On Improving the Classification Capability of Reservoir Computing for Arabic Speech Recognition

Abdulrahman Alalshekmubarak and Leslie S. Smith

Dept. of Computing Science, University of Stirling,
Stirling FK9 4LA, UK
{aal,lss}@cs.stir.ac.uk

Abstract. Designing noise-resilient systems is a major challenge in the field of automated speech recognition (ASR). These systems are crucial for real-world applications where high levels of noise tend to be present. We introduce a noise robust system based on Echo State Networks and Extreme Kernel machines which we call ESNEKM. To evaluate the performance of the proposed system, we used our recently released public Arabic speech dataset and the well-known spoken Arabic digits (SAD) dataset. Different feature extraction methods considered in this study include mel-frequency cepstral coefficients (MFCCs), perceptual linear prediction (PLP) and RASTA- perceptual linear prediction. These extracted features were fed to the ESNEKM and the result compared with a baseline hidden Markov model (HMM), so that nine models were compared in total. ESNEKM models outperformed HMM models under all the feature extraction methods, noise levels, and noise types. The best performance was obtained by the model that combined RASTA-PLP with ESNEKM.

Keywords: Reservoir computing, Speech recognition, PLP, MFCC, RASTA-PLP, Speech corpus, Arabic language.

1 Introduction

Speech communication is one of the most distinguishing capabilities of humans. Indeed the ability to conduct a conversation was introduced in the early days of computation as a measurement of intelligence in the well-known Turing test. Automatic speech recognition (ASR), mapping the acoustic signal into a string of words, forms the first part in such an intelligent system. A major challenge in the field of automated speech recognition (ASR) lies in designing noise-resilient systems. These systems are crucial for real-world applications where high levels of noise are often present. In this paper, we introduce a noise robust system based on Echo State Networks (ESN)[1] and the Extreme Kernel Machine (EKM)[2] which we call ESNEKM to improve the performance of ESN in the presence of noise. The proposed model maintains the main attractions of ESN, being very fast to train and able to handle multi-class classification problems. The paper

S. Wermter et al. (Eds.): ICANN 2014, LNCS 8681, pp. 225–232, 2014.

is organised as follows. A brief review of ESN and EKM is presented in the section 2. The proposed system is introduced in section 3, and section 4 contains a detailed report of our experiments to ensure the reproducibility of our results. We discuss our results in section 5, and present our conclusions in section 6.

2 Background

2.1 Reservoir Computing

Reservoir Computing is an emerging field that offers a novel approach to training Recurrent Neural Networks. Originally proposed in 2002, its popularity has grown rapidly due to the simplicity of its implementation and its robust performance [3]. RC contains several techniques that derived from different backgrounds. However, all of them share the main idea of RC: random initialisation of the weights of the recurrent nodes and only learning weights in the output layer which implements a simple readout function. The two major approaches under the umbrella of RC are the Echo State Network (ESN) and the Liquid State Machine (LSM).

Echo State Network. ESN was introduced by Jaeger in 2001 [1] and has been applied in different real world applications where it proved to achieve a superior performance, or similar, compared to the state of the art algorithms. This success lead to a wide acceptance of this technique in the field and encouraged researchers to conduct studies that aim to explore the fundamental properties and behaviour nature of ESN that lie behind its performance. Another, rather more empirical effort has also been made to investigate the applicability of ESN on new and more challenging real world problems and to conduct extensive comparisons among the state of the art techniques [4]. The ESN model is characterised in the following way. First, $\mathbf{W^{in}}$, which is an m by n matrix (where m is the size of the input vector and n is the size of the reservoir), is initialised randomly. Second, $\mathbf{W^{res}}$, which is an n by n matrix, is initialised randomly as well and scaled to obtain the desirable dynamics. Another important component of this model is the fading memory (forgetting) parameter α, which plays a major role in controlling the memory capacity of the reservoir. The model update equations are as follows[5]:

$$\bar{x}(t) = f(\mathbf{W^{in}}[1; u(t)] + \mathbf{W^{res}} x(t-1)) \tag{1}$$

$$x(t) = (1-\alpha)x(t-1) + \alpha\bar{x}(t) \tag{2}$$

where $x(t)$ is the reservoir's state at time t, $u(t)$ is the input signal at time t and f is a nonlinear transfer function: commonly logistic or tanh is applied. The response of the reservoir is dynamic and the class labels of training are used to train a simple linear read-out function by learning the weights of the output layer $\mathbf{W^{out}}$. This is typically accomplished by applying the pseudo-inverse equations:

