Hindawi
Scientific Programming
Volume 2021, Article ID 1704995, 10 pages
https://doi.org/10.1155/2021/1704995

Research Article

Automatic Synthesis Technology of Music Teaching Melodies Based on Recurrent Neural Network

Yingxue Zhang[1] **and Zhe Li**[2]

[1]*Academy of Music, Hubei Engineering University, Xiaogan 432100, China*
[2]*Economic and Management, Hubei Engineering University, Xiaogan 432100, China*

Correspondence should be addressed to Zhe Li; lizhe_lz@hbeu.edu.cn

Received 27 September 2021; Revised 15 November 2021; Accepted 19 November 2021; Published 9 December 2021

Academic Editor: Ahmed Farouk

Computer music creation boasts broad application prospects. It generally relies on artificial intelligence (AI) and machine learning (ML) to generate the music score that matches the original mono-symbol score model or memorize/recognize the rhythms and beats of the music. However, there are very few music melody synthesis models based on artificial neural networks (ANNs). Some ANN-based models cannot adapt to the transposition invariance of original rhythm training set. To overcome the defect, this paper tries to develop an automatic synthesis technology of music teaching melodies based on recurrent neural network (RNN). Firstly, a strategy was proposed to extract the acoustic features from music melody. Next, the sequence-sequence model was adopted to synthetize general music melodies. After that, an RNN was established to synthetize music melody with singing melody, such as to find the suitable singing segments for the music melody in teaching scenario. The RNN can synthetize music melody with a short delay solely based on static acoustic features, eliminating the need for dynamic features. The proposed model was proved valid through experiments.

1. Introduction

With the rapid development of modern computer science, many researchers have shifted their focus to computer-based algorithm composition or automatic music melody generation system. The research results on music melody synthesis and music modeling methods are being applied to various fields. The research of computer music creation aims to quantify and combine the emotional tendencies of music, with the aid of computer and mathematical algorithms. The specific tasks include aided composition, sound simulation and storage, and music analysis and creation [1, 2]. Computer music creation generally relies on artificial intelligence (AI) and machine learning (ML) to generate the music score that matches the original mono-symbol score model or memorize/recognize the rhythms and beats of the music. Despite its broad application prospects, the AI-based composition without needing lots of music knowledge rules is in the theoretical stage [3, 4].

Speech processing has been widely applied in composition and songwriting, record production, and entertainment. Unlike simple speech synthesis, music melody synthesis has two additional processing steps: tone detection and transformation [5, 6]. Wenner et al. [7] preprocessed the musical melody synthesis corpus through automatic note segmentation and voiced/unvoiced sound recognition, constructed a high-quality music melody synthesis system, and proposed a music melody adjustment algorithm, which functions as an adaptive filter capable of detecting musical note cycles.

AI has already been adopted to realize algorithm composition or automatic music generation [8–12]. Bilbao et al. [13] introduced bidirectional long short-term memory (LSTM) neural network to the mixed music generation system and thus realized the training of multi-voice music datasets. Their approach provides effective chord progressions while ensuring melody time and transposition invariance.

Electronic synthetic tones bring rich new sound experience to music of various styles and themes. Electronic musical instruments differ from traditional acoustic instruments in sound rendering principle and acoustic features [14–19]. Miranda et al. [20] expounded the computer-aided means to realize the acoustic features, voice editing, and modulation of electronic sound melodies and provided a valuable reference for applying electronic sound melodies in modern music creation. However, computer-based accompaniment has a rigid chord structure, which cannot easily adapt to diverse music styles. To solve the problem, Taigman et al. [21] put forward an adaptive automatic accompaniment algorithm, including chord series extraction and automatic accompaniment figure acquisition, and created a suitable accompaniment figure database based on chord sequences in the light of the features of music melodies and emotions.

The existing studies at home and abroad mostly concentrate on the methods, melodic forms, and tone synergy of computer music creation [22–26]. However, there are very few music melody synthesis models based on artificial neural networks (ANNs). Some ANN-based models cannot adapt to the transposition invariance of original rhythm training set. To overcome the defect, this paper attempts to develop an automatic synthesis technology of music teaching melodies based on recurrent neural network (RNN).

The remainder of this paper is organized as follows. Section 2 extracts the acoustic features from music melody. Section 3 applies the sequence-sequence model to synthetize general music melodies. Section 4 establishes an RNN to synthetize music melody with singing melody, aiming to find the suitable singing segments for the music melody in teaching scenario. Finally, experiments were carried out to prove the effectiveness of our model.

2. Acoustic Feature Extraction

The automatic synthesis of music melody aims to obtain a melody that is beautiful and pleasant to human ears. To describe the differences in the auditory sensitivity of human ears to music melodies of different frequencies, the linear frequency μ of each music melody was transformed based on mel scale frequency μ_{MR}:

$$\mu_{MR} = 2595 \times \log_{10}\left(1 + \frac{\mu}{700}\right). \tag{1}$$

Under the scale of mel scale frequency μ_{MR}, the multiples of the μ_{MR} difference between two music melodies are roughly equal to those of the tone difference perceived by human ears.

