Neural Processing Letters (2020) 52:1893–1912
https://doi.org/10.1007/s11063-020-10241-8

A Style-Specific Music Composition Neural Network

Cong Jin[1] · Yun Tie[2] · Yong Bai[2] · Xin Lv[3] · Shouxun Liu[4]

Published online: 9 June 2020

Abstract
Automatic music composition could dramatically decrease music production costs, lower the threshold for the non-professionals to compose as well as improve the efficiency of music creation. In this paper, we proposed an intelligent music composition neutral network to automatically generate a specific style of music. The advantage of our model is the innovative structure: we obtained the music sequence through an actor's long short term memory, then fixed the probability of sequence by a reward-based procedure which serves as feedback to improve the performance of music composition. The music theoretical rule is introduced to constrain the style of generated music. We also utilized a subjective validation in experiment to guarantee the superiority of our model compared with state-of-the-art works.

Keywords Automatic composition · Reinforcement learning · Actor–critic network · Music rules

1 Introduction

Music, as a crucial form of art, brings us much pleasure and also satisfies our higher demands for richer spiritual and cultural lives [1]. Original music could have broadened its market in combination with the booming industry of multimedia such as video games [2] or short videos [3], but high-cost professional music production and relatively monotonous style of music fail to meet personalized needs [4]. As machine learning and deep learning spring up, intelligent music composition is a hot research topic in the field of computer music at home and abroad, which is mainly based on one neural network or combined neural networks. From a mathematical perspective, as a major direction applied in music creation, a neural network can model any function up to any given precision with a large number of basis functions.

Cong Jin and Yun Tie have contributed equally to this work.

This document is the results of the research project funded by the National Natural Science Foundation of China (Grant Nos. 61631016, 61901421 and 11571325), National Key R&D Program of China (Grant No. 2018YFB1403903) and supported by the Fundamental Research Funds for the Central Universities (Grant Nos. 2019E002, CUC19ZD003 and CUC200B017).

✉ Shouxun Liu
sxliu@cuc.edu.cn

Extended author information available on the last page of the article

In addition, the relative topics include accompaniment generation, melody generation, style transfer, etc.

According to the different characteristics of intelligent composition, it can be divided into: random generation composition, rule-based knowledge system composition, mathematic-based composition, music grammar composition, and genetic algorithm composition [5] . To a certain extent, these methods can meet the basic requirements of automatic composition, but as for the melody's structure, can not meet the constant changes of music datasets. Since Google's AlphaGo was born and defeated many Go champions, Google launched the Magenta [6] in 2016 to generate music by machine. Later, music composition based on neural network has developed rapidly and receives wide attention. At present, most of the research on intelligent composition relying on neural network is focusing on how to improve the structure of neural network and how to preprocess the training data from signal processing angles. However, a single neural network can no longer meet the requirements of controlling music style that constrained by certain music rules; the preprocessing of training data requires a lot of manual labeling and data cleaning [7], which consumes a lot of computing and storage resources [8]. Therefore, it' s necessary to adopt a preprocessing method and a fusion network to deal with the feature extraction and control of music styles [9].

In this paper, we propose a style-specific Music Composition neural network based on reinforcement learning. Styles include classical, pop, jazz, rock and R&B etc.The so-called "style-specific" can be defined as a certain preset style such as classical music. Moreover, this paper comprehensively uses the superiority of deep learning network in strategy learning and intelligent processing of time series signals, combined with the unique properties of music signals to design a fusion neural network of intelligent composition, and the network mechanism is utilized in the processing of music style's feature extraction and control problems. This not only solves the problem of time-consuming and laborious manual marking of large-scale data [10], but also contributes to adding some music rules, to improve the efficiency of generating music and constrain the music style.

The remaining sections are organized as follows: Sect. 2 presents some mainstream methods of music composition. Section 3 elaborates the structure and algorithm of our model. Section 4 conducts the result of experimental part. Section 5 discusses the comparison and evaluation of different models.

2 Related Work

2.1 Traditional Methods

Researchers have proposed many methods for automatic composition. Among traditional methods, hidden Markov models (HMMs) [11] is a probability-based music generation model while relying too much on mathematical relations. The generated samples are relatively rigid and lack of subjectivity. Another traditional solution, rule-based expert system [12] generates music according to theoretical knowledge of music. However, the formulation of music rules is time-consuming, laborious and difficult to cover all aspects. Genetic algorithms [13] depends on natural selection in biology can be used to deal with nonlinear complex problems. It does not need comprehensive music rules while the whole composition process is too abstract to follow. So it is difficult to extract valuable music ideas and set the fitness function.

2.2 Symbol Model Based on Neural Networks

Nowadays, existing intelligent music composition method can be broadly classified into two types according to format variety of training datasets: an original audio model and a symbol model (MIDI). The symbolic model trained and generated at the note level is currently a popular method. Moreover, with the fast development of neural network, most of music generation technologies are based on one model or multi-model network. Johnson proposes [14] a polyphonic generation framework by combining RNN and general neural network. The part of the loop in the network represents time, and the other not looped represents harmony. Music generated by this method is very poor in structure due to relatively random and it is easy to cause excessive reduction or enlargement of long-term memory during training. Google Brain's Magenta project [15] presents further optimizations based on LSTM to enhance long-term associations between multiple measures; Attention RNN adds mask vectors to loop joins to control model weights for different historical states for the effect of music generation, but the resulting fragments are short and have a relative poor structure. Hadjeres et al. [16] designs the LSTM network called "DeepBach" to solve this problem, which combines two feed forward networks and two LSTM networks to generate music. That is highly similar to Bach's four-part chorus style, called "Style migration". However, the disadvantage of this type of network is lack of high-order structure output, which requires the addition of music rules. Therefore, Lattner et al. [17] raises the RBM network adding some constraint rules. When studying the generation of Mozart-style sonatas, it not only learns the features of texture, but also adds some features of chord and melody manually. Although the RBM network can constrain the sampling for gradient descent optimization, it will cost too long time to calculate and sensitive to sampling noise.

