AP-STDP: A Novel Self-Organizing Mechanism for Efficient Reservoir Computing

Yingyezhe Jin and Peng Li
Department of Electrical and Computer Engineering
Texas A&M University, College Station, Texas, USA
Emails: {jyyz, pli}@tamu.edu

***Abstract*—The Liquid State Machine (LSM) exploits the computation capability of recurrent spiking neural networks by incorporating a randomly generated reservoir, which is often fixed. This standard choice relaxes the challenging need for training the complex recurrent reservoir. The fixed reservoir is used as a generic kernel to map the temporal input signals to the internal network dynamics, and a readout layer is trained to extract the information embedded in the network dynamics to facilitate pattern classification.**

However, the question of how to effectively tune the reservoir for given computational tasks remains to be answered. In this paper, we propose a novel Activity-based Probabilistic Spiking-Timing Dependent Plastic (AP-STDP) mechanism for self-organizing reservoirs. Compared to conventional STDP mechanisms, the proposed rule improves tuning efficiency, prevents the saturation of synaptic memory, and boosts performance. We assess the internal representation ability of the proposed self-organizing mechanism via principal component analysis (PCA) and show that the proposed method is advantageous over other STDP algorithms. Using the spoken English letters adopted from the TI46 speech corpus for performance benchmarking, we demonstrate that AP-STDP consistently outperforms other STDP mechanisms regardless of reservoir size, and is able to boost the performance of the isolated spoken English letter recognition by 2.7% with a small reservoir size.

I. Introduction

Recently, there has been increasing interest in exploring reservoir computing, a biologically plausible computation paradigm, to make use of the computational power of recurrent neural networks without tuning complex recurrent connections [1]. The liquid state machine (LSM) is one specific form of reservoir computing, which has recently emerged in computational neuroscience [2], [3]. Structurally, the LSM (shown in Fig. 1) is typically composed of a fixed "reservoir", a randomly connected recurrent spiking neural network, mimicking generic neural microcircuits in the cerebral cortex, and a group of readout neurons that make the final classification decisions by processing the information coming from the reservoir. The reservoir is used as a generic filter through which the LSM maps the input into a high-dimensional space of network dynamics. Typically, a linear readout output layer can be trained to conduct classification with good accuracy. The LSM is especially competent for processing continuous streams of temporal inputs [4]–[6].

The LSM also offers an interesting paradigm for realizing VLSI-based hardware learning processors. Having a generic common reservoir makes it amenable to realize general-purpose processor architectures on which multiple applications share the same reservoir. The inherent error resilience of the LSM is also appealing for VLSI implementation in highly scaled modern CMOS technologies for which device reliability and process variation are grand challenges. As a result, the hardware implementation of the LSM has emerged [7]–[9].

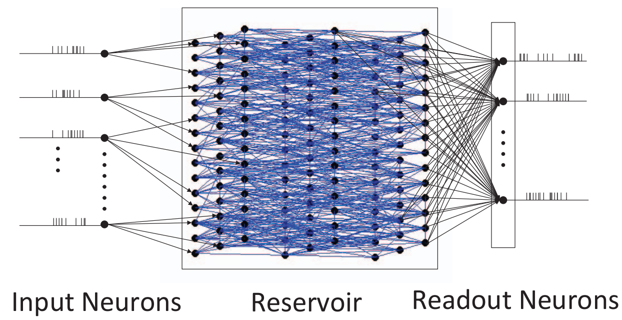

Fig. 1. The LSM consists of the reservoir and a readout layer.

Adapting the recurrent connections in the reservoir is in general very challenging because of complex long-term dependencies in the dynamics [10]. In this regard, the typical LSM model is attractive as it exploits the computational power of the recurrent reservoir without tuning it. Although integrating a generic reservoir presents a simple solution to build general-purpose LSM processors as presented in [9], many studies have argued that randomly generated fixed reservoirs do not act as an effective filter for specific applications [11], [12].

While reservoir tuning has not been well studied in the literature, several attempts have shown that the separation capability of the reservoir and hence learning performance may be boosted by tuning the recurrent connections [12]–[18]. But the question of how to efficiently tune the recurrent reservoir for improving the performance of real-world applications remains unclear. In [12], an iterative refinement approach is proposed to modify the recurrent synaptic connections to boost the separation ability but no performance improvement for real-world tasks is reported. A gene regulatory network (GRN) regulated reservoir is proposed in [18] and shown to have self-organizing behaviors leading to improved performance. However, for both approaches, the global neuronal activities need to be known for altering synaptic weights, which is inefficient and costly to implement.

One can also introduce self-organizing behaviors managed through spike-timing dependent plasticity (STDP), which was both experimentally discovered in biology [19] and theo-

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

retically studied in computational neuroscience [20]. STDP explores the correlation between the firing activities of a pair of presynaptic and postsynaptic neurons and tunes the synaptic weight locally in an unsupervised manner. Therefore, compared to [12], [18], STDP is more suitable for efficient online learning because of its simplicity and locality, as explored in [13]–[17], [21], [22] for tuning reservoirs.

