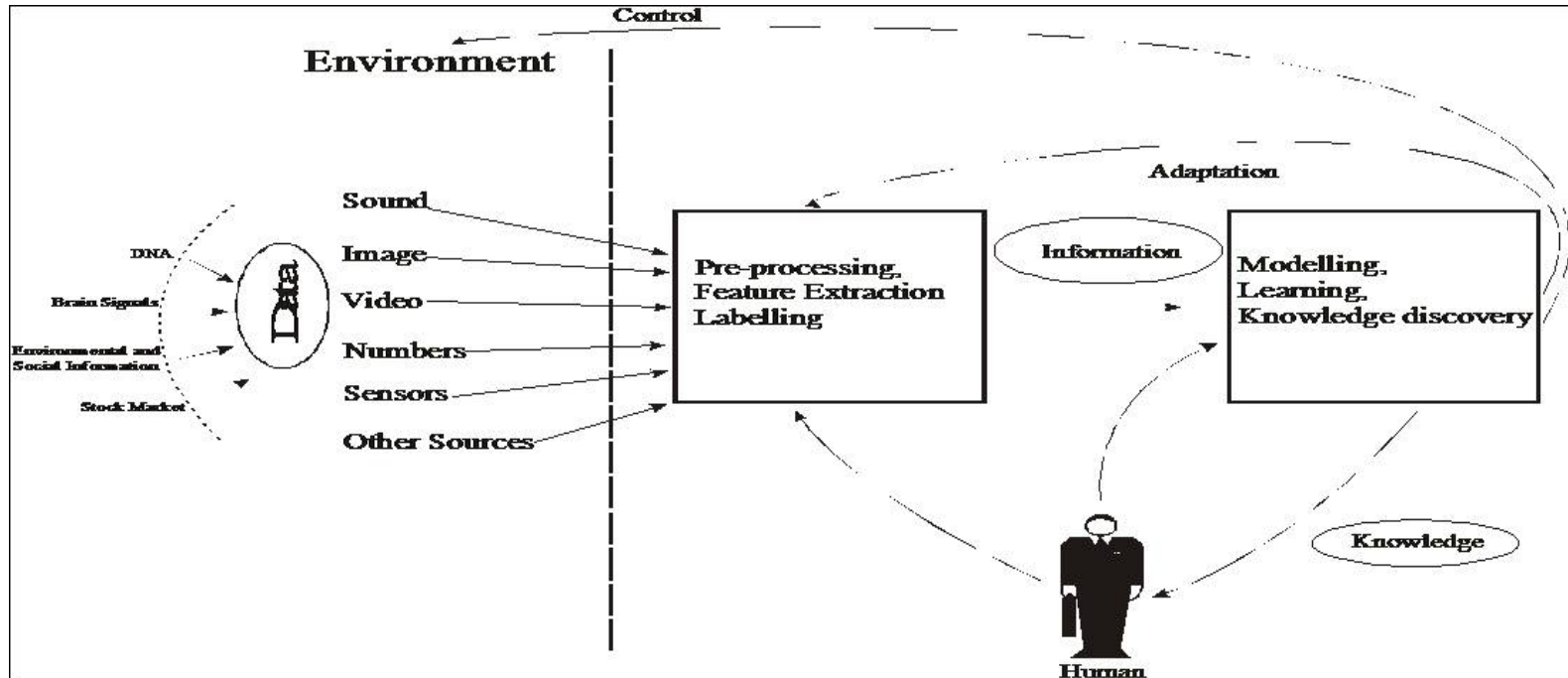
Professor and Director,
Knowledge Engineering and Discovery Research Institute (KEDRI),
Auckland University of Technology, New Zealand

Content

1. Neural Networks and Evolving Connectionist Systems (ECOS)
3. Spiking Neural Networks (SNN) and eSNN
4. eSNN for Spatio/Spectro-Temporal Pattern Recognition
5. eSNN for Early Prediction of Events
6. Future Directions



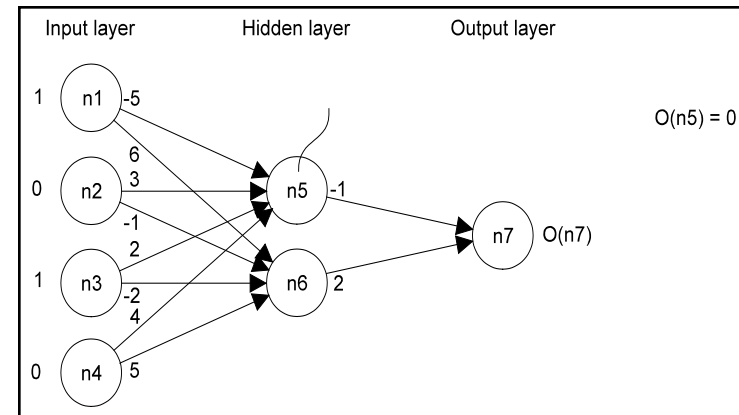
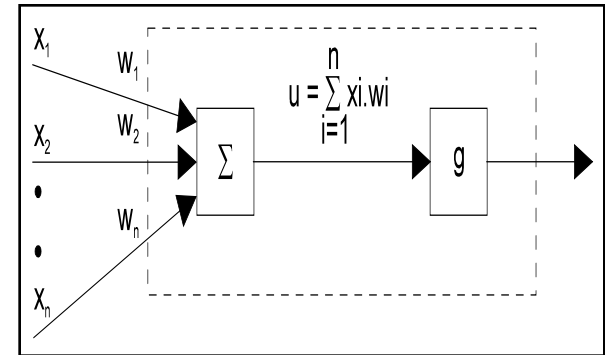
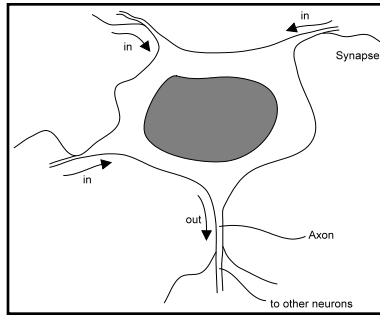
1. Neural Networks and Evolving Connectionist Systems (ECOS)



- Modelling complex processes is a difficult task: adaptation is needed based on new data and new information
- Knowledge discovery – always evolving, improving, changing
- A wide range of real-world on-line applications
- Neural Networks (NN) is a suitable paradigm for the above tasks

Neural Networks

- NN are computational models that mimic the nervous system in its main function of adaptive learning.
- ANN can *learn* from data and make *generalisations*
- ANN are *universal computational models*
- Software and hardware realisation of ANN – Neurocomputing
- Frank Rosenblatt (1928-1971), Perceptron, 1962
- Multilayer perceptrons (second generation of NN)



ANN Development

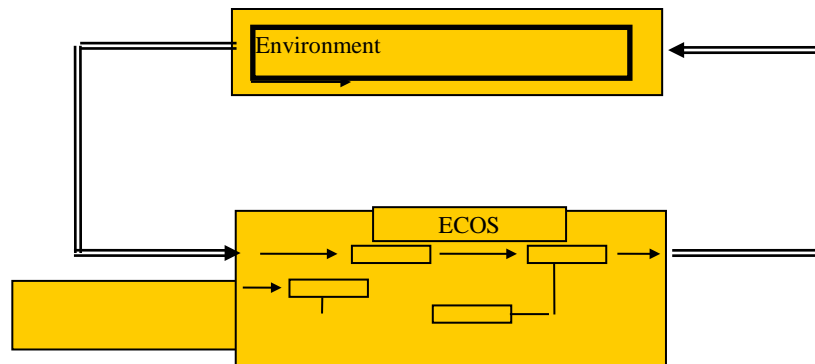
- 1943, McCulloch and Pitts - a model of a neuron,
- 1960, Widrow and Hoff- Adelaine,
- 1962, Rosenblatt - Perceptron,
- 1971- 1986, Amari, Rumelhart and others, Multilayer perceptron,
- 1980, International Neural Network Society, INNS, www.inns.org, Grossberg
- 1990- Hybrid neuro-fuzzy and neuro-symbolic systems (Kosko, Yamakawa, Kasabov, Sun and others)
- 1992, European Neural Network Society, ENNS, J.Taylor (1936-2012)

The Bulgarian connection:

- 1990, Int. School of AI (ISAI): Connectionism & AI, Varna (Braspenning, Taylor, Gallinary, Kasabov)
- First publications in Bulgaria:
 - Kasabov, N. Neural networks and genetic algorithms. *Avtomatika i Informatika*, 8/9:51-60 (1990) (in Bulgarian)
 - Kasabov N., Hybrid connectionist rule based systems, in: *Artificial Intelligence IV Methodology, Systems, Applications*, P. Jorrand and V. Sgurev (eds) Amsterdam, North-Holland (1990) 227- 235

Evolving Connectionist Systems (ECOS)

- ECOS are modular connectionist-based systems that **evolve their structure** and functionality in a continuous, self-organised, in on-line, **adaptive**, interactive way from incoming information facilitating **knowledge discovery** (Kasabov, 1998, 2002, 2007).



- Early ECOS models: RAN (J.Platt, 1991) – evolving RBF NN; Incremental FuzzyARTMAP (Carpenter , Grossberg); Growing gas; EFuNN (Kasabov, 1998, 2001); ESOM (Deng and Kasabov, 2002); DENFIS (Kasabov, Song, 2002); EFuRS, eTS (Angelov, 2002;Filev, 2002).
- *M.Watts, Ten years of Kasabov's evolving connectionist systems, IEEE Tr SMC- part B, 2008.*
- New developments: Ensembles of EFuNNs (T. Ljudemir, 2008-); Application oriented ECOS (B.Gabric, R.Duro, McGinitty et al.); Incremental feature selection (Ozawa, Pang, Kasabov, Polikar, Minhu Lee); evolving spiking neural networks (eSNN); computational neuro-genetic systems; quantum inspired eSNN.

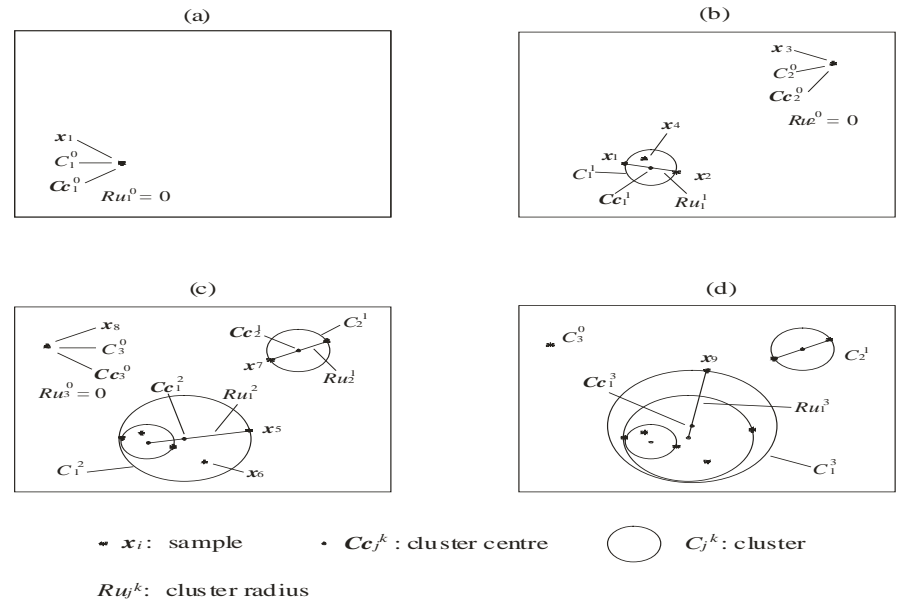
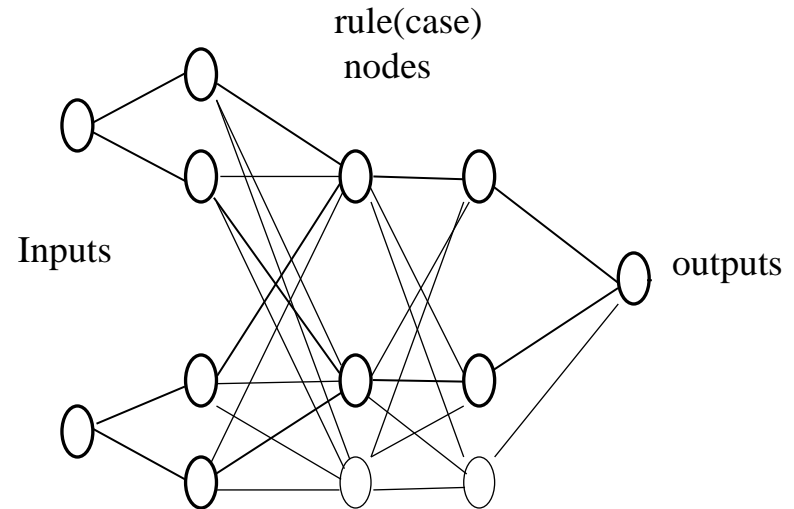
Evolving Fuzzy Neural Network (EFuNN)

- Incremental, supervised clustering
- Input and/or output variables can be non-fuzzy (crisp) or fuzzy
- Hidden nodes evolve to capture clusters (prototypes) of input vectors
- **Input weights change based on *Euclidean distance* between input vectors and prototype nodes (evolving clustering):**

$$\Delta w = \text{irate} * E(x, R_n)$$

- Output weights evolve to capture local output function and change based on output error.

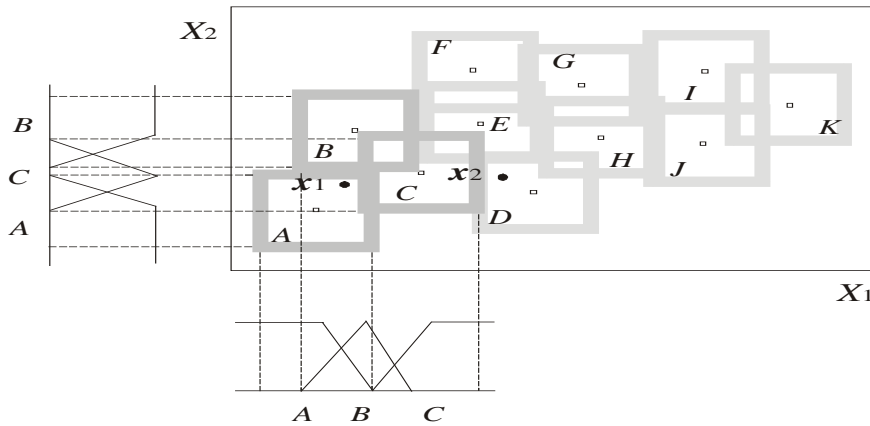
- EFuNN, N. Kasabov, IEEE Tr SMC, 2001
- DENFIS, N.Kasabov, Q.Song, IEEE Tr FS, 2002
- ECOS Toolbox available in MATLAB
- NeuCom Software available: www.kedri.info



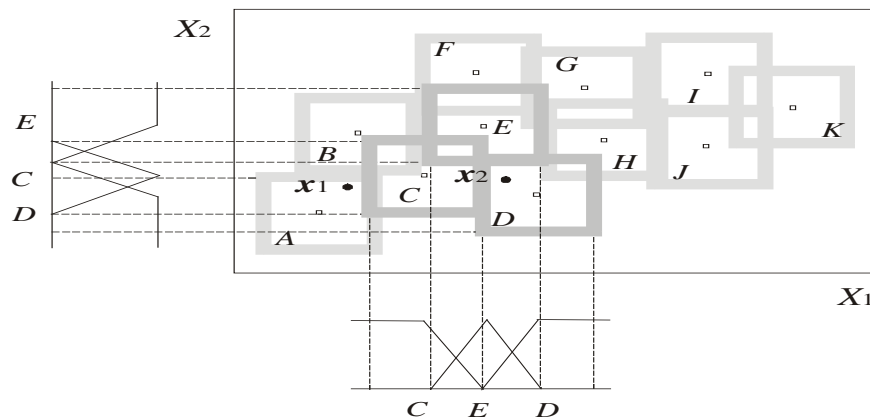
DENFIS: Evolving Neuro-Fuzzy Inference System

(DENFIS, Kasabov and Song, 2002, IEEE Tr Fuzzy Systems, 600 citations)

(a) Fuzzy rule group 1 for a DENFIS



(b) Fuzzy rule group 2 for a DENFIS



DENFIS algorithm:

(1) Learning:

- Unsupervised, incremental clustering.
- For each cluster there is a Takagi-Sugeno fuzzy rule created: IF x is in cluster C_j THEN $y_j = f_j(x)$, where: $y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_q$
- Incremental learning of the function coefficients and weights of the functions through least square error

