IJBT (2017) 01: 49-65
RESEARCH ARTICLE

Artificial neural networks for artificial intelligence

Nikola Kasabov[1]

Abstract Artificial neural networks now have a long history as major techniques in computational intelligence with a wide range of applications for learning from data. There are many methods developed and applied so far, from multiplayer perceptrons (MLP) to the recent ones being deep neural networks and deep learning machines based on spiking neural networks. The paper addresses a main question for researchers and practitioners: Having data and a problem in hand, which method would be most suitable to create a model from the data and to efficiently solve the problem? In order to answer this question, the paper reviews the main features of the most popular neural network methods and then lists examples of applications already published and referenced. The methods include: simple MLP; hybrid systems; neuro-fuzzy systems; deep neural networks; spiking neural networks; quantum inspired evolutionary computation methods for network parameter optimization; deep learning neural networks and brain-like deep learning machines. The paper covers both methods and their numerous applications for data modelling, predictive systems, data mining, pattern recognition, across application areas of engineering, health, robotics, security, finances, etc. It concludes with recommendations on which method would be more suitable to use, depending on the data and the problems in hand, in order to create efficient information technologies across application domains.

Keywords: Artificial intelligence (AI) · Artificial neural networks · Evolving Connectionist Systems (ECOS) · Neuro-fuzzy systems · Spiking Neural Networks (SNN) · Evolving spiking neural networks · NeuCube · Quantum inspired neural networks · Spatio-temporal pattern recognition · Data mining

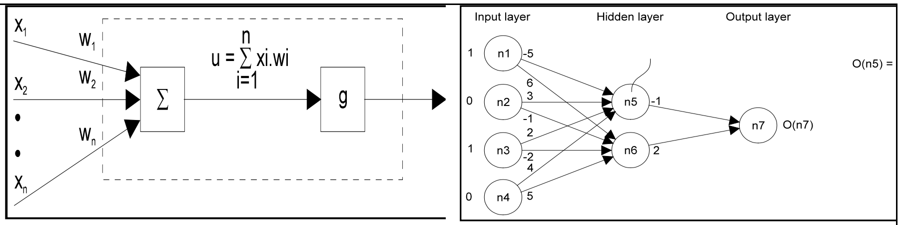
1 Artificial neural networks and hybrid systems

Artificial Neural Networks (ANNs) are computational models that mimic the nervous system in its main function of adaptive learning and generalization. ANNs are universal computational models. One of the most popular artificial neuron models is the McCulloch and Pitts neuron developed in 1943 (Figure 1a). It was used in early ANNs such as Perceptron [1] and multilayer perceptron [2-5] – a simple example is given in Figure 1b.

✉ Nikola Kasabov
nkasabov@aut.ac.nz, www.kedri.aut.ac.nz

1 Fellow IEEE, Director, Knowledge Engineering and Discovery Research Institute – KEDRI
Auckland University of Technology, New Zealand

These ANN are suitable when trained on a small scale static (vector-based) data, but are not adaptive to new data and in most cases they are ‘black boxes’ – they do not reveal internal structures in the data to be used to extract new knowledge. Optimal structures of ANNs are difficult to design.



(a) A simple artificial neuron. *(b) An example of a simple ANN.*

Fig. 1 Examples of simple artificial neuron models

In order to incorporate human knowledge into an intelligent system, an ANN module can be combined with a rule-based module in the same system. The rules can be fuzzy rules as a partial case [6, 7]. An exemplar system is shown in Figure 2, where, at a lower level, an ANN module predicts the next day value of a stock index and, at a higher level, a fuzzy reasoning module combines the predicted values with some macro-economic variables, using the following types of fuzzy rules [8]:

IF <the module stock predicted by the ANN is high> AND <the economic situation is good> (1)
THEN <buy stock>

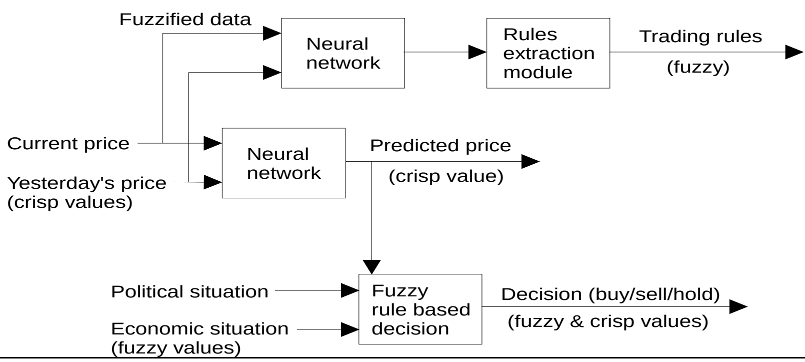


Fig. 2 A hybrid ANN-fuzzy rule-based expert system for financial decision support [8]

Hybrid systems can also use crisp propositional rules, along with fuzzy rules [9]. The hybrid systems from Figure 2 are suitable to use when decision rules are available to integrate with data.

Another group of ANN methods can be used not only to learn from data, but to extract rules from a trained ANN and/or insert rules into an ANN as initialization procedure. These are the neuro-fuzzy systems as discussed in the next section on the case of the evolving connectionist systems (ECOS).

2 Neuro-fuzzy and evolving connectionist systems

2.1 The principles of neuro-fuzzy and evolving connectionist systems

The integration of neural networks and fuzzy systems into one ANN attracted many researchers. The integration of fuzzy rules into a single neuron model and then into larger neural network structures, tightly coupling learning and fuzzy reasoning rules into connectionists structures, was initiated by Professor Takeshi Yamakawa and other Japanese scientists [10]. Many models of fuzzy neural networks were developed based on these principles [8, 11, 12].

In the evolving connectionist systems (ECOS) these ideas were developed further, where instead of training a fixed connectionist structure, the structure and its functionality are evolving from incoming data, often in an online, one-pass learning mode [12-16].

ECOS are modular connectionist based systems that evolve their structure and functionality in a continuous, self-organized, online, adaptive, interactive way from incoming information [13]. 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 items or chunks of data and also incrementally change their input features [15, 17]. Elements of ECOS have been proposed as part of the classical neural network models, such as Self-Organizing Maps, Radial Basis Functions, Fuzzy ARTMap, growing neural gas, neuro-fuzzy systems, Resource Allocation Network (for a review see [17]). Other ECOS models, along with their applications, have been reported in [18] and [19].

