

Evolving Spatio-Temporal Data Machines Based on the NeuCube Neuromorphic Framework: Design Methodology and Selected Applications

Nikola Kasabov^{a,*}, Nathan Scott^{a,*}, Enmei Tu^a, Muhaini Othman^a, Maryam Dobarjeh^a, Stefan Marks^{a,b}, Neelava Sengupta^a, Norhanifah Murli^a, Reggio Hartono^a, Israel Espinosa-Ramos^{a,e}, Elisa Capecchi^a, Lei Zhou^{a,i}, Fahad Alvi^a, Grace Wang^c, Denise Taylor^d, Valery Feigin^f, Sergei Gulyaev^g, Mahmoud Mahmoud^g, Zeng-Guang Hou^h, Jie Yangⁱ

^aKnowledge Engineering and Discovery Research Institute, Auckland University of Technology, Auckland, New Zealand

^bCoLab, Auckland University of Technology, Auckland, New Zealand

^cGambling & Addictions Research Centre, Auckland University of Technology, Auckland, New Zealand

^dHealth & Rehabilitation Research Centre, Auckland University of Technology, Auckland, New Zealand

^eCentro de Investigación en Computación, Instituto Politécnico Nacional, México

^fNational Institute for Stroke and Applied Neurosciences, Auckland University of Technology, Auckland, New Zealand

^gInstitute for Radio Astronomy & Space Research, Auckland University of Technology, Auckland, New Zealand

^hState Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China

ⁱShanghai Jiao Tong University, Shanghai, China

Abstract

The paper describes a new type of evolving connectionist systems (ECOS) called evolving spatio-temporal data machines based on neuromorphic, brain-like information processing principles (eSTDM). These are multi-modular computer systems designed to deal with large and fast spatio/spectro temporal data using spiking neural networks (SNN) as major processing modules. ECOS and eSTDM in particular can learn incrementally from data streams, can include ‘on the fly’ new input variables, new output class labels or regression outputs, can continuously adapt their structure and functionality, can be visualised and interpreted for new knowledge discovery and for a better understanding of the data and the processes that generated it. eSTDM can be used for early event prediction due to the ability of the SNN to spike early, before whole input vectors (they were trained on) are presented. A framework for building eSTDM called NeuCube along with a design methodology for building eSTDM using it is presented. The implementation of this framework in MATLAB and in PyNN is presented, the latter facilitating the use of neuromorphic hardware platforms to run the eSTDM. Selected examples are given of eSTDM for pattern recognition and early event prediction on EEG data, fMRI data, multisensory seismic data, ecological data, climate data, audio-visual data. Future directions are discussed, including extension of the NeuCube framework for building neurogenetic eSTDM and also new applications of eSTDM.

Keywords: Spatio/spectro temporal data, Evolving Connectionist Systems, Evolving Spiking Neural Networks, Computational Neurogenetic Systems, Quantum inspired spiking neural networks, evolving spatio-temporal data machines, NeuCube.

1. Introduction: Spatio & Spectro Temporal Data and the Challenges for Information Sciences

Most problems in nature require spatio- and/or 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 Δt , and the output variables belonging to Y , such that

$$F : X(\Delta t) \rightarrow Y \quad (1)$$

where $X(t) = (x_1(t), x_2(t), \dots, x_n(t))$ and $t = 1, 2, \dots, m$.

It is important for a computational model to capture and learn whole spatio- and spectro-temporal patterns from data streams in order to most accurately predict future events in or from new input data. Examples of problems involving SSTD are: brain cognitive state evaluation based on spatially distributed EEG electrodes (Kasabov et al., 2013b); fMRI data (Mitchell et al., 2003; Chu et al., 2011; Dobarjeh et al., 2015; Kasabov and Dobarjeh, 2015; Just, 2001; N. Murli et al., 2014); moving object recognition from video data (Delbruck and Lichtsteiner, 2007); evaluating risk of disease, e.g. heart attack, stroke (Kasabov et al., 2014); evaluating response of a disease to treatment based on clinical and environmental variables; modelling the progression of a neuro-degenerative disease, such as Alzheimer’s Disease; modelling and prog-

*Corresponding authors.

Email addresses: nkasabov@aut.ac.nz (Nikola Kasabov), nascott@aut.ac.nz (Nathan Scott)

nosis of the establishment of invasive species in ecology. The prediction of events in geology, astronomy, economics and many other areas also depend on accurate SSTD modelling.

The most commonly used models for dealing with temporal information, based on Hidden Markov Models (HMM) and traditional artificial neural networks (ANN), 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 (Schmidhuber, 2014). 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 (*i.e.* genetic information, accumulated through evolution). Thus, the brain is the ultimate inspiration for the development of new machine learning techniques for SSTD modelling. Indeed, brain-inspired Spiking Neural Networks (SNN) (Gerstner et al., 2012, 1997; Buonomano and Maass, 2009) 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) (Song et al., 2000) or Spike Driven Synaptic Plasticity (SDSP) (Fusi, 2003).

Organisation of this paper:

In Section 2 we introduce classical evolving connectionist systems, the conceptual predecessor of this work, including the evolving spiking neural network which this work is based around. The primary contribution of this paper is established in Section 3, where our design methodology for eSTDM in the NeuCube computational framework is proposed. An immersive visualisation for this framework is discussed in Section 4. In the following sections we apply this methodology to build example eSTDM for case studies, in: eSTDM for brain data, including EEG and fMRI (Section 5); neurogenetic models (Section 5.3); personalised modelling, including stroke prediction (Section 6), environmental applications, including invasive pest population prediction (Section 7); video data (Section 8); and general spectro-temporal data, including radioastronomy and audio data (Section 9) are presented. An implementation of the framework for neuromorphic hardware is discussed in Section 10.

2. Principles of Evolving Connectionist Systems and their Development

2.1. Principles of ECOS

The human brain uniquely combines low level neuronal learning in the neurons and the connections between them, and higher level rule abstraction leading to adaptive learning and abstract concept formation. This is the ultimate inspiration for the development of intelligent evolving connectionist systems (ECOS) where specially constructed artificial neural networks (NN) are trained on data, so that after training abstract knowledge representation can be derived that explains the data and can be further interpreted as a knowledge-based system.

ECOS are modular connectionist based systems that evolve their structure and functionality in a continuous, self-organised, on-line, adaptive, interactive way from incoming data (Kasabov, 1998, 2007). They can process both data and knowledge in a supervised and/or unsupervised way. ECOS learn local models from data through clustering of the data and associating a local output function for each cluster represented in a connectionist structure. They can learn incrementally single data records or chunks of data and also incrementally change their input features. ECOS further develops some connectionist information processing principles already introduced in classical NN models, such as SOM, RBF, FuzzyARTMap, Growing Neural Gas, Neuro-Fuzzy Systems, or RAN (Kasabov, 2007).

ECOS perform *adaptive local learning* – neurons are allocated as centres of data clusters and the system creates local models in these clusters. The clustering used in ECOS is on-line, one-pass, evolving clustering, which is in contrast to the traditional fuzzy clustering methods that use pre-defined number of clusters and many iterations (Bezdek, 1987; Yager and Filev, 1994).

The following are the main principles of ECOS as stated in Kasabov (1998):

1. Fast learning from large amount of data, e.g. using ‘one-pass’ training, starting with little prior knowledge;
2. Adaptation in a real time and in an on-line mode where new data is accommodated as it comes based on local learning;
3. ‘Open’, evolving structure, where new input variables (relevant to the task), new outputs (e.g. classes), new connections and neurons are added/evolved ‘on the fly’;
4. Both data learning and knowledge representation is facilitated in a comprehensive and flexible way, e.g. supervised learning, unsupervised learning, evolving clustering, ‘sleep’ learning, forgetting/pruning, fuzzy rule insertion and extraction;
5. Active interaction with other ECOSs and with the environment in a multi-modal fashion;

6. Representing both space and time in their different scales, e.g.: clusters of data, short- and long-term memory, age of data, forgetting, etc.;
7. System's self-evaluation in terms of behaviour, global error and success and related knowledge representation.

2.2. ECOS Development: EFuNN, DENFIS, eSNN

The development of ECOS, as a trend in neural networks and computational intelligence that started in 1998 (Kasabov, 1998) continued as many improved or new computational methods that use the ECOS principles have been developed along many applications.

While the classical ECOS such as EFuNN and DENFIS (Kasabov, 2007) use a simple McCulloch and Pitts model of a neuron, where data is represented as scalars, the further developed evolving spiking neural network (eSNN) architectures use a spiking neuron model, while applying the same or similar ECOS principles. eSNN use data represented as temporal sequences of spikes in a similar mode as information is represented and processed in the brain.

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. A spiking neuron model assumes input information represented as trains of spikes over time. When sufficient input information is accumulated in the membrane of the neuron and the neuron's post synaptic potential exceeds a threshold, the neuron emits a spike at its axon. Some of the state-of-the-art models of a spiking neuron include: early models by Hodgkin and Huxley (Hodgkin and Huxley, 1952); more recent models by Maas, Gerstner, Kistler, Izhikevich and others, e.g.: Spike Response Models (SRM); Integrate-and-Fire Model (IFM) Izhikevich models (Izhikevich, 2004); adaptive IFM; probabilistic neurogenetic model (Kasabov, 2010).

Based on the ECOS principles, an evolving spiking neural network architecture (eSNN) was proposed (Kasabov, 2007; Wysoski et al., 2010). It was initially designed as a visual pattern recognition system. The first eSNNs were based on the Thorpe's neural model (Thorpe, 2001), in which the importance of early spikes (after the onset of a certain stimulus) is boosted, called rank-order coding and learning. Synaptic plasticity is employed by a fast supervised one-pass learning algorithm.

The main advantage of the eSNN when compared with other supervised or unsupervised SNN models is that it is computationally inexpensive and boosts the importance of the order in which input spikes arrive, thus making the eSNN suitable for on-line learning with a range of applications. For a comprehensive study of eSNN see Wysoski et al. (2010).

Different eSNN models have been developed, including:

- Dynamic eSNN (deSNN) - an architecture that uses both rank-order and time-based learning methods

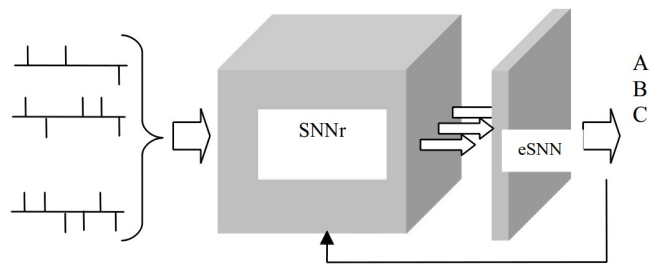


Figure 1: A principle diagram of an eSTDM.

