AN APPLICATION OF RECURRENT NEURAL NETWORKS TO LOW BIT RATE SPEECH CODING

Minoru KOHATA

Faculty of Engineering Tohoku University
Sendai, 980-77, Japan
e-mail koha@commu.ecei.tohoku.ac.jp

ABSTRACT

It is well known that the LSP coefficient which represents the speech spectrum envelope as one of the linear prediction coefficients, shows a good performance of spectral interpolation along the time axis, but it is also known that the duration of interpolation is limited up to 20 $\sim$ 30 ms.

This limitation makes it difficult to reduce the bit rate in very low bit rate speech coding. To resolve this problem, recurrent neural networks (RNN) were applied to interpolate LSP coefficients, and it was possible to increase the duration of interpolation to about 100 ms without so much degradation of the synthesized speech quality.

1. INTRODUCTION

An RNN has asymmetrical weighting coefficients among the units, and a time delay at each unit. Thus, an RNN is expected to be able to memorize or restore time-varying patterns. In this paper, a new spectral interpolation method using RNNs is proposed as an application of an RNN to speech coding; its training method is also proposed.

It is well known that speech spectral patterns are represented by linear predictive coefficients. Among some types of linear predictive coefficients, the LSP coefficients show a good performance of spectral interpolation along the time axis. In most cases, linear interpolation has been used because of its convenience, but the duration of interpolation is limited up to 20 $\sim$ 30 ms.

Thus, we introduce a new interpolation method using RNNs which enables interpolation of LSP coefficients with long time duration (about 100 ms). This method can be applied to very low bit rate speech spectral coding directly. In the experiments in this paper, LSP coefficients were sparsely sampled (segmented), coded, and interpolated by various interpolation methods including the proposed method. These methods were compared based on the spectral distortion in the restored LSP coefficients.

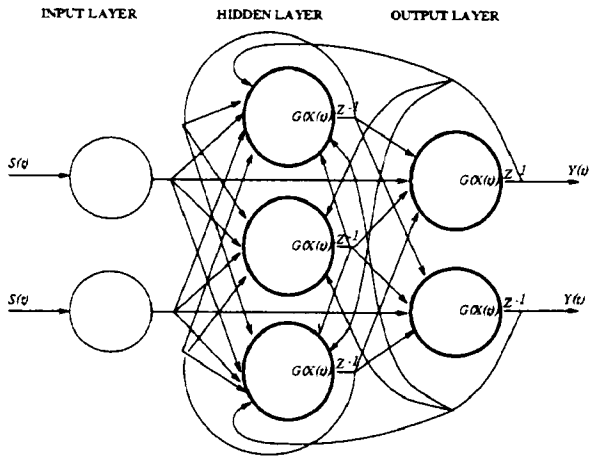

Figure 1: Recurrent neural network.

2. RNN INTERPOLATOR AND ITS TRAINING METHOD

2.1. RNN interpolator

The RNN used in this paper is an asymmetrically and fully connected type, and each unit constituting the RNN has a time delay element at the output. Figure 1 shows the structure of the RNN. The operation of the RNN is denoted by Eq.(1),

$$\begin{aligned} Y_i(t) &= G_i(X_i(t)) \\ X_i(t+1) &= \sum_j^O W_{ij} Y_j(t) + \sum_k^I V_{ik} S_k(t) \\ & \quad (i = 1, \ldots, N) \end{aligned} \tag{1}$$

where $X_i(t)$, $Y_i(t)$, and $S_i(t)$ denote the ith unit's potential, output, and input, respectively. G_i is an output function, and $W_{ij}(t)$ and $V_{ik}(t)$ are weighting coefficients.