The principle of ECOS is based on *local learning* – neurons are allocated as centers of data clusters and the system creates local models in these clusters. Fuzzy clustering, as a means to create local knowledge-based systems, was stimulated by the pioneering work of Bezdek, Yager, and Filev [20, 21].

To summarize, the following are the main principles of ECOS as stated in [13]:

(1) Fast learning from large amount of data, e.g. using “one-pass” training, starting with little prior knowledge;

(2) Adaptation in 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;

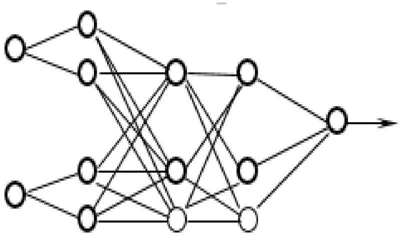
(6) Representing both space and time in their different scales, e.g., clusters of data, short- and long-term memory, age of data, forgetting, etc.;

(7) System's self-evaluation in terms of behavior, global error and success, and related knowledge representation.

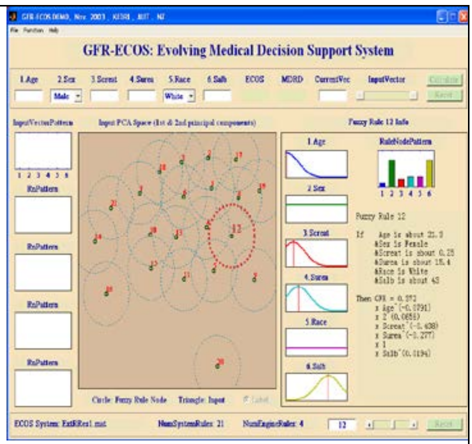
Here the concept of ECOS is illustrated on two implementations: the evolving fuzzy neural network (EFuNN) [14] and the dynamic evolving neuro-fuzzy inference system (DENFIS) [16]. Examples of EFuNN and DENFIS are shown in Figure 3a and Figure 3b, respectively. In ECOS, clusters of data are created based on similarity between data samples either in the input space (this is the case in some of the ECOS models, e.g., DENFIS), or in both the input and output space (this is the case, e.g., in the EFuNN models). Samples (examples) that have a distance to an existing node (cluster center, rule node) less than a certain threshold are allocated to the same cluster. Samples that do not fit into existing clusters form new clusters. Cluster centers are continuously adjusted according to new data samples, and new clusters are created incrementally. ECOS learn from data and automatically create or update a local fuzzy model/function, e.g.:

$$IF \ \textless data\ is\ in\ a\ fuzzy\ cluster\ C_i \textgreater \ THEN \ \textless the\ model\ is\ F_i\textgreater \qquad (2)$$

where F_i can be a fuzzy value, a logistic or linear regression function (Figure 3b) or ANN model [16, 17].



a) An example of EFuNN model (from [14])



b) An example of DENFIS model (from [15] and [17])

Fig. 3 Example implementations of ECOSs

The ECOS methods are realized as software modules as part of the free development system NeuCom (www.theneucom.com).

A special development of ECOS is *transductive reasoning and personalized modelling*. Instead of building a set of local models (i.e., 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 neighbor vectors from the available data. Such ECOS models are the neuro-fuzzy inference system (NFI) and the transductive weighted NFI (TWNFI) [22]. In TWNFI, for every new input vector the neighborhood of closets data vectors is optimized using both the distance between the new vector and the neighboring ones and the weighted importance of the input variables, so that the error of the model is minimized in the neighborhood area [23].

2.2 A survey of neuro-fuzzy and ECOS-based methods

The following is a survey list of selected methods that use the ECOS principles (full publications and details are available from www.ieeexplore.ieee.org, Google Scholar and Scopus):

- Evolving Self-Organized Maps (ESOM) [24];
- Evolving Clustering Method (ECM) [25];
- Incremental feature learning in ECOS [26];
- Online ECOS optimization [27];
- Assessment of EFuNN accuracy for pattern recognition using data with different statistical distributions [28];
- Recursive clustering based on a Gustafson–Kessel algorithm [29];
- Using a map-based encoding to evolve plastic neural networks [30];
- Evolving Takagi–Sugeno fuzzy model based on switching to neighboring models [31];
- A soft computing based approach for modeling of chaotic time series [32];
- Uni-norm based evolving neural networks and approximation capabilities [33];
- Global, local and personalised modelling and profile discovery in Bioinformatics: An integrated approach [34];
- FLEXFIS: a robust incremental learning approach for evolving Takagi–Sugeno fuzzy models [35];
- Evolving fuzzy classifiers using different model architectures [36];
- RSPOP: Rough Set–Based Pseudo Outer-Product Fuzzy Rule Identification Algorithm [37];
- SOFMLS: Online self-organizing fuzzy modified least-squares network [38];
- On-Line Sequential Extreme Learning Machine [39];
- Finding features for real-time premature ventricular contraction detection using a fuzzy neural network system [40];
- Evolving fuzzy rule-based classifiers [41];
- A novel generic Hebbian ordering-based fuzzy rule base reduction approach to Mamdani neuro-fuzzy system [42];
- Implementation of fuzzy cognitive maps based on fuzzy neural network and application in prediction of time series [43];
- Backpropagation to train an evolving radial basis function neural network [44];
- Smooth transition autoregressive models and fuzzy rule-based systems: Functional equivalence and consequences [45];
- Development of an adaptive neuro-fuzzy classifier using linguistic hedges [M46];

- A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system [47];
- Meta-cognitive RBF network and its projection based learning algorithm for classification problems [48];
- SaFIN: A self-adaptive fuzzy inference network [49];
- A sequential learning algorithm for meta-cognitive neuro-fuzzy inference system for classification problems [50];
- Architecture for development of adaptive online prediction models [51];
- Clustering and co-evolution to construct neural network ensembles: An experimental study [52];
- Algorithms for real-time clustering and generation of rules from data [53];
- SAKM: Self-adaptive kernel machine – A kernel-based algorithm for online clustering [54];
- A BCM theory of meta-plasticity for online self-reorganizing fuzzy-associative learning [55];
- Evolutionary strategies and genetic algorithms for dynamic parameter optimization of evolving fuzzy neural networks [56];
- Incremental leaning and model selection for radial basis function network through sleep learning [57];
- Interval-based evolving modeling [58];
- Evolving granular classification neural networks [59];
- Stability analysis for an online evolving neuro-fuzzy recurrent network [60];
- A TSK fuzzy inference algorithm for online identification [61];
- Design of experiments in neuro-fuzzy systems [62];
- EFuNNs ensembles construction using a clustering method and a co-evolutionary genetic algorithm [63];
- eT2FIS: An evolving type-2 neural fuzzy inference system [64];
- Designing radial basis function networks for classification using differential evolution [65];
- A meta-cognitive neuro-fuzzy inference system (McFIS) for sequential classification problems [66];
- An evolving fuzzy neural network based on the mapping of similarities [67];
- Incremental learning by heterogeneous bagging ensemble [68];
- Fuzzy associative conjuncted maps network [69];
- EFuNN ensembles construction using CONE with multi-objective GA [70].