$$\mathbf{W^{out}} = (\mathbf{X^T X})^{-1}\mathbf{X^T Y} \tag{3}$$

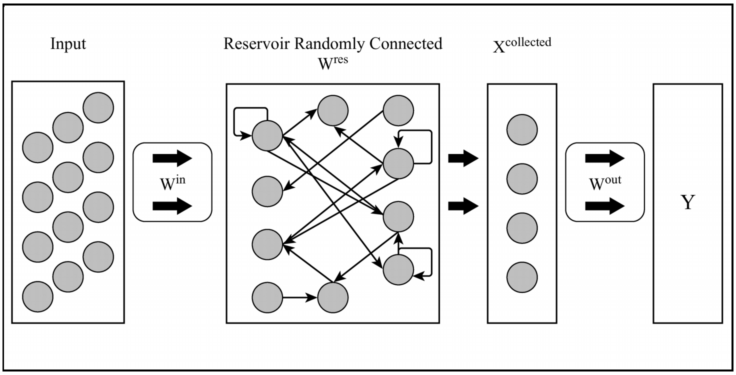

Fig. 1. The structure of the ESN and readout system. On the left, the input signal is fed into the reservoir network through the fixed weights $\mathbf{W^{in}}$. The reservoir network recodes these, and the output from the network is read out using the readout network on the right $\mathbf{W^{out}}$, which uses the the learned weights.

2.2 Extreme Learning Machine (ELM)

The ELM was proposed as an efficient model to train the single-hidden layer feedforward networks (SLFNs) by Huang in 2004 [6]. The basic concept is similar to reservoir computing in that both approaches map the input to a higher dimensional space using random weights and only learn the weights of the output layer. The main difference is that ELM, unlike RC, does not use recurrent nodes, which prevents it from modelling dynamic systems. ELM has been applied successfully in many real-world conditions in a variety of fields [7] [8].

Instead of using a relatively small number of nodes in the hidden layer and applying a powerful optimisation technique such as back-propagation, which suffers from several well-known issues (e.g. local minima, sensitivity to initialisation weights, implementation complexity, tendency to over-fit, long training time), ELM uses a very large number of nodes, typically more than 1,000 and only uses a simple read-out function at the output layer. Despite the use of this large number of nodes, ELM offers superior generalisation performance, which can be explained by the random weights applied on the learning and testing samples. This means the mapping mechanism is not based on the learning dataset. ELM can be described mathematically as follows [2]:

$$f_L(x) = \sum_{n=1}^{L} \beta_i h_i(x) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{4}$$

where $\boldsymbol{\beta}$ $=[\beta_1, ..., \beta_L]^T$ is the output learned weights by the simple linear readout function and $\mathbf{h}(\mathbf{x})$ $=[h(x)_1, ..., h(x)_L]^T$ is calculated by mapping the input vector by the random initialised weights. A variety of nonlinear functions may be chosen in the mapping layer: commonly, logistic or tanh is applied.

Researchers have demonstrated that ELM offers superior, or similar, performance to LS-SVMs and SVMs but trains more rapidly [2]. This, and the limited number of hyper-parameters that need to be selected, encouraged Huang to argue that ELM can promote real-time learning without human intervention. The main limitation of ELM is its inability to handle dynamic systems, which prevents its use in many real-world applications.

Extreme Kernel Machine (EKM). In the EKM version of ELM, the input vector $\boldsymbol{x}$ is not mapped to a higher dimensional space by a random matrix, but by a kernel. Similar to SVMs, a kernel is applied here, which means users do not have to know the actual mapping function. It is important to state the differences in applying the kernel among EKM, SVMs and LS-SVMs. SVMs and LS-SVMs are binary classifiers whereeas EKM is not, which allows it to deal with multi-label tasks efficiently. The main equation of EKM used to estimate the decision function from a training dataset is as follows [2]:

$$f(x) = h(x)\mathbf{H}^T(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{T} \tag{5}$$

$$= \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T \left(\frac{\mathbf{I}}{C} + \boldsymbol{\Omega}_{ELM}\right)^{-1} \mathbf{T}$$

where $\boldsymbol{\Omega}_{ELM_{i,j}} = K(x_i, x_j)$ and C is the regularisation parameter. As can be seen from the previous brief mathematical description, users do not need to specify the number of nodes used in the mapping layer. As in EKM, the length of the mapping function is the number of training samples. In other words, in EKM, the size of the mapping layer cannot exceed the number of training samples. However, this is not guaranteed in ELM.