The signal to be interpolated is sparsely sampled, and its val-

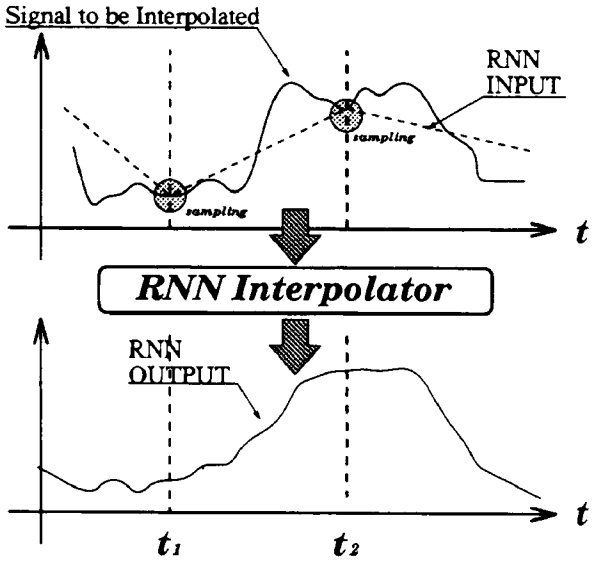

Figure 2: Interpolation by RNN.

ues at the sampling time are linearly interpolated, resulting in the input signal to the RNN. This input signal is shown in Fig.2 as a dotted line and expressed by Eq.(2),

$$S_i(t) = \frac{(Q_i(t_2) - Q_i(t_1))}{(t_2 - t_1)}(t - t_1) + Q_i(t_1) \qquad (2)$$

If the RNN is trained by the proposed method as explained in the following section, the signal is restored so that it is nearly the same as the original signal, by applying $S_i(t)$ of the Eq.(2). The interpolator using an RNN is termed an RNN interpolator.

The interpolation method using RNN has the following advantages.

(1) The RNN interpolator is able to interpolate sparsely sampled coefficients with lower distortion compared with linear or spline interpolation.

(2) After the training of the RNN, sub-optimal interpolation to an original signal can be made without statistical or mathematical modeling of the original signal.

2.2. Training of RNN

In order for an RNN to operate as an interpolator, some training method must be used so as to make the RNN output closer to the original input signal. The RNN interpolator must interpolate multiple input patterns, and so the training method must make an RNN interpolator whose interpolation error is sufficiently small for multiple input patterns. As a method of training an RNN for only one teacher pattern, the BPTT (Back Propagation Through Time) method [1][2] is often used, which is a simple extension of the ordinary BP method to the time domain.

In the BPTT method, the weighting coefficients between the units constituting the RNN is adjusted so as to minimize the square error of Eq.(3),

$$E = \frac{1}{2}\int_{t_1}^{t_2} \sum_{i \in O} (Y_i(t) - Q_i(t))^2 \, dt \qquad (3)$$

where $Q_i(t)$ represents the teacher signal and O means the set of output units.

Adjustment of the weighting coefficients is done by the steepest descent method; its amount, ΔW_{ij} is represented by Eq.(4),

$$\Delta W_{ij} = -\eta \sum_{t=t_1}^{t_2} P_i(t) Y_j(t) \qquad (4)$$

In Eq.(4), $P_i(t)$ is "the back propagation value through time," which is represented by Eq.(5).

$$\begin{aligned} P_i(t-1) &= \sum_j P_j(t) W_{ji} G_i'(X_i(t)) \\ &\quad + (Y_i(t) - Q_i(t)) G_i'(X_i(t)) \; (i \in O) \\ &= \sum_j P_j(t) W_{ji} G_i'(X_i(t)) \quad (i \notin O) \end{aligned} \qquad (5)$$

By using the BPTT method, a single RNN can be trained, the error of which is locally minimized for a single pattern [5]. However an RNN for multiple teacher patterns cannot trained using this method, because the error does not always converge. Thus in this paper, a new algorithm which can train plural RNNs for multiple teacher patterns is proposed. This method is termed the BPSS (Bach Propagation with Selective Study) method.

2.3. BPSS method

The BPSS method incorpolates the BPTT method, and a pre-clustering of teacher signals is adopted before commencement of BPTT to avoid explosion of the error and to achieve lower interpolation error. The algorithm of the BPSS is shown in Fig. 3, and explained below.