2.3 Neuro-fuzzy and ECOS applications for AI

Based on the ECOS concepts and methods, sustained engineering applications have been developed, such as:

- Risk analysis and discovery of evolving economic clusters in Europe [71];
- Adaptive time series prediction for financial applications [72];
- Adaptive speech recognition [73];
- and others [17].

While the ECOS methods presented above use the McCulloch and Pitts model of a neuron (Figure 1a) and have been efficiently used for vector-based data, for rule extraction from data and for classification and prediction purposes, the further developed spiking neural networks (SNN) and evolving SNN (eSNN) architectures use a spiking neuron model and spike information representation. Spike information representation accounts for time in the data and for changes in the data over time. This is where SNN can be chosen as preferred methods and used efficiently.

3 Spiking Neural Networks and the brain-like AI

3.1 Main principles, methods and examples of SNN and evolving SNN (eSNN)

A spiking neuron model receives input information represented as trains of spikes over time. When sufficient input information is accumulated in the membrane of the neuron, the neuron's post synaptic potential exceeds a threshold and the neuron emits a spike at its axon (Figure 4).

Some of the-state-of-the-art models of a spiking neuron include: early models by Hodgkin and Huxley [74]; more recent models by Maas, Gerstner, Kistler, Izhikevich and others, e.g., Spike Response Models (SRM); Integrate-and-Fire Models (IFM) (Figure 4); Izhikevich models; adaptive IFM; probabilistic IFM [75, 76].

Based on the ECOS principles, an evolving spiking neural network architecture (eSNN) was proposed [17]. It was initially designed as a visual pattern recognition system. The first eSNNs were based on Thorpe's neural model [77], 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. Different eSNN models were developed, including:

- Reservoir-based eSNN for spatio- and spectro-temporal pattern recognition (Figure 5) [78];
- Dynamic eSNN (deSNN) [79] – a model that uses both rank-order and time-based spike-time dependent plasticity (STDP) learning rules [80] to account for spatio-temporal data.

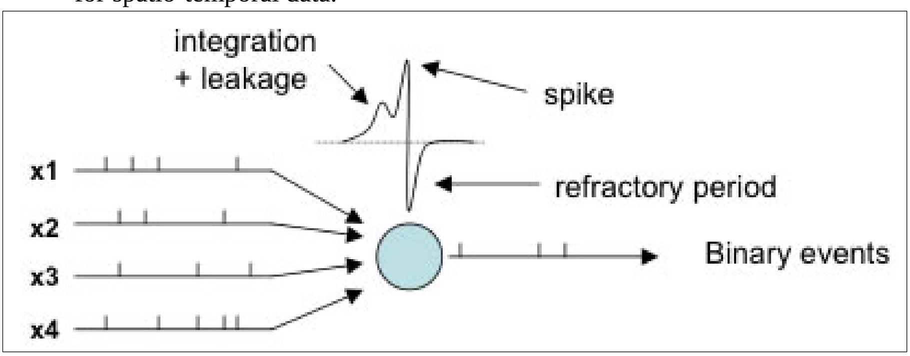


Fig. 4 The structure of the LIFM of a spiking neuron

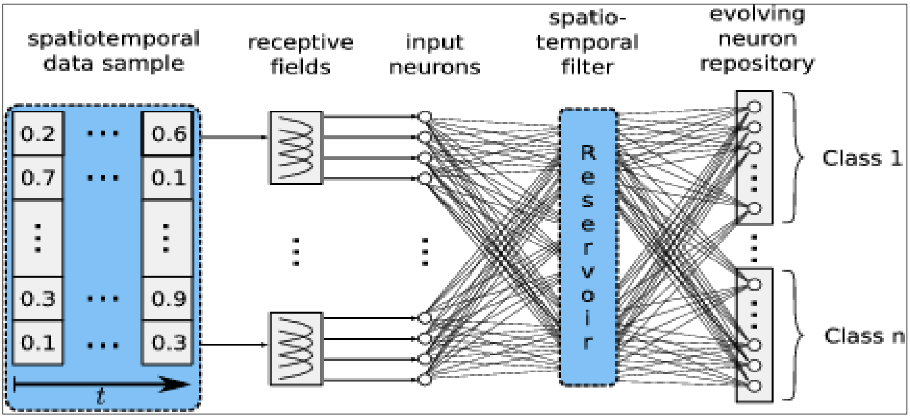
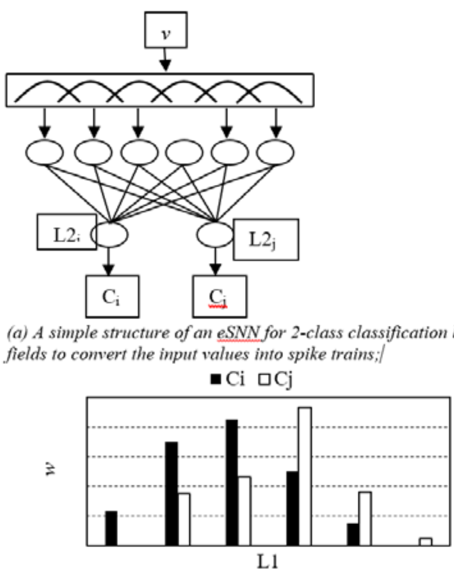


Fig. 5 A reservoir-based eSNN for spatio-temporal data classification

Extracting fuzzy rules from an eSNN would make the eSNN not only efficient learning models, but also knowledge-based models. A method was proposed [81] and illustrated in Figure 6a and Figure 6b. Based on the connection weights (W) between the receptive field layer (L1) and the class output neuron layer (L2), the following fuzzy rules can be extracted:

$$\textit{IF (input variable v is SMALL) THEN class } C_i;$$
$$\textit{IF(v is LARGE) THEN class } C_j \quad (3)$$



(a) A simple structure of an eSNN for 2-class classification based on one input variable using 6 receptive fields to convert the input values into spike trains;

(b) The connection weights of the connections to class C_i and C_j output neurons respectively are interpreted as fuzzy rules.

