ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Continuous Digit Recognition in Noise: Reservoirs can do an excellent job!

Azarakhsh Jalalvand, Fabian Triefenbach, Jean-Pierre Martens

Ghent University - IBBT, ELIS Multimedia Lab, Sint-Pietersnieuwstraat 41, B-9000, Ghent, Belgium
Azarakhsh.Jalalvand@elis.ugent.be

10.21437/Interspeech.2012-491

Abstract

In this paper a formerly proposed continuous digit recognition system based on Reservoir Computing (RC) is improved in two respects: (1) the single reservoir is substituted by a stack of reservoirs, and (2) the straightforward mapping of reservoir outputs to state likelihoods is replaced by a trained non-parametric mapping. Furthermore, it is shown that a reservoir-based method can improve a model trained on clean speech to work better in a noisy condition from which it has a number of unknown digit string recordings available. The first two improvements have lead to a system that outperforms a HMM-based system with the same noise robust features as input. The model adaptation offers a promising supplementary gain at modest noise levels.

Index Terms: Reservoir Computing, Acoustic Modeling, Model Adaptation, Noise Robustness

1. Introduction

Standard Hidden Markov Models (HMMs) incorporate Gaussian Mixture Models (GMMs) to compute state-level acoustic likelihoods. Such systems have reached a high level of performance, but they remain very sensitive to mismatches between the training and the test circumstances. Many research efforts have been directed towards the development of novel front-end and/or back-end techniques [1, 2, 3] for making the systems more resistant to these mismatches.

In this paper an RC-HMM hybrid for continuous digit recognition is investigated. The acronym RC stands for Reservoir Computing [4]. It indicates that the transformation of input features to state likelihoods involves a reservoir, defined as a pool of non-linear and recurrently connected computational nodes (called neurons) with randomly fixed weights.

It was already demonstrated last year [5] that an RC-HMM hybrid comprising one reservoir can yield good performance for isolated and continuous digit recognition in clean and noisy circumstances (tested on Aurora-2). However, for large signal-to-noise ratios (SNR) the hybrid was still outperformed by a traditional HMM system working with the same input features.

In this paper we report on how we succeeded in creating a new hybrid that either equals or surpasses the HMM system for all tested SNRs. Furthermore, we propose a reservoir-based method for adapting a hybrid that was trained on clean speech to work better in a noisy condition after it has seen some untranscribed digit string recordings representative of that condition.

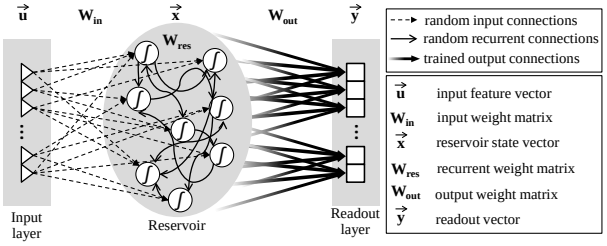

Figure 1: A basic RC system consists of a reservoir of fixed nonlinear & recurrently connected nodes and a set of trainable output nodes.

2. Reservoir Computing

Figure 1 shows the architecture of a basic RC system. It is composed of one reservoir and a layer of linear nodes which 'read out' the reservoir nodes. The weights of the reservoir nodes are fixed (not trained) and drawn from a random distribution. Precautions are taken to guarantee that a stable dynamical system is obtained and that the new inputs and the previous outputs contribute in a balanced way to the new outputs of the reservoir nodes (see [6] for details). The weights of the readout nodes are trained so as to pursue that a particular node is high when the corresponding digit state is visited and low otherwise.

Suppose that u_t, x_t and y_t respectively represent the input vector, the reservoir state vector and the output vector at time t and that the matrices W_{in}, W_{res} and W_{out} comprise the weights of the various connections. Then the RC system performs the following computations:

$$x_t = f_{res}\left(W_{res}\, x_{t-1} + W_{in}\, u_t\right) \quad (1)$$

$$y_t = W_{out}\, x_t \quad (2)$$

with f_{res} being a non-linear function (in our case $f_{res}(x) = \tanh(x)$).

To further extend the integration of information over time, the memoryless reservoir nodes are replaced by

Leaky Integrator Neurons (LIN) [7]. Equation (1) then changes to

$$x_t = (1-\lambda)\, x_{t-1} + \lambda\, f_{res}\,(W_{res}\, x_{t-1} + W_{in}\, u_t)$$

with $0 \leq \lambda \leq 1$ determining the integration time.

