Discriminating Noisy Sentences with Partially Recurrent Neural Networks

Classification des Phrases Bruitées avec les Réseaux Artificiels de Neuronnes Partiellement Récurrents

Ezin C. Eugène

Institut de Mathématiques et de Sciences Physiques
Unité de Recherche en Informatique et Sciences Appliquées
Université d'Abomey-Calavi
Porto-Novo, Bénin
eugene.ezin@imsp-uac.org
&
Faculté des Sciences et Techniques
Département de Mathématiques
Abomey-Calavi, Bénin.

RÉSUMÉ. Nous présentons dans cet article, la classification de 9880 phrases bruitées en utilisant les réseaux artificiels de neuronnes partiellement récurrents de Jordan et Elman. Nous proposons trois algorithmes de prétraitement tous basés sur la théorie de la prédiction linéaire pour l'extraction des indices fondamentaux dans ces phrases bruitées. Nous trouvons que l'approche des fenêtres consécutives introduite donne un bon résultat. La performance moyenne exprimée par un taux de reconnaissance de 99.5 % obtenu à la phase de test avec le réseau artificiel récurrent de Jordan permet de faire une meilleure généralisation. Tous les résultats obtenus avec les réseaux artificiels de Jordan et de Elman sont présentés et analysés.

ABSTRACT. We report on this paper the classification of 9880 noisy sentences using Jordan and Elman recurrent neural networks. We propose three preprocessing algorithms based on the linear predictor coding technique to extract features from noisy sentences. We find that the consecutive frames' approach introduced, gives a good result. The best average performance of 99.5 % over the testing dataset is obtained with Jordan network architecture, showing a good generalization behaviour. Results for both Elman and Jordan network architectures are given and analyzed.

MOTS-CLÉS : Réseaux artificiels de neuronne, algorithmes d'apprentissage, Traitement de la parole, Architecture de Elman, Architecture de Jordan, Extraction des indices.

KEYWORDS : Artificial neural networks, learning algorithms, Speech processing, Elman architecture, Jordan Architecture, Feature extraction.

1. Introduction

Speech enhancement in noisy environment is a challenge problem since decades. Noise is added to speech signal almost in an uncontrolled manner. Speech-processing systems (e.g. speech coding, speech recognition, speaker verification) pick-up those "unwanted" signal along speech. These noise signals result in performance degradation of those systems. Many efforts to design automatic systems for controlling noise are done (see [1] and [2]). Noise classification can be used to reduce the effect of environmental noises on speech processing tasks [3]. Vacher M. et al., proposed three algorithms for signal detection where the best result is achieved with the wavelet filtering algorithm [4]. Neural nets are proposed as alternative optimization techniques to handle problems in automatic speech recognition field. Prior a neural net maps each input feature vector into output vector, it must have first learnt the classes of feature vectors through a process that partitions the set of feature vectors. This is called discrimination (or classification) which involves machines learning. In [5], Looney defines classification as a process of grouping objects together into classes according to their perceived likenesses or similarities. In this paper, we are interested by the classification of noisy sentences into four classes using partially recurrent networks to demonstrate how generalization is not straightforward process namely with Jordan and Elman network models. Different preprocessing methods are used.

2. Database Description

We have built two types of databases. Firstly, we artificially corrupted 1376 sentences extracted from TIMIT database [6], and Italian speech sentences[1] with the same range of noise waves. The sampling frequency was 16 kHz at a rate of 16 bits per sample. The four noise sources used are babble, car, traffic, and white noise. Secondly, likewise the previous one, we artificially corrupted the 1376 sentences with random samples of the four noise sources. In both cases, the total of 5504 noisy sentences is splitted into three parts to define the training, the validation, and the testing sets. The training set is used to train the net. During the learning, the weights and biases are updated dynamically using the back propagation algorithm (see [7] for more details). The validation set is used to determine the performance of the net on patterns that are not trained during learning. Its major goal is to avoid the over training during the learning phase. The testing set is used to check the overall performance of the net. Table 1 shows the noisy sentences repartition for the three datasets.

