ESANN'2005 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 27-29 April 2005, d-side publi., ISBN 2-930307-05-6.

# Isolated word recognition using a Liquid State Machine

David Verstraeten and Benjamin Schrauwen and Dirk Stroobandt *

Ghent University - ELIS
Sint-Pietersnieuwstraat 41, 9000 Ghent - Belgium
{dvrstrae,bschrauw,dstr}@elis.ugent.be

**Abstract**. An implementation of the recently proposed concept of the Liquid State Machine using a Spiking Neural Network (SNN) is trained to perform isolated word recognition. We investigate two different speech front ends and different ways of coding the inputs into spike trains. The robustness against noise added to the speech is also briefly researched. It turns out that a biologically realistic configuration of the LSM gives the best result, and that it performs very well for the task of speech recognition.

## 1 Introduction

The Liquid State Machine is a recent computational model whose structure is depicted in Figure 1. The liquid consists of a recurrent network of non-linear interacting computational nodes (in this case spiking neurons [1], but others are possible: see [2], [3] or [4]) with an internal state. The set of all the internal states of these nodes form the *liquid state*. This liquid state serves as input to a memoryless readout function that extracts the actual output of the LSM. A more complete description of this powerful computational framework can be found in [5].

We tested this LSM for the rather practical task of recognizing isolated spoken digits. In this context, we investigated two different speech front ends: Mel Frequency Cepstral Coefficients (MFCC), a technique often used in traditional speech processing systems, and the Lyon Passive Ear model, a biologically realistic model of the human inner ear. We also tested three different ways of coding the output of these speech processing steps into spike trains: the classical Poisson spike coding, the BSA filter coding scheme , and a Leaky Integrate & Fire (LIF) neuron. Finally we briefly investigate the robustness of our word recognizer against noise added to the speech inputs.

In Section 2 we describe the parameters for the experiments we performed. Next, in Section 3 we introduce the two speech front ends we tested, and in Section 4 we discuss the three spike coding schemes that were used. In Section 5 we present and discuss the results of our measurements. In Section 6 we briefly investigate the performance of the system under different noisy circumstances. Finally, in Section 7, we formulate some conclusions about our research.

*This research was partially funded by the Fund for Scientific Research - Flanders (FWO)

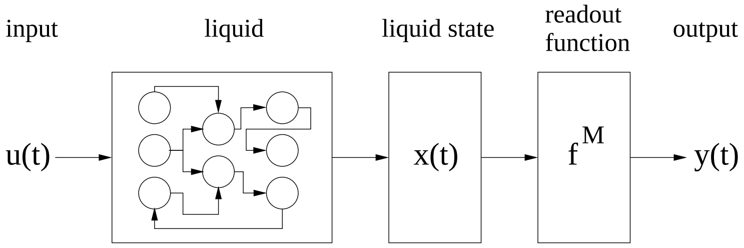

Fig. 1: The liquid state machine.

## 2 Experimental setup

The experiments for this publication were done using the Matlab LSM toolbox, described in [6]. The parameters of the neurons were inspired by biological data and are as follows (taken from [7]): membrane time constant 30 ms, firing threshold 15 mV, reset voltage 13.5 mV and an absolute refractory period of 3 ms for excitatory neurons and 2 ms for inhibitory neurons. A more complete description of the neuron parameters is given in [7].

The network connections in the liquid are randomly generated, with a probability of a connection being placed between neurons $a$ and $b$ defined as $P_{conn}(a,b) = C \cdot e^{\frac{-D^2(a,b)}{\lambda^2}}$ with $D(a,b)$ the Euclidean distance (in 3D space) between the two neurons. Both the average amount of connections as the average distance between connected neurons are controlled by the parameter $\lambda$, which in our case has the value 2. This parameter setting generates mainly local connections.

Because of the stochastic construction of the liquids and their connections, there exists some variation between the performance of different liquids of the same size. We therefore simulated different liquids of the same size and calculated the average performance for a given size, but we were sometimes hindered by memory limitations. Since a liquid of a given size can still have a varying number of internal connections, we were not always able to evaluate every liquid size the same number of times.

We trained and tested our LSM using a subset of the TI46 speech corpus consisting of ten different utterances of the digits ‘zero’ to ‘nine’, spoken by five different speakers. Of this set, 300 samples were used for training and 200 for testing.

The readout function used in our experiments consists of a linear classifier that maps the liquid state onto the output classes using a weight matrix $\mathbf{w}$, i.e. $C[\mathbf{x}(t)] = \Theta[\mathbf{w} \cdot \mathbf{x}(t) + w_0]$. The weight matrix is found using pseudo matrix inversion. Previous research [8] has shown that this simple readout function performs better in this case than more advanced classifiers such as a Fisher discriminant or a pool of parallel perceptrons. The readout function computes an output based on the liquid state every 20 ms, and the final output of the LSM based on a given input is determined using the winner take all principle after the full input sample is presented.

