Neurocomputing 226 (2017) 145–160

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Performance and robustness of bio-inspired digital liquid state machines: A case study of speech recognition

Yingyezhe Jin, Peng Li[*]

Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, United States

ARTICLE INFO

Communicated by Dr. Shen Jianbing Shen

Keywords:
Liquid state machine
Performance
Robustness
Speech recognition

ABSTRACT

This paper presents a systematic performance and robustness study of bio-inspired digital liquid state machines (LSMs) for the purpose of future hardware implementation. Our work focuses not only on the study of the relation between a broad range of network parameters and performance, but also on the impact of process variability and environmental noise on the bio-inspired LSMs from a circuit implementation perspective. In order to shed light on the implementation of LSMs in digital CMOS technologies, we study the trade-offs between hardware overhead (i.e. precision of synaptic weights and membrane voltage and size of the reservoir) and performance. Assisted with theoretical analysis, we leverage the inherent redundancy of the targeted spiking neural networks to achieve both high performance and low hardware cost for the application of speech recognition. In addition, by modeling several types of catastrophic failure and random error, we show that the LSMs are generally robust. Using three subsets of the TI46 speech corpus to benchmark, we elucidate that in terms of isolated word recognition, the analyzed digital LSMs are very promising for future hardware implementation because of their low overhead, good robustness, and high recognition performance.

1. Introduction

Spiking neural networks (SNNs) are the third generation of neural networks. Compared to traditional sigmoidal perceptrons, SNNs possess increased computational power [29] and are biologically more plausible because they model the communication of the temporal information among biological neurons. Recent years have witnessed an increased interest in the theoretical studies of SNNs including bio-inspired learning algorithms [30,16,3,13,37,11] and network structure [17,40]. SNNs closely resemble spiking behavior of biological neural networks with intrinsic temporal information, making it a potentially good model of computation for temporal tasks such as speech recognition [46]. Works targeting practical implementation of SNNs in hardware systems have also emerged [14,41,43,1]. Furthermore, bio-inspired spiking neural networks are shown to have inherent error resilience and fault tolerance [22], a very appealing characteristic for VLSI implementation in highly scaled modern CMOS technologies, for which device reliability and process variability are grand challenges.

Motivated by the anatomical and physiological structure of the cerebral cortex that carries out diverse computational tasks [32], the liquid state machine (LSM), which is more generally termed reservoir computing [26], has been proposed and shown to provide powerful computational capability for many applications [30,31,46,4]. The LSM consists of a reservoir, a recurrent spiking neural network with fixed but randomly chosen connections introduced to preprocess the external input signals, and a group of readout neurons performing classification by further processing and extracting relevant features of the input patterns from the reservoir. With recurrent connections in the reservoir, the LSM can map the input into a high-dimensional space by producing complex nonlinear dynamics in the reservoir, which makes the subsequent classification easier. The decaying dynamic responses of the input signals in the reservoir serve as a transient memory by which critical information about the inputs is captured. As a result, the LSM is especially competitive for dealing with temporal input patterns such as speech signals [12,46].

Compared with other standard methods for recognizing isolated words such as HMM (Hidden Markov Model) [47], template [7] and feature [33] based approaches, LSMs are not only more biologically plausible and general purpose, but also have potential advantages in error resilience when implemented in hardware. HMM-based approaches often use highly tuned acoustic and language models. In contrast, LSMs can be trained merely based upon the statistics of the data presented in the training data set. In addition, compared with other biologically plausible methods such as LSTM neural nets [15], LSMs are more hardware-friendly.

However, the existing works of LSMs either focus only on high-level

[*] Corresponding author.
E-mail addresses: jyyz@tamu.edu (Y. Jin), pli@tamu.edu (P. Li).

http://dx.doi.org/10.1016/j.neucom.2016.11.045
Received 21 September 2015; Received in revised form 16 July 2016; Accepted 22 November 2016
Available online 26 November 2016

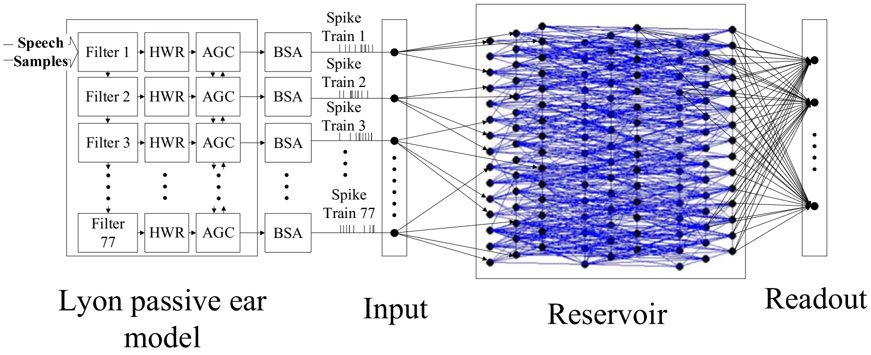


Fig. 1. The LSM-based speech recognition system. 77 channels of spike trains preprocessed by the Lyon passive ear model and BSA algorithm are used as input to the reservoir. Then the reservoir projects the input spikes by a nonlinear transformation to the readout for further processing. Finally, the readout is trained to classify different input signals by their temporal responses in the reservoir.

computational principles [17,35] without a real-world application background or on an application level without much theoretical analysis ([43,46]). [39] studied the design and VLSI implementation of the readout stage for LSMs based on perceptrons and the p-Delta learning algorithm, which were less biologically inspired and only applied to simple two-class recognition and rate-sum retrieval problems. Most importantly, little has been investigated for VLSI hardware implementation of LSMs while considering the optimization and trade-offs involving learning performance, hardware overhead, and error resilience when using real-world challenging applications to benchmark.

To this end, we provide here a systematic examination of performance and robustness issues of LSMs, targeting specifically at speech recognition and digital VLSI implementation. It has been shown recently that digital liquid state machines with biologically plausible training rules can achieve highly competitive performance for isolated word recognition, outperforming several existing state-of-the-art techniques [49]. In this paper, we perform a systematic design space exploration of the LSMs proposed in [49] and show that it is possible to attain good recognition performance while noticeably reducing design complexity. More specifically, we show that recognition performance can be traded off favorably for a potentially significant reduction in reservoir sizes, synaptic weights and membrane voltage resolutions. It shall be noted that these three key network design parameters have a significant impact on the silicon area and power overhead of the VLSI implementation. To shed a deeper light on how these design parameters influence the internal dynamics of the network and finally recognition performance, we use several theoretical measures to characterize the computational power of the LSM as a function of the design parameters. We correlate these theoretical measures with the corresponding real-life speech recognition performance by using the widely adopted TI46 speech corpus [8] as a benchmark. Finally, to evaluate the robustness of the hardware-based LSM, a key design concern for VLSI implementation in modern CMOS technologies, we model various manufacturing and noise induced failure and error mechanisms and show the presented LSMs are in general tolerant to failures and errors. While our study is conducted without referring to a specific digital VLSI implementation, the presented findings are rather general and can provide immediate guidance for designing highly efficient and robust VLSI-based digital liquid state machines.

The rest of the paper is organized as follows: Section 2 provides a brief background of the presented work including the experimental setups. Section 3 introduces the adopted three theoretical measures for estimating the computational power of the LSMs. In Section 4, the LSM design space exploration is presented, and the achievable recognition performance and theoretical measures of computing power as functions of design parameters are shown. The robustness of the targeted LSMs is discussed in Section 5. In Section 6, the performance study of the LSM on two other datasets is presented to provide a more complete understanding on the trade-offs between design cost and performance. Finally, the key insights on network design and robustness obtained from this study are summarized in Section 7.

2. Background

2.1. Speech recognition using the liquid state machine

The liquid state machine consists of a random recurrent neural network (RNN), or a reservoir, and a tunable readout layer that is fed by the reservoir. The sustained temporal dynamics activated by the inputs allows the reservoir to memorize the past inputs, making it possible for the readout layer to extract and process the context information [26,30]. With the inherent advantages in temporal pattern processing, the liquid state machine, and more broadly reservoir computing, may be well suited for a number of classification tasks such as speech recognition. Practically, the readout layer can be trained into a linear classifier, which greatly simplifies the training task of reservoir computing.

The LSM based speech recognition can be constructed as depicted in Fig. 1 ([46,49]). Speech signals are first preprocessed by the Lyon passive ear model [28] then encoded into spike trains by the BSA algorithm ([44,46]), and fed into a group of randomly selected neurons in the reservoir.

Input signals are processed in two steps. The first step takes place in the reservoir, where an incoming spike train $u(t)$ gets mixed and mapped to the responses of the reservoir, represented by a higher dimensional transient state, rendering complex patterns more likely to be separable [6]. In the second step, the responses of the reservoir are projected to the readout through plastic synapses. For each readout neuron at time t, the net current it receives from the reservoir is given by:

$$I_o(t) = \sum_i w_{oi} \cdot f_i(t) = \sum_i w_{oi} \cdot f_i[(u(t)], \tag{1}$$

where $f_i(t)$ is the response of the i^{th} neuron in the reservoir, and w_{oi} is the synaptic weight between the i^{th} reservoir neuron and the readout neuron. The integrated net current over [0, T] is:

$$\int_0^T I_o(t) = \sum_i w_{oi} \cdot \int_0^T f_i(t) = \sum_i w_{oi} \cdot \int_0^T f_i[u(t)]. \tag{2}$$

As the integrated net current to each readout neuron is a linear combination of integrated outputs coming from all reservoir neurons, each readout neuron can be treated as a linear classifier of the responses of the reservoir, which can be considered falling into a feature space. Ideally, only reservoir responses produced by input signals from the same class are expected to activate the correspondent readout neuron. Therefore, conceptually in the feature space, the hyperplane defined by all w_{oi}'s separates these inputs from others.

Finally, the task of speech recognition is a problem of solving all these linear classification problems by tuning these synaptic weights between the reservoir and readout layer. This can be achieved by applying a learning algorithm.

2.2. Learning algorithm

Hebb's postulate, which claims that neurons fire together wire together, was proposed and widely accepted ([18]). In particular, if the firing activity of neuron i tends to excite/inhibit the firing activity of neuron j, the synapse connected from i to j will be potientiated/depressed. Based on this principle, a number of biologically plausible learning algorithms can be used for training the readout layer of an LSM. In the literature, temporal encoding has been adopted in several learning rules, e.g. ReSuMe [37], I-Learning [11], tempotron [16] and SPAN [34]. Firing rate encoding has also been adopted for developing an abstract learning rule [3].

The focus of this paper is not to study a specific learning algorithm under the context of LSMs, but to examine the performance and robustness of LSMs when a typical biologically plausible learning rule is adopted. In other words, the emphasis of this work is placed upon the key dynamical and network characteristics of the liquid state machine rather than a behavior of a given learning rule. For this purpose, we have opted to use a hardware-friendly learning rule [49], which is motivated by the abstract rule of [3]. We succinctly describe the key features of this adopted rule below.

The adopted rule is based on the principle of Hebbian learning, under which the goal of the learning process is to modulate the activity of the readout neurons according to the desired level, and then tune the weights of plastic synapses correspondingly. More precisely, when a certain readout neuron is expected to fire actively, we drive its firing activity to a high level, with the help of certain teacher signals that implement supervised learning; while at the same time, we inhibit the firing activities of other readout neurons.

To implement the adopted learning rule, we define the calcium variables c_r and c_d to indicate the actual and desired firing activity of a postsynaptic neuron, respectively. To distinguish highly active neurons from inactive ones, a threshold c_θ of the calcium variables is imposed - if the calcium level is higher (lower) than the threshold, the neuron is considered to be at a high (low) activity. The update of plastic synapses only happens when a presynaptic neuron fires and the actual calcium concentration c_r of the postsynaptic neuron is higher (or lower) than c_θ:

$$\begin{cases} w_{ij} \rightarrow w_{ij} + \Delta w \text{ with prob. } p_+ \\ \text{iff } neuron_j \text{ fires } \& \ c_\theta < c_r < c_\theta + \Delta c \\ w_{ij} \rightarrow w_{ij} - \Delta w \text{ with prob. } p_- \\ \text{iff } neuron_j \text{ , fires } \& \ c_\theta - \Delta c < c_r < c_\theta, \end{cases} \tag{3}$$

where w_{ij} is the weight of the plastic synapse from neuron j to neuron i, Δw is the potentiation/depression granularity, and the parameter Δc is used for good generalization performance. Potentiation or depression of synapses happens with the probability of p_+ or p_-, respectively. The potentiation/depression of synapses is only activated when c_r is in the specific ranges specified by c_θ and Δc. This mechanism is used to avoid saturation of the plastic weights. This learning process is visualized in Fig. 2.

