Kohonen network, automatic disorderds speech recognition, waveblaster, cwt, continuous wavelet transform, bark scale, sound repetition

Ireneusz CODELLO, Wiesława KUNISZYK-JÓŹKOWIAK, Elżbieta SMOŁKA, Adam KOBUS

DISORDERED SOUND REPETITION RECOGNITION IN CONTINUOUS SPEECH USING CWT AND KOHONEN NETWORK

Automatic disorders recognition in speech can be very helpful for therapist while monitoring therapy progress of patients with disordered speech. This article is focused on sound repetitions. The signal is analyzed using Continuous Wavelet Transform with 16 bark scales, the result is divided into vectors and passed into Kohonen network. Finally, the Kohonen winning neuron result is put on the 3-layer perceptron. The recognition ratio was increased by about 20% by adding a modification into the Kohonen network training process as well as into CWT computation algorithm. All the analysis was performed and the results were obtained using the authors' program "WaveBlaster". The problem presented in this article is a part of our research work aimed at creating an automatic disordered speech recognition system.

1. INTRODUCTION

Speech recognition is a highly important branch of informatics nowadays – oral communication with a computer can be helpful in real-time document writing, language translation or simply in using a computer. Therefore the issue has been analyzed for many years by the researches which resulted in creating many algorithms, such as Fourier transform, Linear Prediction, spectral analysis. Disorders recognition in speech is quite a similar issue – one attempt to find where speech is not fluent instead of trying to understand the speech, therefore the same algorithms can be used. Automatically generated statistics of disorders can be used as a support for therapists in their attempts at estimating therapy progress.

Several methods for disordered speech detection have been used by researches for disordered speech recognition, like: Fourier Transform, third octave filters, fuzzy logic [15], Hidden Markov Models, MFCC coefficients [19], Linear Prediction [20] or Kohonen networks [13]. In this paper a relatively new algorithm is used – Continuous Wavelet Transform (CWT) [1,2,11] as - by using it - the most suitable scales (frequencies) can be chosen. Fourier transform and Linear Prediction [6] are not so flexible – we have to chose if we want to have more precise time scale (small window) and more precise frequencies or the opposite - for the whole spectrogram. In CWT such a decision can be made for each scale separately. The bark scales set was taken which is, besides the Mel scales and the ERB scales, considered as a perceptually based approach [12]. The CWT result is divided into fixed-length windows and each one is converted into a vector. The vectors, using another bigger window, are grouped and marked if the group starts with a sound repetition or not and then passed onto the Kohonen network which receives the 3D data and produces the 2D data. Such a dimensionally reduced signal is passed to a 3-layer perceptron.

After creating recognition statistics a few algorithm improvements were added which significantly increased the recognition ratio, especially the created modification of Kohonen training algorithm (see section 3.2).

2. INPUT SIGNAL PROCESSING BY CWT

2.1. MOTHER WAVELET

Mother wavelet is the heart of the Continuous Wavelet Transform:

$$CWT_{a,b} = \sum_t x(t) \cdot \psi_{a,b}(t) \quad \text{, where} \quad \psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \tag{1}$$

where $x(t)$ – input signal, $\psi_{a,b}(t)$ – wavelet family, $\psi(t)$ – mother wavelet, a – scale (multiplicity of mother wavelet), b – offset in time. The Morlet wavelet represented by the equation [7] was used:

$$\psi(t) = e^{-t^2/2} \cdot \cos(2\pi \cdot 20 \cdot t) \tag{2}$$

which has center frequency F_C=20Hz. Mother wavelets have one significant feature: length of the wavelet is connected with F_C which is a restraint. Morlet wavelet is different because the length can be chosen and then its F_C can be set by changing the cosines argument.

