Asian Journal of Information Technology 5(7): 686-690, 2006
© Medwell Online, 2006

Arabic Speech Synthesis Using Optimized Neural Networks with Genetic Algorithms

Rachid Hamdi and Mouldi Bedda
Department of Electronic, Faculty of the Engineer, Annaba University, Algeria

Abstract: This study describes a concept of an Arabic speech synthesis based on optimized neural networks. The genetic algorithm is used to perform the connection weight update. An Arabic database which contains subjects, verbs and complements and a speech synthesizer, whose objective is to make educational oriented verbal Arab sentences, is developed. Based on parallel architecture of neural networks, the system receives a set of sentences of three words. The structure of the network, the algorithms and the results are detailed.

Key words: Arabic speech synthesis, neural networks, genetic algorithms

INTRODUCTION

In recent years artificial neural networks have played an important role in various aspects of speech processing. One of the important applications of neural networks is solving speech synthesis problems. Text-to-speech synthesis includes the whole process permitting to a system to convert a written text (grapheme) into a vocal message (phoneme). This conversion is necessary for the application of speech synthesis from text such that the applications for learning machines, for blind persons and mail reading. The two traditional methods of grapheme-to-phoneme conversion, synthesis by rule and synthesis by concatenation have some disadvantages such as the need of huge database and the quality of the synthesized speech is not satisfactory[1-3]. So, several new techniques that improve text-to-speech synthesis by using neural network have been developed.

As a matter of fact, the neural networks have been used years ago, since 1890 by W.Jams and in 1949 by D. Hebb. The first success was in 1957 by F. Rosemblat who developed the perceptron model. The Nettalk system of J.J Sejnowski and C.R Rosenberg [SEJ87][4] which uses the neural networks has been developed to give better results to speech synthesis systems. In this paper, we propose to use a solution of this kind to synthesize Arabic sentences. The artificial neural network model is defined by several parameters that can be optimized. The method of setting the values of the weights is an important distinguishing characteristic of different neural networks. The wrong choice of the initial weights can lead to a phenomenon known as premature saturation[5]. Most training algorithms such as back propagation and conjugate gradient descent algorithms are based on gradient descent. But back propagation has drawbacks due to its use of gradient descent, slow convergence and possibility of being trapped at locally minimum value[6]. In order to get better performance of the model and to overcome the disadvantages of the back propagation, we use the genetic algorithms during the training of the initial weights of the network connections.

ARABIC SPEECH PROCESSING

Arabic standard language: Learning to read Arabic text aloud is a difficult task because the correct pronunciation of a letter depends on the context in which the letter appears. The presence of the diacritic marks in the Arabic text is essential for the implementation of the automatic text-to-speech system. Unfortunately, most modern written Arabic, as in books and newspapers, is at the best partially vowelized. A verbal sentence is a straightforward in canonical order the situation[7]. In this case grapheme-to-phoneme conversion for a verbal sentence can be performed with less complexity using optimized neural networks.

The basic unit for describing how speech convoys linguistic meaning is the phoneme. The standard Arabic language has basically 34 phonemes which are 28 consonants, 3 short vowels and 3 long vowels. The short vowels are /a/ ‘ةحتف ‘, /i/ ‘ةرسك ‘, /u/ ‘ةمضلا’ and the long vowels which are of the same quality but of long duration /aa/ ‘ا’,

/uu/ ‘و’, /ii/ ‘ى’. Several factors effect the pronunciation of the phoneme in the syllable as initial, closing, intervocalic or suffix[8]. There are 19 ways of pronunciation of an Arabic letter. A consonant can be pronounced with 6 vowels and with signs as tanween [˘ , ˝ , ˏ], tachdid or gemination [ˉ] and sukune [˚˚].

Corresponding Author: Rachid Hamdi, Department of Electronic, Faculty of the engineer, Annaba University, Algeria

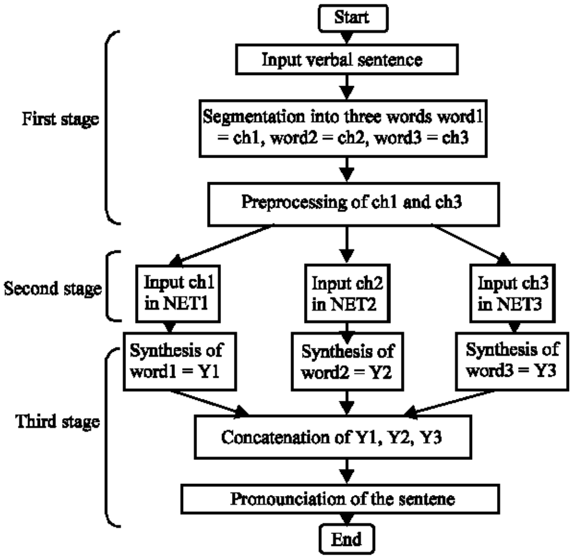

Fig. 1: Structure of the arabic synthesizer

Structure of the arabic speech synthesizer: Arabic is a rule based language in which the pronunciation of the word obeys known rules. Thus, before the word enter the neural network, pre-processing is useful to achieve a desired accuracy of the pronunciation.

