

Online Spatio-Temporal Pattern Recognition with Evolving Spiking Neural Networks utilising Address Event Representation, Rank Order, and Temporal Spike Learning

Kshitij Dhoble*, Nuttapod Nuntalid*, Giacomo Indiveri[†] and Nikola Kasabov*[†]

*Knowledge Engineering & Discovery Research Institute (KEDRI)

Auckland University of Technology, Auckland

Email: {kdhoble, nnuntali, nkasabov}@aut.ac.nz

[†]Institute of Neuroinformatics (INI),

University of Zurich and ETH Zurich (UZH — ETHZ)

Email: giacomo@ini.phys.ethz.ch

Abstract—Evolving spiking neural networks (eSNN) are computational models that evolve new spiking neurons and new connections from incoming data to learn patterns from them in an on-line mode. With the development of new techniques to capture spatio- and spectro-temporal data in a fast on-line mode, using for example address event representation (AER) such as the implemented one in the artificial retina and the artificial cochlea chips, and with the available SNN hardware technologies, new and more efficient methods for spatio-temporal pattern recognition (STPR) are needed. The paper introduces a new eSNN model dynamic eSNN (deSNN), that utilises both rank-order spike coding (ROSC), also known as time to first spike, and temporal spike coding (TSC). Each of these representations are implemented through different learning mechanisms - RO learning, and temporal spike learning - spike driven synaptic plasticity (SDSP) rule. The deSNN model is demonstrated on a small scale moving object classification problem when AER data is collected with the use of an artificial retina camera. The new model is superior in terms of learning time and accuracy for learning. It makes use of the order of spikes input information which is explicitly present in the AER data, while a temporal spike learning rule accounts for any consecutive spikes arriving on the same synapse that represent temporal components in the learned spatio-temporal pattern.

I. INTRODUCTION

Spatio- and spectro-temporal data (SSTD), that are characterised by a strong temporal component, are the most common types of data collected in many domain areas, including engineering (e.g. speech and video data), bioinformatics (e.g. gene expression data), neuroinformatics (e.g. EEG, fMRI), ecology (e.g. establishment of species), environment (e.g. global warming phenomenon), medicine (e.g. patients risk of disease and recovery data), economics (e.g. financial time series), etc. However, there is lack of efficient methods for modeling such data and for spatio-temporal pattern recognition (STPR) that can facilitate the discovery of complex STP from streams of data and the prediction of new spatio-temporal events. The brain-inspired spiking neural networks (SNN) [1], [2], considered the third generation of neural networks, are

a promising paradigm for STPR as these 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, phase, and to deal with large volumes of data in an adaptive and self-organising manner. With the development of new techniques to capture spatio-temporal data in a fast on-line mode, e.g. using address event representation (AER), such as the implemented one in the artificial retina chip [3] (see example in Fig.1) and the artificial cochlea chip [4], and with the advanced SNN hardware technologies [5], new opportunities have been created for efficient STPR across domain areas. But this still requires efficient and suitable methods.

The paper extends the known evolving SNN (eSNN) model [6], [7]), that utilises rank-order spike coding (ROSC) and RO learning rule (described in section 2 with temporal spike coding (TSC) representation and TSC learning rules namely the Fusis Spike Driven Synaptic Plasticity (SDSP) rule [8] (described in section 3) to arrive at a new model dynamic eSNN (deSNN) (section 4). The deSNN is demonstrated on a simple moving object classification problem where data was collected using AER in an artificial retina camera [3] (section 5). A comparative analysis of results between eSNN, deSNN, and an SNN that uses only SDSP learning rule shows the advantage of the proposed deSNN in terms of fast and accurate learning of AER data for STPR. Section 6 talks about future directions that include hardware realisations for real time applications.

II. RANK ORDER LEARNING AND EVOLVING SPIKING NEURAL NETWORKS (ESNN)

The RO learning rule in the eSNN uses important information from the input spike train the rank of the first incoming spike on each synapse. It establishes a priority of inputs (synapses) based on the order of the spike arrival on