2.2. SCALES

For frequencies of scales, a perceptually based approach was assumed – because it is considered to be the closest to the human way of hearing. Hartmut scales were chosen [18]:

$$B = \frac{26.81}{1+1960/f} - 0.53\,, \; f - \text{freq. in Hz} \tag{3}$$

The frequency F_a of each wavelet scale a was computed from the equation

$$F_a = F_C F_S / a\,, \qquad F_S - \text{sampling frequency} \tag{4}$$

Due to the discrete nature of the algorithm, it was not always possible to match the scale a with the scale B perfectly (Table 1). During the research some Hartmut scales were found as insignificant in the recognition process. Therefore eventually only 16 scales were used.

Table 1. 16 scales a with corresponding frequencies f and bark scales B.

***a* [scale]**	***f* [Hz]**	***B* [bark]**
57	7736	20,9
68	6485	20,1
83	5313	19,1
100	4410	18
119	3705	17
140	3150	16
163	2705	15
190	2321	14

***a* [scale]**	***f* [Hz]**	***B* [bark]**
220	2004	13
256	1722	12
297	1484	11
347	1270	10
408	1080	9
479	920	8
572	770	7
700	630	6

2.3. SMOOTHING SCALES

Because CWT values are similarity coefficients between signal and wavelet (the sign of its value) is – therefore - irrelevant in all computations the following modules are taken – $|CWT_{a,b}|$. We went one step further and the $|CWT_{a,b}|$ was smoothed by creating a contour (see Figure 1) because of its good recognition ratio influence [4].

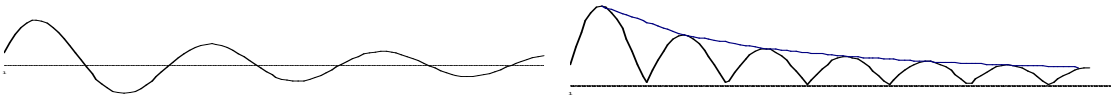

Fig. 1. Left: Cross-section of one CWTa,b scale. Right: Cross-section of one |CWTa,b| scale and its contour (smoothed version).

2.4. WINDOWING

Thus, the spectrogram consists of 16 smoothed bark scales vectors. Then the spectrogram is cut into 23.2ms frames (512 samples when F_S=22050Hz), with a 100% frame offset. Because each scale has its own offset – one window of fixed width (e.g. 512 samples) will contain different number of CWT values (CWT similarity coefficients) in each scale (see Figure 3), therefore the CWT arithmetic mean of each scale value is taken.

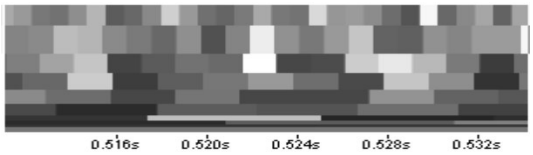

Fig. 2. One CWT window (512 samples when F_S=*22050Hz*).

From one *i*-th window the vector *V* of the form presented in eq. 5 is obtained. Such consecutive vectors are then passed into the Kohonen network.

$$\vec{V} = \left\{mean\left(\left|CWT_{57,i}\right|\right), mean\left(\left|CWT_{68,i}\right|\right), ..., mean\left(\left|CWT_{572,i}\right|\right), mean\left(\left|CWT_{700,i}\right|\right)\right\} \tag{5}$$

3. MODIFIED KOHONEN NETWORK ALGORITHM

The Kohonen network [5,8,9,10,16,17] (or "self-organizing map" or SOM, for short) was developed by Teuvo Kohonen. The basic idea behind the Kohonen network is to establish a structure of interconnected processing units ("neurons") which compete for the signal. While the structure of the map may be quite arbitrary rectangular maps were used in the research.

Let's assume that:

- Kohonen network has *K* neurons,
- *n* is the dimension of each input vector X,
- each element $x_i \in X$ is connected to all *K* neurons, so we have $K \times n$ connections. Each connection is represented by its weight w_{ij}, $i=1..n$, $j=1..K$ which is adjusted during the training.

