OSJ | OPEN SCIENCE JOURNAL

RESEARCH ARTICLE

Speech synthesis using neural network

Noor Abbood[1]*, Sarah Abbood[2], Mohammed Salah[3]

[1]Middle Technical University, Technical Institute Baquba, Department of computer systems, Iraq
[2]Ministry of Education, General director of Education, Diyala Baqubah Iraq
[3]National University of Malaysia, Faculty of engineering and built environment, Malaysia

*Corresponding author: Noor Abbood: nooraldarraji@techbaq.mtu.edu.iq

OPEN ACCESS

Citation: Abbood N., Abbood S., Salah M. (2018) Speech synthesis using neural network. Open Science Journal 3(1)

Received: 28th August 2017

Accepted: 10th October 2017

Published: 28th February 2018

Funding: The author(s) received no specific funding for this work

Competing Interests: The author have declared that no competing interests exists.

Abstract:

In this letter presents a speech learning machine and develop it using Neural-Network. We benefited from the past researches, that depend on Neural network actually called NeTtallk and compare Net Talk model with Hidden Markov Model (HMM). This letter introduced the net Talk model to simulation, this model has ability to mimic the pronunciation of English vocabulary essentially the silent pronounced and the vowel sounds letters.

Keywords: Speech learning, Neural-Network, Net Talk, Hidden Markov Model (HMM)

Introduction

One of most important computer application is User interface with computer, User interfaces are used increasingly special in speech synthesis. The goal of Speech is to produced speech patterns through the text input, that processed always happened by a Back-Propagation Neural Network (BPNN) as explained in [1][2].

In 1943 Warren McCulloch and Walter Pitts, were prescribed a paper, in that paper review that how neurons works through moulding the neuron with in Roberts. Neural Network has many applications in different fields such as in classification, machine learning, experts systems and pattern recognition. As all the techniques that used in computer sciences, Neural Network have major challenge is slow in progress as compare with another optimized tool, for example

Fuzzy Logic. In Neural Network takes extended time to pick up the drive the correct or desired output. In spite of that, Neural Network is still the most widely used.

In the speech synthesis, has many products in the market that have able to transformation the text to speech. For example, one software used the python language called an Application Programming Interface (API)[3] and the other software was built by the deep learning called Ersatz [4][5].

There are many researchers interested in the speech synthesis field, (Lo, W& et al .1996) propose a modified vowel diagram to provide flexible articulatory control as well as saving a phone template in a network in the form of variables. The system was applied on Cantonese dialect [6]. Developed a time delay neural-network to perform the phonetic to acoustic conversion. The advantage was to reduce the memory used and improvement in performance [8][7].

In the middle of 1980, NET Talk developed at Johns Hopkins University [9]. NetTalk works based on previous learning phase, by develop a speech pronunciation that reinforce the strength of the network to vocalize abnormal phonological regularities and vowel sound for English sentences [9].

NETTalk

NETTalk system is consist of three layers as the following: the first layer called input layer, it's responsible for the introduction of the English text, the second layer called hidden layer, it's responsible for receives the text from the input layer and sends its to the output layer, third layer called the output layer, it's responsible for transformation the text to sound. As shown in figure 1.

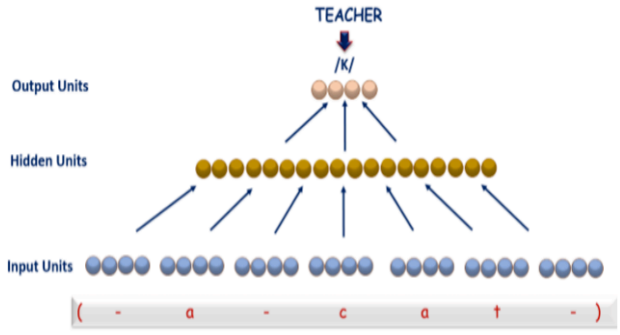

Figure 1. NETTalk Architecture [9]

The activation function in NETTalk of this letter is representing a sigmoid function, when the input is negative then output will be zero. When the input positive then the output equal one [10].

Processing unit

The network is consists from two units: the first unit called processing unit .in the processing unit will be used nonlinearly transform with totals input. The units are linking together, the value of the unit may be negative or positive

values, and the first unit either has an exciting or discouraging effect on the second unit.