Fig. 6 Knowledge extraction from evolving spiking neural networks

The eSNN use spike information representation, spiking neuron models, and spike learning and encoding rules, and the structure is evolving to capture spatio-temporal relationship from data.

3.2 Applications of eSNN for AI

Numerous applications based on different eSNN models have been reported, among them:

- Advanced spiking neural network technologies for neurorehabilitation [82];
- Object movement recognition [83];
- Multimodal audio and visual information processing [84];
- Ecological data modelling and prediction of the establishment of invasive species [85];
- Integrated brain data analysis [86];
- Predictive modelling method and case study on personalized stroke occurrence prediction [87].

3.3 Quantum inspired optimization of eSNN

eSNN have several parameters that need to be optimized for an optimal performance. Several successful methods have been proposed for this purpose, among them are: Quantum-inspired evolutionary algorithm (QiEA) [88], and Quantum-inspired particle swarm optimization method (QiPSO) [89].

Quantum inspired optimization methods use the principle of superposition of states to represent and optimize features (input variables) and parameters of the eSNN [17]. Features and parameters are represented as qubits, which are 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 value of 1 or 0.

3.4 Neuromorphic implementations of SNN

Using SNN neuromorphic computational models can be developed. Opposite to the traditional von Neumann computational architecture, where memory, control and ALU are separated, in neuromorphic models all these modules are integrated together as they are in the brain.

To make the implementation of SNN models more efficient, specialized neuromorphic hardware has been developed, including:

- A hardware model of an integrate-and-fire neuron [90];
- A silicon retina [91];
- INI Zürich SNN chips [92, 93];
- IBM True North [94]. The system enables parallel processing of 1mln spiking neurons and 1 billion synapses;
- DVS and silicon cochlea (ETH, Zurich);
- Stanford NeuroGrid [95]. The system has 1 million neurons on a board, 6 3 billion connections, and is realized as hybrid analogue/digital circuits;
- SpiNNaker [96]. The system is a general-purpose, scalable, multichip multicore platform for real-time massively parallel simulations of large scale SNN.

The neuromorphic platforms are characterized by massive parallelism, high speed and low power consumption. For their efficient application, they require the development of SNN computational models for learning from data.

4 Deep learning neural networks and brain-like AI machines. NeuCube

4.1 Deep learning neural networks (DNN)

Deep learning neural networks (DNN) are ANN that have several layers of neurons and connections in their structures (rather than 3 as shown in Figure 1b). A class of DNN is the convolutional DNN, where neurons at the first layer learn features only within a small subsection of the input vector data (e.g., a small square of pixels from an image). These neurons are connected to the next layer where features are combined, until the output classification layer, where output classes are determined. An example is shown in Figure 7.

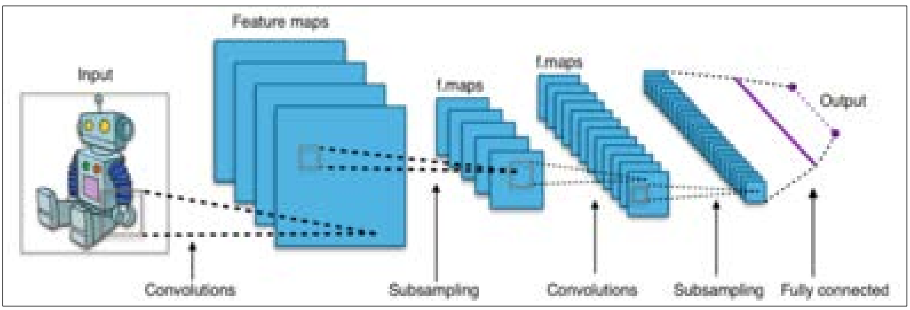


Fig.7 An example of a convolutional DNN (from https://en.wikipedia.org/wiki/Convolutional_neural_network)

DNNs are excellent for vector- or frame-based data, but not much for temporal (or spatio-/spectro-temporal data). There is no *time of asynchronous events* learned in the model. They are difficult to adapt to new data and the structures are not flexible.

4.2 Brain-like AI machines. NeuCube.

Inspired by the deep learning in the brain, a deep learning machine was developed, named NeuCube [86]. It was initially designed for spatio-temporal brain data modelling, but then it was also used for climate data modelling and stroke occurrence prediction and other applications [87].

The NeuCube framework is depicted in Figure 8. It consists of the following modules:

- Input information encoding module;
- 3D SNN reservoir module (SNNr);
- Output (e.g. classification) module;
- Gene regulatory network (GRN) module (optional);
- Optimization module (optional).

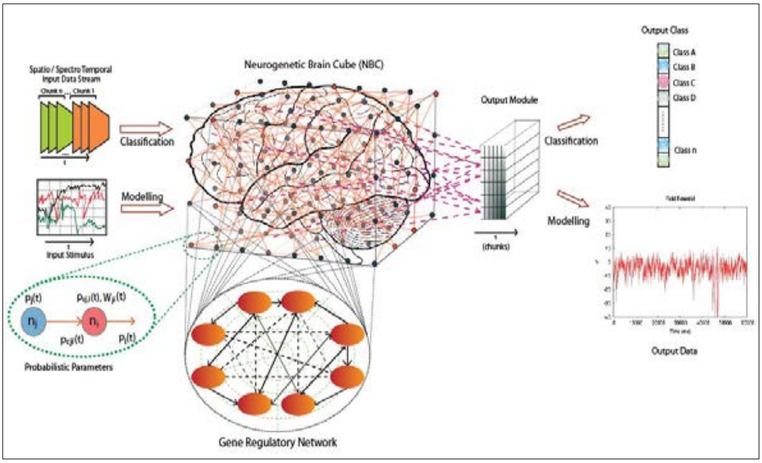


Fig. 8 A block diagram of the NeuCube deep learning machine (from [86])

The input module transforms input data into trains of spikes. Spatio-temporal data (such as EEG, climate, cybersecurity, financial, etc.) is entered after the encoding into the main module – the 3D SNN reservoir (SNNr). Input data is entered into *pre-designated spatially distributed* areas of the SNNr that correspond to the spatial location in the origin where data was collected (if there is such).