In recent years, there are many cases of variational auto-encoder (VAE) for music generation. The essence is the compression algorithm of encoders and decoders, however, it is difficult to handle harmony and polyphony. Akbari et al. [18] proposes a sequence generation algorithm and a combined network based on VAE–GAN structure, in which each generated continuous frame is encoded, then the generator predicts the generated content of the next frame according to the coding sequence of the previous frame, finally the discriminator determines between the real data and the generated data, but this model is difficult to model long sequence. To solve this problem, Google Brain's Roberts et al. [19] advances the MUSIC-VAE model, which adopts a layered decoder to model the sequence of notes and it shows better performance than the model with "flat" baseline in sampling, interpolation and reconstruction. Moreover, Brunner et al. [20] presents MIDI-VAE, a multi-track polyphonic music generation model, using interpolation algorithm to automatically change the pitch, dynamics and instruments of musical works, to transform the style, and to verify the effect of style migration by selecting a proper classifier. However, the segments of training data for extracting style features is too short, and the generated segments are too short. Nowadays, generative adversarial network (GAN) has achieved remarkable results in image generation and processing, and Yang et al. [21] introduces MidiNet, which combines GAN network and CNN network to generate popular melody. The architecture consists of a generator, a discriminator and a conditional network with four convolutional layers to define the harmony direction. In addition to this, MuseGAN proposed by Dong et al. [22] first divides the music into five levels including passage, phrase, measure, beat and pixel when studying multi-track sequence music generation and automatic accompaniment, then generates it layer by layer when each track is relatively independent and cooperates with each other. Finally, all the tracks and bars are taken as an input sample into the discriminator together with the samples in the training set for training. However, the pre-trained work is too cumbersome, and requires some knowledge of music and early labeling work. Based on above, SeqGAN proposed by Yu et al. [23] can solve the problem. GAN network is difficult to update the generation model of discrete sequence, which can be used to model the data generator as a random policy in reinforcement learning, and to bypass the difference of the generator by implementing updates to the gradient strategy. When referring to style migration, Brunner et al. [24] proposes that the CycleGAN model can be applied to the style migration of symbolic music, adding another discriminator to keep the generator's structure features of the original music, but the generated style is not rich enough.

2.3 Audio Model Based on Neural Networks

The symbolic model captures the long-term dependence of the fusion structure, but unable to understand the nuances and richness of the original audio generation. The original audio model directly trains the sampled audio waveform [25] to produce a realistic sound, i.e. timbre. WaveNet [26] introduced by DeepMind is based on probability and auto-regression. The prediction distribution of each audio sample depends on the previous model. The training data is an audio waveform file, which can generate realistic music fragments. However, the model is trained for a long time, continuous sound segments in every second, a calculation unit generates 24,000 samples, which has the low efficiency for music generation. Considering the problem of WaveNet, Manzelli et al. [27] presents an automatic music generation method that combines the original audio model and the symbol model to generate music with better structure and better melody. This method adopts LSTM to learn music with different melody structure styles, and then takes this unique and symbolic result as a conditional input for original audio generators to generate a model that automatically produces new music.

3 Methodology

We propose an automatic music composition neutral network (MCNN) which consists of generator, control network and probability network, with its structure shown in Fig. 1. The novelty of MCNN is making full use of the superiority of LSTM model in the aspect of sequence prediction and actor–critic (AC) network in constraints of music theory to generate high-quality music, with probability network [28,29] integrated into AC network [30,31]. We first represent a generator combined actor–critic (AC) network with LSTM network as a result of music generation model with the construction and parameters. Then we formulate the process of sequence prediction as inference with LSTM hidden layers following the given weight matrix in Sect. 3.1. In Sect.3.2 we mainly expound the control network, which introduces music rules as restriction for a particular style's generation, calculates the loss function of actor–critic updating process and defines an algorithm for the reward function of scoring music generated by actor network. Finally, the probability network is composed of CNN discriminator network and probability output, and it can return P_D when generated sequence is true data.

3.1 Generator

In this model, LSTM network can be regarded as a generator, and CNN network is regarded as a discriminator. Music sequence is generated by LSTM based on a specific probability distribution, while CNN network as a probability discriminator is difficult to judge a single or

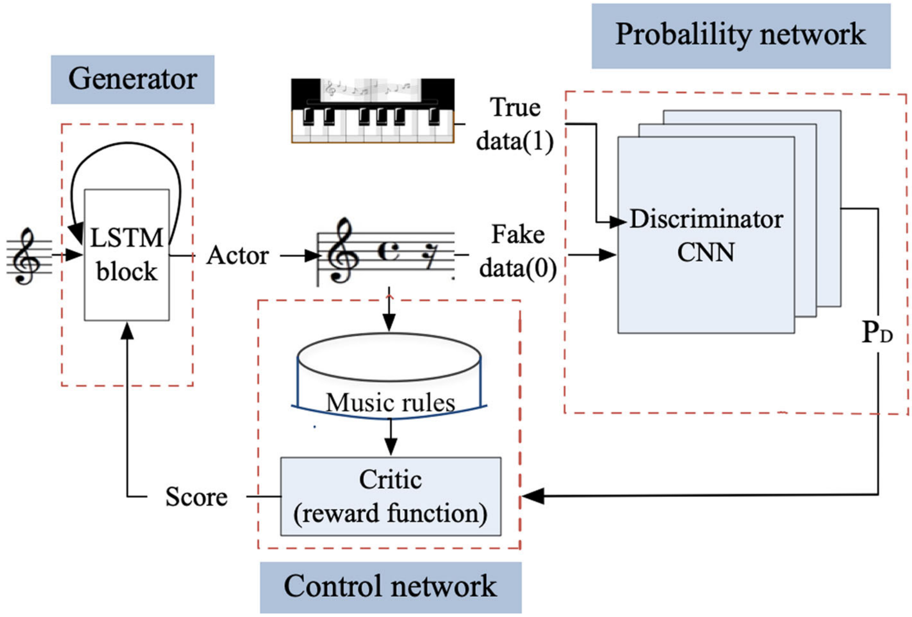


Fig. 1 Music composition neutral network (MCNN). (In this mixed network, LSTM works as a generator, music rules as reward function are added to the control network and CNN performs as a discriminator to predict the probability output called probability network.)

consecutive music sequence. Therefore, this paper designs a one-step update and round update mode. That is to say, for the generation of a complete music sequence, the critic network with music rules and the probability network act as a reward function and participates in the updating of the generated network. For generating music sequence, critic network updates weight parameters by adding the music theoretical rules as the reward function and return scores.

In this paper, the generation of music sequence is realized by LSTM network [32,33]. The parameters of generator are trained by the dataset, and actor–critic (AC) network is used to make fine-tuning. Actor–critic (AC) network is a reinforcement learning algorithm that updates the probability of action selection of music sequence by stochastic policy gradient algorithm. This probabilistic actor selecting method is conducive to generate diversified music sequences. Critic network is a feedback and control network that actually scores the sequences generated by actor network. The reward function of critic network consists of the probability output of CNN network and music rules

1. Input and output

The training model is presented in Fig. 2a. The LSTM network is pre-trained by inputting the classical piano database in MIDI format. Music is generated randomly and labeled as 0 while the real samples are labeled as 1. Meanwhile a binary model was constructed to pre-train the CNN network. Figure 2b demonstrates how the model generates music sequence. By setting initial notes, music sequence is generated by LSTM network and labeled as 1. After CNN network, the output binary discriminant probability is output. The output of music rules adjust LSTM network through weighted calculation.