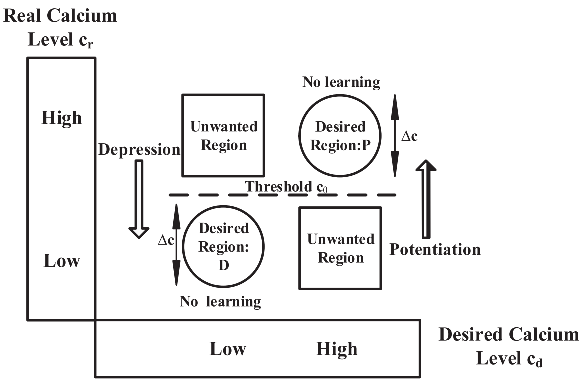


Fig. 2. Learning process of the LSM. Four regions in the diagram show how different combinations of c_d and c_r of the postsynaptic neuron determine the synaptic plasticity (c_d and c_r are only defined for readout neurons). The two arrows represent depression and potentiation implemented by the teacher signals, driving the activity of a neuron to the desired region (highly active or inactive), where the corresponding synaptic weights are tuned. More precisely, potentiation of the corresponding synapses happens in the desired region marked by “P” and depression happens in the desired region marked by “D”.

Teacher signals are introduced to the readout layer to implement supervised learning. A teacher signal injects a large positive or negative current to the corresponding neuron for the purpose of modulating its real calcium concentration c_r to the desired level of calcium concentration c_d. More specifically, when the samples from a certain input speech class are presented, the readout neuron corresponding to this class is expected to be highly active (i.e. c_d is high). Therefore, its firing activity and calcium concentration are both driven to a high level by the teacher signals, whereas other readout neurons are supposed to be inactive with a low c_d. Consequently, these neurons are driven to the highly inactive region where c_r is low by their teacher signals. The combined use of the learning rule (shown in Eq. (3)) and the teacher signals potentiates or suppresses the plastic synapses in a way that leads towards separation of different input classes.

2.3. Digitized simulation models

In terms of simulation of LSMs, we adopt the digitized leaky-integrate-and-fire (LIF) model for neurons and second order response model for synapses adopted from [49]. The dynamics of the membrane voltage of a neuron can be described by the following equation:

$$V_m^n = V_m^{n-1} - \frac{V_m^{n-1}}{\tau_m} + I_{syn}R_{syn} + I_t R_t, \tag{4}$$

where the superscript of V is the index of the time step, V_m and τ_m are the membrane voltage and the time constant of its first-order dynamics, respectively, I_{syn} models synaptic input current, I_t models the input current from the teacher signal, and R_{syn} and R_t model the synaptic resistance and the input resistance associated with the teacher signal. If the membrane voltage (V_m) of a neuron reaches or exceeds the threshold voltage V_{th}, the neuron fires and its membrane voltage is reset to the resting potential V_{rest}. There is an absolute refractory period τ_{refrac} associated with each spike, during which a fired neuron cannot fire again.

The calcium level of a neuron is used to model its firing activity to trigger the learning rule. The dynamics of the calcium level is modeled as:

$$C^n = C^{n-1} - \frac{C^{n-1}}{\tau_c} + \sum_i \delta_{T^n, T_i}. \tag{5}$$

Here τ_c is the time constant of this first-order model, i is the index of the spike emitted from this neuron, and $\delta_{x,y}$ is the Kronecker delta whose value is 1 if x=y, and 0 otherwise. T_i is the time when the neuron transmits its i_{th} spike and T^n is the simulation time.

The synaptic current I_{syn} to each neuron is modeled as:

$$I_{syn} = \sum_i \sum_j W_i \cdot Syn(T^n, T_{ij} + D_{ij}), \tag{6}$$

where i and j are indices of the presynaptic neurons and the spikes, respectively. Specifically, W_i represents the weight of the synapse that connects to the i^{th} presynaptic neuron. T_{ij} is the firing time of the j^{th} spike emitted from the i^{th} presynaptic neuron, and D_{ij} is the corresponding synaptic propagation delay. $Syn(\cdot)$ is the digitized second-ordered dynamic response of a synapse to an incoming spike:

$$Syn(T^n, T_{ij} + D_{ij}) = \frac{1}{\tau_{s1} - \tau_{s2}} \cdot e^{-\frac{T^n - T_{ij} - D_{ij}}{\tau_{s1}}} \cdot S(T^n - T_{ij} - D_{ij}) - \frac{1}{\tau_{s1} - \tau_{s2}} \cdot e^{-\frac{T^n - T_{ij} - D_{ij}}{\tau_{s2}}} \cdot S(T^n - T_{ij} - D_{ij}). \quad (7)$$

τ_{s1} and τ_{s2} are two time constants of the model. $S(\cdot)$ is the unit step function. The term $\frac{1}{\tau_{s1} - \tau_{s2}}$ normalizes the response function such that the integrated response of each spike is normalized to one.

The setup of the parameters in the neuron and synpatic model can be found in Section 4.

2.4. Adopted speech benchmarks

In order to benchmark the performance of our LSMs for speech recognition, three subsets of TI46 speech corpus ([8]) are used. The speech samples in TI46 were collected in a low noise sound isolation booth using an Electro-Voice RE-16 Dynamic Cardiod microphone at 12.5 kHz sample rate.[1]

The first benchmark is widely used in testing the performance of reservoir computing based speech recognition ([32,46,45,12]). It contains isolated word utterances of 5 different speakers. 10 different utterances of each word from "zero" to "nine" are recorded for each speaker. Thus, this benchmark contains 500 speech samples. This is the main benchmark used to study the performance and robustness of the LSMs in this paper. The second benchmark includes 1,000 utterances of isolated digits in the training set of the TI46 speech corpus. This large subset contains speech samples from 10 speakers. For each speaker in the subset, there are 10 recorded samples of each spoken digit. The third subset contains 10 utterances of each English letter from "A" to "Z", which were recorded from a single speaker. There are 260 samples in the third benchmark.

These speech samples are preprocessed by Lyon's ear model, which consists of three prepassing stages: a band-pass filter bank, a half wave rectifier with automatic gain control ([28]), and BSA, an algorithm converting time domain input signals into spike trains ([44,46]). The average input spike rates of different spoken digits in the first benchmark are illustrated in Table 1. Each reported rate is the average rate of different recordings of the same digit. To visualize these speech samples, we show several representative input spike trains of the words "zero", "three", "six" and "nine" with the corresponding reservoir responses in Fig. 3. The resolutions of synaptic weights and membrane voltages in the reservoir are 10 bits and 16 bits, respectively. Other detailed parameter settings can be found in Section 4.

It is worth noting that although the spike rates of input spikes remain roughly the same for different utterances, it can be observed from Fig. 3 that the reservoir is able to produce responses with distinctive spatio-temporal characteristics in response to different input speech samples. It can be expected that the mapping from the space of input spikes to the higher-dimensional space of reservoir responses contributes to differentiability across different speech classes.

2.5. Network and training setup

The spiking network and training are set up according to [49]. As illustrated in Fig. 1, the reservoir has a grid structure. 20% of neurons in the reservoir are randomly chosen to be inhibitory while the rest are excitatory. The connectivity in the reservoir is constructed randomly under a distribution such that the wiring probability of any two neurons (N_i and N_j) drops exponentially in the distance between them ([30]):

Table 1
Average input spike rates for different words in the first benchmark. Each spike rate is the average rate of different recordings of the same digit.

Digit	0	1	2	3	4	5	6	7	8	9
Spike rate (*kHz*)	9.0	7.0	8.5	7.7	7.2	8.2	7.5	7.5	5.7	7.6

$$P_{connect}(N_i, N_j) = k \cdot e^{-\frac{D^2(N_i, N_j)}{r^2}} \; (i \neq j), \quad (8)$$

where $D(N_i, N_j)$ is the Euclidean distance between these two neurons, r is chosen to control the exponential decay of the probability, and k is a constant depending on the neuron type. The parameters are chosen according to the values suggested by [49].

In the network, each input spike train generated in the preprocessing stage is sent to four randomly chosen reservoir neurons through synapses with fixed weights randomly chosen to be W_{max} or W_{min}, where W_{max} and W_{min} are maximum and minimum synaptic weights used in the simulation, respectively. Reservoir neurons are fully connected to each readout neuron by plastic synapses, whose weights are randomly initialized between W_{max} and W_{min}. The plastic synapses are trained by the adopted learning algorithm. The synaptic weights in the reservoir are fixed according to the neuron type. The detailed parameter settings of synaptic connections can be found in Section 4.

To test the performance of the various LSM designs considered in this paper, we adopt a 5-fold cross validation scheme to determine the speech recognition rate. In this setup, all speech samples are randomly divided into five groups. Based on these samples, a fixed LSM is trained and tested for five times with different training and testing datasets. For the i_{th} ($i = 1, 2, 3, 4, 5$) time, the i_{th} group is used for testing and the remaining data for training. The recognition decision is made after each testing speech sample is played. At this time, the readout neuron that has fired most frequently is the winner and its associated class label is deemed to be the classification decision of the LSM. Finally, the five classification rates obtained in the testing stage are averaged as final performance measure.

2.6. Performance of the base-line LSM

We set up a baseline LSM as a reference for the presented design space exploration. There are 135 neurons in this baseline LSM and the resolutions for membrane voltages and synaptic weights are set to be 16 bits and 10 bits, respectively. The detail of other parameter settings will be discussed in Section 4. The best recognition rate of this LSM is 99.2% based upon the first benchmark described in Section 2.4.

3. Theoretical measures of computational performance

To gain insights into the LSM network dynamics and its relation to learning performance, we adopt three theoretical measures of computational power to analyze the presented LSMs. First, we measure the "fading memory" of the reservoir of a given LSM ([30,32]), characterizing how well the reservoir "memorizes" temporal input patterns. From a dynamic system point of view, we examine the operating regime of an LSM and quantify its distance to the so-called edge between order and chaos ([2]). Finally, from a task-oriented point of view, we analyze the LSMs in terms of their separation property and generalization capability ([24]).

3.1. Fading memory - how well the LSM can remember temporal input patterns

First of all, we theoretically estimate how well the dynamics in the

[1] More information of TI 46 is available from the Linguistic Data Consortium (https://catalog.ldc.upenn.edu/LDC93S9).

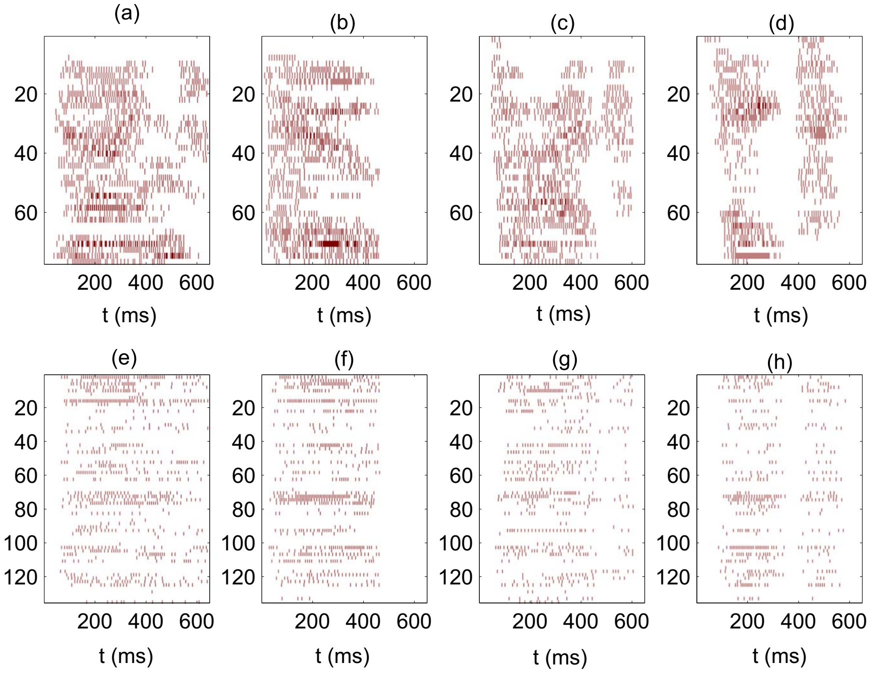


Fig. 3. (a)–(d) are the input spike trains of the utterances “zero”, “three”, “six” and “nine”, with the y-axis showing the indices of the input channels. (e) - (h) show the corresponding neuron activities in the reservoir for the words “zero”, “three”, “six’ and “nine”, respectively. The y-axis represents the indices of the reservoir neurons.

reservoir helps to “memorize” different input patterns by measuring its fading memory[2] (Fig. 4).