Learning in the SNN is performed in two stages:

- Unsupervised training, where spatio-temporal data is entered into relevant areas of the SNNr over time. Unsupervised learning is performed to modify the initially set connection weights. The SNNr will learn to activate the same groups of spiking neurons when similar input stimuli are presented, also known as a *polychronization* effect [76].
- Supervised training of the spiking neurons in the output classification 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 SNNr 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 SNNr, including the deSNN [79] and SPAN models [97]. The latter is suitable for generating motor control spike trains in response to certain patterns of activity of the SNNr.

Memory in the NeuCube architecture is represented as a combination of the three types of memory described below, which are mutually interacting:

- Short-term memory, represented as changes of the PSP and temporary changes of synaptic efficacy;
- Long-term memory, represented as a stable establishment of synaptic efficacy – long-term potentiation (LTP) and long-term depression (LTD);
- Genetic memory, represented as a genetic code.

In NeuCube, similar activation patterns (called "polychronous waves") can be generated in the SNNr 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 SNNr retains the connections as a long-term memory. Since the SNNr learns functional pathways of spiking activities represented as structural pathways of connections, when only a small initial part of input data is entered the SNNr 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 with a wide scope of applications.

4.3 Applications of NeuCube for AI

For a classification or a regression problem based on temporal- or spatio/spectro-temporal data, a NeuCube based system can be designed and implemented following the steps below:

(a) Input data transformation into spike sequences;
(b) Mapping input variables into spiking neurons;
(c) Unsupervised learning spatio-temporal spike sequences in a scalable 3D SNN reservoir;
(d) On-going learning and classification of data over time;
(d) Dynamic parameter optimization;
(e) Evaluating the time for predictive modelling;
(f) Adaptation on new data, possibly in an on-line/real time mode;
(g) Model visualization and interpretation for a better understanding of the data and the processes that generated it;
(h) Implementation of the SNN model as both software and a neuromorphic hardware system (if necessary).

A NeuCube development system, that allows for the above steps to be explored for a final design of an efficient application system, is available from: http://www.kedri.aut.ac.nz/neucube/. Several applications of NeuCube are described in [98]. In [99], a method for using SNN for efficient data compression is introduced, with wide range of applications in telecommunication. In [100] a survey of applications using NeuCube SNN machine can be found.

5 Conclusion

This paper presents briefly methods of artificial neural networks (ANN) and directs users which method to use depending on the data and the problem in hand. According to the type of the data, these applications can be classified as:

- Static (vector-based) data;
- Temporal data (e.g. climate, financial);
- Spatio-temporal data with fixed spatial location of variables (e.g., cybersecurity, brain data);

- Spatio-temporal data with changing locations of the spatial variables (e.g., moving objects);
- Spectro-temporal data (e.g. radio-astronomy, audio).

According to the characteristics of the data, the applications can be:

- Sparse features/low frequency (e.g. climate data, ecological data, multisensory data);
- Sparse features/high frequency (e.g. EEG brain signals, seismic data related to earthquakes);
- Dense features/low frequency (e.g. financial data);
- Dense features/high frequency (e.g. radio-astronomy data).

The paper presents first principles of various ANN and then surveys their use in practical systems. The paper specifically addresses the problem of designing systems for adaptive learning and knowledge discovery from complex data, emphasizing on one particular direction named ECOS. Different types of ANN and ECOS methods in particular are suitable for different applications as discussed in the paper. The paper advocates that integrating principles derived from neural networks, fuzzy systems, evolutionary computation, quantum computing and brain information processing, could lead to more efficient information technologies [100].

Further use of SNN is anticipated in several directions:

- Spatio-temporal data compression [101];
- Security systems [102, 103];
- Personal assistance [104,105]
- Automated financial, banking and trading systems [106].

Acknowledgements The work on this paper is supported by the Knowledge Engineering and Discovery Research Institute (KEDRI, http://www.kedri.aut.ac.nz) under a SRIFT AUT grant. I would like to thank Prof. Ravi and the editors of the journal for inviting me to submit this paper.

References

1. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychology Review. **65,** 386-402 (1958)
2. Amari, S.: A theory of adaptive pattern classifiers. IEEE Transactions on Electronic Computers. **3** (vol.EC-16), pp. 299-307 (1967)
3. Amari, S.: Mathematical foundations of neurocomputing. Proceedings of the IEEE. **9** (vol.78), pp. 1443-1463 (1990)
4. Rumelhart, D., McLelland, J. (eds): Parallel and distributed processing. MIT Press, 1986
5. Werbos, P. Backpropagation through time: Proc. of the IEEE. 87:10 (1990)
6. Zadeh, L.A.: Fuzzy sets. Information and Control. **3** (vol.8), pp. 338-353 (1965)
7. Zadeh, L. A.: Fuzzy logic. IEEE Computer, **4** (vol.21), pp. 83-93 (1988)
8. Kasabov, N. K.: Foundations of neural networks, fuzzy systems and knowledge engineering. MIT Press, Cambridge, Massachusetts, USA (1996)
9. Kasabov, N. K., Shishkov, S. I.: A connectionist production system with partial match and its use for approximate reasoning. Connection Science. **3-4,** 5, pp. 275-305 (1993)