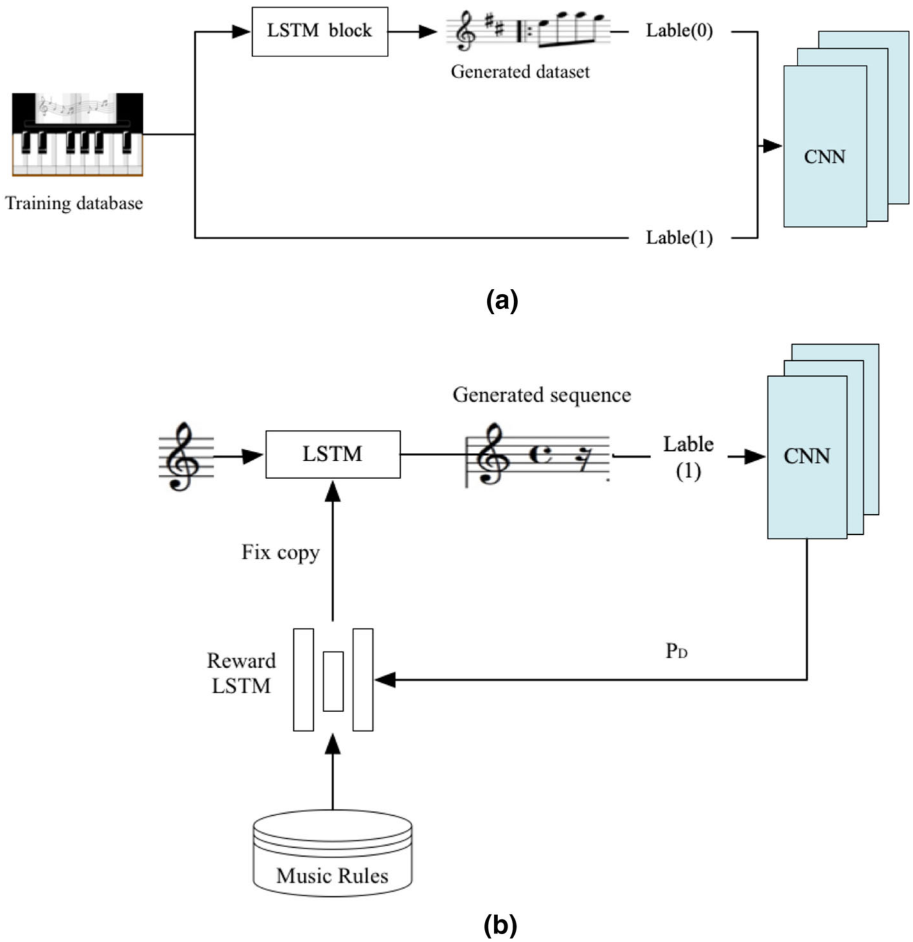


Fig. 2 Generating network. **a** Training model, **b** generation model

2. Generate music sequence using LSTM

LSTM has special temporal Memory function. Based on the ordinary multi-layer BP neural network, the structure increases the horizontal connection between each unit in the hidden layer. Through a weight matrix, the previous information can be connected to the current task, that is, the next sequence action can be predicted by the previous generation sequence. LSTM network structure conforms to the acoustic timing [34–36], and its sequence generation and prediction process is as Fig. 3.

Set the input sequence $(x_0 \ldots x_t)$, the hidden layer $(h_0 \ldots h_t)$, and the output sequence $(y_0 \ldots y_t)$, then the output of the hidden layer is calculated as Eq. (1),

$$h_t = f\,(Ux_t + Wh_{t-1}) \tag{1}$$

where U is the weight matrix of input, W is the weight matrix of the hidden layer at the previous moment, and f is the activation function of the hidden layer.

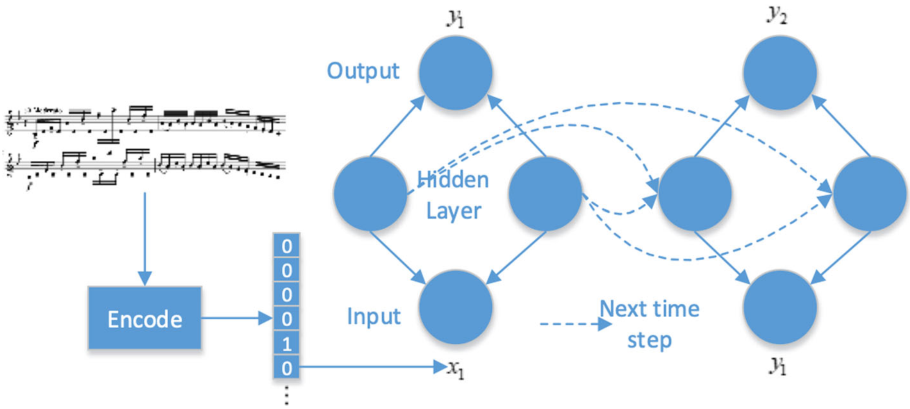


Fig. 3 LST sequence generation principle

After calculating the output of the hidden layer, the output sequence y_t can be derived from the activation function of the output layer δ and the weight matrix V.

$$y_t = \delta(Vh_t) \tag{2}$$

The algorithm flow of LSTM sequence generation is as follows.

Algorithm 1 Training Process of LSTM Generator

Input: preprocessed training data are fed into the network
Output: model parameters of neural network
1: Input training data, set iteration times, batch size, the number of hidden layer's cells and network layers.
2: Take the cross entropy between the output of the neural network at the current time and the input at the next time as the loss function to update the network's parameters.
3: LOOP step 2 until the iteration is completed, the loss is converged.
4: Output weight parameters of LSTM Generator.

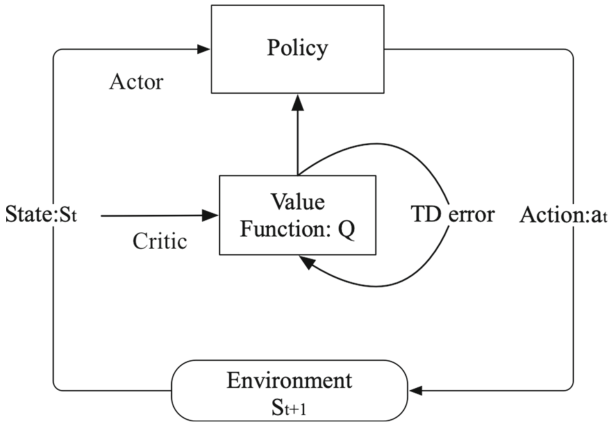
3.2 Control Network

1. Music-rule based reward

Similar to the language grammar in Chinese, the generation of music follows music rules. In this section we focus on how to use rule-based constraints to modify the performance of music composition. In order to generate classical style's music, we choose several composition principles from the common introduction to classical music composition.

Following the principles of Refs. [37–39], we define the reward function $r_{mr}(a, s)$ for the following classical music's characteristics. In a piece of music, the interval difference between two adjacent notes should be less than an octave; all of the notes should be on the same pitch, and the beginning and end of the song should be controlled by the pitch of the main note. We take the key of C major as an example, the note appears in the first and last 4 beats should be middle C; the pitch can't rise or fall all the time, and also can't remain on the same pitch; one musical note can not be repeated above four times in succession. The change in the rhythm of the music can't be too fast; the highest note in a piece of music is

Fig. 4 Actor–critic network



unique, and the lowest note is unique; the occurrence of a split note should be avoided as much as possible. In order to adapt to the matching of chords, we have added rules of chords: the notes that appear in the strong beat are generally the inner chord; the weak beats in the bar must be within one octave of the main tone; the last note should avoid the appearance of fa. The trend of chords can provide a good constraint on the generation of the melody. We order the rewards that meet these rules get positive values, otherwise will get negative values, and finally the sum of these values is the score $r_{mr}(a, s)$ of music rules.

