

# Effects of synaptic connectivity on liquid state machine performance

Han Ju<sup>a,c</sup>, Jian-Xin Xu<sup>b</sup>, Edmund Chong<sup>a</sup>, Antonius M.J. VanDongen<sup>a,\*</sup>

<sup>a</sup> Program for Neuroscience and Behavioral Disorders, Duke–NUS Graduate Medical School, Singapore

<sup>b</sup> Department of Electrical and Computer Engineering, National University of Singapore, Singapore

<sup>c</sup> Graduate School for Integrative Science and Engineering, National University of Singapore, Singapore

## ARTICLE INFO

### Article history:

Received 23 March 2012

Received in revised form 26 September 2012

Accepted 6 November 2012

### Keywords:

Liquid state machine

Genetic algorithm

Neural microcircuit optimization

Spatiotemporal pattern classification

## ABSTRACT

The Liquid State Machine (LSM) is a biologically plausible computational neural network model for real-time computing on time-varying inputs, whose structure and function were inspired by the properties of neocortical columns in the central nervous system of mammals. The LSM uses spiking neurons connected by dynamic synapses to project inputs into a high dimensional feature space, allowing classification of inputs by linear separation, similar to the approach used in support vector machines (SVMs). The performance of a LSM neural network model on pattern recognition tasks mainly depends on its parameter settings. Two parameters are of particular interest: the distribution of synaptic strengths and synaptic connectivity. To design an efficient liquid filter that performs desired kernel functions, these parameters need to be optimized. We have studied performance as a function of these parameters for several models of synaptic connectivity. The results show that in order to achieve good performance, large synaptic weights are required to compensate for a small number of synapses in the liquid filter, and *vice versa*. In addition, a larger variance of the synaptic weights results in better performance for LSM benchmark problems. We also propose a genetic algorithm-based approach to evolve the liquid filter from a minimum structure with no connections, to an optimized kernel with a minimal number of synapses and high classification accuracy. This approach facilitates the design of an optimal LSM with reduced computational complexity. Results obtained using this genetic programming approach show that the synaptic weight distribution after evolution is similar in shape to that found in cortical circuitry.

© 2012 Elsevier Ltd. All rights reserved.

## 1. Introduction

The neuronal wiring pattern in the human brain is one of the most remarkable products of biological evolution. The synaptic connectivity in the brain has been continuously and gradually refined by natural selection, and finally evolved to possess extraordinary computational power. The human brain not only can memorize experiences, learn skills, and create ideas, but it is also a superior pattern classifier that can process multi-modal information in real-time.

The hippocampus, a brain region critical for learning and memory processes, has been reported to possess pattern separation functionality similar to the Support Vector Machine (SVM) (Baker, 2003; Bakker, Kirwan, Miller, & Stark, 2008), a popular machine classifier. The cerebellum has been modeled based on similar principles (Yamazaki & Tanaka, 2007). The SVM is a kernel ‘machine’ that nonlinearly transforms input data into high dimensional feature space, where accurate linear classification can be obtained

by drawing a hyperplane. This kernel method is quite popular in pattern recognition. The Liquid State Machine (LSM) (Maass, Natschläger, & Markram, 2002) is a biologically plausible neural network model inspired by the structural and functional organization of the mammalian neocortex. It uses a kernel approach similar to the SVM. The kernel part (the ‘liquid filter’) is an artificial spiking neural network consisting of hundreds of neurons and thousands of synaptic connections, whose model parameters are set to mimic properties measured in real cortical neurons and synapses. The input neurons inject spike train stimuli into the liquid filter, and a readout neuron that is connected to all the kernel neurons can be trained to perform classification tasks. LSMs have been applied to many applications, including word recognition (Verstraeten, Schrauwen, Stroobandt, & Van Campenhout, 2005), real-time speech recognition (Schrauwen, D’Haene, Verstraeten, & Campenhout, 2008) and robotics (Joshi & Maass, 2004), and its performance is comparable to state-of-the-art recognition systems.

The traditional sigmoidal recurrent neural networks (RNNs) have a fully-connected structure. A fully-connected network can be reduced to a partially-connected version by setting certain synaptic weights to zero, but such networks suffer from high computational complexity if the number of neurons is large, because the number of connections increases exponentially with

\* Correspondence to: Program for Neuroscience and Behavioral Disorders, 8 College Road, Singapore 169857, Singapore. Tel.: +65 6515 7075.

E-mail address: [antonius.vandongen@duke-nus.edu.sg](mailto:antonius.vandongen@duke-nus.edu.sg) (A.M.J. VanDongen).

the number of neurons. The LSM consists of a partially connected spiking neural network containing hundreds of neurons. In the proposed formalism, the connections in the liquid filter are initialized at random, with random synaptic weights, which do not change during training. The model parameters that determine the network connectivity and the distribution of synaptic weights are critical determinants of performance of the liquid filter. The LSM is a biologically realistic model, which suggests that the pattern of neuronal wiring in brain networks and the topology of synaptic connections could be taken into consideration when constructing the LSM kernel.

A well-studied paradigm for network connectivity is the small-world topology (Watts & Strogatz, 1998), in which nodes (neurons) are highly clustered, and yet the minimum distance between any two randomly chosen nodes (the number of synapses connecting the neurons) is short. Small-world architectures are common in biological neuronal networks. It has been shown that neuronal networks in the worm *C. elegans* have small-world properties (Amaral, Scala, Barthelemy, & Stanley, 2000). Simulations using cat and macaque brain connectivity data (Kaiser, Martin, Andras, & Young, 2007) have shown these networks to be scale-free, a property also found in small-world networks. For human brain networks, small-world properties have been shown from MEG (Stam, 2004), EEG (Micheloyannis et al., 2006), and fMRI data (Achard, Salvador, Whitcher, Suckling, & Bullmore, 2006). The small-world property is also important in neural network simulations. For a feed-forward network with sigmoidal neurons, small-world architectures produced the best learning rate and lowest learning error, compared to ordered or random networks (Simard, Nadeau, & Kröger, 2005). In networks build with Hodgkin and Huxley neurons, small-world topology is required for fast responses and coherent oscillations (Lago-Fernandez, Huerta, Corbacho, & Siguenza, 2000). It has also been suggested that small-world networks are optimal for information coding via poly-synchronization (Vertes & Duke, 2009). As the LSM is a 3D spiking neural network with a lamina-like structure, it is worthwhile to explore the effects of the small-world properties on the performance of LSMs.

In addition to small-world properties, the orientation of the synaptic connections in brain networks may also be important. If a neuron fires, an action potential will travel along the axon, distribute over the axonal branches, and reach the pre-synaptic terminals and boutons (*en passant*), causing transmitter release which excites or inhibits the post-synaptic cell. During brain development, axons tend to grow along a straight line until a guidance cue is encountered. As a result, much of the information flow in biological neuronal networks is not radial, but displays directionality. Models with directional connectivity have not yet been explored for LSMs.

A previous study by Verstraeten, Schrauwen, D'Haene, and Stroobandt (2007) investigated the relation between reservoir parameters and network dynamics with a focus on Echo State Networks (ESN), which is a computational framework similar in structure to the LSM, but built from analog neurons. ESN and LSM architectures both belong to the reservoir computing family. However, the relation between network parameters and performance is still poorly understood for the LSM. Various neuron models have been explored to boost performance of the LSM. It has been shown that compared to deterministic models, using a probabilistic neuron model (Kasabov, 2010) could offer potential advantages for both LSMs (Schliebs, Mohammed, & Kasabov, 2011) and spiking neural networks with a reservoir-like structure (Hamed, Kasabov, Shamsuddin, Widiputra, & Dhole, 2011). In this paper we focus on how LSM performance depends on parameters associated with synaptic connectivity, including network topology and synaptic efficacies. Several connectivity models are studied:

the original radial connection model proposed by Maass et al., small-world network topologies, and a directional axon growth model. The effects of both the connection topology and connection strength were studied. The main purpose of this paper is not to determine which model performs best, but rather to derive general rules and insights, which may facilitate optimal LSM design. More than 12,000 LSMs with different connection topologies were simulated and evaluated. Based on the results, we propose a method that uses genetic algorithms to evolve the liquid filter's connectivity to obtain a structure with high performance and low computational complexity. One of the LSM's main merits is its ability to perform classification in real-time. The complexity of the liquid filter directly affects the computation speed and the real-time performance. Thus, a minimum kernel structure is always desired.

## 2. Models

The simulations were implemented using MATLAB with the CSIM (a neural Circuit Simulator) package (Natschläger, Markram, & Maass, 2003).

### 2.1. Neuron model

A network of leaky integrate-and-fire (LIF) neurons is created as the liquid filter, with each neuron positioned at an integer point in a three dimensional space. 20% of the neurons in the liquid filter are inhibitory and 80% are excitatory. Each neuron is modeled by a linear differential equation:

$$\tau_m \frac{dV_m}{dt} = -(V_m - V_{resting}) + R_m(I_{syn} + I_{inject}) \quad (2.1)$$

where the parameters are: membrane time constant  $\tau_m = 30$  ms, membrane resistance  $R_m = 1$  M $\Omega$  and steady background current  $I_{inject} = 13.5$  pA. No random noise is added to the input current. For the first time step in the simulation, the membrane potential  $V_m$  was set to an initial value randomly selected between 13.5 and 15 mV. When  $V_m$  is larger than the threshold voltage 15 mV,  $V_m$  is reset to 13.5 mV for an absolute refractory period of 3 ms for excitatory neurons and 2 ms for inhibitory neurons (Joshi, 2007).

Input neurons receive and inject stimuli into the liquid filter through static spiking synapses with delays. Each input neuron is randomly connected to 10% of the neurons in the liquid filter, and is restricted to connect to excitatory neurons only.