Song et al. (2000) to account for spatio-temporal learning Dhoble et al. (2012); Kasabov et al. (2013a);

- Reservoir-based eSNN for spatio- and spectro-temporal pattern recognition (for principles of reservoir computing, see Verstraeten et al. (2007);
- Specialised architectures for EEG modelling and moving object recognition systems (Kasabov et al., 2013b)

3. The NeuCube Framework and a Design Methodology for Evolving Spatio-Temporal Data Machines (eSTDM)

3.1. General Architecture and Functionality of eSTDM

Our approach here to modelling large and fast stream SSTD is based on a common architecture of eSTDM as depicted in Figure 1. The functionality of an eSTDM is based on the following procedures:

1. Converting multivariable input stream data into spike sequences;
2. Unsupervised learning of spatio-temporal patterns from data in a SNN reservoir;
3. Supervised learning of classification/regression output system for classification/regression problems;
4. Using the evaluated/tested accuracy of the system as a feedback for improving the performance of this system in an iterative way if necessary.

The structure of the eSTDM resembles the structure of a LSM (Verstraeten et al., 2007), but the methodology for building such eSTDM in a specially proposed SNN computational framework called NeuCube departs significantly the classical neuro-computation and artificial intelligence approaches.

3.2. NeuCube: A Framework for eSTDM

The latest development in the direction of eSNN systems was proposed as a new architecture of an evolving spatio-temporal data machine (eSTDM) called NeuCube (Kasabov, 2014). It was initially proposed for spatio-temporal brain data modelling, but then it was further

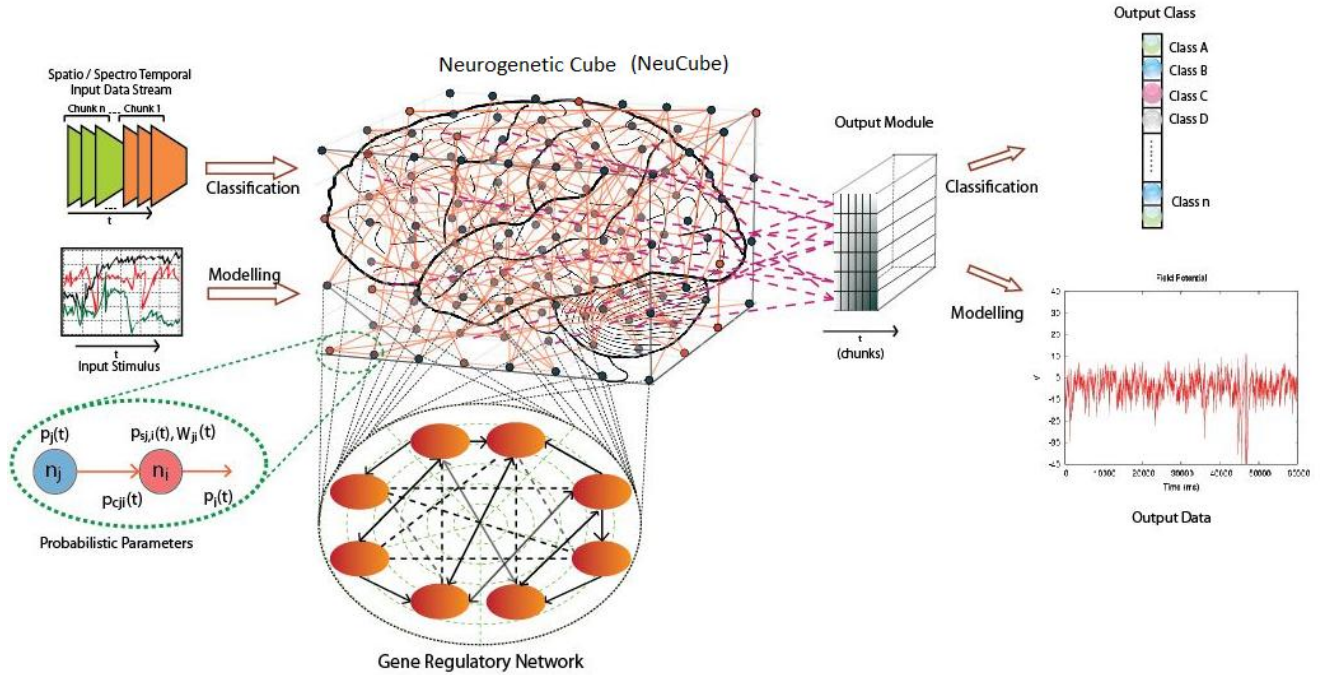


Figure 2: A block diagram of the NeuCube architecture initially proposed for brain data modelling but later used for a wide range of SSTD (Kasabov, 2014).

developed for other types of data as presented in this paper

A block diagram of the NeuCube architecture is provided in Figure 2. It consists of the following modules:

- Input information encoding module;
- Input mapping module;
- 3D SNN module (the Cube);
- Output classification/regression module;
- Gene regulatory network (GRN) module (Optional);
- Parameter optimisation module; and a
- Visualisation and knowledge extraction module.

The input module transforms input data into trains of spikes. Spatio-temporal data (such as EEG, fMRI, climate) is entered into the main module – the 3D SNNcube (SNNc). Different types of data can be used. This data is entered into pre-designated spatially located areas of the SNNc that correspond to the spatial location of the origin where data was collected (if such a location exists.)

Learning in the SNN is performed in two stages:

1. Unsupervised training, where spatio-temporal data is entered into relevant areas of the SNNc over time. Unsupervised learning is performed to modify the initially set connection weights. The SNNr will learn to activate same groups of spiking neurons when similar

input stimuli are presented, also known as a polychronization effect (Izhikevich, 2004).

2. Supervised training of the spiking neurons in the output module, where the same data that was used for unsupervised training is now propagated again through the trained SNN and the output neurons are trained to classify the spatio-temporal spiking pattern of the SNNc into pre-defined classes (or output spike sequences). As a special case, all neurons from the SNN are connected to every output neuron. Feedback connections from output neurons to neurons in the SNN can be created for reinforcement learning. Different SNN methods can be used to learn and classify spiking patterns from the SNNc, including the deSNN (Kasabov et al., 2013b) and SPAN models (Mohammed and Kasabov, 2012). The latter is suitable for generating motor control spike trains in response to certain patterns of activity of the Cube.

In an eSTDM similar activation patterns (called ‘polychronous waves’) can be generated in the SNNc with recurrent connections to represent short term memory. When using STDP learning, connection weights change to form LTP or LTD, which constitute long-term memory.

Results of the use of the NeuCube suggest that the NeuCube architecture can be explored for learning long spatio-temporal patterns and to be used as associative memory. Once data is learned, the SNNc retains the connections as a long-term memory. Since the SNNc learns functional pathways of spiking activities represented as

structural pathways of connections, when only a small initial part of input data is entered the SNNc will ‘synfire’ and ‘chain-fire’ *learned connection pathways* to reproduce *learned functional pathways*. Thus a NeuCube can be used as an associative memory and as a predictive system for event prediction when only some initial new input data is presented.

3.3. Design Methodology of eSTDM in NeuCube

In order to design an appropriate eSTDM for a given data source, a number of factors must be taken into consideration. Here, we identify these considerations.

- Which input transformation function do we use to encode the data as trains of spikes?
- Which input variable mapping into the SNNc? Is there some a-priori information we can use to spatially locate these?
- Which learning method do we use in the SNNc?
- Which output function is appropriate? In this, do we desire classification or regression?
- How to visualize an eSTDM for an improved understanding?
- Which parameter optimisation method will we apply?

For rapid prototyping and exploration of the NeuCube model, a version has been implemented in MATLAB. An up-to-date version of the NeuCube system accompanied with a detailed user manual can be freely obtained from our website¹.

3.3.1. Data Encoding

There are different coding schemes in SNN, primarily rate (information as mean firing rates) or temporal (information as *temporally significant*) coding. For NeuCube, we use temporal coding to represent information. So far four different spike encoding algorithms have been integrated into the existing implementation of the NeuCube, namely the Ben’s Spiker Algorithm (BSA), Address Event Representation (AER), Step-Forward Spike Encoding Algorithm (SF) and Moving-Window Spike Encoding Algorithm (MW). Figure 3 shows different results of the same SSTD, in this case an EEG signal, encoded by these four algorithms.

Different spike encoding algorithms have distinct characteristics when representing input data. BSA is suitable for high frequency signals and because it is based on the Finite Impulse Response technique, the original signal can be recovered easily from the encoded spike train. Only positive (excitatory) spikes are generated by BSA, whereas all other techniques mentioned here can also generate negative (inhibitory) spikes. AER was originally implemented

in hardware (Delbruck and Lichtsteiner, 2007) in the artificial silicon retina. It represents significant changes in signal intensity over a given threshold, where the ON and OFF events are dependent on the sign of the changes. However if the changes of the signal intensity vary dramatically, it may not be possible to recover the original signal using the encoded spike train generated by AER. Therefore, we propose here an improved spike encoding algorithm, SF, to better represent the signal intensity.

For a given signal $S(t)$ where $(t = 1, 2, \dots, n)$, we define a baseline $B(t)$ variation during time t with $B(1) = S(1)$. If the incoming signal intensity $S(t_1)$ exceeds the baseline $B(t_{1-1})$ plus a threshold defined as Th , then a positive spike is encoded at time t_1 , and $B(t_1)$ is updated as $B(t_1) = B(t_{1-1}) + Th$; and if $S(t_1) \leq B(t_{1-1}) - Th$, a negative spike is generated and $B(t_1)$ is assigned as $B(t_1) = B(t_{1-1}) - Th$. In other situations, no spike is generated and $B(t_1) = B(t_{1-1})$.

As to the Moving-Window Spike Encoding Algorithm, the baseline $B(t)$ is defined as the mean of previous signal intensities within a time window T , thus this encoding algorithm can be robust to certain kinds of noise.

Before choosing a proper spike encoding algorithm, we need to figure out what information the spike trains shall carry for the original signals, like AER for significant changes. After that, the underlying spike patterns in the spike trains will be better understood.

3.3.2. Input Variable Mapping

Mapping input variables into spatially located spiking neurons in the SNNc is a new approach towards modelling SSTD and a unique feature of the eSTDM. The main principle is that if spatial information about the input variables is known it can help in a) building more accurate models of the SSTD collected through these variables and b) a much better interpretation of the model and a better understanding of the SSTD. This is very important for data such as brain data such as EEG (see Kasabov (2014); Kasabov and Capecchi (2015)) and for fMRI data (see Figures 9 and 10) where patterns of interaction of brain signals can be learned and discovered. In some implementations we have used the Talairach brain template, mapped spatially into the SNNc (see Figure 2. Another way of mapping, when there is no spatial information available for the input variables, is to measure the temporal similarity between the variables to map variables with similar patterns into closer neurons in the SNNc. In way this is the vector quantisation principle, where by ‘vector’ here we use time series, not necessarily with the same length.

3.3.3. Learning

Learning in a eSTDM is a two-phase process as it was described in the NeuCube framework (*cf.* Section 3.2). The accuracy of a NeuCube model depends a great deal with the SNNc learning parameters and the classifier/regressor parameters. Optimisation procedures are discussed in Section 3.3.5.