The Arabic speech synthesizer consists of three stages; the structure of the adopted synthesizer with the three detailed stages is presented in Fig. 1.

The major stages are:

- The input stage which consists of a set of three words that are pre-processed in respect to Arabic rules and coded to ASCII, then placed in a matrix.
- The second stage consists of three networks (parallel architecture) optimized by genetic algorithms.
- The last stage is the transformation of the network results to phonemes which are concatenated to pronounce the written sentence.

To implement the Arabic speech synthesis process, we adopt the following algorithm which depends on some pre-processing and post-processing of the Arabic words. Before the word enters the neural network the suffix and prefix are separated from the original word[9]. To illustrate the function of the adopted Arabic speech synthesizer, we have developed the following algorithm:

Step 1: Input unvowelized verbal sentence.

Step 2: Segmentation of the sentence into three words, respectively denoted ch1, ch2 and ch3.

Step 3: The Alif and Lam at the start of the word are separated from ch1 and ch3.

Step 5: Coding the word in ASCII code (Coding each letter of the word).

Step 6: Input the coded word into the appropriate neural network (NET1, NET2, NET3)

Step 7: Decoding outputs of NET1, NET2 and NET3 for synthesis.

Step 8: Resolving the problem of Alif and Lam at the start of the word, Taa marbouta and maad at the end of the word

Step 9: Output phoneme

Step 10: Concatenation of the pronounced words to make the sentence.

STRUCTURE OF THE NEURAL NETWORKS

The proposed artificial neural networks consist of three multi-layers-perceptrons and are based on parallel architecture[10]. As it is mentioned earlier, the synthesized verbal sentence consists of three words (subject, verb, complement). So each network does the training of one word. A file containing the most common Arabic words, along with their phonetic spelling, is used to train the neural network. The training is supervised and the update of the weights is carried out by the genetic algorithm. The inputs are made up with standard Arab graphemes. These graphemes are coded in ASSCII code and then coded by place, which means that we use as many neurons as characters. One neuron corresponds to one character. At the input, the neuron takes the value of one if the character is present, otherwise it takes the value zero.

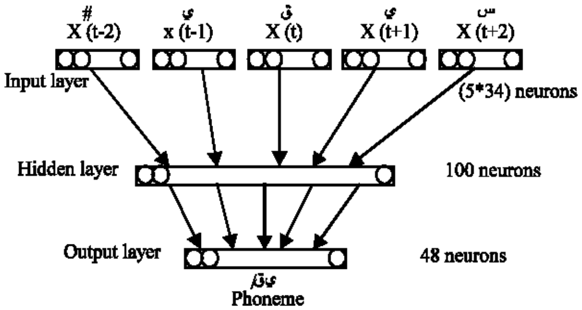

Fig. 2: Example of processing the letter 'ق' in the sequence 'سيقي #'

The first layer consists of 170 neurons split up into five blocks of 34 neurons which represent the 34 Arabic characters. The second layer (hidden layer) is made up with 100 neurons. Several tests for the number of neurons to be used in this layer have been done between 50 and 200. The network presents better architecture for 100 neurons in the hidden layer. The output layer contains 48 neurons for the 28 letters, 19 for the different ways of pronunciations and one for the silence. Each network has thus, 318 neurons and 21800 connections.

The system learns to transform an unvowelized written sentence into series of phonemes which correspond to its pronunciation. But the phoneme depends on the context in which the letter appears in the word. For example the letter ‘ق‘ is pronounced differently in ‘يقول‘ and in ‘سيقي‘. Consequently, we consider a sliding window, similar as to NetTalk[1]. The network receives sequences of five letters at the input which are the target letter to be processed, the two previous letters and the two following letters. Figure 2 shows this kind of principles for the word ‘سيقي‘. Thus one neuron from the 34 neurons of one block is activated by one character of the sequence to be processed. So during the training, just one neuron is activated from the 34 neurons of each activated block.

OPTIMIZATION OF THE NEURAL NETWORKS

Genetic algorithms: More recently, approaches of utilizing genetic algorithms to improve generalization ability of neural network in different areas have been successfully used[11,12]. The process of the GA leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptations. They can exceed other traditional methods with their robustness and are basically different according to four principal axes:

- GA uses a coding of the parameters and not the parameters themselves.
- They work on a population of points, instead of a single point.
- They use only the values of the function, not its derivative, or knowledge auxiliary.
- They use probabilistic and no deterministic rules of transition.