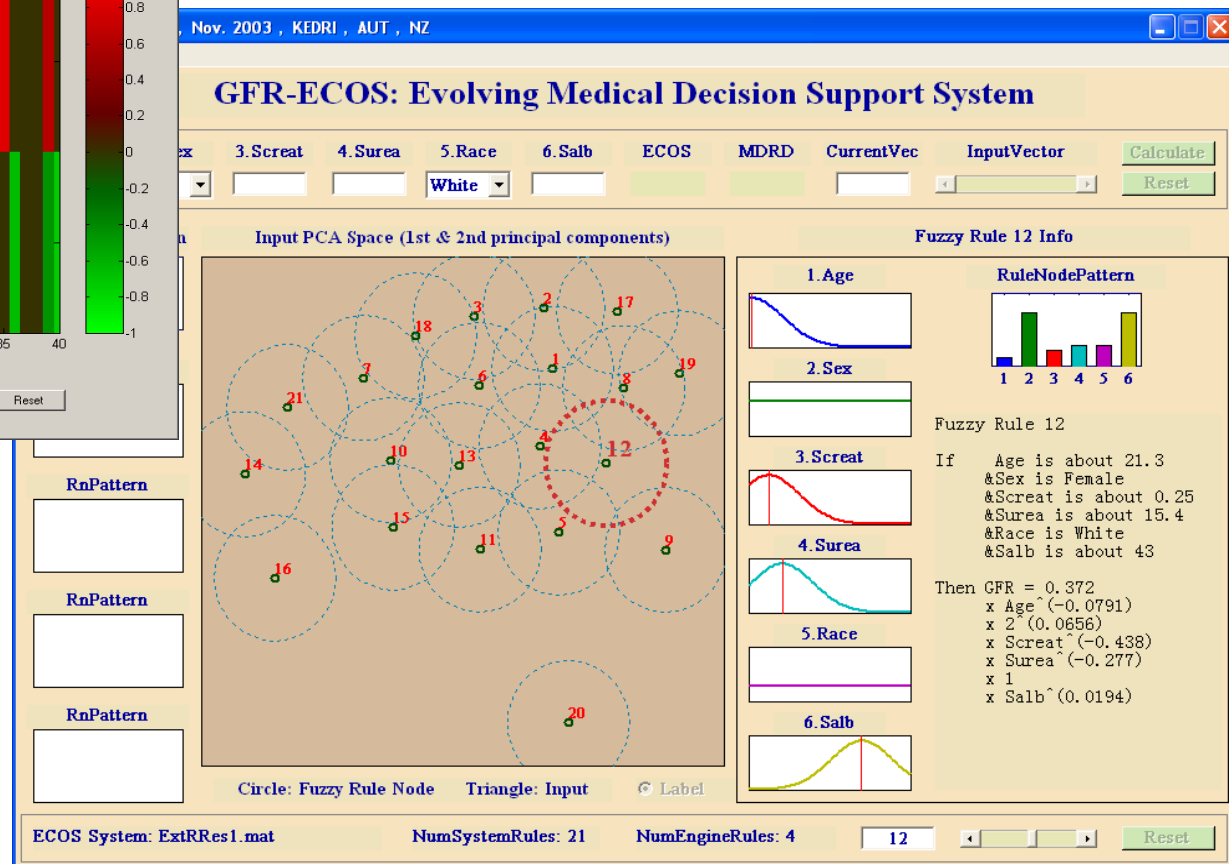
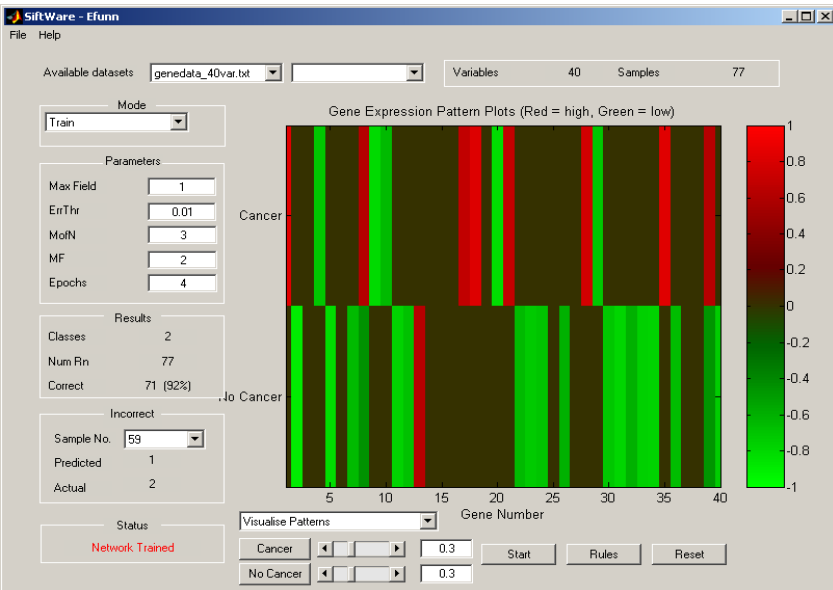
(2) Fuzzy inference over fuzzy rules:

- For a new input vector $x = [x_1, x_2, \dots, x_q]$ DENFIS chooses m fuzzy rules from the whole fuzzy rule set for forming a current inference system.
- The inference result is:

$$y = \frac{\sum_{i=1,m} [\omega_i f_i(x_1, x_2, \dots, x_q)]}{\sum_{i=1,m} \omega_i}$$

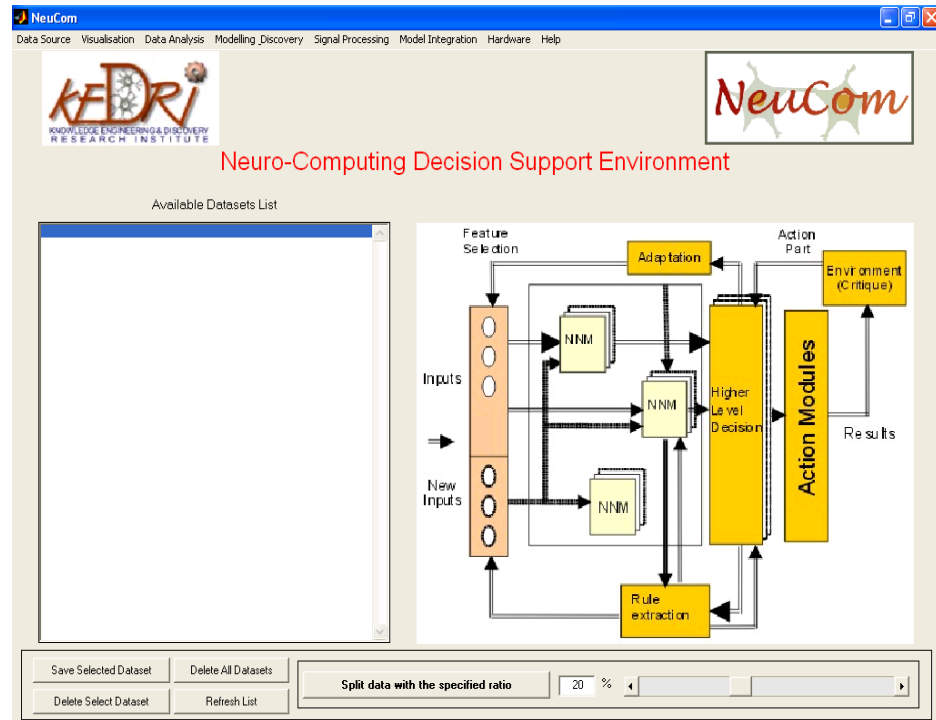
Applications of ECOS

- Bioinformatics -
- Neuroinformatics
- Decision support systems



NeuCom: A Software Environment for NeuroComputing, Data Mining and Intelligent System Design (www.theneucom.com)

- A generic environment, that incorporates 60 traditional and new techniques for intelligent data analysis and the creation of intelligent systems, including:
 - Statistical methods
 - Neural networks
- Methods for feature selection
- Methods for classification
- Methods for prediction
- Methods for knowledge extraction
- Fast data analysis and visualisation
- Fast model prototyping
- A free copy available for education and research from: www.theneucom.com
- DENFIS for prediction
- ECF for classification



2. Spiking Neural Networks (SNN) and eSNN

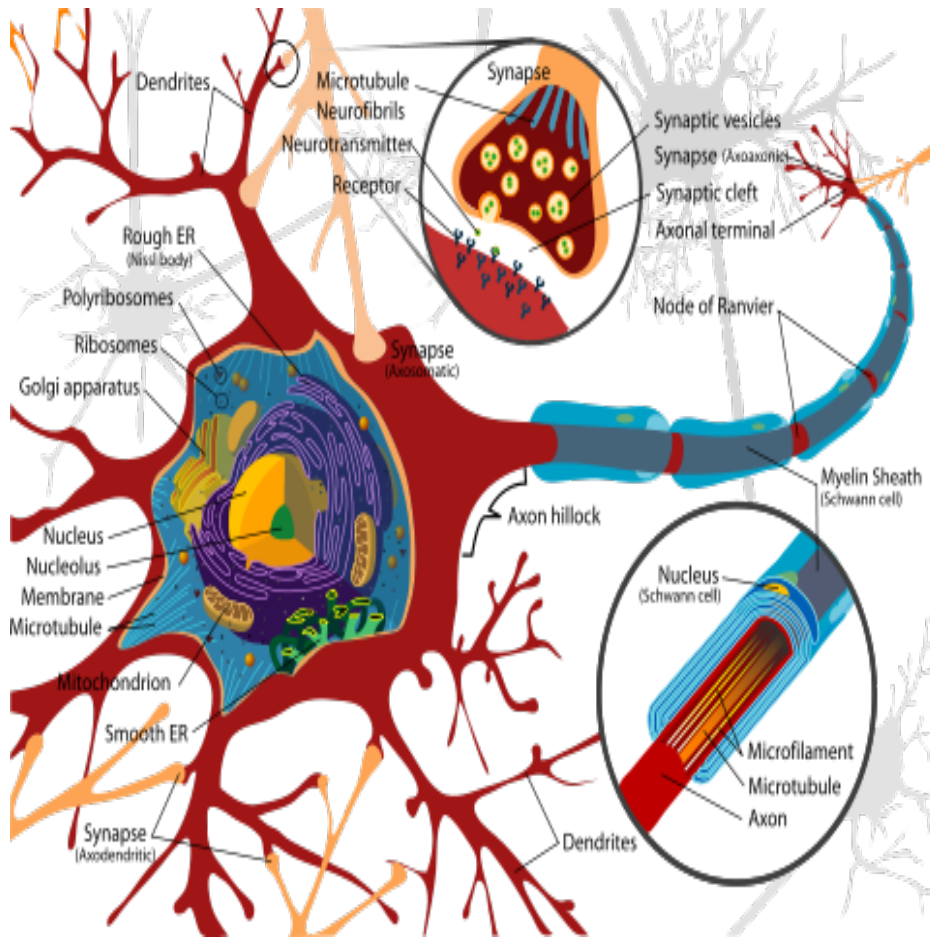
Brain-like NN – third generation of NN

A single neuron is very rich of information processes: time; frequency; phase; field potentials; molecular (genetic) information; space.

Three, mutually interacting memory processes:

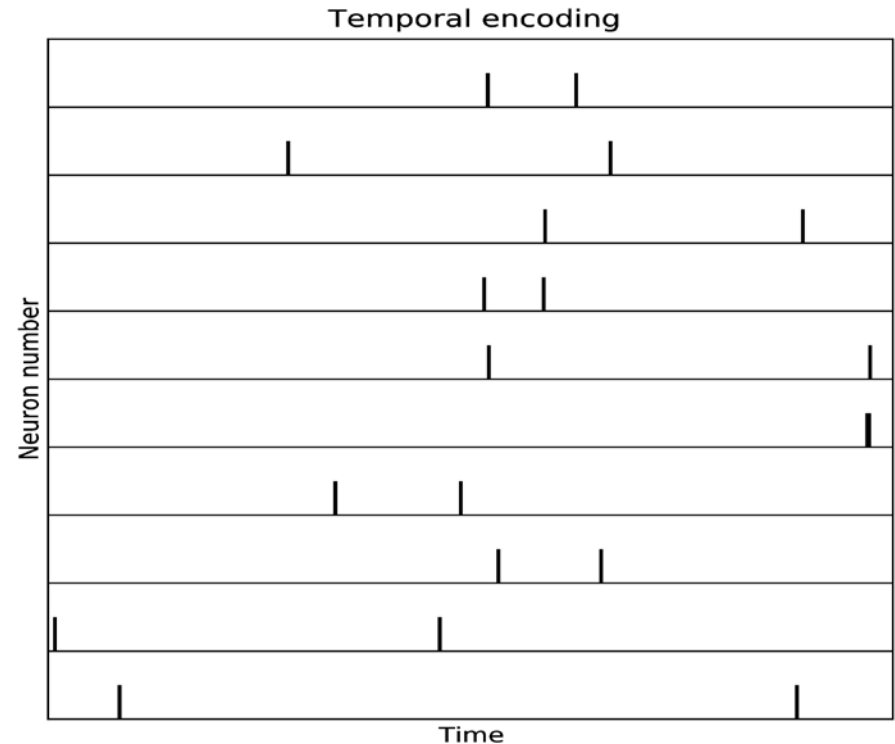
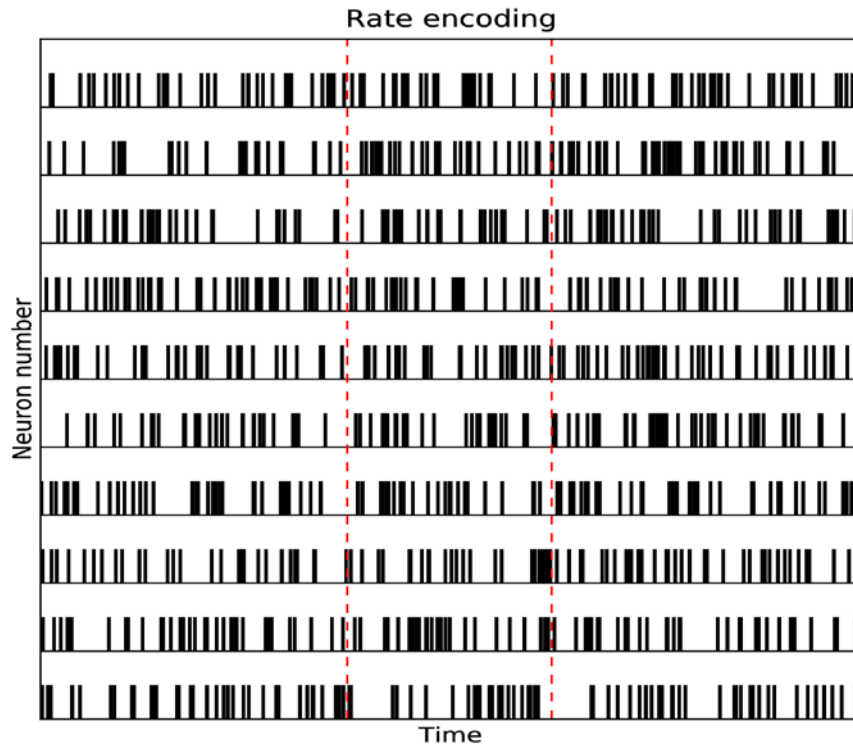
- short term (membrane potential);
- long term (synaptic weights)
- genetic (gene and protein information)

SNN can accommodate both spatial and temporal information as location of neurons/synapses and their spiking activity over time.



Representing information as spikes: Rate vs time-based

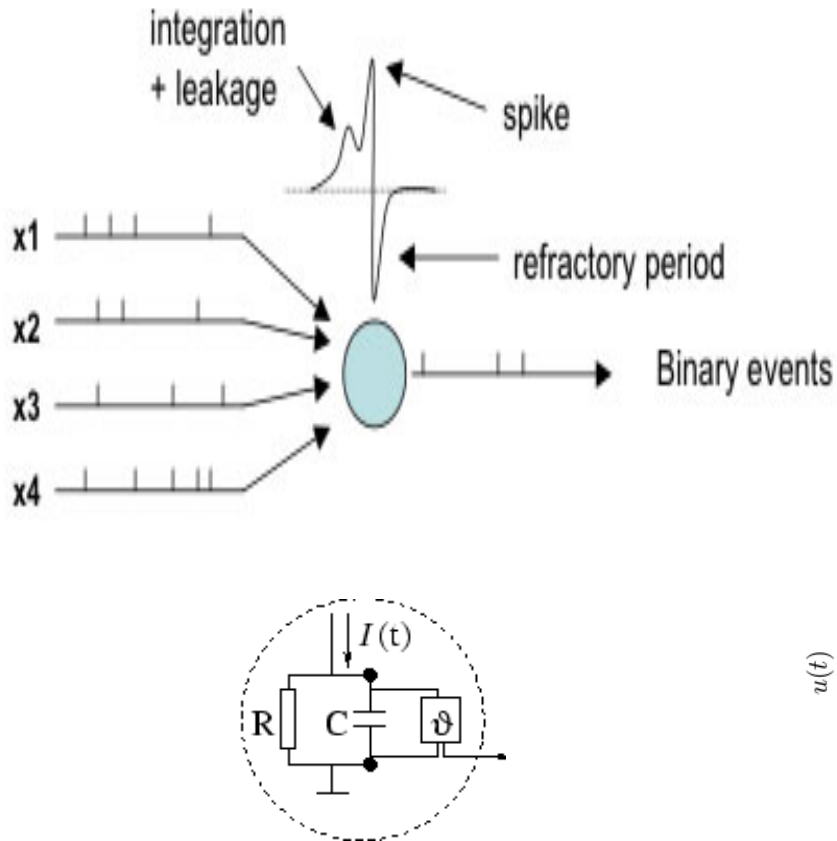
- ❖ Rate-based coding: A spiking characteristic within a time interval, e.g. frequency.
- ❖ Time-based (temporal) coding: Information is encoded in the time of spikes. Every spike matters! For example: class A is a spike at time 10 ms, class B is a spike at time 20 ms.



