Number of Output Nodes of Artificial Neural Networks for Korean Prosody Generation

Kyoung-Joong Min, *Chan-Goo Kang, Un-Cheon Lim

Department of Electronics Eng., Graduate School, Hoseo University, San 29-1, Sechul-Ri, Baebang-Myun, Asan-Si, Chungnam-Do, 336-795, Korea
* Dept. of Electronic Eng., Anyang Technical College
kjmin@bcline.com, alberto@aytc.anyang-c.ac.kr, uclim@office.hoseo.ac.kr

Abstract

We’d been studying artificial neural networks(ANNs) that can learn and generate the prosody of a Korean sentence. To hear more natural synthetic speech generated by a Korean TTS (Text-To-Speech) system, we have to know all the possible prosodic rules about Korean language and integrate all of these rules into an algorithm. We can get these rules from linguistic, phonetic knowledge or by analyzing real speech. But this algorithm cannot cover all the possible prosodic rules in a language, so the quality of synthesized speech cannot be as good as we expect.

We had trained BP (Back Propagation) ANNs that can learn the energy contour and the pitch contour of a phoneme in a sentence and generate the polynomial parameters of the contours that can be used in TTS system. The prosodic contours of a phoneme can be approximated by polynomial equations and the order of the polynomial equations can be determined according to the various conditions. In this paper, we had compared the performances of ANNs with different number of output nodes.

1. Introduction

To improve the naturalness of synthetic speech of TTS system, a set of accurate prosodic rules has to be integrated into an algorithm in both methods. These prosodic rules are extracted by syntactic analysis or statistically from natural speech. But these rules are not complete and inaccurate to cover up the prosody of natural speech exactly. So the algorithm for prosody generation in TTS has to be modified continuously to adopt new prosodic rules for more natural synthetic speech.

We propose ANNs with different number of output units, which will learn the prosodic contours from real speech and generate them for Korean TTS, and we evaluated the estimation rates of ANNs. To train and test these ANNs, and to compare the estimation results of them, we made a corpus that consists of some meaningful sentences that were collected from a phonetically balanced(PB) isolated word corpus.

The sentences of this corpus were read by a male speaker three times in a row, recorded, and gathered as raw speech data. The sentences of this corpus were read by a male speaker three times in a row, recorded, and gathered as raw speech data. And using short-time autocorrelation method, we made speech data base(DB) that consisted of 10 linear predictive coding parameters and energy and pitch information per an analysis frame. We had extracted the prosodic information of each phoneme from speech DB and made an approximation to the prosodic contour and got prosody DB using curve fitting method. We had built up 2 sets of prosody DB for ANNs with different number of output units. We had trained ANNs and tested and compared their estimation rates.

2. Prosody in a Korean sentence

We call the variations of duration, intensity and pitch of phonemic segment in a sentence, in a phrase, or in a word according to various environmental conditions, the prosody. These parameters for each phoneme are supra-segmental features of time, not restricted to a single segment and they vary at from acoustic level to emotional level. At emotional level the prosodic parameters are varied by speaker's affect, intent, personality and speaking rate. It's not easy to make prosodic rules that take account of all these variations, so we have to make some restrictions such as that a male speaker, normal speaking rate, flat reading style at emotional level. Preceding and following phonemes, if a phoneme is stressed or not, if there is a stop or a pause before/after it, the number of phonemes in a prosodic phrase, and a relative frequency of the word that includes this one, are the factors that affected the length (duration) of that phoneme in a phrase.

There were some studies that said the length of a vowel with following unvoiced consonant was shorter than that with voiced one and the duration of a vowel preceding a fricative consonant was longer than that stop consonant etc. As the number of syllables in a word increases, the duration of each component should be decreased with different rates within a certain limit.