For the above reason, mel scale was adopted to extract acoustic features in our music melody synthesis system. Since the music melody signal in the high-frequency band is relatively weak, such a signal was compensated through preemphasis. Let β be the preemphasis factor. Then, the preemphasis of music melody $a(\tau)$ can be described by

$$b(\tau) = a(\tau) - \beta \cdot a(\tau - 1). \tag{2}$$

To prevent spectrum leakage and enhance the continuity of the left end and right ends of the signal frame, it is necessary to perform framing and windowing of the music melody signal with short-time stationarity features. Let $CH(m)$, $m = 0, 1, \ldots, M-1$, be the framed music melody signal and $CK(m)$ be the function of the Hamming window. Then, the windowed signal $CH^*(m)$ can be described by

$$CH^*(m) = CH(m) \times CK(m), \tag{3}$$

where $CK(m)$ can be described by

$$CK(m) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi m}{M-1}\right), & 0 \le m \le M-1, \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

It is difficult to extract the features from time-domain music melody signal. The general practice is to convert the signal to the frequency domain through short-time Fourier transform (STFT) before further analysis. Let $CH^*(m)$ be the input signal of STFT and M be the number of Fourier points. Then, the fast Fourier transform of the M points of the windowed framed time-domain music melody signal $CH^*(m)$ can be expressed as

$$CH(l) = \sum_{m=0}^{M-1} CH^*(m)\exp\left(-\frac{j2\pi lm}{M}\right), \quad 0 \le l \le M. \tag{5}$$

In the frequency domain, the absolute value of the spectrum of the music melody signal can be described as

$$PT(l) = |CH(l)|. \tag{6}$$

The human ears can only detect the frequency components in a certain range. Therefore, the human auditory system could be treated as a filter bank that only allows some frequency signals to pass through. This paper simulates the human auditory system with a mel filter bank. Let μ_n, $\mu_{\max}$, and $\mu_{\min}$ be the central, upper, and lower frequencies of the filter bank, respectively. Then, the transfer function of the filter bank can be described by

$$QF_n(l) = \begin{cases} 0, & k < \mu_{\min}, \\ \dfrac{l - \mu_{\min}}{\mu_n - \mu_{\min}}, & \mu_{\min} \le l \le \mu_n, \\ \dfrac{\mu_{\max} - l}{\mu_{\max} - \mu_n}, & \mu_n \le l \le \mu_{\max}, \\ 0, & l > \mu_{\max}. \end{cases} \tag{7}$$

Let N be the number of triangular filters. Then, the signal o_n passing through the mel filter bank $QF_n(l)$ can be expressed as

$$o_n = \sum_{l=\mu_{\min}}^{\mu_{\max}} PT(l)QF_n(l) \quad n = 0, 1, 2, \ldots, N-1. \tag{8}$$

The mel spectrum can be extracted by taking the logarithm of o_n.

3. Sequence-Sequence Model-Based Music Melody Synthesis

Both speech synthesis based on statistical parameters and music melody synthesis based on neural networks face the following defects: the complexity of model construction and the dependence of front-end text processing on texts with strong linguistics knowledge. Unlike these approaches, sequence-sequence model-based speech synthesis can directly transform phonetic notations to waveforms and significantly simplify the front-end module. Since a standard neural network cannot directly process input and variable sequences with variable length, a sequence-sequence model is needed to handle the scenario that input sequence is not equal to the output sequence.

This paper constructs a sequence-sequence model of music melody to realize the automatic synthesis of melody sequences. Figure 1 shows the structure of the music melody synthesis system based on the sequence-sequence model. In the proposed model, the music melody recognition module receives a music melody sequence and outputs a singing melody sequence. The synthesis module receives the target melody sequence and outputs an audio sequence.

The RNN model consists of an encoder and a decoder, using an activation function Γ. The hidden state g_τ at the current moment τ depends on the input g_τ at the current moment and the hidden state $g_{\tau-1}$ at the previous moment $\tau - 1$:

$$g_\tau = \Gamma(g_{\tau-1}, a_\tau). \tag{9}$$

Let $g_{\psi a}$ be the hidden layer state of the neural network at moment ψ_a and $s(^*)$ be the nonlinear transform. Then, the middle vector p of the encoder can be obtained through the nonlinear transform of each hidden layer states:

$$p = s\left(g_1, g_2, g_3, g_4 \ldots g_{\psi_a}\right). \tag{10}$$

The middle vector p is equivalent to the final encoded state of the hidden layer:

$$p = s\left(g_1, g_2, g_3, g_4 \ldots g_{\psi_a}\right) = g_{\psi_a}. \tag{11}$$

The next output b_i of the encoder can be generated based on the middle vector p and historical outputs b_1, b_2, …, b_{i-1}. The encoder is often adopted to predict the next acoustic feature in music melody composition. It is necessary to determine the middle vector p and the existing acoustic features:

$$b_i = \arg\max O(b_i) = \prod_{i=1}^{n} o(b_i | b_1, b_2, \ldots, b_{i-1}, p). \tag{12}$$

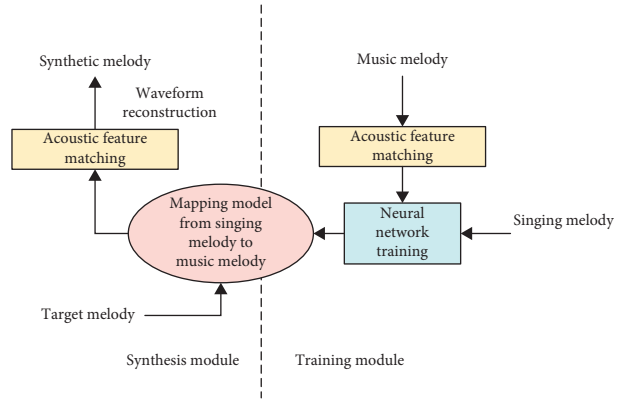

Figure 1: Music melody synthesis system based on the sequence-sequence model.

Let r_{i-1} be the state of a hidden layer node in the RNN of the decoder; b_{i-1} be the output of that node at the moment $i - 1$; and $s(^*)$ be the nonlinear transform. Then, formula (12) can be simplified as

$$b_i = s(b_{i-1}, r_{i-1}, p). \tag{13}$$

The sequence-sequence model is prone to a potential problem: the input sequence has the same influence on the weight of each element in the output sequence. This problem can be solved by the attention mechanism. Figure 2 shows the structure of the attention-based model.

