ATLANTIS PRESS

Advances in Computer Science Research, volume 93

2nd International Conference on Mathematics, Modeling and Simulation Technologies and Applications (MMSTA 2019)

Adaptive Resonance Theory Neural Network for Phoneme Perception and Production

Marius CRISAN

Department of Computers and Information Technology, Polytechnic University of Timisoara, Blvd. V. Parvan 2, 300223 Timisoara, Romania

***Abstract*—The paper discusses the possibility of developing a hybrid adaptive resonance theory neural network architecture that can model the dynamics of speech perception and production starting from the sound constituents of phonemes. The architecture is composed of an adaptive resonance theory network coupled with a recurrent neural network. The hybrid network was trained to learn and generate successfully the elemental patterns of the main single vowel sounds in the English alphabet. The proposed configuration proved adequate to self-stabilize in real-time its learning independently of a teacher.**

Keywords—adaptive resonance theory; speech perception and production; adaptive pattern recognition; competitive learning; recurrent neural networks; time-series analysis

I. INTRODUCTION

One main research interest in the field of neural networks is to develop models for the functions of mind and brain. The progress on this direction was dependent on the results obtained from the interdisciplinary research into brain function involving the fields of neurophysiology, psychology and mathematics. Among other issues, of much interest is modeling how the somatosensory cortex represents and processes the time-varying sensory stimuli. This is a challenging task because it requires the account for network self-organization, at one hand, and plasticity or network's stability in a dynamic regime of learning, at the other hand. One of the first proposed structures that considered the effect of time on the network processing were the recurrent neural networks (RNNs) [1]. More modern approaches of RNNs employed the concept of 'reservoir' computing (a randomly generated RNN) in order to obtain a spatial representation of a time varying input signal, that may put in evidence a memory trace or trajectory of the recent network inputs [2][3]. In order to obtain a higher stability of the unsupervised learning, different mechanisms of plasticity were combined, using the same framework of reservoir computing as was discussed in [4]. Another perspective to the problem of learning stability in competitive networks was offered by the adaptive resonance theory (ART) [5][6][7]. A key problem of many contemporary neural networks is that they cannot learn rapidly new information in a dynamic environment without forgetting old information. ART addresses this so-called *stability–plasticity dilemma*. The ART model belongs to the class of competitive learning models and was initially inspired by researches that identified adaptive perceptual mechanisms in the visual cortex of mammals [8]. In principle, these mechanisms pertain to the process of vision normalization by which the visual perception can adapt to different optical factors and overcome the biological limitations of the primary visual apparatus. The initially developed adaptative resonance architectures were named ART1 and ART2 [9][10]. ART1 is the basic architecture that can stably learn a corresponding classification code of an arbitrary sequence of binary input vectors. ART2 networks can accept both binary and analog patterns, but the learning mechanism is the same. The major components of the architecture are called the attentional subsystem and the orienting subsystem. The attentional subsystem has two layers of neurons fully interconnected. One layer receives the input vector patterns and has the role of a short-term memory (STM). The activity traces of STM exist only during the presence of the input vector. The second layer has the role of a long-term memory (LTM) because it can record information for an extended period. The LTM layer allows the competitive learning after both layers achieve a resonant state. Information circulates back and forth between these layers, due to a feedback mechanism of the orienting subsystem, until a stable state ensues (equivalent of resonance). In this state, the learning of new information takes place without destroying old information. Learning doesn't take place prior to achieving the resonant state. ART architectures proved to overcome the problem of learning stability and are considered suitable building blocks for developing hierarchical structures that can manifest complex behavior [11][12].

In the present paper, the objective is to develop an ART hierarchical architecture that may account for adaptive speech perception and generation starting from the lowest level of sound constituents that establish phonemes. A challenging issue in the speech perception modeling is the compositionality constraint. This requires the explanation of how the combination of the sound constituents and the temporal order in which they appear lead to the formation of the perception of the meaning of words and language ultimately. The model should also be capable to reveal how the phonetic sounds combine in a proper dynamic sequence to form a pattern in the cortex that is classified according to a corresponding meaning. For instance, if the phonemes that compose a word are uttered in a correct sequence, but more distantly separated in time, they do not create a proper pattern for meaningful classification. The impressions of the phonemes fade away without combining one to another in the series, although one can still remember what phonemes have been uttered and their order. Another

requirement in modeling the neural network is the adaptability to the input length. Speech is a continuous process. Words may come in series, one after another, practically of any length and number and the network should be able to cope with them. The present approach considers that the ART model offers the necessary flexibility to deal with the requirements mentioned above and might be a suitable candidate for modeling the adaptive speech perception and generation.

