Neurocomputing 292 (2018) 104–110

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Music auto-tagging using deep Recurrent Neural Networks

Guangxiao Song, Zhijie Wang*, Fang Han*, Shenyi Ding, Muhammad Ather Iqbal

College of Information Science and Technology, Donghua University, Shanghai 201620, China

ARTICLE INFO

Article history:
Received 14 June 2017
Revised 10 February 2018
Accepted 15 February 2018
Available online 27 February 2018

Communicated by Dr Zhiyong Wang

Keywords:
Music auto-tagging
Deep learning
Music information retrieval
Recurrent Neural Network

ABSTRACT

Musical tags are used to describe music and are cruxes of music information retrieval. Existing methods for music auto-tagging usually consist of preprocessing phase (feature extraction) and machine learning phase. However, the preprocessing phase of most existing method is suffered either information loss or non-sufficient features, while the machine learning phase depends on heavily the feature extracted in the preprocessing phase, lacking the ability to make use of information. To solve this problem, we propose a content-based automatic tagging algorithm using deep Recurrent Neural Network (RNN) with scattering transformed inputs in this paper. Acting as the first phase, scattering transform extracts features from the raw data, meanwhile retains much more information than traditional methods such as mel-frequency cepstral coefficient (MFCC) and mel-frequency spectrogram. Five-layer RNNs with Gated Recurrent Unit (GRU) and sigmoid output layer are used as the second phase of our algorithm, which are extremely powerful machine learning tools capable of making full use of data fed to them. To evaluate the performance of the architecture, we experiment on Magnatagatune dataset using the measurement of the area under the ROC-curve (AUC-ROC). Experimental results show that the tagging performance can be boosted by the proposed method compared with the state-of-the-art models. Additionally, our architecture results in faster training speed and less memory usage.

1. Introduction

In music information retrieval (MIR) area, musical tags are important for artist identification, genre classification or other purposes. Tags represent the high-level information of each music clip, such as instrument (piano, guitar, strings), mood (quiet, soft, weird), genre (classical, rock, jazz) and so on. In the past, they are collected manually by musicians or some music fans. In order to save both time and labor, automatic tagging techniques, namely content-based MIR techniques, have been researched and developed [1].

MIR techniques consist of two phases, preprocessing (feature extraction) and machine learning phase. It is desired that preprocessing phase maintains a delicate balance between feature extraction and information integrity, and that machine learning phase makes use of more information as possible. Most of these existing MIR techniques use traditional machine learning methods such as support vector machine, random forest and decision tree [2–4] to gain complicated relationship from original musical signals to abstract tags. But these machine learning algorithms do not have the capacity of feature extraction. Their performances heavily depend on the performance of the time-consuming features carried out in the preprocessing phase. For example, mel-frequency cepstral coefficient (MFCC) which is widely used in audio processing, is efficient extracting way when using short time scales. When applied to traditional machine learning algorithms, features of short time scales do not perform well in musical tagging task. So, it is necessary to enlarge the scale to make it more suitable for music tagging applications. Another popular transformation way of musical data is mel-frequency spectrogram. It can achieve the purpose of larger scale with stability to time-warping deformation but losing information which has significant influence on the machine learning phase [5–8].

As a powerful and popular learning method, deep learning has successfully been applied in computer vision [9–13], speech recognition [14–16], audio recognition [17] and natural language processing (NLP) [18–20] in recent years. The primary reason of these successes is that deep learning related algorithms can automatically extract high-level features relevant to certain tasks from raw data or processed data. Researchers also have applied deep learning to music auto-tagging task with different preprocessing methods. In [21], a feedforward artificial neural network (multilayer perceptron, MLP) is used for classification. Nam et al. [22] uses unsupervised learning on bag-of-features to initialize a generative stochastic neural network (restricted Boltzmann machine, RBM), then fine-tune the neural network with musical tags. Convolutional

* Corresponding authors.
E-mail addresses: wangzj@dhu.edu.cn (Z. Wang), yadiahan@163.com (F. Han).

https://doi.org/10.1016/j.neucom.2018.02.076
0925-2312/

Neural Network (CNN) gains a lot of success in recognition tasks, such as image classification and speech recognition. Base on inspiration of its outstanding performance in feature extraction, Choi et al. [23] uses deep full convolutional neural networks (FCNs) with mel-spectrogram inputs to deal with music auto-tagging task. Contrast to CNN, which learns high-level features layer by layer from static data, Recurrent Neural Network (RNN) can learn correlations through different time steps well [24,25], especially from sequential data. Musical data are sequential and different kinds of tags need various time scales. Specifically, instrument (guitar, strings, piano) is at the scale of milliseconds and the rhythm, genre (classic, rock, pop) of music is at the scale of seconds and musical mood (slow, quiet, soft) needs longer. Therefore, learning short and long term correlations through time in musical data is important for auto-tagging task. This suggests that RNN architecture is a suitable for musical tagging task potentially.