2.1. An RC-HMM hybrid decoder

The RC-HMM hybrid we proposed in [5] comprised a single-state HMM to model inter-digit silences and a left-to-right model with no skips and $S = 5$ states per digit. Since we will test our system on Aurora-2 we discern 11 digits (two variants of '0') leading to a total of $11 \times 5 + 1 = 56$ HMM states. The joint probability of the acoustic feature sequence U and an HMM state sequence Q is computed as

$$P(Q,U) = \prod_{t=1}^{T} P(q_t|q_{t-1})\; P(u_t|q_t), \qquad (3)$$

where $P(u_t|q_t)$ is derived from the readouts y_t (see further) and $P(q_t|q_{t-1})$ is a state transition probability.

The weights of the readout nodes minimize the mean squared distance between the output vectors y_t and the corresponding desired output vectors d_t in a training set. The desired output vectors d_t point to the state q_t^* of the state sequence Q^* that maximizes the aforementioned joint probability. One finds the weights by solving a set of linear equations (see [5]). However, since a retrained reservoir can lead to another q_t^*, the process is repeated a few times until convergence.

3. Improvements of the RC-HMM hybrid

Our research on RC-based continuous phoneme recognition [6] demonstrated that the recognition can be improved by putting two to three basic RC-systems in cascade. Apparently, the second RC-system is capable of discovering regularities in the shortcomings of the first one, and so on. Even though the cascading approach did not lead to improved large vocabulary recognition, we wanted to test it for connected digit recognition. The architecture of the new RC-HMM hybrid is depicted on Figure 2. The inputs to the first reservoir are the noise-resistant MSVA features proposed in [2], plus their first and second order derivatives. The feature normalization is performed on complete utterances. The inputs to the second reservoir are the readout nodes of the first RC-system, and so on.

3.1. Training a multi-layer reservoir system

The training of a multilayer system proceeds layer per layer. The training of one layer is achieved by means of a Viterbi-training, as described in [5]. In a nutshell, it is an iterative process consisting of two steps per iteration: (1) use the current system to align the input vectors u_t to the states q_t of an utterance model derived from the known digit string and (2) solve a set of linear equations to determine the weights of the readout nodes that minimize the mean squared error between the computed and the desired values. In [5] it is indicated how to bootstrap the training of the first layer.

3.2. Mapping reservoir outputs to likelihoods

In [5] we assumed that[1] the reservoir outputs y_{ti} ($i = 1 \ldots 56$) are good approximations of the posterior probabilities $P(s_i|u_t)$ of the eligible states s_i (the value set of q_t). Based on that assumption, we formerly used $y_{ti}/P(s_i)$ as a proxy of the likelihood $P(u_t|s_i)$ needed in Equation (3). However, when measuring the posterior probabilities $P(s_i|y_i)$ on a development set, it turned out that the aforementioned assumptions are violated. Therefore, we now retrieve for each state s_i a lookup-table to map y_{ti} to a better estimate of $P(s_i|u_t)$.

The lookup table for state s_i is derived from a histogram of the y_{ti} of the frames of a development set that were assigned to state s_i and from the global histogram of all the y_{ti}. Starting with bins of 0.01 wide, the bins collecting too few examples (global histogram) are joined with their neighbors: the least populated bin is joined with the least populated neighboring bin and the process is continued until all bins contain enough examples (we used 100 examples for this).

Although the new mapping could have been applied to each readout layer, it was only applied to the last one as only these readouts have to be converted to likelihoods.

3.3. Experimental validation

The proposed modifications were validated on the Aurora-2 benchmark [8].

The readout layers were always trained on clean data (8440 utterances from 110 speakers) but tests were performed on clean and noisy data. We report mean results over test sets A - C and use the word error rate (WER) as the performance measure. We compare the RC-systems to a reference HMM embedding digit models with 16 states and GMMs with three mixtures in each state.

Table 1 shows the performances as a function of the number of layers, the number of reservoir nodes per layer and the activation/inhibition of the new output mapping. Per row, the clean speech result, the average result over SNRs between 20 and 0 dB, and the result for SNR = -5 dB are provided. The main findings are that doubling the size of a single layer system is not as helpful as adding an extra layer and non-linear readout mapping is always beneficial.