3 Proposed System

The proposed model aims to improve the classification capability of the ESN by applying the EKM classifier on the output layer instead of the linear classifier used in the conventional approach. We have found few attempts in the literature [9][10] to overcome the limitation of the linear readout function and replace it with a nonlinear classifier. The added training time (or the binary nature of some classifiers such as SVMs) makes this impractical for many tasks, specifically for multi-label tasks with a relatively large number of classes. However, using EKM yields the benefits of using the nonlinear function while maintaining a single-shot convex solution that can handle multi-label tasks even when the number of labels is large. This is important not only from the efficiency aspect but also from reproducibility: many nonlinear classifiers such as multilayer perceptron are sensitive to their initial weights. In addition, we have found that dividing the signal into subparts and separately feeding these subparts to the reservoir improves the performance, particularly in the presence of noise. The proposed model is summarised in the following steps:

1. Divide the signal into subparts (in our experiments, we have found two parts is adequate).
2. Process these subparts by randomly initialising the reservoir and store the reservoir responses.
3. Train EKM classifier using reservoir responses (after normalisation) and the teaching signal (label).

4 Experiments

4.1 Datasets

Spoken Arabic Digits. Spoken Arabic Digits (SAD) is the only publicly accessible Arabic speech corpus. It is published in a processed format (as MFCCs) which prevents the development of novel feature extraction methods and comparison between them. It contains the Arabic digits 0- 9, with each digit spoken 10 times by 88 native speakers, 44 males and 44 females. Thus, it contains 8800 samples which are divided as follows: 6600 instances for training and 2200 instances for testing.

Arabic Speech Corpus for Isolated Words. The development of an Arabic speech corpus for isolated words has been conducted at the Department of Management Information Systems, King Faisal University. It contains about 10000 utterances of 20 words spoken by 50 native male Arabic speakers. The corpus has been made freely accessible for non-commercial use in the raw format (.wav files) and other formats to allow researchers to apply different feature extraction methods. The corpus has been recorded with a 44100 Hz sampling rate and 16-bit resolution, as well as two channels-stereo mode.

4.2 Feature Extraction and Noise Addition

In the SAD dataset, we used the processed MFCC features. It is not possible to compare different feature extraction methods or evaluate the performance of the proposed model in the presence of different types of noise. However, in the Arabic Speech Corpus for Isolated Words, dataset a variety of feature extraction methods have been considered, namely MFCCs, perceptual linear prediction (PLP) and RASTA-perceptual linear prediction. In addition, white noise and babble noise have been added to the test set in three levels of noise: 30 db, 20 db, 10 db.

4.3 Hyperparameters Optimisation

A validation set is used to optimise the hyperparameters of each model. This validation set is extracted from the training in the SAD set, and we report the result on the unseen test set. The optimisation has been carried out on the clean dataset only, meaning that all of these models have not been exposed to noisy data in the optimisation phase or in the training phase. In the ESN and ESNEKM we suggest fixing the size of the reservoir to a relatively small number, while optimising the rest of parameters reduces the required time to select the hyperparameters.

5 Results and Discussions

The results on the SAD, summarised in table 1, show the superior performance of the proposed system compared to the systems found in the literature. To account for the stochastic behaviour of ESN, we repeated the experiments ten times and report the mean and standard deviation. In this dataset (no noise added), the performance of ESN and ESNEKM is relatively similar; however, ESNEKM is more stable as can be seen from the value of its standard deviation.

Table 1. The results obtained by the proposed system , ESN and from the compared studies

System	Result
TM (Nacereddine Hammami et al, 2010) [11]	93.10%
CHMM (Nacereddine Hammami et al 2012) [12]	94.09%
LoGID (Paulo R. Cavalin et al,2012)[13]	95.99%
ESN(This work)	99.06% (0.23)
ESNEKM(This work)	99.16% (0.12)

Table 2 shows the results for the Arabic Speech Corpus for Isolated Words. In the absence of noise, ESN and ESNEKM were examined under all the considered feature extraction approaches, and approximately all of them provide a similar performance when fed to ESNEKM. This result does not hold true in the noisy sets as PLP-RASTA provides a superior performance regardless of the type or the level of noise. As in the previous experiment, ESNEKM provides much more stable results and outperforms the baseline model in all the sets. The best results on the clean and the noisy sets are achieved when combining PLP-RASTA with ESNEKM.

In our experiments ESNEKM provides a better performance compared to ESN even if the reservoir size is relatively small (100 nodes). However, there are some limitations related to the proposed system, and the added complexity in the output layer of the network (resulting from replacing the linear readout function with a nonlinear function) can be seen as the major issue. This issue includes the selection of the kernel and optimising its parameters. In addition, the testing time depends on the training size. Unlike support vector machines (SVM), the solution in EKM is not sparse.

Table 2. The results obtained by the proposed system, ESN and a baseline hidden Markov model (HMM). In ESN and ESNEKM, we report the means and the standard deviations over ten runs.