By observing the responses in the reservoir, we intuitively approximate the fading memory. As proposed in [30], one way to empirically measure fading memory is to count the number of firing neurons and calculate the duration of firing activity in the reservoir after injecting random temporal signals into the LSM. Long lasting and strong firing activity is usually desirable because it implies the strong memory capacity possessed by the reservoir.

3.2. Edge of chaos - whether or not the LSM operates on the transition boundary

Second, we theoretically analyze an LSM from a dynamic system point of view. Related studies ([21,23]) suggest that dynamic systems operating near the phase transition between order and chaos (i.e. “edge of chaos”) possess a good amount of computational power. To determine the ordered and chaotic regimes for discrete dynamical systems driven by online inputs, [2] proposed to track the evolution of state difference resulting from two close initial states while the system is driven by the same input. The state difference of a chaotic system is highly amplified while that of an ordered system vanishes quickly. One can quantitatively analyze the phase transition by Lyapunov exponents ([27]). We look for the exponent λ that is determined by

$$\delta_{\Delta T} \approx \delta_0 \cdot e^{\lambda \Delta T},\ \delta_0 \to 1,\ \Delta T \gg 1. \tag{9}$$

Here δ_0 represents the initial state difference at time 0 and $\delta_{\Delta T}$ is state separation at time ΔT. As depicted in Fig. 5, $\lambda > 0$ suggests that the system is chaotic while $\lambda < 0$ indicates an ordered system. The dynamical system sits on the transition boundary if its λ is equal to 0. In our measurement, we define the state of the LSM as a binary vector $s(t) = [s_1(t), s_2(t), \ldots, s_n(t)]$, with s_i setting to one when the i_{th} reservoir neuron fires at time t. The Hamming distance between two states is defined as the state difference.

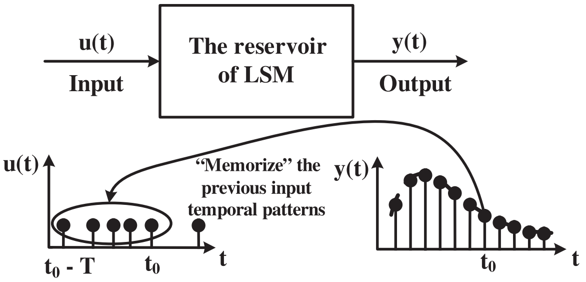


Fig. 4. Fading memory of an LSM. The temporal input stream $u(t)$ is transformed by the reservoir into a high-dimensional signal $y(t)$, which holds the information about the recent history of the input $u(t)$ ($[t_0 - T, t_0]$).

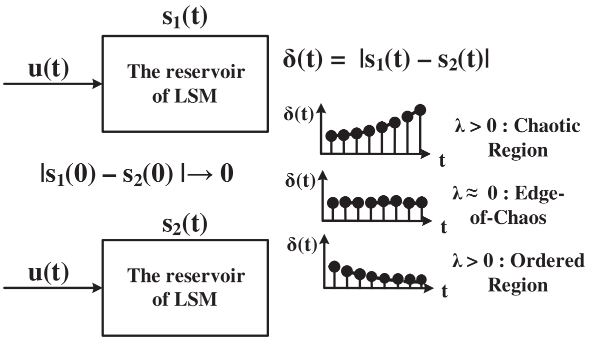


Fig. 5. Edge of chaos of an LSM. With the same input $u(t)$ and two initially close states ($s_1(t)$ and $s_2(t)$), the difference between two states is recorded and measured as the dynamics of the LSM evolves. The Lyapunov exponent λ theoretically reveals whether or not the system is on the phase transition boundary.

3.3. Separation and generalization - how well the LSM performs for a given task

Third, as illustrated in Fig. 6, we investigate the theoretical computational power by quantitatively analyzing two essential properties, i.e. separation and generalization, of an LSM to characterize its performance from an application perspective ([24]). Separation of the

[2] see [30] for a detailed definition.

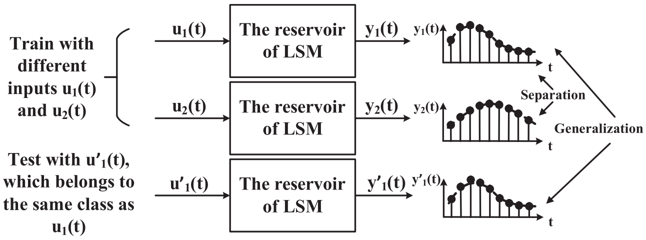


Fig. 6. Generalization and separation of an LSM. When an LSM is trained with two different inputs ($u_1(t)$ and $u_2(t)$), the outputs of the reservoir ($y_1(t)$ and $y_2(t)$) are expected to be distinct because of the separation property. While tested with the input $u'_1(t)$ which belongs to the same class as $u_1(t)$, the output of the reservoir $y'_1(t)$ should be similar to $y_1(t)$ because of good generalization.

reservoir reflects the kernel-quality of the neural circuit and generalization measures how well the reservoir can generalize a learned function to new input streams.

As suggested in [24], the separation and generalization properties of the reservoir are estimated by computing the rank of an $n \times m$ matrix M, where n is the number of state variables of the reservoir, m is number of inputs, and each column of M is the state vector $x_{u_i}(t_0)$ under the incoming input stream u_i at a fixed time point t_0. To measure separation property in our case, randomly generated input streams are used; while for approximating the generalization capability, application-specific speech signals are used. According to [24], a large difference between r_S (the rank estimating the separation capability) and r_G (the rank representing the generalization capability) is usually a good indicator of strong computational power with respect to the specific task at hand.

4. Performance of LSMs and Its dependencies on key model/design parameters

The performance of an LSM immediately depends on several network design parameters and these parameters in turn greatly determine the resulting hardware implementation cost. When it comes to the application of speech recognition, one major design choice that needs to be made is what level of precision should be maintained to guarantee the good performance of LSMs when emulating the behaviors of neurons and synapses. This question is meaningful both from a biological modeling and an engineering point of view. First, nervous systems in nature exhibit trial-to-trial variability ([10]) in the present of intrinsic noise, therefore it is not necessary to have extremely high precision in modeling the dynamics of neurons and synapses. Second, from an engineering perspective, it is critical to choose an appropriate level of precision for the targeted application because excessive precision leads to unnecessary increase in implementation overhead.

To achieve the above goal, we conduct behavior-level simulations to study the performance of LSMs with a broad range of design parameters. The considered parameters, including ones for the digitized LIF neuron model, synaptic model, and learning rule, are summarized in Tables 2–4. Some of the parameters have been adopted from [49]. In Table 2, two extra parameters, V_{max} and V_{min}, are imposed as the upper bound and lower bound upon the membrane voltage V_m, for the purpose of discreteness. The same discreteness is also applied to synaptic weights and calcium concentrations. In Table 3, E and I indicate excitatory and inhibitory neurons, respectively. $E \rightarrow I$ denotes connections from excitatory presynaptic neurons to inhibitory postsynaptic neurons. In Table 4, the strength of the teacher signal I_t is set to be $\frac{V_{th}}{R_t}$ for potentiation, and $-\frac{3V_{th}}{4R_t}$ for depression.

Table 2
Parameters of the neuron model.

Parameter	Value
Threshold voltage V_{th}	20 mV
Resting potential V_{rest}	0 mV
Time constant τ_m	32 ms
Time constant τ_c	64 ms
Refractory period τ_{refrac}	2 ms
Upper bound of membrane voltage V_{max}	32 mV
Lower bound of membrane voltage V_{min}	−32 mV
Granularity of membrane votlage δV	$\frac{V_{max} - V_{min}}{2^{n_{mem-bit}}}$ [a]

[a] $n_{mem-bit}$: the resolution of the membrane voltage.

Table 3
Parameters of the synaptic model.

Parameter	Value	
	type	value
Fixed weights in the reservoir	$E \rightarrow E$	3
	$E \rightarrow I$	6
	$I \rightarrow E$	−2
	$I \rightarrow I$	−2
Upper bound of synaptic weights W_{max}	$E/I \rightarrow E/I$	8
Lower bound of synaptic weights W_{min}	$E/I \rightarrow E/I$	−8
Time constant τ_{s1} of the second-order synaptic dynamics	$E \rightarrow E/I$	4
	$I \rightarrow E/I$	8
Time constant τ_{s2} of the second-order synaptic dynamics	$E \rightarrow E/I$	4
	$I \rightarrow E/I$	2
Synaptic propagation delay D_{ij}	1	
Synaptic resistance R_{syn}, R_t	1Ω	
Granularity of synaptic weights δW	$\frac{W_{max} - W_{min}}{2^{n_{syn-bit}}}$ [a]	

[a] $n_{syn-bit}$: the resolution of the synaptic weight.

Table 4
Parameters used in the learning rule.

Parameter	Value
Granularity of calcium level δc	$2^{n_{cal-bit}-4}$ [a]
Upper bound of calcium level c_{max}	$16 \times \delta c$
Lower bound of calcium level c_{min}	0
Threshold of calcium level c_θ	$5 \times \delta c$
Generalization parameter Δc	$3 \times \delta c$
Teacher signal strength I_t	$\frac{V_{th}}{R_t}$ or $-\frac{3V_{th}}{4R_t}$
Learning probability p_+,p_-	$\frac{0.004}{2^{n_{syn-bit}-4}}$
Potentiation/depression granularity ΔW	δW

[a] $n_{cal-bit}$: the resolution of the calcium level.

To attain the simulated recognition performance of each sample point of the design parameter space, five randomly generated LSMs are trained and tested for speech recognition. To optimize the performance, we train the LSMs for multiple iterations. For each generated LSM, the best recognition rate is computed after multiple training iterations and we average five obtained best recognition rates of the LSMs as the reported recognition rate. The standard deviation (SD) of the five best recognition rates is measured to report variation of recognition performance.

To comprehensively study the relation between a broad range of parameters and performance, we investigate from three aspects: resolution of the neuron model, resolution of the synaptic model and size of the reservoir. As mentioned in Section 2.6, we use the base-line LSM as the reference design (see Table 5 for the key parameter settings) and apply the first adopted benchmark for conducting the performance study. In addition to simulation, we also theoretically characterize the computational power of the targeted LSMs under various parameter settings.

Table 5
Key design parameters of the reference design.

Design Parameter	Neuron Type	Resolution /Value
Calcium Level	Readout[a]	14 bits
Membrane Voltage	Reservoir	16 bits
	Readout	16 bits
Synaptic Weight	Reservoir	10 bits
	Readout	10 bits
Size of Reservoir	N.A.	135 neurons

[a] Reservoir neurons do not have this variable because no weight adaption happens in the reservoir.

4.1. Resolution of membrane voltage and calcium level

First of all, we examine how the precision of membrane voltage and calcium level of the neurons can influence recognition performance. While reservoir and readout neurons have different roles in the LSM, we separately analyze these two types of neurons. The performance of the LSM with different resolution settings is plotted in Figs. 7 and 8. As mentioned in Section 4, each plotted point in Figs. 7 and 8 is the averaged recognition rate of the five recognition rates obtained by training and testing five randomly generated LSMs with different random seeds for the generation of random connections inside the reservoir. Each error bar in the figure represents the standard deviation (SD) of the five recognition rates with its length being 2 × *SD*.