¹<http://www.kedri.aut.ac.nz/>

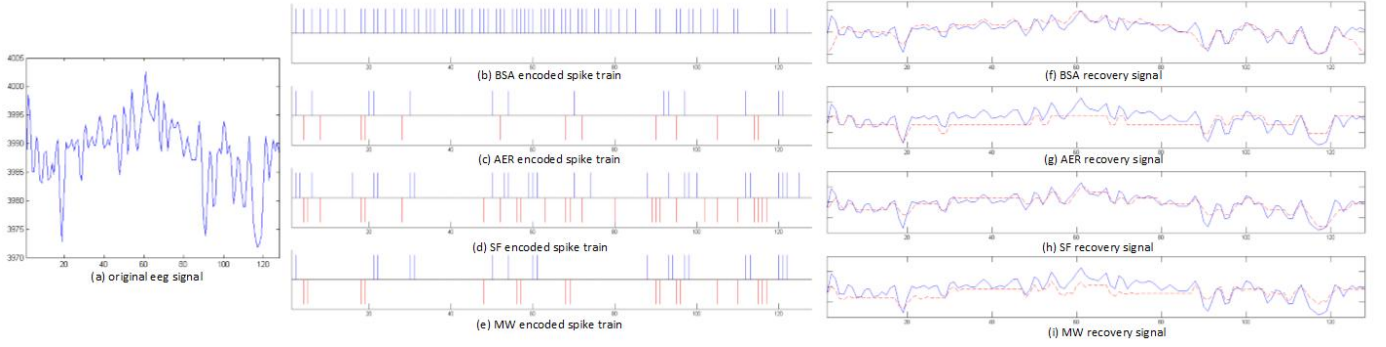


Figure 3: Spike trains generated by four different spike encoding algorithms in NeuCube with corresponding recovery signals. The blue(red) lines in (b),(c),(d),(e) are positive(negative) spikes, and the blue lines in (f),(g),(h),(i) are the original signals while the red dash lines are the signals reconstructed by corresponding spike trains.

3.3.4. Output Classification or Regression

An eSNN evolves its structure and functionality in an on-line manner, from incoming information. For every new input data vector, a new output neuron is dynamically allocated and connected to the input neurons. The neuron’s connections are initially established using the RO rule for the output neuron to recognise this vector (frame, static pattern) or a similar one as a positive example. The weight vectors of the output neurons represent centres of clusters in the problem space and can be represented as fuzzy rules (Soltic and Kasabov, 2010).

In some implementations neurons with similar weight vectors are merged based on the Euclidean distance between them. That makes it possible to achieve a very fast learning (only one pass may be sufficient), in both supervised and unsupervised modes (Kasabov et al., 2013a). When in an unsupervised mode, the evolved neurons represent a learned pattern (or a prototype of patterns). The neurons can be labelled and grouped according to their class membership if the model performs a classification task in a supervised mode of learning – an example is shown in Figure 4.

During a *learning phase*, for each M-dimensional training input pattern (sample, example, vector) P_i a new output neuron i is created and its connection weights $w_{j,i} (j = 1, 2, \dots, M)$ to the input (feature) neurons are initially calculated based on the order of the incoming spikes on the corresponding synapses using the RO learning rule:

$$w_{i,j} = \alpha \text{mod}^{\text{order}(j,i)} \quad (2)$$

where: α is a learning parameter (in a partial case it is equal to 1); mod is a modulation factor that defines how important the order of the first spike is; $w_{j,i}$ is the synaptic weight between a pre-synaptic neuron j and the postsynaptic neuron i ; $\text{order}(j, i)$ represents the order (the rank) of the first spike at synapse j, i ranked among all spikes arriving from all synapses to the neuron i ; $\text{order}(j, i) = 0$ for the first spike to neuron i and increases according to the input spike order at other synapses.

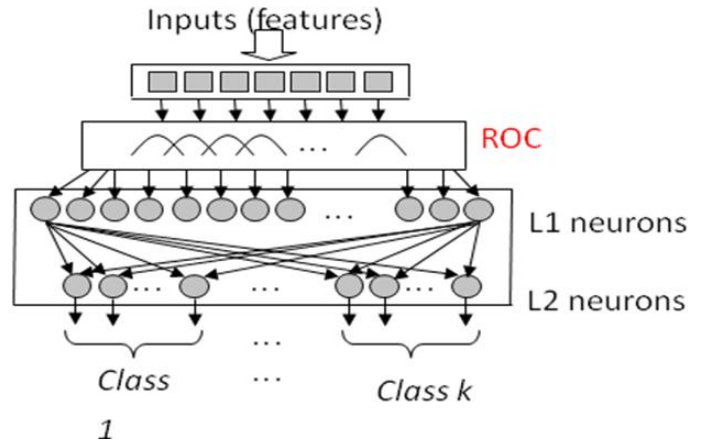


Figure 4: Example of an eSNN for classification using population RO coding of inputs (Soltic and Kasabov, 2010). Each input is connected to several feature neurons representing overlapping Gaussian receptive fields and producing spikes according to how much the current input variable value belongs to the receptive field: the higher the membership degree, the earlier a spike is generated and forwarded to the output neurons for learning or recall. A pool of output neurons representing different input vectors or prototypes is evolved for each class.

While the input training pattern (example) is presented (all input spikes on different synapses, encoding the input vector are presented within a time window of T time units), the spiking threshold Θ of the neuron i is defined to make this neuron spike when this or a similar pattern (example) is presented again in the recall mode. The threshold is calculated as a fraction (C) of the total PSP_i (denoted as PSP_i^{max}) accumulated during the presentation of the input pattern:

$$PSP_i^{max} = \sum_j \text{mod}^{\text{order}(j,i)} \quad (3)$$

$$\Theta = C PSP_i^{max} \quad (4)$$

If the weight vector of the evolved and trained new neuron is similar to the one of an already trained neuron (in a supervised learning mode for classification this is a neuron from the same class pool), *i.e.* their similarity is above a certain threshold Sim , the new neuron will be merged with the most similar one, averaging the connection weights and the threshold of the two neurons (Kasabov, 2007; Wysocki et al., 2010). Otherwise, the new neuron will be added to the set of output neurons (or the corresponding class pool of neurons when a supervised learning for classification is performed). 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. The merged neuron has weighted average weights and thresholds of the merging neurons.

While an individual output neuron represents a single input pattern, merged neurons represent clusters of patterns or prototypes in a transformed spatial – RO space. These clusters can be represented as fuzzy rules (Soltic and Kasabov, 2010) that can be used to discover new knowledge about the problem under consideration.

The eSNN (deSNN) learning is adaptive, incremental, theoretically ‘lifelong’, so that the system can learn new patterns through creating new output neurons, connecting them to the SNNc neurons, and possibly merging the most similar ones. The eSNN implement the 7 ECOS principles from Section 1.

During the recall phase, when a new spike sequence is presented, the spiking pattern is submitted to all created neurons during the learning phase. An output spike is generated by neuron i at a time l if the $PSP_i(l)$ becomes higher than its threshold Th_i . After the first neuron spikes, the PSP of all neurons are set to an initial value (e.g. 0) to prepare the system for the next pattern for recall or learning.

The postsynaptic potential $PSP_i(l)$ of a neuron i at time l is calculated as:

$$PSP_i(l) = \sum_{t=0,1,2,\dots,l} \sum_j e_j(t) \text{mod}^{\text{order}(j,i)} \quad (5)$$

where: $e_j(t) = 1$ if there is a first spike at time t on synapse j ; $\text{order}(j, i)$ is the rank order of the first spike at synapse j among all spikes to neuron i for this recall pattern.

The parameter C , used to calculate the threshold of a neuron i , makes it possible for the neuron i to emit an output spike before the presentation of the whole learned pattern (lasting T time units) which the neuron was initially trained to respond to. As a partial case $C = 1$. This is an important property of eSNN that can be utilized to train an eSNN on whole temporal input patterns and to recall the eSNN on a partial input pattern, *e.g.* only initial input data, so that the eSNN can predict an outcome earlier.

An alternative recall procedure implies the creation of a new output neuron for each recall pattern, in the same way as the output neurons were created during the learning phase, and then – comparing the connection weight vector of the new one to the already existing neurons using Euclidean distance. The closest output neuron in terms of synaptic connection weights is the ‘winner’. This method uses the principle of transductive reasoning and nearest neighbour classification in the connection weight space. It compares spatially distributed synaptic weight vectors of a new neuron that captures a new input pattern and existing ones. This model is denoted as eSNNs.

In one version of eSNN (the so-called deSNN) the connection weights, initially established with the RO learning rule, are modified based on the following spikes on the same synapse (Dhoble et al., 2012; Kasabov et al., 2013a).

3.3.5. Parameter Optimisation of NeuCube Models

eSTDM behaviours can be easily manipulated by changes in their large number of parameters. For example, differing neuron reset voltages can lead to a number of different spiking dynamics, and differing encoding parameters can significantly change the information density of the spike trains. To this end, a parameter search is usually performed in order to extract the best performance. Two primary techniques are discussed here; Grid Search, and the Quantum-Inspired search.

Grid Search. Grid search is a straightforward but effective method to tune parameters. Suppose there are P parameters that have to be optimized simultaneously. For each parameter there are three hyperparameters to be specified manually: the minimal value m and the maximal value M of the searching interval, and the searching step size s . Given these three hyperparameters of each optimizing parameter, we first create a P -dimension matrix, each dimension of which corresponding to a optimizing parameter, from m to M divided into $(M - m)/s$ entries. In this case, each entry of the matrix corresponds to a group of values of the optimizing parameters. Figure 5a shows a 2-dimension matrix, as an example of optimizing two parameters. Then we randomly split the training set into two equal-size parts, a training part and a validation part. For a specific group of values, we run the NeuCube system in a two-fold cross-validation way and the error rate of the cross-validation is added to the entry of the P -dimension matrix corresponding to that group of parameter values.

This process is repeated 10 times for each group of parameter values and the optimisation stops after all the groups of values corresponding to the whole matrix entries are evaluated.

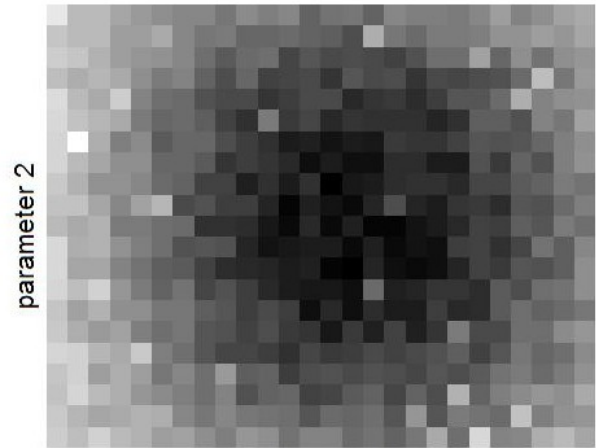
Instead of directly choosing the group of parameter values corresponding to the minimal entry in the matrix, we adopt another more robust method to determine the optimal parameters. We first apply a low pass filter by replacing each matrix entry with the average of its adjacent neighboring entries, and then, after all entries are filtered, we choose the group of parameter values corresponding to the minimal entry of the filtered matrix as the optimal one. The justification is that the performance surface of the system varies smoothly in parameter space, and after filtering some highly unstable points (entry whose value is extremely larger or smaller than its all adjacent neighboring entries, as shown in Figure 5c) will be reduced. Thus the minimal value of the matrix can capture the general trend of the performance surface, as illustrated in Figure 5b.