(1) Firstly, choose a teacher signal randomly from the entire of teacher signals defined by Q, and make it an element of the set Q^+.

(2) Execute BPTT training on an RNN using teacher signals in Q^+. Adjustment of weighting is done once by

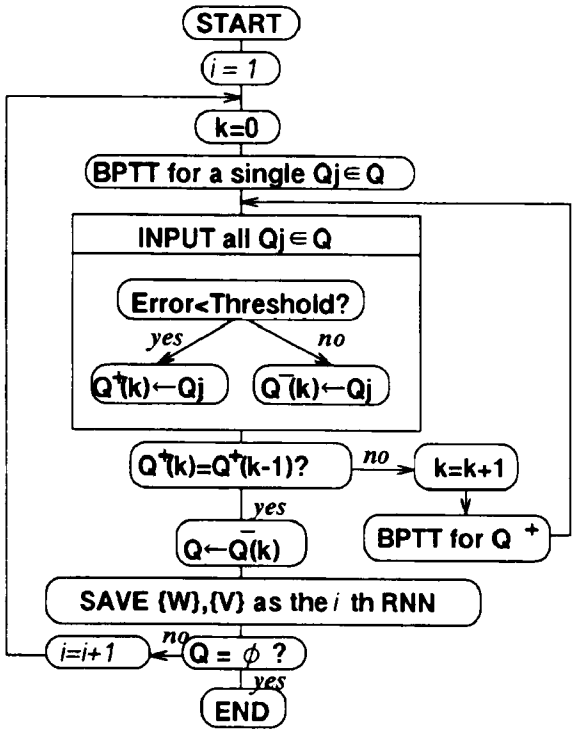

Figure 3: BPSS training method.

one teacher signal sequentially. As the initial condition, the potentials of hidden units are reset to zero and the outputs are set to the first values of the teacher signal.

(3) Make input signals corresponding to all the teacher signals in Q and input them to the above-trained RNN. Observe the interpolation errors and select teacher signals whose interpolation error is lower than a threshold value, and replace the element of Q^+ by them. The teacher signals which are not selected here makes the set Q^-, that is,

$$Q = Q^+ \bigcap Q^-. \tag{6}$$

(4) Repeat steps (2) and (3) until the elements of Q^+ are fixed, or the number of repetition reaches the limit. Then, save the weighting coefficients, W_{ij} and V_{ik}, for the RNN interpolator corresponding to Q^+.

(5) Replace Q by Q^-, prepare a new RNN, and repeat (1) $\sim$ (4) until Q becomes empty or the number of RNN interpolators reaches the limit.

In the proposed method, an error explosion in BPTT study is prevented by clustering teacher signals. The teacher signals in the cluster can be interpolated by a single RNN interpolator with lower interpolation error than the threshold. So, the interpolation errors and the number of created RNNs depend on the threshold and it must be determined experimentally [3].

In the interpolation process, an input pattern is applied to all RNNs to find the best RNN interpolator, the output of which is the signal closest to the original one, by full search.

3. INTERPOLATION AND CODING OF LSP

3.1. Segmentation

In this subsection, the method of making teacher and input signals for the RNN interpolator from a pattern of LSP coefficients which continuously varies with time is explained.

(1) Calculate the dynamic scale defined by Eq.(7),

$$D(t) = 10log_{10} \sum_{i=1}^{p} a_i^2(t) \tag{7}$$

where $a_i(t)$ is the local gradient of LSP along time axis, and p is the order of the LSP analysis.

(2) Search for the local peaks of $D(t)$, and make them $D(t_{p_1})$, $D(t_{p_2})$, $D(t_{p_3})\ldots$.

(3) Search for the time when $D(t)$ becomes minimum between the times corresponding to $D(t_{p_1})$, $D(t_{p_2})$, $D(t_{p_3})\ldots$, and denote it as t_{s_j}. That is,

$$t_{s_j} = argmin(D(t)|t_{p_j} < t < t_{p_{j+1}}). \tag{8}$$

(4) Chop up the LSP coefficients at the time t_{s_j} and make teacher signals and input signals.