The curve with circles in Fig. 7 shows the simulated recognition rates of LSMs with a decreasing resolution of membrane voltage for reservoir neurons while the other design parameters are fixed according to Table 5. The simulation results suggest that the recognition performance only degrades slightly when the precision of membrane voltage for reservoir neurons is reduced down to 6 bits. This phenomenon may be understood by noticing that under a low membrane voltage resolution, fixed and recurrent connections in the reservoir may still be able to propagate the firing activities initialized by a few neurons to create rich dynamics in the network. But the resolution cannot be too low (e.g. below 3 bits or 4 bits) because the firing activity of the reservoir will not reflect the critical information of speech samples well under a very low resolution. Similarly, a low resolution for the membrane voltage of readout neurons will not cause much performance degradation. As shown in the curve marked with “x” in Fig. 7, LSMs start to perform poorly only when the resolution drops below 3 bits, where the average recognition performance is about 40%. The result suggests that the activation of the correct winning readout neuron for a given input speech is not very sensitive to the bit resolution used for the membrane voltage of readout neurons.

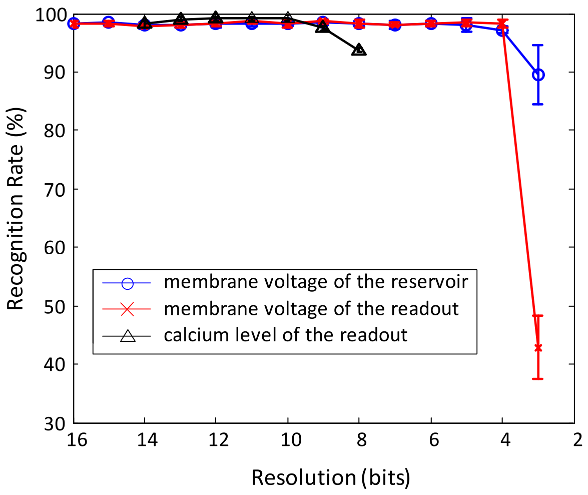


Fig. 7. Performance of the LSMs drops as a function of the decreasing bit-resolutions of the membrane voltage and calcium level.

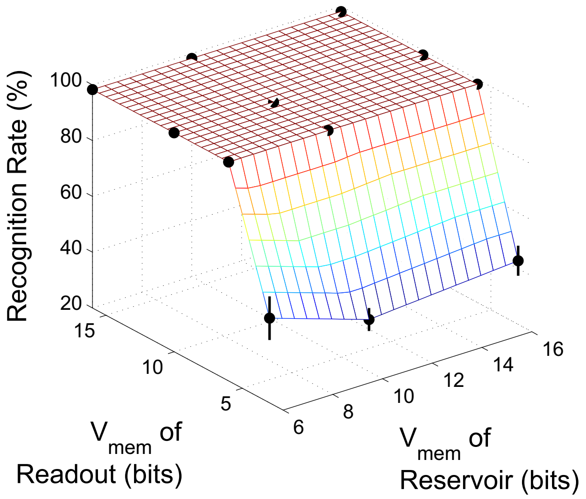


Fig. 8. Performance of the LSMs with different combinations of reservoir and readout membrane voltage resolutions.

To have a more comprehensive study of performance sensitivity to membrane voltage resolutions, we test the performance under different combinations of resolution settings for the reservoir and readout (shown in Fig. 8). The performance is not sensitive to wide range variations of membrane voltage resolutions, particularly for the case of the resolution of reservoir neurons.

The calcium level, used in the learning algorithm, is another important parameter associated with readout neurons. The curve denoted by triangles in Fig. 7 manifests that the recognition performance degrades noticeably as the resolution of the calcium level is lower than 10 bits. This suggests that the calcium concentration plays an important role in learning and hence a fine resolution may be needed to accurately tune the plastic synaptic weights of readout neurons to achieve good performance. We also examine various combinations of the calcium level threshold (c_θ) and generalization parameter (c_Δ), which are two parameters of the learning rule (see Table 4), to investigate their impacts on the performance. As shown in Fig. 9, the choices of the two learning parameters within the considered range have no significant impact on the performance.

In addition to the simulated recognition performance presented above, we further use the three theoretical measures of computational power mentioned before to characterize the performance impacts of the

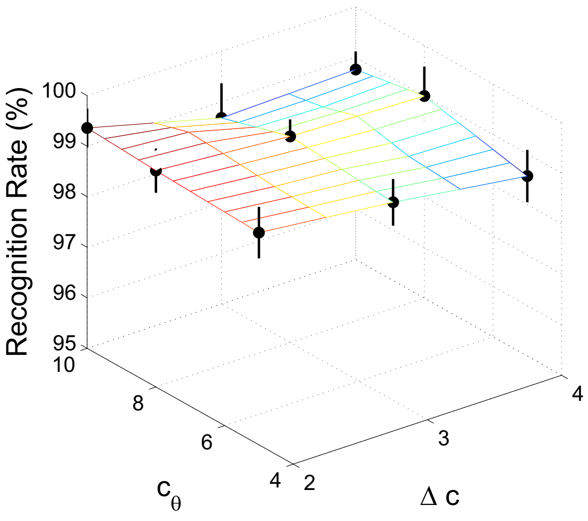


Fig. 9. Performance of the LSMs with different combinations of parameters in the learning rule.

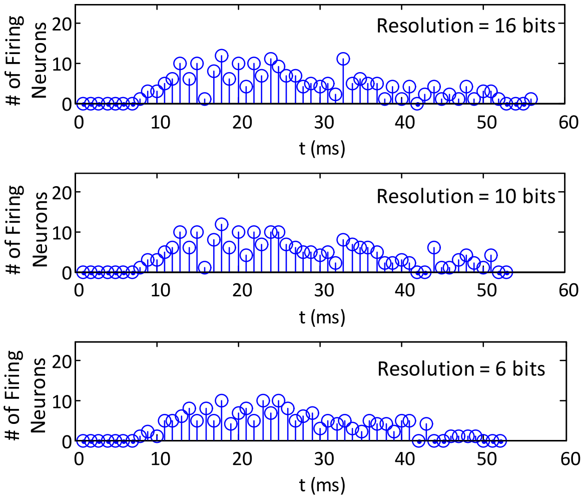


Fig. 10. Firing activities (fading memory) in the reservoir when the resolution of membrane voltage for the reservoir is reduced. Three reservoirs with descending resolutions are tested by injecting 77 random input spike trains that are all ended at 23 ms. A lowered resolution does not necessarily lead to reduced fading memory.

resolution of membrane voltage. As will be seen, these theoretical measures provide largely consistent performance evaluations of the LSMs while offering additional understanding of network dynamics and associated computing power. Note that the three theoretical measures aim at examining the computational power resulting from the dynamics created in the reservoir, which focus on the following analysis.

Fading Memory. To measure the fading memory, we inject 77 random temporal spike trains ending at 23 ms into the reservoir and record the responses of the reservoir (i.e. the number of firing neurons and the duration of the firing activities). The impacts of lowering precision of reservoir neurons' membrane voltage on fading memory are shown in Fig. 10. The results clearly suggest that the reservoir has large fading memory even when the resolution of its membrane voltage is reduced to 6 bits. The observation is in agreement with Fig. 7 and once again implies that the implementation of reservoir neurons can be simplified to lower hardware overhead without any significant degradation of performance.

Edge-of-Chaos. Applying an approach similar to that is adopted in [42], we introduce an infinitesimal initial state difference to characterize the operating regime of an LSM by using a pair of two slightly different inputs. Two such input pairs are used in order to eliminate the randomness introduced by the choice of specific inputs. For the first input pair, the only difference between the two input spike trains results from one missing spike at 24 ms from one spike train. For the second input pair, the difference between two spike trains is due to one missing spike at 42 ms from one of the input spike trains. The two slight different inputs are only used to introduce an initial state difference and after that a future state difference is used to compute the Lyapunov exponent. By independently feeding the two different inputs to an LSM, the resulting state difference $\delta_{\Delta t}$ is observed at a future time point. To be specific, $\delta_{\Delta t}$ is measured at $\Delta t = 300$ ms across the two slightly different input streams, where Δt is the elapsed time from when the initial state difference occurs to the observation time point. Then the Lyapunov exponent λ is determined by ([42])

$$\lambda = \frac{1}{\Delta t}\ln\left(\frac{\delta_{\Delta t}}{\delta_{ini}}\right), \tag{10}$$

where δ_{ini} is the initial state difference introduced by the two slight different inputs.

Fig. 11 shows how the state separation temporally evolves due to a small input difference for three LSMs with a descending membrane voltage resolution. It is obvious that although the precision is reduced, the state difference remains roughly at the same level. In other words, the reservoir's membrane voltage resolution might not significantly influence its dynamics, which is further confirmed by the calculated Lyapunov exponent λ shown in Table 6.

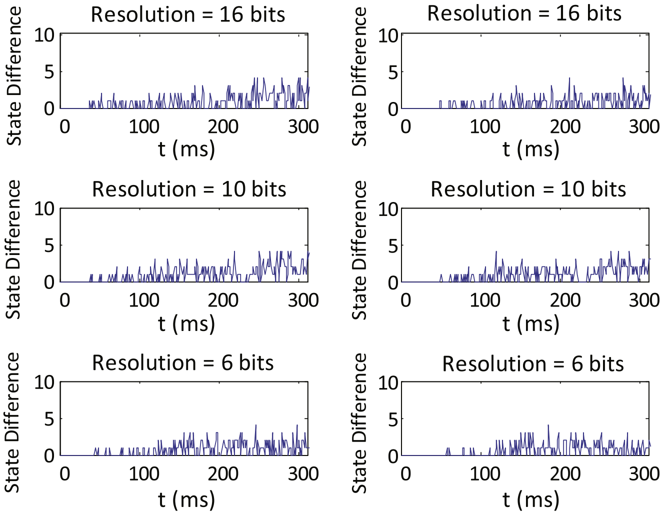


Fig. 11. State separation under different resolution settings for membrane voltage in the reservoir. The temporal evolution of the Hamming distance between the two resulting states $x_u(t)$ and $x_v(t)$ is shown. For the sub-figures on the left and right sides, the input u differs from input v due to only one missing spike at time 24 ms and 42 ms, respectively.

Table 6
Lyapunov exponents of LSMs with various resolutions of reservoir neurons' membrane voltage. The Lyapunov exponents λ_1 and λ_2 are measured under two different input pairs for which the input difference is introduced by a missing spike at 24 ms and 42 ms, respectively. The reported recognition rates of the LSMs under the three resolution settings are 98.16%, 98.24% and 98.28%, respectively.

LSM	Resolution	λ_1	λ_2
1	16 bits	0.36	0.30
2	10 bits	0.32	0.28
3	6 bits	0.22	0.26

As can be seen from Table 6, the Lyapunov exponents of the three LSMs are relatively small and close to zero, indicating that they operate in a region that is close to the transition boundary, which is consistent to the corresponding good recognition performance. In addition, the calculated Lyapunov exponents suggest that a low resolution of membrane voltage may be sufficient for achieving good recognition performance.

Separation and Generalization. We use the method for estimating the separation and generalization capabilities mentioned in Section 3.3. After applying 500 randomly generated input streams and 500 application specific speech signals, we measure the ranks r_G and r_S respectively of the matrix M at five fixed time points from 394 ms to 399 ms and report the maximum obtained ranks for estimation of separation and generation. The rank difference Δ_{SG} between r_S and r_G is calculated as shown in Table 7. As shown in Table 7, interestingly, in the range which is considered for the membrane voltage resolution of the reservoir, lowering the resolution does not affect the computational

Table 7
Estimated separation and generalization capabilities of the LSM as a function of reservoir neurons' membrane voltage resolutions. The reported recognition rates of the LSMs under the three resolution settings are 98.16%, 98.24% and 98.28%, respectively.

