Regular Paper

Evolving, Probabilistic Spiking Neural Networks and Neurogenetic Systems for Spatio- and Spectro-Temporal Data Modelling and Pattern Recognition

Nikola Kasabov, FIEEE, FRSNZ

Knowledge Engineering and Discovery Research Institute - KEDRI, Auckland University of Technology,
and Institute for Neuroinformatics - INI, ETH and University of Zurich
nkasabov@aut.ac.nz; www.kedri.info, ncs.ethz.ch/projects/evospike

Abstract

Spatio- and spectro-temporal data (SSTD) are the most common types of data collected in many dominant areas, including engineering, bioinformatics, neuroinformatics, ecology, environment, medicine, economics, etc. However, there is lack of methods for the efficient analysis of such data and for spatio-temporal pattern recognition (STPR). The brain functions as a spatio-temporal information processing machine and deals extremely well with spatio-temporal data. Its organisation and functions have been the inspiration for the development of new methods for SSTD analysis and STPR. The brain-inspired spiking neural networks (SNN) are considered the third generation of neural networks and are a promising paradigm for the creation of new intelligent ICT for SSTD. This new generation of computational models and systems are potentially capable of modelling complex information processes due to their ability to represent and integrate different information dimensions, such as time, space, frequency, and phase, and to deal with large volumes of data in an adaptive and self-organising manner. The paper reviews methods and systems of SNN for SSTD analysis and STPR, including single neuronal models, evolving spiking neural networks (eSNN) and computational neuro-genetic models (CNGM). Software and hardware implementations and some pilot applications for audio-visual pattern recognition, EEG data analysis, cognitive robotic systems, BCI, neurodegenerative diseases, and others are discussed.

Keywords: Spatio-temporal data, spectro-temporal data, pattern recognition, spiking neural networks, gene regulatory networks, computational neuro-genetic modeling, probabilistic modeling, personalized modelling; EEG data.

1. Spatio- and Spectro-Temporal Data Modeling and Pattern Recognition

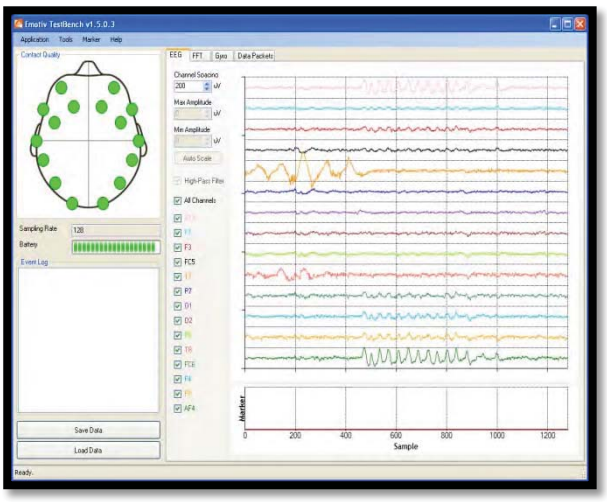
Most problems in nature require spatio- or/and spectro-temporal data (SSTD) that include measuring spatial or/and spectral variables over time. SSTD is described by a triplet (**X**,**Y**,**F**), where **X** is a set of independent variables measured over consecutive discrete time moments t; **Y** is the set of dependent output variables, and **F** is the association function between whole segments ('chunks') of the input data, each sampled in a time window d_t, and the output variables belonging to Y:

$$\mathbf{F}: \mathbf{X}(d_t) \rightarrow \mathbf{Y}, \quad \mathbf{X}(t)=(\mathbf{x}_1(t),\mathbf{x}_2(t),\ldots,\mathbf{x}_n(t)),\ t=1,2,\ldots,n \quad (1)$$

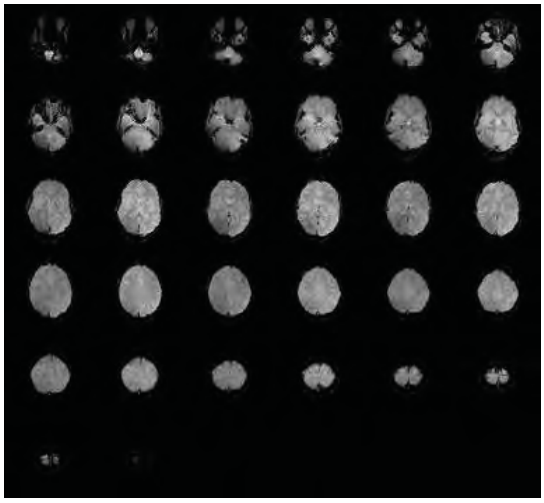
It is important for a computational model to capture and learn *whole* spatio- and spectro-temporal patterns from data streams in order to predict most accurately future events for new input data. Examples of problems involving SSTD are: brain cognitive state evaluation based on spatially distributed EEG electrodes [70, 26, 51, 21, 99, 100] (Fig.1(a)); fMRI data [102] (Fig.1(b)); moving object recognition from video data [23, 60, 25] (Fig.15); spoken word recognition based on spectro-temporal audio data [93, 107]; evaluating risk of disease, e.g. heart attack [20]; evaluating response of a disease to treatment based on clinical and environmental variables, e.g. stroke [6]; prognosis of outcome of cancer [62]; modelling the progression of a neuro-degenerative disease, such as Alzheimer's Disease [94, 64]; modelling and prognosis of the establishment of invasive species in ecology [19, 97]. The prediction of events in geology, astronomy, economics and many other areas also depend on accurate SSTD modeling.

The commonly used models for dealing with temporal information based on Hidden Markov Models (HMM) [88] and traditional artificial neural networks (ANN) [57] have limited capacity to achieve the integration of complex and long temporal spatial/spectral components because they usually either ignore the temporal dimension or over-simplify its representation. A new trend in machine learning is currently emerging and is known as *deep machine learning* [9, 24, 112]. Most of the proposed models still learn SSTD by entering single time point frames rather than learning whole SSTD patterns. They are also limited in addressing adequately the interaction between temporal and spatial components in SSTD.

The human brain has the amazing capacity to learn and recall patterns from SSTD at different time scales, ranging from milliseconds to years and possibly to millions of years (e.g. genetic information, accumulated through evolution). Thus the brain is the ultimate inspiration for the development of new machine learning techniques for SSTD



(a)



(b)

Fig.1(a) EEG SSTD recorded with the use of Emotive EEG equipment (from McFarland, Anderson, Müller, Schlögl, Krusienski, 2006); (b) fMRI data (from http://www.fmrib.ox.ac.uk)

modelling. Indeed, brain-inspired Spiking Neural Networks (SNN) [32, 33, 68] have the potential to learn SSTD by using trains of spikes (binary temporal events) transmitted among spatially located synapses and neurons. Both spatial and temporal information can be encoded in an SNN as locations of synapses and neurons and time of their spiking activity respectively. Spiking neurons send spikes via connections that have a complex dynamic behaviour, collectively forming an SSTD memory. Some SNN employ specific learning rules such as Spike-Time-Dependent-Plasticity (STDP) [103] or Spike Driven Synaptic Plasticity (SDSP) [30]. According to the STDP a connection weight between two neurons increases when the pre-synaptic neuron spikes before the postsynaptic one. Otherwise, the weight decreases.

Models of single neurons as well as computational SNN models, along with their respective applications, have been already developed [33, 68, 73, 7, 8, 12], including evolving connectionist systems and evolving spiking neural networks (eSNN) in particular, where an SNN learns data incrementally by one-pass propagation of the data via creating and merging spiking neurons [61, 115]. In [115] an eSNN is designed to capture features and to aggregate them into audio and visual perceptions for the purpose of person authentification. It is based on four levels of feed-forward connected layers of spiking neuronal maps, similarly to the way the *cortex* works when learning and recognising images or complex input stimuli [92]. It is an SNN realization of some computational models of vision, such as the 5-level HMAX model inspired by the information processes in the cortex [92].

However, these models are designed for (static) object recognition (e.g. *a picture of a cat*), but not for moving object recognition (e.g. *a cat jumping to catch a mouse*). If these models are to be used for SSTD, they will still process SSTD as a sequence of static feature vectors extracted in single time frames. Although an eSNN accumulates incoming information carried in each consecutive frame from a pronounced word or a video, through the increase of the membrane potential of output spike neurons, they do not learn complex spatio/spectro-temporal associations from the data. Most of these models are deterministic and do not allow to model complex stochastic SSTD.

In [63, 10] a computational neuro-genetic model (CNGM) of a single neuron and SNN are presented that utilize information about how some proteins and genes affect the spiking activities of a neuron, such as fast excitation, fast inhibition, slow excitation, and slow inhibition. An important part of a CNGM is a dynamic gene regulatory network (GRN) model of genes/proteins and their interaction over time that affect the spiking activity of the neurons in the SNN. Depending on the task, the genes in a GRN can represent either biological genes and proteins (for biological applications) or some system parameters including probability parameters (for engineering applications).

Recently some new techniques have been developed that allow the creation of new types of computational models, e.g.: probabilistic spiking neuron models [66, 71]; probabilistic optimization of features and parameters of eSNN [97, 96]; reservoir computing [73, 108]; personalized modelling frameworks [58, 59]. This paper reviews methods and systems for SSTD that utilize the above and some other contemporary SNN techniques along with their applications.

2. Single Spiking Neuron Models

2.1 A biological neuron

A single biological neuron and the associated synapses is a complex information processing machine, that involves short term information processing, long term information storage, and evolutionary information stored as genes in the nucleus of the neuron (Fig.2).

2.2 Single neuron models

Some of the-state-of-the-art models of a spiking neuron include: early models by Hodgkin and Huxley [41] 1952;

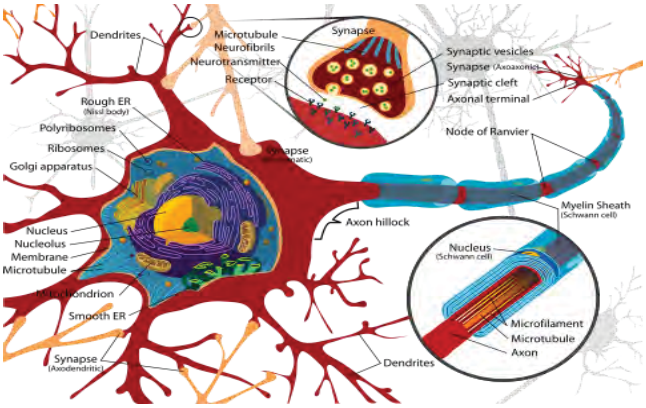


Fig.2. A single biological neuron with the associated synapses is a complex information processing machine (from Wikipedia)

more recent models by Maas, Gerstner, Kistler, Izhikevich and others, e.g.: Spike Response Models (SRM) [33, 68]; Integrate-and-Fire Model (IFM) [33, 68]; Izhikevich models [52-55], adaptive IFM, and others.