In this paper, we explore efficient reservoir tuning in the context of hardware implementation where synaptic weights are realized digitally with a finite resolution. From a biological point of view, targeting discretized synaptic weights is reasonable because there is evidence that modification of individual biological synaptic strength is done in an all-or-none (i.e. digital) instead of graded (i.e. analog) manner [23], [24]. Furthermore, since we deal with realistic synapses, it is not possible to reduce synaptic modifications to an arbitrarily small value [25].

Unfortunately, standard STDP rules conduct continuous weight updates and may trigger a large number of small-valued weight updates throughout the reservoir, degrading the efficiency of hardware-based realizations on FPGA or in ASIC. Furthermore, there exist important tradeoffs between the range of weight tunability, the resolution of weights, performance and hardware cost. High resolution and wide tunability range can lead to good learning performance at the cost of high hardware overhead. Therefore, STDP rules must be carefully designed for cost-effective realization of the desired self-organizing properties. As such, low-resolution synapses represented by a small number of bits are strongly preferred.

Equally importantly, there is no prior STDP work that addresses the issue of synaptic weight saturation for reservoir tuning. Without a specific stop-learning mechanism in place, continuous on-going weight modifications may quickly saturate a synaptic weight, making it unresponsive to future inputs. This situation of synaptic memory saturation is very likely to happen in hardware because synapses have a limited number of states due to low resolution and narrow tuning range. Therefore, from a memory retention point of view, suitable stop-learning conditions are desired in order to prevent saturation such that synapses can learn from the new experience without being over-interfered by the past experience [25].

To address the above challenges, this work proposes a novel activity-based probabilistic STDP (AP-STDP) rule to tune plastic reservoirs. The proposed rule achieves good learning performance with low synaptic weight resolutions by incorporating a probabilistic weight update process. Furthermore, AP-STDP prevents memory saturation by introducing activity-level based weight tuning with a stop-learning condition. By performing principal component analysis of the reservoir dynamics, we demonstrate that AP-STDP gives rise to more effective internal representations of input samples compared to other simpler STDP rules. We use the spoken English letters from the widely adopted TI46 Speech Corpus [26] as a real-world speech recognition benchmark to test the performance of liquid state machines tuned with AP-STDP. It is demonstrated that the proposed AP-STDP outperforms other conventional STDP rules and can produce a performance that is better than some of the best reported recognition performances obtained using fixed reservoirs [6].

The rest of the paper is organized as follows. Section II provids a brief description of the background and motivations of this work. Section III presents our proposed AP-STDP rule. In Section IV, the PCA analysis of reservoir dynamics is introduced. The network settings and benchmark are described in Section V. The experimental results are reported in Section VI. Finally, Section VII concludes this work.

II. Background

We briefly discuss the standard STDP rules and their limitations pertaining to self-organizing reservoirs.

A. Standard STDP Rules

STDP is a local unsupervised Hebbian learning mechanism realizing synaptic plasticity based on the respective firing timing orders of the presynaptic and postsynaptic neurons. The synapse w_{ij} from neuron j to neuron i is potentiated if a causal order (i.e., pre fires before post) is observed, or depressed if the postsynaptic neuron fires before the presynaptic neuron. The change in weight depends on the temporal difference $\Delta t = t_{post} - t_{pre}$ between the specific pair of pre- and postsynaptic spikes:

$$\begin{aligned} \Delta w^+ &= F_+(w) \cdot e^{-\frac{|\Delta t|}{\tau_+}} \qquad \textit{if } \Delta t > 0 \\ \Delta w^- &= F_-(w) \cdot e^{-\frac{|\Delta t|}{\tau_-}} \qquad \textit{if } \Delta t < 0, \end{aligned} \tag{1}$$

where Δw^+ and Δw^- represent the weight modification induced by long-term potentiation (LTP) and long-term depression (LTD), and $F_\pm(w)$ describes the dependency of the update on the current weight value. If $F_\pm(w) = A_\pm$ is fixed, it is called an additive STDP rule. If $F_\pm$ is proportional to the current weight w, it is called a multiplicative STDP rule. Additive STDP rules are usually applied to the reservoir because they have been shown to generate good self-organizing behaviors [27].

A typical additive STDP curve is plotted in Fig 2(a). And there are generally two pairing rules for the implementation of STDP: all-pairing and nearest-neighbor (shown in Fig. 2(b)). In the first scheme, synaptic updates are triggered by all possible pre-post spike pairs before and at the current time t. In nearest-neighbor pairing, instead, at each firing time, a presynaptic (post-synaptic) spike is only paired with the closed preceding post-synaptic (pre-synaptic) spike. As a common practice, excitatory synapses in the reservoir are usually assumed to be plastic and are tuned with STDP while inhibitory synapses are fixed.