LSM	Resolution	r_S	r_G	Δ_{SG}
1	16 bits	135	101	34
2	10 bits	135	102	33
3	6 bits	135	99	36

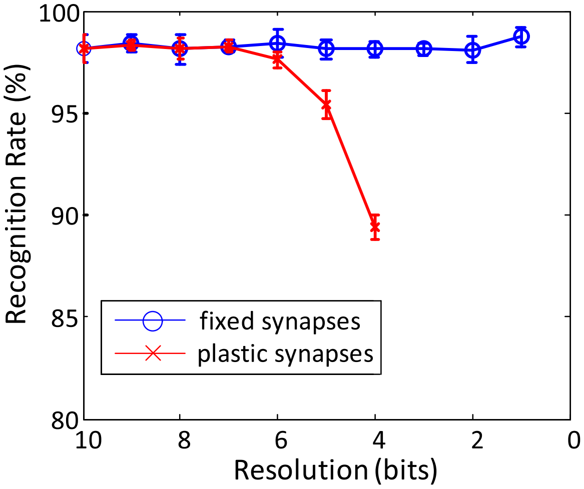


Fig. 12. Performance of the LSMs degrades as a function of the reduced bit-resolutions of the synaptic weights.

capability very much. The approximated computational ability reconfirms that the good classification performance might be achieved given the low resolution of the reservoir neurons' membrane voltage.

4.2. Resolution of synaptic weights

Here we study how the resolution for synaptic weights affects the overall performance. Since synapses within the reservoir have fixed efficacy while synapses connected to the readout are plastic, we consider them separately when changing the resolution.

First, we take a close look at the synapses in the reservoir. The curve with circles in Fig. 12 shows the recognition rates of LSMs with different resolutions of synaptic weights for the reservoir synapses while the resolutions for other parameters are fixed according to Table 5. Clearly, reducing the resolution of fixed synaptic weights has little impact on performance. This observation may be understood by recalling that no synaptic weight adaption takes place in the reservoir and the main functionality of the reservoir is to create rich dynamics to map the input to a higher dimensional space. As a result, low-resolution or even binary synapses may be sufficient for the reservoir.

However, in terms of the plastic synapses between the reservoir and the readout, fine resolution is desirable because synaptic weights with high precision guarantee the accuracy of adjusting the location and orientation of the hyperplane implemented by the linear readout. We examine how the precision of plastic synaptic weights influences performance experimentally. In the simulations, 10-bit synapses are used in the reservoir. The performance degradation of LSMs with different resolution settings for plastic synapses can be seen in Fig. 12. We conclude that 8-bit resolution is needed for efficacy of plastic synapses because further reduction will cause a fairly large performance loss.

We vary the resolutions of fixed and plastic synaptic weights together to obtain a complete picture of how synaptic weight resolutions can affect performance. The simulation results shown in Fig. 13 reconfirm that the resolution of fixed synapses has a limited effect on the performance while the plastic synapses do immediately affect the overall recognition rates.

In the following, we use the three theoretical measures of computational power to correlate with the above simulated recognition performance. The same experimental setup is used here as what is mentioned in Section 4.1.

Fading Memory. After injecting 77 random spike trains into the reservoir and counting the number of fired neurons and measuring the duration of the response, we obtain the fading memory of the reservoir shown in Fig. 14. As depicted in Fig. 14, the synaptic resolution in the reservoir has limited influence on fading memory, which explains why binary synapses can be used in the reservoir for reducing the complexity of LSMs.

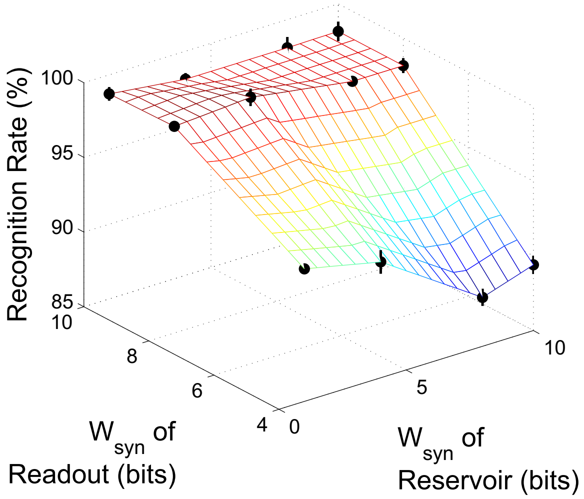


Fig. 13. Performance of the LSMs with different combinations of resolutions of fixed and plastic synaptic weights.

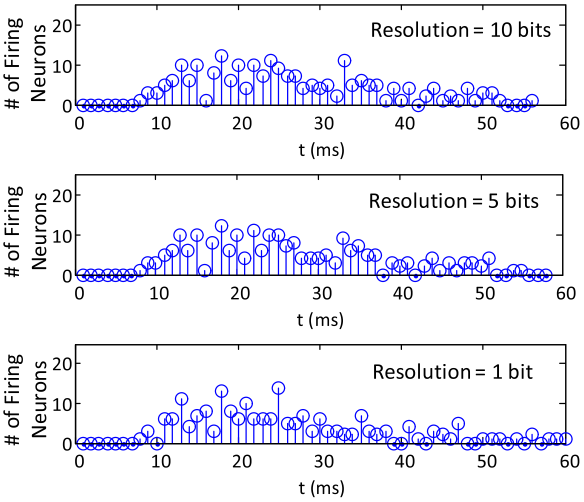


Fig. 14. Firing activity in the reservoir when the resolution of synaptic weights for fixed synapses gets reduced. Similarly to the results shown in Fig. 10, reduction of synaptic weights for the reservoir does not affect fading memory significantly.

Edge-of-Chaos. Two slightly different input streams are injected into the same reservoir and Fig. 15 shows how state divergence temporally evolves due to a small input difference for three LSMs with a descending resolution of the fixed reservoir synapses. By lowering the precision of synaptic weights, we observe that the state difference decreases slightly. The characteristics of the reservoir dynamics are reflected by the Lyapunov exponents λ shown in Table 8.

The computed Lyapunov exponents in Table 8 are relatively small and close to zero, suggesting that the corresponding dynamics of the all three reservoirs are close to the "edge-of-chaos", and thus good performance can be achieved. And the Lyapunov exponent of the LSM with binary reservoir synapses is even closer to zero, indicating that its dynamics is closer to the transition boundary. In other words, by reducing the resolution of synaptic weights, it may be possible for us to move the dynamics of the reservoir from the chaotic region towards the ordered region, and hence having the system operating at the transition boundary.

Separation and Generalization. Randomly generated input streams and application specific speech signals are applied to the reservoir separately and we measure the ranks r_S and r_G respectively of the

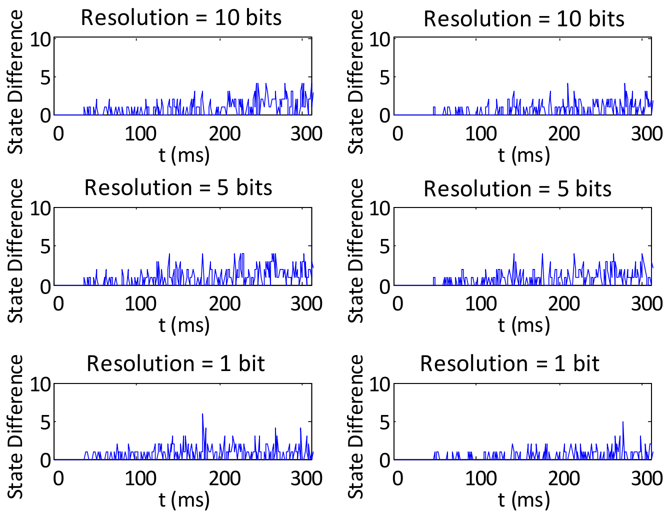


Fig. 15. State separation under different resolutions for synaptic weights in the reservoir. Similar to Fig. 11, the temporal evolution of the Hamming distance between the two resulting states is shown. For the sub-figures on the left and right sides, the input u differs from input v due to only one missing spike at time 24 ms and 42 ms, respectively.

Table 8
Lyapunov exponents of LSMs with various resolutions of the fixed reservoir synapses. The Lyapunov exponents λ_1 and λ_2 are measured under two different input pairs for which the input difference is introduced by a missing spike at 24 ms and 42 ms, respectively. The reported recognition rates of the LSMs under the three resolution settings are 98.16%, 98.40% and 98.72%, respectively.

LSM	Resolution	λ_1	λ_2
1	10 bits	0.36	0.30
2	5 bits	0.38	0.36
3	1 bits	0.10	0.18

Table 9
Estimated separation and generalization capabilities of the LSM as a function of fixed synaptic resolutions. The reported recognition rates of the LSMs under the three resolution settings are 98.16%, 98.40% and 98.72%, respectively.

LSM	Resolution	r_S	r_G	Δ_{SG}
1	10 bits	135	101	34
2	5 bits	135	102	33
3	1 bits	134	98	36

matrix M as mentioned in Section 3.3. As seen in Table 9, the rank difference Δ_{SG} almost remains the same when lowering the resolution of the fixed synaptic weights, which suggests that low resolution does not have a significant impact upon the computational capability. And therefore, good recognition performance can be achieved under low resolution of reservoir's synaptic weights.

In conclusion, low resolution of synaptic weights may be adequate to attain good computational capability of the reservoir, and thus guarantees good separation and generalization. In terms of the readout layer, however, high resolution synapses are required. It is worth mentioning that by applying a coarser synaptic resolution, we can altogether lower the resolution of membrane voltage for the reservoir and readout neurons. Finally in terms of neurons in both the reservoir and readout, 6-bit resolution for membrane voltage and 10-bit resolution for calcium concentration are sufficient to guarantee good performance.

4.3. Size of the reservoir

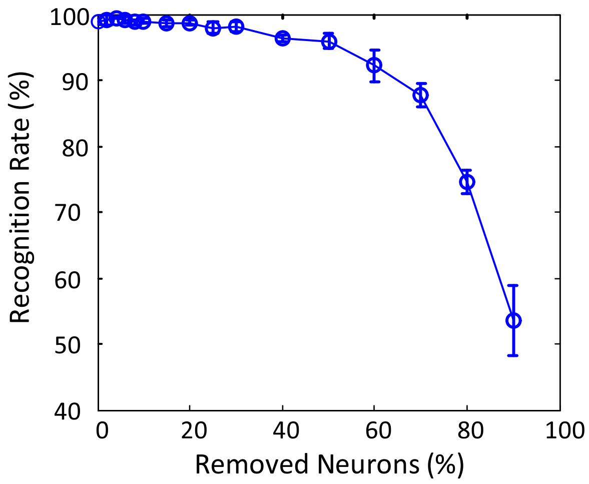


Fig. 16. Performance degrades as the percentage of removed reservoir neurons increases.

Another way to reduce the implementation cost is to cut down the size of the reservoir. To examine this reduction's impact on performance, we randomly remove a certain percentage of neurons from the reservoir. The original reservoir contains 135 neurons. The percentage of the removed neurons is varied from 1–90% and the resulting recognition rates are plotted in Fig. 16. Here we use binary synapses for the reservoir, 8-bit synaptic resolution for the readout, 6-bit membrane voltage resolution for both reservoir and readout neurons, and 10-bit calcium concentration resolution for the readout. In the simulation, each time an LSM is trained and tested with randomly chosen neurons being removed from the reservoir. The simulation results suggest that it is possible to get reasonably high performance without 30% of neurons but the performance begins to degrade noticeably if more neurons get removed. In order to shed light in a theoretical perspective, we again extract the three measures of computational power for these LSMs with the same experimental setup.