The most popular for both biological modeling and engineering applications is the IFM. The IFM has been realised on software-hardware platforms for the exploration of patterns of activities in large scale SNN under different conditions and for different applications. Several large scale architectures of SNN using IFM have been developed for modeling brain cognitive functions and engineering applications. Fig. 3(a) and (b) illustrate the structure and the functionality of the Leaky IFM (LIFM) respectively. The neuronal post synaptic potential (PSP), also called membrane potential u(t), increases with every input spike at a time *t* multiplied to the synaptic efficacy (strength) until it reaches a threshold. After that, an output spike is emitted and the membrane potential is reset to an initial state (e.g. 0). Between spikes, the membrane potential leaks, which is defined by a parameter.

An important part of a model of a neuron is the model of the synapses. Most of the neuronal models assume scalar synaptic efficacy parameters that are subject to learning, either on-line or off-line (batch mode). There are models of dynamics synapses (e.g. [67, 71, 72]), where the synaptic efficacy depends on synaptic parameters that change over time, representing both long term memory (the final efficacy after learning) and short term memory – the changes of the synaptic efficacy over a shorter time period not only during learning, but during recall as well.

One generalization of the LIFM and the dynamic synaptic models is the probabilistic model of a neuron [66] as shown in fig.4a, which is also a biologically plausible model [45, 68, 71]. The state of a spiking neuron n_i is described by the sum $PSP_i(t)$ of the inputs received from all m synapses. When the $PSP_i(t)$ reaches a firing threshold $\vartheta_i(t)$, neuron n_i fires, i.e. it emits a spike. Connection weights ($w_{j,i}$, j=1,2,...,m) associated with the synapses are determined during the learning phase using a learning rule. In addition to the connection weights $w_{j,i}(t)$, the probabilistic spiking neuron model has the following three probabilistic parameters:

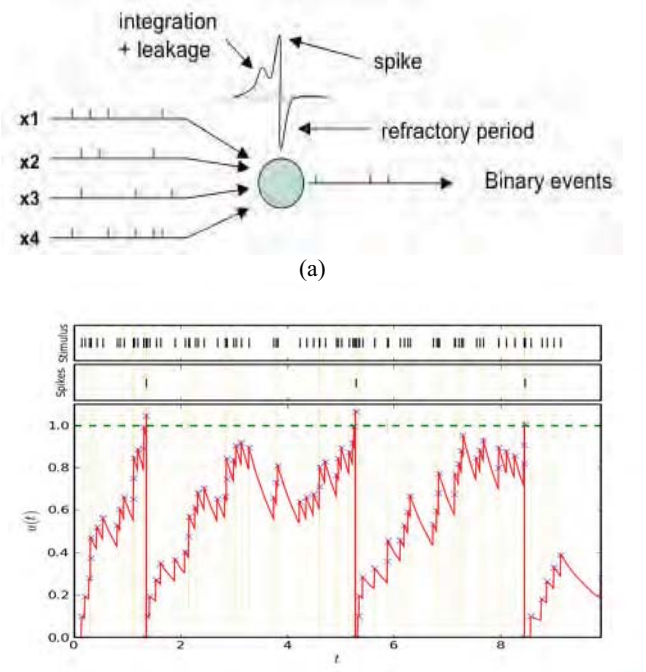


Fig.3. (a) The structure of the LIFM. (b) functionality of the LIFM

- A probability $p_{cj,i}(t)$ that a spike emitted by neuron n_j will reach neuron n_i at a time moment *t* through the connection between n_j and n_i. If $p_{cj,i}(t)=0$, no connection and no spike propagation exist between neurons n_j and n_i. If $p_{cj,i}(t) = 1$ the probability for propagation of spikes is 100%.
- A probability $p_{sj,i}(t)$ for the synapse $s_{j,i}$ to contribute to the PSPi(t) after it has received a spike from neuron n_j.
- A probability $p_i(t)$ for the neuron n_i to emit an output spike at time *t* once the total $PSP_i(t)$ has reached a value above the PSP threshold (a noisy threshold).

The total $PSP_i(t)$ of the probabilistic spiking neuron n_i is now calculated using the following formula [66]:

$$PSP_i(t) = \sum_{p=t_0,..,t} \left(\sum_{j=1,..,m} e_j f_1(p_{cj,i}(t-p)) f_2(p_{sj,i}(t-p)) w_{j,i}(t) + \eta(t-t_0) \right) \quad (2)$$

where e_j is 1, if a spike has been emitted from neuron n_j, and 0 otherwise; $f_1(p_{cj,i}(t))$ is 1 with a probability $p_{cji}(t)$, and 0 otherwise; $f_2(p_{sj,i}(t))$ is 1 with a probability $p_{sj,i}(t)$, and 0 otherwise; t_0 is the time of the last spike emitted by n_i; $\eta(t-t_0)$ is an additional term representing decay in the PSP_i. As a special case, when all or some of the probability parameters are fixed to "1", the above probabilistic model will be simplified and will resemble the well known IFM. A similar formula will be used when a leaky IFM is used as a fundamental model, where a time decay parameter is introduced.

It has been demonstrated that SNN that utilises the probabilistic neuronal model can learn better SSTD than traditional SNN with simple IFM, especially in a noisy environment [98, 83]. The effect of each of the above three probabilistic parameters on the ability of a SNN to process

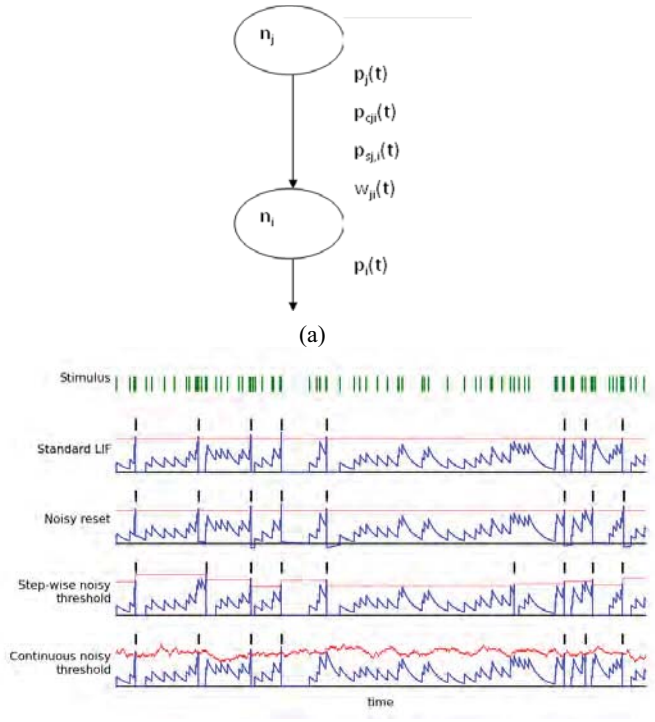


Fig.4 (a) A simple probabilistic spiking neuron model (from [66]); (b) Different types of noisy thresholds have different effects on the output spikes (from [99, 98]).

noisy and stochastic information was studied in [98]. Fig. 4(b) presents the effect of different types of nosy thresholds on the neuronal spiking activity.

2.3 A neurogenetic model of a neuron

A neurogenetic model of a neuron is proposed in [63] and studied in [10]. It utilises information about how some proteins and genes affect the spiking activities of a neuron such as *fast excitation, fast inhibition, slow excitation, and slow inhibition*. Table 1 shows some of the proteins in a neuron and their relation to different spiking activities. For a real case application, apart from the GABAB receptor some other metabotropic and other receptors could be also included. This information is used to calculate the contribution of each of the different synapses, connected to a neuron n_i, to its post synaptic potential PSPi(t):

$$\varepsilon_{ij}^{synapse}(s) = A^{synapse}\left(\exp\left(-\frac{s}{\tau_{decay}^{synapse}}\right) - \exp\left(-\frac{s}{\tau_{rise}^{synapse}}\right)\right) \quad (3)$$

where $\tau_{decay/rise}^{synapse}$ are time constants representing the rise and fall of an individual synaptic PSP; A is the PSP's amplitude; $\varepsilon_{ij}^{synapse}$ represents the type of activity of the synapse between neuron j and neuron i that can be measured and modelled separately for a fast excitation, fast inhibition, slow excitation, and slow inhibition (it is affected by different genes/proteins). External inputs can also be added to model background noise, background oscillations or environmental information.

An important part of the model is a dynamic gene/protein regulatory network (GRN) model of the dynamic interactions between genes/proteins over time that affect the spiking activity of the neuron. Although biologically plausible, a GRN model is only a highly simplified general model that does not necessarily take into account the exact chemical and molecular interactions. A GRN model is defined by:

(a) a set of genes/proteins, $G= (g_1,g_2,\ldots, g_k)$;
(b) an initial state of the level of expression of the genes/proteins $G(t=0)$;
(c) an initial state of a connection matrix $L = (L_{11},\ldots, L_{kk})$, where each element L_{ij} defines the known level of interaction (if any) between genes/proteins g_j and g_i;
(d) activation functions f_i for each gene/protein g_i from G. This function defines the gene/protein expression value at time (t+1) depending on the current values G (t), L (t) and some external information E(t):

$$g_i(t+1)= f_i (G(t), L(t), E(t)) \quad (4)$$

3. Learning and Memory in a Spiking Neuron

3.1 General classification

A learning process has an effect on the synaptic efficacy of the synapses connected to a spiking neuron and on the information that is memorized. Memory can be:

- Short-term, represented as a changing PSP and temporarily changing synaptic efficacy;
- Long-term, represented as a stable establishment of the synaptic efficacy;
- Genetic (evolutionary), represented as a change in the genetic code and the gene/ protein expression level as a result of the above short-term and long term memory changes and evolutionary processes.

Learning in SNN can be:

- Unsupervised - there is no desired output signal provided;
- Supervised – a desired output signal is provided;
- Semi-supervised.

Different tasks can be learned by a neuron, e.g:

- Classification;
- Input-output spike pattern association.

Several biologically plausible learning rules have been introduced so far, depending on the type of the information presentation: (3)

- Rate-order learning, that is based on the average spiking activity of a neuron over time [18, 34, 43];
- Temporal learning, that is based on precise spike times [44, 104, 106, 13, 42];
- Rank-order learning, that takes into account the order of spikes across all synapses connected to a neuron [105, 106].

Rate-order information representation is typical for cognitive information processing [18].

Table 1. Neuronal action potential parameters and related proteins and ion channels in the computational neuro-genetic model of a spiking neuron: AMPAR - (amino- methylisoxazole- propionic acid) AMPA receptor; NMDR - (N-methyl-D-aspartate acid) NMDA receptor; GABA$_A$R - (gamma-aminobutyric acid) GABA$_A$ receptor, GABA$_B$R - GABA$_B$ receptor; SCN - sodium voltage-gated channel, KCN - kalium (potassium) voltage-gated channel; CLC - chloride channel (from Benuskova and Kasabov, 2007)

Different types of action potential of a spiking neuron used as parameters for its computational model	Related neurotransmitters and ion channels
Fast excitation PSP	AMPAR
Slow excitation PSP	NMDAR
Fast inhibition PSP	GABA$_A$R
Slow inhibition PSP	GABA$_B$R
Modulation of PSP	mGluR
Firing threshold	Ion channels SCN, KCN, CLC

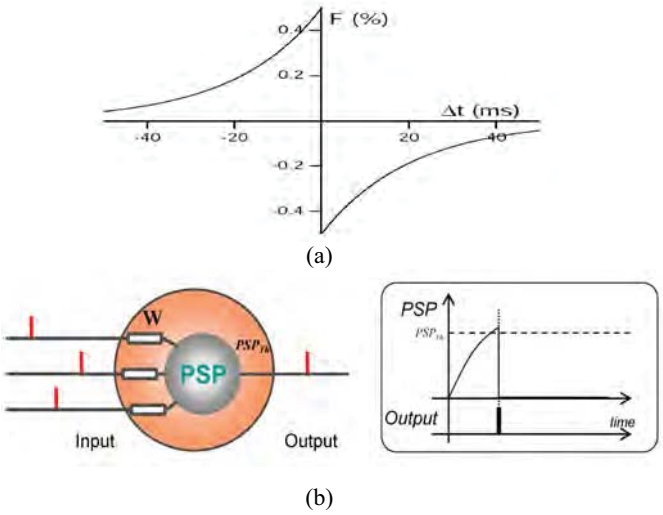


Fig.5. (a) An example of synaptic change in a STDP learning neuron [103]; (b) Rank-order learning neuron. .