We don't claim these characteristics are exhaustive, but adding the rules will guide our model towards traditional composition structure and make the generated music more structural and more obvious in classical style.

2. Act-critic network via policy gradient

An agent conducts interaction with an environment in reinforcement learning. Actor–critic (AC) network is an intensive learning algorithm based on time difference (TD) [40,41], which updates neural network parameters in order to optimizes the probability distribution of action selection, so that agents can get the maximum cumulative reward in the interaction with the environment. In this model, policy $\pi\colon S_t \to a_t$ is the mapping of state space to action space. At time t, the intelligent body actor selects the action a_t according to the policy $\pi(a/s)$ on the current state S_t, executes the action and transfers to the state at the next time S_{t+1}, and feedback the reward R of the action. From Fig. 4 we can conclude, critic's reward function can be used in the network to score music generated by actor network. If the critic network outputs a higher score when action a_t is taken at time t and state S_t, the probability of such action is increased; otherwise, the probability of such action is reduced. It can be seen that the build of the reward function will directly affect the quality of the actor network generating music. The update of actor network is determined by training data set, probability output of CNN discriminator and music rules.

(1) Reward function

Reward function can be used in the network to score music generated by the actor network. If the Reward outputs a higher score when action a is taken at time t under state s, the probability of such action is increased. Otherwise, the probability of such action is reduced. It can be seen that the build of the reward function will directly affect the quality of generated music. Reward function in this paper consists of probability output P_D and music rules $r_{mr}(a, s)$.

Set Q_a as the reward fuction of generating network, and T as the length of the generated music sequence. According to the policy π, get the sequence $\pi(a_1, a_2, a_3 \ldots a_t, \ldots a_T)$, then

$$Q_a \leftarrow r_{mr}(a, s) + P_D \tag{3}$$

where $r_{mr}(a, s)$ is the music rule and P_D is the the probability output of CNN discriminator.

In this case, with the policy of $\pi(a/s)$, the total reward value R obtained by taking action a under state s becomes:

$$R = \log \pi(a/s) + r_{mr}(a, s) + P_D. \tag{4}$$

(2) Updating process

At time t, the parameter of the actor network is θ_a^t, and the parameter of the critic network is θ_v^t, and the loss function $\sigma(\theta_v)$ in the critic network structure can be obtained by the time difference algorithm, and the parameter θ_a^t of the actor network are updated by the loss function. The calculation of Q network is expressed as the mean value of accumulated rewards R_t between time $t1$ (the beginning of the action sequence) to time T (the end of the action sequence), in which R_t represents the reward function value of generated note. That is, for the action sequence $\pi(a_1, a_2, a_3 \ldots a_t, \ldots a_T)$, its Q function is calculated as follows,

$$Q(s, a, \theta_v) = \mathbb{E}[R_t / S_t = S, a_t = a, \theta_v] \tag{5}$$

$$R_t = \sum_{t1}^{T} \omega^{t-t1} r_t \tag{6}$$

where ω is the weight of each future reward value r_t, and the expectation $\mathbb{E}$ can be approximated by sampling methods.

Based on TD time difference algorithm, critic network is composed of Q target network and Q action network [42–44]. Q target network is copied from Q action network every N steps, and its value function is calculated as follows:

$$J\left(s, a, \theta_v^-\right) = R + \gamma \max_{a'} Q\left(s', a'; \theta_v^-\right) + V\left(s; \theta_v^-\right) \tag{7}$$

where s' is the next state after action a, and a' is the next action. θ_v^- is target value network parameters. γ is the discount factor and V is the state value function.

The loss function of Q action network and target network can be calculated as,

$$\sigma(\theta_v) = \mathbb{E}_{(s,a,r,s')}\left[\lambda\big(J\left(s, a, \theta_v^-\right) - Q\left(s, a, \theta_v\right)\right] \tag{8}$$

where θ_v^- is target value network parameters, λ is the discount factor of the value of the objective function.

The loss function σ is optimized end to end by the stochastic gradient descent algorithm. The parameters θ_v of critic network are updated as follows,

$$Q_v \leftarrow Q_v + \nabla_{\theta_v} \sigma^2(\theta_v) \tag{9}$$

and the update mode of actor network parameters θ_a is:

$$\theta_a \leftarrow \theta_a + \nabla_{\theta_a} \pi(a/s, \theta_a) \sigma(\theta_v) \tag{10}$$

Algorithm 2 Actor–critic network updating process

Require: An actor policy $\pi(a/s, \theta_a)$ and a critic $Q(s, a, \theta_v)$ with weights θ_a and θ_v respectively; Pre-train LSTM with weight θ and pre-train CNN discriminator with weight φ.
Output: The loss function of the reward network $L(\theta_v)$.
1: Initialize delayed actor π_0 and critic Q_0 with the weight: $\theta_a' = \theta, \theta_v' = \theta$.
2: **while** Not Converged **do**
3: Initialize a random Notes a_0.
4: Generate a sequence of actions $(a_1, a_2, a_3 \ldots a_t, \ldots a_T)$ from policy π.
5: Compute Music rules $r_{mr}(a, s)$ and probability output P_D.
6: Compute targets for the critic in Eq. (7) and Eq. (4).
7: Compute the loss function of Q action network and target network in Eq. (8), then the loss function of the reward network is
$L(\theta_v) = \sigma^2(\theta_v)$
8: Update the critic weights θ_v using the gradient from Eq. (9),
$Q_v \leftarrow Q_v + \nabla_{\theta_v} L(\theta_v)$
9: Update actor weights θ_a using the following gradient in Eq. (10),
$\theta_a \leftarrow \theta_a + \nabla_{\theta_a} \pi(a/s, \theta_a)\sigma(\theta_v)$
10: **end while**