The attention mechanism highlights that different nodes belong to different parts of the input sequence. Let g_v be the hidden layer output of the encoder at the moment v; $r_{\tau-1}$ be the hidden layer output of the encoder at the moment $\tau - 1$; and e be the alignment model. Then, the matching degree $DO_{\tau j}$ between the location of input layer nodes and output layer nodes can be calculated by

$$DO_{\tau v} = e(r_{\tau-1}, g_v), \tag{14}$$

where e is a nonlinear function to compare and compute the matching degree between g_v and $r_{\tau-1}$. The greater DO_τ is, the more necessary it is to emphasize the input sequence at the current moment during the decoding of music melody signal. Let U, V, and W be weight matrices. Then, the point multiplication, weighting, weight stitching, and sensing of the alignment model e can be, respectively, described by

$$\begin{aligned} e\left(r_{\tau-1}, g_j\right) &= r_{\tau-1}^{\psi} g_j, \\ e\left(r_{\tau-1}, g_j\right) &= r_{t-1}^{\psi} V g_j, \\ e\left(r_{\tau-1}, g_j\right) &= V\left[r_{\tau-1}^{\psi}; g_j\right], \\ e\left(r_{\tau-1}, g_j\right) &= U^{\psi} \ \tanh\ V r_{\tau-1} + W g_j. \end{aligned} \tag{15}$$

$DO_{\tau j}$ can be normalized by

$$\beta_{\tau j} = \frac{\exp\left(DO_{\tau j}\right)}{\sum_{l=1}^{\psi_a} \exp\left(DO_{\tau l}\right)}. \tag{16}$$

The middle vector p_τ can be obtained through weighted summation:

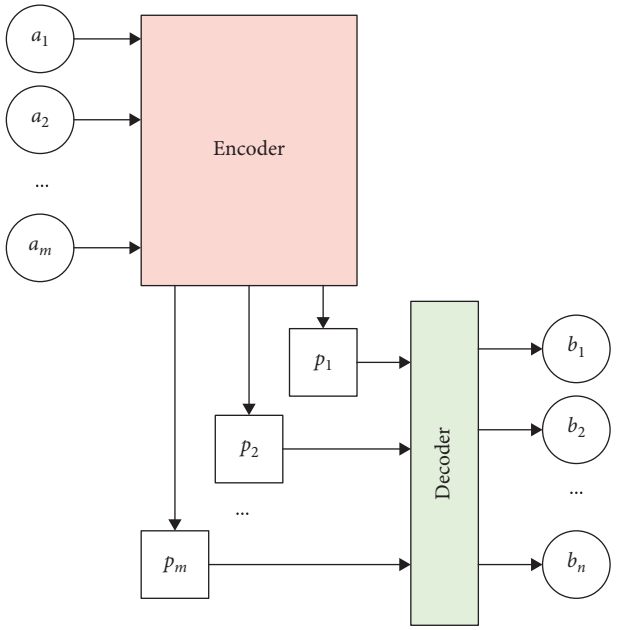

Figure 2: Attention-based model.

$$p_\tau = \sum_{j=1}^{\psi_a} \beta_{\tau j} g_j. \tag{17}$$

Let $s(*)$ be the nonlinear transform. Then, the next hidden layer state can be calculated by

$$r_\tau = s\left(r_{\tau-1}, b_{\tau-1}, p_\tau\right). \tag{18}$$

To realize frame-level feature mapping from the input to the output, the proposed model needs to transform the contextual features on the phoneme level and the frame level. This section puts forward a length prediction model for music melody, which supports the time supervision labeling. Suppose a phoneme sequence is given for a rhythm in a music melody sequence of the length M. Let ε be the model parameter of the RNN; w be the rhythm state sequence of the melody; and L_m be the number of rhythm states of the melody. Then, the length prediction of music melody states can be regarded as the forecast of the length of the state allocation sequence. The prediction goal is to maximize the likelihood in the following formula:

$$\log O(w|\varepsilon, \psi) = \sum_{m=1}^{M} \sum_{l=1}^{L_m} \log o_{m,l}\left(\delta_{m,l}\right). \tag{19}$$

Let $o_{m,l}(\delta_{m,\,l})$ be the probability density function of the length model of the music melody; $\delta_{m,l}$ be the time of the l-th state of the m-th phoneme; Ψ be the total length constraint; $n_{m,l}$ be the length of rhythm state predicted by network model; and $\varphi_{m,l}$ be the variance obtained from the individual length of each phoneme in the music melody database. Solving the maximum likelihood of formula (19), the rhythm state length of each melody can be obtained by

$$\delta_{m,l} = n_{m,l} + \sigma \cdot \phi_{m,l}^2. \tag{20}$$

Let ψ_S be the phoneme length specified in the music score. Then, σ can be calculated by

$$\sigma = \frac{\left(\psi - \sum_{m=1}^{M} \sum_{l=1}^{L_m} n_{m,l}\right)}{\sum_{m=1}^{M} \sum_{l=1}^{L_m} \phi_{m,l}^2}. \tag{21}$$

4. RNN-Based Melody Synthesis

To find the suitable singing segments for the music melody in the teaching scenario, the RNN-based statistical music melody synthesis algorithm needs to realize the following goal in the synthesis phase: identifying the most possible acoustic feature sequence $\tilde{u}$ from the signing melody sequence k with given linguistic features and a series of trained context-dependent music melodies Φ. Then, we have

$$\begin{aligned}
\tilde{u} &= \arg\max_u o\left(u|k, \tilde{\Phi}\right) \\
&= \arg\max_u \sum_{\forall w} \chi\left(u, w|k, \tilde{\Phi}\right) \\
&= \arg\max_{u,w} \chi\left(u, w|k, \tilde{\Phi}\right) \\
&= \arg\max_{u,w} \chi\left(u, w|k, \tilde{\Phi}\right) \Delta\left(w|k, \tilde{\Phi}\right) \\
&= \arg\max_u \chi\left(u|\hat{w}, \tilde{\Phi}\right),
\end{aligned} \tag{22}$$

where $\hat{w}$ is the rhythm state of a melody:

