Music Genre Classification Using Independent Recurrent Neural Network

Wenli Wu
College of Information Science and Technology
Donghua University
Shanghai, China
Email: w_wuwenli@163.com

Guangxiao Song
College of Information Science and Technology
Donghua University,
Shanghai, China
Email: guangxiaosong@hotmail.com

Zhijie Wang*
College of Information Science and Technology
Donghua University,
Shanghai, China
Email: wangzj@dhu.edu.cn

Fang Han
College of Information Science and Technology
Donghua University,
Shanghai, China
Email: yadiahan@163.com

***Abstract*—Genre is one of the most widely mentioned music labels which have a great influence on accuracy of music recommendation. Machine learning is often used to tackle with genre classification task, but the result of the approach heavily depends on the performance of feature extraction. Deep neural network automatically learns advanced features layer by layer, which makes excellent results in many areas. Music signal is sequential and Recurrent Neural Network (RNN) is widely employed for sequential data. Among variant units of RNN, Independently Recurrent Neural Network (IndRNN) can learn long-term relationship better than popular units such as Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). In addition, IndRNN has better computational efficiency. Consequently, multi-layer IndRNN is used as the main part of our model to classify music genres on the GTZAN dataset. In order to keep the information loss as less as possible, scattering transform is used to preprocess the raw music data. The experimental results show that the model achieves a competitive result in music genre classification task compared with the state-of-the-art models.**

Keywords—music genre classification, independently recurrent neural network, deep learning, music information retrieval

I. INTRODUCTION

Automatic music recommendation becomes an indispensable function in music software in recent years. Genre classification of music is quite important for recommendation task [1-3]. Recently, machine learning has a good performance in music genre classification task. However, most of these machine learning methods for instance, Support Vector Machine (SVM) [4] and Decision Trees [5], are heavily depends on the features extracted from the raw music contents. The operation of feature extraction is generally independent of the classification process, so the quality of feature is often affected by many factors. Differently, deep neural network can automatically perform feature extraction layer by layer efficiently [6], and the extracted feature often performs better than that based on manual feature engineering, which makes the final classification result better.

As a model of deep learning, Recurrent Neural Network (RNN) is widely used for sequential data and has the ability of manipulating long-term relationship. Unfortunately, it is well-known that vanilla RNN is difficult to learn long-term patterns due to the gradient vanishing and exploding [7]. Long Short-term Memory (LSTM) [8] and Gated Recurrent Unit (GRU) [9] emerge as RNN variants to solve these problems. But the use of sigmoid and hyperbolic tangent functions in these two variations of RNN may result in gradient decay in deep network. Consequently, the network cannot work in long time scale like long music clip. To solve this problem in practical application of music, we use IndRNN [10] to tackle with genre classification task. IndRNN can learn long-term dependencies better than LSTM and GRU, which is more suitable for music signal processing problem depends on multi-scale features. The problems of gradient vanishing and exploding of IndRNN can be solved by adjusting the time-based gradient back propagation.

Furthermore, we use scattering transform for data preprocessing in order to keep the information loss as less as possible and GTZAN dataset [11] is used to verify the performance of the model in our work. Experimental results show that the network based on IndRNN in this paper can achieve high recognition rate on GTZAN dataset compared with other state-of-the-art models and it has a good performance both in speed and accuracy in music classification task.

II. INDEPENDENTLY RECURRENT NEURAL NETWORK FOR MUSIC GENRE CLASSIFICATION

The overall framework of the method used in this paper is shown in fig.1. The whole process consists of three parts. The first part is using scattering transform to preprocess dataset with preliminary feature extraction. Then, the data is trained using a 5-layer IndRNN with tagged data, which is the core part of the entire work. IndRNN can solve the problem of gradient vanishing and exploding as LSTM and GRU to learn long-term dependencies. Additionally, the trained IndRNN has strong robustness with the Rectified Linear Unit (ReLU) function. At last, we use softmax classifier to complete the final classification task.