### 2.2. Dynamic synapse

All the connections established between neurons in the liquid filter are dynamic synapses. Following the literature (Legenstein & Maass, 2007), the dynamic synapse model incorporates short-term depression and facilitation effects:

$$\begin{aligned} A_k &= w \cdot u_k \cdot R_k \\ u_k &= U + u_{k-1}(1 - U) \exp(-\Delta_{k-1}/F) \\ R_k &= 1 + (R_{k-1} - u_{k-1}R_{k-1} - 1) \exp(-\Delta_{k-1}/D) \end{aligned} \quad (2.2)$$

where  $w$  is the weight of the synapse,  $A_k$  is the amplitude of the post-synaptic current raised by the  $k$ th spike and  $\Delta_{k-1}$  is the time interval between the  $k-1$ th spike and the  $k$ th spike.  $u_k$  models the effects of facilitation and  $R_k$  models the effects of depression.  $D$  and  $F$  are the time constants for depression and facilitation respectively and  $U$  is the average probability of neurotransmitter release in the synapse. The initial values for  $u$  and  $R$ , describing the first spike, are set to  $u_1 = U$  and  $R_1 = 1$ .

Depending on whether the neurons are excitatory ( $E$ ) or inhibitory ( $I$ ), the values of  $U$ ,  $D$  and  $F$  are drawn from pre-defined

Gaussian distributions. According to the published synapse model (Joshi, 2007), the mean values of  $U$ ,  $D$ ,  $F$  (with  $D$ ,  $F$  in seconds) are 0.5, 1.1, 0.05 for connections from excitatory neurons to excitatory neurons ( $EE$ ), 0.05, 0.125, 1.2 for excitatory to inhibitory neurons ( $EI$ ), 0.25, 0.7, 0.02 ( $IE$ ), 0.32, 0.144, 0.06 ( $II$ ), respectively. The standard deviation of each of these parameters is chosen to be half of its mean.

Depending on whether a synapse is excitatory or inhibitory, its synaptic weight is either positive or negative. To ensure that no negative (positive) weights are generated for excitatory (inhibitory) synapses, the synaptic strength for each synapse follows a Gamma distribution. The mean for the distribution is set to  $W \times W_{scale}$ , where the parameter  $W$  is  $3 \times 10^{-8}$  ( $EE$ ),  $6 \times 10^{-8}$  ( $EI$ ),  $-1.9 \times 10^{-8}$  ( $IE$ ,  $II$ ) (Maass et al., 2002);  $W_{scale}$  is a scaling factor, which is one of the parameters that we will investigate in this paper. The standard deviation for the synaptic strength is chosen to be half of its mean, i.e. the coefficient of variation is 0.5.

The value of the post-synaptic current ( $I$ ) passing into the neuron at time  $t$  is modeled with exponential decay  $I = \exp(-t/\tau_s)$ , where  $\tau_s$  is 3 ms for excitatory synapses and 6 ms for inhibitory synapses. Information transmission is not instantaneous for chemical synapses: transmitter diffusion across the synaptic cleft causes a delay, which is set to 1.5 ms for connections between excitatory neurons ( $EE$ ), and 0.8 ms for all other connections ( $EI$ ,  $IE$ ,  $II$ ).

### 2.3. Readout neuron

A single readout neuron is connected to all the LIF neurons in the liquid filter, and it is trained to make classification decisions. Each LIF neuron in the liquid filter provides its final state value to the readout neuron, scaled by its synaptic weight. The final state value  $s_m^f(i)$  of the LIF neuron  $i$  with the input stimulus  $m$  is calculated based on the spikes that the neuron  $i$  has emitted:

$$s_m^f(i) = \sum_n \exp\left(-\frac{t_{sim} - t_i^n}{\tau}\right) \quad (2.3)$$

where  $\tau$  is a time constant set to 0.03 s,  $t_i^n$  is the time of the  $n$ th spike, and  $t_{sim}$  is the duration of simulation for each input stimulus.

Network training is done by finding a set of optimal weights  $W$  for the readout using Fisher's Linear Discriminant. The output of the readout in response to a stimulus  $m$  is:

$$O(m) = W^T \begin{bmatrix} s_m^f(1) \\ s_m^f(2) \\ \vdots \end{bmatrix} = W^T S(m). \quad (2.4)$$

## 3. Connection topologies

### 3.1. Original connection topology

In neuronal networks, the probability of finding a connection between two neurons decreases exponentially with distance. A possible explanation of such connection mechanism is that axons tend to grow along the direction with a high concentration of axon guidance molecules. The concentration of the molecules decays exponentially with distance, and thus, neurons closer to the source of the molecules will have a higher probability to detect the signal (Kaiser, Hilgetag, & van Ooyen, 2009; Yamamoto, Tamada, & Murakami, 2002). Synaptic connections in the original LSM paper (Maass et al., 2002) are initialized according to the Euclidean distance between pre- and post-synaptic neurons. The probability of creating a connection between two neurons is calculated by the following equation:

$$p = C \cdot \exp\left[-\left(\frac{D(a, b)}{\lambda}\right)^2\right] \quad (3.1)$$

where  $\lambda$  is a connection parameter, and  $D(a, b)$  is the Euclidean distance between neurons  $a$  and  $b$ . We will refer to the above connection model as the “lambda model”. In this study, depending on whether neurons are inhibitory ( $I$ ) or excitatory ( $E$ ),  $C$  was set at 0.3 ( $EE$ ), 0.2 ( $EI$ ), 0.4 ( $IE$ ), or 0.1 ( $II$ ), respectively. These values are taken from LSM models used in previous studies (Maass et al., 2002) and are based on measurements of synaptic properties in cortical brain areas (Gupta, Wang, & Markram, 2000). Note that by using this equation, the connection range for each neuron has a sphere shape, i.e. there is no directional preference.

### 3.2. Small world networks

A small world network (Watts & Strogatz, 1998) is a type of graph that has two properties: (i) nodes (neurons) are highly clustered compared to a random graph, and (ii) a short path length exists between any two nodes in the network. It has been shown that many real world networks are neither completely ordered nor purely random, but instead display small-world properties. A small-world network can be obtained by randomly rewiring the connections in a network with a lattice structure. There is a range for the rewiring probability for which the rewired networks will display small world properties.

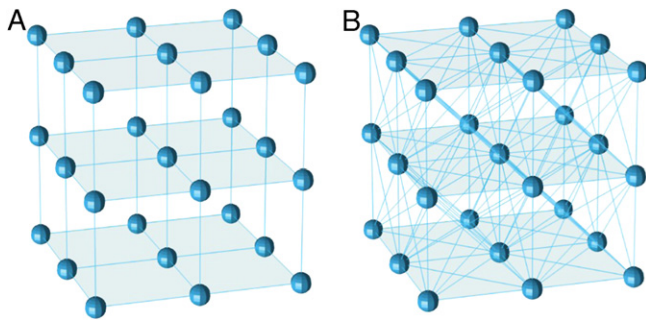
The average shortest path length measures the average number of edges that a piece of information needs to be passed through to reach the destination node (global property), i.e. a measure for “averaged distance” between nodes in the graph. The average clustering coefficient measures the degree of cliquishness (local property) existing in the network. The shortest path length between any two neurons in the liquid filter is calculated by the minimum number of synapses one must travel to get from one neuron to the other. The average shortest path length is obtained by averaging the shortest path for each pair of neurons across the whole liquid filter network. For each neuron, the clustering coefficient is the number of connections it made with its neighbors (excluding itself), divided by the total number of possible connections. The average clustering coefficient is the mean of the clustering coefficients for all the neurons in the network.

The liquid filter used here consisted of 540 LIF neurons placed in a grid having the dimensions  $6 \times 6 \times 15$ . We have tested two lattice connectivity structures (Fig. 1). The connections between the neurons are initially constructed according to one of the two lattice structures. After the construction of the lattice liquid filter, each synapse has a probability  $P$  to be rewired to another randomly chosen neuron. Self-connections and duplicated connections (having the same pre- and post-synaptic neurons) are not allowed. It should be noted that this rewiring process does not alter the total number of synapses in the liquid filter. Lattice (A) will generate 2808 synapses in the liquid filter, and (B) will generate 10,468 synapses. As the rewiring probability  $P$  increases, more long-range synaptic connections will be generated, which will greatly reduce the average shortest path length. The liquid filter will become a totally random network when  $P$  is one. We tested the performance of such liquid filters by varying the rewiring probability  $P$ , and  $W_{scale}$  which is the global scaling factor for synaptic weights.

### 3.3. Axon model

Axon growth cones follow straight lines unless guidance cues are present or pathways are blocked (Yamamoto et al., 2002). Previous simulations (Kaiser et al., 2009) have shown that a





**Fig. 1.** The lattices used to generate small-world networks. Lattice (A) will result in 6 outgoing degrees and 6 incoming degrees for the central vertex (neuron). Its clustering coefficient is 0 because there is no connection between any vertex's neighbors. Lattice (B) has 26 outgoing and incoming degrees for the central vertex.

simple rule of axonal straight outgrowth in random directions in two dimensional space with a neuron density of around 4% results in a connection length distribution similar to that found experimentally in neuronal networks of rat brain and *C. elegans*. We followed this work to construct a liquid filter in 3 dimensional space with the size of  $25 \times 25 \times 25$  (Fig. 2). 540 neurons (the same number of neurons as the previous two models) are randomly placed at integer coordinates in space. This setting leads to a sparse neuron density of 3.46%. For each neuron, a random vector is generated to be its axon growing direction. Axons are assumed to be growing straight until it touches the border of the liquid filter. As each neuron has a finite dendritic surface which limits the number of synaptic contacts, vacancies for the pre- and post-synaptic connections are constrained: no additional incoming or outgoing connections can be established if both pre- and post-synaptic vacancies are occupied. A connection will be built from neuron A to B only if the distance from B to A's axon line is smaller than  $R$  units, and at the same time, neuron A has at least one post-synaptic vacancy, and neuron B has at least one pre-synaptic vacancy. Axons are built one by one, in random sequence. Therefore, neurons whose axons are built first will have an advantage to connect to whichever neurons they want, while the others whose axons are built later will have to selectively make connections to neurons that still have vacancies. A lower limit for pre-synaptic versus post-synaptic vacancies will cause competition for the pre-synaptic vacancies between neurons. The limit of pre- and post-synaptic vacancies was set to 15 and 30, respectively. In this

topology, all outgoing connections are “directional”, in contrast to radial connectivity of the original ‘lambda model’. Each neuron's outgoing connections' range has a cylindrical shape with radius  $R$  and the axon being the central axis. We will investigate the performance of such liquid filters with different  $R$  values (axon cover range) and different scales of synaptic strength  $W_{scale}$ .