10. Yamakawa, T., Uchino, E., Miki, T., Kusanagi, H.: A neo fuzzy neuron and its application to system identification and prediction of the system behavior. Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks. Japan, pp. 477-483 (July 1992)
11. Furuhashi, T., Hasegawa, T., Horikawa, S., Uchikawa, Y.: An adaptive fuzzy controller using fuzzy neural networks. Proc. of 5th Int'l Fuzzy System Association World Congress. Korea, pp.769-772 (July 1993)
12. Kasabov, N. K., Kim, J. S., Watts, M. J., Gray, A. R.: FuNN/2 – A fuzzy neural network architecture for adaptive learning and knowledge acquisition. Information Sciences – Applications. **3-4** (vol.101), pp. 155-175 (1997)
13. Kasabov, N. K.: Evolving fuzzy neural networks – algorithms, applications and biological motivation. In: Yamakawa, T. and Matsumoto, G. (Eds.). Methodologies for the Conception, Design and Application of Soft Computing, World Scientific, pp 271 – 274 (1998)
14. Kasabov, N. K.: Evolving fuzzy neural networks for on-line supervised/unsupervised. Knowledge–Based Learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, **6** (vol.31), 2001, pp. 902-918
15. Kasabov, N. K.: Evolving connectionist systems: methods and applications in bioinformatics, brain study and intelligent machines. Springer-Verlag, London, England, 2002
16. Kasabov, N. K. Song, Q.: DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Transactions on Fuzzy Systems, **2** (vol.10), 2002, pp. 144-154
17. Kasabov, N. K.: Evolving connectionist systems: the knowledge engineering approach (2nd ed.), Springer-Verlag, London, England (2007)
18. Futschik, M. E., Kasabov, N. K.: Fuzzy clustering in gene expression data analysis. Proc. of the IEEE Int'l Conf. on Fuzzy Systems, USA. pp. 414-419 (2002)
19. Watts, M. J.: A decade of kasabov's evolving connectionist systems: a review. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews. 3 (vol.39), 2009, pp. 253-269
20. Bezdek, J. C.: Analysis of fuzzy information. CRC Press, Boca Raton, Florida, USA (1987)
21. Yager, R. R., Filev, D. P.: Generation of fuzzy rules by mountain clustering. Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology. **3** (vol.2), pp. 209-219 (1994)
22. Song, Q., Kasabov, N. K.: TWNFI – A transductive neuro-fuzzy inference system with weighted data normalization for personalized modelling. Neural Networks, 10 (vol.19), 2006, pp. 1591-1596
23. Kasabov, N. K., Hu, Y.: Integrated optimisation method for personalised modelling and case study applications. Int'l Journal of Functional Informatics and Personalised Medicine. 3 (vol.3), pp. 236-256 (2010)
24. Deng, D., Kasabov N. K.: On-line pattern analysis by evolving self-organizing maps. Neurocomputing. (vol.51), pp. 87-103 (2003)
25. Song Q. and N. Kasabov, NFI: A neuro-fuzzy inference method for transductive reasoning. IEEE Trans. on Fuzzy Systems. **13,** 6, 799-808 (2005)
26. Ozawa, S., S.Too, S.Abe, S. Pang and N. Kasabov: Incremental learning of feature space and classifier for online face recognition. Neural Networks, 575-584 (August 2005)
27. Chan Z.and N.Kasabov: Evolutionary computation for on-line and off-line parameter tuning of evolving fuzzy neural networks. Int. J. of Computational Intelligence and Applications. Imperial College Press, **4**, 3, 309-319 (2004)
28. de Moraes, R. M.: Assessment of EFuNN accuracy for pattern recognition using data with different statistical distributions. Proc. of the 2nd Brazilian Congress on Fuzzy Systems. Brazil, pp. 672-685 (November 2012)
29. Dovžan, D., Škrjanc, I.: Recursive clustering based on a gustafson–kessel algorithm. Evolving Systems. **1** (vol.2), pp. 15-24 (2011)
30. Tonelli, P., Mouret, J.-B.: Using a map-based encoding to evolve plastic neural networks. Proc. of the IEEE Workshop on Evolving and Adaptive Intelligent Systems. France, pp. 9-16 (April 2011)
31. Kalhor, A., Araabi, B. N., Lucas, C.: Evolving takagi–sugeno fuzzy model based on switching to neighboring models. Applied Soft Computing, **2** (vol.13), pp. 939-946 (2013)
32. Vajpai, J., Arun, J. B.: A soft computing based approach for modeling of chaotic time series. Proc. of the 13th Int'l Conf. on Neural Information Processing. China. pp. 505-512 (October 2006)
33. Bordignon, F., Gomide, F.: Uninorm based evolving neural networks and approximation capabilities. Neurocomputing, (vol.127), pp. 13-20 (2014)
34. Kasabov, N.: Global, local and personalised modelling and profile discovery in bioinformatics: An integrated approach. Pattern Recognition Letters. **28**, 6, 673-685 (2007)
35. Lughofer, E. D.: FLEXFIS: A robust incremental learning approach for evolving takagi–sugeno fuzzy models. IEEE Transactions on Fuzzy Systems. **6** (vol.16), pp. 1393-1410 (2008)