The Kohonen neurons were numbered by rows from the top to the bottom

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14

For every 2D CWT vector (see eq. 5) one winning neuron is obtained. Therefore the Kohonen network is used to convert 3-dimention CWT spectrogram (which consists of 2D CWT vectors laying one next to another) into 2-dimention winning neuron contour as depicted on Fig. 3 [13,14]. This reduction of data from 3D into 2D which is later passed on to MLP occurred to have positive impact on non-fluencies recognition ratio [13,14] (the whole, 3D spectrogram seems to be too large for MLP to find general features).

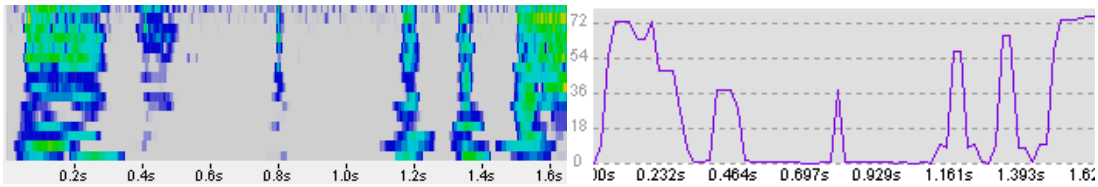

Fig. 3. Converting 3D CWT (Left picture. *Y* axis: the bark scale, *X* axis: the time) into 2D Kohonen winning neuron contour (Right picture. *Y* axis: winning neuron, *X* axis: the time). In this example Kohonen network was of the size 8x9 giving 72 neurons.

3.1. BASIC ALGORITHM

The basic training algorithm is quite simple [10,17]:

- the input vectors from the training set need to be taken (consecutively or randomly),
- the neuron which is closest to the given input vector needs to be found (i.e. the distance between $W_j = \{w_{ij} : i = 1..n\}$ and X is a minimum). The metric can be arbitrary, usually Euclidean, where the distance between the input vector and i-th neuron is defined as

$$d_i = \sqrt{\sum_{j=1}^{n} \left(x_j - w_{ij}\right)^2}, \tag{6}$$

- the weight vectors of the closest node need to be adjusted and the nodes around it in the way they move towards the training data (α is a coefficient factor, usually connected with the distance from the winning neuron),

$$\vec{W}_j = \vec{W}_j + \alpha\left(\vec{X} - \vec{W}_j\right) \tag{7}$$

- all steps need to be repeated for a fixed number of repetitions.

As a result of such learning neurons physically located next to each other in a Kohonen network correspond to classes of input vectors that are likewise next to each other. That is why such regions are called maps.

3.2. LEARNING ALGORITHM MODIFICATION

A little modification was added into the learning algorithm. One input vector can produce very different results on Kohonen network (Fig 4.) as each learning process can place winning maps in different parts of the network (top left corner, or bottom right corner). In automatic process the network needs to behave likewise. The most desired situation is to have silence in the top left corner (neuron number 0) and strong signal in the bottom right corner – then the contour will look more or less like an envelope but with similarity information (because this is not an exact envelope but Kohonen network result). A few methods were introduced [3] but “neighbour modification” is so strong that it often reverts the map. Now a new ‘modified Kohonen algorithm’ is introduced. This is a very simple and efficient method:

- Initiate 0-th neuron with zeros and mark its weights as read-only,
- It takes part in all computations but when it comes to weight changing we do not allow it.

Therefore 0-th neuron always pulls silence (which is always the weakest signal) to the top left corner, then the top left corner (with neighbours) gathers weak signal, therefore strong signal is naturally placed in the bottom right corner.