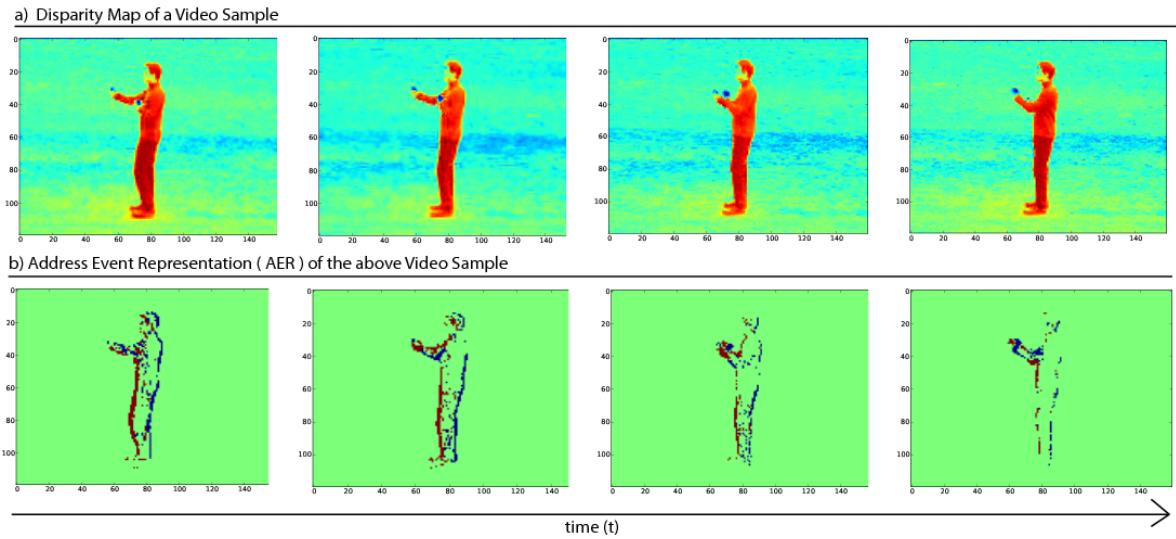


Fig. 1. Example of a spatio-temporal pattern (human movement) represented as spikes using AER. Figure (a) shows the disparity map of a video sample (from KTH dataset). Figure (b) shows the Address Event Representation (AER) for the above video sample shown in Fig.(a). Here the red and blue color represents the On and Off events respectively.

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 [9]. ROSC and RO learning makes use of the extra information of spike (event) order. RO learning utilises ROSC and was introduced in [9]. 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 RO learning is most appropriate for AER input data streams as the events and their addresses are entered into the SNN one by one, in the order of their happening. The eSNN structure and a supervised learning algorithm based on the RO were introduced in [6], [7]. They make use of the integrate-and fire (IF) model of a neuron [1] (fig.2). 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 using the RO rule 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 [6], [7]. 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. The postsynaptic potential of a neuron i at a time t is calculated as:

$$PSP(i, t) = \sum mod^{order(j)} W_{j,i} \quad (1)$$

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

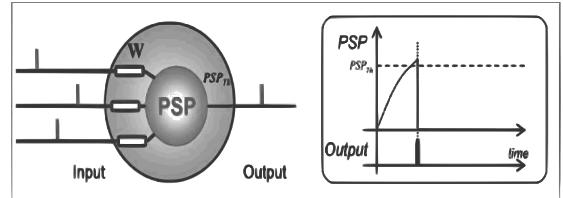


Fig. 2. Integrate-and-fire neuron with RO learning

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) a new output neuron is created and the connection weights are calculated based on the order of the incoming spikes. In the eSNN, the connection weights of on-line created connections between a neuron ni , representing an input pattern of a known class, and an activated input (feature) neuron nj , are established using the RO rule [9]:

$$\Delta W_{j,i} = mod^{order(j,i(t))} \quad (2)$$

After the whole input pattern (example) is presented, the threshold of the neuron ni is defined to make this neuron spike when this or a similar ST pattern (example) is presented again in the recall mode. The threshold is calculated as a fraction (C) of the total PSP, calculated as:

$$PSP_{max} = \sum_{j=1}^m \sum_{t=1}^T (mod^{order(j,i(t))} W_{j,i(t)}) \quad (3)$$

$$PSP_{Th} = C \cdot PSP_{max} \quad (4)$$

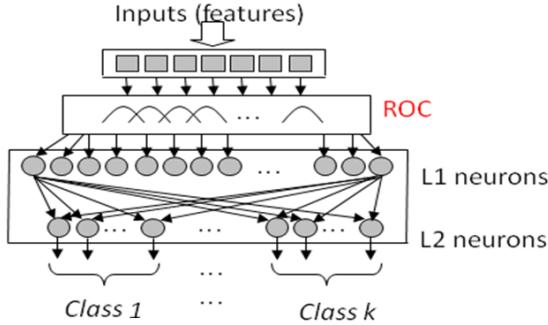


Fig. 3. eSNN for classification using population coding of inputs. Taken from [7]