Adding one layer to the baseline (which is 1 layer, 4k nodes, old mapping) leads to a large gain (46% relative)

[1] making abstraction of a trivial linear mapping

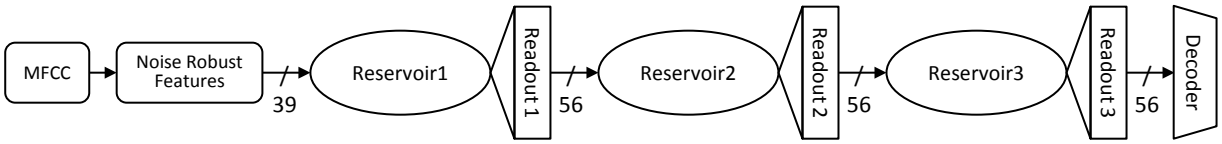

Figure 2: A noise robust multilayer RC-HMM hybrid for continuous digit recognition. Each reservoir is stimulated by the readouts of the previous reservoir.

Table 1: Continuous digit recognition results (average WER over Test A-C) for systems with different topologies, sizes and probability mappings.

nr of layers	nr of nodes	new map	Clean	Avg. 0-20dB	-5dB
HMM			**0.98**	16.86	76.66
1	4k	-	2.56	17.55	66.92
1	4k	y	2.32	15.68	64.00
1	8k	-	2.04	16.42	65.73
2	4k	-	1.37	14.68	61.17
2	4k	y	1.29	**13.73**	**59.34**
3	3k	-	1.25	14.99	60.36
3	4k	-	1.20	14.49	60.41
4	2k	-	1.46	16.49	61.79

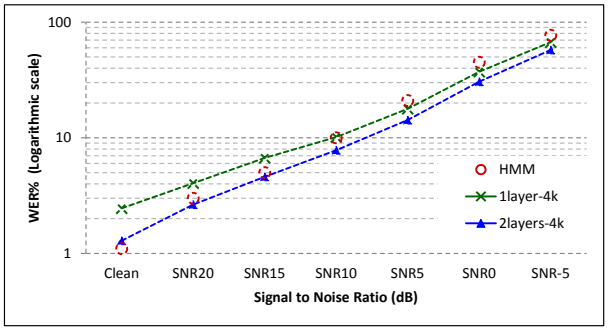

Figure 3: Recognition results (WER) as a function of SNR (Test A) for the reference (HMM), the single layer reservoir system (1layer-4k) and the two-layer system with non-linear output mapping (2layers-4k)

for clean speech and significant gains for noisy speech. Adding a third layer further improves the clean speech result, but it does not bring anything extra for the noisy conditions. From Figure 3 it follows that for high SNRs the two-layer system competes well with the HMM system now, whereas for low SNRs it is consistently better.

4. Model adaptation in RC-HMM hybrids

Given these good results, we wondered whether it would be possible to improve the performance in a certain noise condition by unsupervised adaptation (based on untranscribed recordings) of the system to that condition. We investigated three approaches

4.1. Linear transformation of the outputs

A very simple approach is to train a linear transformation of the readouts y_t to new outputs $y'_t = A\, y_t + b$. The aim is to minimize the mean squared distance between the new outputs and the desired outputs in the adaptation data. As the desired outputs are a priori unknown, they are identified from the states on the most likely path through the digit loop utterance model. Strategies for computing digit confidences and for retaining only the desired outputs corresponding to digits which were recognized with sufficient confidence, did not lead to a better result and are therefore not further considered here.

4.2. Retraining the reservoir weights

Since the readout nodes are linear, the linear transformation of readouts is equivalent to a linear transformation of the readout node parameters (weights). In order to learn the latter transformation one can see the problem as one of training the readouts with the original training data supplemented with the adaptation data. If X and X_A are matrices whose columns represent the reservoir states of the N training and N_A adaptation frames, and if D and D_A represent the corresponding desired outputs, then the adapted readout weights can be obtained as

$$W_{out}^{A} = (X^T X + \alpha X_A^T X_A + \epsilon\,(N + \alpha N_A)\, I)^{-1} (X^T D + \alpha X_A^T D_A) \quad (4)$$

where α is a factor that controls how much the adaptation data contribute to these weights.