Temporal spike learning is observed in the auditory [93], the visual [11] and the motor control information processing of the brain [13, 90]. Its use in neuro-prosthetics is essential, along with applications for a fast, real-time recognition and control of sequence of related processes [14].

Temporal coding accounts for the precise time of spikes and has been utilised in several learning rules, most popular being Spike-Time Dependent Plasticity (STDP) [103, 69] and SDSP [30, 14]. Temporal coding of information in SNN makes use of the exact time of spikes (e.g. in milliseconds). Every spike matters and its time matters too.

3.2 The STDP learning rule

The STDP learning rule uses Hebbian plasticity [39] in the form of long-term potentiation (LTP) and depression (LTD) [103, 69]. Efficacy of synapses is strengthened or weakened based on the timing of post-synaptic action potential in relation to the pre-synaptic spike (example is given in Fig.5(a)). If the difference in the spike time between the pre-synaptic and post-synaptic neurons is negative (pre-synaptic neuron spikes first) than the connection weight between the two neurons increases, otherwise it decreases. Through STDP, connected neurons learn consecutive temporal associations from data. Pre-synaptic activity that precedes post-synaptic firing can induce long-term potentiation (LTP), reversing this temporal order causes long-term depression (LTD).

3.3 Spike Driven Synaptic Plasticity (SDSP)

The SDSP is an unsupervised learning method [30, 14], a modification of the STDP, that directs the change of the synaptic plasticity V_{w0} of a synapse w_0 depending on the time of spiking of the pre-synaptic neuron and the post-synaptic neuron. V_{w0} increases or decreases, depending on the relative timing of the pre- and post-synaptic spikes.

If a pre-synaptic spike arrives at the synaptic terminal before a postsynaptic spike within a critical time window, the synaptic efficacy is increased (potentiation). If the post-synaptic spike is emitted just before the pre-synaptic spike, synaptic efficacy is decreased (depression). This change in synaptic efficacy can be expressed as:

$$\Delta V_{w0} = \frac{I_{pot}(t_{post})}{C_p}\Delta t_{spk} \quad \text{if } t_{pre} < t_{post} \tag{5}$$

$$\Delta V_{w0} = -\frac{I_{dep}(t_{post})}{C_d}\Delta t_{spk} \quad \text{if } t_{post} < t_{pre} \tag{6}$$

where Δt_{spk} is the pre- and post-synaptic spike time window.

The SDSP rule can be used to implement a supervised learning algorithm, when a teacher signal, that copies the desired output spiking sequence, is entered along with the training spike pattern, but without any change of the weights of the teacher input.

The SDSP model is implemented as an VLSI analogue chip [49]. The silicon synapses comprise bistability circuits for driving a synaptic weight to one of two possible analogue values (either potentiated or depressed). These circuits drive the synaptic-weight voltage with a current that is superimposed on that generated by the STDP and which can be either positive or negative. If, on short time scales, the synaptic weight is increased above a set threshold by the network activity via the STDP learning mechanism, the bi-stability circuits generate a constant weak positive current. In the absence of activity (and hence learning) this current will drive the weight toward its potentiated state. If the STDP decreases the synaptic weight below the threshold, the bi-stability circuits will generate a negative current that, in the absence of spiking activity, will actively drive the weight toward the analogue value, encoding its depressed state. The STDP and bi-stability circuits facilitate the implementation of both long-term and short term memory.

3.4 Rank-order learning

The rank-order learning rule uses important information from the input spike trains – the rank of the first incoming

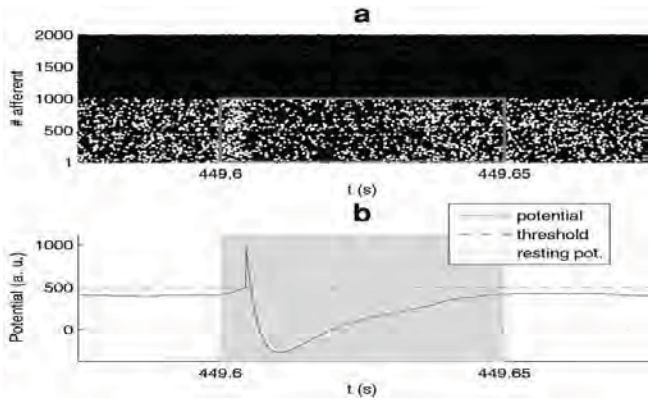


Fig.6. A single LIF neuron with simple synapses can be trained with the STDP unsupervised learning rule to discriminate a repeating pattern of synchronised spikes on certain synapses from noise (from : T. Masquelier, R. Guyonneau and S. Thorpe, PlosONE, Jan2008))

spike on each synapse (Fig.5(b)). It establishes a priority of inputs (synapses) based on the order of the spike arrival on these synapses for a particular pattern, which is a phenomenon observed in biological systems as well as an important information processing concept for some STPR problems, such as computer vision and control [105, 106]. This learning makes use of the extra information of spike (event) order. It has several advantages when used in SNN, mainly: fast learning (as it uses the extra information of the order of the incoming spikes) and asynchronous data entry (synaptic inputs are accumulated into the neuronal membrane potential in an asynchronous way). The learning is most appropriate for AER input data streams [23] as the events and their addresses are entered into the SNN 'one by one', in the order of their happening.

The postsynaptic potential of a neuron i at a time t is calculated as:

$$PSP(i,t) = \sum_j \mathrm{mod}^{order(j)} w_{j,i} \qquad (7)$$

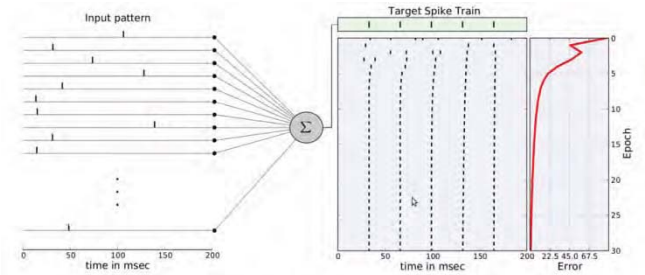
where *mod* is a modulation factor; j is the index for the incoming spike at synapse j,i and $w_{j,i}$ is the corresponding synaptic weight; $order(j)$ represents the order (the rank) of the spike at the synapse j,i among all spikes arriving from all m synapses to the neuron i. The $order(j)$ has a value 0 for the first spike and increases according to the input spike order. An output spike is generated by neuron i if the PSP (i,t) becomes higher than a threshold PSPTh (i).

During the training process, for each training input pattern (sample, example) the connection weights are calculated based on the order of the incoming spikes [105]:

$$\Delta w_{j,i}(t) = mod^{\,\mathrm{order}\,(j,i\,(t))} \qquad (8)$$

3.5 Combined rank-order and temporal learning

In [25] a method for a combined rank-order and temporal (e.g. SDSP) learning is proposed and tested on benchmark data. The initial value of a synaptic weight is set according to the rank-order learning based on the first incoming spike on this synapse. The weight is further modified to



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

(a)

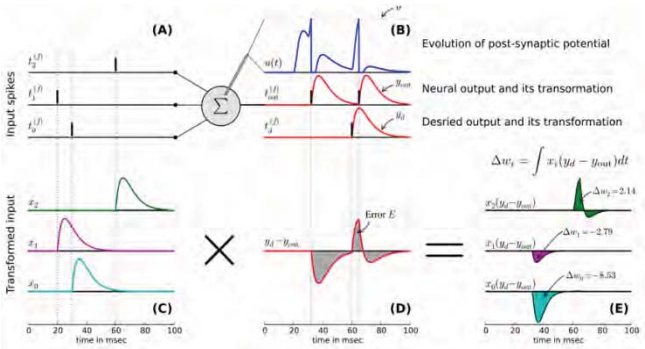
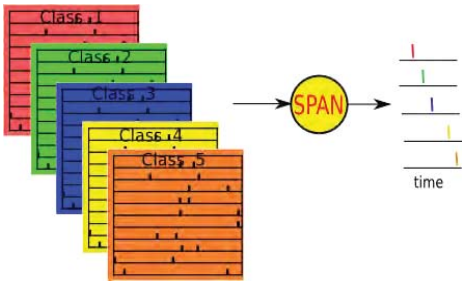
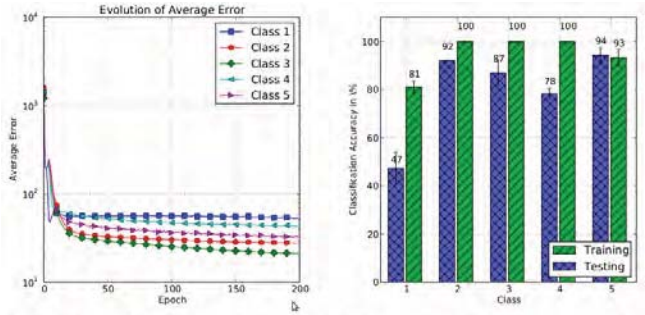


Illustration of the proposed training algorithm.

(b)



(c)



Evolution of the average erros obtained in 30 independent trails for each class of the training samples, and the average accuracies obtained in the training and testing phase.

(d)

Fig.7 (a) The SPAN model [77]. (b) The Widrow-Hoff Delta learning rule applied to learn to associate an output spike sequence to an input STP [77, 30]. (c) The use of a single SPAN neuron for the classification of 5 STP belonging to 5 different classes [77]. (d) The accuracy of classification is rightly lower for the class 1 – spike at the very beginning of the input pattern as there is no sufficient input data).

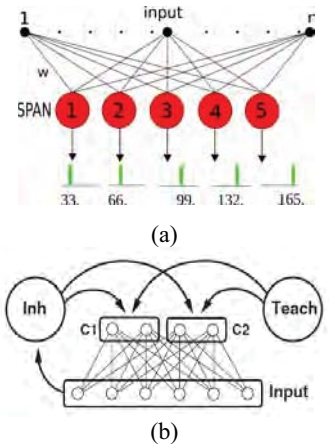


Fig.8: (a) Multiple SPAN neurons [76]. (b) Multiple SDSP trained neurons [14]

accommodate following spikes on this synapse with the use of a temporal learning rule – SDSP.