Dataset		Feature Extraction	HMM	ESN	ESNEKM
Clean		MFCCs	97.65%	98.97%(0.15)	99.59%(0.05)
		PLP	98.45%	99.16%(0.11)	99.31%(0.09)
		RASTA-PLP	98.8 %	99.38%(0.11)	99.69%(0.06)
White Noise	30 db	MFCCs	96.4%	98.03%(0.21)	99.05%(0.13)
		PLP	91.3%	90.13%(0.36)	97.59%(0.17)
		RASTA-PLP	98.1%	99.04%(0.11)	99.59%(0.06)
	20 db	MFCCs	85.29%	94.91%(0.37)	94.82%(0.30)
		PLP	51.13%	56.07%(6.66)	75.39%(0.97)
		RASTA-PLP	96.05%	97.32% (0.33)	98.41%(0.07)
	10 db	MFCCs	45.67%	77.19%(2.12)	79.50%(0.85)
		PLP	12.06%	19.83%(3.83)	35.35%(1.96)
		RASTA-PLP	81.99%	87.48%(1.47)	90.29%(0.53)
Babble Noise	30 db	MFCCs	95.85%	97.23 %(0.29)	99.35 %(0.18)
		PLP	97.05%	97.87%(0.36)	99.02%(0.06)
		RASTA-PLP	98.65%	99.22%(0.19)	99.65%(0.06)
	20 db	MFCCs	78.49%	89.72%(0.87)	94.41%(0.34)
		PLP	86.64%	89.47% (2.43)	96.64%(0.22)
		RASTA-PLP	96.75%	97.18%(0.42)	98.30%(0.14)
	10 db	MFCCs	31.77%	64.12%(2.31)	65.48%(0.86)
		PLP	54.23%	56.23%(4.82)	81.23%(0.32)
		RASTA-PLP	85.14%	85.45%(8.6)	90.76%(0.44)

6 Conclusions

A novel speech recognition model based on RC and EKM which we call ESNEKM was proposed, and was evaluated on a newly developed corpus and the well-known spoken Arabic digits (SAD). Different feature extraction methods considered in this study include mel-frequency cepstral coefficients (MFCCs), perceptual linear prediction (PLP) and RASTA-perceptual linear prediction. The result was compared with a baseline hidden Markov model (HMM), so that nine models were compared in total. These models were trained on clean data and then tested on unseen data with different levels and types of noise. ESNEKM models outperformed HMM models under all the feature extraction methods, noise levels, and noise types. The best performance was obtained by the model that combined RASTA-PLP with ESNEKM. Future work will include an investigation of the system usability in Arabic continuous speech and the possible use of a language model.

References

1. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Tecnical report GMD report 148 (2001)
2. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42, 513–529 (2012)
3. Verstraeten, D.: Reservoir computing: computation with dynamical systems. Electronics and Information Systems, Gent. Ghent University (2009)
4. Lukoševičius, M., Jaeger, H., Schrauwen, B.: Reservoir computing trends. KI-Künstliche Intelligenz, 1–7 (2012)
5. Lukoševičius, M.: A practical guide to applying echo state networks. Neural Networks: Tricks of the Trade, 659–686 (2012)
6. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: A new learning scheme of feedforward neural networks. In: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, vol. 2, pp. 985–990. IEEE (2004)
7. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing 70, 489–501 (2006)
8. Huang, G.B., Wang, D.H., Lan, Y.: Extreme learning machines: a survey. International Journal of Machine Learning and Cybernetics 2, 107–122 (2011)
9. Triefenbach, F., Martens, J.P.: Can non-linear readout nodes enhance the performance of reservoir-based speech recognizers? In: 2011 First International Conference on Informatics and Computational Intelligence (ICI), pp. 262–267 (2011)
10. Alalshekmubarak, A., Smith, L.S.: A novel approach combining recurrent neural network and support vector machines for time series classification. In: 2013 9th International Conference on Innovations in Information Technology (IIT), pp. 42–47 (2013)
11. Hammami, N., Bedda, M.: Improved tree model for arabic speech recognition. In: 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 5, pp. 521–526 (2010)
12. Hammami, N., Bedda, M., Nadir, F.: The second-order derivatives of mfcc for improving spoken arabic digits recognition using tree distributions approximation model and hmms. In: 2012 International Conference on Communications and Information Technology (ICCIT), pp. 1–5 (2012)
13. Cavalin, P.R., Sabourin, R., Suen, C.Y.: Logid: An adaptive framework combining local and global incremental learning for dynamic selection of ensembles of hmms. Pattern Recognition 45, 3544–3556 (2012)