Fading Memory. 77 random spike trains are injected into the reservoir for measuring fading memory. In terms of the reservoir size, although fading memory is slightly weakened when squeezing the reservoir, removing too many neurons yields a performance penalty since fading memory will quickly die out or even vanish (shown in Fig. 17), making the reservoir incapable of generating rich dynamics. In this case, good performance can still be obtained given that 40 neurons (30% of the original size) get removed from the reservoir because

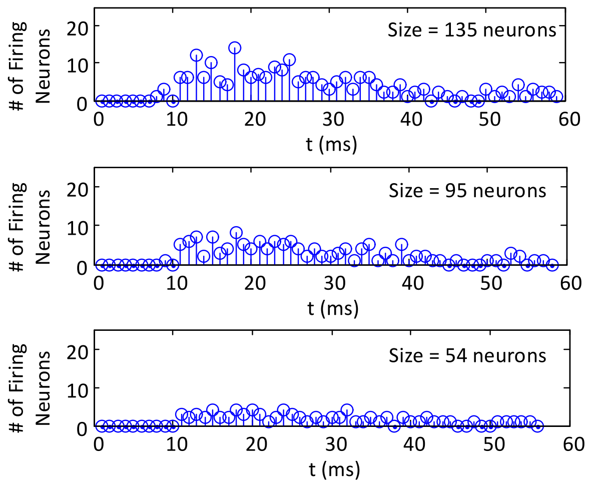


Fig. 17. Firing activity in the reservoir when its size gets reduced. Compared to the results shown in Figs. 10 and 14, the fading memory stays nearly the same after removing 30% neurons from the reservoir. Both the magnitude and duration of the responses are largely reduced if more neurons are removed.

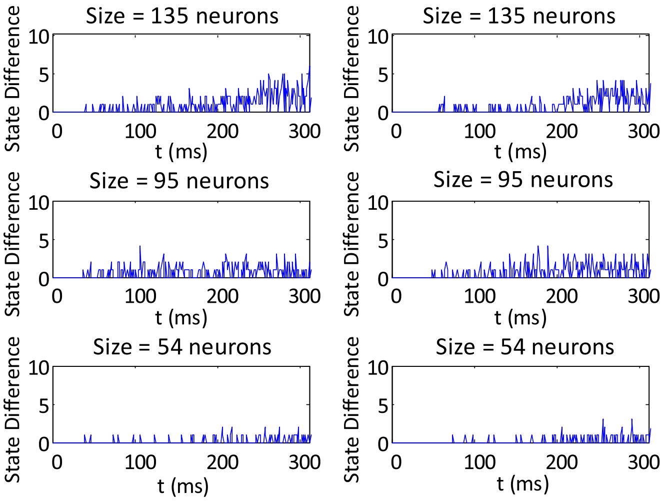


Fig. 18. State separation for the reservoir with different sizes. Similar to Fig. 11, the temporal evolution of the Hamming distance between the resulting states is shown. For the sub-figures on the left and right sides, the input u differs from input v due to only one missing spike at time 24 ms and 42 ms, respectively.

fading memory is well preserved.

Edge-of-Chaos. Two slightly different input signals are injected into the reservoir and Fig. 18 shows how state separation temporally evolves due to a small input difference for three LSMs with descending reservoir sizes. It is clear that by reducing the size of the reservoir, the state difference gradually decreases; in other words, we might be able to change the dynamics of the reservoir from the chaotic region to the ordered region and phase transition happens around the point where 30% neurons get removed, which is further confirmed by the calculated Lyapunov exponent λ shown in Table 10. The Lyapunov exponent of the second LSM in the Table 10 is reasonably close to zero, indicating that the dynamics at this point lies very close to the edge of chaos, which theoretically explains why decent performance can be obtained with this size. Furthermore, in this particular case, LSMs will not function very well if the dynamics is in the ordered region.

Separation and Generalization. The ranks r_S and r_G of the matrix M (see Section 3.3 for details) are obtained by feeding randomly generated input streams and application specific speech signals into the reservoir, respectively. As seen in Table 11, although the first LSM is shown to possess powerful separation and generalization capabilities and has very good performance, the second LSM, whose Δ_{SG} is not the largest, also achieves good recognition performance. This interesting phenomenon can be understood by noting that multiple dynamic regions can provide the LSM with good performance ([24]). Nonetheless, further reduction beyond this point is reconfirmed to cause performance degradation.

4.4. Summary

Finally, we summarize the relation between a broad range of key network parameters and the recognition rates with different levels of performance sensitivity in Table 12. Clearly, there exists a large design space in which various network design parameters can be explored to trade off between hardware overhead and performance. In particular, our experimental study presented here demonstrates the possibility of reducing the network complexity, hence implementation overhead, without incurring any significant degradation of performance. For instance, with the resolutions and the reservoir size getting reduced to the suggested values in Table 12, the recognition rate can still reach up to 98.16%.

Table 10
Lyapunov exponents of LSMs with various reservoir sizes. The Lyapunov exponents λ_1 and λ_2 are measured under two different input pairs for which the input difference is introduced by a missing spike at 24 ms and 42 ms, respectively. The reported recognition rates of the LSMs under the three reservoir settings are 98.92%, 98.16%, and 92.24%, respectively.

LSM	Reservoir Size	λ_1	λ_2
1	135 neurons	0.10	0.18
2	95 neurons	−0.05	0.00
3	54 neurons	−0.40	−0.32

Table 11
Estimated separation and generalization capabilities of the LSM as a function of the reservoir size. The reported recognition rates of the LSMs under the three reservoir sizes are 98.92%, 98.16%, and 92.24%, respectively.

LSM	Reservoir Size	r_S	r_G	Δ_{SG}
1	135 neurons	135	98	37
2	95 neurons	95	69	26
3	54 neurons	54	36	18

The adopted theoretical measures of computational power are normally consistent with the real-world performance. It is worth mentioning that the correlation between the theoretical measures and performance might not be straightforward sometimes, because it is found that the LSM can have numerous dynamical regimes with rich computational power for real-world applications.

5. Robustness of LSMs with respect to catastrophic failures and random errors

The resilience of digital VLSI circuit has been one of the greatest design challenges in the past decades due to the scaling of IC manufacturing technologies and aggressive sizing of transistors. Modern integrated circuits (both analog and digital) are susceptible to a very wide range of failure mechanisms. Therefore, it is worthwhile to examine the robustness of a given LSM when implementing it using highly-scaled modern VLSI technologies for which process variations (e.g. variations in transistors and interconnect parameters) and manufacturing defects (e.g. stuck-at-0 and stuck-at-1 faults) may introduce unavoidable parameter fluctuations, and cause various levels of performance variation or even permanent failures ([25,36]). In addition, modern VLSI chips are prone to errors in operation, which may result from environmental effects (e.g. temperature variation and random power supply noises) and soft errors (e.g. single-event upsets due to cosmic rays and crosstalk noise), potentially causing transient errors and rendering a VLSI-based LSM to fail in numerous ways ([5,20]).

Note that the above failure mechanisms may render rather different types of failure behavior. Catastrophic manufacturing defects may cause permanent failures of certain circuit blocks. Statistical manufacturing process variations may lead to increased circuit delay, hence producing timing errors under certain inputs. While many different inputs are applied to a given logic circuit block, the input dependency of timing errors may be viewed as adding random errors to the output of the circuit block. Environmental effects, in particular, power supply noise can lead to timing failures and hence errors in digital circuits. Since power supply noise has a significant random component, the resulting errors may be modeled as random both temporally and in terms of occurrence probability. Soft errors may lead to erroneous computations of certain output bits and also have a strong random component. These failures are highly process and design dependent and an accurate failure analysis is feasible only for a given the design of the targeted integrated circuit and the adopted manufacturing process. Therefore, given the scope of this work, we only model these failure mechanisms with certain abstraction and assess the general robustness of the proposed LSM rather its particular hardware implementations.

Table 12
Key network parameters and corresponding performance.

Design Specifications	Type	Suggested setting	Range	Best Performance	Worst Performance	Performance Sensitivity
Calcium Level	Readout	10 bits	14 – 8 bits	99.16%	93.76%	Low/ Medium
Membrane Voltage	Reservoir	6 bits	16 – 4 bits	98.52%	97.2%	Low
	Readout	6 bits	16 – 4 bits	98.68%	98.12%	Low
Synaptic Weight	Reservoir	1 bit	10 – 1 bits	98.72%	98.08%	Low
	Readout	8 bits	10 – 4 bits	98.36%	89.4%	High
Size of Reservoir	N.A.	95 neurons	135 – 54 neurons	98.92%	92.24%	Low/ Medium

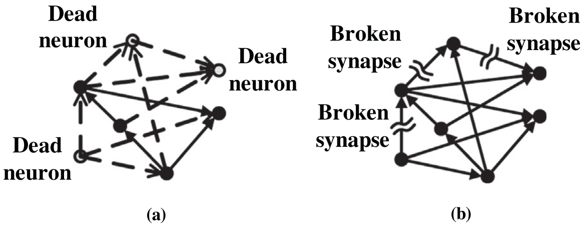


Fig. 19. Modeling broken synapses and dead neurons in the LSM. (a): dead neurons; (b): broken synapses.

As mentioned above, the failures cited above have two main effects in a digital circuit: 1) catastrophic failures that cause certain circuit blocks to fail completely, and 2) random errors in terms of both occurrence and magnitude. First of all, we model the first effect as broken synapses and dead neurons. In the simulation, we assume catastrophic failures may result in permanent malfunction of certain synapses or neurons, that is, some of them fail to respond to the coming spikes and become completely nonfunctional (Fig. 19). This modeling approach effectively removes such broken synapses or dead neurons from the network. To provide an insight about how LSMs perform when subjected to random errors, we consider to introduce a random error probability for key arithmetic blocks (i.e. adders, shifters and comparators) in the network. Furthermore, once an error occurs, a normal distribution is used to model the amount of error produced relative to the correct value.

Similar to the experimental settings in Section 4, in this experimental study, five LSMs are generated with random reservoir connections. For each LSM, we perform the 5-fold cross validation mentioned in Section 2.5 to attain the recognition rate at each targeted failure/error level. The five obtained recognition rates are averaged as the reported rate and the corresponding standard deviation (shown as the error bar with its length being 2 × *SD*) is plotted.

5.1. *Catastrophic failures*

To acquire a quantitative understanding of robustness, we model the effect of catastrophic failures as causing the critical blocks (i.e. synapses and neurons) of the LSM to be non-functional. In the simulation, this effect is equivalent to removing a certain amount of synapses or neurons from the network.

5.1.1. *Broken synapses*

After a certain number of broken synapses being deleted from the LSMs, we measure the recognition rates at different levels of severity of catastrophic failures and plot the results in Fig. 20. Note that even at a fairly large failure level (such as 20%), the LSM can still achieve pretty good performance (around 97%) under the cases where fixed or plastic synapses are broken, respectively. It implies that LSMs are robust to potential "broken synapses" caused by process variations and manufacturing defects. Furthermore, the classification performance is more sensitive to failures of plastic synapses than those of fixed synapses. This can be understood again by noting that the former play a key role in classification conducted by the linear readout neurons. Therefore, plastic synapses shall be implemented with more robust circuit-level techniques.

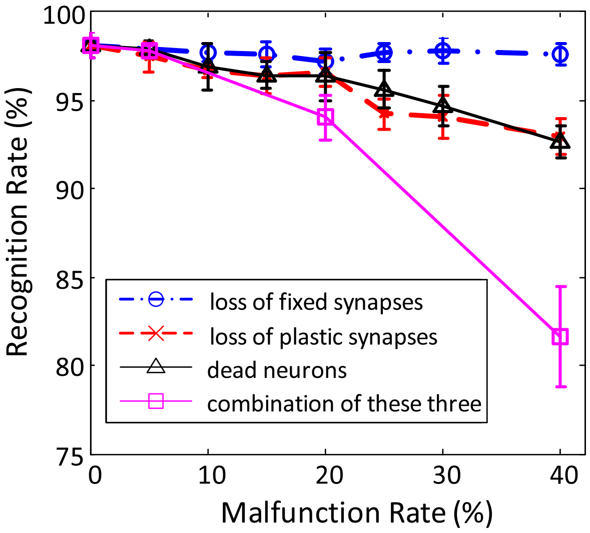


Fig. 20. Performance degradation as a function of malfunction rates. The malfunction rates are the percentages of broken synapses or dead neurons.

5.1.2. *Dead neurons*

We show the recognition rates of the LSM with different percentages of dead reservoir neurons in Fig. 20. As can be observed, the recognition performance of the LSM is more sensitive to dead reservoir neurons than broken reservoir synapses. A possible explanation for this discrepancy is that reservoir neurons are fundamental processing/computing elements in the network. Broken reservoir synapses may not necessarily knock out any neurons from the reservoir while the existence of dead neurons certainly does. Hence, necessary steps of preventing a large number of neurons to fail are important for ensuring good performance. However, thanks to intrinsic resilience and redundancy presented in the LSM, the recognition rate is still about 96% even with 20% of neurons stopping functioning.

