Novel Applications of Complexity Inspired RDT Transform for Low Complexity Embbeded Speech Recognition in Automotive Envirnoments.

Mihai Bucurica
Doctoral School in Electronics
Telecommunications and Information Technology,
University "Politehnica" of Bucharest, Romania
mihai-ionut.bucurica@sdettib.pub.ro

Ioana Dogaru, Radu Dogaru
Natural Computing Laboratory,
Dept. of Applied Electronics and Information Eng.
University "Politehnica" of Bucharest, Romania
radu_d@ieee.org, ioana.dogaru@upb.ro

***Abstract*— Embedded dictation, i.e. recognizing vocal commands in noisy environments, with good accuracy and using low complexity implementations is a desirable task with many applications. Such applications include automotive infotainment solutions particularly when no connectivity is available, personal assistants including embedded dictation solutions for disabled people, and so on. This paper reports our novel results in applying a nonlinear transform (RD-transform) introduced in a previous work and inspired form complexity measurements of signals generated in cellular automaton. Such a transform is compact and has a low computational complexity yet it previously proved quite efficient in terms of accuracy for a standard task of recognizing user independent dictation of digits. Herein, we report results on employing RD-transform on a specially designed sound database containing commands for the non-critical automotive equipment in a realistic, noisy environment. In addition to specific nonlinear transforms, low complexity FSVC classifiers were employed proving that good accuracies can be achieved using a very convenient implementation solution.**

Keywords— speech processing; nonlinear signal processing; radial basis neural networks; complex nonlinear networks; support vector classifiers.

I. INTRODUCTION

In many circumstances, the use of simple and compact electronic devices capable to recognize vocal commands, are highly desired. Such circumstances include automotive dictation systems [1], disabled people assistants [2] among others. Herein the focus would be on an automotive application using a proprietary database with 9 particular commands uttered inside the vehicle with specific ambient noise. The usual approach in speech recognition and dictation is often based on implementing relatively complex speech recognition systems running on a remote server which may be accessed from a remote client with internet access. However, in the case of limited or no connectivity access such a system become useless. Therefore, in such circumstances embedded dictation systems that can be operated independently on any wireless connection, are needed. This paper focuses on such kind of solution where the aim is to provide a low complexity, yet efficient in terms of recognition accuracy, solution. While many similar isolated word recognition systems reported in the literature employ Mel frequency cepstral coefficients (MFCC) analysis combined with HMM or, more recently, using deep learning approaches [3], recurrent neural networks [4][5], liquid-state machines [6] still the implementation complexity remains high, therefore we employ a method based on an original approach first reported in [7]. The key ingredient of our approach is to consider a simple to implement (low complexity, thus convenient for embedded systems) nonlinear transform, herein denoted as RDT standing for Reaction Diffusion Transform combined with a low complexity classifier described in [8]. A remainder of the transform and its properties is given in Section II, it basically transforms a signal sequence of size 2^n into a compressed "spectrogram" with only *n-3* components. When applied to isolated speech recognition problems, the entire utterance representing a speech command is split into *M* sequences the average RDT spectrum is computed for each of the *M* sequences resulting in a feature vector with $M(n-3)$ components. Details regarding the construction of the feature vector are provided in Section III. Such annotated feature vectors provide training and test databases for a special radial basis neural network classifier optimized to provide an efficient, low complexity, implementation while providing a very good accuracy. The very fast training allows accurate optimization of its radius parameter to optimize accuracy while synapses in the output layer are simply 0 or 1 thus leading to a very convenient implementation. A detailed description of the classifier is given in Section IV, it representing the unsupervised version of [8], herein called FSVC-NT. The experimental setup including the construction of a proprietary database (since no other similar was found available on the public domain) and the best results achieved with our solution (average accuracy of up to 89% for 9 vocal commands with 100% accuracy for some of the commands) are reported in Section V. Concluding remarks and research perspectives are given in Section VI.