If the connection weight vector of the trained neuron is similar to the one of an already trained neuron in a repository of output neurons for the same class, the new neuron will merge with the most similar one, averaging the connection weights and the threshold of the two neurons [6], [7]. Otherwise, the new neuron will be added to the class repository. The similarity between the newly created neuron and a training neuron is computed as the inverse of the Euclidean distance between weight matrices of the two neurons. An example of an eSNN for classification is given in fig.3 [6]. The recall procedure can be performed using different recall algorithms:

- The first one is when RO is used for a new input pattern (for recall, test) (Eq.2) and the connection weight vector for this input is compared with the patterns of existing neurons for which the output class is established during training. The closest neuron is the winner and defines the class of the new input pattern. This algorithm uses the principles of transductive reasoning [10] and nearest neighbour classification [11]. It compares synaptic weight vectors of a new neuron that captures a new input pattern and existing ones. We will denote this model eSNNs.
- A modification of the above algorithm is when spikes of the new input pattern are propagated as they arrive to all trained neurons and the first one that spikes (its PSP is greater than its threshold) defines the class. The assumption is that the neuron that best classifies the input ST pattern will spike earlier. This eSNN is denoted as eSNNm.

The main advantage of the eSNN is that it is computationally inexpensive and boosts the importance of the order in which spikes arrive to the neuron, thus making the eSNN suitable for on-line learning of mainly static data vectors (for some applications see [6], [7]). The problem of the eSNN is that there is no mechanism to deal with multiple spikes arriving at different times on the same synapse and representing same spatio-temporal pattern, which is needed for STPR. While the synapses capture long term memory during the learning phase, they have limited abilities (only

through the PSP growth) to capture short term memory, which is necessary for complex STPR tasks. Section 4 proposes an extended eSNN model with the use of SDSP learning [8], thus combining the two representations ROSC and STC. Section 5 demonstrates that the new model deSNN performs better than either the eSNN or the SDSP alone for a STPR problem.

III. TEMPORAL SPIKE CODING AND LEARNING RULES

TSC and temporal spike learning are observed in the auditory- and visual information processing in the brain as well as in motor control [12]. Its use in neuro-prosthetics is essential along with applications for a fast, real-time recognition and control of sequence of related processes [13]. 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) [14] and SDSP [8], the latter being implemented in a SNN hardware chip [13]. Temporal coding of information in SNN makes use of the exact time of spikes (e.g. in milliseconds). This is biologically observed in the visual-, auditory-, and pre-frontal cortex and the motor control brain area. Every spike matters and its time too. The TSC is used in several SNN models and learning algorithms, the most popular ones perhaps being STDP and SDSP as described below.

A. The Spike Timing Dependent Plasticity (STDP) learning rule

The STDP learning rule uses Hebbian form of plasticity in the form of long-term potentiation (LTP) and depression (LTD) [14]. Efficacy of synapses is strengthened or weakened based on the timing of post-synaptic action potentials in relation to the pre-synaptic spike (example is given in fig.4). 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).

B. The Fusi's Spike Driven Synaptic Plasticity (SDSP) Learning Rule

The SDSP is an unsupervised learning method [8], a modification of the STDP [13], that directs the change of the synaptic plasticity V_{w_0} of a synapse w_0 depending on the time of spiking of the pre-synaptic neuron and the post-synaptic neuron. V_{w_0} 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:

$$\Delta V_{w_0} = \frac{I_{pot}(t_{post})}{C_p} \Delta t_{spk} \quad \text{if } t_{pre} < t_{post} \quad (5)$$

$$\Delta V_{w_0} = -\frac{I_{dep}(t_{post})}{C_p} \Delta t_{spk} \quad \text{if } t_{post} < t_{pre} \quad (6)$$

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. In [13] the SDSP model 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 $0Hz$ 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.5) and trained for several hundreds of iterations. The SDSP model is implemented in the INI analogue SNN silicon chip [5]. 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 bistability 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 bistability 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 bistability circuits facilitate the implementation of both long-term and short term memory.