No matter which pairing rule is adopted, realizing (1) can produce weight updates that are arbitrarily small as the temporal difference Δt increases. The fact that a high synaptic resolution is needed to accommodate small updates and the number of such updates can be very large leading to a high level of inefficiency. On the other hand, simply reducing

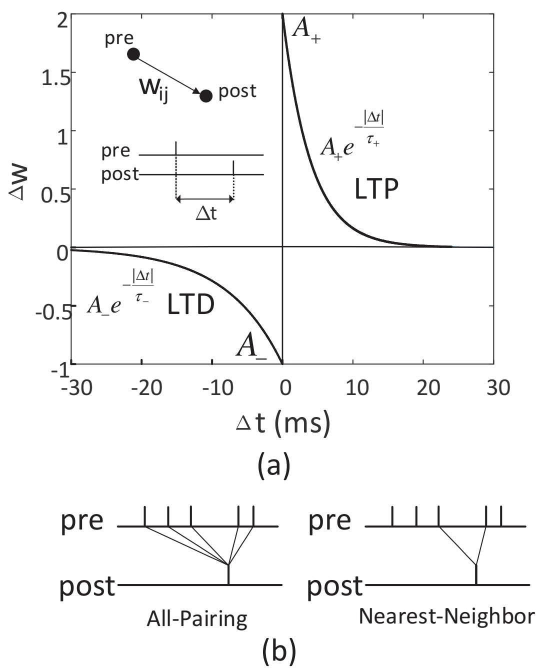

Fig. 2. (a) A typical additive STDP curve. (b) Two pairing schemes.

hardware overhead by adopting a low resolution can lead to learning performance degradations. Another disadvantage of this standard STDP rule is that persistent firing activities can eventually saturate the storage capacity of a given synapse and push the weight to the upper/lower limit, preventing the reservoir from adapting to subsequent input samples and degrading learning performance.

III. The Proposed Activity-based Probabilistic STDP

We first propose a simple probabilistic STDP rule for reservoir computing targeting low synaptic weight resolutions. Using the probabilistic rule as a starting point, we further propose an Activity-based Probabilistic STDP (AP-STDP) rule incorporating a stop-learning condition based upon the post-synaptic neural firing activity level. AD-STDP addresses both the resolution and memory saturation challenges discussed in Section II.

A. The Proposed Probabilistic STDP

Inspired by two stochastic learning rules developed under contexts different from reservoir computing [28], [29], we propose a simple probabilistic STDP rule for reservoir tuning. The proposed probabilistic rule first computes the weight change $\Delta w^+/\Delta w^-$ according to (1). Instead of directly applying the weight change, the rule probabilistically commits a fixed amount of weight update with a probability that is proportional to the magnitude of $\Delta w^+/\Delta w^-$:

$$\begin{aligned} w &\leftarrow w + \Delta W \;\textit{with}\; p \propto |\Delta w^+| \quad \textit{if}\; \Delta t > 0 \\ w &\leftarrow w - \Delta W \;\textit{with}\; p \propto |\Delta w^-| \quad \textit{if}\; \Delta t < 0, \end{aligned} \tag{2}$$

where ΔW is the fixed weight update, representing the resolution of synaptic weights, i.e., a large ΔW leads to a low resolution and a lower hardware overhead. Furthermore, since the total number of weight updates is limited by the probabilities, the update process of this rule is much more efficient than that of the standard STDP. It is worthwhile mentioning that the probabilistic update of synaptic weights can be easily realized in hardware by using efficient random number generator (RNG) primitives.

In terms of performance, it has been argued that continuously fast weight updates of synapses with a limited number of states (e.g. due to a low synaptic resolution) can result in bad memory performance. This manifests itself in such a way that the most recent experiences are represented and learned by the synapses better than the older ones [30], [31]. The proposed probabilistic STDP addresses the above problem by slowing down the learning procedure and helping maximally utilize the synaptic storage capacity [25]. As a result, it more evenly distributes the memory across the network so as to well represent both the new and old experiences and allow for retrieval them at a later time. However, the issue of synaptic memory saturation remains unresolved.

B. The Proposed Activity-based Probabilistic STDP

Although the probabilistic STDP rule helps to slow down the learning for better memory performance, it has no stop-learning condition imposed to tackle the issue of synaptic memory saturation. In the unsupervised learning process of STDP rules, new input samples are fed into the reservoir and synaptic weights are tuned to capture the internal patterns of the inputs. The reservoir will not be able to learn from new stimuli once most of its synapses are over-potentiated or over-depressed (i.e., the synaptic memory is saturated). Incorporating a stop-learning condition provides an effective way to prevent memory saturation [25], which is particularly desirable for synapses with a finite number of states.

Conceptually, we may deactivate LTP when over-potentiation of synapses happens and deactivate LTD when over-depression takes place. This can be done for each synapse by monitoring the firing activity of the postsynaptic neuron. For example, if the postsynaptic neuron is overly active and has fired a lot of spikes, we could “turn off” the LTP of the STDP rule because the afferent synapses of this neuron may have been over-potentiated by this time. Similarly, if the postsynaptic neuron is inactive, we might “switch off” the LTD of the STDP rule to stop/prevent the synapses from being over-depressed. The above mechanism regulates the network dynamics and helps produce more orderly self-organizing behaviors in the reservoir which are crucial to good performance.