In Korean, syntactic boundaries such as phrase boundaries, accents in word, and segmental phonemes were said to affect the pitch variation of each phoneme in a sentence. In English, the pitch contour has been considered as an indication of the syntactic structure in spoken sentences, and most of TTS systems used the concept of base line and rise and fall approach to generate the fundamental frequency F0(a reversal of pitch period) and implemented it in a sentence or in a clause.

There are few papers about the intensity variations of phonemes in a sentence. But in Korean it’s known that the intensity of a phoneme is varying according to it's own sonority, the stress pattern in a word, and the relative position in a sentence.

There were some papers said that prosodic cues played an active role not only at syntactic level, but also at lexical and syllabic level as well. The lexical stress pattern in a word played an important role, and the sentential level stress was more important. It's pointed out that base line of pitch could be detected in a prosodic phrase rather than in a sentence.
And it’s known that the relative position of the phoneme in a prosodic phrase was more important than in a sentence.

3. ANNs with different output units

If there is a way to learn all these rules about prosody from natural speech, and there will be no need to modify them again and again. We thought ANNs would learn the prosodic rules in Korean sentences and generate them for Korean TTS system to improve the naturalness of synthetic speech. We designed the input and the output units of ANNs and the number of hidden layer and the number of each node in each layer so to let them learn the prosodic rules residing in natural speech. We have to consider the environmental conditions of a phoneme in a prosodic phrase to learn and generate it’s prosodic information accurately.

BP(Back Propagation) neural network shown in Figure 1 that consisted of 11 input phones and 4 to 5 output units was set to learn one of the prosodic contours of each phoneme in a phrase. Two BP neural networks had been trained to learn pitch and energy variations of the center phoneme from the 11 phoneme strings in a prosodic phrase. To determine the architecture of ANNs, we had to know the prosodic features in Korean spoken language.

We have to consider the environmental conditions of a phoneme in a prosodic phrase. To determine the order of the equation is set to higher number, the prosodic contour. As the order changes, the prosody DB and the architecture of ANN should be determined.

The number of phonemes in a prosodic phrase can be from 2 to over 10. In order to take account of the supra-segmental effects of the preceding and following phonemes and the lengthening effect of the final syllable in a prosodic phrase, we assign 11 phoneme strings to the input layer of ANN. From these phoneme strings, the 6th phoneme will be the center phoneme, and the prosodic features of this one will become the output units of ANN.

There are 18 initial consonants, 21 medial vowels and 7 final consonants available in a syllable after going through a proper phonological processing on a Korean sentence. In addition to these phonemes, we have to include some syntactic symbols such as a period, a comma, a question mark and a blank etc. To represent all these symbols and phonemes, we just need 6 bits for a phoneme or a symbol, but we add extra 2 bits to use them later if there is a need to cover other information useful for the prosody generation. So the total number of bits in input layer of ANN will become 88 bits.

For nonlinear mapping between the input and output pattern, we have to adopt hidden layers. We select one hidden layer for the learning of prosodic information, and let the number of nodes of hidden layer be the same as the number of input nodes.

We can see the spectral and prosodic variations of real speech through short-time analysis on speech data. Especially we can draw a pitch contour and an energy contour of one phoneme or a sentence. We can approximate these contours using curve fitting method.

The pitch contour of a Korean vowel and it’s estimated one is shown in Figure 2.

To approximate this pitch contour, we used 2nd and 3rd order polynomial equation and built up prosody DB using curve fitting method and determined the output units of pitch neural network.

The pitch contour of a phoneme can be approximated with 2nd order polynomial equation (1),

\[ p_d(n) = p_{22}n^2 + p_{12}n + p_{02}, \quad 0 \leq n < d \]  

where \( d \) is the number of frames of a phoneme, \( p_{22} \) and \( p_{12} \) are polynomial parameters of it’s pitch contour, and \( p_{02} \) is it’s initial pitch period. With this equation and the other 10 phonemes, there is no need to consider the fall-rise pattern.