(5) Normalize the lengths of the teacher and input signals, and subtract the mean values for each of the LSP orders. These processes are required to improve the performance of the RNN interpolator.

This segmentation method chops up the LSP coefficients into patterns where they are stationary near the top of the segment, transient in the middle frames, and stationary near the bottom. The reason why this method is adopted is that the RNN interpolator shows a better performance near the top and the bottom of the segments, and it can interpolate stationary periods of speech with lower distortion. This results in good subjective quality of the synthesized speech.

3.2. Experiments of coding

Finally, the coding of LSP coefficients was carried out by using the proposed interpolation method. For comparison, linear interpolation, spline interpolation, and segment quantization (SQ) were applied and the spectral distortions were compared with that of the RNN interpolator. In order to interpolate LSP coefficients at a receiver, the following values must be coded and transmitted: the LSP coefficients in the

Table 1: Bit allocation.

	LSP order		
	1~3	4~6	7~10
LINEAR	9bit	9bit	7bit
SPLINE	9bit	9bit	7bit
SQ	9bit	9bit	7bit
RNN	7bit	7bit	6bit

RNN Index = 5bit, Signal length = 7bit

Table 2: Experimental condition.

Sampling freq.	8kHz
LSP analysis window	32ms
LSP analysis period	4ms
LSP order	10
RNN input/output units	10
RNN hidden units	30
Output function	tanh

Table 3: Spectral distortion.

	CD[dB]	bit rate
LINEAR	3.855	287bps
SPLINE	4.458	287bps
SQ	3.653	287bps
RNN	3.365	287bps

top or the bottom frame of the input signal, the index of the RNN interpolator to be used, and the length of the original signal. The bit allocation to these parameters are listed in Table 1. For the length of the original signal, 7 bits were allocated in all methods. The LSP coefficients were split into three vectors and coded by split-VQ [4]. The RNN interpolation requires additional information about which RNN is to be used. In order to share 5 bits for this and to make the total bits equal in all the methods, the bits for the split-VQ were decreased in the proposed method. The other experimental conditions are shown in Table 2.

Then the spectral distortions of the LSP coefficients after coding and interpolation were measured and compared. The final results are shown in Table 3. This test was carried out for the data outside the training data. The RNN interpolation method achieved lower spectral distortion than the other methods by about $0.5dB$ and the mean duration of interpolation was more than $100ms$.

The bit rate shown in Table 3 is only for coding the LSP coefficients, which does not include the residual information. Using the restored LSP coefficients, speech signals were synthesized where the raw residual signals were used as the excitation signal. An informal listening test showed a perceived improvement of quality in comparison with the other methods although there was some degradation compared with the original signal.

4. CONCLUSION

An RNN interpolator was newly proposed with its training method for multiple input patterns. Then it was applied to the interpolation of spasely sampled LSP coefficients, where the sampling period was more than $100ms$. Simulation of coding was carried out, and the proposed method outperformed the other interpolation methods, below the bit rate of $300bps$.

The LSP coefficients interpolated by the RNN interpolators were well restored and showed a lower spectral distortion than the other interpolation methods. The training method introduced here is a sub-optimal method, and it may be possible to slightly improve the obtained RNNs' interpolation error.

5. REFERENCES

1. M.Sato, "A learning algorithm to spatiotemporal patterns to recurrent neural networks," Biol. Cybern., Vol.62, pp.259–263, 1990.
2. B.A.Pearlmutter, "Learning state space trajectories in recurrent neural networks," Neural Computation, Vol.1, pp.263-269, 1989.
3. S.Geman and D.Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." IEEE Trans. Patt. Ana. Mach. Int, Vol.PAMI-6, no.6, pp.721–741, Nov. 1984.
4. R.M.Gray, "Vector Quantization," IEEE ASSP Magazine, Vol.1, no.2, Apr. 1984.
5. K.Funahashi, "On the Approximate Realization of Continuous Mappings by Neural Networks," Neural Networks, Vol.2, pp.183–192, 1989.