4. STPR in a Single Neuron

In contrast to the distributed representation theory and to the widely popular view that a single neuron cannot do much, some recent results showed that a single neuronal model can be used for complex STPR.

A single LIF neuron, for example, with simple synapses can be trained with the STDP unsupervised learning rule to discriminate a repeating pattern of synchronised spikes on certain synapses from noise (from: T. Masquelier, R. Guyonneau and S. Thorpe, PlosONE, Jan2008) – see Fig. 6.

Single neuron models have been introduced for STPR, such as: Tempotron [38]; Chronotron [28]; ReSuMe [87]; SPAN [76, 77]. Each of them can learn to emit a spike or a spike pattern (spike sequence) when a certain STP is recognised. Some of them can be used to recognise multiple STP per class and multiple classes [87, 77, 76].

'Fig.7(a)-(d) show a SPAN neuron and its use for classification of 5 STP belonging to 5 different classes [77]. The accuracy of classification is rightly lower for the class 1 (the neuron emits a spike at the very beginning of the input pattern) as there is no sufficient input data – Fig.7(d).) [77].

5. Evolving Spiking Neural Networks

Despite the ability of a single neuron to conduct STPR, a single neuron has a limited power and complex STPR tasks will require multiple spiking neurons.

One approach is proposed in the evolving spiking neural networks (eSNN) framework [61, 111]. eSNN evolve their structure and functionality in an on-line manner, from incoming information. For every new input pattern, a new neuron is dynamically allocated and connected to the input neurons (feature neurons). The neurons connections are established for the neuron to recognise this pattern (or a similar one) as a positive example. The neurons represent centres of clusters in the space of the synaptic weights. In some implementations similar neurons are merged [61, 115]. That makes it possible to achieve a very fast learning in an eSNN (only one pass may be necessary), both in a supervised and in an unsupervised mode.

In [76] multiple SPAN neurons are evolved to achieve a better accuracy of spike pattern generation than a single SPAN – Fig.8(a).

In [14] the SDSP model from [30] has been successfully used to train and test a SNN for 293 character recognition (classes). Each character (a static image) is represented as 2000 bit feature vector, and each bit is transferred into spike rates, with 50Hz spike burst to represent 1 and 0 Hz to represent 0. For each class, 20 different training patterns are used and 20 neurons are allocated, one for each pattern (altogether 5 860) (Fig.8(b)) and trained for several hundreds of iterations.

A general framework of eSNN for STPR is shown in Fig.9. It consists of the following blocks:

- Input data encoding block;
- Machine learning block (consisting of several sub-blocks);
- Output block.

In the input block continuous value input variables are transformed into spikes. Different approaches can be used:

- population rank coding [13] – Fig.10(a);
- thresholding the input value, so that a spike is generated if the input value (e.g. pixel intensity) is above a threshold;
- Address Event Representation (AER) - thresholding the difference between two consecutive values of the

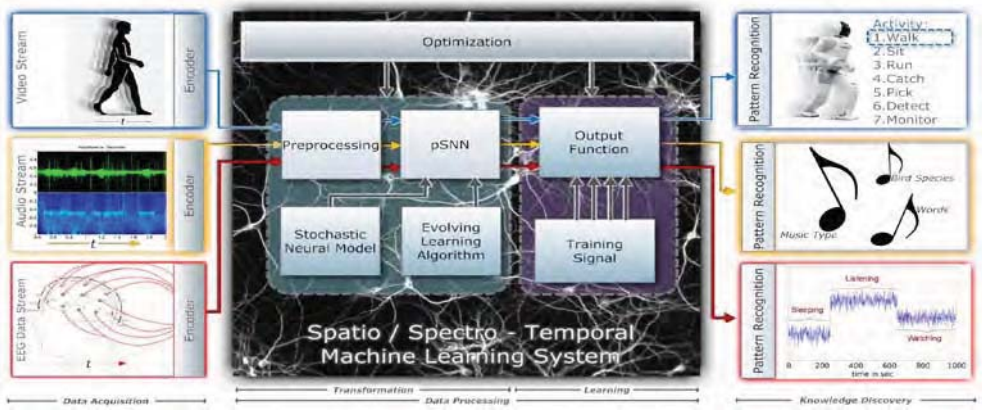


Fig.9. The eSNN framework for STPR (from: http://ncs.ethz.ch/projects/evospike)

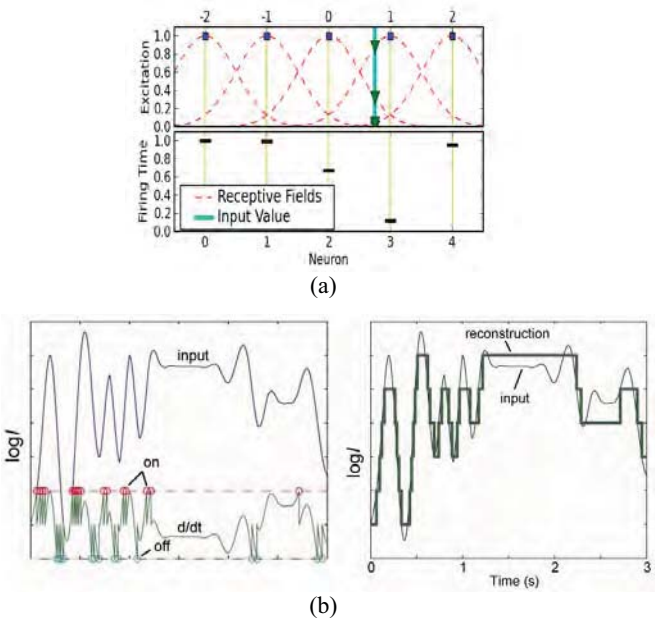


Fig.10. (a) Population rank order coding of input information; (b) Address Event Representations (AER) of the input information [23].

same variable over time as it is in the artificial cochlea [107] and artificial retina devices [23] – Fig.10(b).

The input information is entered either on-line (for on-line, real time applications) or as a batch data. The *time* of the input data is in principal different from the internal SNN *time* of information processing.

Long and complex SSTD cannot be learned in simple one-layer neuronal structures as the examples in Fig.8(a) and (b). They require neuronal 'buffers' as shown in Fig.11(a). In [82] a 3D buffer was used to store spatio-temporal 'chunks' of input data before the data is classified. In this case the size of the chunk (both in space and time) is fixed by the size of the reservoir. There are no connections between the layers in the buffer. Still, the system outperforms traditional classification techniques as it is demonstrated on sign language recognition, where eSNN classifier was applied [61, 115].

Reservoir computing [73, 108] has already become a popular approach for SSTD modelling and pattern recognition. In the classical view a 'reservoir' is a homogeneous, passive 3D structure of probabilistically connected and fixed neurons that in principle has no learning and memory, neither it has an interpretable structure – fig.11b. A reservoir, such as a Liquid State Machine (LSM) [73, 37], usually uses *small world recurrent connections* that do not facilitate capturing explicit spatial and temporal components from the SSTD in their relationship, which is the main goal of learning SSTD. Despite difficulties with the LSM reservoirs, it was shown on several SSTD problems that they produce better results than using a simple classifier [95, 73, 99, 60]. Some publications demonstrated that probabilistic neurons are suitable for reservoir computing especially in a noisy environment [98, 83].

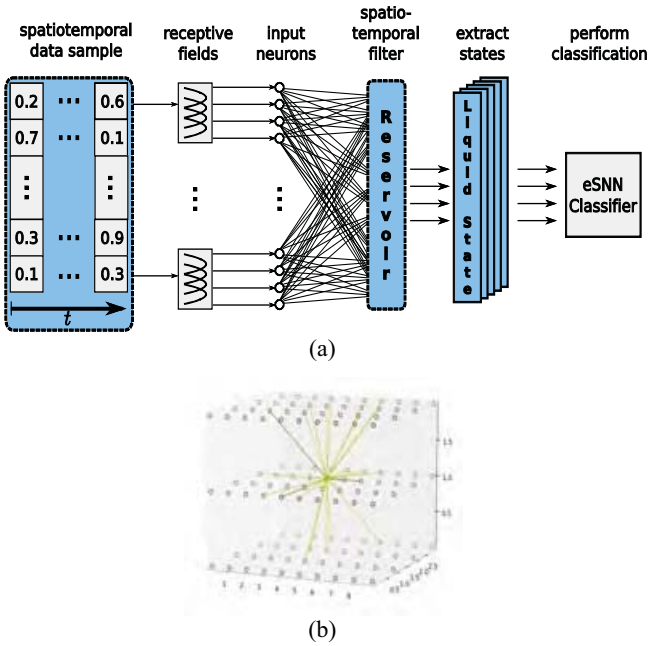


Fig.11. (a) An eSNN architecture for STPR using a reservoir; (b) The structure and connectivity of a reservoir

In [81] an improved accuracy of LSM reservoir structure on pattern classification of hypothetical tasks is achieved when STDP learning was introduced into the reservoir. The learning is based on comparing the liquid states for different classes and adjusting the connection weights so that same class inputs have closer connection weights. The method is illustrated on the phone recognition task of the TIMIT data base phonemes – spectro-temporal problem. 13 MSCC are turned into trains of spikes. The metric of separation between liquid states representing different classes is similar to the Fisher's *t*-test [27].

After a presentation of input data example (or a 'chink' of data) the state of the SNN reservoir $S(t)$ is evaluated in an output module and used for classification purposes (both during training and recall phase). Different methods can be applied to capture this state:

- Spike rate activity of *all* neurons at a certain time window: The state of the reservoir is represented as a vector of n elements (n is the number of neurons in the reservoir), each element representing the spiking probability of the neuron within a time window. Consecutive vectors are passed to train/recall an output classifier.
- Spike rate activity of spatio-temporal clusters C_1, C_2, … C_k of close (both in space and time) neurons: The state $S_{Ci}(t)$ of each cluster C_i is represented by a single number, reflecting on the spiking activity of the neurons in the cluster in a defined time window (this is the internal SNN time, usually measured in '*msec*'). This is interpreted as the current spiking probability of the cluster. The states of all clusters define the current reservoir state $S(t)$. In the output function, the cluster states $S_{Ci}(t)$ are used differently for different tasks.
- Continuous function representation of spike trains: In contrast to the above two methods that use spike rates to evaluate the spiking activity of a neuron or a neuronal

cluster, here the train of spikes from each neuron within a time window, or a neuronal cluster, is transferred into a continuous value temporal function using a kernel (e.g. α-kernel). These functions can be compared and a continuous value error measured.

In [95] a comparative analysis of the three methods above is presented on a case study of Brazilian sign language gesture recognition (see Fig.18) using a LSM as a reservoir.

Different adaptive classifiers can be explored for the classification of the reservoir state into one of the output classes, including: statistical techniques, e.g. regression techniques; MLP; eSNN; nearest-neighbour techniques; incremental LDA [85]. State vector transformation, before classification can be done with the use of adaptive incremental transformation functions, such as incremental PCA [84].

6. Computational Neurogenetic Models (CNGM)

Here, the neurogenetic model of a neuron [63, 10] is utilized. A CNGM framework is shown in Fig.12 [64].