$$\hat{w} = \arg\max_w \Delta\left(w|k, \tilde{\Phi}\right). \tag{23}$$

Let $\lambda_{\hat{w}i}$ be the mean vector under state $\hat{w}_i$; $\sum_{\hat{w}i}$ be the corresponding covariance matrix; and $\lambda_{\widehat{w}i} = \lambda_{\widehat{w}1}^T, \ldots, \lambda_{\widehat{w}Q}^T$ and $\Sigma_{\widehat{w}} = \Sigma_{\widehat{w}1}^T, \ldots, \Sigma_{\widehat{w}Q}^T$ be the mean vector and covariance matrix under the given state w of singing melody sentence, respectively. If the output probability of the neural network obeys single Gaussian distribution, then formula (25) can be rewritten as

$$\begin{aligned}
\tilde{u} &= \arg\max_u o\left(u_\tau|\tilde{w}_\tau, \Phi\right) \\
&= \arg\max_u \prod_{\tau=1}^{Q} M\left(u_t; \lambda_{\widehat{w}_\tau}, \sum_{\widehat{w}_\tau}\right) \\
&= \arg\max_u \Re\left(u_\tau; \lambda_{\widehat{w}}, \sum_{\widehat{w}}\right) \\
&= \lambda_{\widehat{w}}.
\end{aligned} \tag{24}$$

From the statistical features of output probability (Figure 3), it can be learned that $\lambda_{\hat{w}i}$ is a jump series because the rhythm states of a melody are discrete and independent. The music melody signal reconstructed from $\lambda_{\hat{w}i}$ has a discontinuous boundary of rhythm states. To solve the problem, this paper introduces an observation vector u, which covers the static acoustic feature and its first- and second-order derivatives with respect to time:

$$u_\tau = \left[\theta_\tau^T, \Delta\theta_\tau^T\right]^T. \tag{25}$$

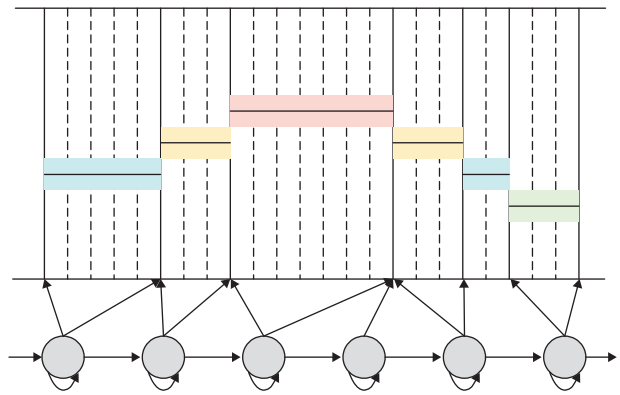

Figure 3: Statistical features of output probability.

Let SC be the sparse coefficient matrix. Then, the relationship between the observation vector sequence $u = [u_1^T, \ldots, u_Q^T]$ and the acoustic eigenvector sequence $\theta = [\theta_1^T, \ldots, \theta_Q^T]$ can be described by

$$\begin{bmatrix} u_{\tau-1} \\ u_\tau \\ u_{\tau+1} \end{bmatrix} \begin{bmatrix} \vdots \\ \theta_{\tau-1} \\ d\theta_{\tau-1} \\ d_\tau \\ d\theta_\tau \\ d_{\tau+1} \\ d\theta_{\tau+1} \\ \vdots \end{bmatrix} = SC = \begin{bmatrix} \cdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ \cdots & 0 & I & 0 & 0 & \cdots \\ \cdots & -1/2I & 0 & 1/2I & 0 & \cdots \\ \cdots & 0 & 0 & I & 0 & \cdots \\ \cdots & 0 & -1/2I & 0 & 1/2I & \cdots \\ \cdots & 0 & 0 & 0 & I & \cdots \\ \cdots & 0 & 0 & -1/2I & 0 & \cdots \\ \cdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \theta_{\tau-2} \\ \theta_{\tau-1} \\ \theta_\tau \\ \theta_{\tau+1} \\ \vdots \end{bmatrix}. \tag{26}$$

Combining formulas (24) and (26):

$$\begin{aligned} \widetilde{u} &= \arg\max_u \Re\left(u_\tau; \lambda_{\widehat{w}}, \sum_{\widehat{w}}\right) \\ &= SC \cdot \theta. \end{aligned} \tag{27}$$

The maximization of the output probability is equivalent to finding the maximum of θ:

$$\begin{aligned} \widetilde{\theta} &= \arg\max_\theta \Re\left(SC \cdot \theta; \lambda_{\widehat{w}}, \sum_{\widehat{w}}\right) \\ &= \arg\max_\theta \log\ \Re\left(SC \cdot \theta; \lambda_{\widehat{w}}, \sum_{\widehat{w}}\right). \end{aligned} \tag{28}$$

Find the partial derivative of θ in formula (28):

$$\frac{\partial \log\ \Re\left(SC \cdot \theta; \lambda_{\widehat{w}}, \sum_{\widehat{w}}\right)}{\partial \theta} \propto \frac{\partial}{\partial \theta}\left(SC \cdot \theta - \lambda_{\widehat{w}}\right)^T \sum_{\widehat{w}}^{-1}\left(SC \cdot \theta - \lambda_{\widehat{w}}\right) = SC^T \sum_{\widehat{w}}^{-1} SC \cdot \theta - SC^T \sum_{\widehat{w}}^{-1} \lambda_{\widehat{w}}. \tag{29}$$

Make formula (29) equal to 0, and a linear equation about θ can be obtained:

$$SC^T \sum_{\widehat{w}}^{-1} SC \cdot \theta = SC^T \sum_{\widehat{w}}^{-1} \lambda_{\widehat{w}}. \tag{30}$$