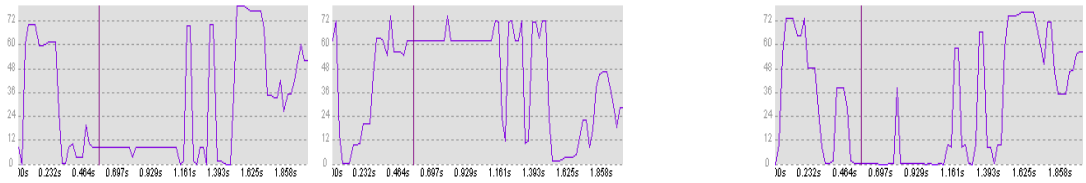

Fig. 4. Different Kohonen winning neuron contours for the same itterance. Right-most result is obtained using modified algorithm therefore silence is always placed in or near 0-th neuron. (*Y* axis: winning neuron, *X* axis: the time).
In this example Kohonen network was of the size 8x9 giving 72 neurons.

4. DISORDERED SOUND REPETITIONS RECOGNITION

4.1. INPUT DATA

The Polish speech recordings of 10 stuttering persons were taken of the summary length equal 548s. Based on [13,14] research – 3- second fragments were chosen with a disordered sound repetition. The rest (that is fluent speech) were divided automatically into 3-second fragments. That gave us 288 disordered repetitions fragments of sounds: b,d,g,k,n,o,p,t and 275 fluent fragments. The statistics are the following:

Table 2. Disordered sound repetition fragments counts

b	d	G	k	N	o	p	t	**sum**	fluent	**all**
18	6	10	86	2	1	61	104	**288**	275	**563**

4.2. TRAINING ALGORITHM

The procedure of finding sound repetitions in the file is the following:

- Compute CWT spectrogram of the continuous speech,
- Divide all the utterance into ‘small’ windows (23.2 ms) with the offset (23.2 ms). By using windowing (see section 2.4 for details) each ‘small’ window is converted into a set of 16 element vectors (each element of a vector corresponds to one bark scale),
- Divide the result into ‘big’ windows (3000ms). This process is not automatic. Disorders are found, placed more or less at the beginning of the 3s window and the window is marked as disordered. All remaining signal is automatically divided into 3s windows (cutting offset set to 300ms) which are marked as fluent speech (see Table 2),
- Each window which consists of 16-element vectors is automatically passed into the Kohonen network. After the training process a winning neuron graph is obtained (Fig 3). The 5x5 Kohonen network is used with the following parameters: 100 epochs, learning coefficient 0.20-0.10, and neighbour distance 2.5-0.5. Each graph is marked as fluent or non-fluent (this information is ‘the teacher’ in perceptron learning algorithm).

4.3. RECOGNITION ALGORITHM

All above steps are done using our tool – ‘WaveBlaster’

- Then STATISTICA neural network solver tool is used and 3-layer perceptrons with the best recognition ratio is found. As an input of perceptrons the Kohonen graphs are used and as an output the decision fluent/non-fluent is obtained. The input vectors are divided randomly into teaching set (50%), verifying set (25%) and testing set (25%),
- The recognition ratio is calculated with the use of these formulas:

$$sensitivity = \frac{P}{A};\ predictability = \frac{P}{P+B} \tag{8}$$

where *P* is the number of correctly recognized disorders, *A* is the number of all disorders and *B* is the number of fluent sections mistakenly recognized as disorders.

5. RESULTS

We wanted to test how ‘modified Kohonen algorithm influences the recognition ratio (see section 3.2 for details). We also wondered if 3s Kohonen graph contains too much unnecessary information which can lower the recognition ratio so we also did tests when only 1s ‘prefix’ out of 3s graph is passed onto perceptron. Therefore we also tested how the Kohonen ‘prefix’ influences the recognition ratio.

The following four tests were performed (the results for the best perspetrons found by STATISTICA ‘neural network solver’ are presented in the Table 3):

1. modified Kohonen algorithm: NO, Kohonen prefix: YES,
2. modified Kohonen algorithm: YES, Kohonen prefix: YES,
3. modified Kohonen algorithm: NO, Kohonen prefix: NO ,
4. modified Kohonen algorithm: YES, Kohonen prefix: NO.