5.1.3. *Combination of broken synapses and dead neurons*

Furthermore, we look at the compound effect of having simultaneous broken synapses and dead neurons in Fig. 20. With 5% combined malfunction rate (5% of fixed and plastic synapses are broken and 5% of reservoir neurons stop functioning), only about 1% performance drop is observed. With 20% combined malfunction rate, the performance can still reach up to 94%.

In conclusion, it is clear that the catastrophic failures do have some impacts on performance. Although the examined failure rates are fairly large, the learning performance does not degrade a lot, which reveals the good robustness of this neuromorphic system.

5.2. *Random errors*

To perform this robustness study of the LSMs under random errors, we perturb outputs of crucial arithmetic blocks (i.e. adders, shifters and

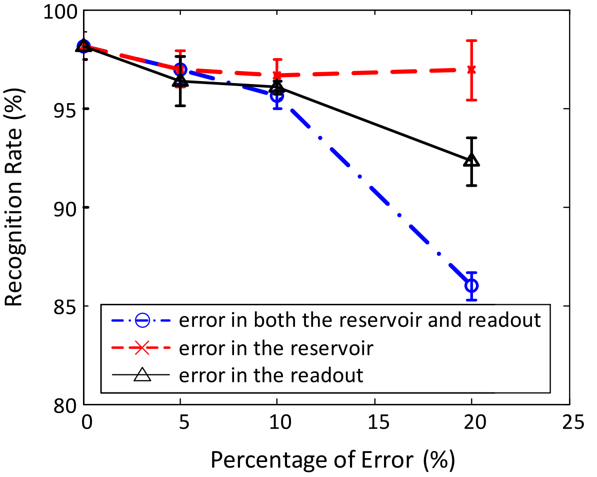


Fig. 21. Performance degradation as a function of the amount of error in adders.

comparators) with errors. We assume that the error probability for each adder and shifter is 0.1 for simplicity. Once an error happens, a normal distribution is used to model the amount of error introduced relative to the correct value for adders and shifters. For each comparator, we consider the probability of generating erroneous output because a comparator only outputs “0” or “1”.

5.2.1. Error in adding operations

To consider a somewhat stressed error scenario, we assume all adders used to implement the neurons and synapses in the LSM are subjected to an error probability P_{err} of 10%. When an error occurs, a normal distribution is used to model the magnitude of error as described before. As the amount of injected error increases, we observe a significant performance degradation in Fig. 21. For example, the LSM largely fails when the amount of error is larger than 20%.

We further study how the network performance may depend on adding errors occurring in parts of the LSM. As shown in Fig. 21, when the reservoir is subjected to the error, the performance only degrades by no more than 2% within the considered range. Hence, the recognition performance appears to be insensitive to errors in the reservoir. In comparison, the performance is much more sensitive to errors in the readout as performance drops down to 86% in the worse case. This phenomenon may be understood by noting that additions are needed for both simulating the neurodynamics of the readout neurons and implementing the learning rule. The former ultimately determines the firing activity of a reservoir neuron which is a basis for interpreting recognition decision; while the latter is responsible for tuning the plastic synaptic weights according to the firing activities in the reservoir, playing a critical role in adjusting the corresponding hyperplanes implemented by the linear readout neurons. However, given that the LSM still performs well under a large amount of error (e.g. 10%), a fairly noisy environment indeed, we do observe good robustness of the network even with respect to errors in the readout.

5.2.2. Error in shifting operations

In this work, shifters are used to simplify multiplication and division operations ([49]). Similar to modeling output errors in adders, we introduce random errors to the outputs of the shifters in the LSM with the error probability of 0.1 and the error amount modeled by a normal distribution. The resulting performance as a function of the amount of shifting error is shown in Fig. 22. As the shifters suffer from more error, the performance degrades very gracefully. Even 20% of error, the performance degradation is less than 2%. Thus we can conclude that LSMs are insensitive to error from shifters. This argument can be further supported by Fig. 22 where we separately consider errors in the reservoir and readout. No significant performance penalty can be seen. It is evident that the network performance is less sensitive to errors in shifters compared with adders. This interesting phenomenon may be understood by noting that multiplication or division is only used when calculating dynamics of neurons and synapses while addition is not only responsible for the computation of dynamics, but also the update of plastic synaptic weights.

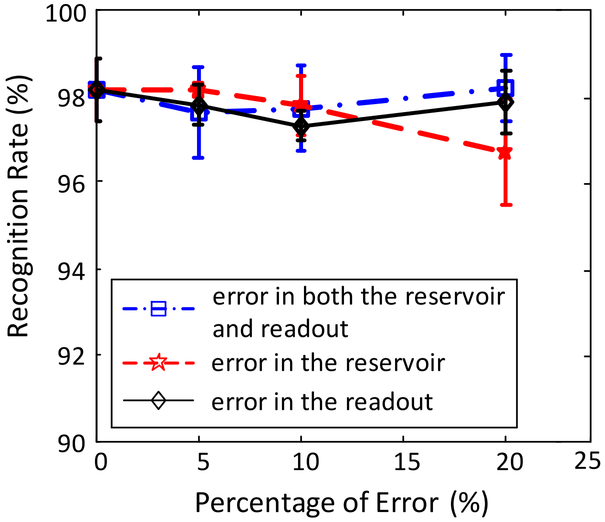


Fig. 22. Performance degrades gracefully with an ascending amount of shifting error.

5.2.3. Error in comparison operations

The way to model random errors in comparators is slightly different from the previous cases. Since comparators only output “0” or “1”, we consider the probability of generating the erroneous outputs. As shown in Fig. 23, the performances at three error probability levels (5%, 10% and 20%) are measured. Since the worst performance penalty is nearly 90%, we conclude that error in comparators does affect the performance. By further perturbing comparators in the reservoir and the readout (Fig. 23), it appears that the performance is sensitive to errors in both the reservoir and the readout. The reason might be that comparison is involved in determining the firing activities of neurons and the adaptation of plastic synaptic weights. Any error in comparison might give rise to unpredictable behaviors of neurons and synapses. For example, an error-prone comparator can render a neuron fire a spike though its membrane voltage is less than the threshold. Similarly, an error-prone comparator in a plastic synapse might mistakenly initiate weight adaption with the learning rule being violated. Therefore, the LSM can be very sensitive to errors in comparison.

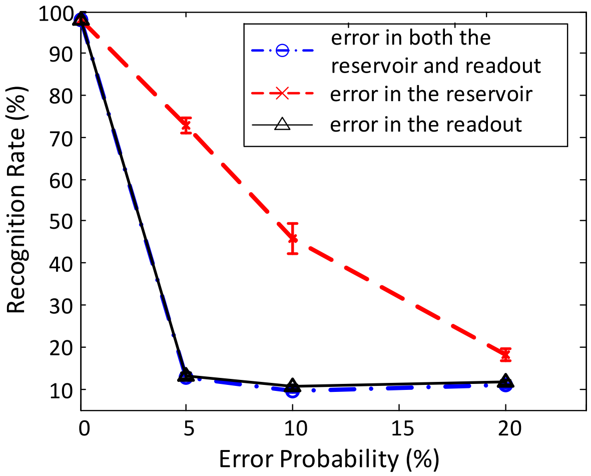


Fig. 23. Performance drops dramatically with an increasing error probability. The impact of comparing error from different parts of the LSM is shown.

Table 13
Typical types of failure and error with their impacts on performance.

Failure/ Error	Type	Range	Worst case performance degradation	Performance sensitivity
Catastrophic failures in reservoir	Dead neurons	0% – 40%	5.48%	Low/Medium
	Broken synapses	0% – 40%	0.96%	Low
Catastrophic failures in readout	Broken synapses	0% – 40%	5.2%	Low/Medium
Error in reservoir	Error in adders	0% – 20%	1.52%	Low
	Error in shifters	0% – 20%	1.44%	Low
	Error in comparators	0% – 20%	79.94%	High
Error in readout	Error in adders	0% – 20%	5.88%	Low/Medium
	Error in shifters	0% – 20%	0.84%	Low
	Error in comparators	0% – 20%	87.68%	High

5.3. Summary

Table 13 summarizes two types of catastrophic failure and three types of random error with their impacts on performance. In general, the reservoir is much more insensitive to failure and error compared to the readout, therefore the readout layer should be the main target of fault and noise tolerance for hardware implementation. Furthermore, error effects associated with comparators appear to have a high impact on performance. Hence, the comparators may be designed with a high-level of robustness. Besides, the recognition performance is sensitive to addition error that happens in the readout, and thus the adders in the readout should also be designed with a high-level of robustness. The other arithmetic operations have been shown to be less critical for the overall performance and hence a less robust implementation may be explored to gain benefits in area and energy consumption. Generally speaking, the studied LSMs appear to be robust to various types of catastrophic failure and random error. This is very appealing and can be leveraged for efficient hardware implementation while maintaining a good level of robustness.

6. Performance study of LSM on other subsets of TI46

To provide a more complete understanding on the trade-offs between design cost and performance, we introduce two additional benchmarks, which are the second and third benchmark described in Section 2.4, and three levels of implementation complexity with design parameters shown in Table 14. The speech signals of the second and third benchmark are preprocessed by the Lyon passive ear model ([28]) then encoded into 83 spike trains by the BSA algorithm ([44,46]), and fed into a group of randomly selected neurons in the reservoir. And other experimental settings are the same as described in Section 4. We test the performance of the LSM on these two additional subsets of the TI46 speech corpus, and the recognition rates of the LSMs on the three different benchmarks adopted in this paper are depicted in Fig. 24. In Table 14, the complexity level 3 is the original setting in [49] and the complexity level 2 is the suggested setting of this paper (see Table 12). It is clear from Fig. 24 that the recognition rate degrades pretty gracefully as the design complexity decreases for all three benchmarks. This suggests that within a reasonably wide range, performance and design overhead can be rather nicely traded off with each other, providing a rather good flexibility in achieving the overall design objectives.

Table 14
Design parameters of different levels of design complexity.

Design Specifications	Type	Complexity Level 3	Complexity Level 2	Complexity Level 1
Calcium Level	Readout	14 bits	10 bits	10 bits
Membrane	Reservoir	16 bits	6 bits	6 bits
Voltage	Readout	16 bits	6 bits	6 bits
Synaptic Weight	Reservoir	10 bits	1 bit	1 bit
	Readout	10 bits	8 bits	8 bits
Size of Reservoir	N.A.	135 neurons	95 neurons	54 neurons

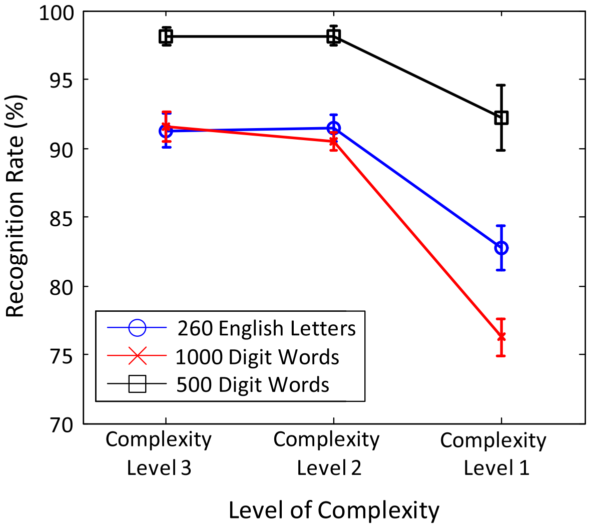


Fig. 24. Classification performance of the LSM on the three adopted benchmarks decreases as a function of design complexity. A reasonably good performance can be attained with reduced complexity.

7. Discussion of other reservoir computing (RC) methods

We expect that the general characteristics (i.e., strong computation capability, redundancy and robustness) of LSMs we found in this paper may exist for a broad set of reservoir computing methods like Echo State Networks (ESNs) ([19]). [45] reported that good performance could be achieved over a broad range of network parameters for different RC methods by using the reservoirs with different neural models. It is found by [38] that a minimum complexity ESN could be constructed to achieve good memory capability by reducing redundancy of the reservoir. Redundancy in the synapses between a reservoir and a readout of an ESN can also be reduced to facilitate the generalization ability as reported in [9]. Noise resilience of the RC methods is observed in [48] where additional noise is injected into reservoir neurons to avoid over-fitting.