In CSIM, the length of connections between neurons does not have any effect on the simulation. Hence, in the axon growth model, once the liquid filter is build, the physical locations of neurons do not affect the simulations. Since this axon model produces a biologically realistic distribution of connection lengths (Kaiser et al., 2009), it is desirable that the lengths of connections affect the outcome of the simulations. Spike propagation along the axon proceeds with a constant speed, so longer axons produce larger axonal delays. Therefore, we further adjust each axon's delay to be proportional to its length (the Euclidean distance between the pre and post-synaptic neurons), such that longer axons have larger delays.

## 4. Methods

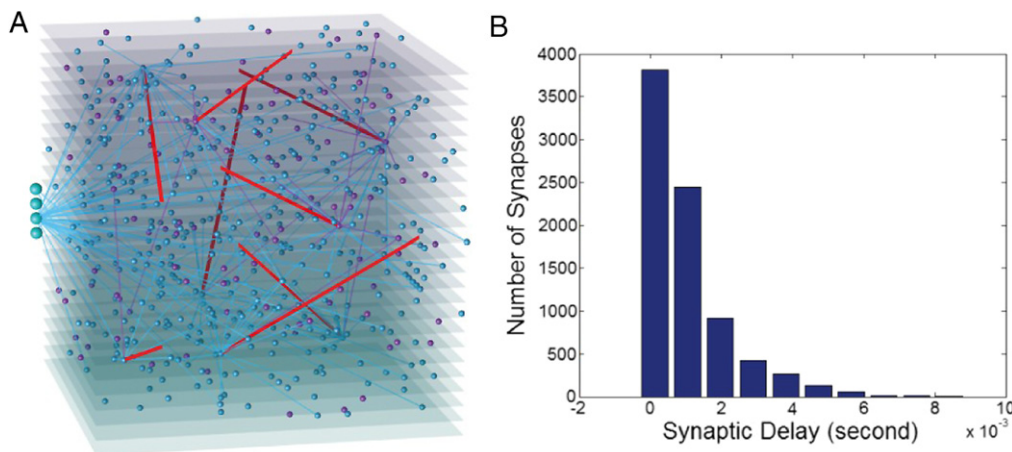
### 4.1. Classification tasks description

The liquid state machine operates on spikes; therefore, spike train classification is a suitable task for the LSM. A binary classification task is used to evaluate the performance and the properties of the liquid filter with the three connection models described above. This task is a benchmark problem and is likely to be relevant for computations in cortical neural microcircuits (Legenstein & Maass, 2007).

#### Poisson spike train templates classification

80 templates are generated, each consisting of four 20 Hz Poisson spike trains with a duration of 200 ms. The 80 templates are divided into two groups, each of which denotes a class. From these two classes of templates, stimuli are produced by adding Gaussian jitter to the templates with mean zero and standard deviation of 4 ms. A total of 2500 jittered stimuli are generated, 2000 for training and 500 for testing. The LSM is trained to classify which group of templates the input spike trains are generated from. The performance is measured by the percentage of jittered inputs being correctly classified for the test group.

Since the 80 templates are randomly generated from the same Poisson distribution, there are no obvious discriminant features or



**Fig. 2.** The axon model and the distribution of synaptic delays. (A) The axon model. Red lines denote axons. 540 neurons, 20% inhibitory (purple) and 80% excitatory (blue), are randomly placed in a  $25 \times 25 \times 25$  space. The 4 neurons shown on the left are input neurons. Each input neuron is randomly connected to 10% of the 540 neurons in the liquid filter. Only a few neurons' axons are drawn for clarity. (B) The distributions of synaptic delays. We set the delays to be proportional to the synaptic length, thus, the synaptic delays follow an exponential distribution. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

rules that distinguish the two classes, making this classification task quite challenging. The LSM needs to somehow “memorize” which template belongs to which group during training, and then apply this ‘memory’ to the testing data set to perform classification.

#### 4.2. Separation, generalization, and the kernel quality

An optimal classifier combines an ability to distinguish between many inputs (separation property) and generalize from learned inputs to new examples (generalizability). The liquid filter has the ability to operate in a chaotic regime, in which small differences between inputs result distinct network states. Although this improves kernel discrimination quality, it also lowers generalizability (Legenstein & Maass, 2007). Highly chaotic networks might produce vastly different network states due to the presence of small amounts of noise in the input. This magnification of noise is certainly undesired in classification. When the liquid filter is at the edge of chaos, this tradeoff could be optimal.

The liquid filter possesses fading memory due to its short-term synaptic plasticity and recurrent connectivity. For each stimulus, the final state of the liquid filter, i.e. the state at the end of each stimulus, carries the most information. It has been proposed (Legenstein & Maass, 2007) that the rank of the final state matrix  $F$  can reflect the separation and generalization ability of a kernel:

$$F = \begin{bmatrix} S(1)^T \\ S(2)^T \\ \dots \\ S(N)^T \end{bmatrix} \quad (4.1)$$

where  $S(n)$  is the final state vector of the liquid filter for the stimulus  $n$ . Each column of  $F$  represents one neuron’s response for all the  $N$  stimuli. If all  $N$  inputs are very different from each other, i.e. they are from  $N$  classes; a higher rank in  $F$  indicates better kernel separation. If  $N$  inputs are from very few classes, a lower rank in  $F$  means better generalization.

The numerical rank of a matrix is very sensitive to noise, especially for a chaotic liquid filter. We therefore refine the measurement by taking the *effective rank* of the matrix, which is not only robust to noise, but also shows the degree of linear dependency in the final state matrix. In cases where the network activity is high, the state matrix could be full rank. However, there may be groups of neurons whose activity is inter-dependent. Those dependent neurons are doing redundant work. Therefore, the numerical rank does not reveal the true rank of the state matrix, while the effective rank does.

The effective rank is calculated by a Singular Value Decomposition (SVD) on  $F$ , and then taking the number of singular values that contain 99% of the sum in the diagonal matrix as the rank. i.e.

$$F = U \Sigma V^T \quad (4.2)$$

where  $U$  and  $V$  are unitary matrices, and  $\Sigma$  is a diagonal matrix  $\text{diag}(\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_N)$  that contains non-negative singular values in descending order. The effective rank is then determined by:

$$k_{\text{effective}} = \min_k \left( \sum_{i=1}^k \lambda_i \geq 99\% \times \sum_{j=1}^N \lambda_j \right). \quad (4.3)$$

A similar method has been used to estimate the number of hidden units in a multi-layer perceptron (Teoh, Tan, & Xiang, 2006).

## 5. Results and discussion

For every liquid filter, the simulation is divided into three steps. First, 500 templates (500 spike trains that are different from each

other) are generated, and then injected into the liquid filter, to measure its separation ability by calculating the effective rank of the final state matrix  $F$ . Second, 500 jittered spike trains generated from 4 different templates are injected into the liquid filter, to measure its generalization ability. Finally, the Poisson spike train classification task is performed using this liquid filter, to calculate its training and testing accuracy. Simulations were performed using a range of values for the synaptic weight scale  $W_{\text{scale}}$  and the connection range, in the case of the lambda and axon models. For the small world networks, we have investigated the effects of varying  $W_{\text{scale}}$  and the rewiring probability.

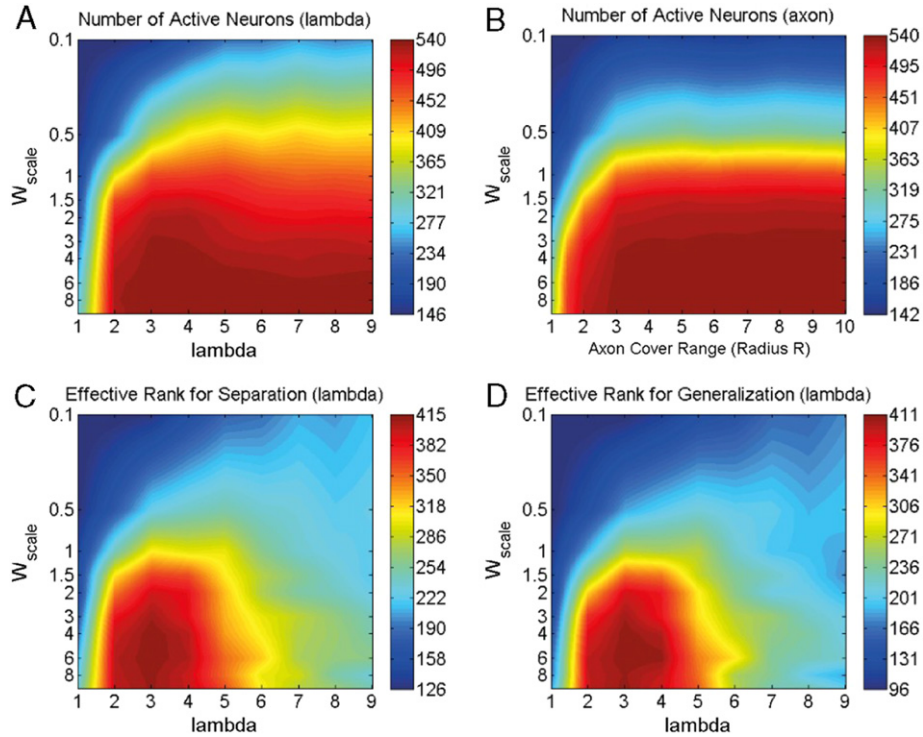
For the spike template classification experiments, the number of ‘active’ neurons was calculated and plotted against the parameter values for synaptic weight and connectivity (Fig. 3). Active neurons are defined as the neurons that fire at least one spike over the entire test data set. This plot indicates how many neurons have responses to the inputs, and reflects the network’s activity level.