3. Preprocessing Algorithms

Prior to evaluation with neural networks, the database requires preparation since this process significantly influences the network learning capabilities. In this paragraph, we present three approaches to preprocess the noisy sentences using *linear prediction coding* LPC algorithm. We found that a 12^{th} order model was enough good when the sampling

1. Italian sentences are recorded by us in a quiet room to avoid as much as possible the background noise with a mono-channel microphone.

Table 1. *Repartition of the training, validation and testing datasets.*

	Training dataset			Validation dataset			Testing dataset		
Sentences	Italian	English	Total	Italian	English	Total	Italian	English	Total
babble	8	494	510	4	460	464	4	398	402
car	8	494	510	4	460	464	4	398	402
traffic	8	494	510	4	460	464	4	398	402
white	8	494	510	4	460	464	4	398	402

frequency is 16 kHz. Let us review each of them starting from the linear prediction analysis.

3.1. Linear Prediction Analysis

One of the major development is speech coding is LP Coders that approximate the spectral envelop of speech with the spectrum of an all-pole model (see [8]). LPC is a particular technique of linear prediction analysis defined as a time-domain technique which attempts to predict a speech sample through a linear combination of several previous samples. This is given in [9] and [8] by

$$s_n = -\sum_{j=1}^{p} a_k s_{n-k} + G u_n$$

where s_n is the predictor signal, u_n some input, G is the gain factor, p the model order, and a_k the predictor coefficients. The coefficients are determined based on the minimum mean square error between the speech segment and the estimate of the speech (see [9]). As the order of LP model increases, more details of the power spectrum of speech can be approximated. The LPC algorithm finds the predictor coefficients, a_k of an pth order forward linear predictor and the gain factor G such that the sum of the square of the errors

$$e_n = s_n + \sum_{j=1}^{p} a_k s_{n-k}$$

is minimized. For the problem we faced, we set the model order to 12 after trial and error process.

3.2. Overlapped Windows' Approach

In this approach, the noisy sentence is segmented into 6 frames by applying a 200 ms window length every 100 ms. From each frame, 12 coefficients are extracted. The resulting coefficients are computed on each noisy sentence of 700 ms as duration and arranged into a single observation vector of 72 coefficients. Such a vector becomes an input to the net, used as classifier.

3.3. Consecutive Windows' Approach

In this approach, a noisy sentence is segmented into consecutive frames by applying a 200 ms as window's length. From each frame, 12 coefficients are extracted and arranged into a single vector of 72 coefficients. These coefficients are computed on noisy speech signal of 1.2 second as duration.

3.4. Mixture of Overlapped and Consecutive Windows' Approaches

This approach combines the two previous ones on a speech signal of 1.1 second and can be described as follows : the first 300 ms is segmented into frames by applying a 200 ms as window's length overlapped every 100 ms following by the overlapped windows' technique. That leads to 24 coefficients. The remaining noisy sentence is segmented into four consecutive windows of 200 ms length according to the consecutive windows approach. That leads to 48 coefficients. Both sets of coefficients are concatenated into a single vector of 72 coefficients.

4. Recurrent Neural Networks

Recurrent networks are logical candidates when identifying a nonlinear dynamical process (see [10] and [11]) like the problem we faced. Such nets are attractive with their capabilities to perform highly nonlinear dynamic mapping (see [12]) and their ability to store information for later use. Moreover, they can deal with time-varying input or output through their own natural temporal operation. There are two types of recurrent neural networks : *fully recurrent neural networks and partially recurrent neural networks*. Many learning algorithms have been developed for both models (see [13], [14] and [15]). We refer the reader to [12] and [17] for more details about fully recurrent networks since it is out of scope of this paper. Partially recurrent networks are back-propagation networks with proper feedback links. They allow the network to remember cues from the recent past. In these architectures, the nodes receiving feedback signals are context units (see [12]). According to the kind of feedback links, two major models of partially recurrent networks are encountered in literature as presented below.

4.1. Jordan Sequential Network

This network model is realized in adding recurrent links from the network's output to a set of context units C_i, of a context layer and from the context units to themselves. Context units copy the activations of output node from the previous time step through the feedback links with unit weights. Their activations are governed by the differential equation

$$C_i'(t) = -\alpha C_i(t) + y_i(t)$$

where the y_i's are the activations of the output nodes and α is the strength of the self-connections. Despite the use of the Jordan sequential network to recognize and distinguish different input sequences with sequences of increasing length, this model of network encounters difficulties in discriminating on the basis of the first cues presented.