*Corresponding Author

978-1-7281-1312-8/18/$31.00 ©2018 IEEE

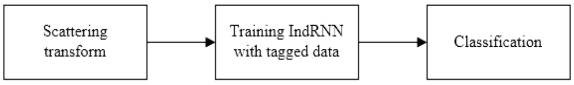

Fig.1. The architecture of the used method in the paper

A. Scattering transform

In most audio classification tasks, long time scales (>500ms) signal representations need to be captured [12]. There are two widely used methods in audio processing. Mel-Frequency Cepstral Coefficients (MFCC) is more efficient only when the feature window length is about 25ms, the same as Mel-spectrogram is that expanding the time scale will lead to severe information loss. Therefore, neither of these methods is suitable for large time-scale signal processing up to 500ms. In addition, the modulation spectrum decomposition utilizes the autocorrelation or Fourier coefficients to represent the process of Mel-spectrum over a longer time scale. However, this modulation spectrum is also affected by the instability of the time warping deformation which has a negative effect on the classification performance. Unlike the methods mentioned above, using scattering transform has better performance on large time scales with less information loss.

Scattering transform can recover the information lost by Mel-frequency with a cascade of wavelet decompositions and modulus operators. According to [13], for an audio signal x , Mel-frequency spectral coefficients can be calculated $\left|x*\psi_{\lambda_1}\right|*\phi(t)$ instead. ψ_{λ_1} represents the wavelets. The set of wavelet modulus coefficients recovers the high frequencies removed by low pass filter $\phi(t)$. Scattering transform is defined as cascading this procedure.

For a signal x , scattering coefficient can be defined as $S_n x$, when n means the order. A time-average operation on x is to obtain a local translation invariant descriptor $S_0 x(t) = x*\phi(\mathrm{t})$, and it removes the high frequencies. These high frequencies can be recovered by wavelet modulus transform $|W_1|$:

$$|W_1|x = \left(x*\phi(\mathrm{t}), \left|x*\psi_{\lambda_1}(t)\right|\right) \tag{1}$$

For audio signals, we define wavelets having the same frequency resolution as Mel-frequency filters. In order to make the wavelet modulus coefficient invariant to the translation, time average unit is used to this transform. The approximated Mel-spectrum spectral coefficients are obtained by averaging the wavelet modulus coefficients with . Therefore, the first-order scattering coefficients are defined as:

$$S_1 x(t,\lambda_1) = \left|x*\psi_{\lambda_1}\right|*\phi(t) \tag{2}$$

For each $\left|x*\psi\lambda_1\right|$ has an operation of a second wavelet modulus transform $|W_2|$:

$$|W_2|\left|x*\psi_{\lambda_1}\right| = \left(\left|x*\psi_{\lambda_1}(t)\right|*\phi, \left\|x*\psi_{\lambda_1}\right|*\psi_{\lambda_2}\right|\right) \tag{3}$$

The lost high frequencies are recovered by the operation of ψ_{λ_2} on wavelet modulus coefficients. These coefficients are all averaged by the same low-pass filter as the first layer as to ensure that each layer of operation has invariance for time-shifts Therefore, the second-layer scattering coefficients are defined as:

$$S_2 x(t,\lambda_{,1},\lambda_2) = \left\|x*\psi_{\lambda_1}\right|*\psi_{\lambda_2}\right|*\phi(t) \tag{4}$$

According to the above, n-order scattering coefficient can defined as:

$$S_n x(t,\lambda_1,\lambda_2,\ldots,\lambda_n) = \left\|\left|x*\psi_{\lambda_1}\right|*\ldots\right|*\psi_{\lambda_n}\right|*\phi(t) \tag{5}$$

B. Independently Recurrent Neural Network

A new variant of RNN—IndRNN [10] is adopted in the genre classification task. RNN has been widely used in the sequence recognition problems such as action recognition [14], language processing, and music auto-tagging [15]. Unlike a feed-forward neural network like Convolutional Neural Network (CNN), RNN has a recurrent connection in which the last hidden state is the input to the next state. The status update can be expressed as follows:

$$h_t = \sigma\left(Wx_t + Uh_{t\text{-}1} + b\right) \tag{6}$$

h_t is defined as hidden status at time step t and $h_{t\text{-}1}$ is the hidden state at the last time step. W and U represent the weights at different stage and b is the basis. σ means the activation function. Due to the continuous multiplication of the recurrent weight matrix, there are serious gradient vanishing and exploding in the training of RNN. So, LSTM was proposed by HochReiter [8] to solve this problem. LSTM is a neural network layer which has four special ways to interact with each other, which is different from the neural network layer of RNN alone. The key to LSTM are the state of cells. LSTM uses the gate structure to add or delete information to the cell state as to protect and control the cell state. Gates are a way of selecting passing information, and their output has a sigmoid or hyperbolic tangent layer. However, the use of hyperbolic tangent and sigmoid as activation functions in LSTM may cause gradient decay in the deep network layer, so the construction of deep LSTM based on RNN is still difficult. Different from the traditional status update method of RNN, the recurrent input is processed with the Hadamard product as:

$$h_t = \sigma\left(Wx_t + \mathbf{u}\odot h_{t\text{-}1} + b\right) \tag{7}$$

The recurrent weight $\mathbf{u}$ is a vector instead of the matrix U in traditional RNN, and $\odot$ represents Hadamard product. It means that each neuron only accepts the input at this moment and its own hidden state at time step t - 1 as input information at time step t . Differently, in the traditional RNN, each neuron at time step t receives the states of all neurons at time step t - 1 as input. Therefore, each neuron in IndRNN can independently process a time and space model. Because neuron in the same layer is independent, the hidden state of the n-th neuron is described as follows:

$$h_{n,t} = \sigma\left(w_n x_t + u_n h_{n,t\text{-}1} + b_n\right) \tag{8}$$

w_n and u_n respectively represent the n-th line of input weight and recurrent weight. Because the units of $h_{t\text{-}1}$ and h_t are independent from each other, which indicates that w is responsible for extracting the spatial characteristics of the input, and u is responsible for extracting temporal characteristics.

The basic architecture of IndRNN is shown in figure 2, where "weight" and "Recurrent" represent the input processing and recurrent processing of each step with ReLU as an activation function.

The connection between neurons is achieved by stacking two or more layers of IndRNNs. A deep IndRNN network can be built by stacking the basic architecture as figure 3.

For the n-th neuron $h_{n,t}$, suppose the objective trying to minimize at time step T is J_n. The gradient back propagated to the time step t is

$$\begin{aligned}\frac{\partial J_n}{\partial h_{n,t}} &= \frac{\partial J_n}{\partial h_{n,T}} \frac{\partial h_{n,T}}{\partial h_{n,t}} = \frac{\partial J_n}{\partial h_{n,T}} \prod_{k=t}^{T-1} \frac{\partial h_{n,k+1}}{\partial h_{n,k}} \\ &= \frac{\partial J_n}{\partial h_{n,T}} \prod_{k=t}^{T\text{-}1} \sigma'_{n,k+1} u_n = \frac{\partial J_n}{\partial h_{n,T}} u_n^{T\text{-}t} \prod_{k=t}^{T\text{-}1} \sigma'_{n,k+1}\end{aligned} \tag{9}$$

The $\sigma'_{n,k+1}$ means the derivative of activation function and the gradient of activation is often bounded in a certain range. What can be seen in the formula is that the gradient only involves the exponential term of the scalar value u_n.

The RNN gradients $\frac{\partial J_{n,T}}{\partial h_{n,\mathrm{t}}} = \sum_{t=0}^{T} \frac{\partial J_{n,T}}{\partial \hat{y}_{n,T}} * \frac{\partial \hat{y}_{n,T}}{\partial h_{n,T}} \prod_{j=t+1}^{T} \frac{\partial h_{n,j}}{\partial h_{n,j-1}}$ depend on the matrix product which is changed significantly even though the change to each matrix entries is small [31]. Comparing these two gradient calculations, the gradient of IndRNN directly depends on the value of recurrent weight, which is just changed by a small size because of the learning rate. Therefore, the results of IndRNN are more robust than a traditional RNN. Furthermore, we only need to control the exponent part of $u_n^{T\text{-}t} \prod_{k=t}^{T\text{-}t} \sigma'_{n,k+1}$ in an appropriate range to solve the problem of gradient vanishing and exploding over time.

IndRNN has several advantages comparing to LSTM. Firstly, using IndRNN can preserve long-term memory and handle long sequences information. The experiments show that the length of the processable sequence (>5000) is obviously superior to LSTM (<1000). Our classification task benefits from the characteristic of IndRNN largely. Secondly, the problems of gradient vanishing and exploding can be effectively solved by adjusting the time-based gradient back propagation. Finally, IndRNN makes good use of non-saturation function such as Relu as activation function, and the trained network has strong robustness.