We measured the performance of the LSM for this specific task as the Word Error Rate (WER): the fraction of incorrectly classified words as a percentage of the total number of presented words: $WER = 100 \cdot \frac{N_{nc}}{N_{tot}}$ with $N_{nc}$ the number of incorrectly classified samples, and $N_{tot}$ the total number of samples presented.

## 3 Speech coding

### 3.1 Mel-Frequency Cepstral Coefficients

Mel-Frequency Cepstral Coefficients (MFCC) is the de facto standard preprocessing technique in the field of speech recognition. The MFCC are calculated as follows: (1) the sample data is windowed using a hamming window and a FFT is computed, (2) the magnitude is run through a so-called mel-scale[1] filter bank and the $log_{10}$ of these values is computed, (3) a cosine transform is applied to reduce the correlation among the individual features. The result is the so-called *cepstrum.*

### 3.2 Lyon Passive Ear

The Lyon Passive Ear model [9] is a model of the human inner ear or cochlea, which describes the way acoustic energy is transformed and converted to neural representations. The algorithm models the most important properties of the cochlea and hair cells. The model consists of a filter bank which closely resembles the selectivity of the human ear to certain frequencies, followed by a series of half-wave rectifiers (HWRs) and adaptive gain controllers (AGCs) both modeling the hair cell response.

Note that this form of preprocessing is computationally more intensive than the use of a MFCC front end. It takes about three to five times as long to compute on a conventional processor.

## 4 Spike coding

The communication between the neurons in the liquid is done through the use of spikes. This means that the output from the speech processing front ends needs to be converted from analog values into a series of spike trains. The coding of stimuli into spike trains has been researched extensively [10, 1, 11], and several methods have been proposed, such as time to first spike coding, population coding and correlation coding.

In this publication we compare three different methods for extracting the spike trains from the analog outputs of the speech front ends. First, we use a standard Poisson coding, whereby the spike trains are generated by interpreting the analog outputs of the speech processing as instantaneous firing rates. The spikes are generated by a Poisson process based on these firing rates.

[1] A mel-scale is a non-linear transformation of the frequency domain to model the human selectivity to certain frequency bands.

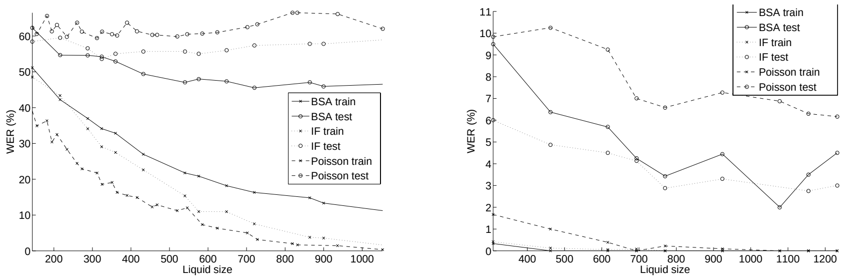

Fig. 2: Performance results for the MFCC front end (left) and the Lyon front end (right) using three different spike coding schemes. Note the difference in scale on the WER axis between both figures.

Secondly, we use a LIF neuron as a way to code the analog values into spike trains: this biologically realistic model of a real neuron takes an input current (an analog value) as input, and produces spikes in response to this current. This coding scheme is computationally the most intensive due to the complexity of the model.

Finally, we use BSA [12] to code the speech processing output into spike trains, a coding algorithm that assumes the decoding of the spike trains will take place using a linear filter. This is the case for our implementation of the LSM: the liquid response (i.e. the set of all spike times of the neurons in the liquid) is decoded into the liquid state (analog values) using an exponential filter, before being fed into the readout function.

## 5 Results

The results of our measurements are given in Figure 2. It appears that the standard MFCC technique is not well suited to be used as a speech front end for the LSM. Performance is poor and for two coding schemes (LIF and Poisson) does not even increase for larger liquids. It can also be noted that the performance for the BSA-coding is worse on the training set than the others, but is better on the test set. This suggests that the LSM generalizes better using BSA in the case of an MFCC front end. The Lyon Passive Ear front end performs far better: the larger liquids achieve an average WER of 3% using the LIF coding scheme, and the best liquid attained a WER of 0.5%.

For comparison, we present two other speech recognition systems tested with a comparable dataset. Sphinx4 is a recent speech recognition system developed by Sun Microsystems [13], using Hidden Markov Models (HMMs) and an MFCC front end. When it is applied to the full TI46 database, a word error rate (WER) of 0.168% is achieved. The best LSM from our experiments achieved a WER of 0.5%. While slightly worse than the state-of-the-art, we point out that the LSM offers a number of advantages over HMMs. HMMs tend to be sensitive to noisy inputs and are usually biased towards a certain speech database.