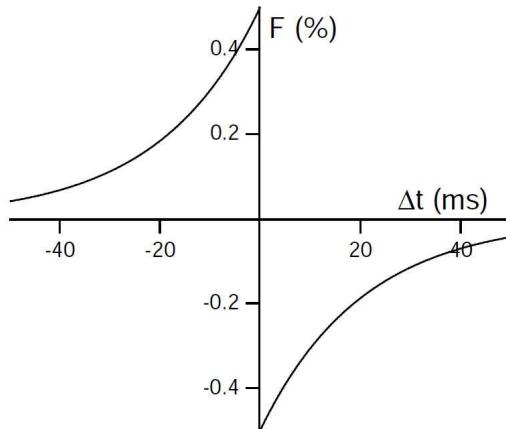


Fig. 4. An illustration of the STDP learning rule. Taken from [14]

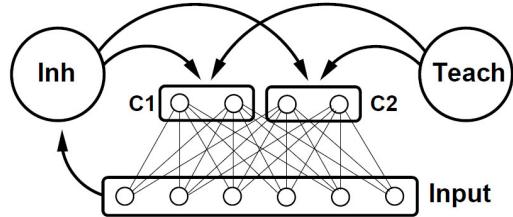


Fig. 5. An example of using a SDSP neurons. Taken from [13]

While successfully used for the recognition of static patterns, the potential of the SDSP SNN model and its hardware realisation have not been fully explored for STPR.

IV. THE PROPOSED deSNN WITH RO- AND SDSP LEARNING RULES

The main disadvantage of the RO learning eSNN is that they adjust their connection weights once only (based on the rank of the first spike), which is appropriate for static pattern recognition, but not for STPR. In the latter case the connection weights need to be further tuned based on following spikes on the same synapse using temporal spike learning. In the proposed deSNN both the RO- and SDSP learning rules are utilised. While the RO learning will set the initial values of the connection weights for a STPR utilising the existing event order information in an AER data, the STDP will adjust these connections based on following spikes (events) as part of the same spatio-temporal pattern.

The following is the training algorithm of deSNN:

- 1: **SET** deSNN parameters (including: Mod, C, Sim and the SDSP parameters)
- 2: **FOR** every input STP i represented as AER **DO**
 - 2a. Create a new output neuron j for this pattern and calculate the initial values of connection weights using the RO learning rule: $w_j = (Mod)^{\text{order}(j)}$
 - 2b. Adjust the connection weights w_j for consecutive spikes on the corresponding synapses using the SDSP learning rule.
 - 2c. Calculate PSP_{max}
 - 2d. Calculate the threshold value $x_i = PSP_{max(i)} * C$
 - 2e. **IF** the new neuron j weight vector w_j is similar to the weight vector of an already trained output neuron using Euclidean distance and a threshold Sim , then merge the two neurons:
 $w = w_{new} + w * N/N + 1,$
 $w = w_{new} + x * N/N + 1$
where N is the number of all previous merges of the merged neuron
 - ELSE**
 - END IF**
- 3: **END FOR** (Repeat to all input STP)

Two types of deSNN are proposed that differ in the recall algorithm and correspond to the two types of eSNN: eSNNs and eSNNm:

- (a) deSNNs
- (b) deSNNm.

V. EXPERIMENTAL SETTING AND RESULTS ON A SIMPLE MOVING OBJECT CLASSIFICATION TASK USING AER

Here we have used an AER data of a moving object classification collected through a silicon retina camera [3] - a moving irregular wooden bar in front of the camera. Two classes of movements are recorded as: crash and no crash. For the crash samples, the object is recorded as it approaches the camera and for no crash other movements such as up/down motion at a fixed distance from the camera are recorded. The size of the recorded area is 7,000 pixels. Each movement is recorded 10 times, 5 used for training and 5 for testing. Five models are created, trained and tested: SDSP, eSNNs, eSNNm, deSNNs and deSNNm. The parameter C for the SDSP, deSNNm and deSNNm has been optimized between 0 and 1 (with 0.1 step). The parameters used in the models are presented in Table I and the classification results along with the number of training iterations in Table II. The training and testing was carried out using 80:20 split.

A. Artificial Silicon Retina: Address Event Representation

Many of the real time machine vision systems have an inherent limitation of processing information on a frame by frame basis. [15] state that this often results in processing of redundant information present both within and across the frames (see Fig.7). However, this drawback is addressed by an Address Event Representation (AER) artificial silicon retina.