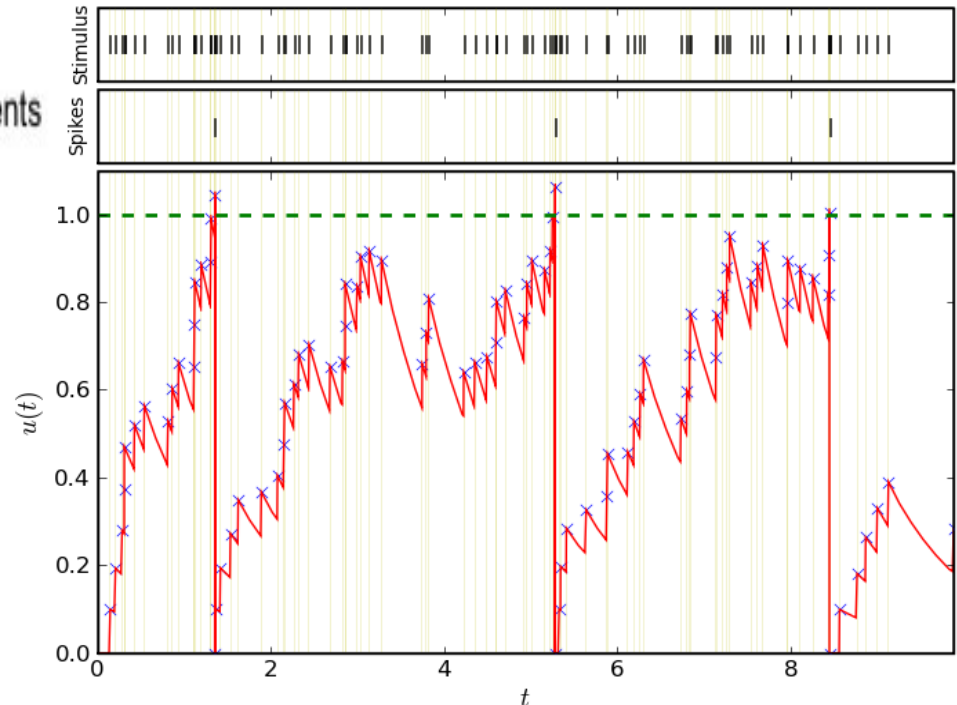
Models of spiking neurons:

(Hodgkin-Huxley 1952; Abbott, 2000; Maas, Izhikevich; other)

Most popular is the Leaky Integrate and Fire Model (LIF) .



$$\tau_m \frac{du}{dt} = -u(t) + RI(t)$$

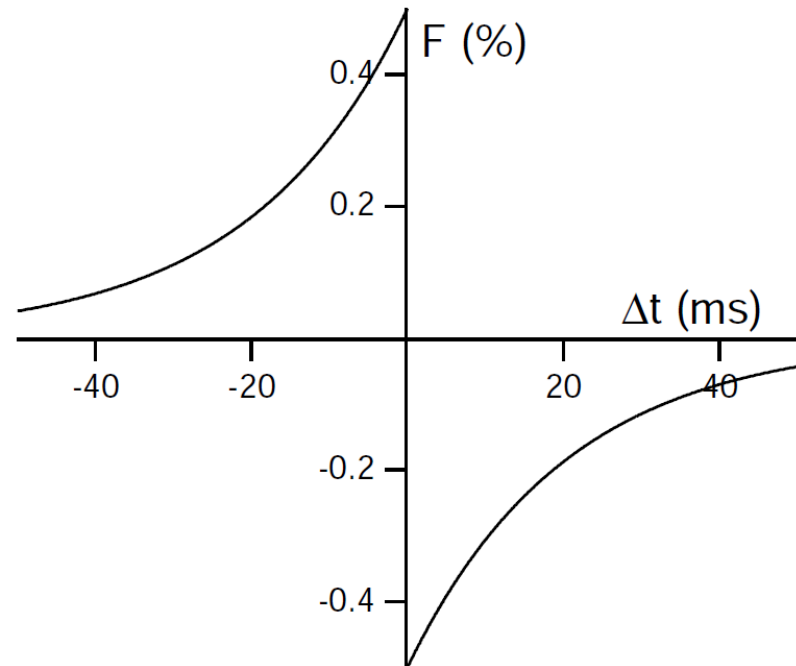


Methods for learning in SNN: Spike-Time Dependent Plasticity (STDP) (Abbott and Nelson, 2000).

- Hebbian form of plasticity in the form of long-term potentiation (LTP) and depression (LTD)
- Effect of synapses are strengthened or weakened based on the **timing** of pre-synaptic spikes and post-synaptic action potential.
- Through STDP connected neurons learn consecutive **temporal** associations from data.

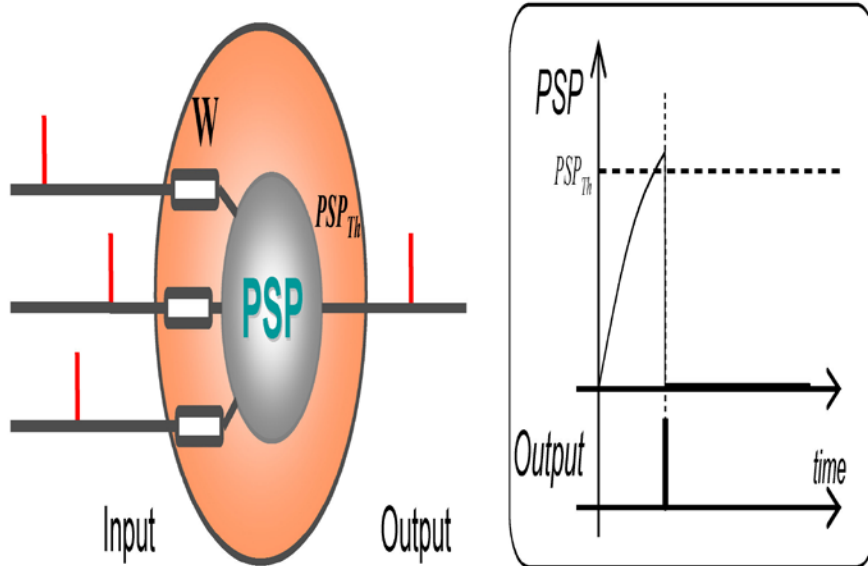
Pre-synaptic activity that precedes post-synaptic firing can induce **LTP**, reversing this temporal order causes **LTD**

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



The rank order (RO) learning rule

(Thorpe et al, 1998)



$$\Delta w_{ji} = m^{\text{order}(j)}$$

$$u_i(t) = \begin{cases} 0 & \text{if fired} \\ \sum_{j|f(j)<t} w_{ji} m_i^{\text{order}(j)} & \text{else} \end{cases}$$

$$\text{PSP max} = \text{SUM} (\text{mod}^{\text{order}(j,i(t))} w_{j,i}(t)), \text{ for } j=1,2,\dots, m; t=1,2,\dots,T$$

$$\text{PSP}_{\text{Th}} = C \cdot \text{PSPmax}$$

Evolving SNN – eSNN

- eSNN: Creating and merging neurons based on localised information (Kasabov, 2007; Wysoski, Benuskova and Kasabov, 2006-2009)
- Uses the first spike principle (Thorpe et al.) for fast on-line training
- For each input vector
 - a) Create (evolve) a new output spiking neuron and its connections
 - b) Propagate the input vector into the network and train the newly created neuron

$$u_i(t) = \begin{cases} 0 & \text{if fired} \\ \sum_{j|f(j)<t} w_{ji} m_i^{\text{order}(j)} & \text{else} \end{cases} \quad \Delta w_{ji} = m^{\text{order}(j)}$$

Weights change based on the spike time arrival

- c) Calculate the similarity between weight vectors of newly created neuron and existing neurons: IF similarity > Threshold THEN Merge newly created neuron with the most similar neuron

$$W \leftarrow \frac{W_{new} + NW}{1 + N}$$

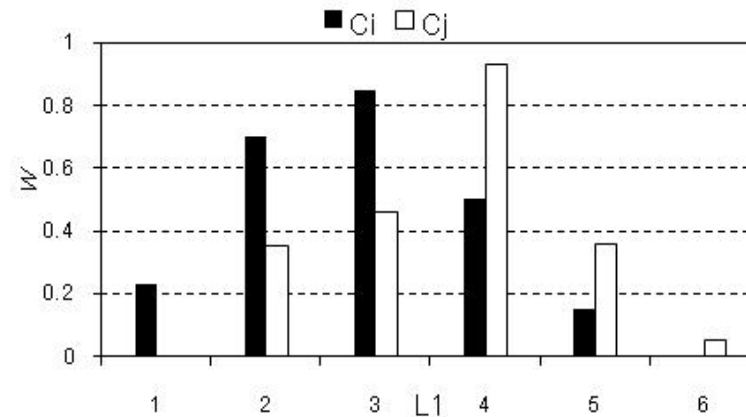
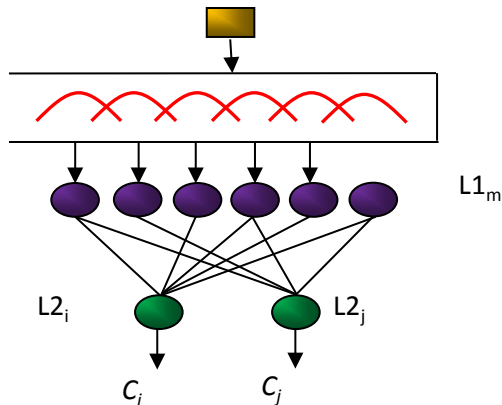
where N is the number of samples previously used to update the respective neuron.

- d) Update the corresponding threshold ϑ :
$$\vartheta \leftarrow \frac{\vartheta_{new} + N\vartheta}{1 + N}$$

- Schliebs, S. and N.Kasabov, Evolving spiking neural networks: A Survey, *Evolving Systems*, Springer, 2013.

Methods for fuzzy rule extraction from eSNN

(S.Soltic, N.Kasabov, Int. J. Neural Systems, World Sc. Publ., 2010)



IF v is SMALL THEN C_i

IF v is LARGE THEN C_j

Progress in neuromorphic computation

Hodgin- Huxley model (1952)

Carver Mead (1989): A hardware model of an IF neuron:
The Axon-Hillock circuit;

INI Zurich SNN chips (Giacomo Indivero, 2008 and 2012)

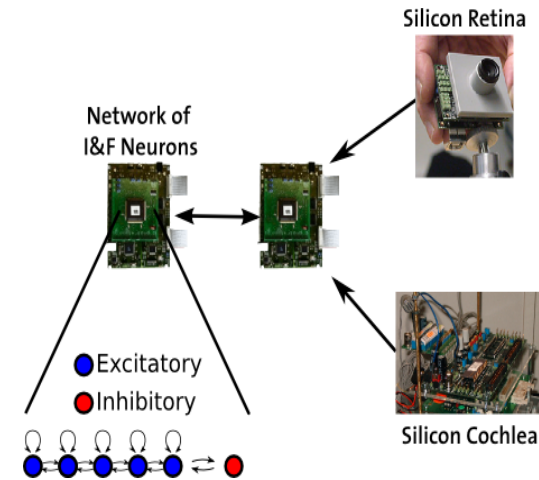
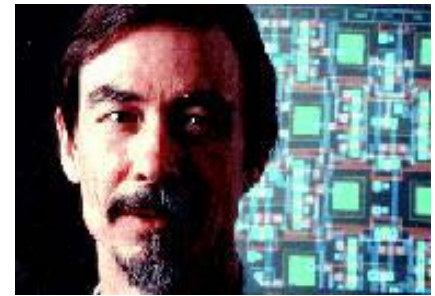
FPGA SNN realisations (McGinnity, Ulster, 2010);

The IBM chip (D.Modha, 2012): 256 LIF neurons and 64k synapses in a chip.

U. Manchester SpiNNaker (2^{16} computer chips, 2011; 1 mln neurons 2013)

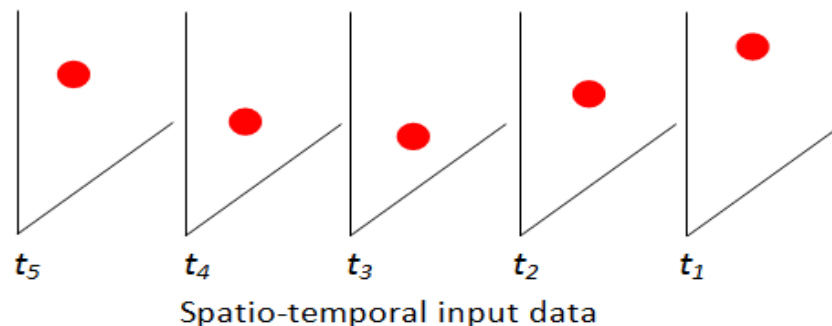
Stanford U., NeuroGrid (Kwabena Boahen et al), 1mln neurons on a board, 63 bln connections ; hybrid - analogue /digital)

The challenge: Technology is available, but how do we use it for engineering applications and specifically – for STPR?