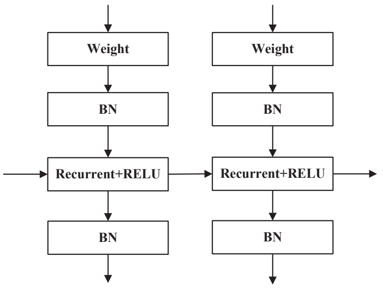

Fig.2. The basic architecture of IndRNN

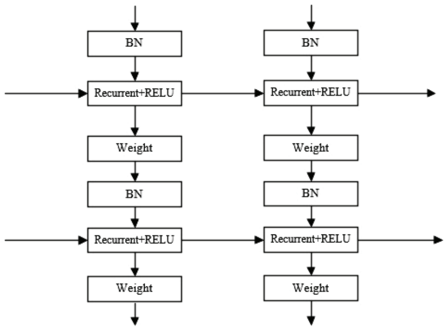

Fig.3. Stacked IndRNN architecture

III. DATASETS AND EXPERIMENT SETUP

The GTZAN dataset is collected by Tzanetakis [16] and is widely used in performance evaluation in the music field. All stereo MP3 audio files are sampled at 16 kHz and converted to mono before being classified. The GTZAN dataset contains ten music classes: Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, and Rock. Each music genre contains 100 clips of 30s duration. The dataset is randomly shuffled into 10 music folders, and nine folders are used for training with the other one for testing. Meanwhile, an average accuracy of 10 times of 10-fold cross-validation is presented for the final test accuracy. Dropout is set as 0.5 and learning rate is set as 1e-5. We stop the training and save the model when the accuracy curve convergence area is stable.

IV. EXPERIMENTAL RESULTS ANALYSIS

Figure 4 shows that the data with scattering transformation converges to a rather high degree of accuracy after 75 epochs in the training process. Figure 5 shows a comparison among the training results of the GTZAN dataset baseds on different typical networks in recent years. We compare network performance based on training time and classification accuracy. It can be seen from the figure 5 that the training time of each iteration of the RNN network is short (0.27s), but it has the disadvantage of the classification accuracy (89%); Meanwhile, the LSTM accuracy reaches high accuracy of 97%, however, its training time per iteration is as long as 0.68s; Differently, we can see IndRNN performed best with both comprehensive training time (0.23s) and classification accuracy (96%). This phenomenon is not only present in the diagram shown, but is consistent in other experiments that have not been shown. In Table 1, we compare our work with some other works in recent years. The experimental results show that the used model excels in music genre classification task.

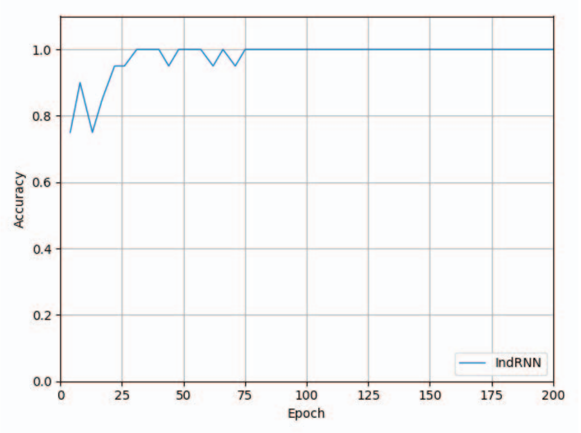

Fig.4. Validation AUC-ROC score of 5-layer IndRNN using scattering transformed input