Therefore genetic algorithms are composed of selection and reproduction (crossover and mutation).

The choices of parameters which will be used to drive the neural network are very important and have an effective influence on quality of produced speech. Evolutionary programming to optimize the weights of neural network have been applied to improve learning efficiency of neural networks[13].

Optimization of the weight connections by genetic algorithms: A model of ANN type is defined by an important number of parameters which can be optimized by genetic algorithms. Thus, the weighting connections can be optimized for a network that has a number of layers known and a number of neurons known as well. During training by genetic algorithms the number of cells increases slightly, but in the meantime the connections number decreases rapidly. This means that there is an important complexity diminution of the population networks.

Our adopted network consists of 3 layers of respectively 170, 100 and 48 neurons which mean that there are 21800 connections or weights. So, the matrix of the weights consists of two parts: the input matrix IW with 100 rows and 170 columns and the output matrix LW with 100 rows and 48 columns. The transfer function of the hidden layer is the tan-sigmoid and the transfer function of the output layer is the linear transfer function. The initial population of the genetic model is represented by the initial weights w (m,n) and w (p,m) of the network as shown in Fig. 3, where in our case n = 170, m = 271 and p = 319. Consequently, the genotype of the network is: w(171,1);w(172,1);... ;w(271,170); w(271,171); w(272,171);...; w(319,271).

The dimension of the genome is 21800 and we have chosen a population of 100 genomes. A different probability of mutation and crossover has been used for the genetic algorithm in order to get better result. Finally, we have chosen a 5% of mutation and 80% of crossover. The fitness function is the error between the target and the actual outputs.

DATABASES

Speech database: Speech database is needed to train the neural networks used in the system. Arabic phonemes and diphones were recorded by a male speaker. The Arabic language has 28 consonants, 6 vowels and 13 different other ways of pronunciations. Thus, there are 19 phonemes for each letter. Consequently, there are 532 phonemes (28x19) to be recorded. In this application, we have used elements composed with 3 phonemes to make vowel letter. For the diphones, each letter is associated to a short vowel, so we recorded 2352 diphones (28x3x28). The recording data properties are:

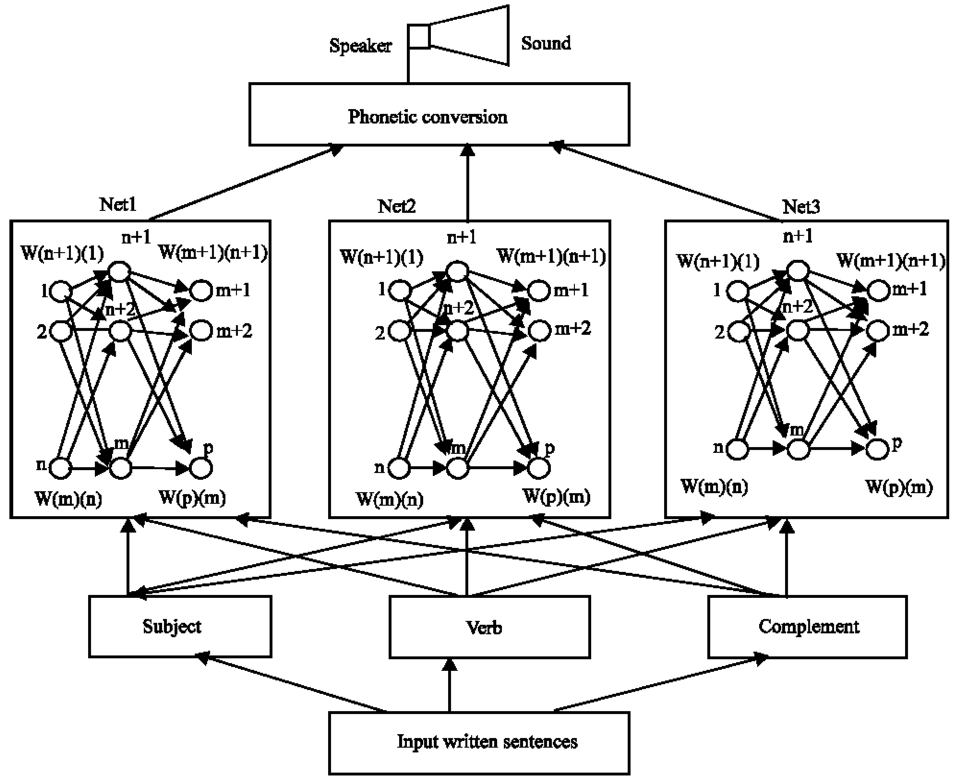

Fig. 3: Arabic grapheme-phoneme conversion system with optimized neural network

- Sampling rate is 11025 Hz.
- Number of bits for each sampling is 8.