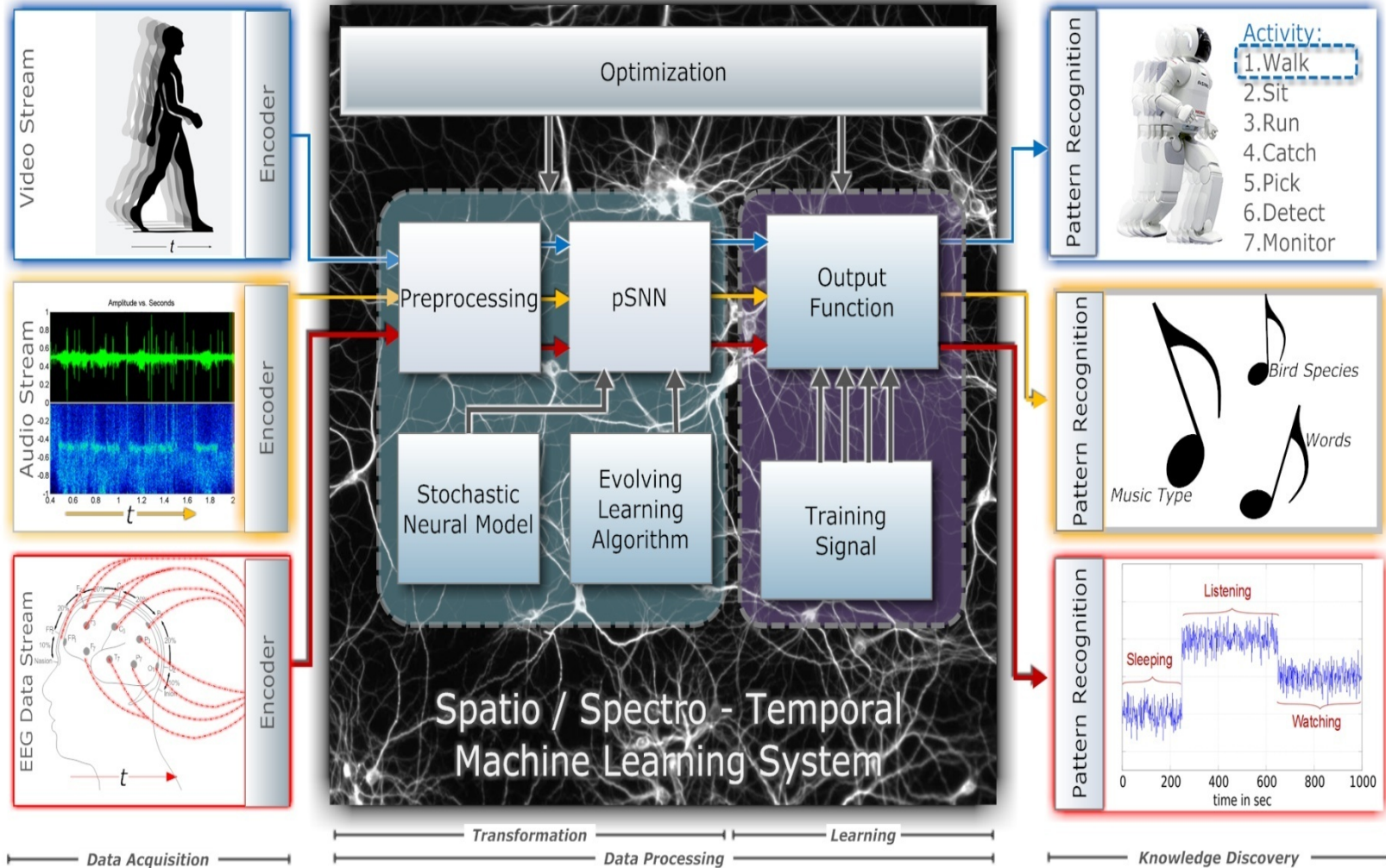


4. eSNN for Spatio/Spectro-Temporal Pattern Recognition

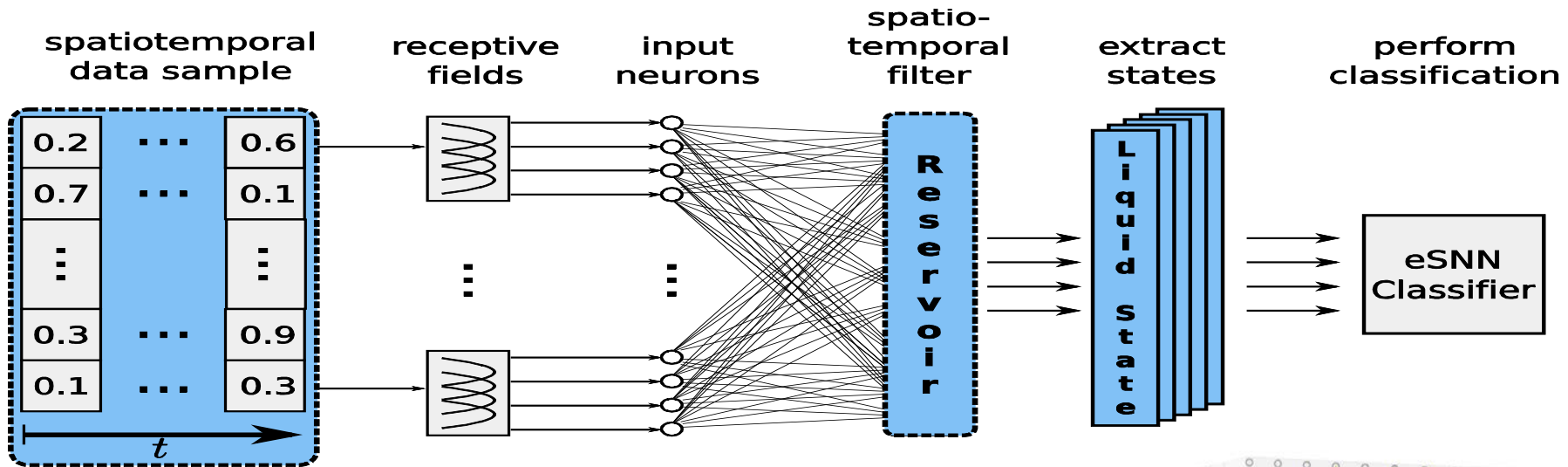
- Most real world data is spatio- or spectro- temporal.
- In STPR problems spatial and temporal components of the information are interrelated.
- Examples of spatio-temporal data and related problems are:
 - a) Object movement recognition from video data
 - b) Audio/video data modelling
 - c) Brain signals (EEG, MEG, fMRI)
 - d) Brain- computer interfaces
 - e) Motor control for prosthetics
 - f) Ecological and environmental data, e.g. earthquake prediction
 - g) Robot control
 - h) Cyber-security data
- Goal: Developing new methods based on ECOS and eSNN for STPR



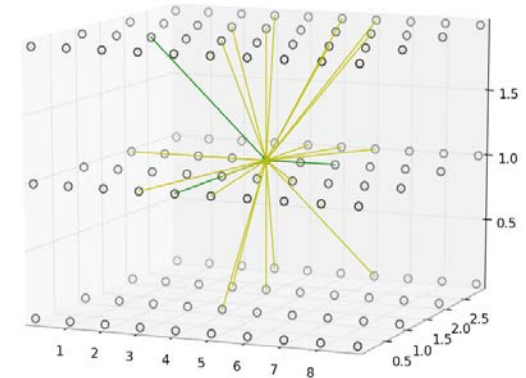
The EvoSpike Project: EU FP7 Marie Curie (<http://ncs.ethz.ch/projects/evospike>)



Reservoir-based eSNN for STPR



- *Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states, Neur. Comp. 14(11),2002;*
- Input (feature) neurons connected to part of the LSM
- Output neurons connected to part of the LSM
- LSM recurrent connections, e.g. small world connections
- Excitatory 80%, Inhibitory 20%
- **Learning in LSM:** STDP; spike time delay???
- **Polychronization** (Izhikevich): ‘opening the box’?

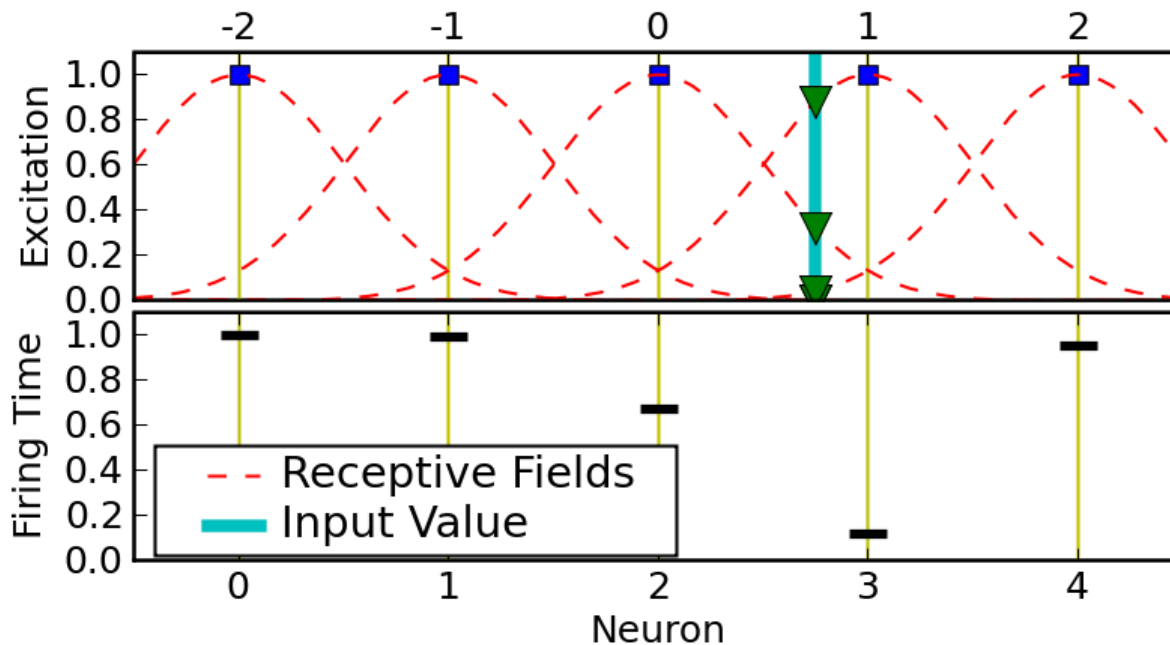


$$p_{a,b} = C \times e^{-D^2_{a,b} / \lambda^2}$$

Encoding input data into spikes

Rank Order Population Encoding

- Distributes a single real input value to multiple neurons and may cause the excitation and firing of several responding neurons
- Implementation based on Gaussian receptive fields introduced by Bothe *et al.* 2002

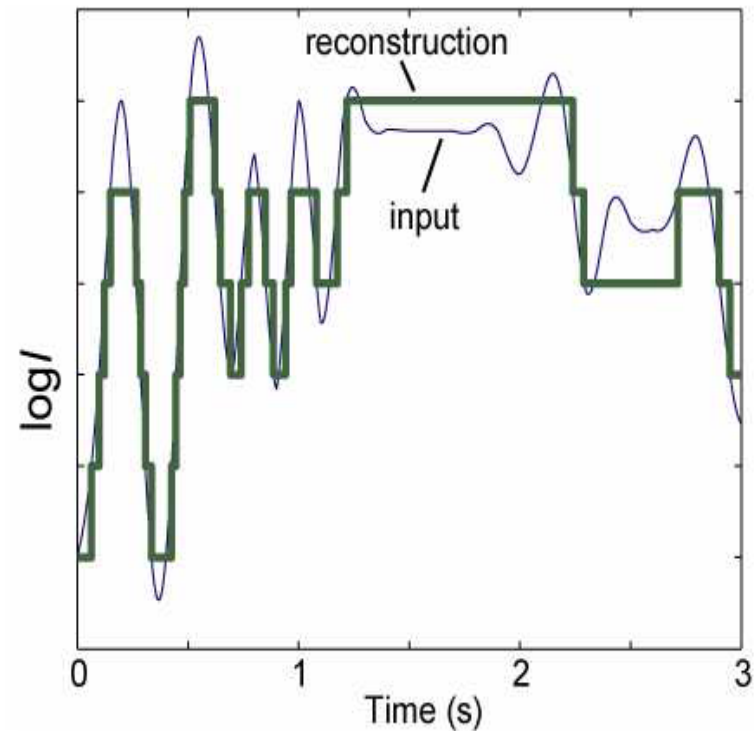
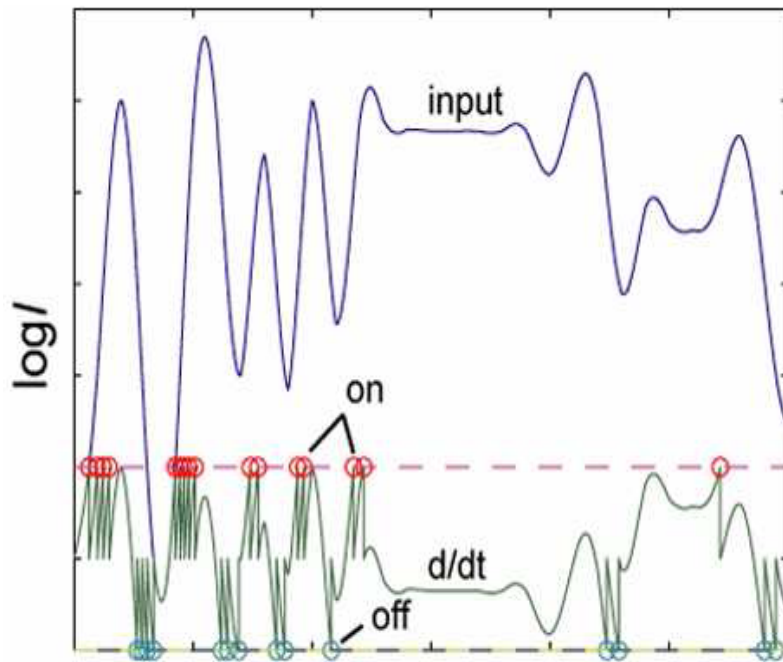


Address Event Representation (AER) Encoding

A spike is generated only if a change in the input data occurs

Silicon Retina (Tobi Delbruck, INI, ETH/UZH, Zurich), DVS128

Silicon Cochlea (Shih-Chii Liu, INI, ETH/UZH, Zurich)



eSNN classifiers for STPR

Dynamic Evolving SNN (deSNN)

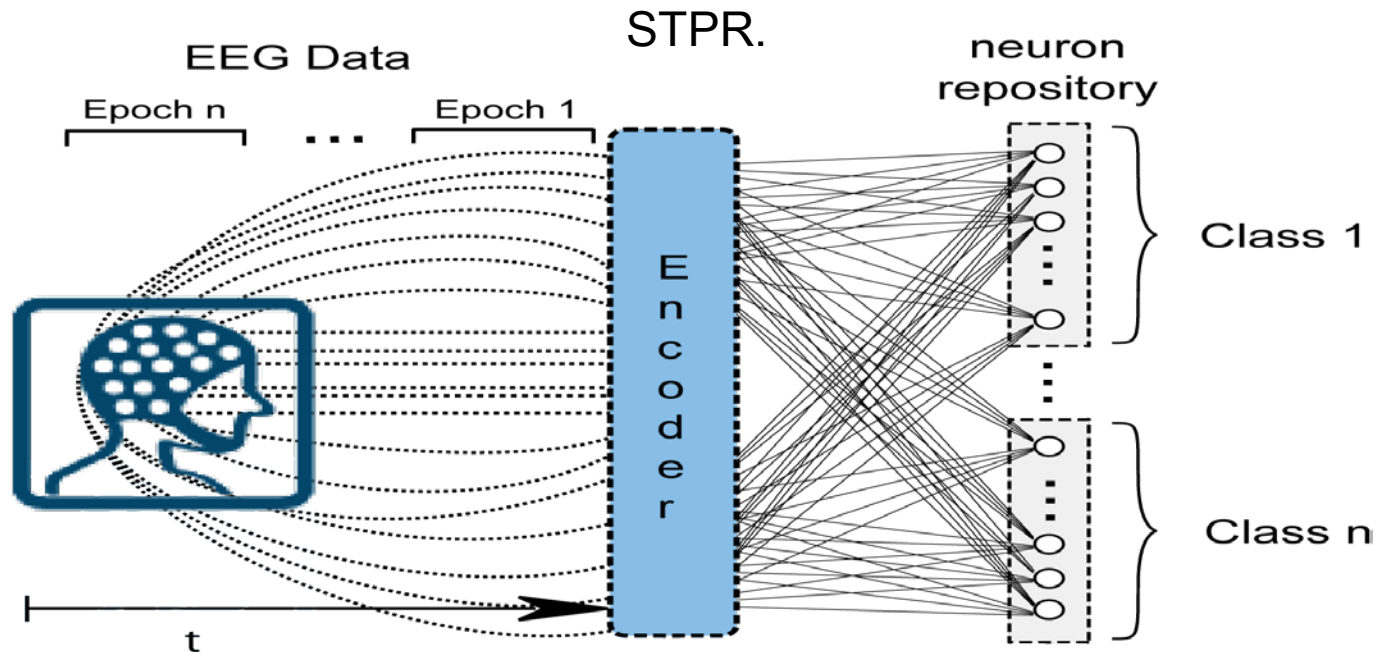
(Kasabov, Dhoble, Nuntalid, Indivery, Neural Networks, 2013)

- Combine: (a) RO learning for weight initialisation based on the first spikes:

$$\Delta w_{ji} = m^{\text{order}(j)}$$

(b) SDSP for learning further input spikes at a synapse.

- A new output neuron is added to a respective output repository for every new input pattern learned. Neurons may merge.
- The figure below shows the deSNN architecture on a case study for EEG

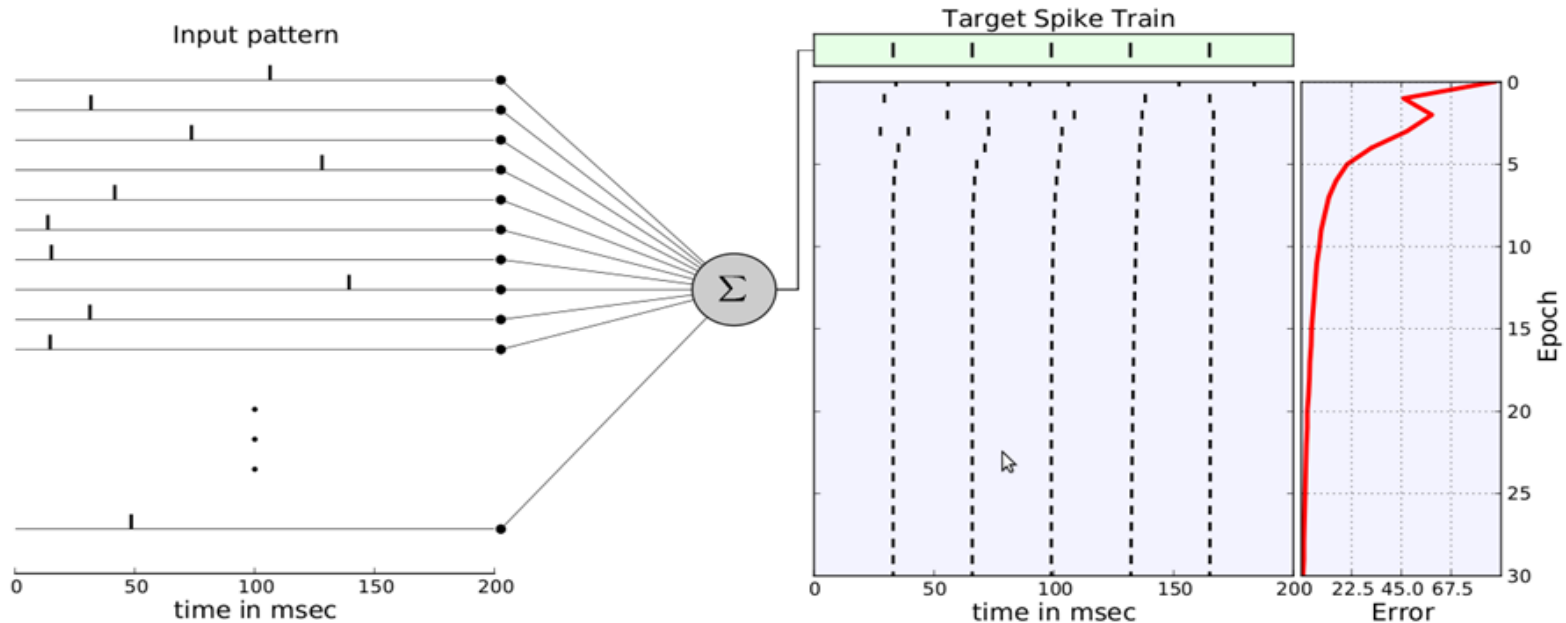


The deSNN training algorithm

- 1: Set deSNN parameters (including: Mod, C, Sim, and the SDSP (STDP) parameters)
- 2: **FOR** every input spatio-temporal spiking pattern P_i **DO**
 - 2a. Create a new output neuron i for this pattern and calculate the initial values of connection weights $w_i(0)$ using the RO learning formula.
 - 2b. Adjust the connection weights w_i for consecutive spikes on the corresponding synapses using the SDSP (STDP) learning rule.
 - 2c. Calculate $PSP_{i\max}$.
 - 2d. Calculate the spiking threshold of the i th neuron.
 - 2e. (Optional) **IF** (the new neuron weight vector w_i is similar in its initial $w_i(0)$ and final $w_i(T)$ values after training to the weight vector of an already trained output neuron using Euclidean distance and a similarity threshold Sim, then merge the two neurons (as a partial case only initial or final values of the connection weights can be considered or a weighted sum of them))
 - ELSE**
 - Add the new neuron to the output neurons repository.
 - END IF**
- END FOR** (Repeat for all input spatio-temporal patterns for learning)