The rest of the paper is organized as follows: In Section 2, the ART-type network structure and learning process are presented. Section 3 presents the simulation results of phoneme perception and generation, and the conclusions are drawn in the last section.

II. ART-TYPE NETWORK TOPOLOGY AND TRAINING

The present approach was thought to be consistent with the classic findings from speech perception and production theory that speech is perceived in direct relation to the capability of producing it even during passive listening [12]. The proposed hybrid architecture is composed of an ART module coupled with a recurrent neural network (RNN) as depicted in Figure 1. In the simplest form, the ART network consists of two interconnected layers. The bottom-up weights of connection from Layer 1 to Layer 2 constitute the matrix W_{21} that has one row for each unit of Layer 2 and one column for each unit of Layer 1. The top-down connection matrix W_{12} has one row for each unit of Layer 1 and one column for each unit of Layer 2. In the beginning, the input activation vector is encoded as bottom-up patterns in Layer 1. The output of Layer 1 is transmitted to Layer 2 where its units compete according to the value of their net-input and determines which row in the W_{12} matrix is closest to the input vector. The output from Layer 2 is the top-down expectation pattern that is send back to Layer 1 for comparison with the input vector. A resonant condition is attained if the patterns match within a *vigilance* parameter. Once reached the resonant state, the weights on both layers are updated to encode the input pattern. If large mismatches occur between bottom-up and top-down patterns, a reset signal is generated to start over the learning process. Learning activity in the two layers of the network leads to the formation of short-term memory (STM) and long-term memory (LTM) patterns. STM traces exist only during a single application of an input vector. Information that remains in the weights of the bottom-up and top-down connections for a longer period is called LTM. The version of ART used in the present experiments is derived from [9]. The top-down weights W_{12} are initialized to 0. This will prevent the network reset when a new unit of Layer 2 is being selected to encode a new input pattern. The bottom-up weights W_{21} can be initialized to small random values, but in order to favor an uncommitted node over previously mismatched nodes a uniform initialization was preferred, with a value very close to $1/[(1 - d)\ \mathrm{Sqrt}[N]]$, where d is a parameter, $0 < d < 1$, and N is the dimension of Layer 1. The processing equation of Layer 1 corresponds to the shunting model:

$$dx_k/dt = -Ax_k + (1 - Bx_k)I^+_k - (C + Dx_k)I^-_k \qquad (1)$$

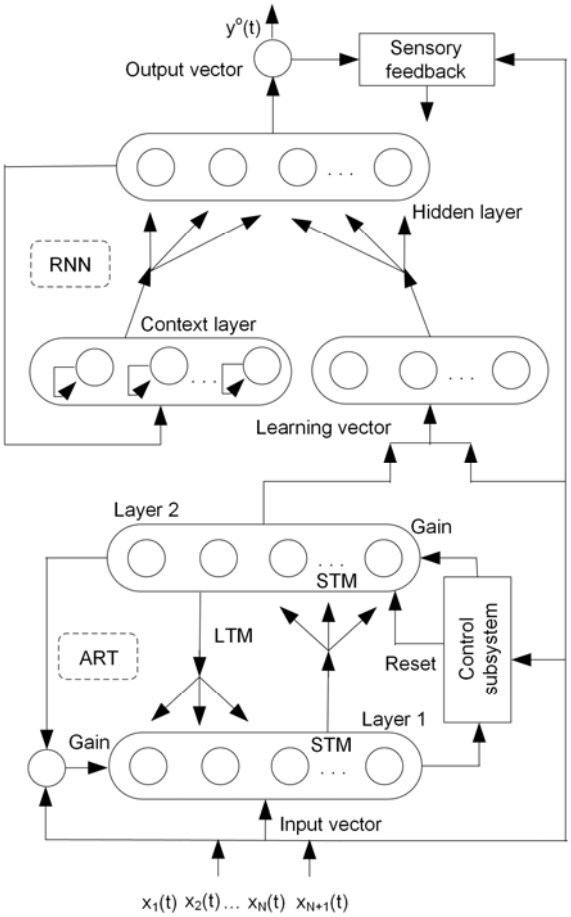

FIGURE I. SCHEMATIC STRUCTURE OF THE HYBRID ART ARCHITECTURE FOR SELF-ORGANIZING PHONEME PERCEPTION AND GENERATION.