While the instantaneous firing frequency acts as a direct measure for the activity level of a neuron, a measure at a longer timescale may correlate better with the average firing level induced by both the new and old inputs. Motivated by [25], we adopt the internal calcium concentration of a biological neuron as an indicator for the firing activity within a specified time window in the proposed AP-STDP rule. The calcium variable $c(t)$ follows the first order dynamics with a large time constant and is a function of the postsynaptic neuron activity:

$$\frac{dc(t)}{dt} = -\frac{c(t)}{\tau_c} + \sum_i \delta(t - t_i), \tag{3}$$

where τ_c is the time constant and the summation is over all postsynaptic spikes arriving at time t_i.

We now discuss the idea of the stop-learning condition based on the calcium concentration. First of all, a threshold of calcium variable c_θ is defined for determining whether the neuron is active or inactive. Suppose a postsynaptic neuron is active and it has fired many spikes. Its corresponding calcium concentration must be very high. Our stop-learning condition basically enables the LTP of the corresponding synapses only when c is within a specific range. Otherwise, it deactivates LTP to avoid over-potentiation. Similarly, if a postsynaptic neuron is inactive and its calcium concentration is too low, implying that afferent synapses are possibly over-depressed, LTD would be disabled. More specifically, the proposed stop-learning condition is combined with the probabilistic STDP rule presented in the last subsection in the following form:

$$\begin{aligned} w \leftarrow w + \Delta W \textit{with } p \propto |\Delta w^+| \quad & \textit{if } \Delta t > 0 \ \&\& \\ & c_\theta < c < c_\theta + \Delta c \\ w \leftarrow w - \Delta W \textit{with } p \propto |\Delta w^-| \quad & \textit{if } \Delta t < 0 \ \&\& \\ & c_\theta > c > c_\theta - \Delta c, \end{aligned} \tag{4}$$

where c_θ is the threshold for determining the postsynaptic neuron firing activity, and Δc is a margin that is introduced to realize the stop-learning condition.

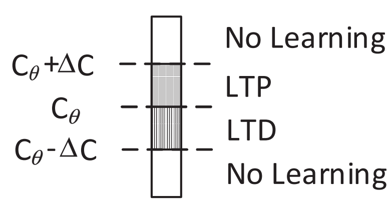

Fig. 3. The AP-STDP rule with the stop-learning condition.

As shown in Fig. 3, we allow a synapse to be potentiated if and only if the calcium concentration c of the postsynaptic neuron is in the range of $[c_\theta, c_\theta + \Delta c]$. Similarly, a synapse is depressed if and only if the calcium concentration falls in the range of $[c_\theta - \Delta c, c_\theta]$. No long-term modification is induced if the calcium level of the postsynaptic neuron is too low or too high. This regulatory mechanism protects the synaptic memory against modifications triggered by the ongoing spontaneous activity, prevents the synapses from over-potientation or over-depression, and allows for full exploration of all input samples in the learning process.

By introducing the stop learning rule based on the calcium level, we correlate the postsynaptic neuron firing activity with the synapse plasticity to avoid saturation. Note that since LTP and LTD are only allowed when the calcium concentration falls within the respective limits, no global positive feedback effect exists. In addition, the calcium concentration follows the first order dynamics of (3), and hence its value can increase or decrease depending on the balance between post-synaptic firing activities and the leakage. As a result, the imposed stop learning rule allows the synaptic weight to change freely without holding it at a specific high or low value.

Since the nearest-neighbor pairing scheme can be done more easily than the all-pairing scheme and the difference in performance between the two schemes is subtle, in this paper, we only focus on nearest-neighbor pairing for our AP-STDP rule to reduce the implementation overhead.

IV. Estimation of Internal Representation Ability of Self-Organizing Reservoirs

To shed light on how the network dynamics may be impacted by the applied STDP mechanisms, hence leading to different learning performances, we perform principal component analysis (PCA) on the reservoir responses induced by various input samples. We define the network state at a given time t as a binary vector $\mathbf{s(t)} \in \{0,1\}^{N^r}$, where N^r is the total number of reservoir neurons, and $s_i(t)$ is 1 if and only if the i^{th} reservoir neuron fires at time t. The network state $\mathbf{s(t)}$ specifies the reservoir response at t. The reservoir response matrix at time t is defined as:

$$\mathbf{R(t)} = \{\mathbf{s^0(t)}, \mathbf{s^1(t)}, \cdots, \mathbf{s^j(t)}, \cdots, \mathbf{s^N(t)}\}, \tag{5}$$

where j is the index of the input samples, and N is the number of the input samples considered. The j-th column vector of $\mathbf{R(t)}$, or $\mathbf{s^j(t)}$, represents the network state at time t given the j^{th} input. The defined reservoir response matrix $\mathbf{R(t)}$ is a snapshot of the complex reservoir dynamics at time t considering all input samples (shown in Fig. 4). By analyzing the response matrix $\mathbf{R(t)}$, we can better understand how well the input samples are represented by the network dynamics.