Table 3.Disordered sound repetition recognition results.

DESCRIPTION: *MLP: 3-2-1 means how many neurons were in each perceptron layer with learning algorithms: e.g. BP100 – back propagation with 100 epochs, CG22b – conjugate gradients 22 epochs A – disordered fragment, P – disorders correctly recognized, B – fluent mistakenly recognized as disorders*					
1. modified Kohonen algorithm: **NO**, Kohonen prefix: **YES**, MLP:43-86-1 (BP100) – 43 inputs because of *1s Kohonen graph input*					
set	A	P	B	**sensitivity**	**predictability**
All	288	213	72	**74%**	**75%**
Teaching	135	114	36	**84%**	**76%**
Verifying	77	56	20	**73%**	**74%**
Testing	76	43	16	**57%**	**73%**
2. modified Kohonen algorithm: **YES**, Kohonen prefix: **YES**, MLP:43-45-1 (BP100,CG21b) – 43 inputs because of *1s Kohonen graph input*					
set	A	P	B	**sensitivity**	**Predictability**
All	288	265	45	**92%**	**85%**
Teaching	150	150	12	**100%**	**93%**
Verifying	66	56	14	**85%**	**80%**
Testing	72	59	19	**82%**	**76%**
3. modified Kohonen algorithm: **NO**, Kohonen prefix: **NO**, MLP:130-103-1 (BP100,CG22b) – 130 inputs because of *3s Kohonen graph input*					
set	A	P	B	**sensitivity**	**predictability**
All	288	243	35	**84%**	**87%**
Teaching	141	138	0	**98%**	**100%**
Verifying	73	52	20	**71%**	**72%**
Testing	74	53	15	**72%**	**78%**
4. modified Kohonen algorithm: **YES**, Kohonen prefix: **NO**, MLP:130-80-1 (BP100,CG22b) – 130 inputs because of *3s Kohonen graph input*					
set	A	P	B	**Sensitivity**	**predictability**
All	288	264	21	**92%**	**93%**
Teaching	149	147	0	**99%**	**100%**
Verifying	65	52	14	**80%**	**79%**
Testing	74	65	7	**88%**	**90%**

6. CONCLUSIONS

As we can see zeroing 0-th neuron gives very good result. In scenario 1 and 2 the sensitivity was increased by about 15%-25% and predictability by about 10% (depending on the set). In scenario 3 and 4 the numbers increased by about 8% for sensitivity and 5% for predictability. This result is rather obvious as this algorithm makes Kohonen more stable (the same sounds are placed in, more or less, the same areas of the network for various network learning) which is crucial for perceptron learning algorithm. For each vector a new Kohonen network is learnt but then results from all those networks are passed onto one MLP network. If some Kohonen's wining contours have silence in the top left corner, some in top right corner and so on – we can say that they are shifted or reversed. This causes MLP to have difficulties in finding general similarities. When MLP receives more stable (not reversed from time to time) winning contours – it can focus on non-fluency features only.

The second idea, learning perceptron with only 1s Kohonen graph prefixes, turned out to be a bad one. The results are significantly lower. By looking on the high recognition ratios (especially in scenario 4) we can see that whole 3s window is needed by perceptron and high verify and test ratios are the proof that the network generalizes well (network didn't learn some insignificant detail which cause high teaching ratio and low verifying and test ratio).

The best results were achieved for perceptron MLP 130-80-1 which- as an input- received all 3s Kohonen vectors trained with 'modified Kohonen algorithm'. 88% sensitivity and 90% predictability are very promising. The next step will be to take such a learnt MLP, implement it into WaveBlaster program and create fully-automatic recognition algorithm.

BIBLIOGRAPHY

[1] AKANSU A.N, HADDAD R.A., Multiresolution signal decomposition, Academic Press, 2001.

[2] CODELLO I., KUNISZYK-JÓŹKOWIAK W., Wavelet analysis of speech signal, Annales UMCS Informatica, 2007, AI 6, pp. 103-115.