The CNGM framework comprises a set of methods and algorithms that support the development of computational models, each of them characterized by:

- Two-tire, consisting of an eSNN at the higher level and a gene regulatory network (GRN) at the lower level, each functioning at a different time-scale and continuously interacting between each other;
- Optional use of probabilistic spiking neurons, thus forming an epSNN;
- Parameters in the eSNN model are defined by genes/proteins from the GRN;
- Can capture in its internal representation both spatial and temporal characteristics from SSTD streams;
- The structure and the functionality of the model evolve in time from incoming data;
- Both unsupervised and supervised learning algorithms can be applied in an on-line or in a batch mode.
- A concrete model would have a specific structure and a set of algorithms depending on the problem and the application conditions, e.g.: classification of SSTD; modelling of brain data.

The framework from Fig.12 supports the creation of a multi-modular integrated system, where different modules, consisting of different neuronal types and genetic parameters, represent different functions (e.g.: vision; sensory information processing; sound recognition; motor-control) and the whole system works in an integrated mode.

The neurogenetic model from Fig.12 uses as a main principle the analogy with biological facts about the relationship between spiking activity and gene/protein dynamics in order to control the learning and spiking parameters in a SNN when SSTD is learned. Biological support of this can be found in numerous publications (e.g. [10, 40, 117, 118]).

The Allen Human Brain Atlas (www.brain-map.org) of the Allen Institute for Brain Science (www.alleninstitute.org) has shown that at least 82% of the human genes are expressed in the brain. For 1000 anatomical sites of the brains of two individuals 100 mln data points are collected that indicate gene expressions of each of the genes and underlies the biochemistry of the sites.

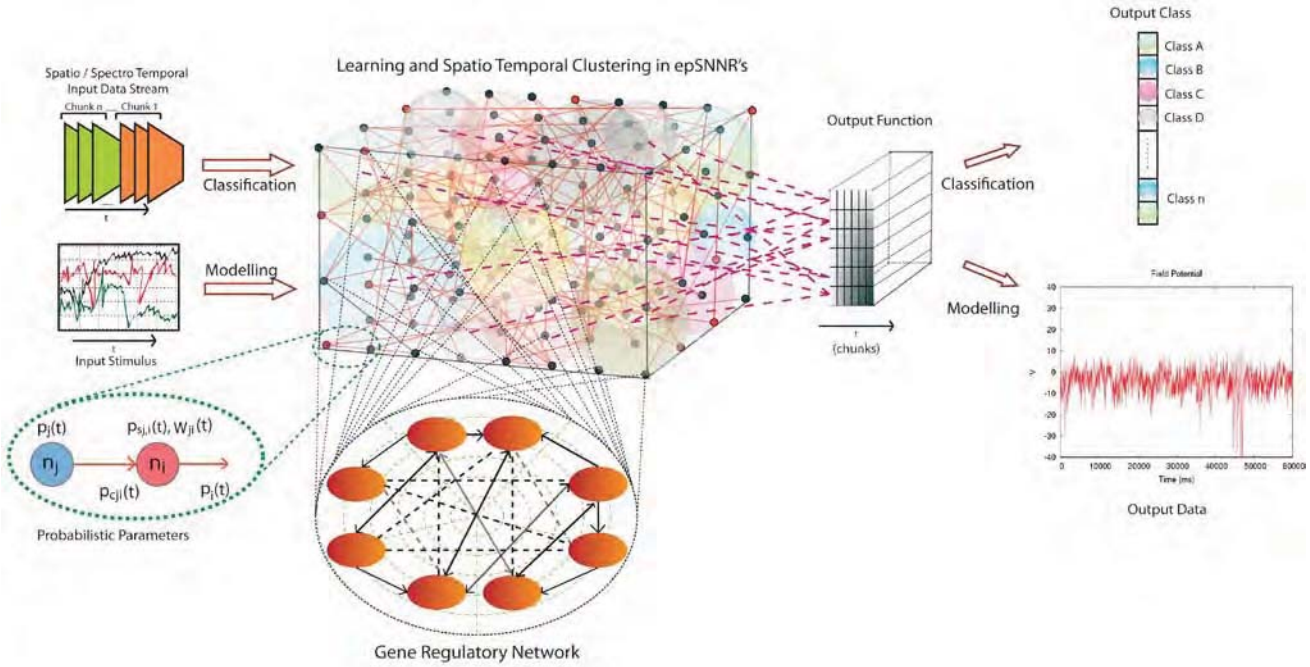


Fig.12. A schematic diagram of a CNGM framework, consisting of: input encoding module; a SNN reservoir output function for SNN state evaluation; output classifier; GRN (optional module). The framework can be used to create concrete models for STPR or data modelling (from [64]).

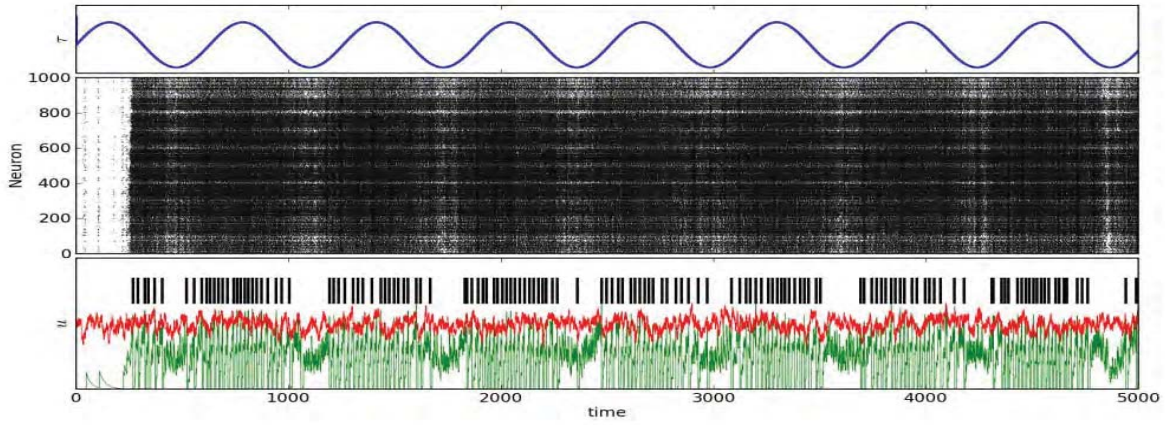


Fig.13. A GRN interacting with a SNN reservoir of 1000 neurons. The GRN controls a single parameter, i.e. the τ parameter of all 1000 LIF neurons, over a period of five seconds. The top diagram shows the evolution of τ. The response of the SNN is shown as a raster plot of spike activity. A black point in this diagram indicates a spike of a specific neuron at a specific time in the simulation. The bottom diagram presents the evolution of the membrane potential of a single neuron from the network (green curve) along with its firing threshold ϑ (red curve). Output spikes of the neuron are indicated as black vertical lines in the same diagram (from [65]).

In [18] it is suggested that both the firing rate (rate coding) and spike timing as spatiotemporal patterns (rank order and spatial pattern coding) play a role in fast and slow, dynamic and adaptive sensorimotor responses, controlled by the cerebellar nuclei. Spatio-temporal patterns of population of Purkinji cells are shaped by activities in the molecular layer of interneurons. In [40] it is demonstrated that the temporal spiking dynamics depend on the spatial structure of the neural system (e.g. different for the hippocampus and the cerebellum). In the hippocampus the connections are scale free, e.g. there are hub neurons, while in the cerebellum the connections are regular. The spatial structure depends on genetic pre-determination and on the gene dynamics. Functional connectivity develops in parallel with structural connectivity during brain maturation. A growth-elimination process (synapses are created and eliminated) dependong ene expression [40], e.g. glutamatergic neurons issued from the same progenitors tend to wire together and form ensembles, also for the cortical GABAergic interneuron population. Connections between early developed neurons (mature networks) are more stable and reliable when transferring spikes than the connections between newly created neurons (thus the probability of spike transfer). Postsynaptic AMPA-type glutamater eceptors (AMPARs) mediate most fast excitatory synaptic transmissions and are crucial for many aspects of brain function, including learning, memory and cognition [10, 31].

It was shown the dramatic effect of a change of single gene, that regulates the τ parameter of the neurons, on the spiking activity of the whole SNN of 1000 neurons – see Fig.13 [65].

The spiking activity of a neuron may affect as a feedback the expressions of genes [5]. As pointed in [118] on a longer time scales of minutes and hours the function of neurons may cause the changes of the expression of hundreds of genes transcribed into mRNAs and also in microRNAs, which makes the short-term, the long-term and the genetic memories of a neuron linked together in a global memory of the neuron and further - of the whole neural system.

A major problem with the CNGM from fig.12 is how to optimize the numerous parameters of the model. One solution could be using evolutionary computation, such as PSO [75, 83] and the recently proposed quantum inspired evolutionary computation techniques [22, 97, 96]. The latter can deal with a very large dimensional space as each quantum-bit chromosome represents the whole space, each point to certain probability. Such algorithms are faster and lead to a close solution to the global optimum in a very short time.

In one approach it may be reasonable to use same parameter values (same GRN) for all neurons in the SNN or for each of different types of neurons (cells) that will results in a significant reduction of the parameters to be optimized. This can be interpreted as 'average' parameter value for the neurons of the same type. This approach corresponds to the biological notion to use one value (average) of a gene/protein expression form illions of cells in bioinformatics.

Another approach to define the parameters of the probabilistic spiking neurons, especially when used in biological studies, is to use prior knowledge about the association of spiking parameters with relevant genes/proteins (neuro-transmitter, neuro-receptor, ion channel, neuro-modulator) as described in [64]. Combination of the two approaches above is also possible.

7. SNN Software and hardware implementations to support STPR

Software and hardware realisations of SNN are already available to support various applications of SNN for STPR. Among the most popular software/hardware systems are [24, 16, 29]:

- jAER (http://jaer.wiki.sourceforge.net) [23];

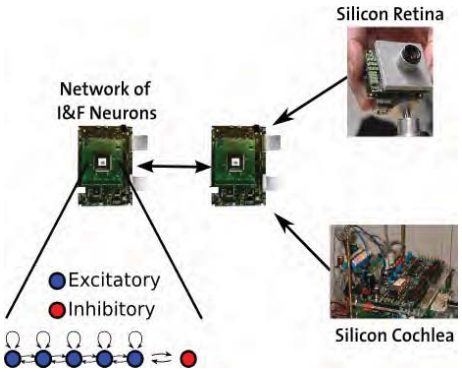


Fig.14. A hypothetical neuromorphic SNN application system (from http://ncs.ethz.ch)

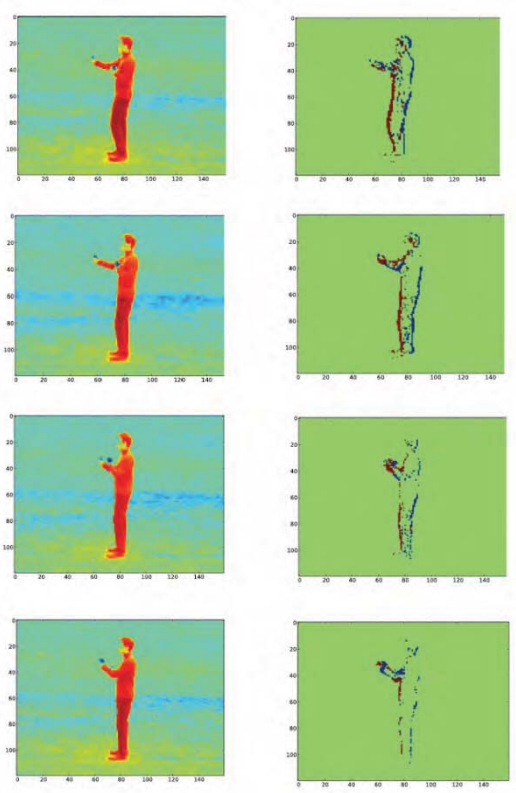
- Software simulators, such as Brian [16], Nestor, NeMo [79],etc;
- Silicon retina camera [23];
- Silicon cochlea [107];
- SNN hardware realisation of LIFM and S DSP [47-50];
- The SpiNNaker hardware/software environment [89, 116];
- FPGA implementations of SNN [56];
- The IBM LIF SNN chip, recently announced.