8. Conclusion

This paper presents a comprehensive performance and robustness study of bio-inspired digital liquid state machines for speech recognition. By examining a broad range of key network design parameters and using real-world meaningful benchmarks, we shed light on the relationship between design parameters and performance. We show that good performance can be maintained while reducing the resolutions and reservoir size, both of which have immediate impacts on hardware implementation overhead. To gain deep insights into the computational capability of LSMs, we adopt three theoretical measures to illustrate the relation between performance and the design parameters and the results generally agree with the simulated performance. By applying the theoretical measures, we also notice that multiple regimes can provide the LSM with sufficiently rich dynamics for separation of different input samples. To provide practical suggestions for future hardware implementation, we study the impacts of failure and error mechanisms introduced by process variations and environmental effects upon the recognition performance, showing that LSMs are fairly robust.

We have several main findings. First, in general, the implementation of the reservoir does not appear to be critical and require a high level of precision and robustness as long as it is capable of creating rich dynamics. One exception is the implementation of comparators which directly impact the firing activities of the reservoir. On the contrary, robust and accurate arithmetic units (comparators and adders especially) are desirable for the readout because they are critical parts of the LSM. These insights are particularly useful in practice as they offer insightful guidance for circuit implementation such that a good level of performance and robustness may be maintained while avoiding unnecessary overdesign.

References

[1] A. Afifi, A. Ayatollahi, F. Raissi, Implementation of biologically plausible spiking neural network models on the memristor crossbar-based cmos/nano circuits, in: Proceedings of the IEEE Circuit Theory and Design ECCTD European Conference, 2009, pp. 563–566.

[2] N. Bertschinger, T. Natschläger, Real-time computation at the edge of chaos in recurrent neural networks, Neural Comput. 16 (2004) 1413–1436.

[3] J.M. Brader, W. Senn, S. Fusi, Learning real-world stimuli in a neural network with spike-driven synaptic dynamics, Neural Comput. 19 (2007) 2881–2912.

[4] H. Burgsteiner, M. Kröll, A. Leopold, G. Steinbauer, Movement prediction from real-world images using a liquid state machine, Appl. Intell. 26 (2007) 99–109.

[5] H. Chang, S.S. Sapatnekar, Statistical timing analysis under spatial correlations, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 24 (2005) 1467–1482.

[6] T.M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, IEEE Trans. Electron. Comput. (1965) 326–334.

[7] M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, D. Van Compernolle, Template-based continuous speech recognition, IEEE Trans. Audio Speech Lang. Process. 15 (2007) 1377–1390.

[8] G.R. Doddington, T.B. Schalk, Computers: speech recognition: turning theory to practice: new ics have brought the requisite computer power to speech technology; an evaluation of equipment shows where it stands today, Spectr. IEEE 18 (1981) 26–32.

[9] X. Dutoit, B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, M. Nuttin, Pruning and regularization in reservoir computing, Neurocomputing 72 (2009) 1534–1546.

[10] A.A. Faisal, L.P. Selen, D.M. Wolpert, Noise in the nervous system, Nat. Rev. Neurosci. 9 (2008) 292–303.

[11] R.V. Florian, The chronotron a neuron that learns to fire temporally precise spike patterns, PloS One 7 (2012) e40233.

[12] A. Ghani, T.M. McGinnity, L.P. Maguire, J. Harkin, Neuro-inspired speech recognition with recurrent spiking neurons, in: Proceedings of the Artificial Neural Networks-ICANN, Springer, 2008, pp. 513–522.

[13] S. Ghosh-Dastidar, H. Adeli, A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection, Neural Netw. 22 (2009) 1419–1431.

[14] B. Glackin, T.M. McGinnity, L.P. Maguire, Q. Wu, A. Belatreche, A novel approach for the implementation of large scale spiking neural networks on fpga hardware, in: Proceedings of the Computational Intelligence and Bioinspired Systems, . Springer, 2005, pp. 552–563.

[15] A. Graves, D. Eck, N. Beringer, J. Schmidhuber, Biologically plausible speech recognition with lstm neural nets, in: Proceedings of the Biologically Inspired Approaches to Advanced Information Technology,. Springer, 2004, pp. 127–136.

[16] R. Gütig, H. Sompolinsky, The tempotron a neuron that learns spike timing-based decisions, Nat. Neurosci. 9 (2006) 420–428.

[17] H. Hazan, L.M. Manevitz, Topological constraints and robustness in liquid state machines, Expert Syst. Appl. 39 (2012) 1597–1606.

[18] D.O. Hebb, The Organization of Behavior: A Neuropsychological Theory, Wiley, New York, NY, 1949.

[19] H. Jaeger, The echo state approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, 148, 2001, 34.

[20] T. Karnik, P. Hazucha, Characterization of soft errors caused by single event upsets in cmos processes, IEEE Trans. Dependable Secur. Comput. 1 (2004) 128–143.

[21] S.A. Kauffman, The Origins of Order: Self-Organization and Selection in Evolution, Oxford University Press, Oxford, UK, 1993.

[22] Y. Kim, Y. Zhang, P. Li, A digital neuromorphic VLSI architecture with memristor crossbar synaptic array for machine learning, in: Proceedings of the IEEE SOC Conference (SOCC), International, 2012, pp. 328–333.

[23] C.G. Langton, Computation at the edge of chaos phase transitions and emergent computation, Physica D: Nonlinear Phenom. 42 (1990) 12–37.

[24] R. Legenstein, W. Maass, Edge of chaos and prediction of computational performance for neural circuit models, Neural Netw. 20 (2007) 323–334.

[25] P. Li, F. Liu, X. Li, L.T. Pileggi, S.R. Nassif, Modeling interconnect variability using efficient parametric model order reduction, in: Proceedings of the conference on Design, Automation and Test in Europe-Volume 2, 2005, pp. 958–963.

[26] M. Lukoševičius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Comput. Sci. Rev. 3 (2009) 127–149.

[27] B. Luque, R.V. Solé, Lyapunov exponents in random boolean networks, Physica A: Stat. Mech. its Appl. 284 (2000) 33–45.

[28] R.F. Lyon, A computational model of filtering, detection, and compression in the cochlea, In Acoustics, Speech, and Signal Processing, in: Proceedings of the IEEE International Conference on ICASSP'82, vol. 7, 1982, pp. 1282–1285.

[29] W. Maass, Networks of spiking neurons the third generation of neural network models, Neural Netw. 10 (1997) 1659–1671.

[30] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states a new framework for neural computation based on perturbations, Neural Comput. 14 (2002) 2531–2560.

[31] W. Maass, T. Natschlager, H. Markram, A model for real-time computation in generic neural microcircuits, Adv. Neural Inf. Process. Syst. (2003) 229–236.

[32] W. Maass, T. Natschläger, H. Markram, Fading memory and kernel properties of generic cortical microcircuit models, J. Physiol.-Paris 98 (2004) 315–330.

[33] R.J. Mammone, X. Zhang, R.P. Ramachandran, Robust speaker recognition a feature-based approach, Signal Process. Mag., IEEE 13 (1996) 58.

[34] A. Mohemmed, S. Schliebs, S. Matsuda, N. Kasabov, Span: Spike Pattern Association Neuron for Learning Spatio-temporal Spike Patterns, International Journal of Neural Systems, vol. 22, 2012.

[35] D. Norton, D. Ventura, Improving liquid state machines through iterative refinement of the reservoir, Neurocomputing 73 (2010) 2893–2904.

[36] M. Orshansky, S. Nassif, D. Boning, Design for Manufacturability and Statistical Design: A Constructive Approach, Springer Science & Business Media, New York, NY, 2007.

[37] F. Ponulak, A. Kasinski, Supervised learning in spiking neural networks with resume sequence learning, classification, and spike shifting, Neural Comput. 22 (2010) 467–510.

[38] A. Rodan, P. Tiňo, Minimum complexity echo state network, IEEE Trans. Neural Netw. 22 (2011) 131–144.

[39] S. Roy, A. Banerjee, A. Basu, Liquid state machine with dendritically enhanced readout for low-power, neuromorphic VLSI implementations, IEEE Trans. Biomed. Circuits Syst. 8 (2014) 681–695.

[40] M. Rubinov, O. Sporns, J.-P. Thivierge, M. Breakspear, Neurobiologically realistic determinants of self-organized criticality in networks of spiking neurons, PLoS Comput. Biol. 7 (2011) e1002038.

[41] J. Schemmel, A. Grubl, K. Meier, E. Mueller, Implementing synaptic plasticity in a VLSI spiking neural network model, in: Proceedings of the IJCNN'06 International Joint Conference Neural Networks, 2006. on, 2006, pp. 1–6.

[42] B. Schrauwen, L. Büsing, R. Legenstein, On computational power and the order-chaos phase transition in reservoir computing, in: Proceedings of the 22nd Annual conference on Neural Information Processing Systems (NIPS 2008) NIPS Foundation vol. 21, 2009, pp. 1425–1432.

[43] B. Schrauwen, M. DHaene, D. Verstraeten, J.V. Campenhout, Compact hardware liquid state machines on fpga for real-time speech recognition, Neural Netw. 21 (2008) 511–523.

[44] B. Schrauwen, J. Van Campenhout, Bsa, a fast and accurate spike train encoding scheme, in: Proceedings of the International Joint Conference on Neural Networks IEEE Piscataway, NJ, vol. 4, 2003, pp. 2825–2830.

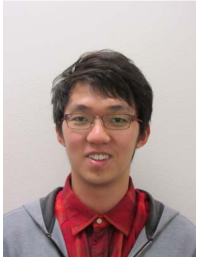
[45] D. Verstraeten, B. Schrauwen, M. dHaene, D. Stroobandt, An experimental unification of reservoir computing methods, Neural Netw. 20 (2007) 391–403.

[46] D. Verstraeten, B. Schrauwen, D. Stroobandt, J. Van Campenhout, Isolated word recognition with the liquid state machine a case study, Inf. Process. Lett. 95 (2005) 521–528.

[47] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, J. Woelfel, Sphinx-4: A Flexible Open Source Framework For Speech Recognition, 2004.

[48] F. Wyffels, B. Schrauwen, D. Stroobandt, Stable output feedback in reservoir computing using ridge regression, Proceedings of the Artificial Neural Networks-ICANN, Springer, 2008, pp. 808–817.

[49] Y. Zhang, P. Li, Y. Jin, Y. Choe, A digital liquid state machine with biologically inspired learning and its application to speech recognition, Neural Netw. Learn. Syst., IEEE Trans. on (2015) 1.



Yingyezhe Jin received the B.S. degree in electronic and information engineering from Zhejiang University, Hangzhou, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Texas A & M University, College Station, TX, USA. His current research interests include simulation and analysis of bio-inspired neural networks and hardware implementation of neural networks.

Peng Li received the B.Eng. degree in information engineering and the M.Eng. degree in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 1994 and 1997, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2003. Since August 2004, he has been on the faculty of the Department of Electrical and Computer Engineering, Texas A & M University, College Station, TX, where he is presently a professor. His research interests are in integrated circuits and systems, electronic design automation, and aspects of computational neuroscience.

His work has been recognized by various awards including the 2013–2014 William and Montine P. Head Fellow from the College of Engineering, Texas A & M University, the IEEE/ACM William J. McCalla ICCAD Best Paper Award in 2012, the 2011–2012 TEES Fellow Award from the College of Engineering, Texas A & M University, three IEEE/ACM Design Automation Conference Best Paper Awards in 2003, 2008, and 2011, the US National Science Foundation CAREER Award in 2008, the ECE Outstanding Professor Award from Texas A & M University in 2008, two MARCO Inventor Recognition Awards in 2006 and 2007, and two SRC Inventor Recognition Awards in 2001 and 2004. Li was an Associate Editor for the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems from 2008 to 2016 and currently is an associate editor for the IEEE Transactions on Circuits and Systems-II: Express Briefs. He is an Fellow of the IEEE.