The parameter  $W_{\text{scale}}$  is a factor that enhances the signal transmission within the liquid filter. Hence, it is expected that when the  $W_{\text{scale}}$  is large, more neurons will fire (Fig. 3(A), (B)). As lambda and axon cover range increases, more synaptic connections are created in the liquid filter (Fig. 4). The synaptic connections can be thought as communication channels between neurons. With more communication channels, the neurons will have more paths to “talk” to each other, and the average shortest path length is also reduced. Because the size of the liquid filter is fixed, the maximum number of active neurons is 540. Thus in the lower right part (the dark red region) of Fig. 3(A) and (B), the number of active neurons becomes invariant to  $W_{\text{scale}}$ , as well as the lambda value and the axon cover range. Another reason for this invariance is that, for the axon model, we set limits for the number of pre and post-synaptic vacancies. Therefore, the number of synapses for the axon model reaches a limit when the axon cover range (radius  $R$ ) is large. This makes the red region in Fig. 3(B) to be more flat than the lambda model (Fig. 3(A)). These two figures show that, when the number of synapses is fixed, larger synaptic weights will increase the number of active neurons, and *vice versa*.

More synapses can certainly increase the numerical rank of the final state matrix used to calculate separation and generalization properties. In fact, the dependence of the numerical rank on  $W_{\text{scale}}$  and lambda shows a pattern very similar to that of the number of active neurons (data not shown). However, this is not the case for the effective rank (Fig. 3(C), (D)), which reaches an optimum for intermediate synapse density. For the lambda model, the number of synapses created increases steadily as lambda increases (Fig. 4). When lambda is 9, more than 50,000 synapses are created, i.e. on average about 100 synapses/neuron. This large number of connections may create more dependencies between neurons; thus the effective ranks of both separation and generalization are decreased for the lambda model when the number of synapses is too high.

#### 5.1. Regions with satisfactory performance

Several interesting phenomenon can be observed from the test performance plots in Fig. 5. The location and the shape of the red region, which indicates a region of good performance for each connection model, are highly dependent on the number of active neurons. By comparing Figs. 3(A) and 5(A), it can be seen that high classification accuracy are located at the edge of the red region in Fig. 3(A). In other words, too many or too few active neurons in the liquid filter will not yield a good performance. The performance becomes satisfactory only for intermediate amount of active neurons. The same conclusion can be made by comparing



**Fig. 3.** Analysis for separation and generalization ability of the lambda and axon models. (A) The amount of active neurons for the lambda connection model. Values are coded in color. (B) The amount of active neurons for the axon connection model. (C) The effective ranks of the final state matrix to measure separation, for the lambda connection model. (D) The effective ranks of the final state matrix to measure generalization, for the lambda connection model. Each plot in this figure is obtained by interpolation of 100 points, and each point is calculated by averaging the results from 10 randomly initialized liquid filters with the parameters specified by the point. The horizontal axis is plotted in linear scale while the vertical is in log scale. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

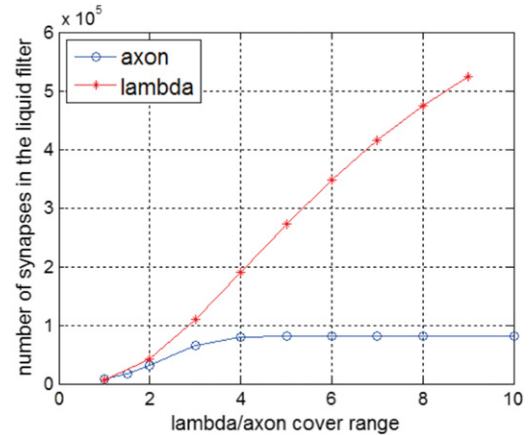
Figs. 3(B) and 5(B). This result is also valid for the small-world models.

We found that the number of synapses and  $W_{scale}$  are highly correlated to performance. In both Fig. 5(A) and (B), the red region spreads from the lower left to the upper right, indicating that as the number of synapses increases, the LSM requires smaller  $W_{scale}$  to obtain good performance. For the small-world networks (Fig. 5(C), (D)), the red region spreads almost horizontally. Note that the number of synapses is invariant to the rewiring probability for the case of the small-world networks. This indicates that the value of  $W_{scale}$  that yields satisfactory performance is not dependent on the rewiring probability, i.e. not sensitive to the small world properties (the average shortest path length and the average clustering coefficient). The red band in Fig. 5(C) is located at the  $W_{scale}$  range of (0.5, 4), while in 5D, the  $W_{scale}$  range is around (0.1, 1.5). Notice that Fig. 5(C) is obtained from the LSM with the lattice A structure having 2808 synapses, significantly less than the lattice B having 10,468 synapses. Based on these observations, we conclude that the range of  $W_{scale}$  for satisfactory performance depends on the number of synapses in the liquid filter. With more synapses, smaller  $W_{scale}$  is required to achieve good classification results.

## 5.2. Regions with optimal performance

In both Fig. 5(A) and (B), it can be seen that the dark red color fades from the lower left to the right. The best testing accuracy over the whole plotting region is located near or at the lower left corner. For larger values of lambda (or axon cover range), the performance of the liquid state machine was suboptimal, and could not be improved by tuning the parameter  $W_{scale}$ .

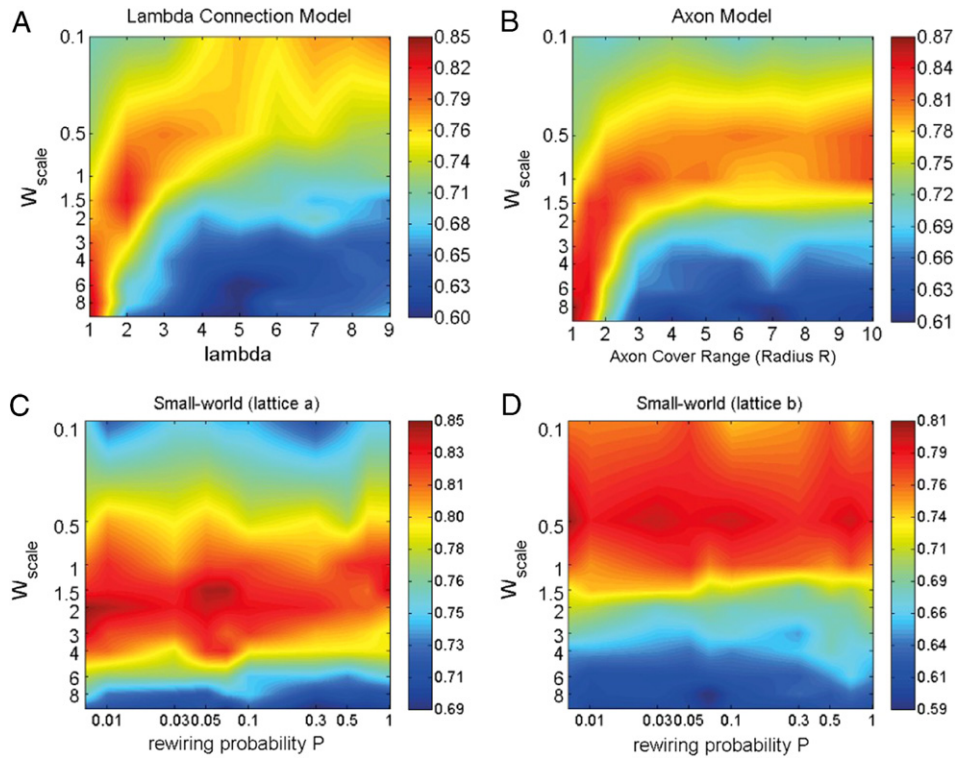
To explain this, we further examined the two types of parameters:  $W_{scale}$ , and lambda or axon cover range. First, increasing lambda or axon cover range will increase the number of synapses



**Fig. 4.** The number of synapses for the lambda and axon connection models. The number of synapses is saturated to 8100 for the axon model when the axon cover range  $R$  is greater than 4, because we limit pre and post-synaptic vacancies.

created in the liquid filter. Second, the  $W_{scale}$  parameter determines the mean synaptic strength. The variance of the synaptic weights is half of the mean, i.e. the coefficient of variation is 0.5 (see Section 2.2). Therefore, greater  $W_{scale}$  values produce larger weight variance. In cases where the number of synapses was small, a larger  $W_{scale}$  was required to obtain satisfactory performance. We therefore hypothesized that a large variance of the weights leads to better performance. This hypothesis explains why the best testing accuracy is always located at the lower left corner, where  $W_{scale}$  is large and the number of synapses is small. It also explains why the highest testing accuracy for the lattice A (Fig. 5(C)) is better than the lattice B (Fig. 5(D)), because the red region in 5C corresponds to higher  $W_{scale}$  values and thus larger weight variance than 5D.