Fig.14 shows a hypothetical engineering system using some of the above tools (from [47, 25]).

8. Current and Future Applications of eSNN and CNGM for STPR

Numerous are the applications of eSNN for STPR. Here only few of them are listed:

- Moving object recognition (fig. 15) [23, 60];
- EEG data modelling and pattern recognition [70, 1, 51, 21, 26, 99, 35, 36] directed to practical applications, such as: BCI [51], classification of epilepsy [35, 36, 109] - (fig.16);
- Robot control through EEG signals [86] (fig.17) and robot navigation [80];
- Sign language gesture recognition (e.g. the Brazilian sign language – fig.18) [95];
- Risk of event evaluation, e.g. prognosis of establishment of invasive species [97] – fig.19; stroke occurrence [6], etc.
- Cognitive and emotional robotics [8, 64];
- Neuro-rehabilitation robots [110];
- Modelling finite automata [17, 78];
- Knowledge discovery from SSTD [101];
- Neuro-genetic robotics [74];
- Modelling the progression or the response to treatment of neurodegenerative diseases, such as Alzheimer's Disease [94, 64] – fig.20. The analysis of the obtained GRN model in this case could enable the discovery of unknown interactions between genes/proteins related to a brain disease progression and how these interactions can be modified to achieve a desirable effect.



(a) (b)

Fig.15.Moving object recognition with the use of AER [23]. (a) Disparity map of a video sample; (b) Address event representation (AER) of the above video sample.

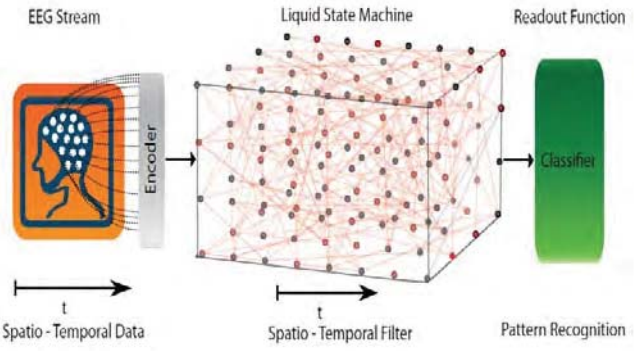


Fig.16. EEG based BCI.

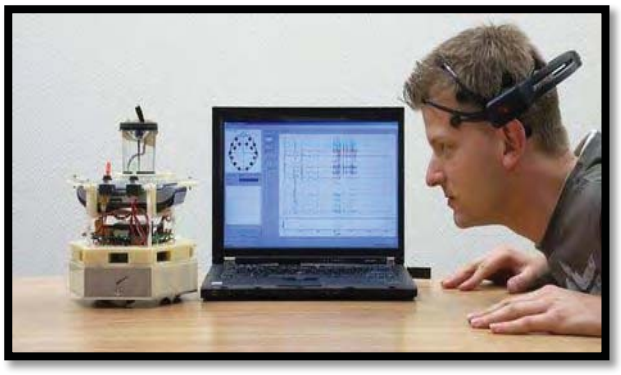


Fig.17. Robot control and navigation

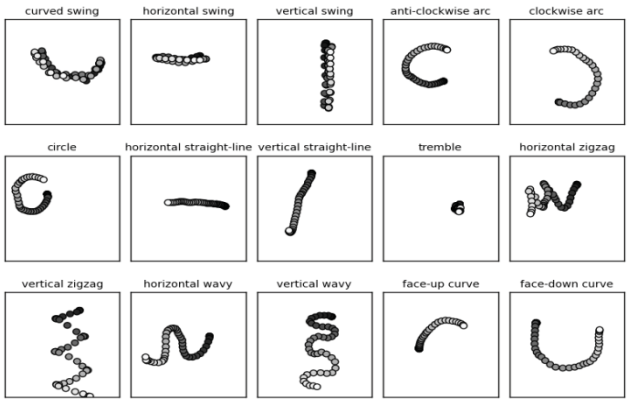


Fig.18. A single sample for each of the 15 classes of the LIngua BRAsileira de Sinais (LIBRAS) - the official Brazilian sign language is shown. The colour indicates the time frame of a given data point (black/white corresponds to earlier/later time points) [95].

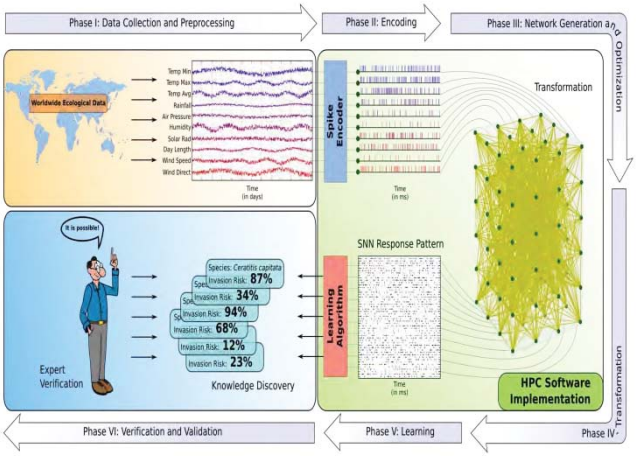


Fig 19. Prognosis of the establishment of invasive species [97]

- Modelling financial and economic problems (neuro-economics) where at a 'lower' level the GRN represents the dynamic interaction between time series variables (e.g. stock index values, exchange rates, unemployment, GDP, prize of oil), while the 'higher' level pSNN states represents the state of the economy or the system under study. The states can be further classified into pre-define classes (e.g. buy, hold, sell, invest, likely bankruptcy) [113];
- Personalized modelling, which is concerned with the creation of a single model for an individual input data [58, 59, 62]. Here as an individual data a whole SSTD pattern is taken rather than a single vector.

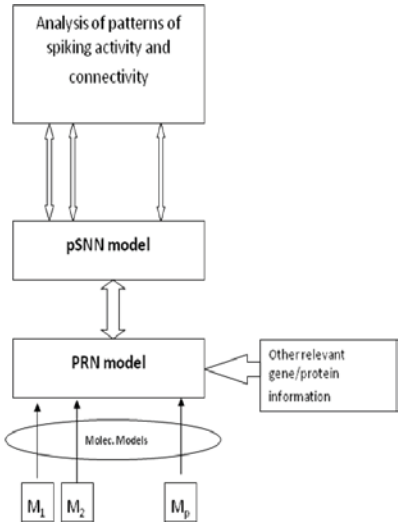


Fig.20.Hierarchical CNGM [64]

Acknowledgement

I acknowledge the discussions with G. Indivery and also with: A.Mohemmed, T.Delbruck, S-C.Liu, N.Nuntalid, K.Dhoble, S.Schliebs, R.Hu, R.Schliebs, H.Kojima, F.Stefanini. The work on this paper is sponsored by the Knowledge Engineering and Discovery Research Institute, KEDRI (www.kedri.info) and the EU FP7 Marie Curie International Incoming Fellowship project PIIF-GA-2010-272006 *EvoSpike,* hosted by the Institute for Neuroinformatics – the Neuromorphic Cognitive Systems Group, at the University of Zurich and ETH Zurich (http://ncs.ethz.ch/projects/evospike). Diana Kassabova helped with the proofreading.

References

1. Acharya, R., Chua, E.C.P., Chua, K.C., Min, L.C., and Tamura, T. (2010), "Analysis and Automatic Identification of Sleep Stages using Higher Order Spectra," *Int.Journal of Neural Systems*, 20:6, pp. 509-521.
2. Arel, I., D. C. Rose, T. P. Karnowski, Deep Machine Learning: A New Frontier in Artificial Intelligence Research, Computational Intelligence Magazine, IEEE, vol.5, no.4, pp.13-18, 2010.
3. Arel, I., D.Rose, and B.Coop (2008) DeSTIN: A deep learning architecture with application to high-dimensional robust pattern recognition, in: Proc. 2008 AAAI Workshop Biologically Inspired Cognitive Architectures (BICA)
4. Arel, I., D.Rose, T.Karnovski (2010) Deep Machine Learning – A New Frontier in Artificial Intelligence Research, IEEE CI Magazine, Nov.2010, 13-18
5. Barbado, M., Fablet, K., Ronjat, M. And De Waard, M. (2009) Gene regulation by voltage-dependent calcium channels, Biochimica et Biophysica Acta, 1793, 1096-1104.
6. Barker-Collo, S., Feigin, V. L., Parag, V., Lawes, C. M. M., & Senior, H. (2010). Auckland Stroke Outcomes Study. Neurology, 75(18), 1608-1616.
7. Belatreche, A., Maguire, L. P., and McGinnity, M. Advances in Design and Application of Spiking Neural Networks. Soft Comput. 11, 3, 239-248, 2006
8. Bellas, F., R. J. Duro, A. Faiña, D. Souto, M DB: Artificial Evolution in a Cognitive Architecture for Real Robots, IEEE Transactions on Autonomous Mental Development, vol. 2, pp. 340-354, 2010
9. Bengio, Y. (2009) Learning Deep Architectures for AI, Found. Trends. Mach. Learning, vol.2, No.1, 1-127.
10. Benuskova, L., and N.Kasabov, Computational neuro-genetic modelling, Springer, New York, 2007, 290 pages