4.2. Elman Network

Elman proposed a simple but powerful two layers back propagation network where time is implicitly represented by the network dynamics. According to Elman (see [14]), increasing the sequential dependencies in a given task does not necessarily result in worse performance. Feedback connections come from the output of the hidden layer to input layer.

4.3. Network Architecture Design

Both Jordan and Elman recurrent networks have the following architecture :

– the input layer is a vector of 72 components ;

– a hidden layer with 20 neurons ;

– four output nodes ; one for each class ;

– number of epochs is fixed to 2000 ;

– learning rate is fixed to 0.002.

5. Experimental Results

We carried out four experiments using both Jordan and Elman models on a workstation machine equipped of the Stuttgart Neural Network Simulator [18].

– The first experiment deals with a database of sentences corrupted with the same range of noise waves. The feature extraction technique is based on the mixture approach. The results obtained are presented in Table 2.

– The second one is carried out with preprocessed data obtained thank to the overlapped windows' approach. The results obtained are also presented in Table 2.

– The third one is done with noisy sentences preprocessed with the consecutive windows' algorithm. The results are presented in Table 3.

– Finally, the fourth one is done with noisy speech database obtained from the mixture approach. The results are also presented in Table 3.

Table 2. *Result obtained from the first and second experiments.*

Methods	First experiment						Second experiment					
Network type	Jordan			Elman			Jordan			Elman		
Classes	Tra	Val	Tst	Tra	Val	Tst	Tra	Val	Tst	Tra	Val	Tst
babble	98.2	94.0	94.5	99.8	99.6	100	96.1	93.1	92.0	90.4	84.3	84.1
car	99.8	97.0	97.3	100	99.6	99.5	99.0	95.9	96.0	97.3	92.0	93.5
traffic	100	98.5	97.0	100	100	100	99.8	99.6	98.5	98.2	98.1	95.8
white	99.8	98.9	97.3	100	100	100	99.4	99.4	98.0	99.2	99.4	97.8
average	99.6	97.1	97.2	99.9	99.8	99.7	98.6	97.0	96.1	96.3	93.5	92.8

Table 3. *Result obtained from the third and fourth experiments.*

Methods	Third experiment						Fourth experiment					
Network type	Jordan			Elman			Jordan			Elman		
Classes	Tra	Val	Tst	Tra	Val	Tst	Tra	Val	Tst	Tra	Val	Tst
babble	96.9	92.9	93.5	97.3	90.9	92.5	97.8	94.0	94.0	95.9	91.8	91.5
car	99.2	97.6	96.3	99.0	97.2	96.5	99.2	96.1	96.3	98.8	96.6	96.0
traffic	99.6	99.1	97.8	99.2	98.9	97.3	100	98.5	97.8	100	98.3	96.5
white	99.6	99.8	99.5	99.6	99.8	99.5	99.8	99.4	98.5	99.6	98.9	97.8
average	98.8	97.4	96.8	98.8	96.7	96.5	99.2	97.0	96.7	98.6	96.4	95.5

5.1. Results Analyses

Taking a careful look on the different preprocessing techniques proposed, one can say that a 12^{th} order model is enough good for coding the noisy sentences for the discrimination task under examination. Both Elman and Jordan models are good classifiers to discriminate the noisy speech sentences. Moreover, one can do the following observations :

– From Table 2, it appears that Elman network performs substantially better than Jordan network. Both networks have the ability to generalize. One can explain such an important result due to the same range of noise waves used to corrupt each speech signal sentence. The average percentages on correct classification (see Table 2) can be justify by the use of the same samples of noise waves of each type of noise. Even though in practical thinking, noise waves that corrupted speech signal changed rapidly, this experiment showed us how much partially recurrent neural nets have the ability to generalize. For the theory about the use of probabilistic techniques for the modeling of generalization, see [16].

– From Table 2, it appears varying noise wave range induces the network to misclassify some patterns.