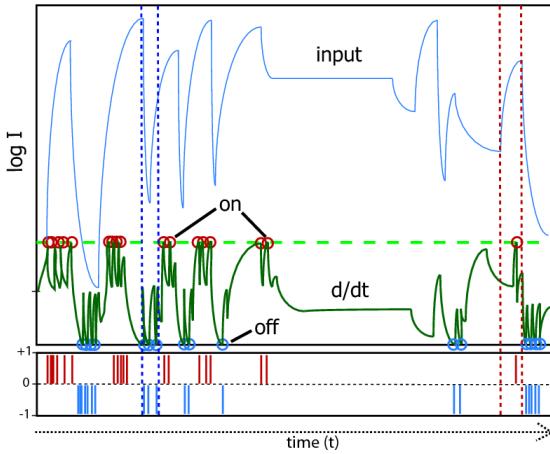


Fig. 7. These figures show the idealized pixel encoding and reconstruction of video data. The ON and OFF events represent significant changes in $\log I$. It can be seen that the changes greater than the threshold generate events. Adapted from [15].

This neuromorphic vision hardware generates events corresponding to the changes in $\log I$. The artificial silicon retina mimics aspects of our biological vision system which utilizes asynchronous spike events captured by the retina [5].

TABLE I
PARAMETER SETTINGS

For neurons and synapses	
Excitatory synapse time constant	2 ms
Inhibitory synapse time constant	5 ms
Neuron time constant (τ_{mem})	20 ms
Membrane leak	20 mV
Spike threshold (V_{thr})	800 mV
Reset value	0 mV
Fixed inhibitory weight	0.20 volt
Fixed excitatory weight	0.40 volt
Thermal voltage	25 mV
Refractory period	4 ms
For learning related parameters (Fusi)	
Up/Down weight jumps (V_{thm})	5 x ($V_{thr}/8$)
Calcium variable time constant (τ_{ca})	5 x (τ_{mem})
Steady-state asymptote for Calcium variable (w_{ca})	50 mV
Stop-learning threshold 1 (stop if $V_{ca} < thk1$)	1.7 x w_{ca}
Stop-learning threshold 2 (stop LTD if $V_{ca} > thk2$)	2.2 x w_{ca}
Stop-learning threshold 2 (stop LTP if $V_{ca} > thk3$)	(8 x w_{ca})- w_{ca}
Plastic synapse (NMDA) time constant	9 ms
Plastic synapse high value ($w_{p hi}$)	6 mvolt
Plastic synapse low value ($w_{p lo}$)	0 mvolt
Bistability drift	0.25
Delta Weight	0.12 x $w_{p hi}$
Other miscellaneous parameters / values	
Input Size	7000 spike train
Simulation time	1600 ms
mod (for rank order)	0.8

This allows fast and efficient processing since it discards the irrelevant redundant information by capturing only that spatio-temporal information corresponding to the temporal changes in $\log I$ (see Fig.7 & 1). For pixel illumination I , the operation for the continuous form can be defined as

$$\frac{d}{dt} \log I = \frac{dI/dt}{I} \quad (7)$$

B. The Address-Event Representation

In AER, each analog element on a sending device is assigned an address. When a spiking element generates an analog pulse its address is instantaneously put on a digital bus, using asynchronous logic (see Fig.10). In this asynchronous representation time represents itself, and analog signals are encoded by the inter-spike intervals between the addresses of their sending nodes. Address-events are the digital pulses written on the bus. In the case of single-sender/single-receiver communication, a simple handshaking mechanism ensures that all events generated at the sender side arrive at the receiver. The address of the sending element is conveyed as a parallel word of sufficient length, while the handshaking control signals require only two lines. Systems containing more than two AER chips can be constructed by implementing additional special purpose off-chip arbitration schemes and custom digital logic circuits [16], [17].

VI. CONCLUSION AND FUTURE DIRECTIONS

Amongst the five online one-pass learning algorithm, deSNNm method performed the best on this particular dataset yielding 90.00 percent accuracy. Other algorithms such as SDSP SNN, DeSNNs, eSNNs and eSNNm gave an accuracy