Applying PCA to the response matrix $R(t)$ allows us to visualize the reservoir responses in the projection space expanded by the first few, say three, principal components (PCs). For the reservoir with weak internal representation capability, the projected responses of different class labels are expected to overlap with each other. In contrast, the formation of tight intra-class clusters with small or no inter-class overlaps is indicative of effective internal representation. We further evaluate the network dynamics by calculating the amount of variance explained by the first several PCs. The greater the variance that is explained by the first several PCs, the more orderly the network dynamics is, suggesting that the internal structures of the input samples can be better captured by the

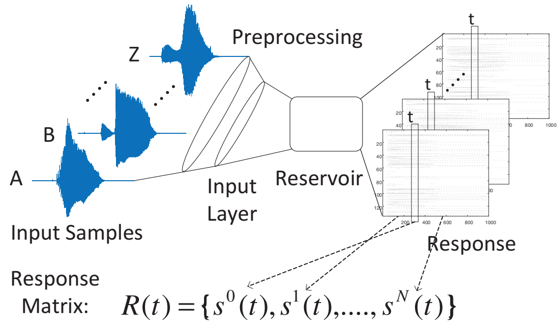

Fig. 4. Extract the response matrix $R(t)$ from the reservoir responses.

network response. The experimental results of PCA on the reservoir responses are reported in Section VI.

V. Experimental Settings and Benchmark

The reservoirs of two liquid state machines are set up using the approach described in [6], giving rise to a recurrent network of 135 and 90 reservoir neurons on a 3D grid, respectively. 80% of the reservoir neurons are excitatory while the rest of them are inhibitory. The connectivity between any two neurons is constructed randomly under a probabilistic distribution function such that the wiring probability of two neurons drops exponentially with the distance between them:

$$P(i,j) = k \cdot e^{-\frac{D(i,j)}{r^2}} \quad (i \neq j), \tag{6}$$

where $D(i,j)$ is the Euclidean distance between neuron i and neuron j, and r and k are two control parameters chosen as suggested in [6]. We adopt the discrete LIF neuronal model and the second-order synaptic model described in [6].

The parameters of all STDP rules described in this work are shown in Table I. The maximum synaptic weight W_{max} is set to 8.0. The initial weights of excitatory synapses are set to ΔW while inhibitory synaptic weights are initialized to $-\Delta W$. We set the bit-width of reservoir synaptic weights to 4 bits. For comparison purposes, we have tuned parameters of the probabilistic STDP rule such that the total number and amount of weight updates are roughly the same for both the AP-STDP and probabilistic STDP rules.

TABLE I
Parameter settings of the standard STDP, probabilistic STDP and AP-STDP Rules.

Parameter	Value
A_+	8.0
A_-	4.0
τ_+	2.0
τ_-	4.0
ΔW	1.0
c_θ	5.0
Δc	3.0
τ_c	64.0

The adopted benchmark is a subset of the TI46 speech corpus [26]. This benchmark contains 10 utterances of each English letter from "A" to "Z", which were recorded from a single speaker. There are 260 samples in this benchmark. The time domain speech signals are preprocessed by Lyon's passive ear model [32], and encoded into 83 spike trains using the BSA algorithm [33]. Each input spike train generated in the preprocessing stage is sent to 32 randomly selected reservoir neurons with a fixed weight randomly chosen to be 2 or -2. The readout layer is fully connected to the reservoir with plastic synapses trained using the bio-inspired supervised learning algorithm proposed in [6].

Before training the readout layer, all speech samples are presented to each plastic reservoir one by one while a STDP rule is applied to tune the reservoir synapses. The process is repeated for a sufficient number of iterations till the reservoir synaptic weights converge. Then, the readout layer is trained with the learning algorithm described by [6]. We adopt a 5-fold cross validation scheme to test the recognition performance for each LSM network setting by randomly dividing all speech samples into 5 groups. The recognition decision is made right after each testing speech sample is presented. At this time, the readout neuron with the highest firing frequency is chosen as the winner whose class label is deemed to be the classification decision.

VI. Experimental Results

Using the experimental setups described in Section V, we compare five reservoir tuning settings, which are abbreviated in Table II, for two reservoir sizes, namely 135 and 90 neurons, respectively.

TABLE II
Reservoirs Tuning Methods.

Abbreviation	STDP Rule and Pairing Scheme Used
Static	Randomly Generated Fixed Reservoir
AAP	Standard Additive STDP w/ All-Pairing
ANN	Standard Additive STDP w/ Nearest-Neighbor
PAAP	Probabilistic STDP w/ All-Pairing
PANN	Probabilistic STDP w/ Nearest-Neighbor
Proposed	Activity-based Probabilistic STDP w/ Nearest-Neighbor

A. Principal Component Analysis for Reservoir Dynamics

We first analyze reservoir dynamics using PCA. For this, we collect the reservoir response matrix $\mathbf{R}(\mathbf{t})$ at a randomly selected time $t_0 = 434\ ms$ and perform PCA to the reservoir responses for both static and plastic reservoirs. We visualize the projected responses in the projection space spanned by the first three PCs in Fig. 5. The static reservoir and the plastic reservoirs tuned using the standard STDP create visible overlaps across different speech classes without forming good intra-class clusters. This again indicates that a randomly generated reservoir may not be effective for a specific application. Fig. 5 also suggests that simply adopting the standard STDP does not produce the desired self-organizing behaviors when the reservoir synapses have a low resolution.