The output layer of pitch neural network with 2nd order polynomial would be consisted of 4 units, which represent the
duration and 2 polynomial parameters with an initial value of central phoneme's pitch contour. When the pitch contour of a phoneme can be approximated with 3\textsuperscript{rd} order polynomial equation (2),
\[ p(n) = p_{33}n^3 + p_{23}n^2 + p_{13}n + p_{03}, \quad 0 \leq n < d \quad (2) \]
where \( p_{33}, p_{23} \) and \( p_{13} \) are polynomial parameters of it's pitch contour, and \( p_{03} \) is it's initial pitch period.

The output layer of pitch neural network with 3\textsuperscript{rd} order polynomial would be consisted of 5 units, which represent the duration and 3 polynomial parameters with an initial value of central phoneme's pitch contour.

The energy contour of a Korean vowel and it’s estimated one is shown in Figure 3.

![Figure 3. The energy contour and it’s estimated one of a Korean vowel.](image)

The energy contour of it can be described by the 2\textsuperscript{nd} order polynomial equation (3),
\[ e(n) = c_{22}n^2 + c_{12}n + c_{02}, \quad 0 \leq n < d \quad (3) \]
and \( c_{22} \) and \( c_{12} \) are polynomial coefficients of it's energy contour, and \( c_{02} \) represents it's initial energy.

The output layer of energy neural network with 2\textsuperscript{nd} order polynomial would be consisted of 4 units, which represent the duration and 2 polynomial parameters with an initial value of central phoneme's energy contour.

The energy contour of a phoneme can be approximated with 3\textsuperscript{rd} order polynomial equation (4),
\[ e(n) = c_{33}n^3 + c_{23}n^2 + c_{13}n + c_{03}, \quad 0 \leq n < d \quad (4) \]
where \( c_{33} \), \( c_{23} \) and \( c_{13} \) are polynomial parameters of it's energy contour, and \( c_{03} \) is it's initial energy.

The architecture of energy neural network will be the same as that of pitch neural network.

If the sampling frequency is 11.025 KHz, the size of short-time window is 256 samples, and the overlapping interval is 128 samples, then the time interval per frame will be 11.61 msec. The number of frames per phoneme ranges from 1 to 26 frames. To represent the duration, it's is enough to assign 6 bits, but we assign 8 bits(minimum data size) to the duration \( d \) module. The initial pitch period varies from 4 to 20 m sec, i.e. from 4 to 220 samples, so we can assign 8 bits for the initial pitch period. To represent polynomial coefficients, we assign 32 bits for each one. The total units of output layer will become 80 bits. In the case of 3\textsuperscript{rd} order polynomial, that will be 112 bits.

The dynamic range of initial energy computed from short-time analysis is too large, so most of synthesis/analysis methods adjust this energy not to exceed 256 with the window size. So we can also assign 80 or 112 bits for the output layer of energy neural network according to the order of polynomial equation.

In BP networks, we used a sigmoid function as a transfer function of an artificial neuron. We adopted gradient descent method to adjust weights in ANNs. When the input phoneme strings of a sentence were given to ANNs, they generated an output pattern of the central phoneme, we compared it with a target pattern by computing the RMS error between the output and target pattern. To minimize this RMS error, the networks had to adjust their weighting values. We called this process one epoch. When we set the maximum number of epochs to a certain value, and the RMS error goes below the threshold, then let the network stop the weight adjustment. We can get the estimation rate in learning phase by counting the number of phonemes that can't be learned with an RMS error below the threshold within maximum epochs.

### 3.2. Evaluating ANNs

We made a corpus to collect the prosodic information residing in natural speech, and recorded natural speech spoken by a male speaker with normal speed based on the corpus. We made a speech DB and 2 sets of prosody DB. With these prosody DB, ANNs have been trained and tested.

We selected some words from 412 PB words and grouped them to meaningful sentences or phrases. We could make a corpus that consists of 100 meaningful sentences or phrases. Some words were used several times.