11. Berry, M.J., D. K. Warland, and M. Meister. The structure and precision of r etinal s piketrains. P NAS, 94(10): 5411–5416, May 1997.
12. Bohte, S., J. K ok, J. LaPoutre, Applications of spiking ne ural networks, Infor mation Processing L etters, vol. 95, no. 6, 519-520, 2005.
13. Bohte, S.M., The evidence for neural information proc essing with pre cise s pike-times: A sur vey. NAT URAL COMPUTING, 3:2004, 2004.
14. Brader, J., W. S enn, a nd S. F usi, Learning re al-world s timuli in a n eural n etwork w ith s pike-driven s ynaptic d ynamics, Neural computation, vol. 19, no. 11, pp. 2881–2912, 2007.
15. Brader, J.M., Walter Senn, Stefano Fusi, Learning Real-World Stimuli i n a N eural N etwork with S pike-Driven S ynaptic Dynamics, Neural Computation 2007 19(11): 2881-2912, 2007.
16. Brette R., R udolph M., C arnevale T., H ines M., B eeman D., Bower J . M ., Diesmann M ., Morrison A ., G oodman P . H ., Harris F . C ., Z irpe M ., N atschläger T., P ecevski D ., Ermentrout B., Djurfeldt M., Lansner A., Ro chel O., V ieville T., M uller E., Davison A. P ., Boustani S . E ., D estexhe A . (2007). Simulation of networks of spiking neurons: a review of tools and strategies. J. Comput. Neurosci. 23, 349–398.
17. Buonomano, D ., W . M aass, S tate-dependent c omputations: Spatio-temporal proc essing in c ortical n etworks, N ature Reviews, Neuroscience, vol. 10, Feb. 2009, 113-125.
18. Chris I. De Zeeuw, Freek E. Hoebeek, Laurens W. J. Bosman, Martijn S chonewille, S patiotemporal fi ring p atterns i n the cerebellum, Nature R eviews N euroscience 12, 327-344 (June 2011) | doi:10.1038/nrn3011
19. Chris R. S hortall, A lison M oore, E mma S mith, Mike J . H all, Ian P. Woiwod, and Richard Harrington. Long-term changes in the a bundance of fl ying i nsects. Ins ect Cons ervation a nd Diversity, 2(4):251–260, 2009
20. Cowburn, P. J., Cleland, J. G. F., C oats, A. J. S., & K omajda, M. (1996). Ri sk s tratification in c hronic he art fa ilure. E ur Heart J, 19, 696-710.
21. Craig, D .A., H .T. N guyen, A daptive E EG Thought P attern Classifier for A dvanced W heelchair Cont rol, Engineering i n Medicine and Biology Society- EMBS'07, pp.2544-2547, 2007.
22. Defoin-Platel, M ., S .Schliebs, N .Kasabov, Quantum-inspired Evolutionary Algorithm: A multi-model E DA, IEEE Tr ans. Evolutionary Computation, vol.13, No.6, Dec.2009, 1218-1232
23. Delbruck, T ., j AER op en s ource pr oject, 2007 , http://jaer.wiki.sourceforge.net.
24. Douglas, R. and M ahowald, M . (1995) S ilicon Neurons , in: The Handbook of Brain Theory and Neural Networks, pp. 282-289, M. Arbib (Ed.), MIT Press.
25. Dhoble, K ., N .Nuntalid, G .Indivery and N .Kasabov, O nline Spatio-Temporal P attern R ecognition w ith Evolving S piking Neural N etworks ut ilising Address Event Re presentation, Rank Order, and Temporal Spike Learning, Proc. IJCNN 2012, Brisbane, June 2012, IEEE Press
26. Ferreira A ., C . A lmeida, P . G eorgieva, A . T omé, F . S ilva (2010): A dvances i n E EG-based Bi ometry, Int ernational Conference on Image Analysis and Recognition (ICIAR), June 21-27, 2010 , Povoa de V arzim, P ortugal, t o a ppear i n Springer LNCS series.
27. Fisher, R.A., The use of multiple measurements in taxonomic problems, Annals of Eugenics, 7 (1936) 179-188)
28. Florian, R .V., The c hronotron: a n euron t hat learns t o f ire temporally-precise spike patterns.
29. Furber, S., Temple, S., Neural systems engineering, Interface, J.of the Royal Society, vol. 4, 193-206, 2007
30. Fusi, S., M. Annunziato, D. Badoni, A. Salamon, and D. Amit, Spike-driven s ynaptic pl asticity: the ory, s imulation, V LSI implementation, N eural Com putation, vo l. 12 , no. 10, pp. 2227–2258, 2000.
31. Gene and Disease (2005), NCBI, http:/www.ncbi.nlm.nih.gov
32. Gerstner, W . (1 995) T ime s tructure of t he activity of ne ural network models, Phys. Rev 51: 738-758.
33. Gerstner, W . (2 001) W hat's di fferent w ith s piking n eurons? Plausible N eural N etworks fo r Bi ological M odelling, i n: H.Mastebroek and H. V os (Eds.), K luwer A cademic Publishers, pp. 23- 48.
34. Gerstner, W., A. K. Kreiter, H. Markram, and A. V. M. Herz. Neural c odes: f iring r ates a nd b eyond. P roc. N atl. A cad. S ci. USA, 94(24):12740–12741, 1997.
35. Ghosh-Dastidar S. and A deli, H. (2009), "A Ne w S upervised Learning A lgorithm for M ultiple S piking N eural N etworks with A pplication i n Epilepsy and S eizure D etection," *Neural Networks*, 22:10, 2009, pp. 1419-1431.
36. Ghosh-Dastidar, S . and A deli, H. (2007), Im proved S piking Neural N etworks for E EG Cl assification a nd Epilepsy and Seizure D etection, *Integrated Com puter-Aided E ngineering*, Vol. 14, No. 3, pp. 187-212
37. Goodman, E . a nd D . V entura. S patiotemporal pa ttern recognition v ia l iquid s tate m achines. In Neural Ne tworks, 2006. IJ CNN ' 06. Int ernational J oint Confe rence on , p ages 3848–3853, Vancouver, BC, 2006)
38. Gutig, R., and H. S ompolinsky. T he tempotron: a ne uron that learns s pike t iming-based de cisions. N at N eurosci, 9(3):420–428, Mar. 2006.
39. Hebb, D . (1949). The O rganization of Be havior. N ew Y ork, John Wiley and Sons.
40. Henley, J .M., E .A.Barker and O .O.Glebov, R outes, destinations and de lays: re cent advances i n A MPA receptor trafficking, Trends i n N euroscience, M ay 2011, vol.34, N o.5, 258-268
41. Hodgkin, A . L. a nd A . F . H uxley (1952) A qua ntitative description of membrane c urrent and i ts a pplication t o conduction and excitation in nerve. Journal of Physiology, 117: 500-544.
42. Hopfield, J ., P attern recognition c omputation us ing a ction potential timing for stimulus representation. Nature, 376:33–36, 1995.
43. Hopfield, J .J. (1982) N eural networks a nd ph ysical s ystems with emergent collective computational abilities. P NAS USA, vol.79, 2554-2558.
44. Hugo, G .E. a nd S . Ine s. Time and c ategory information i n pattern-based codes. Frontiers in Computational Neuroscience, 4(0), 2010.
45. Huguenard, J. R. (2000) Reliability of axonal propagation: The spike doesn't stop here, PNAS 97(17): 9349-50.
46. Iglesias, J . a nd Villa, A .E.P.(2008), Emergence of P referred Firing S equences i n Large S piking N eural N etworks D uring Simulated N euronal D evelopment, *Int. J ournal of Ne ural Systems*, 18(4), pp. 267-277.
47. Indiveri, G., B. Linares-Barranco, T. Hamilton, A. Van Schaik, R. E tienne-Cummings, T . D elbruck, S . Liu, P . D udek, P . H¨afliger, S . R enaud et a l., " Neuromorphic s ilicon ne uron circuits," Frontiers in Neuroscience, 5, 2011.
48. Indiveri, G ; Chicca, E ; Douglas, R J (2009). Artificial cognitive s ystems: F rom V LSI ne tworks of s piking n eurons to neuromorphic c ognition. Cognitive Com putation, 1(2):119-127.
49. Indiveri, G ; Stefanini, F ; Chicca, E (2010). Spike-based learning w ith a generalized integrate and fire s ilicon ne uron. In: 2010 I EEE Int. S ymp. Ci rcuits a nd S yst. (IS CAS 2010), Paris, 30 May 2010 - 02 June 2010, 1951-1954.
50. Indiviery G., a nd T .Horiuchi (2011) F rontiers i n Neuromorphic Engineering, Frontiers in Neuroscience, 5:118.