However, straightforwardly feeding raw musical data to RNN is impracticable because of the limitation of current hardware. To exploit the advantages of deep learning algorithms, musical data have to be shrunk by preliminary feature extraction. This preprocessing should be moderate, and retain useful information as much as possible in order that deep learning algorithms can extract features further contrast to traditional machine learning algorithms, whose performance depends on extracted features heavily. For more compatibly combining the preprocessing with deep learning algorithms, we use scattering transform to reduce the size of musical data in this paper. This method not only retains the stability but also recovers the information lost by a mel-frequency averaging with modulus operators and wavelet decompositions [26]. We believe these advantages can maximize the capacity of deep learning algorithms, albeit the combination of scattering transform and deep learning is rare up to date. Furthermore, Gated Recurrent Unit (GRU), a structure to manipulate the hidden states of RNN, can deal with long-term relationships which is necessary for auto-tagging task.

Base on the discussions above, we propose an architecture combines scattering transform with five-layer RNNs using GRU [27] in this paper. We use sigmoid output layer for the last RNN layer, and binary cross-entropy loss to compute the objective of the training. The test result achieves a competitive score of the area under the ROC-curve (AUC-ROC) on Magnatagatune dataset, which is higher than the state-of-the-art models.

During the experiments, with the intention of finding the best RNN unit and the numbers of layers and hidden states, Long Short-Term Memory (LSTM) [28], Batch-Normalized LSTM (BN-LSTM) [29] units using different hyperparameters have been evaluated as well. For better comparison, CNN with scattering coefficients has been tested. And the proposed algorithm converges to a quite high accuracy rapidly at early epoch and has more efficient training process because of fewer trainable parameters, relative to CNN models.

2. Proposed architecture

Fig. 1 shows the overall structure of the proposed method to tackle music auto-tagging task. Deep learning techniques are often used for automatically feature extraction. But in MIR, raw music data are too huge to existent processors if used as inputs for deep neural networks directly. Therefore, our architecture consists of two parts. One is machine learning part using multilayer RNNs with GRU, because RNN is suitable and powerful algorithm for sequential data, meanwhile, GRU is a variation of gated unit structure which can learn long-term relationships by training process. And multilayer structure can improve the feature extraction and learning capacity further. Considering the situation of the existing hardware, musical data should be reduced. In order to develop the ability of feature extraction as much as possible, we use scattering transform as preprocessing stage. This transform can extract useful features from raw data and recovery the information loss in feature extraction operation at the same time. It makes the raw data decrease to an appropriate size and retains abundant information for deep neural networks. In the next two sections, we describe the scattering transform and multilayer RNN architecture in detail.

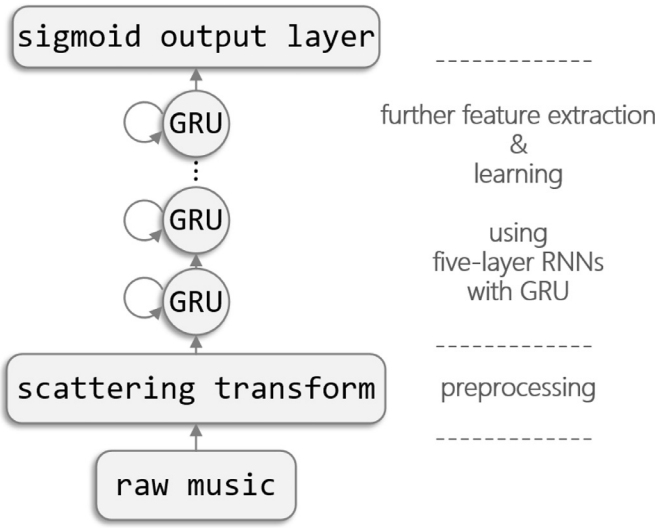


Fig. 1. Overall structure of the proposed architecture.

3. Scattering transform

Choosing scattering transform as preprocessing part is because it can enlarge time scale and recover the information loss, which commonly used preprocessing methods in MIR, such as the mel-frequency spectrogram and MFCC lack. We first describe the calculation process of two-order scattering transform used in our architecture. Then introduce the scattering transform normalizer, in order to reduce redundancy and increase the invariance of scattering coefficients.