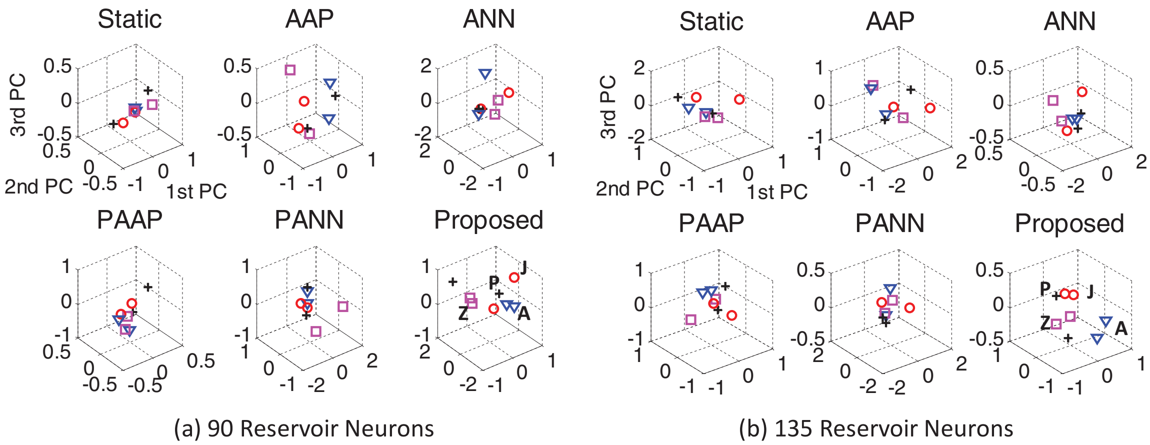

Fig. 5. Visualization of the reservoir responses in the PCs space for different spoken letters. For simplicity, we visualize the responses of input speech samples with four different class labels (letters). The four class labels 'A', 'J', 'P' and 'Z' are marked as '$\bigtriangledown$', circle '$\circ$', cross '+' and square '$\square$', respectively.

Although intra-class clusters are partially formed in some cases with the probabilistic STDP rule, overlaps across different classes still exist. This suggests that the resulting poor input representation power may be attributed to the occurrence of synaptic memory saturation due to the lack of stop-learning rules. This conclusion is supported by the results of the proposed AP-STDP rule, which produces compact intra-class clusters and also well separates different classes.

TABLE III
AMOUNT OF VARIANCE EXPLAINED BY THE FIRST SEVERAL PCS FOR DIFFERENT RESERVOIR TUNING METHODS.

	Principal Components					
	135 Reservoir Neurons			90 Reservoir Neurons		
# of PCs	5	20	65	5	20	50
Static	21.8%	55.4%	94.2%	23.5%	65.3%	96.6%
AAP	20.3%	55.1%	94.6%	23.3%	65.8%	96.6%
ANN	18.7%	52.4%	93.2%	24.1%	67.1%	96.5%
PAAP	17.8%	51.6%	92.8%	23.0%	64.8%	96.2%
PANN	19.6%	54.0%	93.6%	24.3%	68.4%	96.0%
Proposed	**21.5%**	**57.2%**	**94.4%**	**28.4%**	**73.9%**	**97.8%**

Further analysis of the variance explained by the first several PCs offers additional insights on the internal representation effectiveness of different reservoirs as reported in Table III. We use the static reservoir case as a baseline reference and visualize the change in the explained variance due to the adoption of different STDP rules in Fig. 6. As shown in Table III and Fig. 6, the standard and probabilistic STDP rules either make no significant improvements over the static baseline or even underperform it, reaffirming the potential weaknesses of these rules.

In the case of AP-STDP, there is a greater amount of variance explained by the first several PCs compared to the static baseline and other STDP rules. This is consistently the case for the reservoirs with 135 neurons and 90 neurons. As seen in Fig. 6, up to 2% and 6% more variance can be explained with AP-STDP compared to the static reservoir. These results suggest that the network dynamics induced by AP-STDP have an improved internal representational structure.

B. Recognition Performances

TABLE IV
RECOGNITION PERFORMANCES OF THE LSMS WITH DIFFERENT RESERVOIRS.

Reservoir Tuning	**135 Reservoir Neurons**	**90 Reservoir Neurons**
Static	92.3%	89.6%
AAP	92.3%	88.4%
ANN	90.4%	89.6%
PAAP	93.5%	88.1%
PANN	92.3%	89.2%
Proposed	**94.2%**	**92.3%**

We use the adopted benchmark described in Section V to test the LSM recognition rates as reported in Table IV. We also plot the performance boosts over the static baseline achieved by the plastic reservoirs in Fig. 7. To the best knowledge of the authors, the best reported performance on the same benchmark achieved by a static LSM with 135 reservoir neurons is 92.3% [6]. The recognition performance of our static reservoir with 135 neurons achieves the same recognition rate of 92.3%. When the reservoir size is reduced to 90 neurons, the recognition rate of our static reservoir becomes 89.6%.