SPAN: Spike Pattern Association Neuron and the Delta Rule

(A.Mohhemed et al, EANN 2011, ICONIP2011, IJNS, 2012; Neurocomputing, 2012))



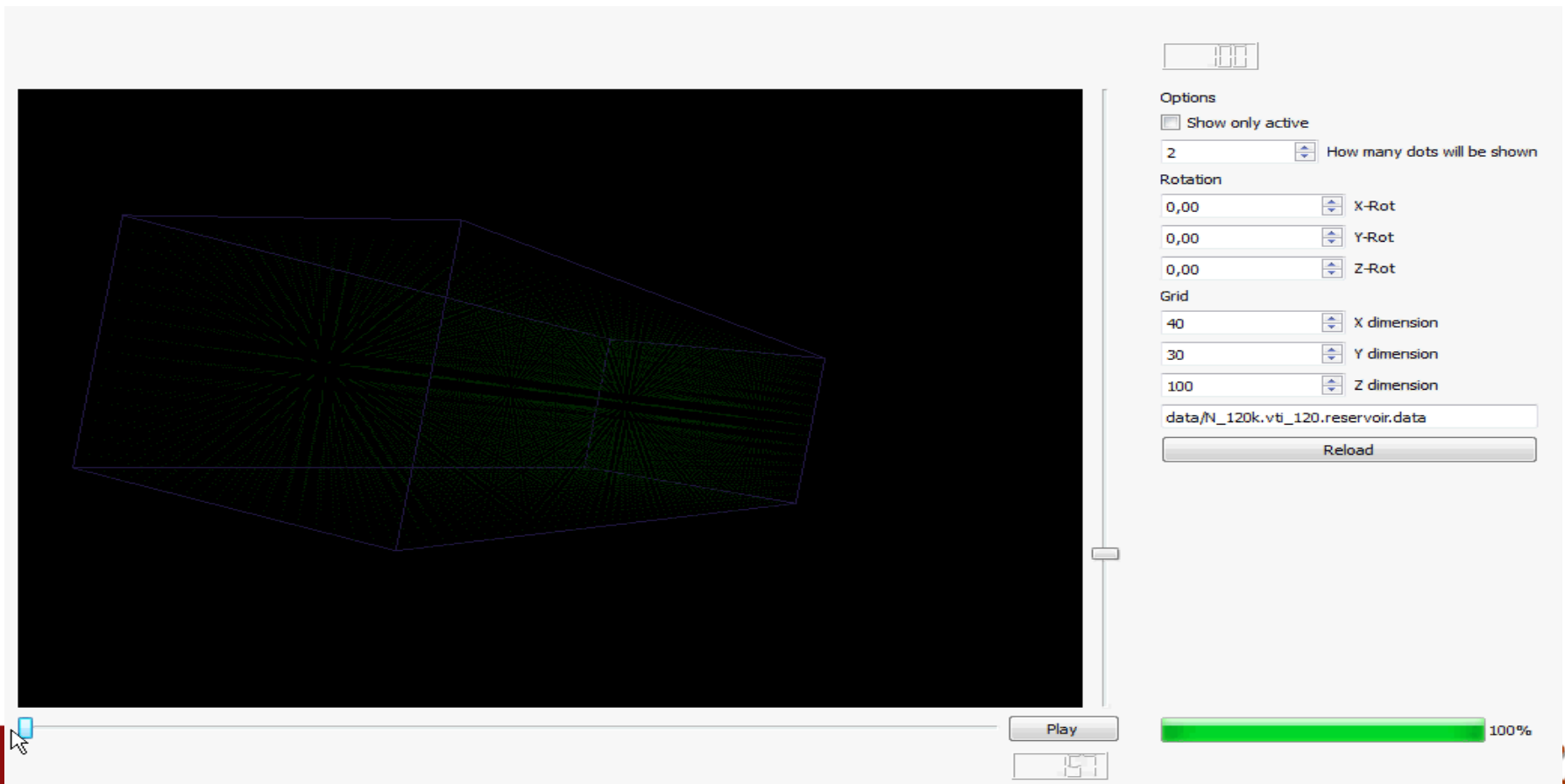
A single output neuron is trained to respond with a temporally precise output spike train to a specific spatio-temporal input.

Spike pattern association neuronal models: SpikeProp; ReSuMe; Tempotron; Chronotron.

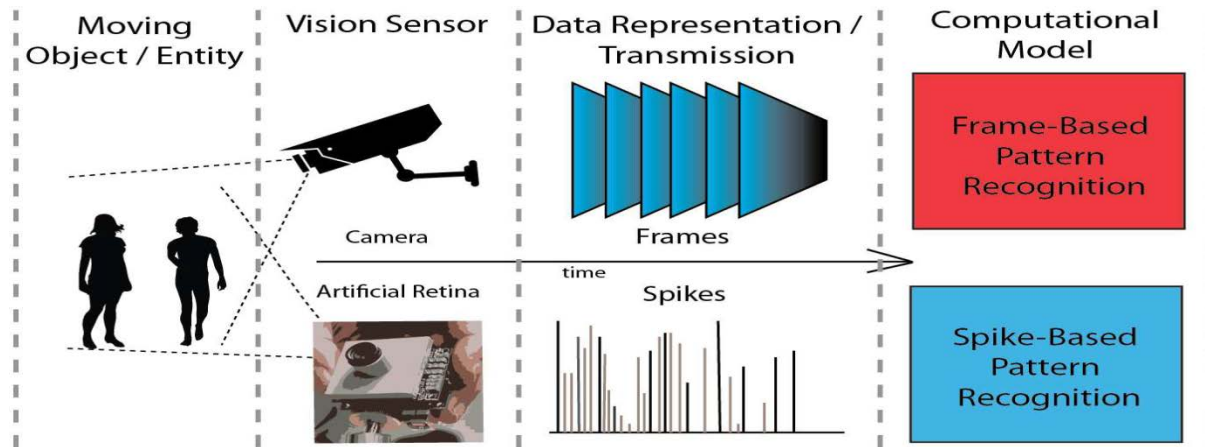
The EvoSpike Simulator

A collection of modules and functions written in Python using functions from Brian library:

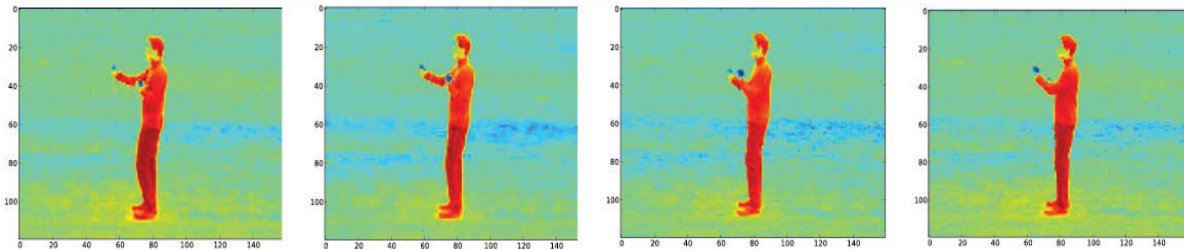
- Converting continuous-value input data into spike trains;
- SNN for spatio-temporal pattern recognition (SPAN, deSNN, LSM deSNN, ...);
- Knowledge extraction from trained eSNN;
- Presenting results and visualisation of learning processes ;
- Connecting software modules with neuromorphic hardware.



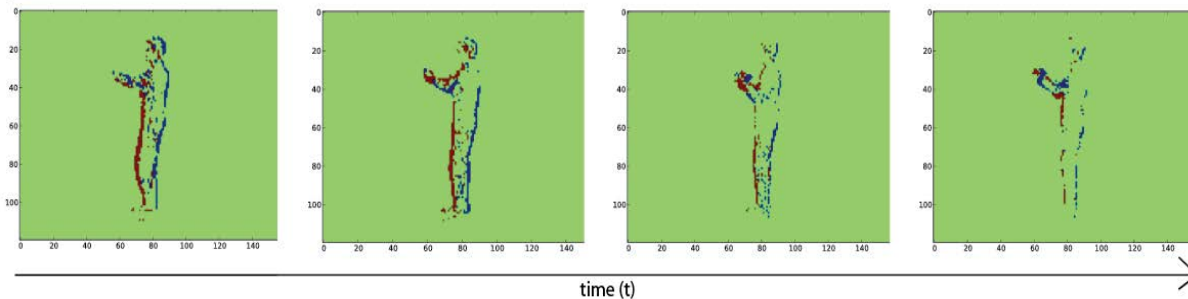
Moving object recognition – frame-based vs AER



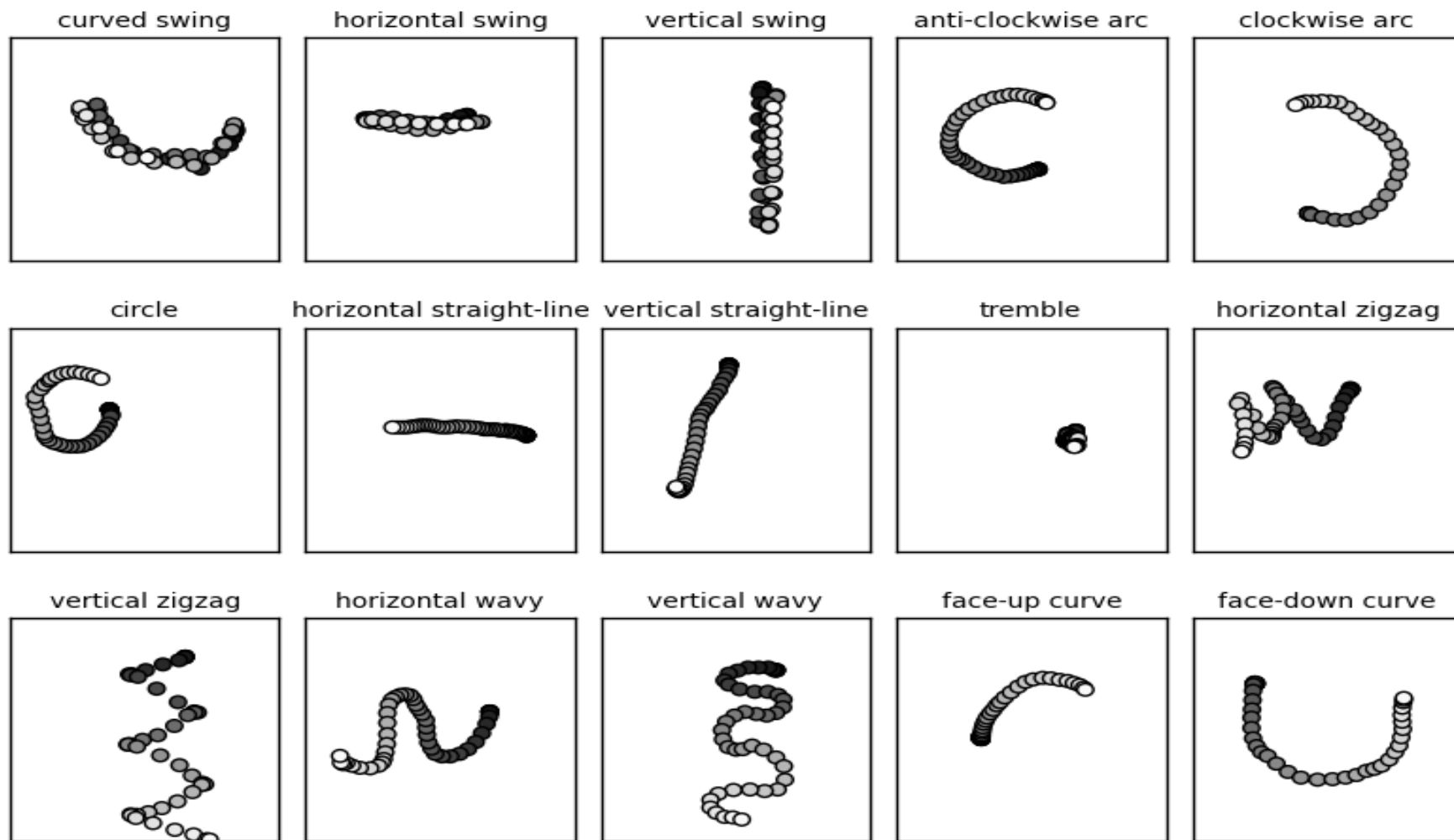
a) Disparity Map of a Video Sample



b) Address Event Representation (AER) of the above Video Sample

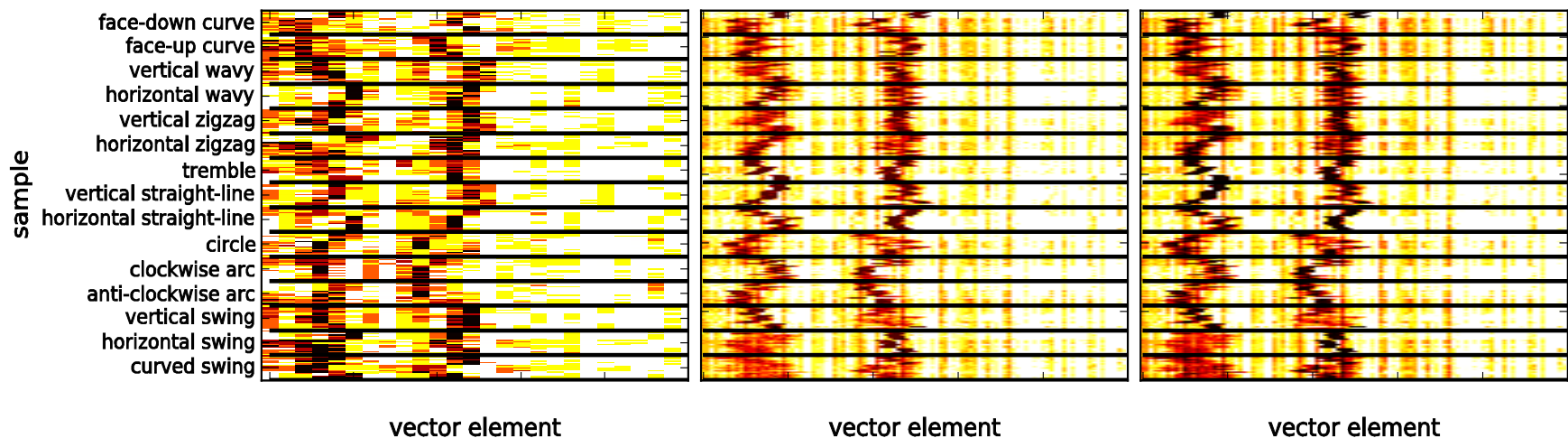
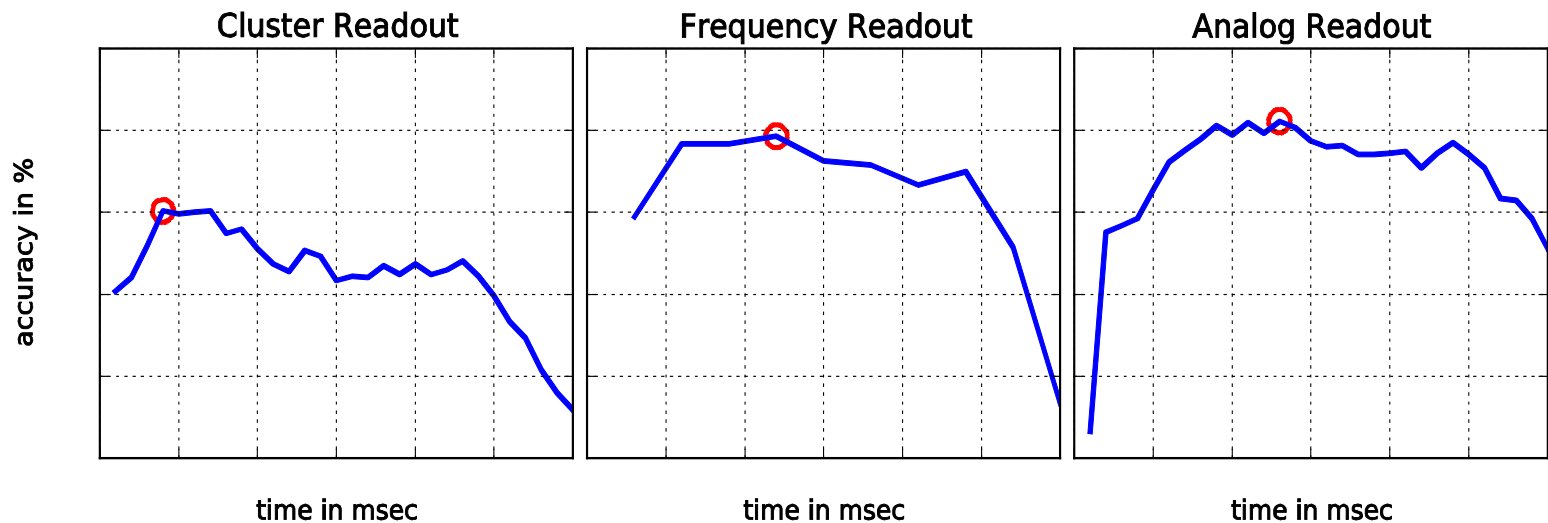


Brazilian Sign Language LIBRAS Pattern Recognition



A single sample for each of the 15 classes is shown. The colour indicates the spatial position in 2D of a single point in time (black/white corresponds to earlier/later time points).