The multiplicity of musical information at different time scales is the major difficulty in auto-tagging task. Instrument is at the scale of milliseconds, genre is at the scale of seconds and musical mood needs longer scale. Existing effective and popular used methods in MIR are MFCC and mel-spectrogram. MFCC is efficient at time scales up to 25 ms. And mel-spectrogram can enlarge the scale but cause information loss. The lost information are quite important for many audio applications. So mel-spectrogram is often calculated over small time windows about 25 ms. To enlarge the time scale without too much information loss, we adopt scattering transform as our preprocessing method.

For an audio signal x, Andén and Mallat [5] show that mel-spectrogram coefficients are approximately equal to averaged squared wavelet coefficients $|x \star \psi_{\lambda_1}|^2 \star |\phi|^2(t)$, where ψ_{λ_1} is a bandpass filter of wavelets to handle with high bandwidth frequencies, $\phi(t)$ is a lowpass filter to handle with low frequencies, and $\star$ is convolution operation as it in discrete wavelet transform (DWT). To avoid amplifying the outliers among these coefficients by the square operator, the square is removed and $|x \star \psi_{\lambda_1}| \star \phi(t)$ is computed instead. The information loss emerges here due to the application of the time averaging filter $\phi(t)$. To solve this information loss problem, we recover the lost information by using a modulus of wavelet transform ψ_{λ_2}, formalized as $||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}|$. The previous wavelet modulus coefficients averaged by the lowpass fil-

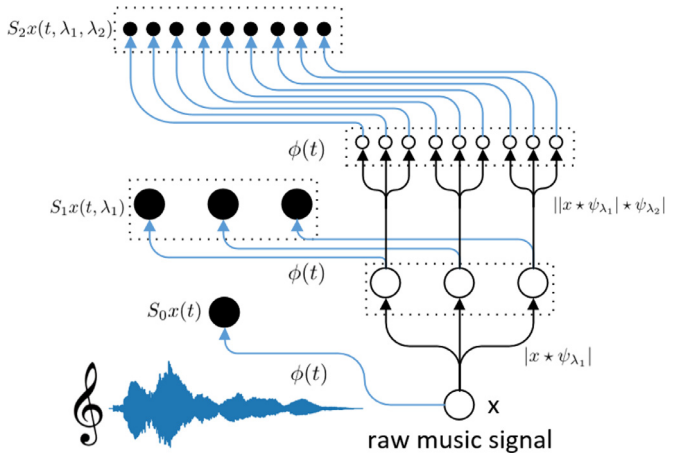


Fig. 2. Process of scattering transform used in proposed architecture.

ter ϕ of size T are called first-order scattering coefficients:

$$S_1x(t,\lambda_1) = |x \star \psi_{\lambda_1}| \star \phi(t) \tag{1}$$

And the second-order scattering transform is applying the same averaging unit $\phi(t)$ on the recovery operation in the last order, defined as:

$$S_2x(t,\lambda_1,\lambda_2) = ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \phi(t) \tag{2}$$

Cascaded operations can be done to form deeper scattering network. But in this paper, two-order scattering transform is used. In addition, applying a $\phi(t)$ to original signal x is called zero-order scattering transform and formalized as:

$$S_0x(t) = x \star \phi(t) \tag{3}$$

The whole process of two-order scattering transform is described by Fig. 2. And to reduce redundancy and increase the invariance of scattering coefficients, we use the same operator named "scattering transform normalizer" as it in [5], defined by the below equation:

$$\widetilde{S}_2x(t,\lambda_1,\lambda_2) = \frac{S_2x(t,\lambda_1,\lambda_2)}{S_1x(t,\lambda_1)}$$

$$\widetilde{S}_1x(t,\lambda_1) = \frac{S_1x(t,\lambda_1)}{S_0x(t)} \tag{4}$$

After two-order scattering transform, the size of raw data can be reduced and low-level invariant features of the original signal are provided without much information loss. Combinations of S_0, S_1 and S_2 scattering coefficients are used as input of our deep neural network.

4. Recurrent Neural Network with gated recurrent unit

In this section, we first introduce the RNN architecture of deep learning algorithms. Then for better manipulating long-term relationships among scattering coefficients, GRU which is used in the proposed architecture is described. Finally, on the purpose of enlarging the representation ability, we stack the RNNs up, forming a multilayer network.

4.1. Vanilla Recurrent Neural Network

RNNs are good for handling with sequential information, such as NLP [18–20], speech recognition [14–16]. RNN structure can be regarded as transitions of hidden states from previous to current ones using fixed transitions. The fixed transition of states is described as a function:

$$h_{t-1}, x_t \rightarrow h_t \tag{5}$$

where h_t, x_t are hidden states and input vectors at time step t, respectively. For vanilla RNN which is the earliest models of neural networks to deal with sequential inputs. Its transitions of hidden states are as follows:

$$h_t = f(W_h[x_t, h_{t-1}] + b_h) \tag{6}$$

where W_h is parameter matrix that connects the input of current time step and the hidden state of last time step. b_h is a bias vector, and f represents the activation function. The output at different time step is transformed from the hidden states connected by another matrix, formalized as:

$$y_t = g(W_y h_t + b_y) \tag{7}$$

where W_y and b_y are the parameter matrix and bias vector, respectively, g is an activation function.