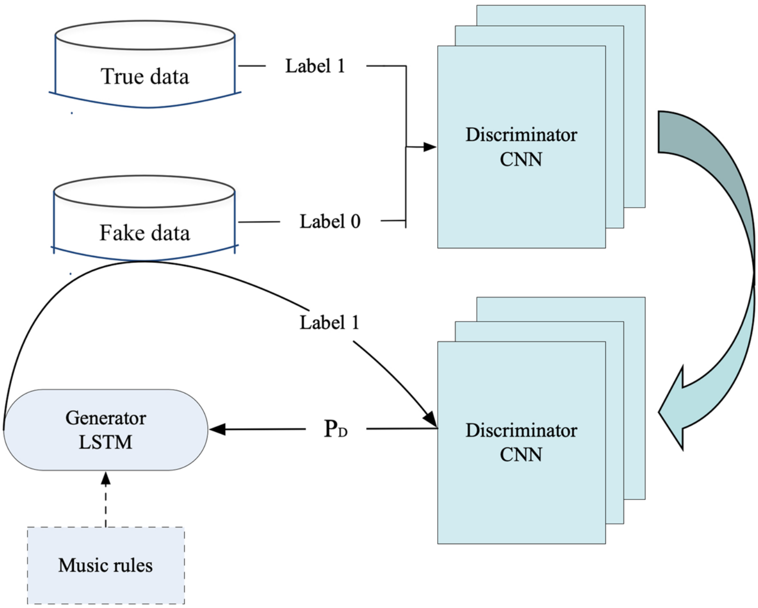


Fig. 5 Probability network

3.3 Probability Network

1. CNN discriminator

The algorithm takes the probability output of CNN discriminator as an important part of the reward function in critic network. This discriminator can be understood as a probability indicating how likely a generated sequence is from true data or fake data. Actor–critic reinforcement learning algorithm is integrated into generating network to generate music with consistent styles but diverse tunes, where, LSTM network is the generator and CNN network is the discriminator. As illustrated in Fig. 5, the CNN discriminator [45,46] is trained by positive samples from the true data and negative samples from the fake data which is generated by the LSTM generator, labeled as 1 and 0 respectively. Moreover, the discriminator predicts the probability that a generated sequence is the true data and outputs the probability P_D.

2. Probability output

We construct a binary classification model based on CNN. In the process of network updating and optimizing, the music rules are added to make the generated sequence close to the real sequence in style, and the probability outputs by CNN to distinguish the similarity between the generated sequence and the real sequence. The discriminator only provides a reward value for completing the sequence, so we define the period T as the completed time of a generated sequence, and the probability P_D will output simultaneously.

When the generated sequence is sufficiently confused with the real sequence, the probability output of CNN network is close to 0.5. If $t < T$, the generated sequence is incomplete, it is difficult for CNN network to fully discriminate the generated sequence, so setting the default P_D to 0. When $t = T$, we consider the estimated probability of being real by the discriminator $D_\varphi(a_{1:T})$ as the reward. In order to enhance the effectivity of the reward function, subtract base value 0.5 when calculating the probability, as shown in Eq. (11).

$$P_D = \begin{cases} 0 \\ D_\varphi(a_{1:T}) - 0.5. \end{cases} \tag{11}$$

4 Experiment

In the experiment part, we first introduce the Database and preprocessing in Sect. 4.1 and model setting and training details in Sect. 4.2. Section 5.2 shows the results of our model and compares the results with other models to show our proposed model is more efficient.

This paper realized the intelligent generation of music on the basis of a deep learning framework “tensorflow”. The server used was Intel Xeon 2640v4 2.4 GHz processor and 11 GB video memory NVIDIA 1080Ti GPU, with 512 GB system memory composed of 8 blocks of the same standard 64 GB memory chips. Programming language used in this experiment was Python3.6.1.

4.1 Database and Data Preprocessing

We employed the matched subset of the Lakh MIDI dataset (LMD) and Classical Piano Midi Page dataset as the training dataset, Lakh MIDI dataset [47] provides a rich collection of real-world MIDI files and some associated meta-data. Classical Piano Midi Page dataset collected classical music sets of composers from the eighteenth century to the nineteenth century, including works of Bach, Beethoven, Chopin and other 25 composers. In the experiment, we select 2000 music samples of Classical Piano Midi Page dataset with specific composers' styles in MIDI format [48] as test dataset, and each sample was a single orbital with an average time of 2–4 min. To meet the requirements of the experiment, each sample was divided into 20 s to obtain more than 20,000 classical music samples.

As is shown in Fig. 6, midi files in the dataset were transformed into sequences. First, a discretization algorithm was utilized to decode a midi file to a matrix, which can be showed as a piano roll. And an example of piano roll [49] is shown in Fig. 6. The value of the row of a matrix ranges from 54 to 86, representing the note number in the midi music. The value of

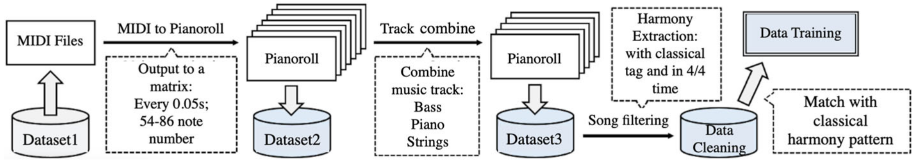


Fig. 6 Data pre-processing

the column starts from 0 and increases by 0.05, representing every 0.05 s in the midi music. Next matrix was transformed into a sequence, which was the input for our model.

4.2 Training

In order to train the generation and the reward model, we utilize the preprocessed classical piano music as training data. For each small batch, music segments are stochastically selected and not replaced, and then a sub-segment is stochastically selected from the whole music segments, so we can determine the size of the small batch. We employ a random gradient descent with 0.9 momentum and remain the learning rate at 0.001. We find directly training the generation model with pre-trained parameters could not perform well, and the loss almost never decreases. Therefore, we constructed AC optimization network to compute the reward function and update the loss. MCNN network is mainly composed of LSTM, reward function and CNN network.

LSTM network is mainly divided into three layers: input layer, hidden layer and output layer, each of which contains 512 LSTM cells. This experiment stipulates that the length of each note sequence is 192, and each note sequence is one-hot coded so that each note sequence corresponds to a one-hot vector. Softmax was taken as the probability output layer of each musical note sequence in the experiment, and both the reward network and CNN discriminator acted on the output layer of softmax.

The AC network consists of two parts: actor and critic. While actor is the generating part of the music sequence, critic is created by the reward function. In order to enhance the evaluation ability of critic network, Q network and target Q network have been established. In the experiment, the parameter of target Q network is copied from the Q network. Q network was updated synchronously while target Q Network was updated every 2 steps, with an update frequency being 20.

CNN discriminator is divided into four layers: embedded layer, convolutional layer, maximum pooled layer and softmax layer. The activation function Relu is added between convolutional layer and maximum pooled layer, and the output of softmax layer is taken as the probability output of the discriminator. In the experiment, reward function will generate a Q value for each pair of state action (s, a), which will be applied to return the music score to LSTM, adjust network parameters, and optimize the probability of distribution generated by the sequence. Better music samples are continuously obtained through multiple iterative updates of the network. There are two aspects that affect the performance of music generation, so we intend to discuss the two aspects and also how the two aspects influence the performance of experimental results.

To test the influence of the number of cells in different hidden layers and various iterations on experimental results, we conduct three comparative experiments by choosing 128, 256 and 512 hidden neurons from 1 to 3500 epochs. Experimental results show that the more cells in the hidden layer, the stronger the ability of LSTM network to learn and extract characteristics of data, and the more effectively reduce the loss between predicted value and target value. The dimension of hidden layers has an important influence on experimental results, however, the more extensive and deeper neural network will greatly increase the experimental computation. Meanwhile, the effects of different iterations with different hidden cells on loss function show that the more iterations, the more learning and adjustment of weight parameters, and to some extent, the accuracy of the model can be improved. In addition, as the iterations increases, the decreasing range of loss function gradually slows down. Of course, too many iterations will result in more computation.