Quantum-Inspired Methods. QeSNNs use the principle of superposition of states to represent and optimize features (input variables) and gene parameters of an eSNN model (Kasabov, 2007). They are optimized through quantum inspired genetic algorithm (Defoin-Platel et al., 2009) or QiPSO. Features or genes are represented as qu-bits in a superposition of 1 (selected), with a probability α , and 0 (not selected) with a probability β . When the model has to be calculated, the quantum bits ‘collapse’ into a state of either 1 or 0. QeSNN need to be developed further in terms of both theory and applications.

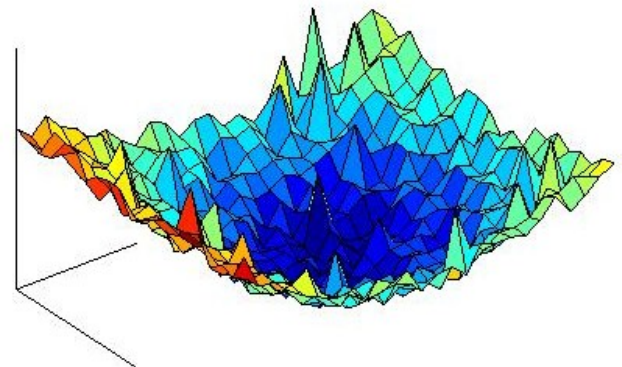
4. Dynamic and Immersive Visualisation of Neu-Cube Models

The number of neurons and connections within Neu-Cube as well as the 3-dimensional structure requires a visualisation that goes beyond a simple 2D connectivity/weight matrix or an orthographic 45-degree view of the volume. We created a specialised renderer for Neu-Cube datasets using JOGL (Java Bindings for OpenGL) and GLSL (OpenGL Shading Language) shaders to be able to render up to 1.5 million neurons and their connections with a steady framerate of 60 fps. In this view, neurons are displayed as stylised spheres, and connections are rendered as lines with green colour for excitatory connections and red for inhibitory connections. Spiking activity is shown as signals travelling along the connections.

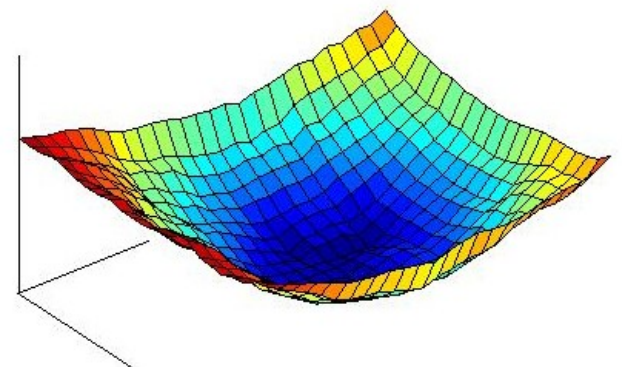
In conjunction with a 3D stereoscopic HMD (Head Mounted Display) like the Oculus Rift, it is easy for users to perceive the spatial structure of the network and the neuron positions. Furthermore, interaction mechanisms allow for playback of spiking patterns and the development of connection weights throughout the learning period. In addition, the visualization includes analysis functionality for the usage of connections to find ‘hot paths’, connection length analysis, and the ability to view the 3D structure



(a) Parameter Space



(b) Unstable Parameter Space



(c) Filtered Parameter Space

Figure 5: Grid search parameter optimisation and performance surfaces

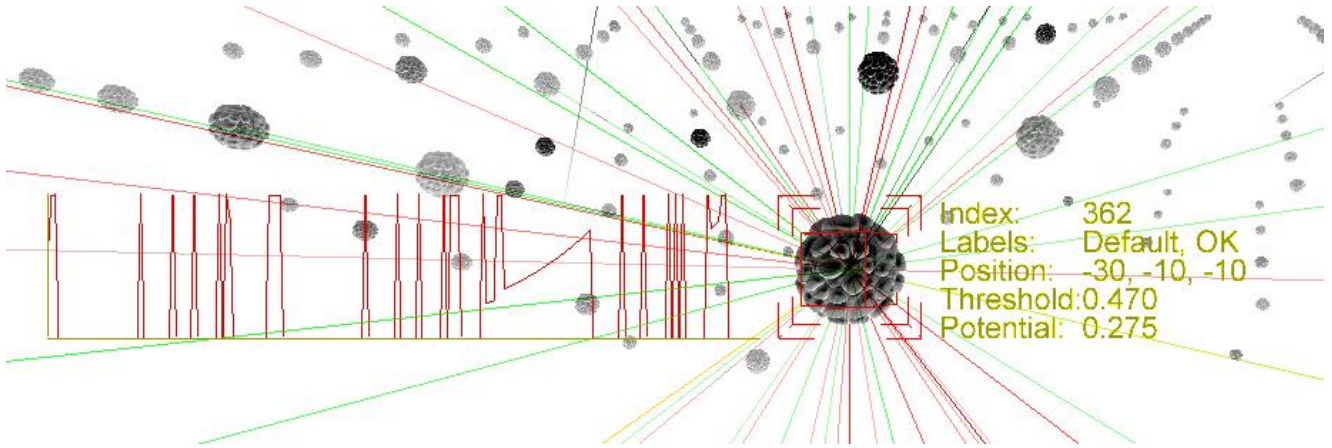


Figure 6: The cursor node can be used to view additional information about a specific neuron and its activity in the SNNc.



Figure 7: A user navigating through the virtual representation of the NeuCube network, using an intuitive, hand position based 3D cursor.

in ‘slices’. A 3D cursor metaphor is employed to look at neurons individually, their parameters, and their spiking history (see Figure 6).

The NeuCube visualization can run as a standalone program on a PC with a reasonable modern 3D graphics card and can be used with keyboard and mouse control. However, the full potential of the visualization is possible in a motion capture space, where the camera perspective and the cursor node position and orientation are controlled by markers that are attached on the actual HMD and a cursor implement (see Figure 7). This setup makes it possible for the user to literally walk through NeuCube and point out

individual neurons with the cursor in a natural manner.

In comparison to other scientific visualisation tools for neural networks such as BrainGazer (Bruckner et al., 2009) and Neuron Navigator (NNG) (Lin et al., 2011), our solution differs in that the user can naturally navigate through the 3D space by simply walking and gesturing instead of using mouse and keyboard shortcuts.

Closer to our visualisation is the work of von Kapri et al, who are using a Computer Assisted Virtual Environment (CAVE) to visualise the spatial structure and activity of a spiking neural network (von Kapri et al., 2011). However, due to the limited space within a cave environment, navigation by simply walking is not possible and requires indirect ways, *e.g.*, by using a controller.

We have not yet conducted a systematic user study, but so far, around 50 visitors of the Immersive VR space have experienced this visualisation. We have observed that, in general, people quickly start to move around and look at structures and point out individual neurons using the 3D cursor. The visualisation and interaction metaphors are very intuitive for new and experienced users.

5. eSTDM for Spatio-Temporal Brain Data Modelling and Understanding

5.1. eSTDM for EEG STBD

EEG has been used for the study of human neural activity recorded from the scalp for nearly a century. It can retrieve functional changes in the brain that occur over a period of milliseconds, is easy to manage, and is non-invasive. For these reasons, EEG has been used in brain computer interface (BCI) based systems to allow users to control devices, and for studying and staging of neurodegenerative disorders, and other clinical diagnostic purposes. As average human lifespan increases have been followed by the dramatic rise in the appearance of neurological diseases, the importance of such tools is clear.

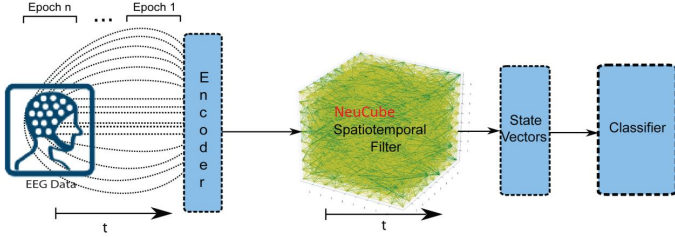


Figure 8: The NeuCube framework for EEG data classification and knowledge extraction. The picture shows the cube’s three principal modules: the input module, where input data are encoded into trains of spikes that are then presented to the main module, the SNNc; the NeuCube module, where time and space characteristics of the STBD are captured and learned; and the output module for data classification (or regression) and new knowledge discovery from the SNNc visualization.

EEG data contains temporal, spatial, and spectral information that is difficult to truly explore using standard statistical or ML techniques. Though these techniques are often used to process STBD, they lack the ability to classify differences in neurological dynamics that occur over the time, to identify the functional brain area involved, and to quantify the information involved. SNN however, are shown to be capable of such tasks (Taylor et al., 2014; Hu et al., 2014; Kasabov et al., 2013b).

In Kasabov and Capecci (2015) for example, an SNN methodology based in the NeuCube eSTDM was used for the study of 19-channel EEG data recorded from the scalp of subjects performing mental tasks. This research identifies that the NeuCube is able to classify and analyse changes in functional brain activities. This is significant, as it allows for the identification of the appearance of mild cognitive impairment (MCI) and the staging of its degeneration toward Alzheimer’s Disease (AD).

To study the EEG data, we have used a 3D SNNc of 1471 brain-mapped spiking neurons. Each of these neurons represented the centre coordinates of 1cm^3 of the Talairach Atlas, a human brain template (Talairach and Tournoux, 1988). The spike trains, obtained after encoding the real time EEG data using Address Event Representation (AER) algorithm, were entered into the SNNc from the 19 corresponding brain-mapped input neurons. The data was first learnt in an unsupervised way using Spike Time Dependent Plasticity Learning Rule (STDP) (Song et al., 2000) and then classified via supervised learning with the Dynamic Evolving SNN (deSNN) (Kasabov et al., 2013a). After training, the SNNc connectivity can be analysed and interpreted for a better understanding of the data and to identify differences in brain activity. A methodology diagram is given in Figure 8.

Excellent classification results of 100% test accuracy have been achieved. These have also been compared with

other traditional machine learning approaches, such as the Multi Layer Perceptron (MLP), Support Vector Machine (SVM), Inductive Evolving Classification Function (IECF) (Kasabov, 2007) and Evolving Clustering Method for Classification (ECMC). The LOOCV method was used to evaluate all the outputs. See the paper by Kasabov and Capecci (2015) for detailed results.

NeuCube performed significantly better compared with the other methods and with the highest accuracy, sensitivity and specificity over all. Thus, we believe that the NeuCube eSTDM can be successfully used for on-line learning and recognition of STBD. It also offers a better understanding of the information and the phenomena of study. Further improvement of the understanding and use of the model proposed are believed to contribute to the advancement in machine learning for the prediction and understanding of brain data and more specifically for data related to neurodegenerative pathologies, such as AD.