Although vanilla RNNs is simple and effective on numbers of tasks, there are two main problems in their training phase called exploding and vanishing gradients problems [30]. Fortunately, the exploding gradients problem can be solved by gradient clipping efficiently [31]. As for the vanishing gradients, one solution is using Hessian-free optimization algorithms which is more sophisticated than the traditionally used Stochastic Gradient Descend (SGD) [32]. The other is to adjust the structure of neural networks by gated units which are introducing next.

4.2. Gated recurrent unit

The concept of gated units in RNNs is introduced firstly in [28] to overcome the vanishing gradients problems, named Long-Short Term Memory. The vanishing gradients are alleviated by allowing the network to conserve its memory over quantity of time steps during both forward and backward phase. LSTM can be formatted by following equations:

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma(W_i[x_t, h_{t-1}] + b_i) \\ \sigma(W_f[x_t, h_{t-1}] + b_f) \\ \sigma(W_o[x_t, h_{t-1}] + b_o) \\ f(W_g[x_t, h_{t-1}] + b_g) \end{pmatrix}$$
$$c_t = f_t * c_{t-1} + i_t * g_t \tag{8}$$
$$h_t = o_t * f(c_t)$$

where i_t, o_t and f_t are input, output and forget gates at time step t, respectively. σ and f are sigmoid and hyperbolic tangent activation function. g_t is cell updates vector and cell vector c_t is used to update the hidden state h_t. "*" represents the elementwise multiplication.

GRU is a variant of RNN with gated units introduced recently [27]. Similar to LSTM, two gates named z_t and r_t are used to update the hidden state h_t. These gates are formalized as follows.

$$\begin{pmatrix} z_t \\ r_t \end{pmatrix} = \begin{pmatrix} \sigma(W_z[x_t, h_{t-1}] + b_z) \\ \sigma(W_r[x_t, h_{t-1}] + b_r) \end{pmatrix}$$
$$g_t = f(W_g[x_t, r_t * h_{t-1}] + b_g) \tag{9}$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * g_t$$

GRU is found to be comparable to LSTM by experiments on polyphonic music and audio signal modeling [33]. Meanwhile, it has been verified that gated units (GRU, LSTM) are indeed better than vanilla RNNs. Because GRU has less parameters, which implies faster training speed, we choose GRU to deal with the potential long-term correlations of scattering coefficients in the proposed architecture. The dataflows of vanilla RNN, LSTM and GRU are shown in Fig. 3.

4.3. Deep Recurrent Neural Network

To improve the representation ability of the network, we construct the deep RNN model, similar to the model first proposed

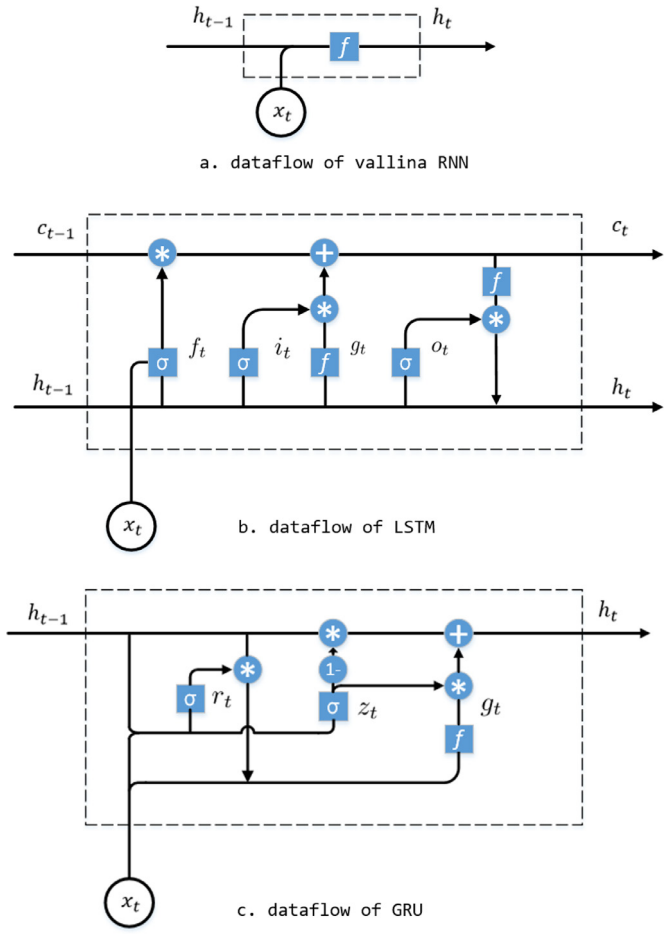


Fig. 3. Dataflows of vanilla RNN, LSTM and GRU.