Unlike other deep neural networks, the RNN combines the output of the input layer and the output of the hidden layer at the previous moment into the input of the hidden layer. Therefore, the network can capture the dynamic law of sequential music melodies from the periodic connections between hidden layer nodes. Let ω_{ga}, ω_{bg}, and ω_{gg} be the weight matrices of input layer-hidden layer, hidden layer-output layer, and hidden layer-hidden layer, respectively; γ_g and γ_b be the bias vectors of hidden layer and output layer, respectively; $G(.)$ be the activation function between hidden layers; and $\{a_\tau\}\psi_{\tau=1}$, $\{g_\tau\}_{\tau|=1}^{\psi}$, and $\{b_\tau\}^{\psi}{}_{\tau=1}$ be the input music melody features, hidden layer sequence, and output features, respectively. Then, g_τ can be expressed as

$$g_\tau = G\left(\omega_{ga}\tau + \omega_{gg}g_{\tau-1} + \gamma_g\right). \tag{31}$$

Besides, b_τ can be given by

$$b_\tau = \omega_{bg}g_\tau + \gamma_b. \tag{32}$$

For the traditional RNN, vanishing gradient problem might occur during network training, owing to the use of backpropagation algorithm. To prevent this problem, the LSTM, a time RNN model, was adopted to synthetize the teaching audios including both singing melody and music melody. As shown in Figure 4, an LSTM unit contains an input gate IG_τ, a forget gate FG_τ, and an output gate OG_τ, as well as a memory cell MC_τ. Let g_τ and a_τ be the hidden layer output and input signal of the network at the moment τ, respectively; $\omega_I{}^*$ and $\omega_{XH}{}^*$ be the weight matrices of input layer nodes and hidden layer nodes, respectively; $C_{PH}{}^*$ and γ^* be the weight and bias, respectively; and $\circ$ be the Hadamard product of the elements of a matrix. Then, the operations of the input gate IG_τ, the forget gate FG_τ, the memory cell MC_τ, and the output gate OG_τ can be, respectively, expressed as

$$\begin{aligned}
IG_\tau &= \text{sigmoid}\left(\omega_I^{IG}a_\tau + \omega_{XH}^{IG}g_{\tau-1} + C_{PH}^{IG}\circ MC_{\tau-1} + \gamma^{IG}\right),\\
FG_\tau &= \text{sigmoid}\left(\omega_I^{FG}a_\tau + \omega_{XH}^{FG}g_{\tau-1} + C_{PH}^{FG}\circ MC_{\tau-1} + \gamma^{FG}\right),\\
MC_\tau &= FG_\tau\circ MC_{\tau-1} + IG_\tau\circ\ \text{Tanh}\left(\omega_I^{MC}a_\tau + \omega_{XH}^{MC}g_{\tau-1} + \gamma^{MC}\right),\\
OG_\tau &= \text{sigmoid}\left(\omega_I^{OG}a_\tau + \omega_{XH}^{OG}g_{\tau-1} + C_{PH}^{OG}\circ MC_{\tau-1} + \gamma^{OG}\right).
\end{aligned} \tag{33}$$

Then, g_τ can be calculated by

$$g_\tau = OG_\tau\circ\ \text{Tanh}\left(MC_\tau\right). \tag{34}$$

To reduce the delay of singing melody relative to music melody, the LSTM was adopted to build the acoustic model, and recurrent output layers were configured to further smoothen the acoustic features between adjacent frames. Figure 5 shows the framework of the low-delay synthesis model of singing melody relative to music melody. Let ω_{bb} be the weight matrix of the recurrent connections of the output layer extended from the traditional RNN. Then, we have

$$b_\tau = \omega_{bg}g_\tau + \omega_{bb}b_{\tau-1} + \gamma_b. \tag{35}$$

Dynamic features are needed to smoothen the parameter trajectories. It is possible to obtain smooth parameter trajectories by smoothing the acoustic parameters with recurrent output layers. The LSTM-based recurrent output layers receive the activation g_τ of the hidden layer and the output $b_{\tau-1}$ at the moment $\tau-1$, process them with the activation function and the input gate operation, and save some of the information to the state of the memory cell:

$$IG_\tau = \text{sigmoid}\left(\omega_I^{IG}g_\tau + \omega_{XH}^{IG}b_{\tau-1} + C_{PH}^{IG}\circ MC_{\tau-1} + \gamma^{IG}\right). \tag{36}$$

The state MC_τ of the memory cell at time τ can be obtained through forget gate operation and scrapping some useless information:

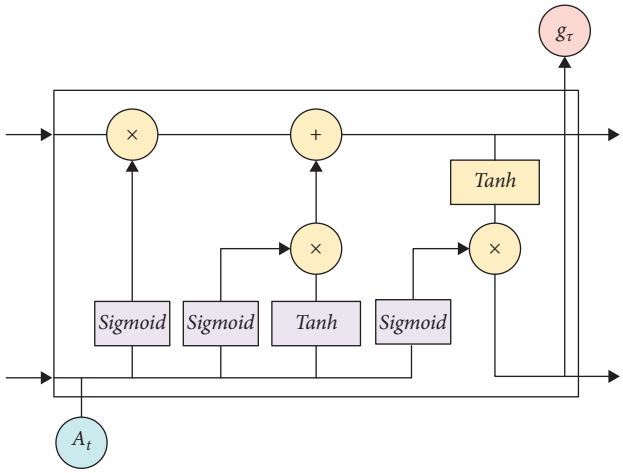

Figure 4: Structure of an LSTM unit.