5 Comparison and Evaluation

5.1 Comparision

Depth policy gradient based on AC network updates the network's parameters by continuously calculating the gradients of policy parameters with the expected total reward, and eventually converges to the optimal policy. The relationship between LSTM generator and CNN discriminator enables the network to produce more appropriate samples, which is conducive to generate music with a specific topic under the multi-style topics. In order to generate music sequences with a certain style and diversity, we propose an automatic music composition neutral network to build a framework that can generate music with a specific style, which is called MCNN.

In this subsection, in order to confirm the superiority of our model, we conduct experiments of another three models that generate music intelligently and then compare all the generation results. Following are these generation models.

(1) *RNN + DQN* [50] This network is proposed by Google DeepMind, which mainly consists of a value-based Q-learning network and RNN network. But the value-based algorithm is easy to numerical jitter caused by state changes, resulting in insufficient convergence, in addition, an Agent of DQN network can only interact with a single environment. In contrast, MCNN contains the actor and critic algorithm, and the actor algorithm is based on policy gradients by updating the reward. This policy-based method can directly solve in the policy space, which is smoother than the value-based method in numerical jitter. Moreover, MCNN can also improve learning efficiency and robustness through asynchronous and multi-threaded methods compared with DQN network.

(2) *SeqGAN* [23] The novel collision of adversarial network and reinforcement learning, which is devoted to generating discontinuous sequence, such as text sequence generation. This model is similar to our model, which includes generator and discriminator, sequence generation as generator and reward as discriminator respectively. While we add music rules as the reward function in order to generate music with specific styles.

(3) *VAE–GAN* [18] The variational auto-encoder and generative adversarial networks are both the generative models based on the unsupervised learning method, which can integrate the adversarial ideas into the variational auto-encoder. Thereby, we can train encoder, generator and discriminator synchronously to generate images on the basis of mutual compensation between GAN and VAE. In VAE network, constraints can be added to the coding part to extract chord's feature, and this constraint generates potential vectors that follow Gaussian distribution. Also adding some constraints, but due to this coding scheme it is difficult to model long sequences. On the contrary, MCNN adds the

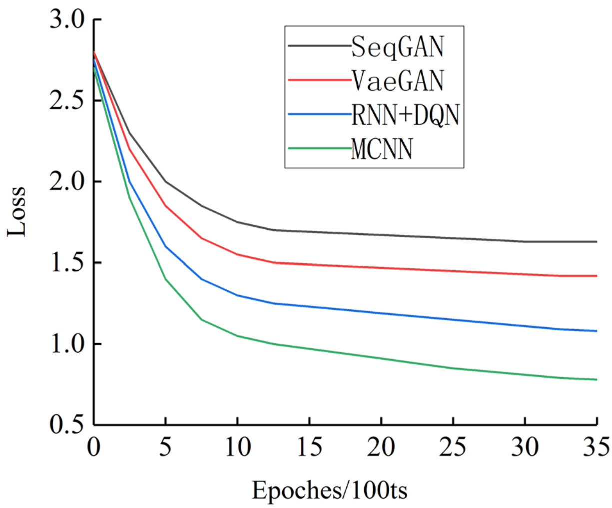


Fig. 7 Comparison results for different models

constraints of music rules into the critic part by controlling the music style, which can improve the efficiency and quality of sequence generation.

To ensure the objectivity of the experiment, we adopt the same training data in each experiment among different models, with 1024 neurons. Different time steps have been assessed: RNN + DQN, SeqGAN, VAE–GAN and MCNN, and considered in the truncated back propagation.

To compare different models, Fig. 7 shows the loss values of the experiments conducted. All the models adopt 1024 neurons and have trained the same datasets. Execution are shown for RNN + DQN, SeqGAN, VAE–GAN and MCNN with 100 timesteps.

SeqGAN and VAE–GAN models shown in Fig. 7 get a slow decline in loss and obtain a similar precision no matter how many time steps there are. It may be caused by the experimental results that these models cannot learn long-term dependence. By contrast, there is a slight and relatively large variety between the experiment with 100 time steps in RNN + DQN and MCNN models respectively. In this occasion, the results indicate that the MCNN performs better than RNN + DQN in terms of the problem solving, reaching the minimum loss below 1.0 using 1000 time steps while RNN + DQN reaching the loss above 1.0 using the same time steps. Therefore, the neural network architecture with the best performance in the training data has been the MCNN model, evaluated in the iteration from 0 to 3500 time steps.

5.2 Evaluation

It is a complicated task to evaluate musical samples because there is no standard procedure for evaluating music's quality. In order to test the music generation effect of MCNN, this paper evaluates the model from the subjective and objective perspectives. To demonstrate the model capability, we have uploaded music samples generated by our MCNN model and other models in the website (https://github.com/jwang44/MCNN-music-generation-demo).

Table 1 Compliance rate of generation music of 4 algorithms

Algorithm	SeqGAN	VAE–GAN	RNN–DQN	MCNN
Compliance rate	0.6234	0.8152	0.9489	0.951

1. Objective evaluation

This paper introduces an objective evaluation method based on the minimum distance by measuring the distance between the generated sample and the training sample to evaluate. Minimum distance classifier (MDC) is a simple classification algorithm based on vector space model [51,52]. The basic theory of MDC is to generate a center vector U_k ($k = 1, 2, \ldots, m$; m is the number of the class) representing this class according to the arithmetic average of the training set. For each data tuple x to be classified, first calculate the distance d_x^2 between x and U_k, and finally it is determined that x belongs to the class closest to it. Euclidean distance is most common used by the minimum distance classifier.

In the experiment, we selected 500 training samples for training and 300 generated samples for testing, and extracted 8 features of classical music as the basis for comparison, which are: range, repeated notes, vertical perfect fourths, rhythmic variability, parallel motion, vertical tritones, chord duration, number of pitches [39,47].

For the selection of reasonable distance threshold d0, we divided the training sets into two parts: the former 250 data were input as training sets, and the latter 250 data were input as test sets. Adjust the size of the d0 until the accuracy of classifying the latter 250 is close to 1, ultimately classification distance was adjusted to be $d0 = 42.4$ [51,53]. When $d < d0$, it is determined that the generated sample is a classical music that meets the requirements, and labeled as 1. Otherwise, it is not consistent with the sample, which is labeled as 0. According to recorded data, the compliance rate of generation music could be calculated by Eq. (12),

$$\tau = \frac{\sum_{i=1}^{300} g\left(i\right)}{300} \tag{12}$$

where $g\left(i\right)$ represents the labeled score of the samples.

In the experiment, the algorithm in this paper is compared with the results of three music generation algorithms of VAE–GAN, SeqGAN and RNN–DQN, as shown in Table 1.

According to the experimental results, music sample generated by MCNN network has the highest compliance rate, far exceeding SeqGAN and VAE–GAN, slightly exceeding RNN–DQN. It can be seen that music generation algorithm based on MCNN network can basically guarantee the validity of music generation.