II. REACTION-DIFFUSION TRANSFORMS

A. Definition of the reaction-diffusion transform

The basic idea to define the RDT comes from the definition of a clustering coefficient *C* ranging in the domain [0,1] from a signal associated with the distribution of states in a 1-

dimensional cellular automata [9]. Let us consider such a sequence $\{s(t)\}_{t=0,..w-1}$ forming a "signal frame" with w samples (in cellular automata framework, w were adjacent cells). In the following is assumed that w is a power of 2, with $n = \log_2(w)$. In the above it is assumed that signal samples are bounded i.e. $s_t \in [-1,1]$. The clustering coefficient is computed as:

$$C = S(0) = \frac{1}{w}\sum_{t=2}^{w-1}\frac{1}{4}\left|s(t-1)+s(t+1)-2s(t)\right| \qquad (1)$$

where *S(0)* denotes the "highest frequency" component of the RDT transform, obtained when all samples of the signals within the w sized frame were considered. Lower frequency "spectral values" *S(k)* are computed similarly but on decimated sequences i.e. applying the same above formulae on frames formed of $w/(2^k)$ samples, with $k = 1,..(n-4)$. A scaled version of the RDT, denoted SRDT (Scaled RDT) is computed as above but (1) is now replaced with equation (2) below. The scaled version proved to improve the recognition accuracy while being less sensitive to the amplitude of spoken utterance:

$$C = S(0) = \frac{1}{w}\sum_{t=2}^{w-1}\frac{1}{4}\frac{\left|s(t-1)+s(t+1)-2s(t)\right|}{\left|s(t-1)\right|+\left|s(t+1)\right|+\left|2s(t)\right|} \qquad (2)$$

In [7] it was shown that for a simple isolated speech recognition task (digit recognition from multiple speakers) SRDT with optimized parameters followed by SVM classifier achieved recognition accuracy close to 88%. As noted, the computational complexity of computing RDT transform is low, no multiplication is needed but only summations and absolute value computations. Computing RDT for a window size w requires *O(w)* computations.

B. *Examples of applying RDT*

In Fig. 1 a raw, not segmented signal (utterance saying "avarie" (damage in Romanian language) from our database is considered (sampling rate 44Khz) and the corresponding RDT and SRDT are presented bellow for window sizes *w=2048*.

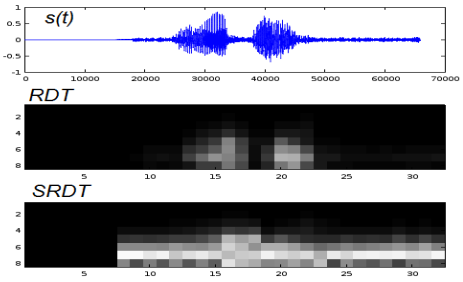

Fig. 1. Spectograms obtained from an utterance using RDT and SRDT applied to sound frames of size *w=2048* composing the signal.

Note that in the case of SRDT (less sensitive to sound amplitude) even the noisy regions present bold spectrograms. In a practical isolated sound application, SRDT or RDT is applied for a segmented part of the utterance containing the relevant sound. In this work an "economical" segmentation is applied i.e. from a sequence of sound like the one in Fig.1 only the samples satisfying $|s(t)| > \varepsilon$ are preserved. The value $\varepsilon = 0.01$ was experimentally found to produce a relatively good segmentation for the recorded signals in our database. This proposed segmentation scheme can be easily applied and although it removes some samples from the main part of the utterance, the remaining signal remains perfectly intelligible and thus is expected that an automated recognition system would correctly classify the utterance. Fig. 2 shows the result of applying the above mentioned scheme to the signal in Fig. 1.

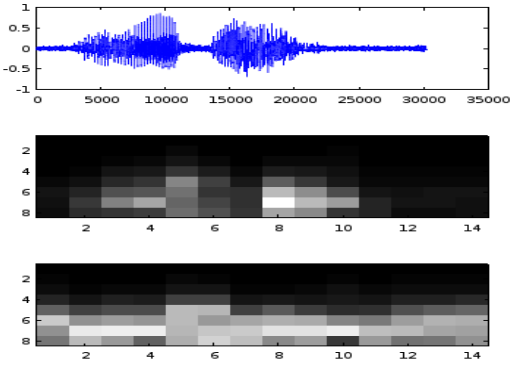

Fig. 2. Segmented sound signal using the threshold $\varepsilon = 0.01$. From top to bottom: siganal, RDT spectograms, SRDT spectograms.

III. Constructing Feature Vectors for the Embedded Dictation Problem

In any recognition system the raw signal must be transformed into a fixed size annotated feature vector in order to be learned by a classifier. Since segmented raw signals have a variable number of samples, it is not possible to define the feature vector simply as a collection of their RDT spectrograms. Therefore, a specific scheme of constructing a feature vector must be specified, as follows:

The algorithm considers M as a predefined number of segments in which the segmented voice signal is split. Let us assume *M=3*. For the segmented signal in Fig.2 composed of 14 spectrograms there should be 5=round (14/3) spectrograms per segment for the first *M-1* segments and the remaining 4 are left to represent the last segment. For each segment an average spectrogram is computed thus resulting in a feature vector formed of $M(n-3)$ average spectrograms. The above scheme would produce the same size of the feature vector for any variable number of spectrograms associated to different lengths of signals. Consequently, for any given isolated signal to be recognized using the RDT method, there are two parameters to be optimized in order to achieve maximal accuracy: the number of segments M and the size w of the signal window (or equivalently $n = \log_2(w)$) submitted to the RDT or SRDT.