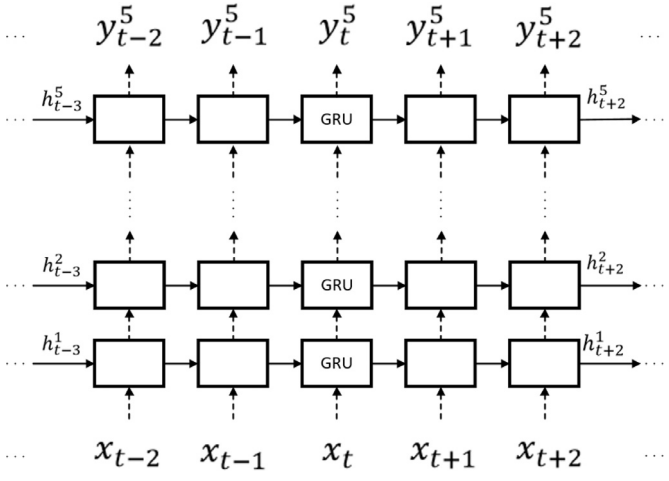


Fig. 4. Structure of RNN for the learning part of the proposed architecture. Each rectangle block represents a unit of GRU and the upper right number means the layer number of hidden state and output.

in [15], by stacking the hidden layers on the top of each other, namely, using the hidden states of the last layer as the input of current layer. In our architecture, we use a five-layer RNNs architecture to learn higher-level features. The structure is displayed in Fig. 4

5. Experiments

5.1. Dataset

To evaluate the performance of our architecture using multi-layer GRUs with scattering transformed inputs, we use Magnatagatune dataset [34] which contains 25,863 clips, and each mono clip lasts nearly 29 s with 16 kHz sample rate. The dataset is annotated with 188 different tags, including genre, instrument, mood and so on. We use top 50 tags according to the frequency of occurrence. These clips are distributed in 16 folders ('0' to '9' and then 'a' to 'f'). The first to the twelfth, the thirteenth, and the rest three folders are for training, validation, test, respectively. After applying a filter of top-50 tags to the divided datasets, we get 21,108 samples, of which we use 14,951 for training, 1825 for validation, 4332 for test. Firstly, trim the audio signals to 29 s. Then we apply the scattering transform to each clip's data as described in Section 3, resulting in each clip with dimensions of 433×114. For a comparison to other well-perform methods, also use log-amplitude mel-spectrograms as another preprocessing way since their better performance compared with STFT and MFCCs [23,35]. 96 mel-bins and 256 hop-size are used to mel-spectrograms. The dimensions of each mel-spectrogram are 1813×96. As supplement, constant-Q transform (CQT) coefficients [36], which have the capacity of representing the fundamental frequencies of notes, have been experimented. 84 bins and 256 hop-size are used to CQT, resulting the dimensions of 1813×84.

5.2. Parameter settings and explored models

In the preprocessing step, the raw music data are converted uniformly to 16,000 Hz and mono. The size of $\phi(t)$ in scattering transform is chosen to be 8192 samples ($\sim$512 ms at 16,000 Hz sampling rate). And ψ_{λ_1} is set as a constant filter-bank with 8 Morlet wavelets per octave which has been demonstrated to represent music signal sparsely in [37]. With these settings, the first order scattering transform can be calculated, which is close to the frequency of mel-frequency filters. In the second order scattering transform, we use the same time averaging operation $\phi(t)$, and ψ_{λ_2} is chosen to be another constant filter-bank with 1 Morlet wavelet per octave. Following a primitive mean and variance normalization is applied straightforwardly to the S_0 and first order scattering features S_1. Scattering transform normalizer for the second order transform is used to reduce redundancy and increase the invariance of scattering coefficients, which is defined by Eq. (4). As discussed above, our preprocessing part, capable of reducing the dimensionality of the raw music data greatly, is composed of scattering transform and normalization operation.

In learning stage, we use five-layer GRUs architecture as discussed in Section 4.3. Sigmoid output layer is on the top layer of our neural network and binary cross-entropy loss is calculated in training phase that benefits the backpropagation speed. Because a model with GRU under the medium number of hidden states (512) leads to the highest AUC-ROC score, we set the number of hidden states as 512 in each layer. Dropout without memory loss [38] is set as 0.75. Exponential decay is applied to learning rate while the start learning rate is 0.00001. We regard scattering transformed data of each clip as a sequential data which have 114 time steps, is a similar way to mel-spectrograms. Tensorflow [39] is used to build our model. Adam optimization [40] is used for training. To reduce the influence of the unbalance in Magnatagatune dataset, we use AUC-ROC score to evaluate the performance of our model.