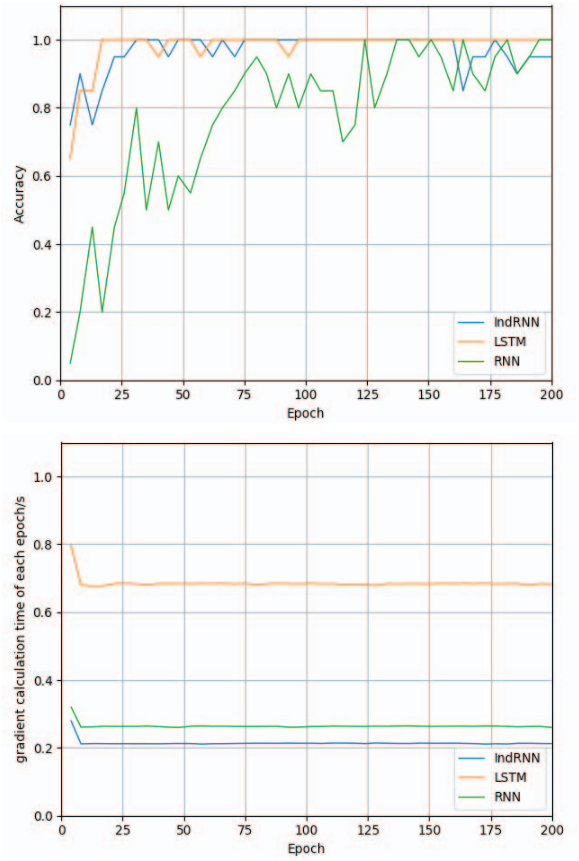

Fig.5. Classification accuracy and time for each iteration among IndRNN、 LSTM and RNN

TABLE 1. AVERAGE TEST ACCURACY OF DIFFERENT MODELS BASED ON GTZAN DATASET

Model	Average test accuracy
Lee,Shih and Yu [12]	90.6%
Andén and Mallat [13]	91.6%
Panagakis and Kotropoulos[17]	92.4%
Dai and Liu [18]	93.4%
IndRNN	96.0%

V. CONCLUSION

In this paper, we use a 5-layer IndRNN with scattering coefficient preprocessing the data to complete the music genre classification work. The experimental results show an excellent performance in term of classification accuracy and training time. Comparing to the state-of-the-art models, our experimental results are very competitive. Therefore, the approach of combining the scattering transform and the IndRNN can well accomplish the classification of music genre on the GTZAN dataset.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grants nos. 11572084, 11472061, 71371046), the Fundamental Research Funds for the Central Universities and DHU Distinguished Young Professor Program (No. 16D210404), the Fundamental Research Funds for the Central Universities (No. CUSF-DH-D-2018097).

REFERENCES

[1] M. Schedl, Y. H. Yang, and P. Herrera-Boyer, Introduction to Intelligent Music Systems and Applications: ACM, 2017.

[2] A. Meng, P. Ahrendt, J. Larsen, and L. K. Hansen, "Temporal feature integration for music genre classification," IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, pp. 1654-1664, 2007.

[3] Y. Song and C. Zhang, "Content-based information fusion for semi-supervised music genre classification," IEEE Transactions on Multimedia, vol. 10, pp. 145-152, 2008.

[4] M. H. Nguyen and F. D. L. Torre, "Optimal feature selection for support vector machines," Pattern Recognition, vol. 43, pp. 584-591, 2010.

[5] Y. Lavner and D. Ruinskiy, "A decision-tree-based algorithm for speech/music classification and segmentation," Eurasip Journal on Audio Speech & Music Processing, vol. 2009, p. 239892, 2009.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436-444, 2015.

[7] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," IEEE Transactions on Neural Networks & Learning Systems, vol. 28, pp. 2222-2232, 2015.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, pp. 1735-1780, 1997.

[9] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Computer Science, 2014.

[10] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN," in Computer Vision and Pattern Recognition, 2018.

[11] B. L. Sturm, "An analysis of the GTZAN music genre dataset," in International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies, 2012, pp. 7-12.

[12] C. H. Lee, J. L. Shih, K. M. Yu, and H. S. Lin, "Automatic Music Genre Classification Based on Modulation Spectral Analysis of Spectral and Cepstral Features," IEEE Transactions on Multimedia, vol. 11, pp. 670-682, 2009.

[13] J. Andén and S. Mallat, "Deep Scattering Spectrum," IEEE Transactions on Signal Processing, vol. 62, pp. 4114-4128, 2014.

[14] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in Computer Vision and Pattern Recognition, 2015, pp. 1110-1118.

[15] G. Song, Z. Wang, F. Han, S. Ding, and M. A. Iqbal, "Music Auto-Tagging Using Deep Recurrent Neural Networks," Neurocomputing, 2018.

[16] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," IEEE Transactions on speech and audio processing, vol. 10, pp. 293-302, 2002.

[17] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Music Genre Classification Using Locality Preserving Non-Negative Tensor Factorization and Sparse Representations," in ISMIR, 2009, pp. 249-254.

[18] J. Dai, W. Liu, C. Ni, L. Dong, and H. Yang, "“Multilingual” Deep Neural Network For Music Genre Classification," in Sixteenth Annual Conference of the International Speech Communication Association, 2015.