IV. Fast Support Vector Classifiers

The architecture of FSVC was first introduced in [10] as a convenient to implement in hardware (but in software as well) alternative to Support Vector Machine (SVM). It is essentially a radial basis function neural network differing from the traditional architecture in several original features: i) there is no

training needed for centroids of the RBF neurons in the hidden layer, instead they are simply selected among the training samples, using a novelty detection algorithm; ii) the training of the system eventually reduces to training the Adaline (linear neurons) in the output layer. As shown in a recent paper [8] using a proper initialization of the output layer there is no need for training the Adaline thus resulting in a very fast training of the classifier, with at least one order of magnitude when compared to extreme learning machine (ELM) considered today one of the fastest to train neural networks. In terms o performance, FSVC exhibits similar accuracy to SVM or ELM on a wide range of problems as shown in [11][12]

Herein we use the "no-tune" unsupervised version of the SFSVC[1] exposed in [8], allowing a very fast training speed and thus a relatively fast optimization of the *r* (radius) and *prag* (threshold) parameters. A brief description of the algorithm follows: The feature vectors computed as indicated in Section III form a training and a test set. **The training set** $TR = \{(\mathbf{x}_k, \mathbf{d}_k)\}$ where $k = 1,..N$ is the sample index, *N* is the number of samples, $\mathbf{x}_k$ is an input (feature) vector, each with a dimension $nf = M(n-3)$. The label of the sample is expressed using a *Q*-dimensional vector $\mathbf{d}_k$ formed of 0 valued elements except $d_{j,k} = 1$ indicating that the sample *k* belongs to class "j" among all possible *Q* classes. In a "no-tune" FSVC structure $d_{j,k} \in \{0,1\}$ are also the fixed-valued synapses of the output Adaline layer. **The test set** *TS* has a similar structure yet contains different *Ns* samples and it is used only to evaluate the generalization performance of the classifier. For the problem considered herein there is a balanced distribution of samples with 50% in the train and the remaining 50% in the test set. The average accuracy (*Acc.*) is considered herein as a performance measure. It represents the fraction of correctly assigned samples in the test set, by the FSVC classifier.

RBF units and radius: It is assumed that each hidden neuron unit is defined by a RBF function where a distance between the actual input sample $\mathbf{u} = (u_1,..u_i,..u_{nf})$ and the corresponding support vector (centroid) $\mathbf{c} = (c_1,..c_i,..c_{nf})$ can be computed in any desired way. In this work we discuss of ***dtype=1*** distance in the case of Hamming distance $d = \sum_{i=1}^{nf} |u_i - c_i|$ and ***dtype=2*** in the case of the Euclidean one. Any other distance formula can be considered. As for the RBF functions, there is a wide palette of possibilities and no restriction, herein we consider only ***rbftype=1*** for a simple (hardware-oriented) triangular function defined as: $RBF(d,r) = \max(0, 1 - 0.4d/r)$ and ***rbftype=2*** for the classic (but not so convenient for hardware-oriented) Gaussian kernel $RBF(d,r) = \exp(-d^2/2r^2)$. In all cases, the radius *r* is an important parameter and is basically the main parameter that has to be optimized to ensure best generalization performance. As seen later, a finer tuning may be considered in using the *ov* (overlap factor) which is usually taken $ov = 1$.