In comparison with the static baseline, the standard STDP rule degrades the performance as shown in Table IV and Fig. 7. The improvements of the probabilistic STDP rule over the static reservoir are not consistent. It in fact leads to performance degradation in some cases. As shown in Table IV, the performance of AP-STDP is superior than other STDP rules. AP-STDP boosts the performance by 1.9% compared to the best reported performance obtained under a static reservoir for the reservoir size of 135 neurons. AP-STDP produces a

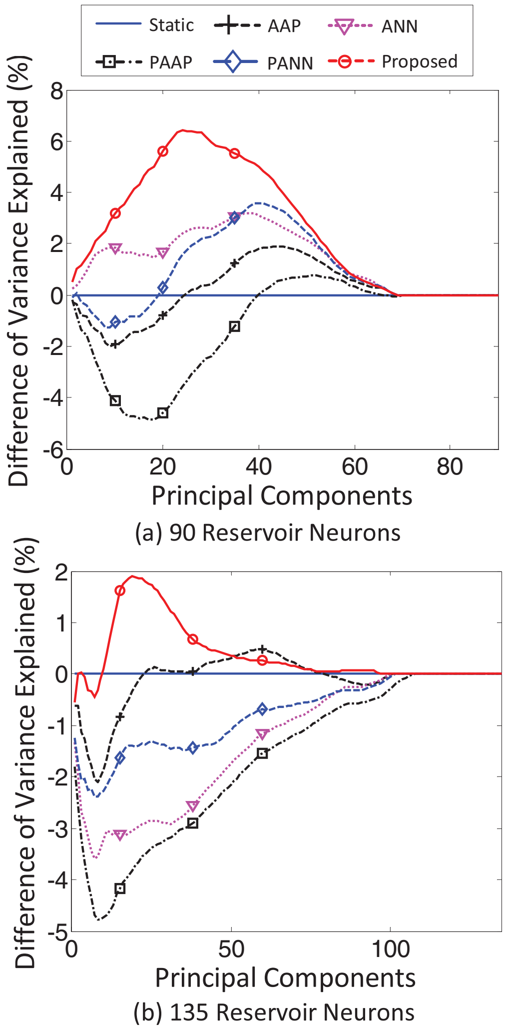

Fig. 6. The differences between the plastic reservoirs and the static reservoir (baseline reference) in terms of the variances explained by the first several PCs.

good recognition rate of 92.3% when the reservoir has only 90 neurons. The performance boost over the static baseline is 2.7% in this case.

VII. Conclusions

In this paper, we have proposed a novel activity-based probabilistic STDP (AP-STDP) rule as a promising self-organizing approach to construct plastic recurrent reservoirs in the context of the liquid state machine. Through a probabilistic update mechanism, AP-STDP achieves good learning performance and facilities efficient reservoir tuning at low synaptic weight resolutions. Furthermore, AP-STDP addresses the issue of synaptic memory saturation by imposing a stop-learning condition based an activity measure. AP-STDP is shown to outperform all other studied STDP rules based on the principle component analysis of the network dynamics and realistic performance benchmarking using speech recognition.

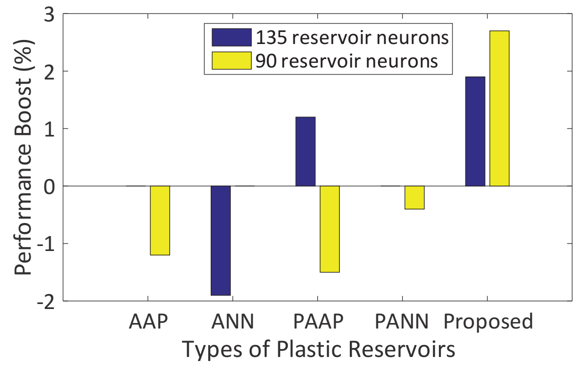

Fig. 7. The performance boosts over the static reservoir achieved by different plastic reservoirs. The proposed AP-STDP significantly boosts the performance for both the 90-neuron and 135-neuron reservoirs. AAP, ANN, and PANN lead to close-to-zero performance boosts in some cases.

References

[1] M. LukošEvičIus and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[2] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[3] W. Maass, T. Natschläger, and H. Markram, "A model for real-time computation in generic neural microcircuits," in *Advances in neural information processing systems*, 2002, pp. 213–220.

[4] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout, "Isolated word recognition with the liquid state machine: a case study," *Information Processing Letters*, vol. 95, no. 6, pp. 521–528, 2005.

[5] A. Ghani, T. M. McGinnity, L. P. Maguire, and J. Harkin, "Neuro-inspired speech recognition with recurrent spiking neurons," in *Artificial Neural Networks-ICANN 2008*. Springer, 2008, pp. 513–522.