5.2. eSTDM for fMRI STBD

Recently there has been a huge interest in using functional magnetic resonance imaging (fMRI) to understand, analyze and predict behavior and cognition. The ability of fMRI to sample high resolution spatial information over time has been successfully used in correlating high-resolution neural activity with behavior. Several attempts have been made (Mitchell et al., 2003; Haxby et al., 2001), not only to identify the spatial distribution of activation across brain regions associated with cognitive tasks, but also to build computational models to distinguish them. The PBAIC 2007 competition was designed to detect cognitive tasks such as ‘seeing a dog’, ‘picking up a weapon’ *etc.* in a virtual reality environment.

Traditional machine learning algorithms like Gaussian Naive Bayes (Mitchell et al., 2003), or the SVM (Chu et al., 2011) has been used previously for this purpose. Some current research also focuses on the transformation of time series information to transformed space like shapelet-similarity, similarity in frequency domain *etc.* All of these techniques are focused mainly on classification accuracy (prediction), rather than *understanding* the spatio-temporal dynamics of the brain.

In contrast to statistical analysis and traditional machine learning methods, NeuCube is a rich computational model for fMRI data analysis (Doborjeh et al., 2015; Kasabov and Capecci, 2015). This method can be applied to fMRI data across areas of brain study and applications. The NeuCube neuromorphic spatiotemporal data machine has been used successfully on one of the benchmark datasets reported in (Kasabov and Doborjeh, 2015). The NeuCube-based methodology for learning, visualisation and classification of a benchmark fMRI data proposed here is shown graphically in Figure 9.

The number of brain voxels vary from one subject to another. The SNNc is scalable to evolve new neurons for different number of voxels. For this purpose, we mapped and analysed a known benchmark fMRI data

called STAR/PLUS (Just, 2001). The 3D size of the SNNc is scalable. This SNNc is composed of 51x56x8 spiking neurons corresponding to the maximum values of the x , y and z coordinates of the STAR/PLUS fMRI data.

In this experiment, we selected subject number “05780” from STAR/PLUS fMRI data sets. This data consists of 5062 voxels in the entire brain data. In order to visualise the whole brain structure, we loaded all voxel coordinates into the SNNc. Then, we fed the activity patterns of the pre-selected voxels into the corresponding allocated input neurons inside the cube. Figure 10 is a comparative illustration of the neuron connections created after the NeuCube learning procedure with different fMRI data streams related to different mental activities.

Another successful experiment involving the NeuCube and STAR/PLUS fMRI is on the classification of a subject looking at a picture or looking at a sentence which is conducted for each 6 subjects (N. Murlu et al., 2014). Comparing with the standard machine learning techniques (i.e. SVM and MLP), NeuCube has achieved more than 80% classification accuracy for all subjects. Neuron connectivity before and after training can help in understanding the data. The results suggest that a NeuCube model is more appropriate in handling complex fMRI data even without filtering the noise from the data. The noise may carry valuable information in defining the association between STBD samples, but failed to be recognized and processed in the standard machine learning techniques.

Recently the NeuCube was successfully used in modelling and predicting response to treatment of clozapine monotherapy responsive and non-responsive Schizophrenia patients using fMRI data. fMRI data was collected of 19 Schizophrenia subjects, 12 of which are Clozapine monotherapy responsive and 7 are non-responsive to clozapine monotherapy. The data was preprocessed for motion correction, coregistration and normalization. The data was also subsampled to approximately 4000 voxels.

As discussed in the previous sections, the standard NeuCube modules are used as part of NeuCube architecture, except some changes in the spatial distribution generation and supervised learning algorithm. As opposed to the canonical edition, the connection generation is not completely random within a small world. A Gaussian probabilistic neighborhood function is used to generate connections in the small world. A tempotron (Gutig and Sompolinsky, 2006) based supervised learning module is used for the classification of the unsupervised spike in these experiments. An average crossvalidated accuracy of 72% with 5% standard deviation was achieved after hyperparameter search. Further work is in progress which not only uses fMRI, but also simultaneously uses other modalities like DTI and EEG for better prediction accuracy and understanding.

5.3. Neurogenetic eSTDM

A neurogenetic model of a neuron is proposed and studied in Benuskova and Kasabov (2007). It utilises informa-

tion 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 the model is a dynamic Gene-Protein Regulatory Network (GRN) model of the dynamic interactions between genes and proteins over time that affect the spiking activity of the neuron (see Figure 2).

Currently, NeuCube-based model implements the STDP learning rule, based on the Hebbian theory, which defines a synaptic competition with respect to the order of incoming spikes, leading to control the postsynaptic action potentials over the time. However, this competition can also be implemented through dynamic mechanisms involving non-Hebbian synaptic growth or decay terms (Song et al., 2000).

In the Central Nervous System, these mechanisms are regulated by two opposite forces controlling the synaptic plasticity functionality. Spiking activity amongst neurons is intrinsically related with Glutamate and GABA neurotransmitters, and their receptors. While AMPA and NMDA Glutamate receptors mediate a fast and a slow excitatory synaptic response, the GABA_a and the GABA_b receptors regulate a fast and a slow inhibitory synaptic transmission. Additionally, these receptors are related with the learning and memory in the hippocampus.

To study how the spiking neuron postsynaptic action potential is affected by the dynamic of these four macromolecules, a new synaptic scaling model was proposed in Espinosa-Ramos et al. (Submitted 2015), which implemented the synaptic plasticity in a evolving Computational Neurogenetic Model (eCNGM). The model can automatically balance the synaptic strengths, making postsynaptic firing irregular but sensitive to presynaptic potentials such as the STDP like rules. It is inspired in the NMDA dependency on the AMPA, and their dependency on the inhibitory receptors.

After a spike is emitted by a neuron n_i and received by a neuron n_j , the activation of these excitatory receptors in a neuron n_j increases up to a maximum threshold value. If no spike is emitted, the inhibitory receptors activity increases in function to the time elapsed after the last spike is emitted, affecting the excitatory behaviour. A probability determines the activation of the GABA receptors, if GABA_a is activated then GABA_b is not, and opposite. The inhibition speed (fast or slow) is also determined by this probability, a higher activation probability means a faster inhibition, and therefore, the GABA_a probability must be higher than the GABA_b probability. .

Threshold values of each neuroreceptor can be modified according to the problem of interest and the data available; therefore, the possible effects that this change may have on the entire model connectivity and spiking activity can be studied.

The eCNGM based on the NeuCube framework need to be further developed in terms of both theory and applications, as they can be used for modelling and prediction of neurodegenerative diseases, such as cognitive impairment

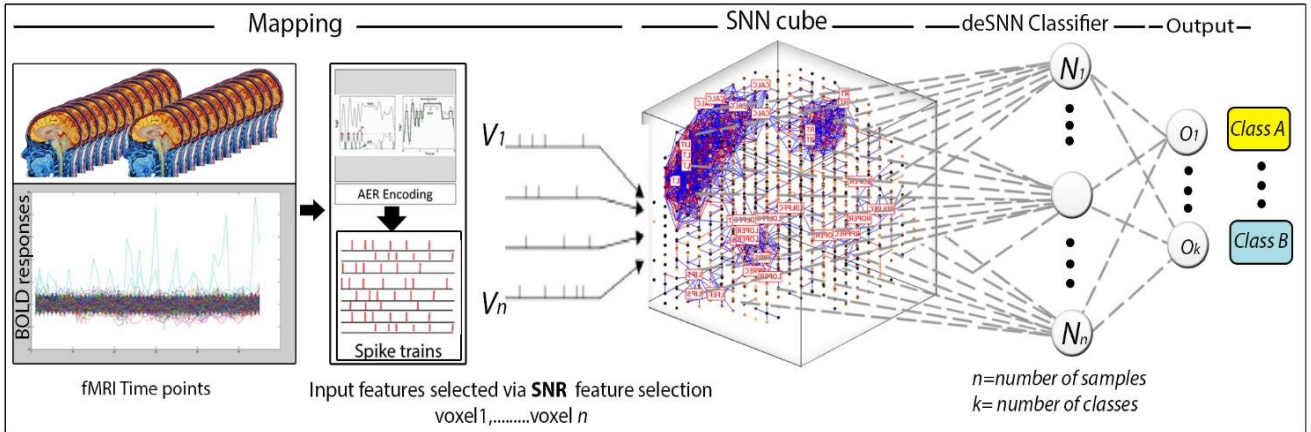


Figure 9: Schematic representation of the NeuCube-based methodology for mapping, learning, visualisation and classification of fMRI STBD

and memory loss that leads to serious disorders such as Alzheimer’s Disease (AD). In addition to brain data they make possible the study of gene data related to the same profile.

5.4. eSTDM for Brain-Machine Interfaces

The feasibility of using NeuCube with EEG data to develop a functional electrical stimulation BCI/BMI system that is able to assist in the rehabilitation of complex upper limb movements was shown in Taylor et al. (2014). A primary modality of the device is for subjects who have no voluntary activity in a limb, who would drive the device using mental imagery. However, the same model could be used for arbitrary output, to control a cursor or speaking device, for example.

In order to provide an effective tool for this purpose, a NeuCube eSTDM was trained on EEG data for a series of relatively complex muscle movements.

The preliminary experiments suggest that NeuCube is much more efficient for this task than standard machine learning techniques, resulting in high recognition accuracy, a better adaptability to new data, and a better interpretation of the models, leading to a better understanding of the brain data and the processes that generated it.

5.5. eSTDM for Neurorehabilitation

eSTDM are uniquely applicable for neurorehabilitation. Their biomimetic learning and information processing timescales are appropriate for integration with mentally-driven tasks. In addition, they offer the fast learning and incremental (continuous) learning required to adapt to the user’s changing abilities as their rehabilitation progresses. This application is a natural extension of eSTDM’s use in a BCI/BMI context.

Repetitive activities of daily living (ADL) and robotic active training are commonly practised in the rehabilitation of paralyzed patients, both of which have been proven

effective in the recovery of locomotor function in impaired limbs. Classification of ADL from EEG is of interest for the active robotic rehabilitation of patients with spinal cord injuries (SCI). This classification is a significant challenge with classical techniques, as these cannot deal effectively with the high noise, variability, and gradual change (due to the subject learning or rehabilitating the task) in the EEG signals.

Hu et al. (2014) performed an experiment using the NeuCube eSTDM to identify the upper-limb ADL of three classes with 14-channel EEG data. The continuous real-number signals are firstly encoded into spike trains through Ben’s Spike Algorithm (BSA). The generated spikes are then submitted into the SNNc reservoir. Spike trains from all neurons of the trained reservoir are finally classified using the dynamic evolving spiking neuron networks (deSNN). Classification accuracy using this technique is shown to be promising despite the highly noisy, low resolution EEG data (Hu et al., 2014). This experiment indicates strong potential for further exploration of the eSTDM for neurorehabilitation tasks.

6. eSTDM for Personalised Modelling and Personalised Event Prediction

6.1. Personalised Modelling

A special direction of ECOS is transductive reasoning and personalised modelling. Instead of building a set of local models (*e.g.* prototypes) to cover the whole problem space and then use these models to classify/predict any new input vector, in transductive modelling for every new input vector a new model is created based on selected nearest neighbour vectors from the available data. Such ECOS models are the Neuro-Fuzzy Inference model, NFI, and the Transductive Weighted Neuro-Fuzzy Inference Model, TWNFI (Kasabov, 2007).