Not only the proposed model, but also the other models are evaluated in this paper. To explore how much the number of layers and hidden units affect the performance, we test the 4–6 layers and GRU, LSTM, BN-LSTM [29] units, respectively. Both 256 and

Table 1
The configurations of 4-layer CNN architecture.

Layers	Configurations
Scattering transform	Input: $433 \times 114 \times 1$
Conv	$3 \times 3 \times 128$, same padding, strides: 1×1
Max-pooling	2×3
Dropout	0.5
Conv	$3 \times 3 \times 384$, same padding, strides: 1×1
Max-pooling	3×6
Dropout	0.5
Conv	$3 \times 3 \times 768$, same padding, strides: 1×1
Max-pooling	4×6
Dropout	0.5
Conv	$3 \times 3 \times 2048$, same padding, strides: 1×1
Max-pooling	4×4
Dropout	0.5
Full-connected	50×1, sigmoid

Table 2
Results of different models on test set and the number of trainable parameters of each model.

Model	AUC-ROC score	Number of parameters
Scat-4L-256H-LSTM	0.857	2295K
Scat-4L-256H-BNLSTM	0.881	2295K
Scat-4L-256H-GRU	0.896	1724K
Scat-4L-512H-LSTM	0.892	8260K
Scat-4L-512H-BNLSTM	0.849	8260K
Scat-4L-512H-GRU	0.903	6202K
Scat-5L-512H-GRU	**0.909**	7776K
Scat-6L-512H-GRU	0.902	9350K
Mel-5L-512H-GRU	0.794	9209K
Mel-5L-512H-LSTM	0.773	12270K
CQT-5L-512H-GRU	0.775	2482K
CQT-5L-512H-LSTM	0.765	3306K
Scat-4L-CNN	0.904	17365K

512 hidden states each layer are evaluated. Additionally, considering that the model CNNs with mel-spectrogram inputs performs well on the same dataset, we evaluate analogous structure of CNN [23] with scattering coefficients. Parameter settings of the model are listed in Table 1.

6. Results and discussion

For simplicity, the model name in tables and figures are formatted as "preprocessing method-layer number-hidden states number-unit" (Scat represents scattering transform, mel represents log-amplitude mel-spectrograms, CQT represents CQT coefficients).

As shown in Table 2, the proposed Scat-5L-512H-GRU architecture outperforms the other structures in our experiments, and reaches the highest average AUC-ROC score of 0.909 on test dataset. Firstly, we choose LSTM, BN-LSTM, and GRU which are commonly used and effective in variant tasks to explore the best unit with different layers and hidden states. When we use 4-layer models with 256 hidden states, BN-LSTM performs better than GRU and LSTM, due to the operation of batch normalization which reduces the internal covariate shift between time steps [29]. But when we enlarge the number of hidden states to 512, the score of BN-LSTM declines to 0.849, and GRU shows the best performance with the score of 0.903. Meanwhile, the result of LSTM is very close to GRU. Both of theirs ROC-AUC scores rise to about 0.900. An illustration on this phenomenon could be the under fitting of the models with 256 hidden states. For increasing the capacity of the network further, we experiment 5-layer and 6-layer architectures with GRU. Results demonstrate that deeper models are more powerful. But the score declines when we use 6-layer model. The explanation of this situation is that the deeper network needs larger scale of data to fit. Secondly, after the best unit and the numbers of layers and hidden states are found, we use the best setting of RNN to test the performance of mel-spectrogram. Results indicate the scattering transform is more suitable with RNN architecture than mel-spectrogram.

Comparing with works of recent years on the same auto-tagging task and dataset, the proposed architecture shows a competitive ROC-AUC score in Table 3, improves the state-of-the-art score. We do experiment on analogous 4-layer CNN to the structure in [23] with scattering coefficients, in order to compare with the CNN model in this task. Result shows that the CNN model can also learning well from scattering transformed inputs, but the proposed RNN model learns better.

Additionally, GRUs in our experiments have less trainable parameters than LSTMs and BN-LSTMs when using the same number of layers and hidden states as calculated in Table 2, which implies

Table 3
The results of proposed architecture and previous methods on the Magnatagatune dataset.