where A, B, C, and D are constants. I^+_k is an excitatory input to the kth unit and I^-_k is an inhibitory input. The first term in (1) is a linear decay term, and the second and third terms provide nonlinear gain control. Layer 2 follows also the shunting model in its general form. The main difference from Layer 1 is that each unit receives a positive feedback from itself and sends inhibitory signals to the other units of the layer. This enables the network to store a pattern for an extended period and, also to perform a winner-take-all competition between units. The net-inputs to Layer 2 are the result of the dot products of the weight vector W_{21} and the output of Layer 1:

$$net_2 = [W_{21} \cdot O_1]. \qquad (2)$$

The winner is that unit with the largest net-input value that will select the prototype pattern closest to the output of Layer 1. Since Layer 2 is a winner-take-all competitive layer, the output value of this layer is $g(y_2) = d$ for the winning unit and 0 otherwise. After resonance has been established, only weights to or from the winning unit will get updated to the output value of Layer 1.

The second part of the proposed architecture has the structure of an RNN. Its input layer contains N units, the same dimension of the input vector of ART. The input to the RNN is provided by the LTM of ART. There are M units in both the hidden layer and the context layer that have the feedback role of a supplementary memory about the previous inputs. Each context unit has also a feedback connection to itself. The training of the RNN follows the standard generalized delta rule. In general, learning of RNNs requires a supervisor to supply the exemplars or input-output pairs. The present architecture is capable to self-stabilize in real-time its learning without using an external teacher. ART block has the role of adaptive learning and, also to supervise the training of the RNN. According to the LTM of ART, the following exemplars are formed: $IoPairs = \{[(x_1, x_2,\ldots, x_N), (x_{N+1})], [(x_2, x_3,\ldots, x_{N+1}), (x_{N+2})],\ldots, [(x_i, x_{i+1},\ldots, x_{i+N-1}), (x_{i+N})],\ldots, [(x_{K-N}, x_{K-N+1},\ldots, x_{K-1}), (x_K)]\}$, $i \in [1, K–N]$. The net-input of the hidden layer is $net_j^h(t) = \sum_i^N w_{ji}^h x_i(t) + \sum_c^M w_{cj}^h y_c^h (t–\tau) + \mu\, net_j^h(t–\tau) + \theta_j^h$, and the corresponding output is $y_j^h(t) = f_j^h(net_j^h(t))$. For the output layer, the net-input is $net^o(t) = \sum_j^M w_j^o y_j^h (t) + \theta^o$, and the produced value at the output becomes $y^o(t) = f^o(net^o(t))$. In the present experiments the activation function f for the hidden and output layers was the hyperbolic tangent function. The training goal is to minimize the mean square error for all the input-output pairs: $E_k = ½ \sum_k \delta_k^2$, where $\delta_k = (y_k^o – o_k)$, y_k^o is the output produced by the RNN and o_k is the desired output for the k-th exemplar. The equations for the weight training of the hidden and output layers are:

$$w_{ji}^h (t+1) = w_{ji}^h(t) + r\delta_{kj}^h x_i(t) + m\Delta w_{ji}^h(t), \quad (3)$$

$$w_j^o (t+1) = w_j^o(t) + r\delta_k^h y_j^h(t) + m\Delta w^o(t). \quad (4)$$

The parameters are the learning rate, $r << 1$, and the momentum term m, $0 < m < 1$. The weight differences are: $\Delta w_{ji}^h(t) = w_{ji}^h(t) – w_{ji}^h(t–1)$, for the hidden layer and, $\Delta w^o(t) = w^o(t) – w^o(t–1)$ for the output layer. The RNN has to learn to produce a new value x at the time $t + 1$ from the previous k time steps according to a function g: $\boldsymbol{R}^N \rightarrow \boldsymbol{R}$: $x(t+1) = g[x(t), x(t–1),\ldots, x(t–k+1)$. Both training and generation follow an iterative cycle. The input vector is formed by N samples from the acoustic time-series. At each iteration, the position $N+1$ is occupied by the previous output value and the sample from the first position is discarded. In this way, the values of the input slide over the entire training data. Once trained, the RNN becomes capable of generating the learned acoustic signals. At each step, the sound amplitude that was produced based on the previous samples is compared with the actual value from the input vector, and a sensory feedback signal confirms the learning process when the minimum expected value of the global error is met. One of the ART specifics is the function of top-down expectation mechanisms that help to self-stabilize the learned patterns and, also to self-organize their selection. In this regard, the sensory feedback functionality can play a role in the self-correction of the produced sounds.