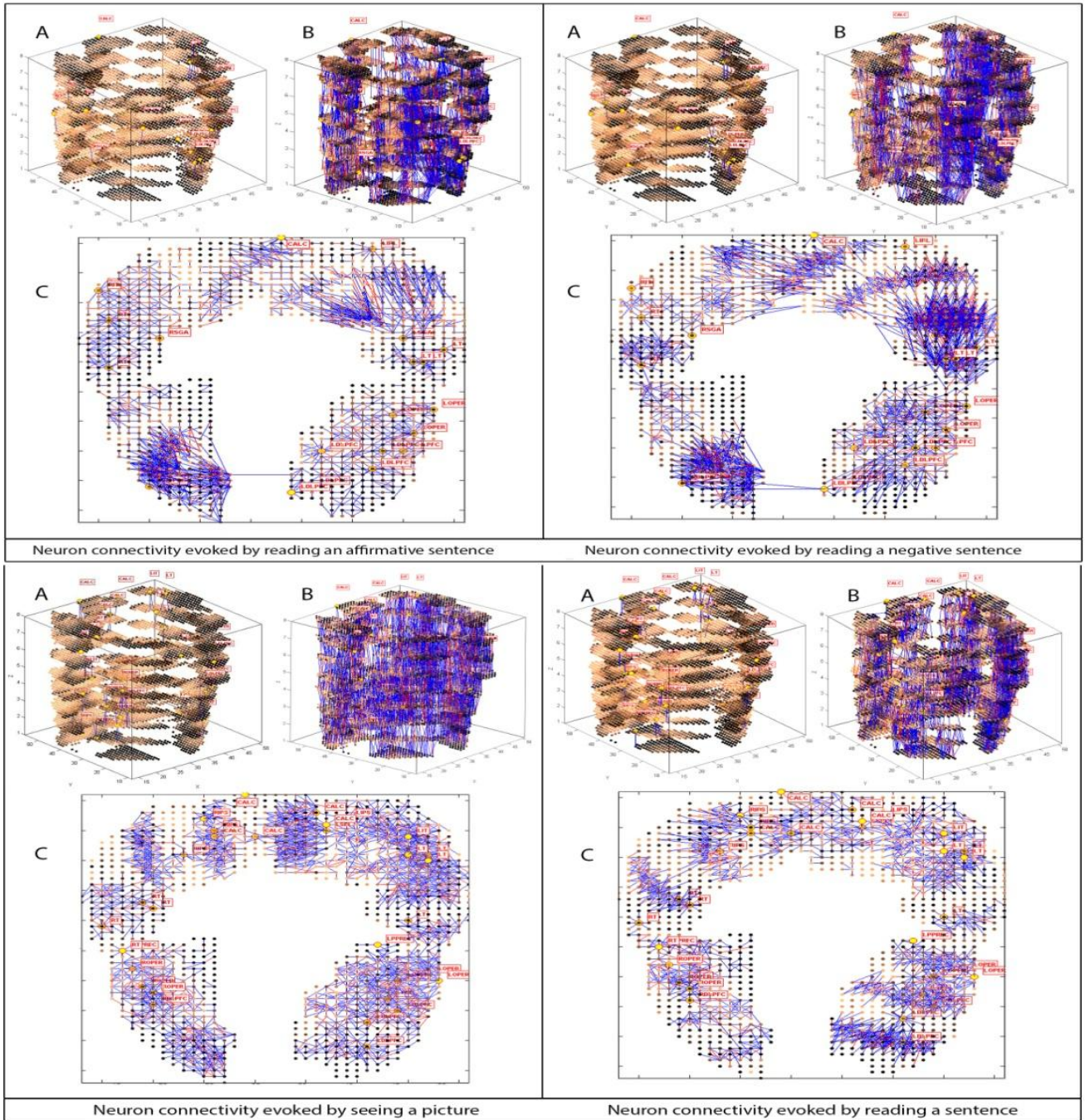


Figure 10: Illustrative mapping of 5062 voxels of a single individual fMRI data to the SNNc and the spatio-temporal connectivity which evolved through unsupervised learning from the spike sequences of 20 input voxels of the affirmative /negative sentence presentation and 33 input voxels of the picture/sentence presentation: (a) 3D visualisation of the initial connections between neurons before SNNc training; (b) 3D visualisation of the spatio-temporal connections after SNNc training; (c) 2D visualisation of the spatio-temporal connections after the SNNc training. Blue lines are positive connections while red lines represent negative connections. The brighter the colour of a neuron, the stronger its activity with neighbouring neurons. Thicknesses of the lines also identify the neuron’s enhanced connectivity. Zooming on particular areas of the connections in the SNNc would reveal more information about the brain processes related to the task.

In Kasabov et al. (2014), a methodology for personalised model creation is proposed based on the NeuCube framework. It builds an eSTDM for every individual based on both static and temporal data.

6.2. A Case Study of Early Stroke Prediction on an Individual Basis

The problem formulation for stroke occurrences is stated as: Given a set of individuals' data (static variables) and a set of environmental data (temporal variables), produce a model for an individual that predicts the earliest time point that individual is likely to suffer a stroke.

A feasibility study on the applicability of NeuCube eSTDM was published in Kasabov et al. (2014) where the dataset was taken from Auckland Regional Community Stroke Study population, consisting of 2805 patients data that suffered a stroke between the years 1981-1982, 1991-1992 and 2002-2003.

A chosen winter case study consists of subjects' age between 50 to 70 years, with a history of hypertension and smoking at the time the stroke occurred, in the year 2002. These subjects are described by eighteen variables which consist of six static features; age, gender, history of hypertension, smoking status, season, date of stroke; along with twelve environmental (temporal) features (continuous daily data) including eight (8) daily mean weather data (*e.g.* wind speed, min & max temperature, humidity); three daily mean air pollution data (*e.g.* NO² concentration); planetary geomagnetic activity, and solar radiation.

As NeuCube eSTDM functionality enables us to do predictive modelling, the experiment was designed in three ways:

1. One day earlier prediction where the whole 100% time period of 20 days was taken for analysis.
2. Six days earlier prediction (75% of the whole time period was taken)
3. Eleven days earlier prediction (50% of the whole time period was taken).

As a comparative experiment, tests were also designed for conventional machine learning methods (SVM, MLP, kNN, wkNN). Table 1 shows the best obtained accuracy from all the experiments.

The result clearly shows that NeuCube eSTDM performed better than conventional machine learning methods since it achieved an overall accuracy of 95% with a misclassification of class 1 (normal class). NeuCube eSTDM managed to 100% identify the stroke subjects (class 2) for this particular winter case study using the whole temporal patterns (one day ahead prediction). Furthermore for earlier time points of 6 days and 11 days prior prediction, NeuCube eSTDM managed to classify at 70% accuracy the risk of stroke occurrences in subjects. With earlier risk prediction, the subject could take precautions to minimize the risk or damage of a stroke.

Through visualisation tools in NeuCube eSTDM, connection patterns of temporal features can be analysed further. In NeuCube we can visualize input feature interactions, not only at group level but also on a personalised level leading to increased understanding of the relationships within the data and how these affect the results.

7. Ecological and Environmental Event Prediction

A NeuCube eSTDM would be suitable for learning the complex spatio-temporal relationships inherent in ecological and environmental data; for ecological applications to predict pest or crop populations; for seismic applications to potentially predict earthquake occurrence; and so on.

7.1. Case Study on Prediction of Risk of Aphid Population

In this section we consider how to use our NeuCube architecture to model and predict the population of a harmful species, *Rhopalosiphum padi*, in Southern New Zealand based on weather and climate factors.

We study a concrete case on aphid population prediction to demonstrate the capability of the NeuCube architecture for modeling ecological and environmental spatio-temporal data. In this study we use 14 weather variables which are recorded week by week from year 1982 to 2004 at the Canterbury Agricultural Research Centre, Lincoln, New Zealand (Hartono et al., 2014). Data preprocessing consisted of bad data removal, and time point alignment. Feature selection was applied to make sure the data was useful before entering the next phase. The real valued weather variables were transformed to spike trains with the Address Event Representation (AER) encoding algorithm. A 5x20x20 SNNs was generated and initialized according to small world connection rule to learn the temporal patterns in the spike trains. Then all the weather variables were mapped into the SNNs using a graph matching algorithm to ensure that the temporally dependent weather variables were mapped into nearby input neurons. The input data was propagated, and after the synaptic weights were learned using STDP, the weights were fixed and the spike trains fed to the SNNr again to obtain each neuron's firing state vector, which serves as the transformed feature of the original input signals in the following learning stage. The firing state vector of the SNNr is fed to a dynamic evolving spike neural network (deSNN) classifier to learn the underlying temporal pattern. After the whole system was trained, we used a validation data set to verify the validity of the system. The accuracy of the performance was evaluated by comparing the ground truth results with the predicted results.

Table 2 shows the results of predicting the autumn aphid population amount with the NeuCube system, as well as results of other traditional methods as a comparison. The testing time length means how many weeks weather measurements we used to predict the autumn aphid population. The fewer weeks used, the harder to predict accurately in the autumn, as shown in the last two columns.

Table 1: Comparative experimental results for all modelling methods (Kasabov et al., 2014) when applied to predicting a stroke occurrence

Method	Overall Accuracy (%)				
	SVM	MLP	kNN	wKNN	NeuCube
1 Day Earlier (100%)	55 (70, 40)	30 (50, 10)	40 (50, 30)	50 (70, 30)	95 (90, 100)
6 Days Earlier (75%)	60 (70, 30)	25 (20, 30)	40 (60, 20)	40 (60, 20)	70 (70, 70)
11 Days Earlier (50%)	50 (50, 50)	25 (30, 20)	45 (60, 30)	45 (60, 30)	70 (70, 70)

Table 2: Aphid Population Prediction Accuracy (%)

	Accuracy of each training and testing time length (weeks)		
	52 (full)	41.6 (early)	39
MLR	36.36	64.63	72.73
SVM	72.73	72.73	63.64
MLP	81.82	81.82	81.82
kNN	72.73	63.64	63.64
wkNN	72.73	63.64	63.64
NeuCube	100	90.91	81.82

7.2. A Feasibility Evaluation of using eSTDM for Seismic Data Modelling

Earthquake prediction is a challenging problem but compelling nonetheless. The immense capacity for destruction of earthquakes prompts for the ability to predict, within a reasonable time horizon, the occurrence of significant earthquakes so preemptive and anticipative actions could be taken to minimize the damage.

One of the potential uses of eSTDM in this field is to analyze the seismogram readings from multiple sites spread spatially across a geographical region to predict the occurrence of large earthquakes. A preliminary study using the waveform data obtained from the New Zealand GeoNet project web services (www.geonet.org.nz) has been done for the Canterbury region of New Zealand. After selecting the appropriate earthquakes from the earthquake catalog, the seismic waveform data collected prior to these earthquakes are fetched from selected four seismograph stations which were picked for their high uptime and availability.