36. Angelov, P. P., Lughofer, E. D., Zhou, X.: Evolving fuzzy classifiers using different model architectures. Fuzzy Sets and Systems. **23** (vol.159), pp. 3160-3182 (2008)
37. Ang, K. K., Quek, C.: RSPOP: Rough set-based pseudo outer-product fuzzy rule identification algorithm. Neural Computation. **1** (vol.17), pp. 205-243 (2005)
38. de Jesús Rubio, J.: SOFMLS: Online self-organizing fuzzy modified least-squares network. IEEE Transactions on Fuzzy Systems, **6** (vol.17), pp. 1296-1309 (2009)
39. Huang, G. B., Liang, N. Y., Rong, H. J.: On-line sequential extreme learning machine. Proc. of the IASTED Int'l Conf. on Computational Intelligence. Canada, pp. 232-237 (July 2005)
40. Lim, J. S.: Finding features for real-time premature ventricular contraction detection using a fuzzy neural network system. IEEE Transactions on Neural Networks. **3** (vol.20), pp. 522-527 (2009)
41. Angelov, P. P., Zhou, X., Klawonn, F.: Evolving fuzzy rule-based classifiers. Proc. of the IEEE Symposium on Computational Intelligence in Image and Signal Processing. USA, pp. 220-225 (April 2007)
42. Liu, F., Quek, C., Ng, G. S.: A novel generic hebbian ordering-based fuzzy rule base reduction approach to mamdani neuro-fuzzy system. Neural Computation. **6** (vol.19), pp. 1656-1680 (2007)
43. Song, H., Miao, C., Roel, W. Shen, Z., Catthoor, F.: Implementation of fuzzy cognitive maps based on fuzzy neural network and application in prediction of time series. IEEE Transactions on Fuzzy Systems. **2** (vol.18), pp. 233-250 (2010)
44. de Jesús Rubio, J., Vázquez, D. M., Pacheco, J.: Backpropagation to train an evolving radial basis function neural network. evolving systems. **3** (vol.1), pp. 173-180 (2010)
45. Aznarte, J. L., Benítez, J. M., Castro, J. L.: Smooth transition autoregressive models and fuzzy rule-based systems: functional equivalence and consequences. fuzzy sets and systems. **24** (vol.158), pp. 2734-2745 (2007)
46. Cetisli, B. Development of an Adaptive Neuro-Fuzzy Classifier Using Linguistic Hedges. Expert Systems with Applications. 8 (vol.37), pp. 6093-6101 (2010)
47. Subramanian, K., Suresh, S.: A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system. applied soft computing. 11 (vol.12), pp. 3603-3614 (2012)
48. Babu, G. S., Suresh, S. Meta-cognitive RBF network and its projection based learning algorithm for classification problems. Applied Soft Computing. **1** (vol.13), pp. 654-666 (2013)
49. Tung, S. W., Quek, C., Guan, C. SaFIN: A self-adaptive fuzzy inference network. IEEE Transactions on Neural Networks. **12** (vol.22), pp. 1928-1940 (2011)
50. Suresh, S., Subramanian, K.: A sequential learning algorithm for meta-cognitive neuro-fuzzy inference system for classification problems. Proc. of the Int'l Joint Conf. on Neural Networks. USA, pp. 2507-2512 (August 2011)
51. Kadlec, P., Gabrys, B.: Architecture for development of adaptive on-line prediction models. memetic computing. **4** (vol.1), pp. 241-269 (2009)
52. Minku, F. L., Ludemir, T. B.: Clustering and co-evolution to construct neural network ensembles: an experimental study. Neural Networks. **9** (vol.21), pp. 1363-1379 (2008)
53. Filev, D. P., Angelov, P. P.: Algorithms for real-time clustering and generation of rules from data. In: Valente di Oliveira, J., and Pedrycz, W. (Eds.). Advances in Fuzzy Clustering and its Applications. John Wiley & Sons, Chichester, UK, (2007)
54. Amadou Boubacar, H., Lecoeuche, S., Maouche, S.: SAKM: Self-adaptive kernel machine: a kernel-based algorithm for online clustering. Neural Networks. **9** (vol.21), pp. 1287-1301 (2008)
55. Tan, J., Quek, C.: A BCM theory of meta-plasticity for online self-reorganizing fuzzy-associative learning. ieee transactions on neural networks. **6** (vol.21), pp. 985-1003 (2010)
56. Minku, F. L., Ludermir, T. B.: Evolutionary strategies and genetic algorithms for dynamic parameter optimization of evolving fuzzy neural networks. Proc. of the IEEE Congress on Evolutionary Computation. Scotland, pp. 1951-1958 (2005)
57. Yamauchi, K., Hayami, J.: Incremental leaning and model selection for radial basis function network through sleep. IEICE Transactions on Information and Systems. **4** (vol.e90-d), pp. 722-735 (2007)
58. Leite, D. F., Costa, P., Gomide, F.: Interval-based evolving modeling. Proc. of the IEEE Workshop on Evolving and Self-Developing Intelligent Systems. USA, pp. 1-8 (March 2009)
59. Leite, D. F., Costa, P., Gomide, F.: Evolving granular neural networks from fuzzy data streams. Neural Networks. (vol.38), pp. 1-16 (2013)
60. de Jesús Rubio, J.: Stability analysis for an online evolving neuro-fuzzy recurrent network. In: Angelov, P. P., Filev, D. P., Kasabov, N. K. (Eds.). Evolving Intelligent Systems: Methodology and Applications. John Wiley & Sons, Hoboken, New Jersey, USA (2010)
61. Kim, K., Whang, E. J., Park, C. W., Kim, E., Park, M.: A TSK Fuzzy inference algorithm for online identification. Proc. of the 2nd Int'l Conf. on Fuzzy Systems and Knowledge Discovery. China, pp. 179-188 (August 2005)

62. Zanchettin, C., Minku, L. L., Ludermir, T. B.: Design of experiments in neuro-fuzzy systems. international journal of computational intelligence and applications. **2** (vol.9), pp. 137-152 (2010)
63. Minku, F. L., Ludermir, T. B.: EFuNNs ensembles construction using a clustering method and a coevolutionary genetic algorithm. Proc. of the IEEE Congress on Evolutionary Computation. Canada, pp. 1399-1406 (July 2006)
64. Tung, S. W., Quek, C., Guan, C. eT2FIS: An evolving type-2 neural fuzzy inference system. Information Sciences. (vol.220), pp. 124-148 (2013)
65. O'Hara, B., Perera, J., Brabazon A.: Designing radial basis function networks for classification using differential evolution. Proc. of the Int'l Joint Conf. on Neural Networks. Canada, pp. 2932-2937 (July 2006)
66. Subramanian, K., Sundaram, S., Sundararajan, N.: A metacognitive neuro-fuzzy inference system (McFIS) for sequential classification problems. IEEE Transactions on Fuzzy Systems. **6** (vol.21), pp. 1080-1095 (2013)
67. Hernández, J. A. M., Castañeda, F. G., Cadenas, J. A. M.: An evolving fuzzy neural network based on the mapping of similarities. IEEE Transactions on Fuzzy Systems. **6** (vol.17), pp. 1379-1396 (2009)
68. Zhao, Q. L., Jiang, Y. H., Xu, M.: Incremental learning by heterogeneous bagging ensemble. Proc. of the Int'l Conf. on Advanced Data Mining and Applications. China, pp. 1-12 (November 2010)
69. Goh, H., Lim, J. H., Quek, C.: Fuzzy associative conjuncted maps network. IEEE Transactions on Neural Networks. **8** (vol.20), pp. 1302-1319 (2009)
70. Minku, F. L., Ludermir, T. B.: EFuNN ensembles construction using cone with multi-objective GA. Proc. of the 9th Brazilian Symposium on Neural Networks. Brazil, pp. 48-53 (October 2006)
71. Kasabov, N.: Adaptation and interaction in dynamical systems: modelling and rule discovery through evolving connectionist systems. Applied Soft Computing. **6,** 3, 307-322 (2006)
72. H.Widiputra, R. Pears, N. Kasabov: Dynamic interaction network versus localized trends model for multiple time-series prediction. Cybernetics and Systems. **42,** 2, 100-123 (2011)
73. Ghobakhlou A., M. Watts and N. Kasabov: Adaptive speech recognition with evolving connectionist systems. Information Sciences. **156,** 71-83 (2003)
74. Hodgkin, A. L., Huxley, A. F.: A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve. Journal of Physiology. **4** (vol.117), pp. 500-544 (1952)
75. Hopfield, J. J.: Pattern recognition computation using action potential timing for stimulus representation. Nature, **6535** (vol.376), pp. 33-36 (1995)
76. Izhikevich, E. M.: Which model to use for cortical spiking neurons? IEEE Transactions on Neural Networks. **5** (vol.15), pp. 1063-1070 (2004)
77. Thorpe, S., Delorme, A., van Rullen, R.: Spike-based strategies for rapid processing. Neural Networks. **6-7** (vol.14), pp. 715-725 (2001)
78. Verstraeten, D., Schrauwen, B., D'Haene, M., Stroobandt, D.: An experimental unification of reservoir computing methods. Neural Networks. **3** (vol.20), pp. 391-403 (2007)
79. Kasabov, N. K., Dhoble, K., Nuntalid, N., Indiveri, G.: Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition. Neural Networks. (vol.41), pp. 188-201 (2013)
80. Song, S., Miller, K. D., Abbott, L. F.: Competitive hebbian learning through spike-timing-dependent synaptic plasticity. Nature Neuroscience. **9** (vol.3), pp. 919–926 (2000)
81. Soltic, S., Kasabov, N. K.: Knowledge extraction from evolving spiking neural networks with rank order population coding. International Journal of Neural Systems. 6 (vol.20), pp. 437-445 (2010)
82. Chen, Y., Hu, J., Kasabov, N. K., Hou, Z., Cheng, L.: NeuroCubeRehab: A pilot study for eeg classification in rehabilitation practice based on spiking neural networks. Proc. of the 20th Int'l Conf. on Neural Information Processing. South Korea, pp.70-77 a(November 2013)
83. EU FP7 Marie Curie EvoSpike Project. INI/ETH/UZH, http://ncs.ethz.ch/projects/evospike (2011-2012)
84. Wysoski, S. G., Benuskova, L., Kasabov, N. K.: Evolving spiking neural networks for audiovisual information processing. Neural Networks. **7** (vol.23), pp. 819-835 (2010)
85. Schliebs, S., Defoin-Platel, M., Worner, S., Kasabov, N. K.: Integrated feature and parameter optimization for evolving spiking neural networks: exploring heterogeneous probabilistic models. Neural Networks. **5-6** (vol.22), pp. 623-632 (2009)
86. Kasabov, N. K.: NeuCube: A Spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. neural networks. (vol.52), pp. 62-76 (2014)
87. Kasabov, N. K., Feigin, V. L., Hou, Z.-G., Chen, Y., Liang, L., Krishnamurthi, R., Othman, M., Parmar, P.: Evolving spiking neural networks for personalized modelling of spatio-temporal data and early prediction of events: a case study on stroke. neurocomputing. (vol.134), pp. 269-279 (2014)