[3] CODELLO I., KUNISZYK-JÓŹKOWIAK W., KOBUS A., Kohonen network application in speech analysis algorithm, Annales UMCS Informatica, 2010 (Accepted paper).

[4] CODELLO I., KUNISZYK-JÓŹKOWIAK W., SMOŁKA E., KOBUS A., Prolongation Recognition in Disordered Speech, Proceedings of International Conference on Fuzzy Computation, Valencia, Spain, October 2010, pp. 392-398.

[5] GARFIELD, S., ELSHAW M., AND WERMTER S., Self-orgazizing networks for classification learning from normal and aphasic speech, 23rd Conference of the Cognitive Science Society, Edinburgh, Scotland, 2001.

[6] GOLD, B., MORGAN, N., Speech and audio signal processing, John Wiley & Sons, INC, 2000.

[7] GOUPILLAUD P., GROSSMANN A., MORLET J., Cycle-octave and related transforms in seismic signal analysis, Geoexploration, Vol. 23, 1984-1985, pp. 85-102.

[8] HORZYK A, TADEUSIEWICZ R, Self-optimizing neural networks, Advances in neural networks - ISNN 2004, pt. 1, Lecture notes in computer science 3173, pp. 150-155.

[9] HORZYK A, TADEUSIEWICZ R, Mechanisms, symbols and models underlying cognition, Proceedings, Lecture notes in Computer Science, pt. 1, 3561, 2005, pp. 156-165.

[10] KOHONEN, T., Self-Organizing Maps, 34, 2001, pp.2173-2179.

[11] NAYAK J., BHAT P.S., ACHARYA R., AITHAL U.V., Classification and analysis of speech abnormalities, Elsevier SAS, Vol. 26, Issues 5-6, 2005, pp. 319-327.

[12] SMITH J., ABEL J., Bark and ERB Bilinear Transforms, IEEE Transactions on Speech and Audio Processing, November, 1999.

[13] SZCZUROWSKA I., KUNISZYK JÓŹKOWIAK W., SMOŁKA E., Speech nonfluency detection using Kohonen networks, Neural Computing and Application, Vol. 18, No. 7, 2009, pp. 677-687.

[14] SZCZUROWSKA I., KUNISZYK-JÓŹKOWIAK W., SMOŁKA E., Application of Artificial Neural Networks In Speech Nonfluency Recognition, Polish Jurnal of Environmental Studies, Vol. 16, No. 4A, 2007 pp. 335-338.

[15] SUSZYŃSKI W., KUNISZYK JÓŹKOWIAK W., SMOŁKA E., DZIEŃKOWSKI M., Automatic recognition of non-fluent stops, Annales UMCS Informatica, 2004, pp. 183-189.

[16] TADEUSIEWICZ R., Elementarne wprowadzenie do sieci neuronowych z przykładowymi programami, Akademicka Oficyna Wydawnicza, Warszawa, 1998, (in Polish).

[17] TADEUSIEWICZ R., Sieci neuronowe, Akademicka Oficyna Wydawnicza, Warszawa, 1993, (in Polish).

[18] TRAUNMÜLLER H. Analytical expressions for the tonotopic sensory scale, J. Acoust. Soc. Am., Vol. 88, 1990, pp. 97-100.
[19] WIŚNIEWSKI M., KUNISZYK-JÓŹKOWIAK W., SMOŁKA E., SUSZYŃSKI W., Improved approach to automatic detection of speech disorders based on the Hidden Markov Models approach, Journal of Medical Informatics & Technologies Vol. 15, 2010, pp. 145-152.
[20] KOBUS A., KUNISZYK-JÓŹKOWIAK W., SMOŁKA E., CODELLO I., Speech nonfluency detection and classification based on linear prediction coefficients and neural networks, Journal of Medical Informatics & Technologies Vol. 15, 2010, pp. 135-144.