LIBRAS recognition with LSM reservoir and eSNN classifier using different methods to read the state of the LSM (Schliebs, Nuzlu and Kasabov, ICONIP 2011)



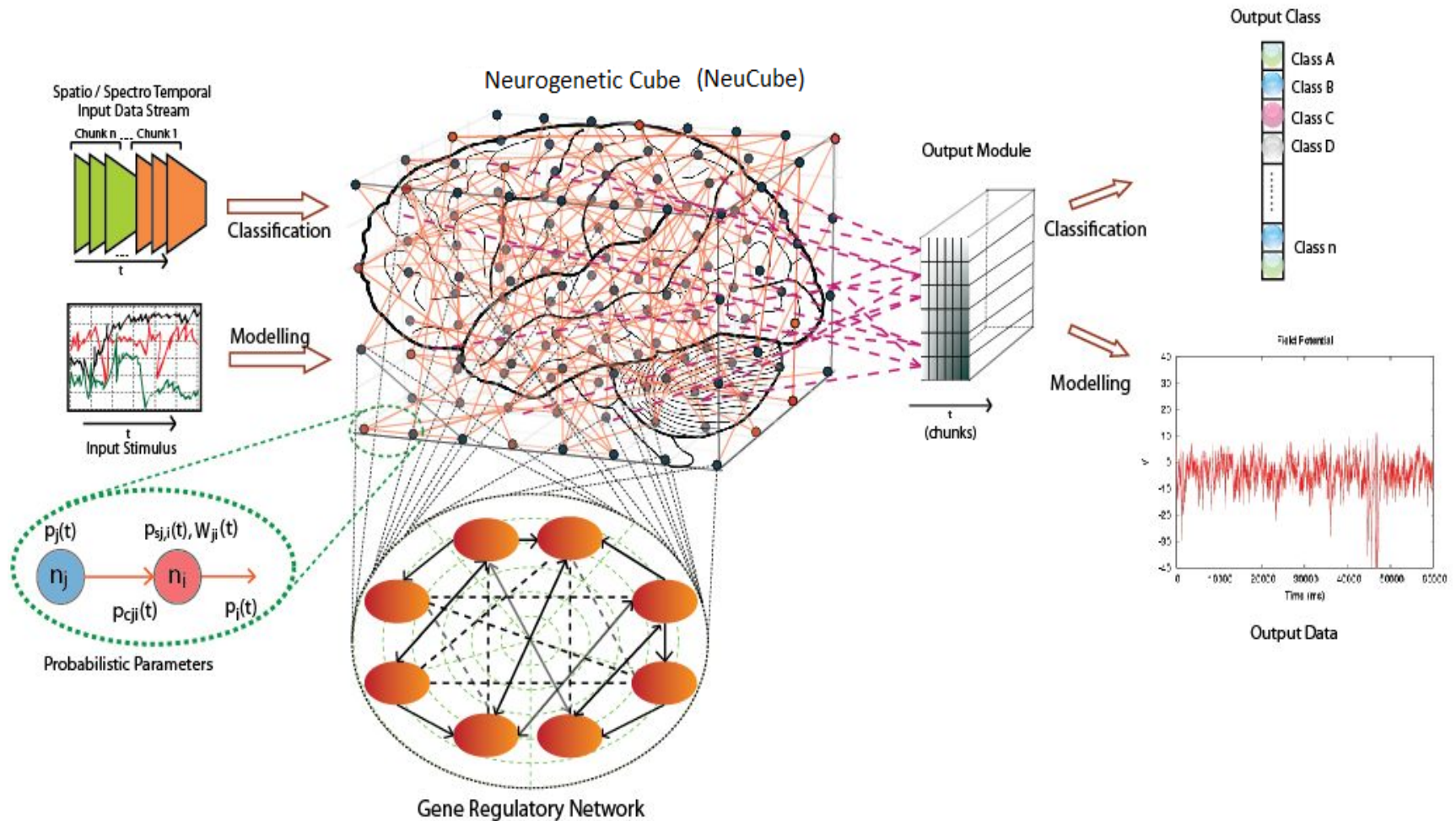
Adaptive, autonomous robot control

(e.g. work by P.Kormushev – IIT, Genoa; R.Duro – U. la Coruna, P.Angelov – U.Lancaster; KIT Japan; U.Ulster, NASA,)



A NeuCube Framework and a Simulator for Brain Data Modelling and brain STPR

(Kasabov, Springer LNAI 7477, 2012; Kasabov, NN 2013)



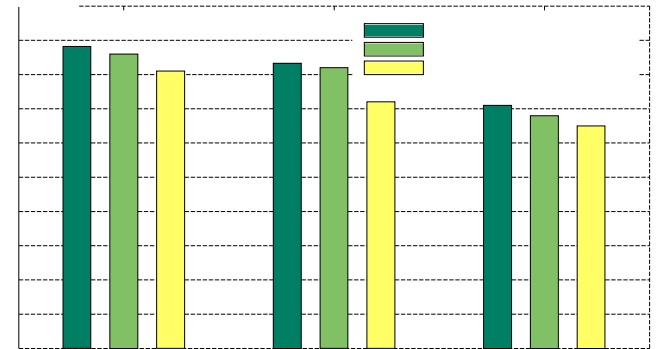
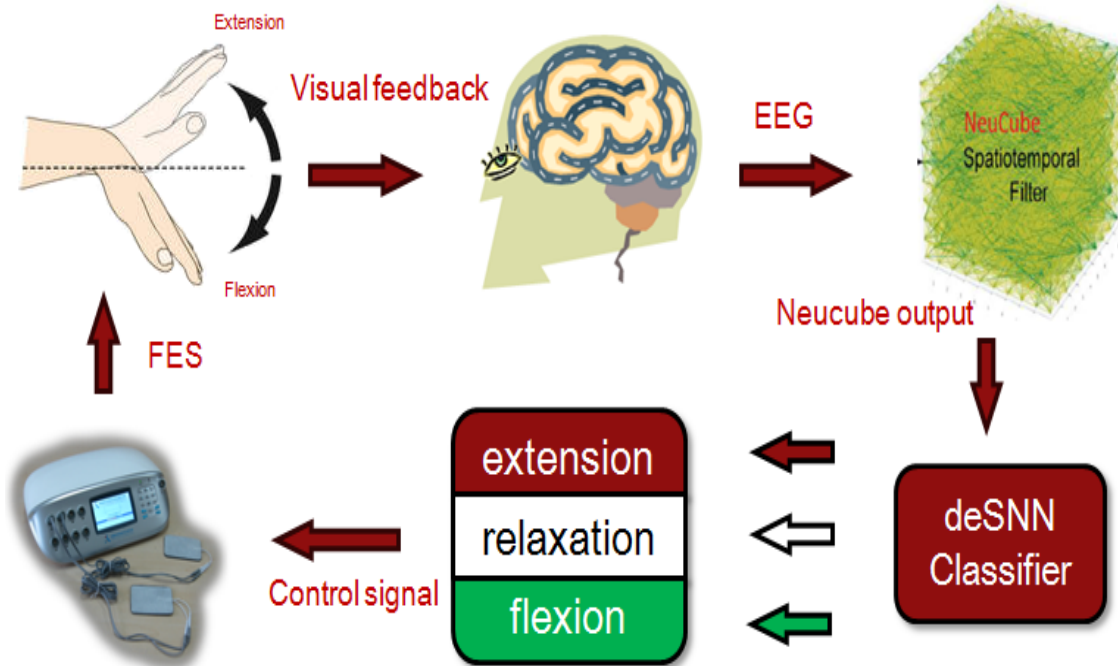
NeuCube Implementation

- Module input data encoding into spikes (e.g. AER)
- Module 3D reservoir (e.g. 1471 neurons)
- Module classifier (e.g. deSNN)

The screenshot displays the NeuCube software interface, which is divided into several functional areas:

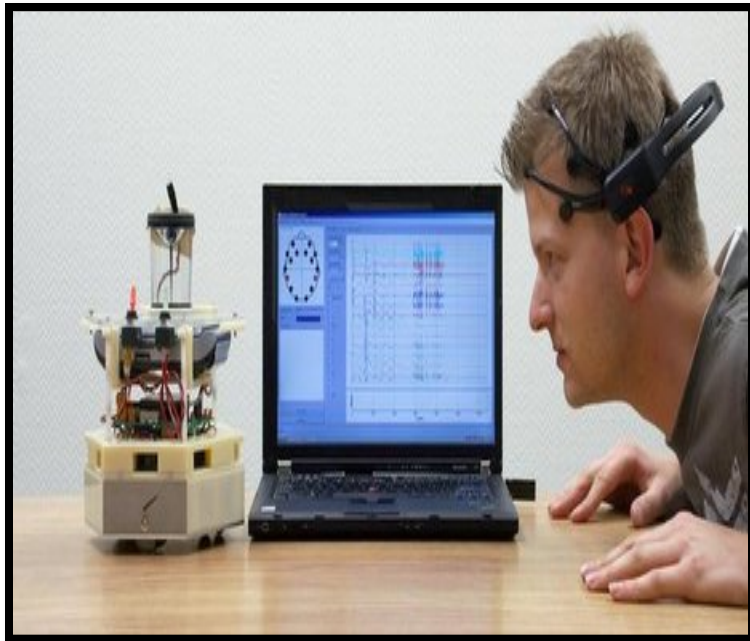
- Setting Panel:** Contains various parameters for training and visualization. Parameters include: AER encoding threshold (0.1), Percentage of the sample used for training (0.6), Number of Samples (40), Number of classes (2), Activity index (0.276), Visualization (NO), Threshold of neural (0.5), Potential leak rate (0.002), STDP rate (0.01), Time to train (1), and SWC (0.15). Buttons for "Load Sample" and "Initialize NeuCube" are present.
- Classifier Panel:** Shows a bar chart with two bars representing class probabilities. The classifier is set to "de...", Mod is 0.4, and Drift is 0.25. A "Cross check" list shows a sequence of 1s and 2s. A "Training and validation" button is at the bottom.
- NeuCube Visualization:** A 3D plot showing the internal structure of the reservoir with neurons and connections. The axes range from -50 to 50. Below the plot, it indicates "NeuCube structure" and "Training progress (%): 100".
- Control Panel:** Includes buttons for "Record" (NO), "The neural to be tracked" (555), "Training neuCube", "Save neuCube", "Load neuCube", and "Replay".
- Monitoring Panel:** Features "Show connection" (0.09), "Activity Level Plot", "Recall" (1), and "Activity Level 3D".
- Display Panel:** Includes "The neural to be displayed" (555), "Weight type" (Pre), and a "Show weight" button.
- Help Panel:** Displays the message "Validation finished".
- Footer:** Includes the logo for Knowledge Engineering & Discovery Research Institute (KEDRI) and the text "AN INSTITUTE OF AUT UNIVERSITY".

NeuCube for Neurorehabilitation



NeuCube for BCI

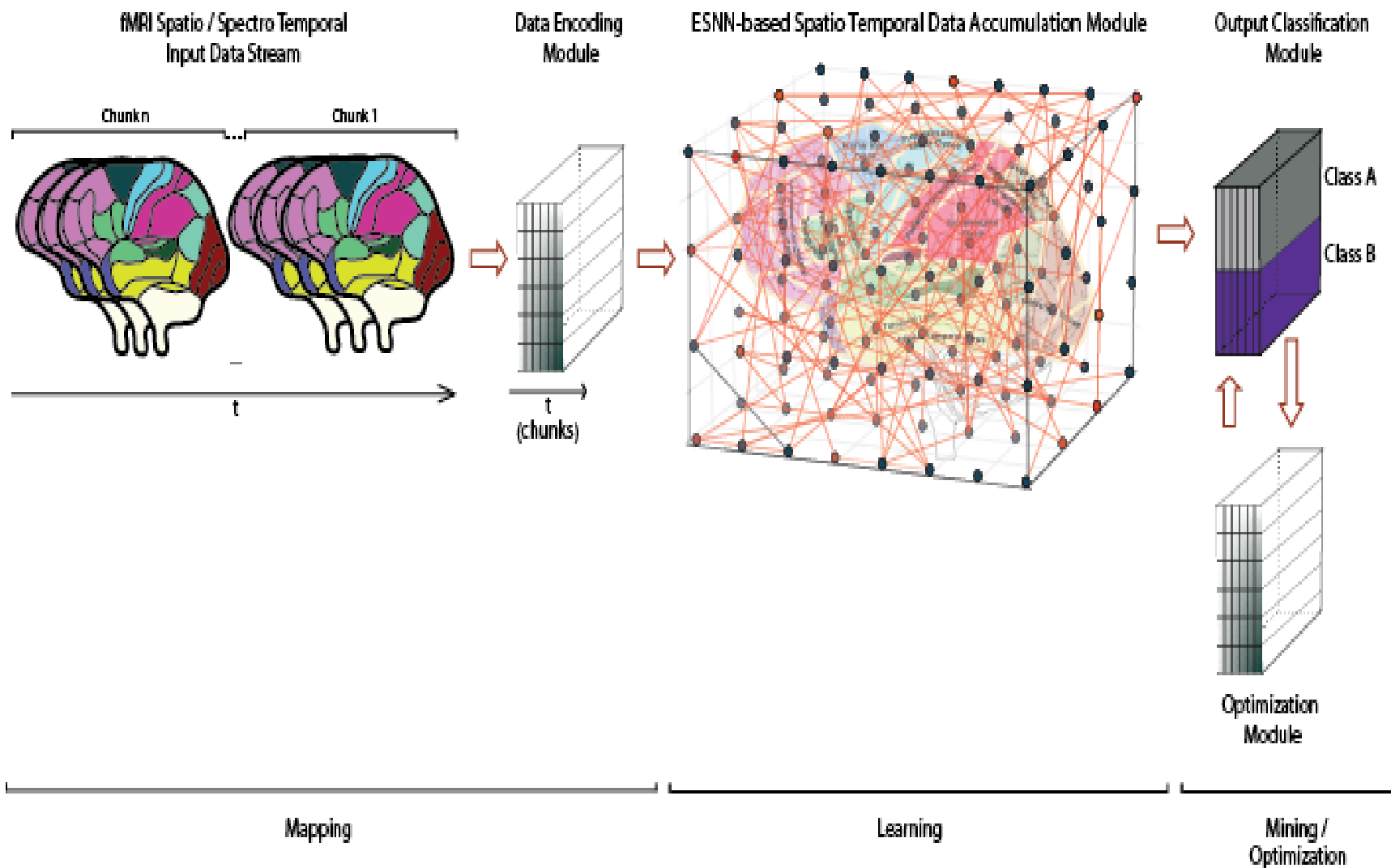
- Brain-Computer Interfaces (BCIs) are interfaces that allow humans to communicate directly with computers or external devices through their brains (e.g. EEG signals)
- Experiments with the WITH robot from KIT, prof. Yamakawa (S.Schliebs)
- Neuro-rehabilitation and neuro-prosthetics (with CAS, Z-G Hou)
- Collaborative work with U.Aveiro (P.Georgieva)