Database for learning: The example of the training consists of two matrixes: an input matrix is used for the input layer of the network and an output matrix for the output layer. The input matrix consists of a list of 400 more common Arabic words. Each character of this list represents one column of 170 rows for the input matrix and one column of 48 rows for the output matrix. The input of the neural network is a series of sequences of 5 letters represented by an input matrix which correspond to a sequence of phonemes represented by an output matrix. For example, for the verb ‘كتب‘ the following sequences are presented to the network: # # كتب, # كتب #, كتب # # associated to the sequence of the phonemes /ك/,/ت/,/ب/. So each letter is mapped in a string of 5 bits. A silence is placed in the empty position. The learning of the network is carried out on a list of couples (sequence of characters, phonemes).

RESULTS AND DISCUSSION

The method to make an Arabic speech synthesis system was implemented under *'Matlab'* language and under Windows. By using genetic algorithms, the neural networks are able to recognize about 95% of the recorded words. During the listening, the pronunciation of the sentences is better than using the back propagation approach. But the training of the neural networks by genetic algorithm is time consuming. We can improve the problem of the training time by adding some Arabic rules in the development of this approach[8]. This rules decrease the amount of the processing time during training and increase the accuracy of synthesizing unknown words.

Table 1 shows the advantages and the disadvantages of using genetic algorithms.

Table 1: Neural networks training results

Neural network	Using back propagation algorithm	training Using genetic algorithm
Training time	10hours	more than 15 hours
Recognized words after training by NETs	336 words from 400	380 words from 400

CONCLUSION

In this study a system was designed for Arabic speech synthesis using neural network optimized by genetic algorithm. This speech synthesizer is used to convert written unvowelized Arabic sentences into phonemes in order to pronounce educational oriented verbal sentences (subject, verb, complement). It is suitable for real time applications because the access time is much lower than the base rule or concatenation methods. During the training, the genetic algorithm allows the neural network to converge very fast to global optimum. The neural network is able to recognize 95% of the recorded words which means that a good performance is reached.

In the future we will use genetic algorithms to set the number of neurons in the hidden layer as in[14]. Further more we are increasing the intelligibility of the system by using the GA for the semantic in order to pronounce only the sentences with the subject and its corresponding verb and complement. For example the system pronounces the following sentence ’بخلادلولالكأ‘ instead of this semantically wrong sentence ‘زبخلادولابرض’.

ACKNOWLEDGMENT

We are grateful to the laboratory of automatic and signal at Annaba University for its contribution in the development of this project.

REFERENCES

1. Karali, O., A. Gorrigan and I. Gerson, 1996. Speech synthesis with neural network World congress on Neural Network, San Diego, pp: 45-50.
2. Ben Sassi, S., R. Braham and A. Belghith, 2001. Neural speech synthesis system for Arabic language using CELP algorithm IEEE, pp: 119-121.
3. Sejnowski, J.J. and C.R. Rosenberg, 1987. Parallel networks that learn to pronounce English text Complex systems, 1 pp: 145-168.
4. Lee and Al, 1991. ‘Handwritten digit using K-nearest neighbour, Radial-basis functions and back propagation neural networks neural computation 3 pp: 440-449.
5. Xin Yao, 1999. ‘Evolving Artificial Neural Networks’ IEEE, pp: 1423-1447.
6. Ramsay, A. and H. Mansur, 2004. The parser from an Arabic text-to-speech system’ le traitement automatique de l’Arabe, JEP-TALN 2004, Fes, avril.
7. Al-Muhtaseb, H., M. Elshafie and M. Al-Ghamdi, 2002. Techniques for high quality Arabic speech synthesis Informatics and computer science V140, issue 3, pp: 255-267.
8. Hendessi, F., A. Ghayoori and T.A. Gulliver, 2005. A speech synthesizer for Persian text using a neural network with smooth ergodic HHM AGM transaction Asian language information processing pp: 38-52.
9. Dragos Burileanu, Mihai Sima and Adrian Neagu, 1999. A Phonetic Converter for Speech Synthesis in Romanian, Proceedings of the 14th Intl. Congress of Phonetic Sci. San Francisco, California, pp: 503-506.
10. Jansheng Wu and Mingzhe Liu, 2005. Improving generalization performance of artificial neural network with genetic algorithms.
11. Tsao, T.P., G.C. and S.H. Chen, 2001. Short-term load forecasting using neural networks and evolutionary programming’ ID proceedings of the fifth Intl. Power Engineering Conference, Singapore, pp: 443-748 .
13. Koza, J.R. and J.P. Rice, 1991. Genetic generation of both the weights and architecture for a network Neural Networks 1991 IEEE international conference pp: 397-044.
14. Liang Tian and Afzel Noore, 2004. Short-term load forecasting using optimized neural network with genetic algorithm’ 8th Intl. conference on probabilistic methods to power systems, Iowa state university, pp: 12-16.