Every one of these units have threshold, that the value of threshold is subtract from the sum input. It can consider the threshold as weight from unit, so when execute the threshold as weight that has a constant value of one so that the same entry and learning methods able to use to the threshold and weight. i_{th} is representing the output unit is calculated by the first combination of all inputs the first combination of all inputs [8].

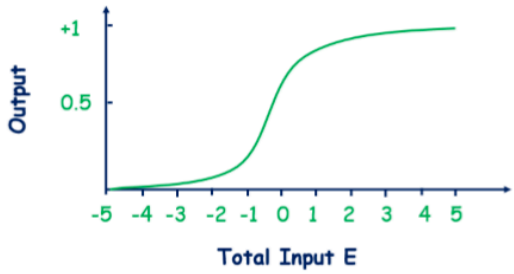

Figure 2. Activation function of single processing unit [9]

$$P(E_i) = \frac{1}{1 + e^{-1E_i}} \quad (1)$$

$$E_i = \sum_j W_{ij} S_j \quad (2)$$

We can define the variables above as the following: The input E_j is the sum of all outputs of the previous layer, denoted as S_j, multiplied with the weight of the synapse which connects the unit i in the previous layer and the unit j in the proceeding layer. The weights value, W_j , could be a positive or negative real value to reflect an inducing/excitation value or a preventing/inhibitory value for the signal that passes through. Besides the weighting of the synapses, each unit is equipped with a threshold that biases the inputs. The threshold is represented as a constant added to the sum of inputs.

Teaching Phase

In this letter, we used a back propagation algorithm in teaching phase. According to this scheme the output is calculated for a given input and compared with the correct pattern that is kept with the teacher. The objective is to minimize that different between the actual calculated output from the network and the preset correct pattern.

$$\text{Error} = \sum_{i=1}^{j} (S_i^* - S_i^{(N)})^2 \quad (3)$$

The superscript N denotes the N_{th} layer of the network, while the superscript i is the i_{th} unit of that corresponding layer and j is the number of units in the output layer.

The procedure of teaching starts with calculating the error gradient in the output layer by (4) and back-calculate the gradient for all the preceding units in the previous layer as in (5)

$$\sigma_i^{(N)} = \left(S_i^* - S_i^{(N)}\right) P' \left(E_i^{(N)}\right) \tag{4}$$

$$\sigma_i^{(N)} = \sum_j \sigma_j^{(n+1)} W_{ji}^{(n)} P' E_i^{(n)} \tag{5}$$

Superscript denotes again the layer number and by noting (5), it is clear that the gradient of proceeding layer relies on the calculated gradient of the top layer. The operation continues form the top layer towards the bottom layer until all the gradients have been calculated and updated their respective weighs. The Updating of the weigh for all the node's inputs are performed by generating to what is known as The Smoothed Weighted Gradient, ΔW_{ij}, by (6) and (7) respectively.

$$\Delta W_{ij}^{(n)}(t+1) = W_{ij}^n(t) + \in \Delta W_{ij}^{(n)} \tag{6}$$

Where

$$\Delta W_{ij}^{(n)}(u+1) = \propto \Delta W_{ij}^{n}(u) + (1-\propto)\sigma_i^{(n+1)} S_j^{(n)} \tag{7}$$

Operator α is called the smoothing parameter and is tuned usually to (0.9), while u is the number of the inputs. By calculating the smoothing parameter in (7), the weights of respective synapses are updated in (6) [9].

Hidden Markov Models

In the end of 20th century HMM has been more popular. But actually, in mid – 1990s the idea of HMM founded speech synthesis [11]. The process of speech synthesis based on HMM pass through the following steps:

1-parametric representations: The first step, from the speech will be extracted the parametric, inclusive to the filter and source parameters from the database that represented the speech and build model through using sub-word in HMM. To assessment the HMM parameters will use the maximum likelihood (ML) criterion, as explain below:

$$\lambda^{\wedge} = \arg max_{\lambda}\, p(o \mid w, \lambda) \tag{8}$$

We can define the parameter in equation (8) as the following:

λ: is represent a set of HMM parameters

O: is represent a set of the training data in HMM

W: is represent a set of the reproductions identical to (O)

From the parameters above, we can produce the set of estimated HMM parameters, as in equation (9)

$$o^{\wedge} = argmax_{o} p(o \mid w, \lambda^{\wedge}) \quad (9)$$

At the end, the parametric representation of speech is responsible for reconstructed a speech waveform. [12] It can be described the HMM based synthesis process which it generating the average of identical sounding speech segments. The utilize of statistical models and parametric representation offers another appealing points. The generality appealing point of it, using a small amount of speech data [13].