To predict ahead of the actual event, the data fetched is offset by around twelve hours. The duration of the observation is 120 hours or five days long. The experiment is done by selecting twenty samples of earthquakes in the Canterbury region which are put into two categories based on the severity of the case (*i.e.* Strong – historically notable, and Weak – low energy seismic events unnoticed by the general population). The small number of samples is the consequence of the fact that strong earthquakes happen very rarely throughout the history and earthquakes before the year 2010 were not included because the availability and quality of the seismic activity data is not as

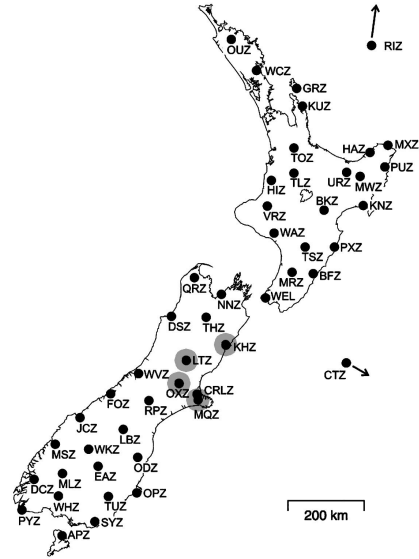


Figure 11: Seismograph sites across New Zealand with the 4 selected sites greyed.

good compared to those which happen after. The performance of the classifiers are measured in terms of the F-measure, which is the harmonic mean of precision and sensitivity of binary classification problems with the formula $F1 = 2TP / (2TP + FP + FN)$. The testing scheme is Leave-one-out cross validation, since the number of samples is small.

The result shown in Table 3 gives us the confidence that seismicity data might be a viable precursor for short-term earthquake prediction. The peak F-score of 0.73 means that the classifier successfully predict 7 out of 10 strong

Table 3: Preliminary results of earthquake prediction on a small data set using a NeuCube eSTDM in comparison with traditional techniques: SVM, MLP, NB.

Measure	SVM	MLP	NaiveBayes	NeuCube
Accuracy (%)	70	65	60	75
F-Score	0.66	0.58	0.55	0.73

earthquakes and raises only 2 false alarms. Though the experiment is in a very preliminary stage, this research has shown a promising way to predict the occurrence of strong earthquakes by training an eSTDM model to differentiate between strong and moderate earthquakes based on spatiotemporal seismicity precursors. For future works, it is important to fine-tune the models to get a better discriminating capability and using a larger dataset and getting more inputs from more seismic monitoring sites across the globe and running the analysis in real-time as the data is collected to produce a useful and practical disaster prediction system. A more comprehensive experiment should also be done to verify the accuracy and find the best prediction horizon and observation period. An interesting aspect is the extraction of spatiotemporal knowledge or rules pertaining to how the seismic activities in different sites affect each other.

8. eSTDM for Video Data Recognition

Video SSTD can be successfully learned in an eSTDM subject to the availability of quality data and the NeuCube eSTDM parameter optimisation. Here we demonstrate the feasibility of NeuCube for this purpose on a case study problem. Specifically, a model is created to classify a given video into one of three age groups based on its assessed age.

Aging is a slow process and its effects are visible only after a few months or a few years. But in spite of being slow, it remains a spatiotemporal phenomenon. The facial features of a person itself can be considered as a subspace and their aging over the years a temporal process. It would be very useful to incorporate the temporal, as well as spatial, patterns in aging data as an important part in classification.

The raw data which has been used in this study is from (Cerniello, 2013). It is five minutes of video containing 8943 frames of size 1920x1080 pixels. First the video is converted into greyscale frames. The nose tip of the subject in the image is manually annotated. The purpose was to locate a small region on the face which remains at a fixed distance from the annotated point. That same region is used for all the images in our study. This region is a part of the texture information of the face image, namely a small part of cheek portion of the face. This is chosen as facial skin is naturally smooth in youth and becomes wrinkled with age, thereby resulting in a change in the textural information present in this area. Based on this assumption 50 pixels are selected from cheek area of each face image.

All frames are divided into three classes. 128 frames of each sample are chosen, for each in a total of 60 samples. Thus the whole data comprises some 7680 images. The first 20 samples comprise young age, the next 20 samples adult age, and the third set of 20 samples represent old age.

In this experiment, the size of the SNNc is 1000 neurons, a relatively simple 10x10x10 cube. It is trained and tested

Table 4: Age Group Classification Accuracy (%) from video data

Method	Accuracy (%)
SVM	55
MLP	26
NeuCube	78

in a hold out method. Firstly we converted the video data into discrete spike trains using the AER encoding method to discretize the continuous signal, following the example of the silicon retina (Delbruck and Lichtsteiner, 2007). The deSNN classified mentioned previously is used here as an output classifier, because deSNN is computationally efficient and emphasizes the importance of the first spike, which has been observed in biological vision systems.

We conducted experiments to compare between traditional modeling methods (SVM and MLP) and our proposed method for age group classification. We designed two experiments for these baseline algorithms. Note that for these baseline algorithms, the time length of training samples and testing samples have to be the same as these methods cannot tolerate different lengths of feature vectors for training and testing.

It was observed that the classification achieved with NeuCube was better than other techniques. See Table 4 for results. Note that the techniques mentioned (other than NeuCube) do not have the capability of representing the spatio-temporal problem space effectively. These traditional techniques are only suitable for static data within a given time segment. Since an eSTDM models the relationships between and within spatio-temporal data, even a small input data will be able to trigger the spiking activities in SNNc, for an accurate pattern (class) recognition from video data.

9. eSTDM for Spectro-Temporal Data

9.1. eSTDM for Audio Information Processing

Audio data is spectro-temporal. It consists of temporal sequences of the intensity of the signal at different frequencies. How to map the frequencies into an SNNc is the first challenge. And then – training the SNNc on spike sequences that represent the audio signals is another challenge. A brief example is given in the subsection below.

9.2. Radioastronomy Data

Radioastronomy data is massively spectro-temporal. The timescale of meaningful background radiation in space is billions of years, and the volume of data to be processed to identify a small event is immense. eSTDM are currently being explored for applications in radioastronomy, as they are effective at learning in a noisy and dynamic environment, and explicitly incorporating the spatial, spectral, and temporal characteristics of such data.

10. Implementing the NeuCube on Neuromorphic Hardware

A system like the NeuCube, with its highly scalable architecture, requires a highly scalable computation platform. As traditional Von Neumann computational architectures reach their limits (Esmailzadeh et al., 2011; Perrin, 2011) in terms of power consumption, transistor size, and communication, new approaches must be sought. Neuromorphic hardware systems, especially designed to solve neuron dynamics and able to be highly accelerated compared to biological time are a response to these concerns. Systems such as analog VLSI or the SpiNNaker are advantageous by comparison to software based simulations on commodity computing hardware in areas such as biophysical realism; density of neurons per unit of processing power; and significantly lowered power consumption (Furber, 2012; Indiveri et al., 2011). This is not to say that simulations of the NeuCube cannot occur on traditional computing architectures; merely that dedicated hardware is advantageous in these areas and may be more appropriate for large-scale modelling. Subsequent to the modular framework for the development of NeuCube neuromorphic implementations written in Python first introduced in Scott et al. (2013), a cross platform version was written utilizing the PyNN API.

PyNN (Davison et al., 2008) is a generic SNN simulation markup framework that allows the user to run arbitrary SNN models on a number of different simulation platforms, including software simulators PyNEST and Brian, and some neuromorphic hardware systems such as SpiNNaker and FACETS/BrainScaleS. It provides a “write once, run anywhere” (where “anywhere” is the list of simulators it supports) facility for the development of SNN simulations. A version of the NeuCube has been implemented in this environment, for application on both commodity Von Neumann computing systems and dedicated neuromorphic hardware.

A key target of this NeuCube version is the SpiNNaker device currently in development. SpiNNaker is a general-purpose, scalable, multichip, multicore platform for the real-time massively parallel simulation of large scale SNN (Furber, 2012). Each SpiNNaker chip contains 18 ARM968 subsystems responsible for modelling up to one thousand neurons per core, at very low power consumption. These chips communicate through a custom multicast packet link fabric, and an arbitrary number of these chips can be linked together, with the assumption that the networks simulated exhibit some kind of connection locality. The small-world structure used in the NeuCube and its scalable nature are appropriate for implementation on this type of hardware.

An alternative implementation of the NeuCube eSTDM for embedded applications is currently being explored using the INI Neuromorphic VLSI chip (Indiveri et al., 2011).

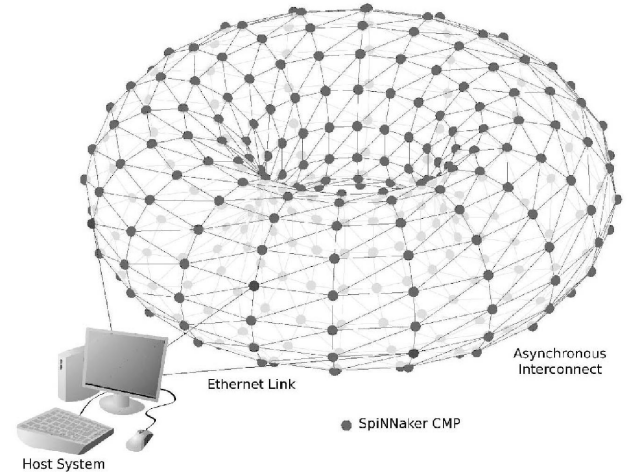


Figure 12: A conceptual schematic of the SpiNNaker system, demonstrating the interface and toroidal connection network.

11. Conclusion and Future Directions

The main goal of ECOS is to facilitate the creation of computational models and systems for adaptive learning and knowledge discovery from complex data. ECOS principles are derived from the integration of principles from neural networks, fuzzy systems, evolutionary computation, quantum computing and brain information processing. ECOS applications are manifold, but perhaps most welcome in the medical, environmental and health sciences, where the diagnostic phenomena are chaotic in nature and the data sets are massive and often incomplete. Here we present a new development of ECOS: the eSTDM, created in the NeuCube SNN environment. eSTDM is a promising approach to deal with big, stream data. Massive (so called ‘big’) data sets with the characteristics just described need to be analyzed, virtually in real time, for prognoses to be made and solutions to the issues sought at a level of urgency. In this sense, eSTDM for adaptive learning and knowledge discovery can make a great contribution to the methodologies employed by the emerging trans-disciplinary, integrative, systemic and problem-solving science.

There are some challenging questions that need to be further explored, for example:

1. What is the capacity of a NeuCube eSTDM in terms of both spatial and temporal characteristics of the data?
2. How much noise can be tolerated in an eSTDM?
3. How do we model transitions between spatio-temporal states triggered by external stimuli?
4. How early and accurately can an eSTDM predict an event from SSTD?

These are some of the questions that need to be addressed as a future work.

Acknowledgements

The work in this paper is supported by the Knowledge Engineering and Discovery Research Institute (KEDRI, <http://www.kedri.aut.ac.nz>). We were helped with the organization of this paper by Joyce D’Mello. Related papers, data, and software systems can be found at <http://www.kedri.aut.ac.nz> and <http://ncs.ethz/projects/evospike/>, where a NeuCube simulator can be downloaded free for use in research and teaching. The authors would also like to thank Steve Furber, Simon Davidson, and the SpiNNaker team at the Advanced Processor Technologies Research Group, University of Manchester; Yixong Chen and Jin Hu, and the Laboratory for Surgical and Rehabilitation Robots at the Institute for Automation, Chinese Academy of Sciences; and Giacomo Indiveri and the Neuromorphic Cognitive Systems group, Institute for Neuroinformatics, ETH & University of Zurich.