51. Isa, T., E . E . Fetz, K . M uller, R ecent advances i n br ain-machine interfaces, N eural N etworks, vol 22 , issue 9, Br ain-Machine Interface, pp 1201-1202, November 2009.
52. Izhikevich, E (2003) S imple m odel of spiking n eurons, I EEE Trans. on Neural Networks, 14, 6, 1569-1572.
53. Izhikevich, E. M. (2004) W hich m odel t o us e for c ortical spiking neurons? IEEE TNN, 15(5): 1063-1070.
54. Izhikevich, E.M., G.M.Edelman (2008) Large-Scale M odel of Mammalian Thalamocortical Systems, PNAS, 105: 3593-3598.
55. Izhikevich, E .(2006) P olychronization: Com putation w ith Spikes, Neural Computation, 18, 245-282.
56. Johnston, S .P., P rasad, G .,Maguire, L., M cGinnity, T .M. (2010), F PGA H ardware/software c o-design m ethodology - towards e volvable s piking ne tworks for robot ics a pplication, *Int. J. Neural Systems*, 20:6, 447-461.
57. Kasabov, N. F oundations of N eural N etworks, Fuzzy S ystems and K nowledge E ngineering. Ca mbridge, Massachussets, MIT Press (1996) 550p
58. Kasabov, N., and Y. Hu (2010) Integrated optimisation method for pe rsonalised m odelling a nd c ase s tudy applications, Int . Journal of F unctional Informatics a nd P ersonalised M edicine, vol.3,No.3,236-256.
59. Kasabov, N ., Data A nalysis and Predictive S ystems and Related Methodologies – Personalised Trait Modelling System, PCT/NZ2009/000222, NZ Patent.
60. Kasabov, N ., D hoble, K ., N untalid, N ., & M ohemmed, A . (2011). E volving proba bilistic spiking ne ural networks for spatio-temporal pattern r ecognition: A pr eliminary s tudy o n moving obj ect r ecognition. In 1 8th Int ernational Confe rence on N eural Infor mation P rocessing, ICO NIP 20 11, S hanghai, China, Springer LNCS .
61. Kasabov, N., Evolving connectionist systems: The knowledge engineering approach, Springer, 2007(2003).
62. Kasabov, N ., G lobal, local an d pe rsonalised m odelling and profile di scovery in Bi oinformatics: An integrated approach, Pattern Recognition Letters, Vol. 28, Issue 6 , April 2007, 673-685
63. Kasabov, N., L .Benuskova, S .Wysoski, A Com putational Neurogenetic Model of a Spiking Neuron, IJCNN 2005 Conf. Proc., IEEE Press, 2005, Vol. 1, 446-451
64. Kasabov, N ., R.Schliebs, H .Kojima (2011) P robabilistic Computational Neurogenetic F ramework: F rom M odelling Cognitive S ystems t o A lzheimer's D isease. IE EE T rans. Autonomous Mental Development, vol.3, No.4,2011, 1-12.
65. Kasabov, N., S.Schliebs and A.Mohemmed (2011) M odelling the E ffect of G enes on t he D ynamics of P robabilistic Spiking Neural N etworks for Com putational N eurogenetic M odelling, Proc. 6 th meeting on Co mputational Int elligence for Bioinformatics and Biostatistics, CIBB 2011, 30 June – 2 July, Gargangio, Italy, Springer LNBI
66. Kasabov, N., To spike or not t o spike: A proba bilistic spiking neuron model, Neural Netw., 23(1), 16–19, 2010.
67. Kilpatrick, Z .P., Br esloff, P .C. (2010) Effect of s ynaptic depression a nd adaptation on s patio-temporal d ynamics of a n excitatory neural networks, Physica D, 239, 547-560.
68. Kistler, G., and W. Gerstner, Spiking N euron Models - Single Neurons, Populations, Plasticity, Cambridge Univ. Press, 2002.
69. Legenstein, R ., C. N aeger, W . Maass, W hat C an a N euron Learn w ith S pike-Timing-Dependent Plasticity? N eural Computation, 17:11, 2337-2382, 2005.
70. Lotte, F ., M . C ongedo, A . Lécuyer, F . Lamarche, B . A rnaldi, A re view of c lassification a lgorithms for EEG-based br ain–computer interfaces, J. Neural Eng 4(2):R1-R15, 2007.
71. Maass, W . and H. M arkram, S ynapses as d ynamic m emory buffers, Neural Network, 15(2):155–161, 2002.
72. Maass, W ., a nd A . M . Z ador, Computing a nd learning w ith dynamic s ynapses, In: P ulsed Neural N etworks, pa ges 321 – 336. MIT Press, 1999.
73. Maass, W ., T. N atschlaeger, H . M arkram, R eal-time computing w ithout s table s tates: A n ew f ramework for ne ural computation ba sed on pe rturbations, N eural Computation, 14(11), 2531–2560, 2002.
74. Meng, Y ., Y . J in, J . Y in, a nd M . Conf orth (2010) H uman activity detection using spiking neural networks regulated by a gene re gulatory ne twork. P roc. Int . J oint Conf . on N eural Networks (IJ CNN), IE EE P ress, pp. 2232-2237, Ba rcelona, July 2010.
75. Mohemmed, A ., M atsuda, S ., Schliebs, S ., D hoble, K., & Kasabov, N. (2011). Optimization of Spiking Neural Networks with Dynamic Synapses for S pike Sequence Generation using PSO. In Proc.Int. Joint Conf. Neural Networks, pp. 2969-2974, California, USA, IEEE Press.
76. Mohemmed, A ., S chliebs, S ., M atsuda, S ., Kasabov, N ., Evolving S pike P attern A ssociation N eurons a nd N eural Networks, Neurocomputing, Elsevier, in print.
77. Mohemmed, A., Schliebs, S., Matsuda, S., Kasabov, N., SPAN: Spike P attern Association N euron for L earning S patio-Temporal Sequences, International Journal of Neural Systems, in print, 2012.
78. Natschläger, T., W. Maass, Spiking neurons and the induction of fi nite s tate machines, T heoretical Com puter S cience - Natural Computing, Vol. 287, Issue 1, pp.251-265, 2002.
79. NeMo spiking n eural ne twork s imulator, http://www.doc.ic.ac.uk/~akf/nemo/index.html
80. Nichols, E ., M cDaid, L .J., and Siddique, N .H. (2010), Case Study on S elf-organizing Spiking N eural N etworks for Robot Navigation, *International Journal of Neural Systems*, 20:6, pp. 501-508.
81. Norton, D. and Dan Ventura, Improving liquid state machines through iterative refinement of the reservoir, Neurocomputing, 73 (2010) 2893-2904
82. Nuzlu, H., K asabov, N ., S hamsuddin, S ., Widiputra, H., & Dhoble. (2011) . A n E xtended E volving S piking N eural Network Model for Spatio-Temporal Pattern Classification. In Proceedings of Int ernational Joint Confe rence on N eural Networks (pp. 2653-2656). California, USA, IEEE Press.
83. Nuzly, H. , N. Kasabov, S .Shamsuddin (2010) P robabilistic Evolving S piking N eural N etwork O ptimization U sing Dynamic Q uantum Ins pired P article S warm O ptimization, Proc. ICONIP 2010, Part I, LNCS, vol.6443.
84. Ozawa, S ., S . P ang and N K asabov, In cremental L earning of Chunk D ata for On-line P attern Classification S ystems, I EEE Transactions of N eural N etworks, vol .19, no .6, J une 2008 , 1061-1074,
85. Pang, S ., S . O zawa and N. Kasabov, I ncremental L inear Discriminant A nalysis for Cl assification of Data S treams, IEEE Trans. SMC-B, vol. 35, No. 5, 2005, 905 – 914
86. Pfurtscheller, G ., R . L eeb, C . K einrath, D . Friedman, C . Neuper, C . G uger, M . S later, Walking from t hought, Bra in Research 1071(1): 145-152, February 2006.
87. Ponulak, F ., and A . K asi´nski. Supervised learning i n spiking neural n etworks w ith Re SuMe: s equence learning, classification, and s pike s hifting. N eural Computation, 22(2):467–510, Feb. 2010. PMID:19842989.
88. Rabiner, L .R., A t utorial on hidden M arkov m odels a nd selected a pplications i n s peech r ecognition, P roc. I EEE, vol . 77, no. 2, pp. 257 - 285, 1989.
89. Rast, A .D., X in Jin, F rancesco Galluppi, L uis A . P lana, Cameron Patterson, S teve F urber, S calable Event-Driven Native P arallel P rocessing: T he S piNNaker Neuromimetic System, P roc. of t he A CM I nternational Co nference on

Computing Frontiers, pp. 21-29, May 17-19, 2010, Bertinoro, Italy, ISBN 978-1-4503-0044-5
90. Reinagel, P. and R. C. Reid. Precise firing events are conserved across neurons. Journal of Neuroscience, 22(16):6837–6841, 2002.
91. Reinagel, R. and R. C. Reid. Temporal coding of visual information in the thalamus. Journal of Neuroscience, 20(14):5392–5400, 2000.
92. Riesenhuber, M. and T.Poggio (1999) Hierarchical Model of Object Recognition in Cortex, Nature Neuroscience, 2, 1019-1025.
93. Rokem, A., S. Watzl, T. Gollisch, M. Stemmler, A. V. Herz, and I. Samengo. Spike-timing precision underlies the coding efficiency of auditory receptor neurons. J Neurophysiol, 2005.
94. Schliebs, R. (2005). Basal forebrain cholinergic dysfunction in Alzheimer´s disease – interrelationship with β-amyloid, inflammation and neurotrophin signaling. Neurochemical Research, 30, 895-908.
95. Schliebs, S., Hamed, H. N. A., & Kasabov, N. (2011). A reservoir-based evolving spiking neural network for on-line spatio-temporal pattern learning and recognition. In: 18th International Conference on Neural Information Processing, ICONIP 2011, Shanghai, Springer LNCS.
96. Schliebs, S., Kasabov, N., and Defoin-Platel, M. (2010), "On the Probabilistic Optimization of Spiking Neural Networks," *International Journal of Neural Systems*, 20:6, pp. 481-500.
97. Schliebs,S., M.Defoin-Platel, S.Worner, N.Kasabov, Integrated Feature and Parameter Optimization for Evolving Spiking Neural Netw.: Exploring Heterogeneous Probabilistic Models, Neural Netw.,22,623-632, 2009.
98. Schliebs, S., Mohemmed, A., & Kasabov, N. (2011). Are Probabilistic Spiking Neural Networks Suitable for Reservoir Computing? In: Int. Joint Conf. Neural Networks IJCNN, pp. 3156-3163, San Jose, IEEE Press.
99. Schliebs,S., Nuntalid, N., & Kasabov, N. (2010). Towards spatio-temporal pattern recognition using evolving spiking neural networks. Proc. ICONIP 2010, Part I, Lect. Notes in Computer Science (LNCS), 6443, 163-170.
100. Schrauwen, B., and J. Van Campenhout, "BSA, a fast and accurate spike train encoding scheme, in Neural Networks, 2003. Proceedings of the International Joint Conference on, vol. 4. IEEE, 2003, pp. 2825–2830.
101. Soltic and S. Kasabov, N. (2010), " Knowledge extraction from evolving spiking neural networks with rank order population coding ," *International Journal of Neural Systems*, 20:6, pp. 437-445.
102. Sona, D., H.Veeramachaneni, E.Olivetti, P.Avesani, Inferring cognition from fMRI brain images, Proc. of IJCNN, 2011, IEEE Press.
103. Song, S., K. Miller, L. Abbott et al., Competitive hebbian learning through spike-timing-dependent synaptic plasticity, Nature Neuroscience, vol. 3, pp. 919–926, 2000.
104. Theunissen, F. and J. P. Miller. Temporal encoding in nervous systems: a rigorous definition. Journal of Computational Neuroscience, 2(2):149–162, 1995.
105. Thorpe, S., and J. Gautrais, Rank order coding, Computational neuroscience: Trends in research, vol. 13, pp. 113–119, 1998.
106. Thorpe,S., Delorme, A., et al. (2001). Spike-based strategies for rapid processing. Neural Netw.,14(6-7),715-25.
107. van Schaik, A., L. Shih-Chii Liu, AER EAR: a matched silicon cochlea pair with address event representation interface, in: Proc. of ISCAS - IEEE Int. Symp. Circuits and Systems, pp. 4213- 4216, vol. 5, 23-26 May 2005.
108. Verstraeten, D., B. Schrauwen, M. D 'Haene, and D. Stroobandt, An experimental unification of reservoir computing methods, Neural Networks, 20(3):391 – 403, 2007.
109. Villa, A.E.P., et al., (2005) Cross-channel coupling of neuronal activity in parvalbumin-deficient mice susceptible to epileptic seizures. Epilepsia, 46(Suppl. 6) p. 359.
110. Wang, X., Hou, Z.G., Zou, A., Tan, M., and Cheng, L., A behavior controller for mobile robot based on spiking neural networks, Neurocomputing (Elsevier), 2008, vol. 71, nos. 4-6, pp. 655-666.
111. Watts, M. (2009) A Decade of Kasabov's Evolving Connectionist Systems: A Review, IEEE Trans. Systems, Man and Cybernetics- Part C: Applications and Reviews, vol.39, no.3, 253-269.
112. Weston, I., F.Ratle and R.Collobert (2008) Deep learning via semi-supervised embedding, in: Proc. 25th Int. Conf. Machine Learning, 2008, 1168-1175
113. Widiputra, H., Pears, R., & Kasabov, N. (2011). Multiple time-series prediction through multiple time-series relationships profiling and clustered recurring trends. Springer LNAI 6635 (PART 2), 161-172.
114. Widrow, B. and M. Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. Proceedings of the IEEE, 78(9):1415 –1442, sep 1990.
115. Wysoski, S., L. Benuskova, N. Kasabov, Evolving spiking neural networks for audiovisual information processing, Neural Networks, vol 23, 7, pp 819-835, 2010.
116. Xin Jin, Mikel Lujan, Luis A. Plana, Sergio Davies, Steve Temple and Steve Furber, Modelling Spiking Neural Networks on SpiNNaker, Computing in Science & Engineering, Vol: 12 Iss:5, pp 91 - 97, Sept.-Oct. 2010, ISSN 1521-961
117. Yu, Y.C. et al (2009) Specific synapses develop preferentially among sister excitatory neurons in the neocortex, Nature 458, 501-504.
118. Zhdanov, V.P. (2011) Kinetic models of gene expression including non-coding RNAs. Phys. Reports, 500, 1-42.