Minimum distance classifier (MDC) algorithm is used to judge whether the generated music is classical music style, and we use receiver operator characteristics (ROC) curve to judge the performance of different models. The ROC curve is a graph between the true positive rate and the false positive rate. In the experiment, as shown in Fig. 8, the ROC curves of VAE–GAN and SeqGAN are more twisted and sharp, and relatively speaking, RNN–DQN and MCNN are more smooth. AUC (Area Under ROC Curve) values of these models have reached to 0.55, 0.54, 0.85, 0.86, with MCNN the biggest AUC value.

2. Subjective evaluation

Listeners' subjective feelings are crucial to music's evaluation. Through the establishment of an expert review panel, the performance of different models in generating classical music was evaluated from the following five aspects:

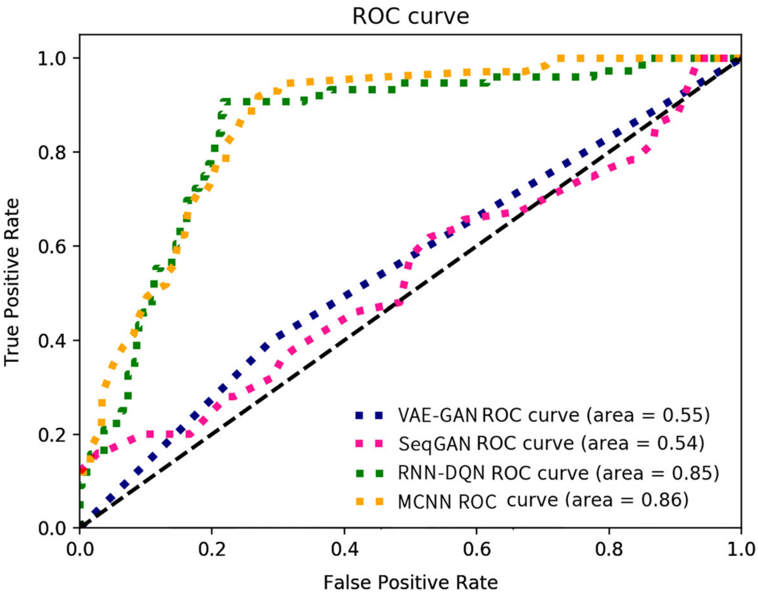


Fig. 8 ROC curve of style classifier for 4 models

- **Melody** The melody of classical music sounds more balanced and symmetrical.
- **Rhythm** Classical style includes unexpected pauses, syncopation, and frequent conversion from long to short notes.
- **Harmony of chord** The progression and trend of chord are very clear and balanced, with triad as the dominant.
- **Musical texture** The use of texture is flexible, and the main tone texture is combined with the polyphonic texture.
- **Emotion** The fluctuation and contrast of classical music's emotion are more obvious.

In this subjective experiment, we invited 15 music professionals composed of 7 teachers from the composition department of the music conservatory, 4 performers from the symphony orchestra and 4 music enthusiasts, including 8 males and 7 females aged between 30 and 40. They listened to the music generated by different models and gave a score on the above aspects to evaluate the performance of those models. They rated each piece of music generated with a score from 1 to 10, among which "1" represents the worst and "10" represents the best. In the end, we gathered the scores, removed outliers that are too high or too low scores compared to other professionals, and averaged the scores of 15 professionals.

We validate the performance of our model based on subjective validation [54] by human in the following aspects: melody, rhythm, harmony of chord, musical texture, emotion. The results are shown in Table 2.

According to the result, we can see that our MCNN model has a great performance and effectiveness on music generation, especially on the perspective of harmony of chord and musical texture.

On the one hand, using GAN individually on generating music creates much similarities on the music sequence and the training data, causing a lack of diversity. On the other hand, if we use reinforcement learning individually, considering there is a bottleneck to construct

Table 2 Score sheet of subjective validation

Model	Melody	Rhythm	Harmony of chord	Musical texture	Emotion
RNN–DQN	6.14	5.13	6.21	6.36	6.00
SeqGAN	6.27	7.07	7.36	6.29	6.07
VAE–GAN	5.71	5.93	5.60	6.14	5.27
MCNN	7.86	7.73	8.00	7.93	7.00

a suitable reward function, it is hard to put it into use to generate music based musical theory rule.

Our model perfectly solved these problems. In MCNN model, we treat the probability output of CNN discriminator as a part of the reward of RNN, which takes part in the feedback of LSTM and contributes to the specific music style element in the music. The reward function is constructed based on the musical rules, effectively promoting the music generation. It brings to the result that our generated music is gracefully structured, having the natural melody as well as the strong rhythm.

6 Conclusion

This paper constructs an end-to-end MCNN music generation network model. The reward function is utilized to optimize the probability distribution of music generating network in real time, so as to generate smooth and beautiful classical music. This network makes use of the advantages of AC reinforcement learning algorithm to select the diversity of actions, as well as avoids the drawback of GAN network generating samples, which are too similar, and improves the creativity of style music generation. Experimental results show that about 90 percent of the samples generated by this method are qualified, and 30 percent of the samples can meet people's actual requirements for style music. Therefore, further research is needed to improve the quality of music generation.

This paper discusses music generation algorithm with specific style. In practical applications, music style is often rich, diverse and constantly changing. Therefore, it is of great significance to build music intelligent generation model with multiple styles and themes, which still needs continuous exploration and research.

Acknowledgements We acknowledge the help of Chao Zhang and Xing Wang of Chenda Music Co., Ltd, Beijing.

References

1. van der Zwaag MD, Janssen JH, Westerink JHDM (2013) Directing physiology and mood through music: validation of an affective music player. IEEE Trans Affect Comput 4(1):57–68
2. Li Z, Tang J, Mei T (2019) Deep collaborative embedding for social image understanding. IEEE Trans Pattern Anal Mach Intell 41(9):2070–2083
3. Li Z, Tang J (2015) Weakly supervised deep metric learning for community-contributed image retrieval. IEEE Trans Multimed 17(11):1989–1999
4. Margounakis D, Politis D (2012) Science of networking: new trends in music production, distribution and management. In: Proceedings of international conference on information society, London, pp 486–491
5. Fernández JD, Vico F (2013) AI methods in algorithmic composition: a comprehensive survey. J Artif Intell Res 48:513–582