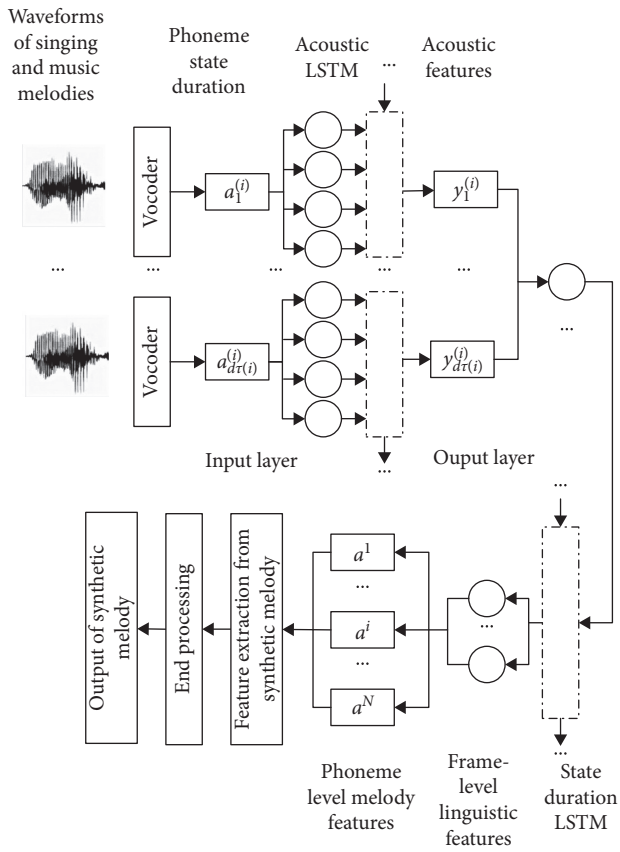

Figure 5: Low-delay synthesis framework of singing melody relative to music melody.

$$\begin{aligned}
FG_\tau &= \text{sigmoid}\left(\omega_I^{FG}g_\tau + \omega_{XH}^{FG}b_{\tau-1} + C_{PH}^{FG}\circ MC_{\tau-1} + \gamma^{FG}\right),\\
MC_\tau &= FG_\tau\circ MC_{\tau-1} + IG_\tau\circ\left(\omega_I^{MC}g_\tau + \omega_{XH}^{MC}b_{\tau-1} + \gamma^{MC}\right).
\end{aligned} \tag{37}$$

Finally, the network output b_τ at time τ can be obtained through output gate operation of MC_τ:

$$OG_\tau = \text{sigmoid}\left(\omega_I^{OG} g_\tau + \omega_{XH}^{OG} b_{\tau-1} + C_{PH}^{OG} \circ MC_{\tau-1} + \gamma^{OG}\right),$$
$$b_\tau = OG_\tau \circ MC_\tau. \tag{38}$$

5. Experiments and Result Analysis

To compare the convergence of different music melody synthesis models, Figure 6 shows the trend of the loss function value on the verification set of four models: DCNN based on static acoustic feature, LSTM, LSTM-RNN, and our model. The value of the loss function is the difference of model output and the actual value. Extended from AlexNet, the DCNN boasts deep layers and numerous parameters and has been widely applied to signal recognition and image processing. As an RNN, the LSTM is suitable for processing and predicting important events with relatively long intervals and delays in the time series. The network has been adopted in many scientific fields, namely, language learning and translation, robotic control, image analysis, document summarization, speech and image recognition, handwriting recognition, chatbot control, disease, click rate and stock prediction, and music synthesis.

The loss of each model continued to decrease with the growing number of iterations and eventually converged. After convergence, the DCNN had the greatest loss, the LSTM and LSTM-RNN had similar losses, and the proposed low-delay LSTM-LSTM realized the smallest loss.

To compare the music melody generated by our model with the original music melody, forty segments of music melodies were randomly selected from a test set containing 577 segments. The music melodies synthetized by different networks were objectively measured by four metrics: BAP distortion, F0 RMSE, LE, and MCD. The results in Table 1 show that MCD had the greatest influence on the synthetic music melodies. Overall, our model, which further smooths the acoustic features between adjacent frames with recurrent outputs, outperformed other networks, as evidenced by the small gaps of the four metrics. Therefore, our model is highly robust in finding the suitable singing segments for teaching.

Table 2 presents the errors of different model configurations in predicting phoneme length. It can be seen that the optimal configuration is our model with four layers, whose RMSE was 5.19 and the sum of RMSE and cross entropy was 4.18. Besides, our model had better phoneme synthesis effect than DCNN and LSTM, a sign of superiority in the modeling of music melody sequence. Based on RMSE + cross entropy, the least mean square (LMS) of the synthetic melody can be predicted, in order to effectively reduce the RMSE of phoneme length prediction. However, the predicted phoneme length might deviate from the actual length of the music melody. It is important to apply a constraint on the phoneme length of the two melodies. Table 3 shows the errors in the phoneme length before and after applying the constraint.

As shown in Table 3, before the constraint was applied, the predicted phoneme length was inconsistent with the given value. After applying the constraint, the mean error between the predicted value and the given value dropped. This proves the reasonability of introducing the constraint on the phoneme level.

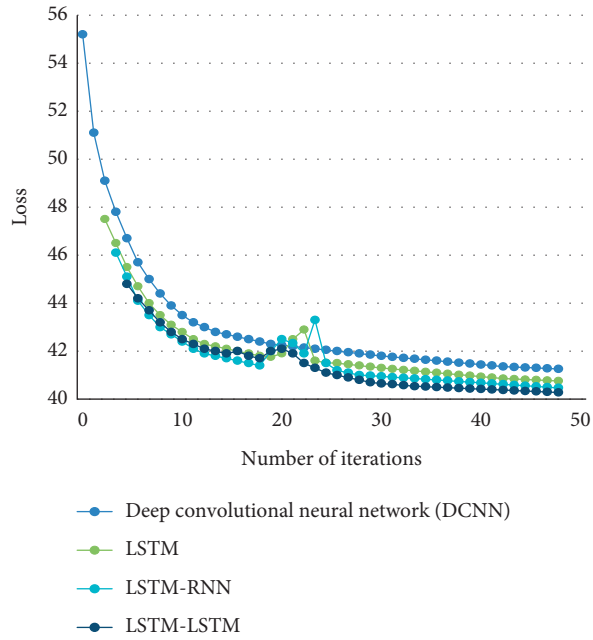

Figure 6: Convergence of different network models.