**Fig. 5.** The testing performance of the Poisson spike templates classification task. (A) Lambda connection model. (B) Axon model. (C) Small-world with lattice *a*. (D) Small-world with lattice *b*. The classification accuracy is coded in color. Each plot is obtained by interpolating 100 points, and each point is calculated by averaging the results from 10 randomly initialized liquid filters with the parameters specified by the point. The horizontal axis for (C) and (D) are plotted in log scale, as the small world properties change fast when *P* is small. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

To test this hypothesis, we used liquid filters with a fixed synaptic connection model, the lattice *A* structure without rewiring, but with various  $W_{scale}$  and synaptic weights' coefficient of variation. The resultant synaptic weight distributions (synaptic weights follow Gamma distributions, see Section 2.2) are shown in Fig. 6(A) and (B). Fig. 6(C) shows that given a  $W_{scale}$  (i.e. given the mean of the synaptic strength) and fixed connection topology, large coefficient of variation produces better results. The satisfactory performance region was highly dependent on the edge of the red region in the active neuron plot; thus, the red region in Fig. 6(C) spreads horizontally, due to the flat edge of the red region in Fig. 6(D), which shows the amount of the active neurons. The best classification accuracy is obtained when the coefficient of variation is close to 1.

### 5.3. Music and non-music classification

The experiments described until now were limited to applying the LSM to benchmark problems. To further support the above observations and discussions, we applied LSM to a real-world task: music classification.

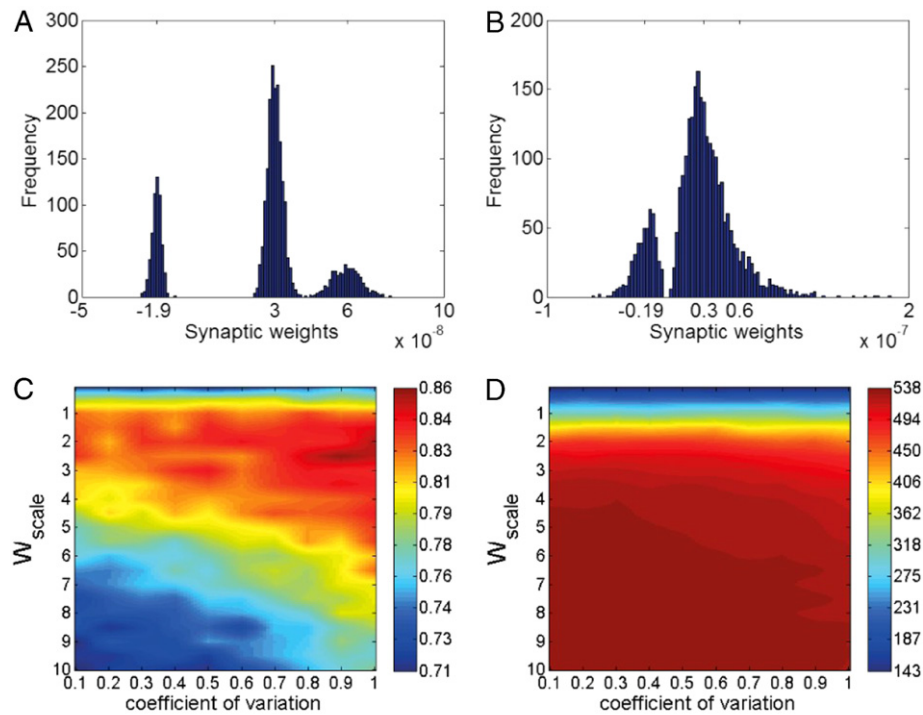
Music classification is a complex problem that requires the analysis of a highly ordered temporal structure. Here we follow up on our previous work on polyphonic music classification (Ju, Xu, & VanDongen, 2010). To create examples of note progressions that do not correspond to music ('non-music'), notes of a short musical piece are randomly swapped 100 times. The training data set contains 234 musical pieces and its corresponding 234 'non-music' versions, segmented from 50 classical music MIDI files. Testing data has  $177 \times 2$  pieces, segmented from 30 MIDI files. The liquid filter consists of 600 neurons with dimensions  $12 \times 10 \times 5$ . To facilitate classification, after coding the music and non-music into input spikes, the input stimuli are speeded up so the stimulus duration is 0.3 s. We let the readout neuron make a decision value

every 0.01 s, and the final classification decision at 0.3 s is the average of all the 30 output values from the readout neuron.

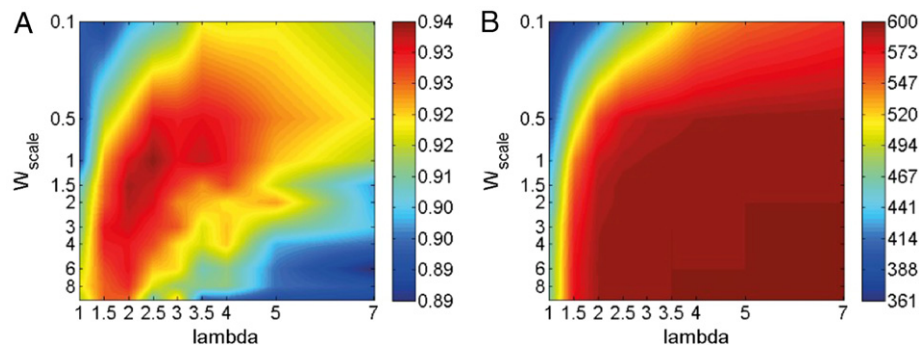
The results (Fig. 7) show that all the conclusions from the benchmark problem experiments described above are still valid in this practical application, except that the variance of the synaptic weights seems to have no obvious effect on the classification accuracy. This indicates that the positive effect of large weight variance is stimulus specific. It may not be topology specific, however, because we have observed improved performance in the spike train template classification task for all the connection models that were discussed so far: the lambda model, the axon growth model and small world networks. During the design of an optimal liquid filter for a given stimulus, enlarging the variance of synaptic weights should be considered as an option to achieve better performance.

Since both the number of synapses and the synaptic strength play important roles, we plotted the total synaptic strength per neuron, defined as the difference between the sum of excitatory synaptic weights and inhibitory weights, against the classification accuracy (Fig. 8). It can be seen that for the Poisson spike train classification problem, the lambda and axon model curves are similar, both reaching an optimum around  $2 \times 10^{-6}$ , while for the music classification task, the peak is at  $5 \times 10^{-7}$ . All three curves drop steeply when the average synaptic strength per neuron is larger than optimal. This indicates that the average synaptic strength is a key parameter for performance, and the optimal value of this parameter seems to be independent of connection topology. The optimum does depend on the stimulus type, since it was slightly left-shifted for the spatiotemporal music classification problem compared to the purely temporal spike train task.

The results are summarized below. First, when the number of synapses in the liquid filter is fixed, larger synaptic weights increased the number of active neurons, and *vice versa*. Second, poor performance was observed when the number of active



**Fig. 6.** Effects of synaptic weights distribution on the performance of the liquid state machine. (A) The distribution of the synaptic weights with a coefficient of variation 0.1. (B) The weights distribution when the coefficient of variation is 0.5. (C) The performance of the liquid filters with different coefficients of variation for the synaptic weights. (D) The number of neurons that fire at least one spike. Each of the plot in (C) and (D) is obtained by interpolating 200 ( $20 \times 10$ ) points with each point representing the average of 10 runs, i.e. 2000 liquid filters were created and simulated. All the liquid filters have the same connection structure with lattice A. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 7.** Music and non-music classification task using the lambda connection model. (A) Classification accuracy on training data. (B) Number of neurons that fire at least one spike. Each data point is the average of 10 liquid state machines.

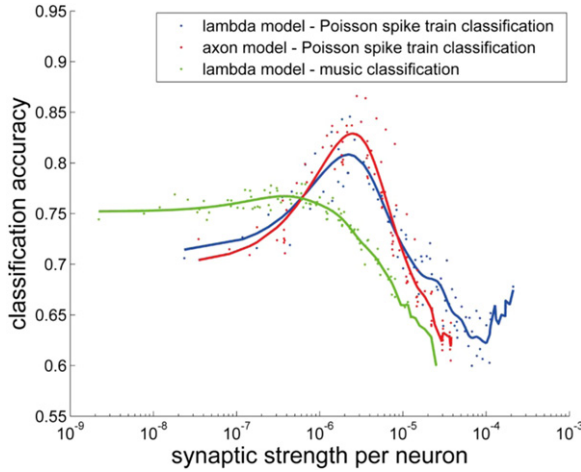
neurons was either large or small; optimum performance was achieved using an intermediate number of active neurons. Third, the performance of the LSM was highly related to the number of synapses in the liquid filter. When the number of synapses was large (small), smaller (larger) values of  $W_{scale}$  were needed to obtain satisfactory performance. Fourth, larger synaptic weight variance may produce better classification accuracy for certain types of stimuli. Fifth, the total synaptic strength per neuron is an important parameter to the performance, and the optimal value of this parameter is stimulus dependent. Finally, the performance of the LSM seems to have little relationship with the small world properties.

## 6. Evolving synaptic connectivity

We have shown that both synapse density and synaptic strength ( $W_{scale}$ ) are important to classification performance.

More synapses may not improve the performance. Satisfactory classification can be obtained with low synapse density, as long as synaptic strength is increased. In fact, optimal performance is sometimes obtained with a small number of very strong synapses. However, for a given problem, it is not easy to find the optimal synaptic connectivity. The experiments described above have used a brute-force approach, in which almost all of the combinations of  $W_{scale}$  and the connection parameters are simulated in order to localize the optimal point in parameter space. In addition to the structural optimization of synapse density and weights, the computational complexity should be taken into consideration. One of the most impressive capabilities of the LSM is real-time classification. The computational complexity of the liquid filter is crucial to real-time tasks, and increases dramatically with the number of synapses. Fewer elements in the liquid filter will greatly increase the computational speed. Therefore, it is desired to design a liquid filter with a minimal number of synapses and neurons, and still possesses good performance.