88. Defoin-Platel, M., Schliebs, S., Kasabov, N. K.: Quantum-inspired evolutionary algorithm: a multimodel eda. ieee transactions on evolutionary computation. **6** (vol.13), pp. 1218-1232 (2009)
89. Hamed, H. N. A., Kasabov, N. K., Shamsuddin, S.: Probabilistic evolving spiking neural network optimization using dynamic quantum inspired particle swarm optimization. Australian Journal of Intelligent Information Processing Systems, **1** (vol.11), pp. 23-28 (2010)
90. Mead, C.: Analog VLSI and neural systems: Addison-Wesley Longman Publishing Co, Boston, Massachusetts, USA (1989)
91. *Mahowald, M., Mead, C.: The silicon retina. Scientific American.* ***264*** *(5), 76–82 (1991)*
92. Corradi, F. and Indiveri, G.: A neuromorphic event-based neural recording system for smart brain-machine-interfaces, biomedical circuits and systems. IEEE Transactions. **9:(5)** 699 - 709 (2015)
93. Binas, J. and Rutishauser, U. and Indiveri, G. and Pfeiffer, M.: Learning and stabilization of winner-take-all dynamics through interacting excitatory and inhibitory plasticity. Frontiers in Computational Neuroscience. **8:(68)** (2014)
94. Sawada, J., et al.: Truenorth ecosystem for brain-inspired computing: scalable systems, software, and applications. proc. of the int'l conf. for high performance computing, Networking, Storage and Analysis. USA, pp. 130-141 (November 2016)
95. Benjamin, B. V., et al.: Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. Proceedings of the IEEE. 5 (vol.102), pp. 699-716 (2014)
96. Furber, S. B., Lester, D. R., Plana, L. A., Garside, J. D., Painkras, E., Temple, S., Brown, A. D.: Overview of the spinnaker system architecture. IEEE Transactions on Computers. 12 (vol. 62), pp. 2454-2467 (2012)
97. Mohemmed, A., Schliebs, S., Matsuda, S., Kasabov, N. K.: Training spiking neural networks to associate spatio-temporal input–output spike patterns. Neurocomputing. (vol.107), pp. 3-10 (2013)
98. Kasabov, N. K., Scott, N., Tu, E., Marks, S., Sengupta, N., Capecci, E., Othman, M., Doborjeh, M., Murli, N., Hartono, R., Espinosa-Ramos, J. I., Zhou, L., Alvi, F., Wang, G., Taylor, D., Feigin, V. L., Gulyaev, S., Mahmoudh, M., Hou, Z.-G., Yang, J.: Design methodology and selected applications of evolving spatio-temporal data machines in the neucube neuromorphic framework. Neural Networks. (vol.78), pp. 1-14 (2016)
99. Sengupta, N., Kasabov, N. K.: Spike-time encoding as a data compression technique for pattern recognition of temporal data. Information Sciences. (vol.406–407), pp. 133-145 (2017)
100. Kasabov, N. (ed): The Springer Handbook of Bio- and Neuroinformatics. Springer pp. 1230 (2014)
101. N. Sengupta, N. Kasabov: Spike-time encoding as a data compression technique for pattern recognition of temporal data. Information Sciences. 406–407, pp. 133–145 (2017)
102. Alvi,F., R. Pears, N. Kasabov: An evolving spatio-temporal approach for gender and age group classification with spiking neural networks, evolving systems, Springer (2017)
103. Kasabov, N.: From multilayer perceptrons and neuro-fuzzy systems to deep learning machines: which method to use? – A survey. Int. Journal on Information Technologies and Security. vol.**9**, No. 20, pp. 3-24 (2017)
104. Kasabov, N., M. Doborjeh, Z. Doborjeh: Mapping, learning, visualisation, classification and understanding of fMRI data in the neucube spatio temporal data machine. IEEE Transactions of Neural Networks and Learning Systems. vol. **28**,4, 887-899, DOI: 10.1109/TNNLS.2016.2612890 (2017)
105. Ge, C., N Kasabov, Z Liu, J Yang: A spiking neural network model for obstacle avoidance in simulated prosthetic vision, Information Sciences **399**, 30-42 (2017)
106. Khansama, R., V. Ravi, N. Sengupta, A.R. Gollahalli and N. Kasabov: Stock market movement prediction using evolving spiking neural networks, IEEE TNNLS, submitted 2017