Model	AUC-ROC score
Scat-5L-512H-GRU	**0.909**
Multi-scale approach [21]	0.898
Mel-4L-CNN [23]	0.894
RBM and bag of features [22]	0.888
Transfer learning [41]	0.88

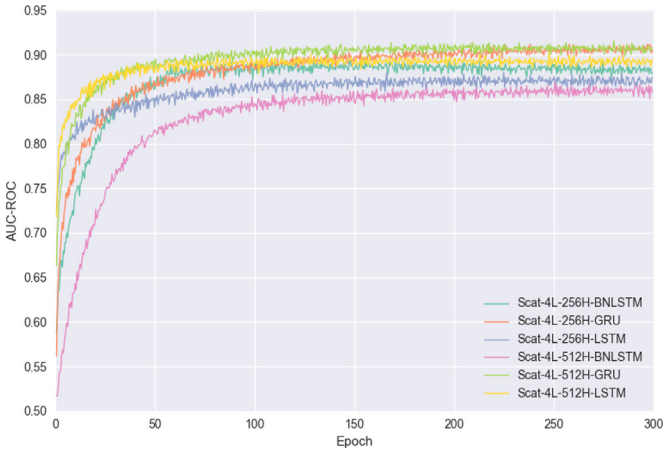


Fig. 5. Training processes of 4-layer RNNs using different number of hidden states and units.

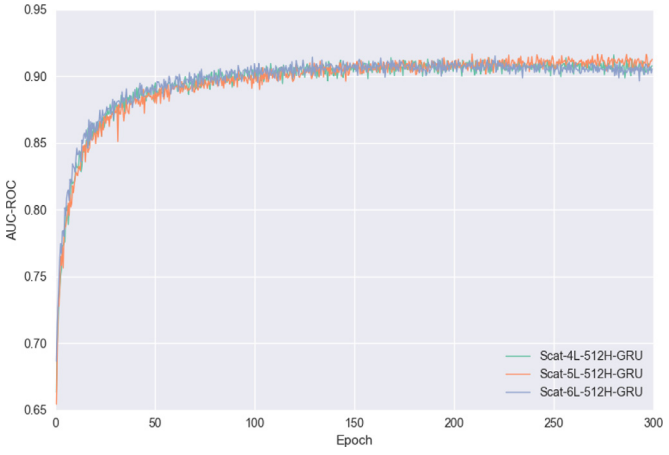


Fig. 6. Training processes of GRUs using different layers with scattering transformed inputs and 512 hidden states.

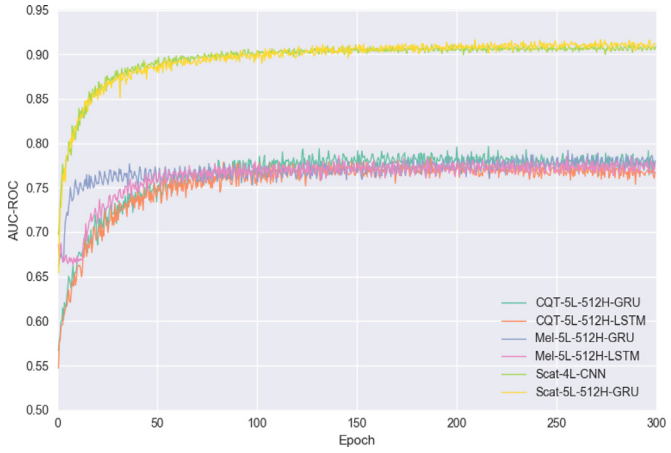


Fig. 7. Comparison of the proposed architecture, similar structure with mel-spectrogram inputs and analogous CNN with scattering transformed inputs.

less memory usage and faster training process. The main reason is that GRU has succincter structure. With the increase of layers, parameters of RNNs grow at the same time. But it is about one third parameters of CNN when using the same number of layers and inputs. Even when the layer number of RNNs increases to 6, the trainable parameters are still less than CNN model. On account of the less number of parameters, the training time of RNNs are shorter than CNN (300 epochs of training on GTX1080 platform, CNN spents about 16 h, and RNNs spent less than 10 h).

Furthermore, RNNs converge rapidly in our experiments, especially in the early 50 epochs. In Fig. 5, different 4-layer RNNs converge to a stable state around 100 epochs. Deeper models show the same behaviors in Fig. 6, while the 5-layer GRU reaches the highest average score. Different with RNNs, the 4-layer CNN model converges slowly mixed with several rapid convergences, then stabilizes around 0.881 in Fig. 7.