**Fig. 8.** Classification accuracy versus ‘synaptic strength per neuron’ for the lambda and axon connection models. The synaptic strength per neuron is obtained by summing up all the synaptic weights in the liquid filter, divided by the number of neurons. Each set of data is fit by a polynomial with degree of 15.

The LSM nonlinearly projects the input data to a high dimensional feature space through the liquid filter that is acting as a kernel. Because the optimal weights for the readout neuron are calculated by Fisher’s Linear Discriminant, the liquid state machine is acting as a Kernel Fisher Discriminant machine. Recently, the method of Kernel Fisher Discriminant Analysis (KFDA) (Kim, Magnani, & Boyd, 2006; Mika et al., 2003) has attracted a lot of interest. The performance of KFDA largely depends on the choice of the kernel, and there are many algorithms for optimal kernel selection in convex kernel space. However, the liquid filter is a highly nonlinear kernel that is very difficult to analyze mathematically. Thus, in most of the previous studies, optimization of the kernel was performed by tuning global parameters.

Approaches applied in evolving systems could be used to tackle complex optimization problems. There are two types of evolving systems: data-based learning and learning through evolution. For the first type, tuning of parameters, such as neurons and connections, is driven by the data samples (Kasabov, 2007). This approach has been used to evolve spiking neural networks in an online evolvable and adaptive fashion (Wysoski, Benuskova, & Kasabov, 2010). This type of system is similar to life-time learning in biological organisms, which is based on accumulation of experience. The mammalian central nervous system has various mechanisms to learn from experience, such as Long-term potentiation and depression (LTP/LTD) and spike-timing dependent plasticity (STDP). STDP-like learning rules have been studied to improve the liquid state machine’s separation ability (Norton & Ventura, 2006, 2010). However, as discussed in Section 4.2, the liquid state machine performance depends not only on its separation ability, but also on how well it can generalize. Enhancement in separation ability is likely to cause degradation in generalization ability. Furthermore, the computational complexity of the liquid filter was not emphasized in these studies. In this paper, we used the second type of evolving systems, in particular, genetic algorithms, to seek an optimal tradeoff between separation and generalization abilities, while minimizing the kernel. Our genetic algorithm approach mimics biological evolution that happens in millions of years, rather than life-time learning of an organism.

Genetic algorithms have been widely used for solving complex optimization problems. In fact, various methods to evolve feed-forward Spiking Neural Networks (SNN) have been proposed and studied, such as the Quantum-inspired SNN (Schliebs, Defoin-Platel, & Kasabov, 2010) and Probabilistic Evolving SNN

(Hamed, Kasabov, & Shamsuddin, 2010). These frameworks directly use genetic algorithms or integrate them with particle swarm optimization (Hamed, Kasabov, & Shamsuddin, 2012) to select relevant features and optimize parameters. They can potentially substitute the readout neuron in the LSM to extract information and select discriminative neurons from the liquid filter. This paper focuses on the connectivity of the liquid filter; optimization of the readout function has not been addressed. It may be possible to achieve even better performance if both the liquid filter and the readout function are evolved, employing the above frameworks.

Neuro-Evolution of Augmenting Topologies (NEAT) (Stanley & Miikkulainen, 2002) is a genetic algorithm that evolves both the neurons and the connections in the network. This algorithm can be used to evolve the liquid filter towards an optimal kernel. NEAT starts from a minimal network structure and gradually increase the network complexity through evolution by adding neurons and synaptic connections. The inherent characteristic of “augmenting topologies” makes NEAT a suitable candidate to find the optimal liquid filter with a minimal number of synaptic connections.

### 6.1. Network settings

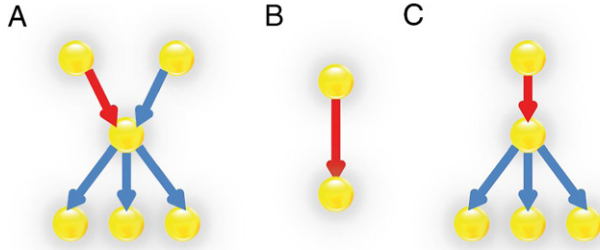
For each individual, the liquid filter is fixed to have  $6 \times 6 \times 15$  neurons with 80% excitatory and 20% inhibitory neurons, the same as before, but with no connections at the first generation. Each input neuron is randomly connected to 10% of the neurons in the liquid filter during the initialization phase, which is the same as in all the connection models discussed above. Since we are focusing on evaluating the quality of the liquid filter, the input connections remain untouched once they are initialized. Therefore, in every generation, the input connections of all the individuals in the population are exactly the same, and they are not involved in the evolution process. Unlike NEAT, which evolves the number of neurons in the network, we only evolve synapses. At the first generation, because there is no kernel connectivity, only those neurons having incoming connections from the input neurons may fire. All the other liquid neurons that do not fire during the simulation are “redundant” neurons, as they do not contribute any information to the readout or the classification result. Thus, the liquid filter starts from a minimal structure that consists of only those neurons receiving input synapses. More neurons will fire and become useful when new synapses are added into the network during evolution. Here we call the neurons that fired at least one spike during the presentation of all the input stimuli ‘active neurons’.

### 6.2. Parameter settings

Each gene, representing a synapse, has three properties: the indexes of the pre- and post-synaptic neurons it connects, and its synaptic weight. When a new synapse is created, depending on the connection type, a random weight is generated in the range of  $[10^{-9} - 10^{-5}]$  (excitatory synapses) or  $[-10^{-5} - 10^{-9}]$  (inhibitory synapses). Like the original work (Stanley & Miikkulainen, 2002), each synaptic weight has a probability of 0.9 to mutate in each generation. Because the weights are represented as real numbers, a mutation adds a random value in the range of  $[-1.5 \times 10^{-8} - 1.5 \times 10^{-8}]$ . Weight values are constrained between  $10^{-10}$  and  $3 \times 10^{-5}$ .

The general concept of speciation is that two individuals belong to different species if their difference is larger than a speciation threshold. The difference  $\delta$  between two individuals is calculated as the weighted sum of the number of different connections ( $G$ ), and the weight difference  $\tilde{W}$  between those same connections, i.e.

$$\delta = \alpha G + \beta \tilde{W}$$



**Fig. 9.** Connection structures that are considered for the synaptic deletion process. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where  $\alpha$  and  $\beta$  are two coefficients, set to 1 and  $2 \times 10^8$  respectively. The speciation threshold is set to 600. 80% of individuals in each species will participate in the crossover process, and 30% of each species with the lowest performance will be discarded in each generation. Interspecies crossover is not allowed. Linear ranking selection is used. To limit computational complexity, the population size is set to 20.

### 6.3. Create/delete synaptic connections

In each generation, a random number of new synapses (in the range of [1–20]) will be added to the kernel. The new connection's pre-synaptic neuron must be an active neuron, because there will not be any effect if a connection is outgoing from a silent neuron. Duplicated connections and self-connections are not allowed.

Besides adding synapses, a synaptic deletion process is needed to prevent an explosion of the number of connections. The deletion process is not enabled at the beginning as there are no connections in the kernel yet. For each generation, a random number of synapses (between [1–20]) are deleted from the kernel. Connections should not be deleted randomly, because it may result in an unconnected graph in the network topology. Unconnected parts are redundant and will never fire, but still consume computational resources. If unconnected parts are allowed to persist, they may become connected by the addition of a new synapse, causing a large change in the network. This is undesirable, as it does not follow the concept of genetic algorithms, where performance adjusts incrementally through each new generation.

Therefore, we have to examine the network topology to determine whether a connection can be deleted or not. In Fig. 9(A) the post-synaptic neuron of the red connection is important in the sense that it is disseminating information to many other neurons. If the red connection is deleted, its post-synaptic neuron still has other incoming connections. The network will not change a lot by deleting such connections. If the post-synaptic neuron does not have any outgoing connections (Fig. 9(B)), it is also safe to delete. However, in Fig. 9(C), many other neurons are connected to the post-synaptic neuron, which is the hub of the network, and the red synapse is the only incoming connection to the post-synaptic neuron. Therefore, the deletion of the red connection may result in an unconnected part. In summary, the connection deletion rule is: a connection can be deleted when either its post-synaptic neuron has more than 1 incoming synapse, or less than 2 outgoing synapses.

### 6.4. Fitness calculation

Fitness calculation for each individual in the population is crucial to Genetic Algorithms as it drives the evolution process. Fitness should directly measure the quality of the kernel quantitatively, such that the algorithm knows which direction to evolve into. As discussed before, the LSM can be thought as a Kernel Fisher Linear Discriminant machine. The measure for the kernel quality, i.e. the fitness value for each individual, should be from

the perspective of the FLD readout, because once the kernel is constructed, the classification performance is determined. Below, we derive the calculation of the individual fitness.