Table 4 presents the prediction errors of acoustic features of different model configurations on the test set. As shown in Tables 3 and 4, our model achieved a lower prediction error of the acoustic features of the test set than LSTM and DNN, during the synthesis of singing and music melodies. This means that our model can establish a good time series dependence and thus achieve an ideal synthesis effect.

For the proposed low-delay synthesis model of singing melody relative to music melody, it is important to evaluate the influence of the decoding consistency between singing melody and music melody on the modeling accuracy of acoustic parameters in the synthetic melody. For this purpose, a contrastive experiment was designed to compare the melody generated from natural music melody and that generated from score notes, under different lengths of historical access points (HAPs).

Table 5 shows the F0 values under different HAP lengths. As shown in Table 5, the F0 RMSE and F0 Pearson correlation did not change with the utilization rate of historical frames and remained independent of the type of source melody (natural melody or score notes). In addition, the HAP length had a limited influence on F0 Pearson correlation and LE.

Next, the mean length of the sliding window was set to 5, 10, 15, and 20 frames in turn for the end processing module. After verification and optimization, it was found that the window of 15 frames led to the best experimental results. Table 6 presents the prediction errors of different models in fundamental frequency and spectrum. Taking the melody generated from score notes for reference, the melody synthetized by our model had a lower F0 RMSE and a higher F0 Pearson correlation than that obtained by DCNN and LSTM, that is, our model can find the

Table 1: Results before and after the addition of singing melodies.

		BAP distortion	F0 root mean square error (RMSE)	Labeling error of voiced/nonvoiced sound (LE)	Mel cepstral distance (MCD)
DCNN	Preaddition	0.248	12.375	5.459	5.135
	Postaddition	0.239	11.892	5.374	4.913
LSTM	Preaddition	0.241	12.841	5.621	5.092
	Postaddition	0.244	12.163	5.548	4.836
Our model	Preaddition	0.246	12.715	5.539	5.051
	Postaddition	0.248	12.734	5.568	4.993

Table 2: Errors of different model configurations in predicting phoneme length.

	RMSE			RMSE + cross entropy		
Number of layers	DCNN	LSTM	Our model	DCNN	LSTM	Our model
1	6.35	5.81	**5.82**	5.31	4.82	**4.76**
2	5.93	5.46	**5.48**	5.13	4.51	**4.34**
3	5.74	5.23	**5.24**	4.86	4.35	**4.23**
4	5.89	5.65	**5.19**	4.92	4.53	**4.18**
5	5.82	5.67	**5.81**	4.89	4.59	**4.47**

Table 3: Errors in the phoneme length before and after applying the constraint.

	Relative to the phoneme length of music melody	Relative to the given phoneme length
Preconstraint	4.15	7.36
Postconstraint	4.07	0

Table 4: Prediction errors of acoustic features of different model configurations on the test set.

Model	Number of layers	F0 correlation	F0 RMSE	LE	LCD
DCNN	1	0.73	41.39	3.95	6.25
	2	0.85	40.72	2.67	5.12
	3	0.89	38.85	2.53	4.87
	4	0.84	39.03	2.51	4.82
	5	0.82	39.73	2.54	4.83
LSTM	1	0.86	39.46	3.76	5.14
	2	0.85	37.12	2.64	4.69
	3	0.84	36.80	2.68	4.23
	4	0.89	36.71	2.63	4.52
	5	0.88	38.62	3.85	4.34
Our model	1	**0.86**	**38.94**	**2.67**	**5.09**
	2	**0.85**	**35.23**	**2.59**	**4.72**
	3	**0.89**	**35.89**	**2.52**	**4.24**
	4	**0.88**	**35.42**	**2.63**	**4.12**
	5	**0.87**	**35.73**	**2.51**	**4.35**

Table 5: F0 values under different HAP lengths.

HAP length		1	2	3	4
F0 RMSE	Natural melody	22.34	21.76	22.32	21.35
	Score notes	19.38	19.34	19.26	23.48
F0 correlation	Natural melody	0.94	0.95	0.93	0.99
	Score notes	0.97	0.92	0.98	0.92
LE		2.35	2.34	2.31	2.39

Table 6: Prediction errors of different models in fundamental frequency and spectrum.

Model		DCNN	LSTM	Our model
F0 RMSE	Natural melody	35.21	21.79	**20.45**
	Score notes	34.68	19.36	**8.59**
F0 correlation	Natural melody	0.85	0.94	**0.93**
	Score notes	0.86	0.95	**0.98**
LE		4.34	2.34	**2.38**
			3.59	**3.53**

singing melody of better tonal consistency with music melody.

6. Conclusions

Based on the RNN algorithm, this paper probes deep into the automatic synthesis of music teaching melodies. After extracting the acoustic features from music melodies, the authors established a sequence-sequence model for synthetizing general music melodies. To find the suitable signing segments for a given music melody in the teaching scenario, an RNN was set up to synthetize music melody with singing melody. After that, the convergence of different network models was compared through experiments, which verifies the feasibility of our model. In addition, the results of different models were compared before and after adding the singing melody, and the difference of the melody generated by our model and the original music melody was quantified accurately. Furthermore, the prediction error of phoneme

time of each model configuration and that after applying the time constraint were obtained through experiments. The relevant results confirm the superiority of our model over DCNN and LSTM in modeling music melody sequence.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] T. Tang, J. Jia, and H. Mao, "Dance with Melody: An Lstm-Autoencoder Approach to Music-Oriented Dance Synthesis," in *Proceedings of the Twenty sixth ACM international conference on Multimedia*, pp. 1598–1606, Seoul, Korea, October 2018.

[2] M. Eppe, T. Alpay, and S. Wermter, "Towards End-To-End Raw Audio Music Synthesis," in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 137–146, Rhodes, Greece, October 2018.