Simulation for speech synthesis models

According to [9], we will arrange the input layer in the neural network to seven groups, to be able accommodate a sentence or word (maximum seven letters). Each group of this groups consisted of (26) units to assimilate the (26) letters of the alphabet. The output layer in the neural network is consisted of the (21) units to assimilate the various phonetic status in the alphabet.

In the input layer, the letters represent in the ASCII format and look up table is use to encrypt the numerical value to transformation into array of binary code of (26) bits. This array will be consider the main source of information of the neural network.

.

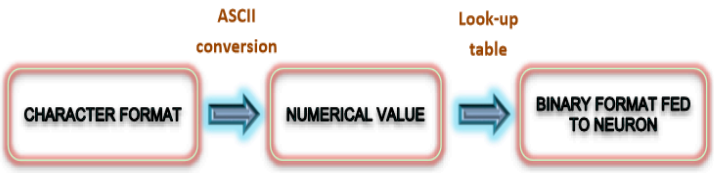

Figure 3. Encoding the letters for neural network input

There is one 5-state HMM for each phoneme, each group is consisted of (26) units to assimilate the (26) letters of the alphabet in every required context. To synthesize a given sentence, at first, use front end to predict the linguistic specification, concatenate the corresponding HMMs and finally generate the model from the HMM. As shown in figure below:

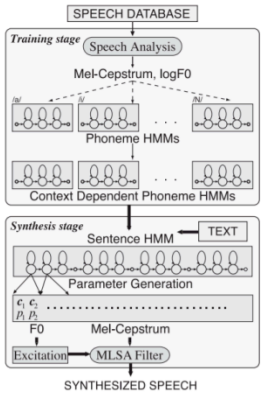

Figure 4. HMM based speech synthesis system

The following codes used to create the network for both neural network structure and Hidden Markov Model:

a) 2-4-3-1

b) 2-4-6-1

When we copy the data from the Microsoft word document to the workspace of mat lab software, Data storage is done in a variable called raw after that will be divided into variables: i for input and tar for output

As in figure 5, the initial simulation in neural network does not present any convergence in the result.

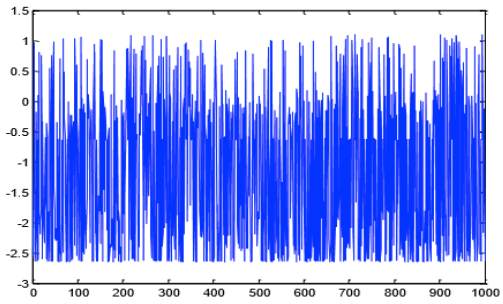

Figure 5. Output of network simulation

While the output of HMM simulation as shown in figure:

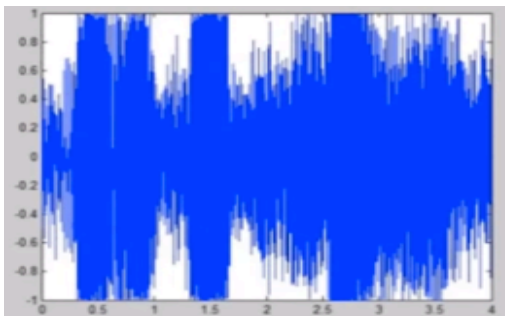

Figure 6. Output of network HMM simulation

In the Net Talk is trained by mean square error (MSE) while in Hidden Markov Model (HMM) is trained under the Maximum Likelihood estimation (MLE). Fig.7 shows that a convergence in the results appears after 1 epoch of training. The Mean Square Error is decrease until the stopping met.