We use  $S_K(m)$  to denote the final state vector that is obtained by injecting input stimulus  $m$  into the liquid filter (kernel)  $K$ . Suppose that we are dealing with a binary classification problem. The stimuli are arranged such that the first  $n_1$  instances belong to class 1, and the remaining  $n_2$  instances belong to the other class. The mean of the final state vectors for the two stimuli classes, when using a kernel  $K$  are

$$\mu_K^1 = \frac{1}{n_1} \sum_{m=1}^{n_1} S_K(m), \quad \mu_K^2 = \frac{1}{n_2} \sum_{m=n_1+1}^{n_1+n_2} S_K(m)$$

and the covariance is

$$\Sigma_K^1 = \frac{1}{n_1} \sum_{m=1}^{n_1} (S_K(m) - \mu_K^1)(S_K(m) - \mu_K^1)^T$$

$$\Sigma_K^2 = \frac{1}{n_2} \sum_{m=n_1+1}^{n_1+n_2} (S_K(m) - \mu_K^2)(S_K(m) - \mu_K^2)^T.$$

The within-class scatter matrix  $M_W^K$  and the between-class scatter matrix  $M_B^K$  for kernel  $K$  are defined as:

$$M_W^K = \Sigma_K^1 + \Sigma_K^2 + \alpha I$$

$$M_B^K = (\mu_K^1 - \mu_K^2)(\mu_K^1 - \mu_K^2)^T.$$

Note that the covariance matrix could be singular; thus we add a small regularization term  $\alpha I$  ( $\alpha > 0$ ) to  $M_W^K$ . Fisher's Linear Discriminant tries to maximize the distance between the means of the two classes and minimizing the variance of each class. We therefore maximize the Fisher discriminant ratio (FDR)  $J$ :

$$J(K, W) = \frac{W^T M_B^K W}{W^T M_W^K W}.$$

By solving  $\frac{\partial J(K, W)}{\partial W} = 0$ , the maximum  $J$  is

$$J_{\max}(K) = (\mu_K^1 - \mu_K^2)^T (M_W^K)^{-1} (\mu_K^1 - \mu_K^2)$$

where the weight vector is

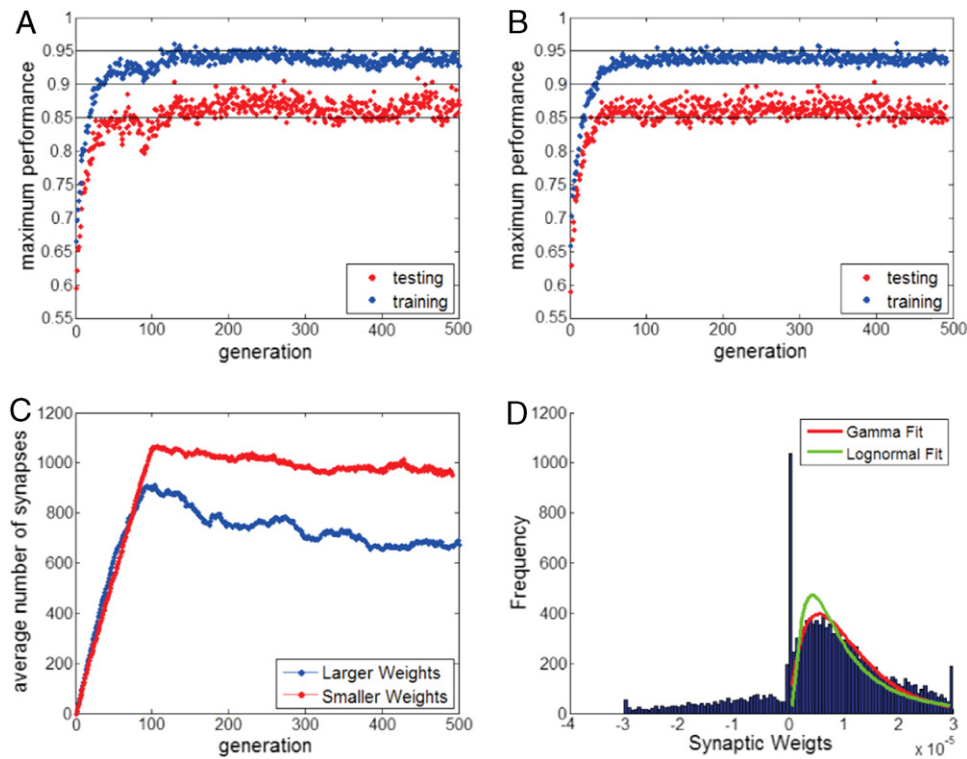
$$W(K) = (M_W^K)^{-1} (\mu_K^1 - \mu_K^2).$$

The maximum classification accuracy that can be achieved by using the kernel  $K$  is determined by the maximum FDR value  $J_{\max}$ . A larger  $J_{\max}$  means that the kernel (liquid filter)  $K$  can better transform the data, i.e. the liquid filter's projection of input stimuli into feature space results in a better separation by FLD.

The main goal of the genetic algorithm is to optimize the fitness value. Therefore,  $J_{\max}(K)$  is used as the fitness for each individual during evolution. Note that two kernels giving the same classification accuracy most probably have different FDRs, showing which kernel is better. Therefore, we are not using the classification accuracy as the fitness, because it is inexact and only indirectly reflects the quality of the liquid filter, as compared to the FDR which measure fitness directly.

### 6.5. Simulation results

The classification accuracy of the liquid state machine is strongly dependent on the number of synapses in the kernel. Therefore, synapse deletion should be enabled at a suitable time during evolution in order to prevent the development of excessive synapses. We cannot start the deletion at the first generation because there is no connection in the liquid filter at the beginning. Therefore, the evolution is divided into two phases: adding synapses and evolving topologies. The phase in which synapses



**Fig. 10.** Simulation results using different synaptic weight limits. (A) The maximum classification accuracy of the population when the synaptic weights have larger limits, capped to  $\pm 3 \times 10^{-5}$ . (B) The maximum classification accuracy when the weights have smaller limits, capped to  $\pm 10^{-5}$ . (C) The average number of synapses in the population during evolution for both the cases of larger and smaller weight limits. (D) The distribution of the synaptic weights for the whole population at generation 500. Excluding the zero bin and the last bin which are the bounds of the positive weights, the fitting curves are  $\sim$ Gamma (1.99,  $5.5e-6$ ), and  $\sim$ lognormal ( $-11.7$ ,  $0.82$ ).

are added has an element of supervision. After the deletion is enabled, the genetic algorithm has to refine the connectivity through evolution and strive for better performance. This second phase is unsupervised.

We start the connection deletion process when the fitness starts to drop due to excessive synapses. Linear regression on the mean population fitness over the past 30 generations was used to detect the performance drop. By using this method, when the performance drop is detected, the number of synapses in the liquid filter is already more than needed because the linear regression is done over the past 30 generations. Thus, it should be expected that the number of synapses will decrease after the synaptic deletion is enabled. The result is shown in Fig. 10(A). At generation 90, the performance started to drop, and the synaptic deletion was enabled. It can be seen that the classification accuracy for both the training and testing started to increase after the deletion was enabled. The number of synapses keeps decreasing, and converges to around 650 synapses.

We also ran another experiment by using smaller synaptic weights: new synapses had 10 times smaller weights and the mutated weights are capped in a smaller range (Fig. 10(B)). The maximum performance seems slightly lower than the previous case with larger weights. Fig. 10(C) plots the average number of synapses in the population for both the experiments together. In Section 5, we have discussed that smaller weights need more synapses to achieve satisfactory performance. This was confirmed by the genetic algorithm, where the number of synapses converged to 900 and 650 for the smaller and larger weights, respectively.

The distribution of the weights for the whole population at generation 500 for the large weight case is plotted in Fig. 10(D). The histogram seems to follow a gamma distribution. The parameters for the distribution are estimated using maximum likelihood estimation. In rat visual cortex, the synaptic strength distribution in layer 5 pyramidal neurons follows a lognormal distribution

(Song, Sjöström, Reigl, Nelson, & Chklovskii, 2005). Lognormal and Gamma distributions have similar shapes. A similar shape of weight distributions was also found in cortical layer 2/3 pyramidal–pyramidal synapses, hippocampal CA3–CA1 synapses, and cerebellar granule cell–Purkinje cell synapses (Barbour, Brunel, Hakim, & Nadal, 2007). Notice that the bin near zero has the highest magnitude, indicating that there are many silent synapses after evolving 500 generations. Experimental results show that in cerebellar cortex, many unitary granule cell–Purkinje cell synapses do not generate detectable electrical responses (Isope & Barbour, 2002), and it has been shown that silent synapses are a necessary by-product of optimizing learning and reliability for the classical perceptron, which is a prototype of feed-forward neural network (Brunel, Hakim, Isope, Nadal, & Barbour, 2004). Hence, the genetically evolved liquid filter optimized for binary classification by FLD has a biologically realistic synaptic weight distribution.

As we optimize classification performance by evolving network connections, the final generation of the population may contain some structural properties of an optimal liquid filter. To investigate this, we performed an additional 3 experiments using NEAT optimization, and pooled all the evolved networks to statistically evaluate the clustering coefficients and average shortest paths length. No significant correlation was found between the fitness of each individual and its clustering coefficient; but there is a weak positive correlation with the average shortest path length (the correlation coefficient is 0.3155,  $p = 0.0014$ ). This indicates that larger path length could be beneficial to performance. However, the results from the small-world experiments have shown that the average shortest path length does not have an obvious effect on performance. A possible explanation for this apparent contradiction is that, for the small world networks described in Section 3.2, the average shortest path length is below 9 due to thousands of connections; however, for the evolved networks with only hundreds of connections, the path lengths ranged between 6



**Table 1**  
Best classification accuracy.

	Highest training performance (%)	Highest testing performance (%)
Lambda model	93.8	88.2
Axon model	95.5	89.2
Small-world lattice <i>a</i>	95.3	87.8
Small-world lattice <i>b</i>	92.2	83.4
GA with NEAT (large weights)	96.1	91.0
GA with NEAT (small weights)	96.3	90.4

and 15. The effect of the path length may be only obvious in this extended range.

In all the above experiments, the simulation results show that, given a synaptic weight scale, performance reaches an optimum at an intermediate synapse density. The proposed genetic algorithm, which evolves both the connection topology and weights in the kernel, eventually finds this range of number of synapses, no matter how many synapses there are at the beginning of the evolution. We performed another experiment by starting the connection deletion process much earlier at generation 10. The network is still able to converge to the optimal number of synapses, but the convergence speed is much slower.