6. http://magenta.tensorflow.org. Accessed 2016
7. Zhu X, Yang J, Zhang C, Zhang S (2019) Efficient utilization of missing data in cost-sensitive learning. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2019.2956530
8. Yan M, Chan CA, Gygax AF, Yan J, Campbell L, Nirmalathas A, Leckie C (2019) Modeling the total energy consumption of mobile network services and applications. Energies 12(1):184
9. Li Z, Tang J (2015) Unsupervised feature selection via nonnegative spectral analysis and redundancy control. IEEE Trans Image Process 24(12):5343–5355
10. Zhu X, Zhu Y, Zheng W (2019) Spectral rotation for deep one-step clustering. Pattern Recognit. https://doi.org/10.1016/j.patcog.2019.107175
11. Balaban M, Ebcioglu K, Laske O (1992) Understanding music with AI: perspectives on music cognition. In: AAAI Press, pp 186–205
12. Ebcioglu K (1990) An expert system for harmonizing chorales in the style of J. S. Bach. J Log Program 8(1–2):145–185
13. Biles JA (1994) GenJam: a genetic algorithm for generating jazz solos. ICMA 94:131–137
14. Johnson D (2015) Composing music with recurrent neural networks. Blog. http://www.hexahedria.com/2015/08/03/composing-musicwith-recurrent-neural-networks. Accessed 2015
15. Waite E (2016) Generating long-term structure in songs and stories. http://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn. Accessed 2016
16. Hadjeres G, Pachet F, Nielsen F (2017) Deepbach: a steerable model for bach chorales generation. In: Proceedings of ICML, pp 1362–1371
17. Lattner S, Grachten M, Widmer G (2018) Imposing higher-level structure in polyphonic music generation using convolutional restricted Boltzmann machines and constraints. J Creat Music Syst. https://doi.org/10.5920/jcms.2018.01
18. Akbari M, Liang J (2018) Semi-recurrent CNN-based VAE–GAN for sequential data generation. In: Proceedings of ICASSP, pp 2321–2325
19. Roberts A, Engel J, Raffel C, Hawthorne C (2018) A hierarchical latent vector model for learning long-term structure in music. In: Proceedings of ICML
20. Brunner G, Konrad A, Wang Y, Wattenhofer R (2018) MIDI-VAE: modeling dynamics and instrumentation of music with applications to style transfer. In: Proceedings of ISMIR
21. Yang L-C, Chou S-Y, Yang Y-H (2017) MidiNet: a convolutional generative adversarial networks for symbolic-domain music generation. In: Proceedings of ISMIR
22. Dong HW, Hsiao WY, Yang LC, Yang YH (2018) MuseGAN: symbolic-domain music generation and accompaniment with multitrack sequential generative adversarial networks. In: Proceedings of AAAI
23. Yu L, Zhang W, Wang J, Yu Y (2017) SeqGAN: sequence generative adversarial nets with policy gradient. In: Proceedings of AAAI
24. Brunner G, Wang Y, Wattenhofer R, Zhao S (2018) Symbolic music genre transfer with CycleGAN. In: Proceedings of ICTAI
25. Zhu X, Zhang S, Hu R, He W, Lei C, Zhu P (2019) One-step multi-view spectral clustering. IEEE Trans Knowl Data Eng 31(10):2022–2034
26. van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A (2016) WaveNet: a generative model for raw audio. arXiv preprint arXiv:1609.03499
27. Manzelli R, Thakkar V, Siahkamari A et al (2018) Conditioning deep generative raw audio models for structured automatic music. In: Proceedings of ISMIR
28. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Proceedings of NIPS, pp 2672–2680
29. Gross R, Gu Y, Li W, Gauci M (2017) Generalizing GANs: a turing perspective. In: Proceedings of NIPS, pp 6316–6326
30. Grondman I, Busoniu L, Lopes GAD, Babuska R (2012) A survey of actor–critic reinforcement learning: standard and natural policy gradients. IEEE Trans Syst Man Cybern 42(6):1291–1307
31. Li L, Li D, Song T, Xu X (2018) Actor–critic learning control based on regularized temporal-difference prediction with gradient correction. IEEE Trans Neural Netw Learn Syst 29(12):5899–5909
32. LeCun Y et al (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551
33. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
34. Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. arXiv:1410.2329
35. Sak H, Senior A, Beaufays F (2014) Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: Proceedings of INTERSPEECH, pp 338–342
36. Fan Y, Qian Y, Xie F, Soong FK (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Proceedings of INTERSPEECH, pp 1964–1968

37. Malandrino D, Pirozzi D, Zaccagnino R (2018) Visualization and music harmony: design, implementation and evaluation. In: Proceedings of 22nd international conference information visualisation, pp 498–503
38. Chiu S, Shan M (2006) Computer music composition based on discovered music patterns. In: Proceedings of IEEE international conference on systems, man and cybernetics, pp 4401–4406
39. Chathuranga EADY, Ratnayake HUW, Premaratne IA (2017) An expert system to generate chords for melodies composed in eastern music format. In: Proceedings of international conference on computer, communications and electronics, pp 501–504
40. Mnih V, Kavukcuoglu K, Silver D et al (2013) Playing Atari with deep reinforcement learning. In: Proceedings of NIPS, pp 201–220
41. Mnih V, Kavukcuoglu K, Silver D et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533
42. Watkin CJCH (1989) Learning from delayed rewards. Robot Auton Syst 15(4):233–235
43. Van Hasselt H, Guez A, Silver D (2016) Deep reinforcement learning with double Q-learning. In: Proceedings of AAAI, pp 2094–2100
44. Van Hasselt H (2010) Double Q-learning. In: Proceedings of AAAI, pp 2613–2621
45. Quan W, Wang K, Yan D, Zhang X (2018) Distinguishing between natural and computer-generated images using convolutional neural networks. IEEE Trans Inf Forensics Secur 13(11):2772–2787
46. Sun T, Wang Y, Yang J, Hu X (2017) Convolution neural networks with two pathways for image style recognition. IEEE Trans Image Process 26(9):4102–4113
47. Wright C (2013) Listening to music. Nelson Education, Scarborough
48. Dhanalakshmy DM, Menon HP, Vinaya V (2017) Musical notes to MIDI conversion. In: Proceedings of international conference on advances in computing, communications and informatics, pp 799–804
49. Colmenares G, Escalante R, Sans JF, Surós R (2011) Computational modeling of reproducing-piano rolls. Comput Music J 35(1):58–75
50. Jaques N, Gu S, Turner RE, Eck D (2017) Tuning recurrent neural networks with reinforcement learning. arXiv:1611.02796
51. Malkauthekar MD (2013) Analysis of Euclidean distance and Manhattan Distance measure in face recognition. In: Proceedings of international conference on computational intelligence and information technology, pp 503–507
52. Wang L, Zhang Y, Feng J (2005) On the Euclidean distance of images. IEEE Trans PAMI 27(8):1334–1339
53. Greche L, Jazouli M, Es-Sbai N, Majda A, Zarghili A (2017) Comparison between Euclidean and Manhattan distance measure for facial expressions classification. In: Proceedings of international conference on wireless technologies, embedded and intelligent systems, pp 1–4
54. Wiriyachaiporn P, Chanasit K, Suchato A, Punyabukkana P, Chuangsuwanich E (2018) Algorithmic music composition comparison. In: Proceedings of international joint conference on computer science and software engineering, pp 1–6

Affiliations

Cong Jin[1] · Yun Tie[2] · Yong Bai[2] · Xin Lv[3] · Shouxun Liu[4]

Cong Jin
jincong0623@cuc.edu.cn

Yun Tie
ieytie@zzu.edu.cn

Yong Bai
627548583@qq.com

Xin Lv
lvxincuc@163.com

[1] School of Information and Communication Engineering, Communication University of China, Beijing 100024, China

[2] School of Information and Engineering, Zhengzhou University, Zhengzhou 450001, China

[3] School of Animation and Digital Arts, Communication University of China, Beijing 100024, China

[4] Communication University of China, Beijing 100024, China