III. Phoneme Perception and Generation

The present research was focused upon the main five single vowel sounds found in the English alphabet, /a/, /e/, /i/, /o/, and /u/. The vowel waveforms have a discernable periodic nature due to the combination of the main formants of their composition. A specific elemental pattern of period length could be identified in the waveform of any vowel. The repetition in time of such elementals produces the final vowel sound. The proposed speech perception and production technique is based on learning and reproducing the elemental patterns of the phonemes. The whole process of phoneme perception and generation consisted of three stages: (i) Stably self-learning the phoneme elemental patterns in the ART network; (ii) training the RNN to learn with a minimum error the shape of a particular elemental; (iii) generation of elemental patterns by the trained RNN in closed loop with previous data based on the ART network.

In the first stage of the experiments, the adaptive perception of ART network was tested for the main period pattern of the vowels. The sound data were sampled at 96 kHz with 16 bits. Layer 1 dimension was 360 and Layer 2 had 5 units. The vocal samples were arranged in the natural order of the alphabet, and the *vigilance* parameter was set to 0.9. The five input samples and the resulted LTM that was stably learned by ART are presented in Figure 2. As expected, the competition in Layer 2 revealed that the network became stable after the second iteration with the resonance established on unit 1 with pattern 1, on unit 2 with pattern 2 and so on, occupying all five units of Layer 2 in the temporal order of application. The high value of the vigilance parameter ensured a very close reproduction of the input vector, as can be noticed from the figure.

As the ART module proved capable of stably memorizing the input patterns for all the five vowels, in the next stage the RNN module was trained to learn the corresponding elemental patterns of the vowels based on the codes that are stored in the ART LTM. The training data was organized as *IoPairs* exemplars out of a series of three periods. The number of units in the hidden layer (context layer) of RNN determines the amount of influence of previous steps. High values of this number, comparable to the input layer, may lead to overfitting. A series of simulations were performed for learning the five elemental patterns of ART LTM with the following parameters: input layer, 360 – 410 units; hidden (context) layer, 100 – 140 units; $r = 0.1$; $m = 0.1$; $\mu = 0.01$. Convergence was noticed after several epochs. The performance of the RNN to learn the shapes of the elemental fluctuations can be observed from Figure 3. The RNN results were quite good. The most challenging shapes are those for /e/ and /i/ due to the presence of higher harmonics. It is useful to test the network capability to produce anticipated elemental patterns in so called “open loop,” when the generated samples are used as previous data. The results are presented in Figure 4, for the challenging case of vowel /i/. The shape “1” is the original input signal. The shape “2” was produced by the RNN with previous data supplied by the ART LTM. The shape “3” was generated in open loop. As expected, the open loop signal has the tendency to gradually become divergent after the first period. Without

the contribution of the ART by its LTM, the RNN alone is not capable of producing long sequence patterns.

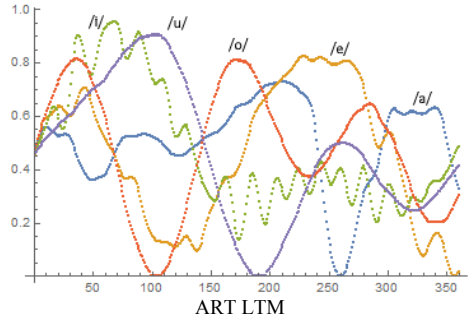

FIGURE II. THE ART LTM PERCEPTION OF THE INPUT SAMPLES FOR THE MAIN VOWELS /A/, /E/, /I/, /O/, AND /U/.