Table 1 lists the best classification accuracy on the Poisson spike templates classification task as described in the Section 4.1, for the exhaustive searches done for all the connection topologies and the genetic algorithm that we have discussed. They all share the same data set for fair comparison. The genetic algorithm using NEAT gives the highest classification accuracy for both training and testing. This could be due to the fact that GA has the freedom to construct any type of topology and adjust individual synaptic strength, as long as the network connections form a connected graph; whereas all the other models are restricted, for example, the axon model has directional connections and the lambda model has more local connections.

## 7. Conclusions

The highly nonlinear, recurrent and (potentially) chaotic nature of the liquid filter makes LSMs difficult to analyze analytically. While deriving analytical methods to construct an optimal kernel is therefore infeasible, empirical rules may facilitate the design of the LSM. As the LSM is a biologically realistic model, biologically inspired approaches could be helpful in refining LSMs for improved performance.

In the experiments described here, we have investigated the effects of synaptic connectivity on the performance of the LSM, in terms of both synaptic weights and topology. Various connection models have been explored, including the lambda model, an axon growth model, and small world networks. The quality of the liquid filter has little relationship with the small world property, but is highly dependent on synapse density and the distribution of synaptic weight. These two parameters are strongly inter-dependent. One possible explanation to the insensitivity of the small-world properties is that the brain functional network is a small-world across brain regions, but not within brain regions. The LSM is a cortical column model that has few hundreds of neurons; therefore, the effect of small-world is not obvious.

In addition, we conclude that large weights are required when the number of synapses is small. For any given distribution of synaptic weights, identifying the optimal synapse density is critical for performance. We propose a method of using genetic algorithms to evolve the liquid filter's connectivity to efficiently identify the optimal number of synapses. The simulation results show that the evolved LSM combines high classification accuracy with low computational complexity. Interestingly, the distribution of synaptic weights arrived at the end of evolution has a shape that is similar to mammalian cortical circuitry.

As there are too many potential connection topologies in biological neural networks, it is impossible for us to address them all; however, we still can find some rules from the models discussed in this paper. In addition, in this paper we simulated the LSM in the scale of hundreds of neurons and thousands of synapses due to the limitation of computation power. Larger scale simulation of neural networks in future could yield better liquid filter from the proposed genetic algorithm, and may provide more insights on brain networks. However, to fully understand the astronomical complexity of brain networks with 100 trillion connections, we still have a long way to go.

## References

- Achard, S., Salvador, R., Whitcher, B., Suckling, J., & Bullmore, E. (2006). A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *The Journal of Neuroscience*, 26, 63–72.
- Amaral, L. A., Scala, A., Barthelemy, M., & Stanley, H. E. (2000). Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97, 11149–11152.
- Baker, J. L. (2003). Is there a support vector machine hiding in the dentate gyrus? *Neurocomputing*, 52–54, 199–207.
- Bakker, A., Kirwan, C. B., Miller, M., & Stark, C. E. (2008). Pattern separation in the human hippocampal CA3 and dentate gyrus. *Science*, 319, 1640–1642.
- Barbour, B., Brunel, N., Hakim, V., & Nadal, J. P. (2007). What can we learn from synaptic weight distributions? *Trends in Neurosciences*, 30, 622–629.
- Brunel, N., Hakim, V., Isope, P., Nadal, J. P., & Barbour, B. (2004). Optimal information storage and the distribution of synaptic weights: perceptron versus Purkinje cell. *Neuron*, 43, 745–757.
- Gupta, A., Wang, Y., & Markram, H. (2000). Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex. *Science*, 287, 273–278.
- Hamed, H. N. A., Kasabov, N., & Shamsuddin, S. M. (2010). Probabilistic evolving spiking neural network optimization using dynamic quantum inspired particle swarm optimization. *Australian Journal of Intelligent Information Processing Systems*, 11(01), 23–28.
- Hamed, H. N. A., Kasabov, N., & Shamsuddin, S. M. (2012). Dynamic quantum-inspired particle swarm optimization as feature and parameter optimizer for evolving spiking neural networks. *International Journal of Modeling and Optimization*, 2, 187–191.
- Hamed, H. N. A., Kasabov, N., Shamsuddin, S. M., Widiputra, H., & Dhoble, K. (2011). An extended evolving spiking neural network model for spatio-temporal pattern classification. In *IEEE international joint conference on neural networks, IJCNN'2011* (pp. 2653–2656).
- Isope, P., & Barbour, B. (2002). Properties of unitary granule cell->Purkinje cell synapses in adult rat cerebellar slices. *The Journal of Neuroscience*, 22, 9668–9678.
- Joshi, P. (2007). From memory-based decisions to decision-based movements: a model of interval discrimination followed by action selection. *Neural Networks*, 20, 298–311.
- Joshi, P., & Maass, W. (2004). Movement generation and control with generic neural microcircuits. *Biologically Inspired Approaches to Advanced Information Technology*, 3141, 258–273.
- Ju, H., Xu, J. X., & VanDongen, A. M. J. (2010). Classification of musical styles using liquid state machines. In *IEEE international joint conference on neural networks, IJCNN'2010* (pp. 1–7).
- Kaiser, M., Hilgetag, C. C., & van Ooyen, A. (2009). A simple rule for axon outgrowth and synaptic competition generates realistic connection lengths and filling fractions. *Cerebral Cortex*, 19, 3001–3010.
- Kaiser, M., Martin, R., Andras, P., & Young, M. P. (2007). Simulation of robustness against lesions of cortical networks. *The European Journal of Neuroscience*, 25, 3185–3192.
- Kasabov, N. (2010). To spike or not to spike: a probabilistic spiking neuron model. *Neural Networks*, 23, 16–19.
- Kasabov, N. (2007). *Evolving connectionist systems: the knowledge engineering approach*. Springer-Verlag.
- Kim, S.-J., Magnani, A., & Boyd, S. (2006). Optimal kernel selection in Kernel Fisher discriminant analysis. In *Proceedings of the 23rd international conference on machine learning* (pp. 465–472).

- Lago-Fernandez, L. F., Huerta, R., Corbacho, F., & Siguenza, J. A. (2000). Fast response and temporal coherent oscillations in small-world networks. *Physical Review Letters*, 84, 2758–2761.
- Legenstein, R., & Maass, W. (2007). Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, 20, 323–334.
- Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, 14, 2531–2560.
- Micheloyannis, S., Pachou, E., Stam, C. J., Vourkas, M., Erimaki, S., & Tsirka, V. (2006). Using graph theoretical analysis of multi channel EEG to evaluate the neural efficiency hypothesis. *Neuroscience Letters*, 402, 273–277.
- Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Smola, A., & Muller, K. R. (2003). Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 623–628.
- Natschläger, T., Markram, H., & Maass, W. (2003). Computer models and analysis tools for neural microcircuits. In *Neuroscience databases: a practical guide* (pp. 123–138). Kluwer Academic Publishers.
- Norton, D., & Ventura, D. (2006). Preparing more effective liquid state machines using hebbian learning. In *IEEE international joint conference on neural networks, IJCNN'06* (pp. 4243–4248).
- Norton, D., & Ventura, D. (2010). Improving liquid state machines through iterative refinement of the reservoir. *Neurocomputing*, 73, 2893–2904.
- Schliebs, S., Defoin-Platel, M., & Kasabov, N. (2010). Analyzing the dynamics of the simultaneous feature and parameter optimization of an evolving spiking neural network. In *IEEE international joint conference on neural networks, IJCNN'2010* (pp. 1–8).
- Schliebs, S., Mohammed, A., & Kasabov, N. (2011). Are probabilistic spiking neural networks suitable for reservoir computing? In *IEEE international joint conference on neural networks, IJCNN'2011* (pp. 3156–3163).
- Schrauwen, B., D'Haene, M., Verstraeten, D., & Campenhout, J. V. (2008). Compact hardware liquid state machines on FPGA for real-time speech recognition. *Neural Networks*, 21, 511–523.
- Simard, D., Nadeau, L., & Kröger, H. (2005). Fastest learning in small-world neural networks. *Physics Letters A*, 336, 8–15.
- Song, S., Sjöström, P. J., Reigl, M., Nelson, S., & Chklovskii, D. B. (2005). Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biology*, 3, e68.
- Stam, C. J. (2004). Functional connectivity patterns of human magnetoencephalographic recordings: a 'small-world' network? *Neuroscience Letters*, 355, 25–28.
- Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10, 99–127.
- Teoh, E. J., Tan, K. C., & Xiang, C. (2006). Estimating the number of hidden neurons in a feedforward network using the singular value decomposition. *IEEE Transactions on Neural Networks*, 17, 1623–1629.
- Verstraeten, D., Schrauwen, B., D'Haene, M., & Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, 20, 391–403.
- Verstraeten, D., Schrauwen, B., Stroobandt, D., & Van Campenhout, J. (2005). Isolated word recognition with the liquid state machine: a case study. *Information Processing Letters*, 95, 521–528.
- Vertes, P., & Duke, T. (2009). Neural networks with small-world topology are optimal for encoding based on spatiotemporal patterns of spikes. *BMC Neuroscience*, 10, O11.
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393, 440–442.
- Wysoski, S. G., Benuskova, L., & Kasabov, N. (2010). Evolving spiking neural networks for audiovisual information processing. *Neural Networks*, 23, 819–835.
- Yamamoto, N., Tamada, A., & Murakami, F. (2002). Wiring of the brain by a range of guidance cues. *Progress in Neurobiology*, 68, 393–407.
- Yamazaki, T., & Tanaka, S. (2007). The cerebellum as a liquid state machine. *Neural Networks*, 20, 290–297.