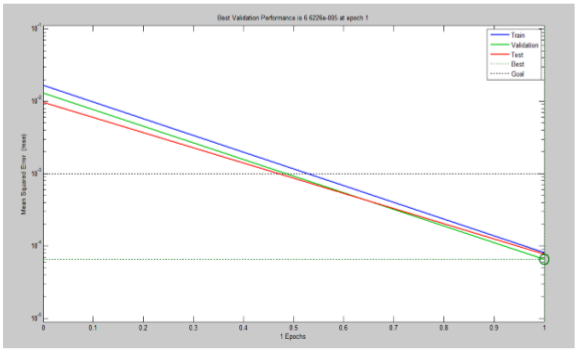

Figure 7. Mean square error for Net Talk

In Hidden Markov Model, the maximum likelihood estimation decrease only in the test and train while in the validation will be increasing and at final, the training and testing met but validation does not met, as show in figure below:

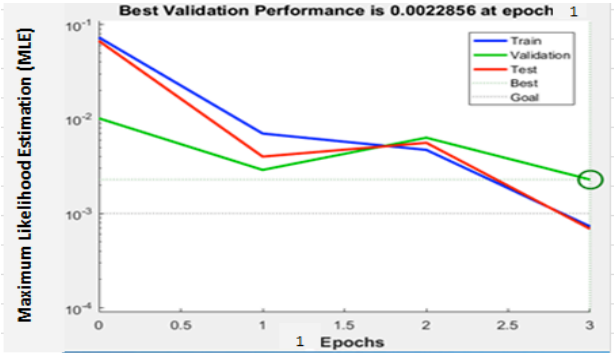

Figure 8. Maximum Likelihood Estimation for HMM

Conclusion

In this letter uses 3 neurons in the input layer leading to make the word size to 3 letters only. The back propagation network is appear an efficient performance in NetTalk, the value of performance of Net talk is (6.6226) while in Hidden Markov Model achieved the value of performance is (0.00228). However, if the number of the words increase may be have problems with the speed learning the model.

Over and above, we must investigate the speech synthesis on big bulk to the back propagation network with great time frame in order to training model.

References:

Rebai, I., &Benayed, Y. (2013). Arabic Textto Speech Synthesis Based on Neural Networks for MFCC Estimation. Computer and Information Technology, World Congress on (pp. 1-5). Sousse: IEEE.

Karaali, O., Corrigan, G., Massey, N., Miller, C., Schnurr, O., & Mackie, A. (1998). A High Quality Text-to-Speech System Composed of Multiple Neural Networks. Acoustic, Speech, and Signal Processing, IEEE International Conference on (volume 2) (pp. 1237-1240). Seattle, WA: IEEE.

Product Tour: Ersatz Lab. (2014). Retrieved from Ersatz Lab: http://www.ersatzlabs.com/services/

Product Demos. (2013). Retrieved from Voiceware: http://www.voiceware.co.kr/eng/product/product1.ph

Tachibana, R., & Nishimura, M. (2009). Patent No. 20090070115.Omah, NE, USA.

Speech synthesis (SpeechTech TTS). (2015). Retrieved from Speech Technology: http://www.speechtech.cz/en/products/speech-synthesis-tts.html

Lo,W., &Ching, P. (1996). Phone Based Speech Synthesis With Neural Network and Articulatory Control. Spoken Language, Fourth International Conference on (Volume 4) (pp. 2227-2230). Philadelphia, PA: IEEE.

Karaali, O., Corrigan, G., &Gerson, I. (1996). Speech Synthesis with Neural Networks. World Congress on Neural Network (pp. 45-50). San Diego, CA: International Neural Netowork Society.

Sejnowski, T. J., & Rosenberg, C. R. (1987). Parallel networks that learns to pronounce English text. Complex Systems (1), 145-168.

Sejnowski, T. J., & Rosenberg, C. R. (1986). NETTalk: a parallel network that learns to read aloud. Baltimore: Johns Hopkins University.

XinWang , Shinji Takaki&Junichi Yamagishi (2016).A Comparative Study of the Performance of HMM, DNN, and RNN based Speech Synthesis Systems Trained on Very Large Speaker-Dependent Corpora. 9th ISCA Speech Synthesis Workshop • September 13 – 15, 2016 • Sunnyvale, CA, USA.

Falaschi A., Giustiniani M., Verola M. (1989). A Hidden Markov Model Approach to Speech Synthesis. Proceedings of Eurospeech 89 (2): 187-190.

Lee K. (1989). Hidden Markov Models: Past, Present, and Future. Proceedings of Eurospeech 89 (1): 148-155.