References

- Benuskova, L., Kasabov, N., 2007. Computational Neurogenetic Modeling. Springer US, Boston, MA.
URL <http://link.springer.com/10.1007/978-0-387-48355-9>
- Bezdek, J., 1987. Analysis of Fuzzy Information. CRC Press, Boca Raton, FL, USA.
- Bruckner, S., Solteszova, V., Groller, E., Hladuvka, J., Buhler, K., Yu, J., Dickson, B., 2009. BrainGazer - visual queries for neurobiology research. IEEE Transactions on Visualization and Computer Graphics 15 (6), 1497–1504.
- Buonomano, D., Maass, W., 2009. State-dependent computations: Spatio-temporal processing in cortical networks. Nature Reviews Neuroscience 10, 113–125.
- Cerniello, A., 2013. A Short Film on Ageing.
URL <http://aeon.co/video/health/danielle-a-short-film-about-the-ageing-process/>
- Chu, C., Ni, Y., Tan, G., J., S. C., Ashburton, J., 2011. Kernel regression for fMRI pattern prediction. Neuroimage 56 (9), 662–673.
- Davison, A. P., Brüderle, D., Eppler, J., Kremkow, J., Müller, E., Pecevski, D., Perrinet, L., Yger, P., Jan. 2008. PyNN: A Common Interface for Neuronal Network Simulators. Frontiers in Neuroinformatics 2, 11.
URL <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2634533&tool=pmcentrez&rendertype=abstract>
- Defoin-Platel, M., Schliebs, S., Kasabov, N., 2009. Quantum-inspired Evolutionary Algorithm: A multi-model EDA. IEEE Trans. Evolutionary Computation 13 (6), 1218–32.
- Delbruck, T., Lichtsteiner, P., May 2007. Fast sensory motor control based on event-based hybrid neuromorphic-procedural system. In: 2007 IEEE International Symposium on Circuits and Systems. IEEE, New Orleans, LA, USA, pp. 845–848.
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4252767>
- Dhoble, K., Nuntalid, N., Indiveri, G., Kasabov, N., Jun. 2012. Online spatio-temporal pattern recognition with evolving spiking neural networks utilising address event representation, rank order, and temporal spike learning. In: Proceedings of the 2012 International Joint Conference on Neural Networks. IEEE, Brisbane, Australia, pp. 1–7.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6252439
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6252439>
- Doborjeh, M. G., E., C., Kasabov, N., 2015. Classification and segmentation of fMRI Spatio-Temporal Brain Data with a NeuCube Evolving Spiking Neural Network Model. In: IEEE SSCI. IEEE.
- Esmailzadeh, H., Blem, E., St. Amant, R., Sankaralingam, K., Burger, D., Jul. 2011. Dark silicon and the end of multicore scaling. ACM SIGARCH Computer Architecture News 39 (3), 365.
URL <http://dl.acm.org/citation.cfm?doid=2024723.2000108>
- Espinosa-Ramos, J., Capecchi, E., Kasabov, N., Submitted 2015. A Synaptic Scaling Plasticity of a Computational Neurogenetic Model. Neural Network Letters.
- Furber, S., Aug. 2012. To Build a Brain. IEEE Spectrum 49 (8), 44–49.
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6247562>
- Fusi, S., Jan. 2003. Spike-driven synaptic plasticity for learning correlated patterns of mean firing rates. Reviews in the Neurosciences 14 (1-2), 73–84.
- Gerstner, W., Kreiter, A. K., Markram, H., M., H. A. V., 1997. Theory and simulation in neuroscience. Proc. Natl. Acad. Sci. USA 94 (24), 12740–1.
- Gerstner, W., Sprekeler, H., Deco, G., 2012. Theory and simulation in neuroscience. Science 338, 60–65.
- Gutig, R., Sompolinsky, H., 2006. The tempotron: a neuron that learns spike timing-based decisions. Nature Neuroscience 9 (3), 420–8.
- Hartono, R. N., Pears, R., Kasabov, N., Worner, S. P., Jul. 2014. Extracting temporal knowledge from time series: A case study in ecological data. In: 2014 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 4237–4243.
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6889918>
- Haxby, J. V., Gobbini, I. M., Furey, M. L., Alumi, I., Schouten, J. L., Pietrini, P., 2001. Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science 293 (5539), 2425–30.
- Hodgkin, A., Huxley, A., 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. Journal of Physiology 117, 500–544.
- Hu, J., Hou, Z.-G., Chen, Y.-X., Kasabov, N., Scott, N., Aug. 2014. EEG-based classification of upper-limb ADL using SNN for active robotic rehabilitation. In: 5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechanics. IEEE, pp. 409–414.
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6913811>
- Indiveri, G., Linares-Barranco, B., Hamilton, T. J., van Schaik, A., Etienne-Cummings, R., Delbruck, T., Liu, S.-C., Dudek, P., Häfliger, P., Renaud, S., Schemmel, J., Cauwenberghs, G., Arthur, J., Hynna, K., Folorosele, F., Saighi, S., Serrano-Gotarredona, T., Wijekoon, J., Wang, Y., Boahen, K., Jan. 2011. Neuromorphic silicon neuron circuits. Frontiers in Neuroscience 5, 73.
URL <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3130465&tool=pmcentrez&rendertype=abstract>
- Izhikevich, E. M., 2004. Which Model to Use for Cortical Spiking Neurons? IEEE Transactions on Neural Networks 15 (5), 1063–1070.
- Just, M., 2001. StarPlus fMRI data.
URL <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-81/www/>
- Kasabov, N., 1998. Evolving Fuzzy Neural Networks - Algorithms, Applications and Biological Motivation. In: Yamakawa, T., Matsumoto, G. (Eds.), World Scientific.
- Kasabov, N., 2007. Evolving Connectionist Systems: The Knowledge Engineering Approach. Springer.
- Kasabov, N., 2010. To spike or not to spike: A probabilistic spiking neural model. Neural Networks 23 (1), 16–19.
- Kasabov, N., 2014. NeuCube: A Spiking Neural Network Architecture for Mapping, Learning and Understanding of Spatio-Temporal Brain Data. Neural Networks 52, 62–76.
- Kasabov, N., Capecchi, E., Feb. 2015. Spiking neural network methodology for modelling, classification and understanding of EEG spatio-temporal data measuring cognitive processes. Information Sciences 294, 565–575.
URL <http://linkinghub.elsevier.com/retrieve/pii/>

S0020025514006562

- Kasabov, N., Dhoble, K., Nuntalid, N., Indiveri, G., 2013a. Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition. *Neural Networks* 41, 188–201.
- Kasabov, N., Doborjeh, M. G., 2015. Mapping, Learning, Visualisation and Classification of fMRI data in the NeuCube Evolving Spiking Neural Network Framework. *IEEE Trans. Neu. Net. & Learning Sys.*
- Kasabov, N., Feigin, V., Hou, Z.-G., Chen, Y., Liang, L., Krishnamurthi, R., Othman, M., Parmar, P., Jun. 2014. Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke. *Neurocomputing* 134, 269–279.
URL <http://linkinghub.elsevier.com/retrieve/pii/S092523121400109X>
- Kasabov, N., Hu, J., Chen, Y., Scott, N., Turkova, Y., 2013b. Spatio-temporal EEG data classification in the NeuCube 3D SNN Environment : Methodology and Examples. In: *Proceedings of the International Conference on Neural Information Processing*. Springer, Daegu, Korea, pp. 63–69.
- Lin, C.-Y., Tsai, K.-L., Wang, S.-C., Hsieh, C.-H., Chang, H.-M., Chiang, A.-S., 2011. The Neuron Navigator: Exploring the information pathway through the neural maze. In: *2011 IEEE Pacific Visualization Symposium (PacificVis)*. pp. 35–42.
- Mitchell, T. M., Hutchinson, R., Just, M. A., Niculescu, R. S., F., P., Wang, X., 2003. Classifying instantaneous cognitive states from fMRI data. In: *AMIA Annual Symposium Proceedings*. American Medical Informatics Association, p. 465.
- Mohammed, A., Kasabov, N., 2012. Incremental Learning Algorithm for Spatio-Temporal Spike Pattern Classification. In: *Proceedings of the IEEE World Congress on Computational Intelligence*. Brisbane, Australia, pp. 1227–1232.
- N. Murli, N., Kasabov, N., Handaga, B., 2014. Classification of fMRI Data in the NeuCube Evolving Spiking Neural Network Architecture. *Neural Information Processing*, 421–428.
- Perrin, D., Jan. 2011. Complexity and high-end computing in biology and medicine. *Advances in Experimental Medicine and Biology* 696, 377–84.
URL <http://www.ncbi.nlm.nih.gov/pubmed/21431578>
- Schmidhuber, J., 2014. Deep learning in neural networks: An overview. *Neural Networks* 61, 85–117.
- Scott, N., Kasabov, N., Indiveri, G., 2013. NeuCube Neuromorphic Framework for Spatio-Temporal Brain Data and Its Python Implementation. In: *Proceedings of the International Conference on Neural Information Processing*. Springer, Daegu, Korea, pp. 78–84.
- Soltic, S., Kasabov, N., 2010. Knowledge extraction from evolving spiking neural networks with rank order population coding. *International Journal of Neural Systems* 20 (6), 437–445.
- Song, S., Miller, K. D., Abbott, L. F., Sep. 2000. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience* 3 (9), 919–26.
URL <http://www.ncbi.nlm.nih.gov/pubmed/10966623>
- Talairach, J., Tournoux, P., 1988. *Co-planar Stereotaxic Atlas of the Human Brain: 3-Dimensional Proportional System*. Theirne Medical Publishers, New York, NY, USA.
- Taylor, D., Scott, N., Kasabov, N., Capecci, E., Tu, E., Saywell, N., Chen, Y., Hu, J., Hou, Z.-G., Jul. 2014. Feasibility of NeuCube SNN architecture for detecting motor execution and motor intention for use in BCI applications. In: *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 3221–3225.
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6889936>
- Thorpe, S., 2001. Spike-based strategies for rapid processing. *Neural Networks* 14 (6-7), 715–25.
- Verstraeten, D., Schrauwen, B., D’Haene, M., Stroobandt, D., 2007. An experimental unification of reservoir computing methods. *Neural Networks* 20 (3), 391–403.
- von Kapri, A., Rick, T., Potjans, T. C., Diesmann, M., Kuhlen, T., 2011. Towards the visualization of spiking neurons in virtual reality. *Studies in Health Technology and Informatics* 163, 685–87.
- Wysoski, S., Benuskova, L., Kasabov, N., 2010. Evolving spiking neural networks for audiovisual information processing. *Neural Networks* 23 (7), 819–35.
- Yager, R., Filev, D., 1994. Generation of fuzzy rules by mountain clustering. *J. of Intelligent and Fuzzy Systems* 2, 209–19.