The fast training of the FSVC-NT (no-tune FSVC) consists in a single epoch for browsing all samples in the training set *TR* and constructing a hidden layer by specifying the centroids for the RBF layer as specific feature vectors (inputs) from the training set (also called support vectors). The result of the training process is an **index table** $TIX = \{i_1, i_2, ..i_p, ..i_m\}$ storing integer values to locate the selected support vector among the feature vectors in the *TR*. Consequently, the *p* RBF-neuron of the hidden layer has the support vector $\mathbf{c}_p = \mathbf{x}_{i_p}$ as centroid. The number of hidden neurons *m* is always smaller than the number of samples *N* in the training set. A novelty mechanism is used to select if the current sample from the train set is selected as a support vector or not (as detailed in [8]). Essentially it is selected only if the activity of the hidden layer is smaller than the overlapping threshold *ov*. Using the test set to evaluate accuracy, a certain number of training cycles using various *dtype*, *rbftype*, radius *r*, and *ov* are executed until the best accuracy is obtained for a given dataset. In order to asses the accuracy on the test set the following **SFSVC Prediction algorithm** is used:

1. FOR *j=1,..Q sc(j)=0*; END // initialize output scores
2. FOR *p=1 .. m*
3. *k = TIX(p)=* i_p *// locate the center in TR*
4. $d = dist(\mathbf{u}, \mathbf{x}_k)$
5. *z = RBF(d,r) // calculate the output of the hidden layer*
6. FOR *j=1,..Q* // calculate the output scores
7. IF $d_{j,k} \neq 0$ *sc(j) = sc(j)+z* ; END
8. END
9. END
10. Predicted_class= Arg(max(sc))

Misclassified samples are counted and the accuracy is reported as the ratio between all correctly classified samples and all samples in the test set.

V. EXERIMENTAL SETUP AND RESULTS

A number of 9 vocal commands were considered and for each of them a number of about 20 different utterances were recorded (44 Khz sampling rate) in the specific noise environment of the automobile. The list of commands and their associated labels is presented below using the format {[label number, utterance (Romanian), English translation, number of samples],…}: {[1, avarii, damage, 21],[2, claxon, horn, 22],[3, frana, brake, 20],[4, inainte, before, 22],[5, inapoi, back, 17],[6, lumina, light, 22],[7, radio, radio, 21],[8 start, start, 22],[9, stop, stop, 24]. There are at all 191 samples, each of the utterances being transformed into feature vectors using the RDT and the SRDT methods with their two parameters to be optimized (*M* – the number of segments, ranging from 4 to 6; *w* – the size of the window with the corresponding *n* ranging from 9 to 11; *r* – radius). Other elements changed during the optimization process are the types of distances and radius, with a preference for ***dtype=1*** and ***rbftype=1*** only for specialized hardware implementations where they have lower complexity than other choices. Table I gives a synthesis of the main results

[1] A Matlab implementation of the FSVC is given at http://www.mathworks.com/matlabcentral/fileexchange/49695-fast-support-vector-classifier--a-low-complexityalternative-to-svm-/ with a GitHub link to a much faster implementation using MEX files.

for the case of RDT while Table II gives the results for SRDT. The cases giving the best accuracies are emphasized. In all cases, the threshold $ov = 0.25$

TABLE I: OPTIMAL STRUCTURES AND PERFORMANCE FOR RDT-BASED CLASSIFICATION USING NT-FSVC

Experiment	**a**	**B**	**c**	**d**	**e**	**f**
M	5	5	6	6	4	6
w (n)	2048 (11)	1024 (10)	1024 (10)	1024 (10)	512 (9)	1024
D-type	2	1	1	2	1	1
RBF-type	2	2	2	2	2	1
r	0.3	1.54	1.63	0.3	1.25	2
Acc. %	**73.9**	**81.2**	**85.4**	**84.3**	**77**	68.7

As seen in the above table, an optimal performance (85.4% recognition accuracy) is achieved using Gaussian radial basis function and Manhattan distance for a proper number of segments (M=6). Last column indicates the best result obtained when using the most convenient RBF function and distance, unfortunately in such cases the recognition accuracy is significantly reduced. In [7] it was found that using properly segmented (in the sense of isolating the utterance from silence) signals, better performance was achieved. When SRDT was used for this automotive dictation database, a better accuracy was observed as well, as reported in Table II.

TABLE II: OPTIMAL STRUCTURES AND PERFORMANCE FOR SRDT-BASED CLASSIFICATION USING NT-FSVC

Experiment	**a**	**b**	**c**	**d**	**e**
M	4	5	6	4	4
w (n)	1024 (10)	1024 (10)	1024 (10)	512 (9)	512 (9)
D-type	1	1	1	1	1
RBF-type	2	2	2	2	1
R	0.23	1.82	1.9	1.3	1.96
Acc. %	**82.2**	**88.5**	**81.2**	**84.3**	**70**

As indicated in the above table, a slightly increased accuracy (88.5%) is now obtained with a proper optimization of the M parameter. Again, the best choice for distance and basis function is Manhattan distance with Gaussian basis function. When triangular basis function is used (column "e") the accuracy decreases. The confusion matrix for the experiment "b" in Table II (best result) displayed in Fig. 3 indicates that samples from some classes are 100% correctly recognized while some utterances (last two commands) have a low recognition rate.