The number of syllables in a sentence varies from 4 to 20, and the number of phonemes varies from 10 to 50. And the number of phonemes in a breath group can be over 20 according to speaker's reading style. But it is rare. We have to limit the number of phonemes in a prosodic phrase because the number of input units of ANNs cannot be too large to include all of them.

A male speaker read one sentence with some prosodic boundaries 3 times in a row. Real speech spoken by the speaker with normal speed were recorded and made into a speech DB. This speech DB was analyzed with short-time analysis algorithm, and after applying a segmentation algorithm to these analyzed data, we could get the pitch and energy contour of each phoneme.

We got the polynomial coefficients and initial values by applying curve fitting method to the pitch and energy contour, made them into target and test patterns for ANNs. With these polynomial parameters, the duration of each phoneme will be grouped into prosody DB.

In training phase, we used the 1st and 2nd set of prosody data. At first the network will be trained within maximum epochs and the threshold value for the minimum RMS error was set. If the error went below that threshold before 200th epoch, we stopped the adjustment of the weighting values of ANNs. We got the estimation rate in learning phase by counting the number of phonemes that can’t be learned with an RMS error below the threshold within maximum epochs.

In test phase, we compared the output pattern of each neural network with the test pattern that was extracted from the 3rd natural speech data, and computed the similarity between these two patterns to get the estimation rates.

The estimation rate of pitch neural network was about 92% in training phase, and about 90% in test phase and that of energy neural network was about 90% in training phase, and about 89% in test phase. They were obtained on the condition...
that the number of input nodes of BP network was set to 11, and the number of output node was set to 4 and the number of hidden layer was set to 1.

In the case of 3rd order polynomial equation, we could see about 1% increase in the estimation rates, but it was not enough when we considered the computational load.

4. Discussion

When the number of phoneme strings was set to 11, the supra-segmental effect and the effect of accent in a word can be covered, but if the number of phoneme strings in a prosodic phrase will exceed 11, then the ANNs cannot fully learn the effect of phoneme's relative position in a prosodic phrase. But if we increase the number of input modules, output modules and hidden layer of ANN, then the computational load will dramatically increase.

The estimation rate of pitch neural network was about 92% in training phase, and about 90% in test phase and that of energy network was about 90% in training phase, and about 89% in test phase. They were obtained on the condition of 2nd order polynomial equation. In the case of 3rd order polynomial equation, we could see about 1% increase in the estimation rates, but it was not enough when we considered the computational load.

When we estimated the duration of each phoneme from speech DB by autocorrelation method, the resolution of the analysis frame was set to 11.61 m sec. If we set the overlapping interval to 64 instead of 128 samples, we could get more accurate data about the duration of each phoneme. It will also need more computational load.

In this study we trained the ANNs with prosody patterns that extracted from a corpus which consisted of 100 sentences, we think it is not large enough to get a good performance of ANNs. If we can expand the size of corpus, and increase the number of speaking for the same sentences from 3 to 10, we can train the ANNs with multiple sets of target patterns and test them with multiple sets of test patterns and we don't have to be trapped into over-learning.

5. Conclusions

In this study we propose ANNs that can learn the prosodic rules in a Korean sentence. MLP neural networks using an error BP algorithm had been selected as ANNs for this study.

To train and test these ANNs, we made a corpus. The sentences of this corpus were read by one male speaker three times in a row, recorded, and collected as a speech DB. We had analyzed this speech DB to extract prosodic information of each phoneme, and built up target and test patterns for ANNs. From this speech DB, we built up 2 prosody DB using curve fitting method.

In case of 2nd order polynomial equation, the estimation rate of pitch neural network was about 92% in training phase, and about 90% in test phase and that of energy network was about 90% in training phase, and about 89% in test phase. The estimation rates of ANNs with 3rd order polynomial increased about 1% compared to those of 2nd order polynomial.

6. References