An additional comparison can be made by looking at the results described in [14]. There, a recurrent SNN is used with so-called Long Short-Term Memory (LSTM). It is trained for the same subset of TI46. A WER of 2% was achieved.

We can conclude that the LSM passes the test of isolated word recognition very well and rivals the performance of standard HMM based techniques and other kinds of SNN solutions.

## 6 Noisy inputs

We also investigated the noise robustness of the LSM by adding three different types of noise commonly found in day to day environments: speech babble (B), white-noise (W) and car interior noise (C) from the NOISEX noise database. We trained a random liquid of 1232 neurons on clean data, added noise to the test set at sound levels of 30, 20 and 10 dB and tested the performance with this corrupted test set. The data was preprocessed using the Lyon Passive Ear model and coded into spike trains using BSA.

We also cite the best results from [15], which describes the Log Auditory Model or LAM as a speech front end specifically designed to be noise-robust which is tested using a HMM based speech recognition system. However, the dataset consists in this case of isolated digits from the TIDIGITS database, which is not identical but still comparable to our dataset. Therefore, the results in table 1 are indicative and not quantitative. Also, note that here the performance is expressed as a recognition score to allow easy comparison.

| | | Clean | 30 dB | 20 dB | 10 dB |
|---|---|---|---|---|---|
| C | **LSM** | **95.5%** | **89.5%** | **87%** | **84.5%** |
| | LAM | 98.8% | 98.6% | 98.8% | 98.6% |
| B | **LSM** | - | **91.5%** | **89.5%** | **82%** |
| | LAM | - | 98.4% | 93.2% | 72.5% |
| W | **LSM** | - | **88%** | **82.5%** | **79.5%** |
| | LAM | - | 98.4% | 95.7% | 72.7% |

Table 1: The robustness of the LSM against different types of noise.

It turns out that the LSM has good noise-robustness. The HMM performs better for low noise levels, but the performance of the LSM decays more gradually with increasing noise levels.

## 7 Conclusion

In this paper we used the SNN interpretation of the LSM to the task of isolated word recognition with a limited vocabulary. We explored two speech front ends: the standard MFCC technique, and the biologically realistic Lyon Passive Ear model. We used three different ways of generating the spike trains: the simple Poisson coding, the BSA filter coding scheme and the biological LIF model. We also investigated the sensitivity to different types of noise commonly found in real-world applications.

Our results show that the LSM is well suited for this task. We have found that the performance is far better when the speech is preprocessed using a biologically realistic model of the human cochlea than using the classic MFCC speech front end, especially when the spikes are generated using a realistic model of a neuron. In addition to good speech recognition performance on clean test data, our results show a very good robustness of the LSM for different types of noise. Results from [5] show that this noise-robustness is also observed with other types of input than speech.

## References

[1] W. Gerstner and W. M. Kistler. *Spiking Neuron Models.* Cambridge University Press, 2002.

[2] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

[3] T. Natschläger, N. Bertschinger, and R. Legenstein. At the edge of chaos: Real-time computations and self-organized criticality in recurrent neural networks. *Submitted for publication*, 2004.

[4] C. Fernando and S. Sojakka. Pattern recognition in a bucket. In *Proc. 7th European Conference on Artificial Life*, pages 588–597, 2003.

[5] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[6] T. Natschläger, H. Markram, and W. Maass. Computer models and analysis tools for neural microcircuits. In R. Kötter, editor, *A Practical Guide to Neuroscience Databases and Associated Tools*, chapter 9. Kluver Academic Publishers (Boston), 2002.

[7] W. Maass, T. Natschläger, and H. Markram. A model for real-time computation in generic neural microcircuits. *Proc. of NIPS 2002*, 15:229–236, 2003.

[8] David Verstraeten. Een studie van de Liquid State Machine: een woordherkenner. Master's thesis, Ghent University, ELIS department, 2004.

[9] R.F. Lyon. A computational model of filtering, detection and compression in the cochlea. *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing*, May 1982.

[10] W. Maass and C. M. Bishop. *Pulsed Neural Networks.* Bradford Books/MIT Press, Cambridge, MA, 2001.

[11] P. Dayan and L. F. Abbott. *Theoretical Neuroscience.* MIT Press, Cambridge, MA, 2001.

[12] Benjamin Schrauwen and Jan Van Campenhout. BSA, a fast and accurate spike train encoding scheme. In *Proceedings of the International Joint Conference on Neural Networks*, 2003.

[13] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel. Sphinx-4: A flexible open source framework for speech recognition. Technical report, Sun Microsystems Inc., 2004.

[14] A. Graves, D. Eck, N. Beringer, and J. Schmidhuber. Biologically plausible speech recognition with LSTM neural nets. In *Proc. of Bio-ADIT*, 2004.

[15] Y. Deng, S. Chakrabartty, and G. Cauwenberghs. Analog auditory perception model for robust speech recognition. In *Proc. IEEE Int. Joint Conf. on Neural Network*, 2004.