[3] S. Nease, A. Lanterman, and J. Hasler, "Applications of current-starved inverters to music synthesis on field programmable analog arrays," *Journal of the Audio Engineering Society*, vol. 66, no. 1/2, pp. 71–79, 2018.

[4] S. Demircan and H. K. Örnek, "Comparison of the effects of Mel coefficients and spectrogram images via deep learning in emotion classification," *Traitement du Signal*, vol. 37, no. 1, pp. 51–57, 2020.

[5] C. Haworth, "Sound synthesis procedures as texts: an ontological politics in electroacoustic and computer music," *Computer Music Journal*, vol. 39, no. 1, pp. 41–58, 2013.

[6] C.-F. Huang and W.-P. Nien, "A study of the integrated automated emotion music with the motion gesture synthesis via ZigBee wireless communication," *International Journal of Distributed Sensor Networks*, vol. 9, no. 11, Article ID 645961, 2013.

[7] S. Wenner, J. C. Bazin, A. Sorkine-Hornung, C. Kim, and M. Gross, "Scalable music: automatic music retargeting and synthesis," *Computer Graphics Forum*, vol. 32, no. 2, pp. 345–354, 2013.

[8] M. Otsuka, S. Okayasu, T. Fukumori, T. Nishiura, and R. Akama, "Sound Reproduction by Concatenative Synthesis for Japanese Traditional Music Box," in *Proceedings of the 2017 International Conference On Culture And Computing (Culture And Computing)*, pp. 153-154, Kyoto, Japan, September 2017.

[9] M. Moussa, W. Guedri, and A. Douik, "A novel metaheuristic algorithm for edge detection based on artificial bee colony technique," *Traitement du Signal*, vol. 37, no. 3, pp. 405–412, 2020.

[10] U. K. Roy, "Instrumental Bengali Music Synthesis from Transcription with Indian Percussion Instruments," in *Proceedings of the 2016 International Conference On Computer, Electrical & Communication Engineering (ICCECE)*, pp. 1–6, Kolkata, India, December 2016.

[11] S. Ouchtati, A. Chergui, S. Mavromatis, B. Aissa, D. Rafik, and J. Sequeira, "Novel method for brain t classification based on use of image entropy and seven hu's invariant moments," *Traitement du Signal*, vol. 36, no. 6, pp. 483–491, 2019.

[12] F. Ofli, E. Erzin, Y. Yemez, and A. M. Tekalp, "Learn2dance: learning statistical music-to-dance mappings for choreography synthesis," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 747–759, 2012.

[13] S. Bilbao, J. Perry, P. Graham et al., "Large-scale physical modeling synthesis, parallel computing, and musical experimentation: the NESS project in practice," *Computer Music Journal*, vol. 43, no. 2-3, pp. 31–47, 2019.

[14] M. R. Velankar, H. V. Sahasrabuddhe, and P. A. Kulkarni, "Modeling melody similarity using music synthesis and perception," *Procedia Computer Science*, vol. 45, pp. 728–735, 2015.

[15] C. d'Alessandro, L. Feugère, S. Le Beux, O. Perrotin, and A. Rilliard, "Drawing melodies: evaluation of chironomic singing synthesis," *Journal of the Acoustical Society of America*, vol. 135, no. 6, pp. 3601–3612, 2014.

[16] R. Liu, B. Sisman, F. Bao, J. Yang, G. Gao, and H. Li, "Exploiting morphological and phonological features to improve prosodic phrasing for Mongolian speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 274–285, 2020.

[17] M. R. Rajan, "Singing Voice Synthesis System for Carnatic Music," in *Proceedings of the 2018 Fifth International Conference On Signal Processing And Integrated Networks (SPIN)*, pp. 831–835, Noida, India, February 2018.

[18] H. Ahn, J. Kim, K. Kim, and S. Oh, "Generative autoregressive networks for 3d dancing move synthesis from music," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3501–3508, 2020.

[19] T. Tanprasert, T. Jenrungrot, M. Müller, and T. J. Tsai, "Midi-sheet Music Alignment Using Bootleg Score Synthesis," in *Proceedings of the Twentyth International Society for Music Information Retrieval Conference*, pp. 91–98, Delft, Netherlands, November 2019.

[20] E. R. Miranda, L. Bull, F. Gueguen, and I. S. Uroukov, "Computer music meets unconventional computing: towards sound synthesis with in vitro neuronal networks," *Computer Music Journal*, vol. 33, no. 1, pp. 9–18, 2009.

[21] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "VoiceLoop: Voice Fitting and Synthesis via a Phonological Loop," in *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[22] H. Romsdorfer, B. Pfister, and R. Beutler, "A Mixed-Lingual Phonological Component Which Drives the Statistical Prosody Control of a Polyglot TTS Synthesis System," in *Proceedings of the International Workshop On Machine Learning For Multimodal Interaction*, pp. 263–276, Martigny, Switzerland, June 2004.

[23] H. Romsdorfer and B. Pfister, "Multi-context Rules for Phonological Processing in Polyglot TTS Synthesis," in *Proceedings of the Eighth International Conference On Spoken Language Processing*, Jeju Island, Republic of Korea, october 2004.

[24] M. Toman, M. Pucher, S. Moosmüller, and D. Schabus, "Unsupervised and phonologically controlled interpolation of Austrian German language varieties for speech synthesis," *Speech Communication*, vol. 72, pp. 176–193, 2015.

[25] E. V. Mistyukov and D. V. Alexandrov, "Musical Synthesis for Certain Music Styles Based on Machine Learning Algorithms," in *Proceedings of the SAI Intelligent Systems Conference*, pp. 543–562, London, UK, September 2016.

[26] H. Wierstorf, C. Hold, and A. Raake, "Listener preference for wave field synthesis, stereophony, and different mixes in popular music," *Journal of the Audio Engineering Society*, vol. 66, no. 5, pp. 385–396, 2018.