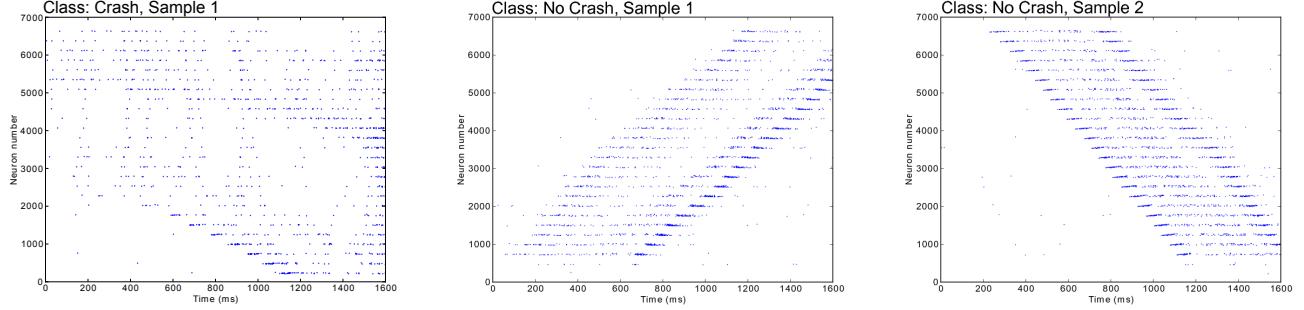


Fig. 6. These figures show the raster plot for the AER encoded samples from the *Crash* and *No crash* classes. It can be seen that there is a similarity between the spike trains of Crash class, sample 1 (left figure) & No crash class, sample 2 (right figure).

TABLE II

CLASSIFICATION ACCURACY FOR SPIKING NEURAL NETWORK BASED CLASSIFIERS. THE PAREMETER C FOR THE deSNNM AND deSNNM HAS BEEN OPTIMIZED BETWEEN 0 AND 1 (WITH 0.1 STEP)

	SNN Classifiers				
	SDSP SNN	eSNNs	eSNNm	deSNNs	deSNNm
Accuracy (%)	70	40	60	60	90
No. of training iteration	5	one-pass	one-pass	one-pass	one-pass

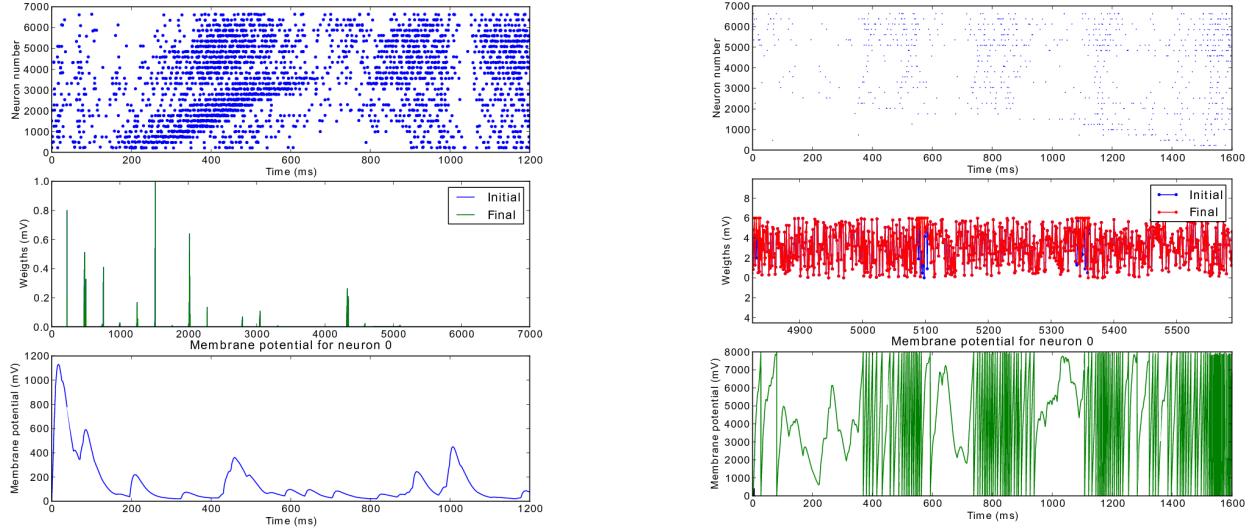


Fig. 8. The figure shows the spike raster plot, weights change and the membrane potential (for neuron 0) for the eSNNs that utilizes rank order without the SDSP dynamics

of 70%, 60%, 40% and 60% respectively. Also, the SDSP SNN method is not one-pass learning, therefore requiring several iterations which in this case is 5 iterations to achieve best results. The methods incorporating only the rank order coding (eSNNs, eSNNm) did not perform very well because (see raster plots of different classes) many of the samples from both the classes have almost similar spike patterns. The eSNNs especially under-performs due to the absence of SDSP mechanism and dynamic synapses. On the other hand it can be