		predicted									Accuracies (per class)
		1	2	3	4	5	6	7	8	9	
actual	1	10	0	0	0	0	0	0	0	0	100 %
	2	0	11	0	0	0	0	0	0	0	100 %
	3	0	0	11	0	0	1	0	0	0	91.6 %
	4	0	0	0	11	0	1	0	0	0	91.6 %
	5	0	0	0	0	6	0	0	0	1	85.7 %
	6	0	0	1	0	0	9	0	0	0	90 %
	7	0	0	0	0	1	0	10	0	0	90.9 %
	8	1	0	0	0	0	0	0	8	2	72%
	9	0	0	0	0	0	0	0	3	9	75%

Fig. 3. Confusion matrix on the test set for the best optimized SRDT+FSVC dictation system. Note that only two classes of commands (last two) give accuracies lower than 85.7%.

VI. CONCLUSIONS

A low complexity system for embedded dictation is proposed, based on a simple to compute RDT transform inspired from computing complexity indices in cellular automata. The name "reaction-diffusion' stands from the 1-dimensional Laplacian used in reaction-diffusion cellular nonlinear networks inspiring this transform. The implementation complexity of RDT is much smaller than needed for MFCC, which is often used is such applications. Accuracies obtained from a proprietary database with utterances representing verbal commands with specific automobile environment background noise are rather good, up to 100% for some of the particular commands, thus comparable to results obtained by other approaches [1] for similar tasks. Further research will focus on improving the segmentation scheme to increase accuracy.

REFERENCES

[1] P. Ding, L. He, X. Yan, R. Zhao and J. Hao, "Robust mandarin speech recognition in car environments for embedded navigation system," in IEEE Transactions on Consumer Electronics, vol. 54, no. 2, pp. 584-590, May 2008.

[2] Hawley, M., Enderby, P., Green, P., Cunningham, S., Palmer, R.: Development of a Voice-Input Voice-Output Communication Aid (VIVOCA) for People with Severe Dysarthria. in: Craddock, G.M., McCormack, L.P., Reilly, R.B., Knops, H. (eds.) Assistive Technology – Shaping the Future, pp. 882–885. IOS Press, Amsterdam (2003).

[3] G Salvi, "An Analysis of Shallow and Deep Representations of Speech Based on Unsupervised Classification of Isolated Words", Volume 48 of the series Smart Innovation, Systems and Technologies pp 151-157, 2016.

[4] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," Neural Networks, vol. 18, no. 5, pp. 602–610, 2005.

[5] G. Evangelopoulos, "Efficient Hardware Mapping of Long Short-Term Memory Neural Networks for Automatic Speech Recognition", MSc. Thesis, KU Leuven, 2016.

[6] Y. Jin, P. Li, "Performance and robustness of bio-inspired digital liquid state machines: A case study of speech recognition", Neurocomputing, Vol. 226, 2017, pp. 145–160.

[7] R. Dogaru, "Fast and Efficient Speech Signal Classification with a Novel Nonlinear Transform," 2007 International Symposium on Information Technology Convergence (ISITC 2007), Joenju, 2007, pp. 43-47.

[8] R. Dogaru and I. Dogaru, "A super fast vector support classifier using novelty detection and no synaptic tuning," 2016 International Conference on Communications (COMM), Bucharest, 2016, pp. 373-376.

[9] R. Dogaru, Systematic design for emergence in cellular nonlinear networks – with applications in natural computing and signal processing, Springer-Verlag, Berlin Heidelberg, 2008.

[10] R. Dogaru, A.T. Murgan, S. Ortmann, M. Glesner, "A modified RBF neural network for efficient current-mode VLSI implementation", in *Proceedings of the Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems (Micro-Neuro'96),* IEEE Computer-Press, Laussane 12-14 Febr. 1996, pp. 265-270, 1996.

[11] M. Bucurica and R. Dogaru, "A comparison between Extreme Learning Machine and Fast Support Vector Classifier," 2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, 2015, pp. Y-9-Y-12.

[12] M. Enache and R. Dogaru, "A benchmark study regarding Extreme Learning Machine, modified versions of Na??ve Bayes Classifier and Fast Support Vector Classifier," 2015 E-Health and Bioengineering Conference (EHB), Iasi, 2015, pp. 1-4.