<http://www.nzherald.co.nz>

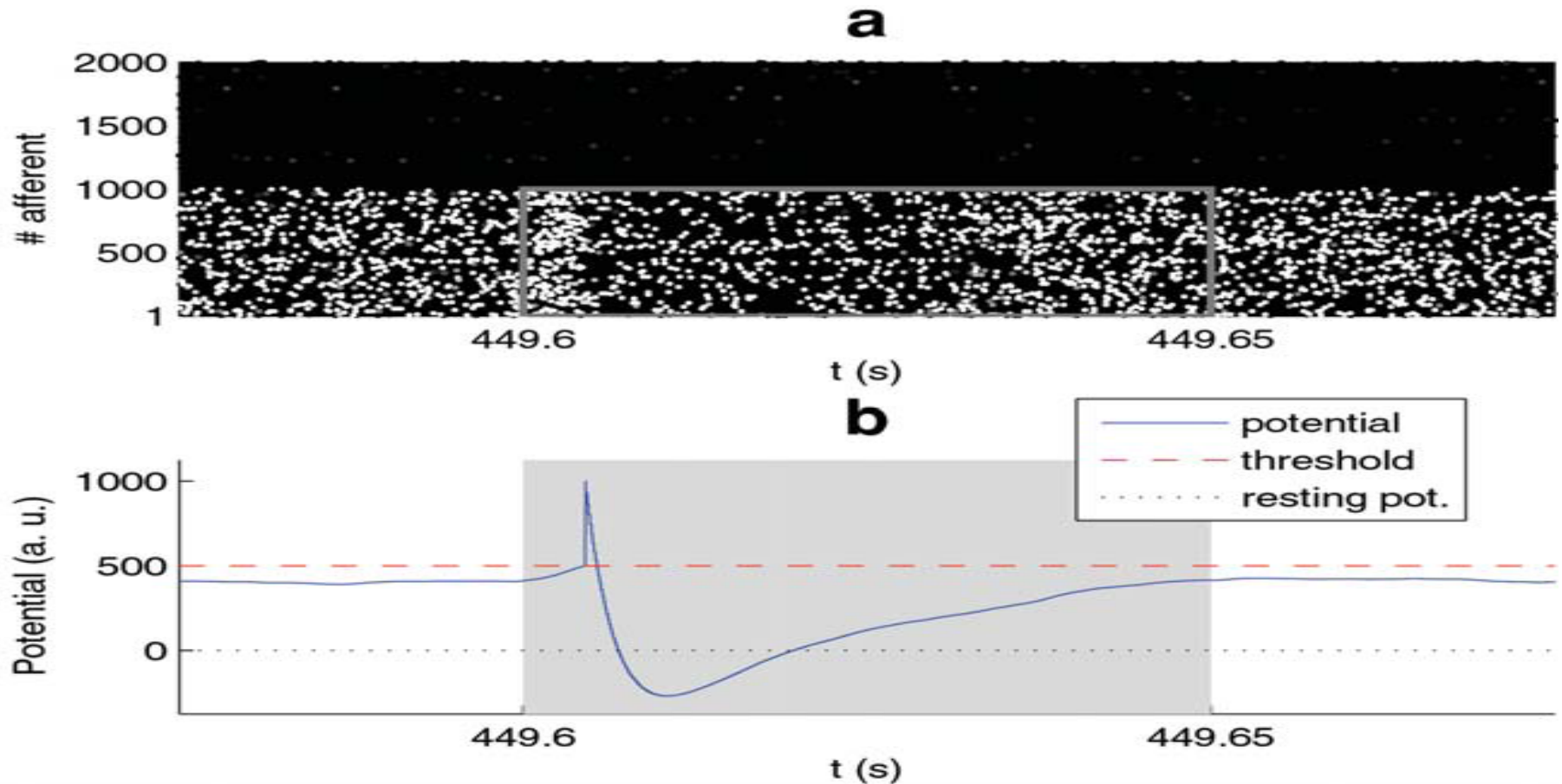


NeuCube for fMRI STPR



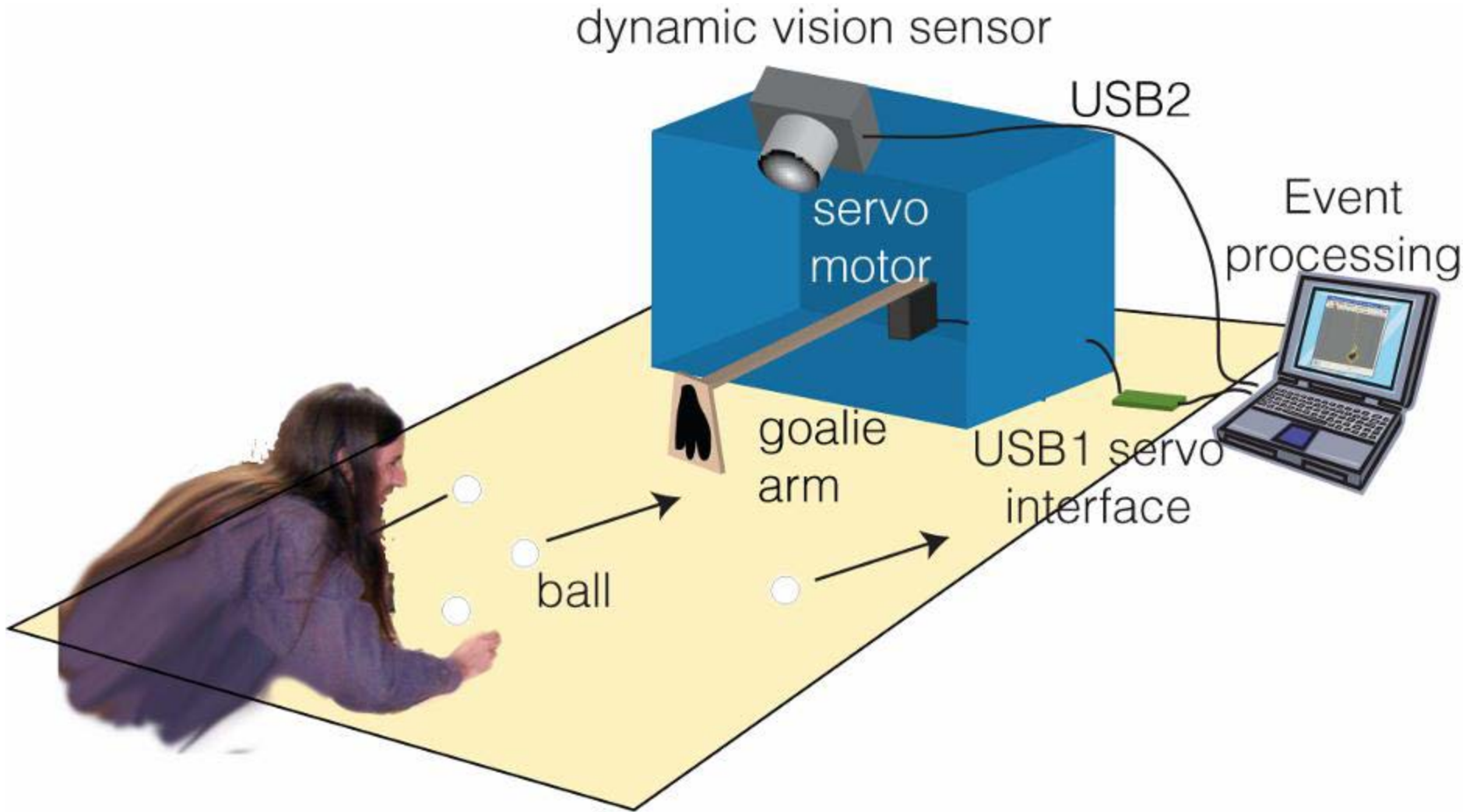
5. eSNN for Early Detection of Events from Spatio-Temporal Data

A single LIF neuron with simple synapses can be trained with the STDP unsupervised rule to discriminate a repeating pattern of synchronised spike trains of thousands inputs from noise (T. Masquelier, R. Guyonneau and S. Thorpe, PlosONE, Jan2008)



Early detection of a moving object with DVS

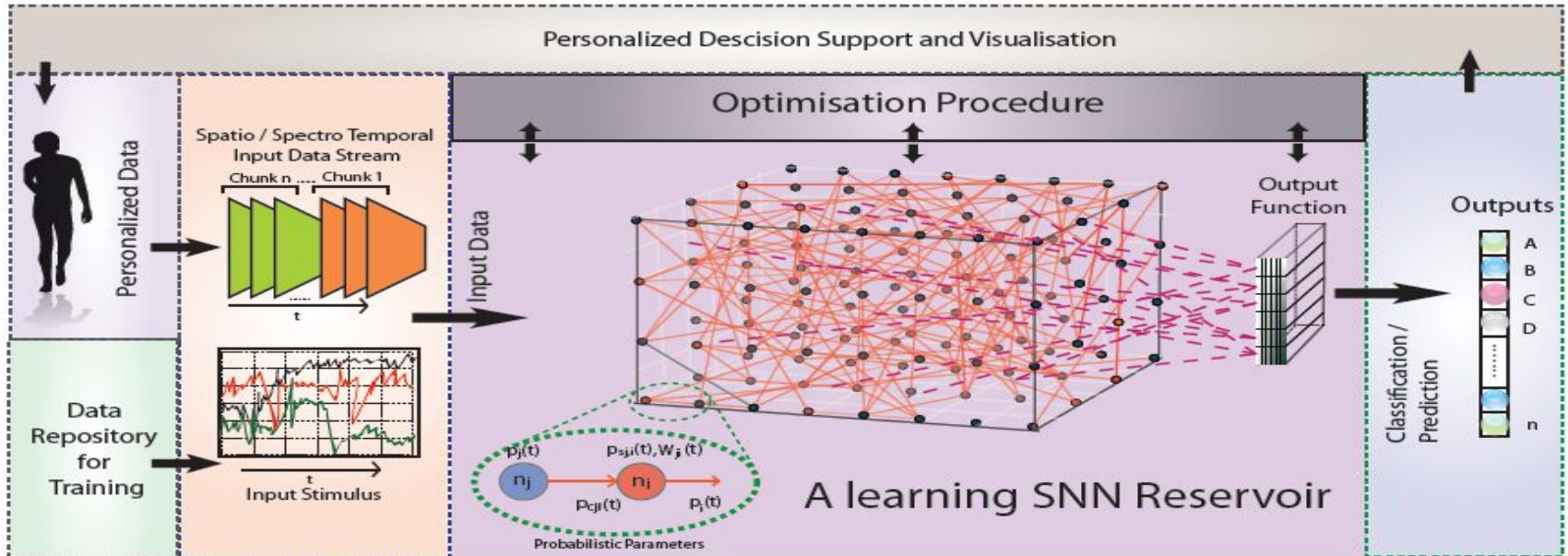
(T.Delbruck, INI, ETH. Zurich)



Personalised Predictive Systems

Can individual risk of stroke be predicted before the event occurrence?

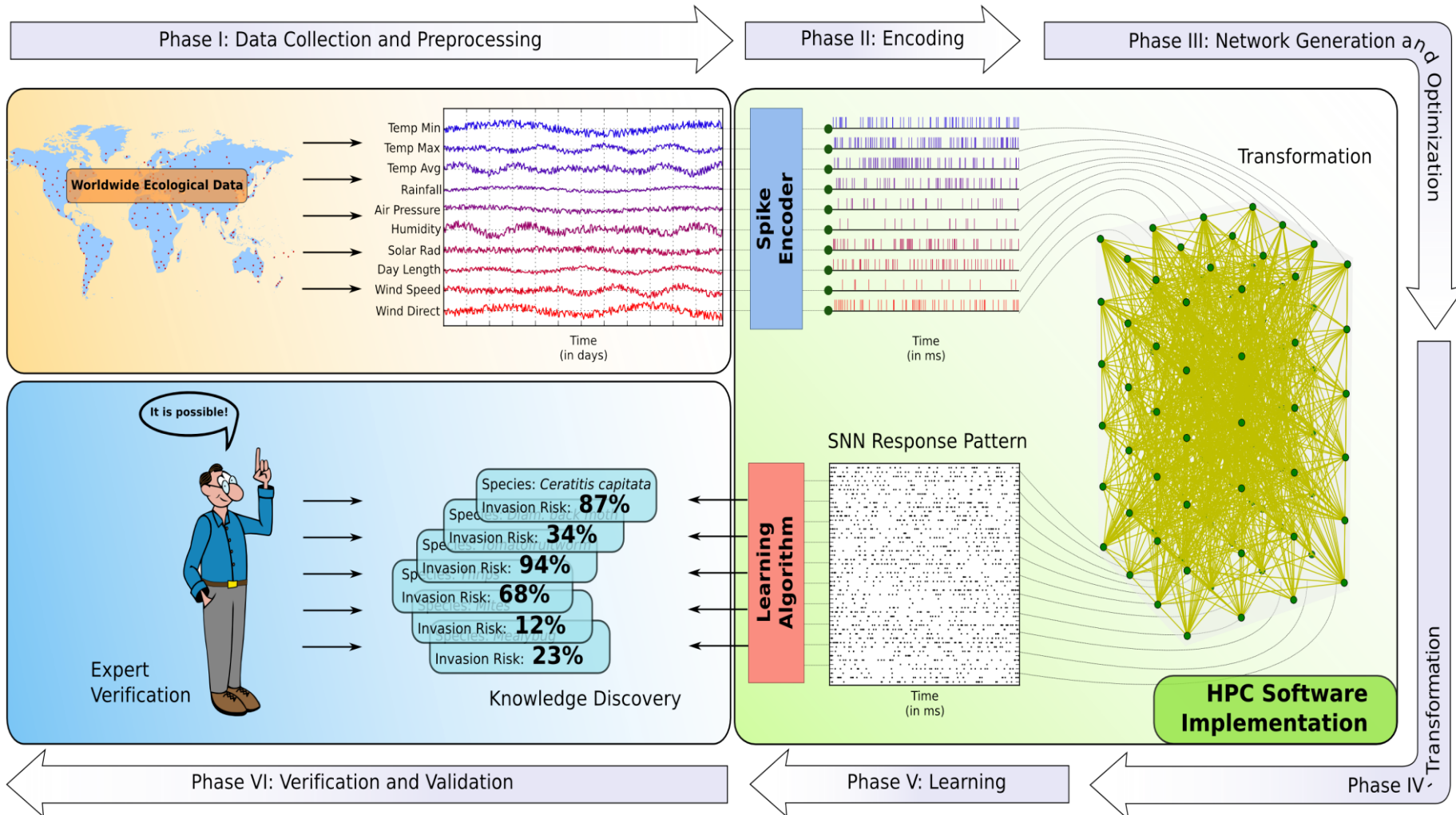
N. Kasabov and R.Hu, Integrated Optimisation Method for Personalised Modelling and Case Study Applications for Medical Decision Support, *J. Functional Informatics and Personalised Medicine*, 2011



Method	Overall accuracy (%)	TP – stroke prediction (%)	TN – no stroke (%)
Multiple Linear regression (MLR)	67.50	65	70
SVM	72.5	65	80
MLP	87.5	85	90
PMeSNNr	94	88	100

Early estimation of risk of establishment of invasive species on a certain *location* at a certain *time*

(S.Schliebs, Defoin-Platel, N.Kasabov, S.Worner et al, Neural Networks, No.22, 2009)



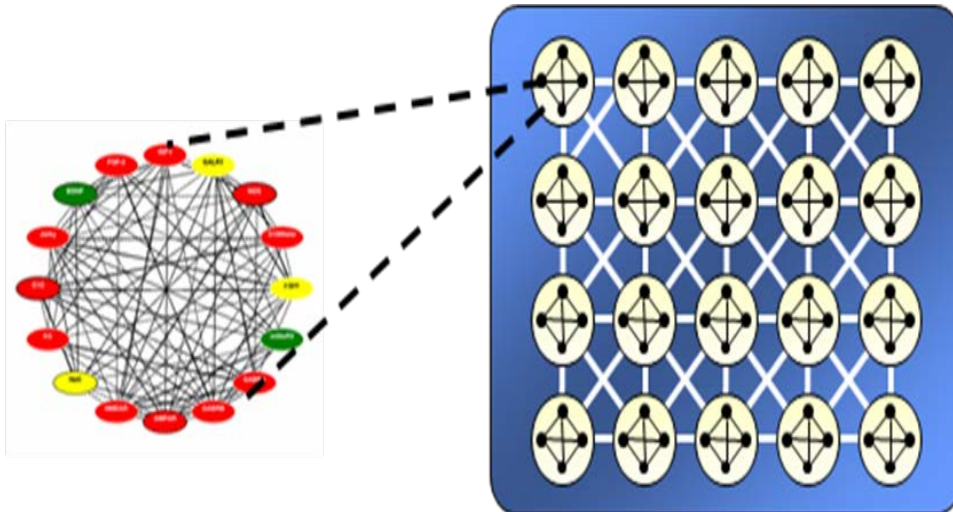
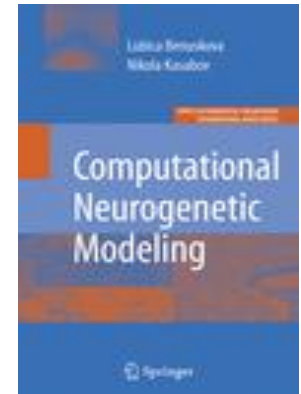
6. Future Directions