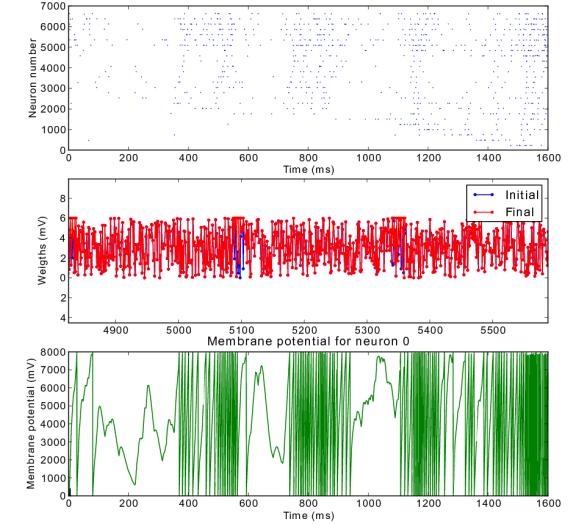


Fig. 9. This figure shows the spike raster plot, weights change and the membrane potential (mV) for the deSNNs. From the weights and the membrane potential (of neuron 0) graph it can be seen that due to the spike driven synaptic plasticity, the synaptic weights adjustments are faster compared to Fig. 8, for the sample from the same class

seen that these spike based classifiers are robust to noise and handle temporal aspect of the data very well. This experiment proves the feasibility of spike based classifiers. Compared to the traditional methods, this is a spike based approach where the need to convert spike times into vectors is eliminated make the whole system online, one-pass learning method.

Further development of the research is expected to be in the direction of utilizing neuromorphic computation for real

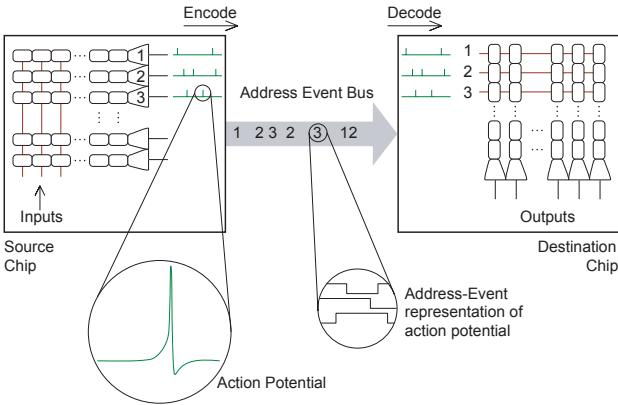


Fig. 10. Schematic diagram of an AER chip to chip communication example. As soon as a sending node on the source chip generates an action potential its address is written on the Address-Event Bus. The destination chip decodes the address-events as they arrive and routes them to the corresponding receiving nodes.

time applications. Example is given in fig.11. A major issue in the future development of deSNN models and systems for STPR is the optimization of the numerous parameters. One way is to combine the local learning of synaptic weights with global optimisation of SNN parameters. Three approaches can be investigated: evolutionary computation methods [18]; gene regulatory network (GRN) model [19], [20]; using both together. Neurogenetic models are promising for cognitive robotic systems and for the prognosis of neurodegenerative diseases such as Alzheimers disease [19] especially when probabilistic neuronal models are employed [21]. Personalized modeling and personalized medicine using personal spatio/spectro-temporal data can be further developed [22], [23] especially in the area of brain-machine interfaces [24].

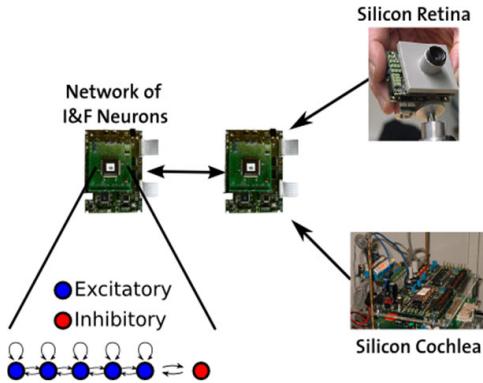


Fig. 11. An example of utilizing neuromorphic computation for real-time applications

ACKNOWLEDGMENT

The research is supported by an EU FP7 Marie Curie Project EvoSpike, the Knowledge Engineering and Discovery Research Institute KEDRI at the Auckland University of Technology and the Institute for Neuroinformatics, ETH and University of Zurich. We acknowledge the advice from Tobi Delbrück from INI.