[6] Y. Zhang, P. Li, Y. Jin, and Y. Choe, "A digital liquid state machine with biologically inspired learning and its application to speech recognition," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, no. 11, pp. 2635–2649, Nov. 2015.

[7] B. Schrauwen, M. DHaene, D. Verstraeten, and J. Van Campenhout, "Compact hardware liquid state machines on fpga for real-time speech recognition," *Neural Networks*, vol. 21, no. 2, pp. 511–523, 2008.

[8] S. Roy, A. Banerjee, and A. Basu, "Liquid state machine with dendritically enhanced readout for low-power, neuromorphic vlsi implementations," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 8, no. 5, pp. 681–695, 2014.

[9] Q. Wang, Y. Jin, and P. Li, "General-purpose lsm learning processor architecture and theoretically guided design space exploration," in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*. IEEE, 2015, pp. 1–4.

[10] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.

[11] E. Goodman and D. Ventura, "Effectively using recurrently-connected spiking neural networks," in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 3. IEEE, 2005, pp. 1542–1547.

[12] D. Norton and D. Ventura, "Improving liquid state machines through iterative refinement of the reservoir," *Neurocomputing*, vol. 73, no. 16, pp. 2893–2904, 2010.

[13] D. Norton and D. A. Ventura, "Preparing more effective liquid state machines using hebbian learning," 2006.

[14] H. Paugam-Moisy, R. Martinez, and S. Bengio, "Delay learning and polychronization for reservoir computing," *Neurocomputing*, vol. 71, no. 7, pp. 1143–1158, 2008.

[15] A. Lazar, G. Pipa, and J. Triesch, "Sorn: a self-organizing recurrent neural network," *Frontiers in computational neuroscience*, vol. 3, 2009.

[16] S. V. Notley and A. Grüning, "Improved spike-timed mappings using a tri-phasic spike timing-dependent plasticity rule," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–6.

[17] F. Xue, Z. Hou, and X. Li, "Computational capability of liquid state machines with spike-timing-dependent plasticity," *Neurocomputing*, vol. 122, pp. 324–329, 2013.

[18] J. Yin, Y. Meng, and Y. Jin, "A developmental approach to structural self-organization in reservoir computing," *Autonomous Mental Development, IEEE Transactions on*, vol. 4, no. 4, pp. 273–289, 2012.

[19] G.-q. Bi and M.-m. Poo, "Synaptic modification by correlated activity: Hebb's postulate revisited," *Annual review of neuroscience*, vol. 24, no. 1, pp. 139–166, 2001.

[20] A. Morrison, M. Diesmann, and W. Gerstner, "Phenomenological models of synaptic plasticity based on spike timing," *Biological cybernetics*, vol. 98, no. 6, pp. 459–478, 2008.

[21] N. K. Kasabov, "Neucube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data," *Neural Networks*, vol. 52, pp. 62–76, 2014.

[22] N. Kasabov, N. M. Scott, E. Tu, S. Marks, N. Sengupta, E. Capecci, M. Othman, M. G. Doborjeh, N. Murli, R. Hartono *et al.*, "Evolving spatio-temporal data machines based on the neucube neuromorphic framework: Design methodology and selected applications," *Neural Networks*, 2015.

[23] C. C. Petersen, R. C. Malenka, R. A. Nicoll, and J. J. Hopfield, "All-or-none potentiation at ca3-ca1 synapses," *Proceedings of the National Academy of Sciences*, vol. 95, no. 8, pp. 4732–4737, 1998.

[24] D. H. O'Connor, G. M. Wittenberg, and S. S.-H. Wang, "Graded bidirectional synaptic plasticity is composed of switch-like unitary events," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 27, pp. 9679–9684, 2005.

[25] J. M. Brader, W. Senn, and S. Fusi, "Learning real-world stimuli in a neural network with spike-driven synaptic dynamics," *Neural computation*, vol. 19, no. 11, pp. 2881–2912, 2007.

[26] "The TI46 Speech Corpus," http://catalog.ldc.upenn.edu/LDC93S9.

[27] E. M. Izhikevich, J. A. Gally, and G. M. Edelman, "Spike-timing dynamics of neuronal groups," *Cerebral cortex*, vol. 14, no. 8, pp. 933–944, 2004.

[28] P. A. Appleby and T. A. Elliott, "Synaptic and temporal ensemble interpretation of spike-timing-dependent plasticity," *Neural computation*, vol. 17, no. 11, pp. 2316–2336, 2005.

[29] S. K. Esser, A. Andreopoulos, R. Appuswamy, P. Datta, D. Barch, A. Amir, J. Arthur, A. Cassidy, M. Flickner, P. Merolla *et al.*, "Cognitive computing systems: Algorithms and applications for networks of neurosynaptic cores," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, 2013, pp. 1–10.

[30] D. J. Amit and S. Fusi, "Constraints on learning in dynamic synapses," *Network: Computation in Neural Systems*, vol. 3, no. 4, pp. 443–464, 1992.

[31] D. J. Amit and S. Fusi, "Learning in neural networks with material synapses," *Neural Computation*, vol. 6, no. 5, pp. 957–982, 1994.

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

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