Computational Neuro-Genetic Modelling (CNGM)

- Benuskova and Kasabov (2007)

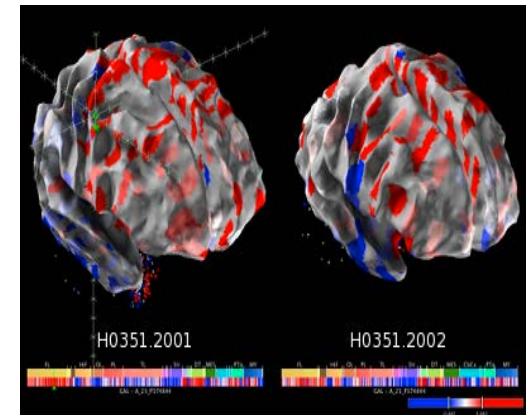
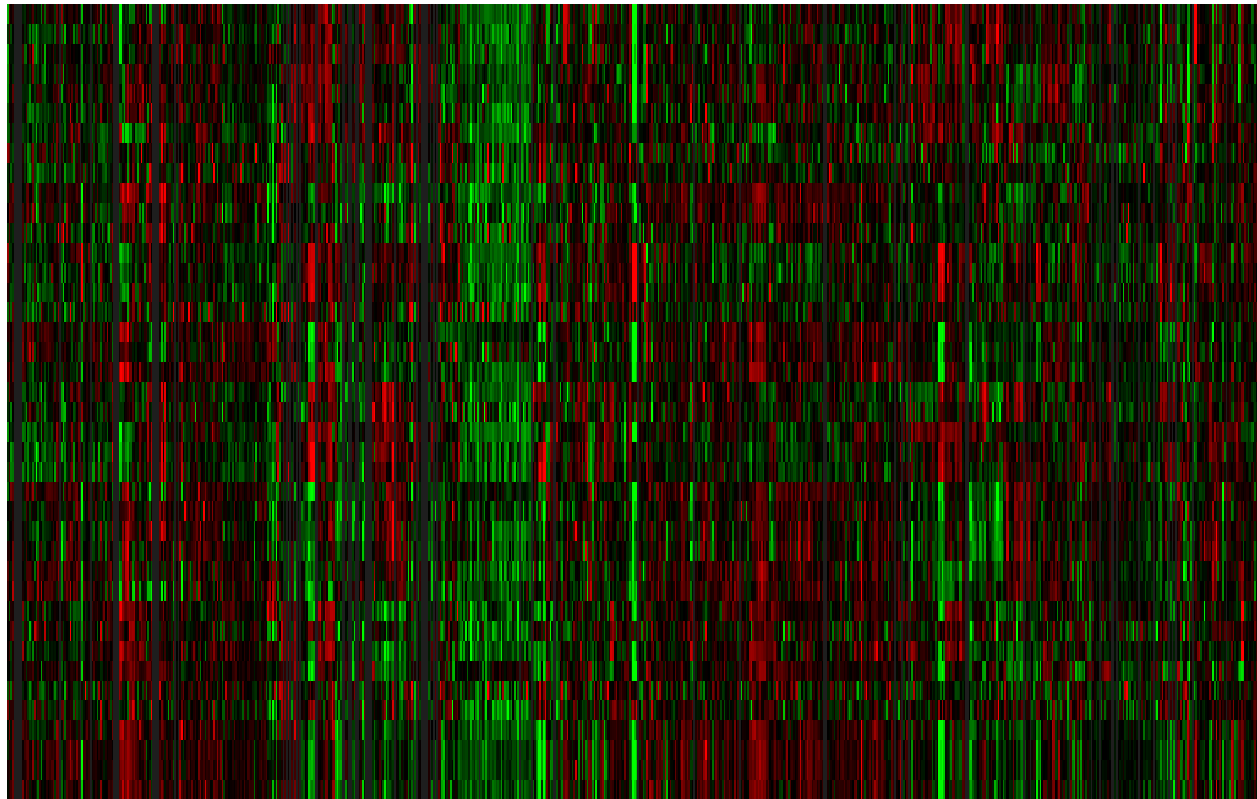
SNN that incorporate a gene regulatory network (GRN) as a dynamic parameter systems to capture dynamic interaction of genes (parameters) related to neuronal activities of the SNN.

- Functions of neurons and neural networks are influenced by internal networks of interacting genes and proteins forming an abstract GRN model.
 - The GRN and the SNN function at different time scales.
 - Mark Sagar's emotional baby



Neurogenetic STBD: The Allen Brain Institute Map

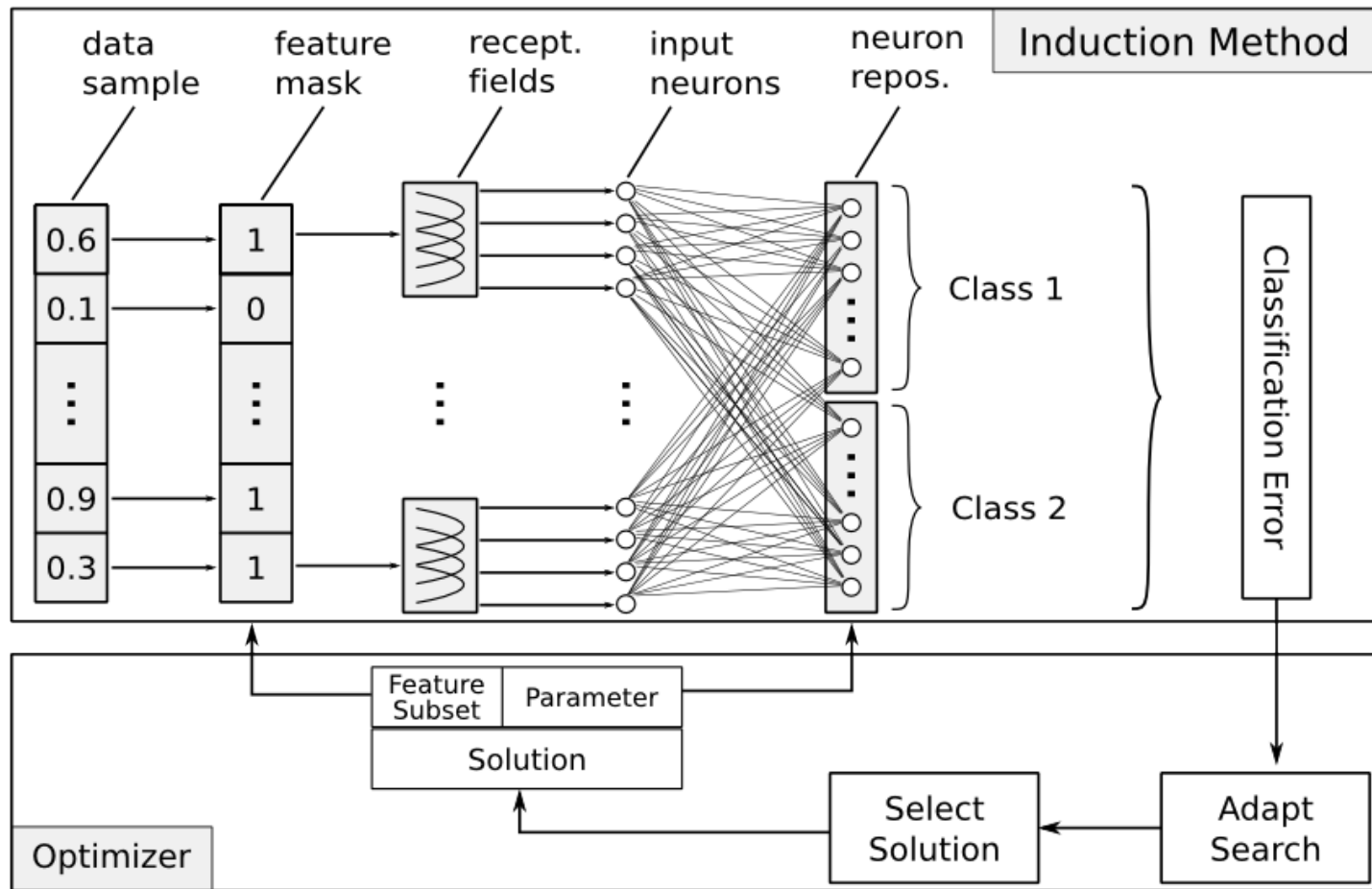
(<http://www.brain-map.org>)



From the Brain Explorer: The Expression level of the genes (on the y-axis): ABAT A_23_P152505, ABAT A_24_P330684, ABAT CUST_52_PI416408490, ALDH5A1 A_24_P115007, ALDH5A1 A_24_P923353, ALDH5A1 A_24_P3761, AR A_23_P113111, AR CUST_16755_PI416261804, AR CUST_85_PI416408490, ARC A_23_P365738, ARC CUST_11672_PI416261804, ARC CUST_86_PI416408490, ARHGEF10 A_23_P216282, ARHGEF10 A_24_P283535, ARHGEF10 CUST_) at different slices of the brain (on the x-axis) (from www.brain-map.org) (<http://www.alleninstitute.org>)

Quantum-inspired EC for the optimisation of eSNN

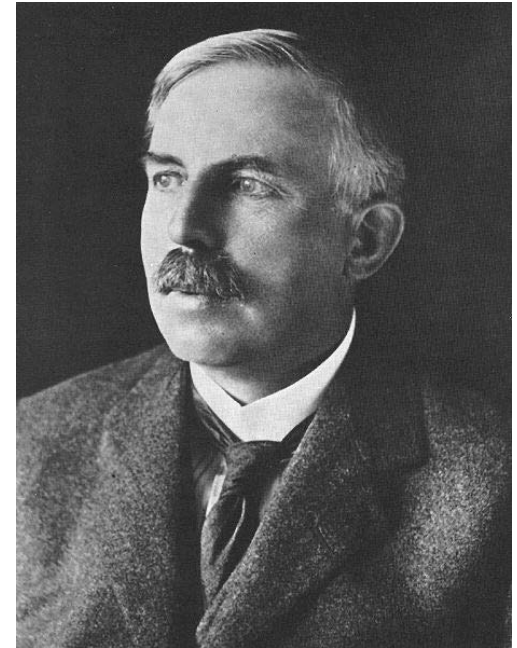
(Kasabov, 2007-2008; S.Schliebs, M.Defoin-Platel and N.Kasabov, 2008)



Quantum Inspired Technologies

- **Quantum principles: superposition; entanglement, interference, parallelism**
 - **Quantum bits (qu-bits)**

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad |\alpha|^2 + |\beta|^2 = 1$$



- **- Quantum vectors (qu-vectors)**

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{array} \right]$$

- **Quantum gates**

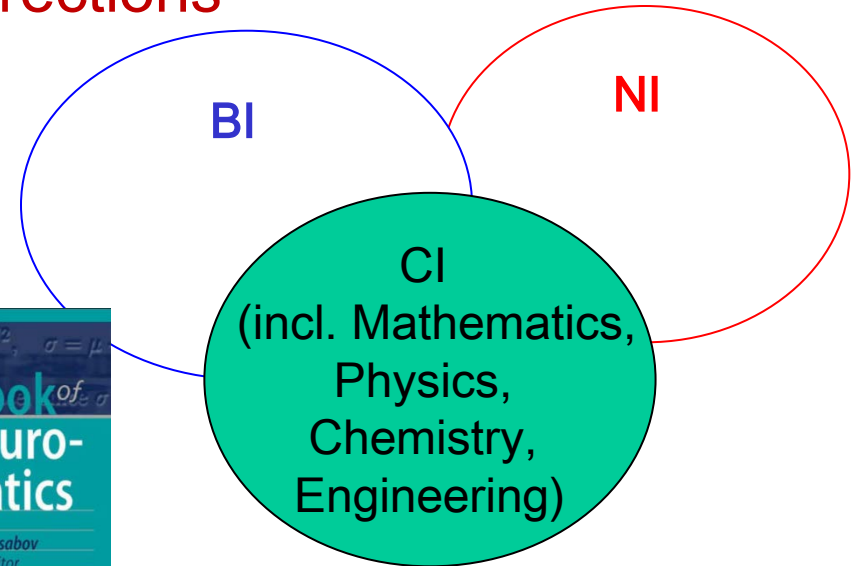
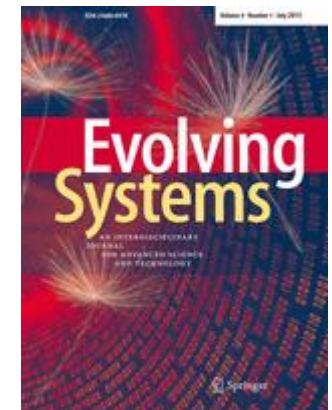
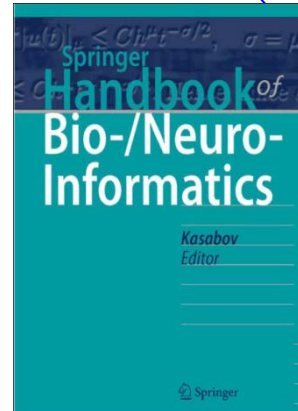
$$\begin{bmatrix} \alpha_i^j(t+1) \\ \beta_i^j(t+1) \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_i^j(t) \\ \beta_i^j(t) \end{bmatrix}$$

- **Applications:**

- Specific algorithms with polynomial time complexity for NP-complete problems (e.g. factorising large numbers, Shor, 1997; cryptography)
- Search algorithms (Grover, 1996), $O(N^{1/2})$ vs $O(N)$ complexity)
- Quantum associative memories
- **Quantum inspired evolutionary algorithms and neural networks**

...Future Directions

- Further interdisciplinary research in the three areas of CI, BI and NI
- The Springer Handbook of Bio-Neuroinformatics, 2013 (N.Kasabov, ed)
- The Springer Series in Bio-Neuroinformatics (N.Kasabov, ed)
- Springer journal *Evolving Systems* (ed. Angelov, Filev, Kasabov)



KEDRI: The Knowledge Engineering and Discovery Research Institute at AUT (www.kedri.aut.ac.nz)



References

- [1] N.Kasabov (2007) *Evolving Connectionist Systems: The Knowledge Engineering Approach*, Springer, London (www.springer.de) (first edition published in 2002)
- [2] S.Wysocki, L.Benuskova, N.Kasabov, *Evolving Spiking Neural Networks for Audio-Visual Information Processing*, *Neural Networks*, vol 23, issue 7, pp 819-835, September 2010.
- [3] N.Kasabov, *To spike or not to spike: A probabilistic spiking neural model*, *Neural Networks*, 23, 1, 2010, 16-19.
- [4] Mohemmed,A., Schliebs,S., Kasabov,N.,SPAN: Spike Pattern Association Neuron for Learning Spatio-Temporal Sequences, *Int. J. Neural Systems*, 2012.
- [5] Kasabov, N., Dhoble, K., Nuntalid, N., G. Indiveri, *Dynamic Evolving Spiking Neural Networks for On-line Spatio- and Spectro-Temporal Pattern Recognition*, *Neural Networks*, 41, 2013, 188-201.
- [6] Benuskova, L and N.Kasabov (2007) *Computational Neurogenetic Modelling*, Springer.
- [7] Defoin-Platel, M., S.Schliebs, N.Kasabov, *Quantum-inspired Evolutionary Algorithm: A multi-model EDA*, *IEEE Trans. Evolutionary Computation*, vol.13, No.6, Dec.2009, 1218-1232
- [8] Nuzly, H., N.Kasabov, S.Shamsuddin (2010) *Probabilistic Evolving Spiking Neural Network Optimization Using Dynamic Quantum Inspired Particle Swarm Optimization*, *Proc. ICONIP 2010, Part I, LNCS*, vol.6443.
- [9] S.Soltic, N.Kasabov, *Knowledge extraction from evolving spiking neural networks with a rank order population coding*, *Int.J.Neural Systems*, Vol. 20, No. 6 (2010) 437-445, World Scientific Publ.
- [10] N.Kasabov (ed) (2013) *The Springer Handbook of Bio- and Neuroinformatics*, Springer, September, 2013..
- [11] Schliebs, Michael Defoin Platel, Susan Worner and Nikola Kasabov, *Integrated Feature and Parameter Optimization for Evolving Spiking Neural Networks: Exploring Heterogeneous Probabilistic Models*, *Neural Networks*, 22, 623-632, 2009.
- [12] S. Pang, Tao Ban, Youki Kadobayashi and Nikola K. Kasabov, *LDA Merging and Splitting with Applications to Multi-agent Cooperative Learning and System Alteration*, *IEEE Transactions On Systems, Man And Cybernetics*, -Part B. 42(2): 552-564, 2012.
- [13] Kasabov, N., *NeuroCube EvoSpike Architecture for Spatio-Temporal Modelling and Pattern Recognition of Brain Signals*, in: Mana, Schwenker and Trentin (Eds) *ANNPR, Springer LNAI 7477*, 2012, 225-243
- [14] Furber, S., *To Build a Brain*, *IEEE Spectrum*, vol.49, Number 8, 39-41, 2012.