REFERENCES

- [1] W. Gerstner and W. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge Univ Pr, 2002.
- [2] W. Maass and A. Zador, "Computing and learning with dynamic synapses," *Pulsed neural networks*, vol. 6, pp. 321–336, 1999.
- [3] T. Delbrück. (2007) jAER open source project. [Online]. Available: <http://jaer.wiki.sourceforge.net>
- [4] A. Van Schaik and S. Liu, "AER EAR: A matched silicon cochlea pair with address event representation interface," in *ISCAS-IEEE International Symposium on Circuits and Systems*. IEEE, 2005, pp. 4213–4216.
- [5] G. Indiveri, B. Linares-Barranco, T. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbrück, S. Liu, P. Dudek, P. Häfliger, S. Renaud *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in neuroscience*, vol. 5, 2011.
- [6] N. Kasabov, *Evolving connectionist systems: The knowledge engineering approach*. Springer-Verlag New York Inc, July 2007.
- [7] S. Wysoski, L. Benuska, and N. Kasabov, "Evolving spiking neural networks for audiovisual information processing," *Neural Networks*, vol. 23, no. 7, pp. 819–835, 2010.
- [8] S. Fusi, M. Annunziato, D. Badoni, A. Salamon, and D. Amit, "Spike-driven synaptic plasticity: theory, simulation, vlsi implementation," *Neural Computation*, vol. 12, no. 10, pp. 2227–2258, 2000.
- [9] S. Thorpe and J. Gautrais, "Rank order coding," *Computational neuroscience: Trends in research*, vol. 13, pp. 113–119, 1998.
- [10] Q. Song and N. Kasabov, "NFL: A neuro-fuzzy inference method for transductive reasoning," *Fuzzy Systems, IEEE Transactions on*, vol. 13, no. 6, pp. 799–808, 2005.
- [11] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.
- [12] A. Morrison, M. Diesmann, and W. Gerstner, "Phenomenological models of synaptic plasticity based on spike timing," *Biological Cybernetics*, vol. 98, no. 6, pp. 459–478, 2008.
- [13] J. Brader, W. Senn, and S. Fusi, "Learning real-world stimuli in a neural network with spike-driven synaptic dynamics," *Neural computation*, vol. 19, no. 11, pp. 2881–2912, 2007.
- [14] S. Song, K. Miller, L. Abbott *et al.*, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," *nature neuroscience*, vol. 3, pp. 919–926, 2000.
- [15] P. Lichtsteiner and T. Delbrück, "A 64x64 aer logarithmic temporal derivative silicon retina," *Research in Microelectronics and Electronics*, vol. 2, pp. 202–205, 2005.
- [16] E. Chicca, A. Whatley, P. Lichtsteiner, V. Dante, T. Delbrück, P. Del Giudice, R. Douglas, and G. Indiveri, "A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity," *IEEE Transactions on Circuits and Systems I*, vol. 5, no. 54, pp. 981–993, 2007.
- [17] D. Fasnacht and G. Indiveri, "A PCI based high-fanout AER mapper with 2 GiB RAM look-up table, 0.8 μ s latency and 66 mhz output event-rate," in *Conference on Information Sciences and Systems, CISS 2011*, Johns Hopkins University, March 2011, pp. 1–6.
- [18] G. Indiveri and T. Horiuchi, "Frontiers in neuromorphic engineering," *Frontiers in Neuroscience*, vol. 5, 2011.
- [19] N. Kasabov, R. Schliebs, and H. Kojima, "Probabilistic Computational Neurogenetic Modelling: From Cognitive Systems to Alzheimer's Disease," *IEEE Transactions on Autonomous Mental Development*, vol. 3, no. 4, pp. 1–12, 2011.
- [20] L. Benušková and N. Kasabov, *Computational neurogenetic modeling*. Springer Verlag, 2007.
- [21] N. Kasabov, "To spike or not to spike: A probabilistic spiking neuron model," *Neural Networks*, vol. 23, no. 1, pp. 16–19, 2010.
- [22] ———, "Global, local and personalised modeling and pattern discovery in bioinformatics: An integrated approach," *Pattern Recognition Letters*, vol. 28, no. 6, pp. 673–685, April 2007.
- [23] N. Kasabov and Y. Hu, "Integrated optimisation method for personalised modelling and case studies for medical decision support," *International Journal of Functional Informatics and Personalised Medicine*, vol. 3, no. 3, pp. 236–256, 2010.
- [24] T. Isa, E. Fetz, and K. Müller, "Editorial: Recent advances in brain-machine interfaces," *Neural Networks*, vol. 22, no. 9, pp. 1201–1202, 2009.