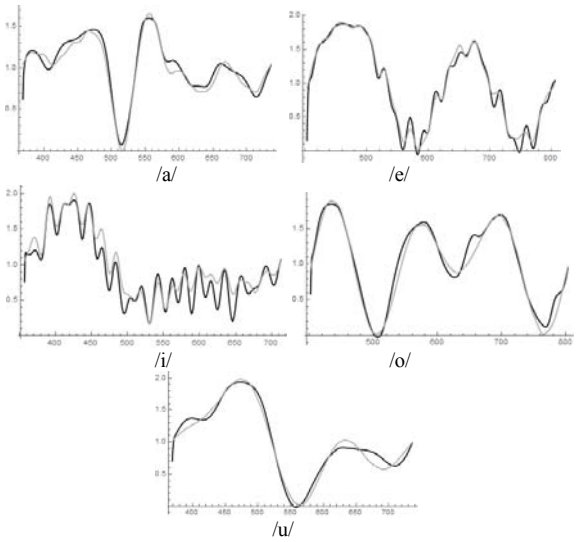

FIGURE III. THE GENERATED PATTERNS (BLACK) BASED ON ART LTM TRAINING DATA (GRAY) .

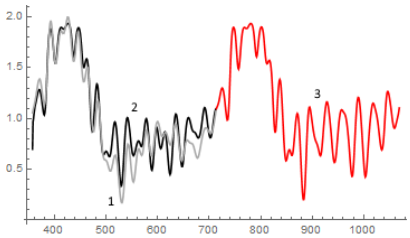

FIGURE IV. ANTICIPATED PATTERNS (VOWEL /I/). "1" – ACTUAL PATTERN (GRAY); "2" – THE PRODUCED SIGNAL IN CLOSED LOOP (BLACK); "3" - THE PRODUCED SIGNAL IN OPEN LOOP.

IV. CONCLUSIONS

The purpose of this work was to explore the possibility of speech perception and production starting from the lowest level of sound constituents of phonemes. The proposed architecture consists of an ART module in the core coupled with an RNN. The ART component has the role of adaptive perception and, also the role of training the RNN module to produce the perceived sound patterns, without the presence of a teacher. The present architecture is capable to self-stabilize its learning in real-time. An advantage of using the ART configuration is the seamless response to an increased capacity for stably learning larger patterns. The LTM part of ART can be easily connected to an RNN that have been proved to offer good results in generating or predicting time-series for relatively short periods as is the case of phonemes. When the input to the RNN is supplied by the LTM of ART, a self-organizing ensemble capable of speech perception and production can be developed. The hybrid network was trained to learn and generate successfully the elemental patterns of the main single vowel sounds in the English alphabet. The results obtained after simulation encourage to continue the research work on this direction.

REFERENCES

[1] J. L. Elman and D. Zipser, (1988), "Learning the hidden structure of speech," J. Acoust. Soc. Am. Vol. 83, pp. 1615–1626, 1988.

[2] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," Computer Science Review, vol. 3, issue 3, pp. 127-149, August 2009.

[3] S. Scardapane, J. B. Butcher, F. M. Bianchi, Z. K. Malik, "Advances in Biologically Inspired Reservoir Computing," Cognitive Computation, vol. 9, issue 3, pp 295–296, June 2017.

[4] A. Lazar, G. Pipa, and J. Triesch, "SORN: a self-organizing recurrent neural network," Frontiers in computational neuroscience vol. 3, article 23, pp. 1-9, Oct. 2009.

[5] S. Grossberg, "Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world," Neural Networks, vol. 37, pp. 1–47, 2013.

[6] L. E. Brito da Silva, I. Elnabarawy, D. C. Wunsch II, "A survey of adaptive resonance theory neural network models for engineering applications," arXiv:1905.11437v1 [cs.NE], submitted on 4 May 2019, in press.

[7] G. A. Carpenter and S. Grossberg, "Adaptive resonance theory," Encyclopedia of Machine Learning and Data Mining. C. Sammut and G. Webb, Eds. Berlin Springer-Verlag, 2016.

[8] S. Grossberg, E. Mingolla and D. Todorovic, "A neural network architecture for preattentive vision," IEEE Transactions on Biomedical Engineering, vol. 36, no. 1, pp. 65–84, Jan. 1989.

[9] G. A. Carpenter and S. Grossberg, "ART2: Self-organization of stable category recognition codes for analog input patterns," Applied Optics, vol. 26, no. 23, pp. 4919–4930, 1987.

[10] J. A. Freeman, Simulating Neural Networks with Mathematica, Addison-Wesley, 1994.

[11] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. De Jesus, Neural Network Design, 2nd ed., Publisher: Martin Hagan, September 1, 2014.

[12] G. A. Carpenter and S. Grossberg, "The ART of adaptive pattern recognition by a self-organizing neural network," Computer, pp. 77-88, March 1988.