– From Table 3, it appears that the use of consecutive frames concept is good for the classification task under examination.

– From Table 3, it appears that the use of the mixture preprocessing approach leads to substantially improvement as expected.

6. Concluding Remarks

Through this paper, we saw that the *Linear Prediction Coding* did a good job for discriminating noisy sentences. It also appears that Jordan model performs as good as Elman network for the task under examination. Also, the concept of consecutive frames we have introduced is a good one. Moreover, the mixture approach introduced as expected has a good performance since it combines two approaches even though the improvement compared to the one obtained with consecutive window's approach is less.

7. Bibliographie

[1] C. AVENDANO, « Temporal Processing of Speech in a Time Feature Space », *PhD Dissertation*, Oregon Graduate Institute of Science and Technology, 1997.

[2] B. WIDROW et al., « Adaptive Noise Canceling : Principles and Applications », *in Proceedings of* IEEE, vol. 63,n° 12, December, pp. 1692 – 1712, 1975.

[3] EL-M. KHALED et al., « Frame-Level Noise Classification in Mobile Environments », *in Proceedings of* the IEEE Conference on Acoustics, speech, Signal Processing, Phoenix, AZ, pp. 237–240, March 1999.

[4] M. VACHER et al., « Life Sounds Extraction and Classification in Noisy Environment », *in Proceedings of* the International Association of Science and Technology for Development, Signal and Image Processing IASTED'SIP 2003, Horiolulu, Hawaii, USA, 13–15 August, 2003.

[5] C.G. LOONEY, « Pattern Recognition Using Neural Networks, Theory and Algorithms for Engineers and Scientists, », *in* Oxford University Press, Inc., 1997.

[6] S. ZUE , T. JACKSON, « Speech Database Development, TIMIT and beyond », *in the Proceedings of* , Speech Communication, pp. 351 – 356, 1990.

[7] R. BEALE , T. JACKSON, « Neural Computing : An Introduction », Institute of Physics Publishing, Bristol and Philadelphia, IOP Publihing, Ltd, 1992.

[8] J. MAKHOUL, « Linear Prediction : A Tutorial Review », *in Proceedings of the* IEEE vol. 63, n° 4, pp. 561 – 579, 1975. Oxford University Press, Inc., 1997.

[9] H. HERMANSKY, « Analysis in Automatic Recognition of Speech », *in* Speech Proceessing, Recognition and Artificial Neural Networks, Chollet et al. editors, pp. 115 – 137, 1998.

[10] O. NERRAND, « Training Recurrent Neural Networks : Why and How ? An Illustration in Dynamical Process Modeling », *in the Proceedings of* the IEEE Transactions on Neural Networks, vol. 5, pp. 178 – 184, 1994.

[11] R.J. WILLIAMS , D. ZIPER, « Technical Report NU-CCS-90-9 », Boston, North-Eastern University, College of Computer Science, April 12, 1990.

[12] C.-T. LIN , C.G.S. LEE, « Neural Fuzzy Systems, A Neuro-Fuzzy Synergism to Intelligent Systems », Prentice Hall, Upper Saddle River, NJ, 1996.

[13] R. J. WILLIAMS, « A Learning Algorithm for Continually Running Fully Recurrent Neural Networks », *in the Proceedings of* Neural Computation, vol. 1, pp. 270 – 280, 1989.

[14] J.L. ELMAN, « Finding Structure in Time », *in the Proceedings of* Cognitive Science, vol. 14, pp. 179 – 211, 1990.

[15] M. JORDAN, « Attractor Dynamics ad Parallelism in a Connectionnist Sequential Machine », *in Proceedings of* the Eigth Annual Conference of the Cognitive Science Society, pp. 531 – 546, 1986.

[16] M. ANTHONY « Mathematical Modeling of Generalization », *Proceedings of Journal*, n° 13, Italian Workshop on Neural Nets, WIRN Vietri sul Mare, Italy,pp. 182–189, 2002.

[17] M.C. BISHOP, « Neural Networks for Pattern Recognition », Oxford University Press, Inc., New York, (1995)

[18] A. ZELL et al., « Stuttgart Neural Network Simulator », User manual, 1995.