The problem with this method is that it needs access to the quite large matrices $X^T X$ and $X^T D$ and that the inversion of a large matrix is time consuming. So the method is merely used as a reference method against which to compare the other methods.

4.3. Adding an extra reservoir

Instead of learning a linear transformation one could also add another reservoir layer (see Figure 4) that is solely trained on adaptation data. This reservoir then achieves a non-linear transformation with memory. The number

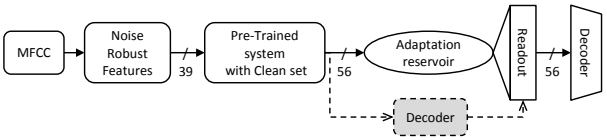

Figure 4: An extra layer is trained on the adaptation data. The grey decoder provides the spoken digit state sequence and therefore the desired outputs for the reservoir training.

Table 2: Performance of the baseline RC-based recognizer on Test A and relative improvements obtained with the three adaptation approaches.

SNR	WER%	Relative Improvement (%)		
	Baseline	Transform	Retrain	Add Layer
Clean	1.48	-8	0	7
SNR20	2.68	14	18	19
SNR15	4.69	19	29	31
SNR10	8.03	5	7	4
SNR5	14.79	9	13	9
SNR0	30.08	-1	3	0
SNR-5	55.66	-8	-5	-8

of trainable parameters is equal to the number of eligible states times the number of nodes in the added reservoir. In a practical implementation one could create a sufficiently large reservoir and connect more of its nodes to the readout nodes as more adaptation data become available. This way the number of trainable parameters can be gradually increased.

Since the construction of a good lookup-table for mapping readouts to posterior probabilities takes a lot of data, we keep the non-parametric output mapping of the readouts of the second reservoir but exploit the new readouts in the traditional way.

4.4. Experimental validation

We conducted experiments with 3 minutes of adaptation data taken from the development set. This way we could continue to use the same test sets as before. Table 2 shows the performances for Test A with the optimal α (method 2) and the optimal adaptation reservoir size (method 3). The latter was equal to 250.

The main conclusions are that method 3 works as well as method 2 and that it offers a promising gain for SNRs of 20 and 15 dB. Obviously there is no improvement for clean speech as this condition already matches with the training condition.

That there is no improvement for SNR $\leq$ 10 dB anymore is owed to the fact that the desired outputs retrieved by the decoder become unreliable for these low SNRs.

5. Conclusions and future work

We proposed several improvements of a formerly presented RC-HMM hybrid for continuous digit recognition. The new system now surpasses a HMM system that is supplied with the same noise robust acoustic features. Moreover a system trained on clean speech can be adapted by means of a reservoir-based method to a new system that works better in another condition (noise type and SNR). The adaptation is performed in an unsupervised way on the basis of a limited amount (3 minutes) of adaptation data. The improvement is only significant (up to 30% relative) though as long as reliable training outputs can be generated by the non-adapted decoder. Our future goal is to investigate whether the proposed adaptation method also works satisfactory for e.g. accent-specific recognition where more adaptation data are likely to be available.

6. Acknowledgments

The research leading to the results presented here has received funding from the European Community's Seventh Framework Program (FP7) under grant agreement 231267 "Self-organized recurrent neural learning of language processing" (ORGANIC) and from FWO under grant G.0088.09N "Reservoir computing for auditive pattern recognition" (RECAP).

7. References

[1] J. F. Gemmeke, "Noise robust asr: Missing data techniques and beyond," Ph.D. dissertation, Radboud Universiteit Nijmegen, The Netherlands, 2011.

[2] C. Chia-Ping and J. A. Bilmes, "Mva processing of speech features," *IEEE Transactions on Audio, Speech and Language processing*, 2007.

[3] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, "Linear hidden transformation for adaptation of hybrid ann/hmm models," *Speech Communication*, vol. 49, pp. 827–835, 2007.

[4] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks," 2001. [Online]. Available: http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf

[5] A. Jalalvand, F. Triefenbach, D. Verstraeten, and J.-P. Martens, "Connected digit recognition by means of reservoir computing," in *Proc. of INTERSPEECH*, 2011, pp. 1725–1728.

[6] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-P. Martens, "Phoneme recognition with large hierarchical reservoirs," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2010, pp. 2307–2315.

[7] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Networks*, vol. 20, pp. 391–403, 2007.

[8] H. Hirsh and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ASR*, 2000.