7. Conclusion

In this paper, we investigate music auto-tagging task using the proposed five-layer RNNs architecture with GRU, and scattering transform is used as preprocessing method. Different numbers of layers and units such as LSTM, GRU and BN-LSTM also have been evaluated. Then for a better comparison to previous works, we also evaluate a four-layer CNN with scattering transform. Finally, five-layer GRU model and LSTM model with mel-spectrogram inputs verify the fitness of scattering transform for music tagging task. Our results show that five-layer GRU with scattering transform inputs outperforms the previous state-of-the-art approaches on the same Magnatagatune dataset. The proposed architecture gets an average AUC-ROC score of 0.909 on test dataset. At the same time, it performs a rapid convergence, faster training speed and less usage of memory contrast to other structures in experiments. This implies that our algorithm has good capacity of processing musical data. Furthermore, due to the requirement for comparing with other models, we uniform the data length. However, because RNN can deal with arbitrary length of inputs, music with different frequency and duration can be applied to our architecture. In addition, our algorithm could be still well-performed potentially when increasing the number of layers under larger dataset.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 11572084, 11472061 and 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] G. Tzanetakis, P. Cook, Musical genre classification of audio signals, IEEE Trans. Speech Audio Process. 10 (5) (2002) 293–302, doi:10.1109/tsa.2002.800560.
[2] Y. Lavner, D. Ruinskiy, A decision-tree-based algorithm for speech/music classification and segmentation, EURASIP J. Audio Speech Music Process. 2009 (1) (2009) 239892, doi:10.1155/2009/239892.
[3] L. Wang, S. Huang, S. Wang, J. Liang, B. Xu, Music genre classification based on multiple classifier fusion, in: Proceedings of the Fourth International Conference on Natural Computation, ICNC'08, 5, IEEE, 2008, pp. 580–583, doi:10.1109/icnc.2008.815.
[4] M.H. Nguyen, F. De la Torre, Optimal feature selection for support vector machines, Pattern Recognit. 43 (3) (2010) 584–591, doi:10.1016/j.patcog.2009.09.003.
[5] J. Andén, S. Mallat, Deep scattering spectrum, IEEE Trans. Signal Process. 62 (16) (2014) 4114–4128.
[6] D.P. Ellis, X. Zeng, J.H. McDermott, Classifying soundtracks with audio texture features, in: Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, pp. 5880–5883, doi:10.1109/icassp.2011.5947699.
[7] M. Ramona, G. Peeters, Audio identification based on spectral modeling of bark-bands energy and synchronization through onset detection, in: Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, pp. 477–480, doi:10.1109/icassp.2011.5946444.
[8] J.K. Thompson, L.E. Atlas, A non-uniform modulation transform for audio coding with increased time resolution, in: Proceedings of the 2003 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5, IEEE, 2003, pp. V–397, doi:10.1109/icassp.2003.1199990.
[9] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of the 2012 Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[10] Z. Wang, Z. Zhao, S. Weng, C. Zhang, Incremental multiple instance outlier detection, Neural Comput. Appl. 26 (4) (2015) 957–968, doi:10.1007/s00521-014-1750-6.
[11] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778, doi:10.1109/cvpr.2016.90.
[12] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: a review, Neurocomputing 187 (2016) 27–48, doi:10.1016/j.neucom.2015.09.116.
[13] Z. Wang, S. Teng, G. Liu, Z. Zhao, H. Wu, Hierarchical sparse representation with deep dictionary for multi-modal classification, Neurocomputing 253 (2017) 65–69, doi:10.1016/j.neucom.2016.11.079.
[14] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97, doi:10.1109/msp.2012.2205597.
[15] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2013, pp. 6645–6649, doi:10.1109/icassp.2013.6638947.
[16] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio, End-to-end attention-based large vocabulary speech recognition, in: Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2016, pp. 4945–4949, doi:10.1109/icassp.2016.7472618.
[17] Y. Xu, Q. Huang, W. Wang, P. Foster, S. Sigtia, P.J. Jackson, M.D. Plumbley, Unsupervised feature learning based on deep models for environmental audio tagging, IEEE/ACM Trans. Audio Speech Lang. Process. 25 (6) (2017) 1230–1241.
[18] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A Convolutional Neural Network for Modelling Sentences, arXiv preprint:1404.2188 (2014). doi:10.3115/v1/p14-1062.
[19] K.M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, in: Proceedings of the 2015 Advances in Neural Information Processing Systems, 2015, pp. 1693–1701.
[20] J. Chung, K. Cho, Y. Bengio, A Character-Level Decoder Without Explicit Segmentation for Neural Machine Translation, arXiv preprint:1603.06147 (2016). doi:10.18653/v1/p16-1160.
[21] S. Dieleman, B. Schrauwen, Multiscale approaches to music audio feature learning, in: Proceedings of the Fourteenth International Society for Music Information Retrieval Conference (ISMIR-2013), Pontifícia Universidade Católica do Paraná, 2013, pp. 116–121.
[22] J. Nam, J. Herrera, K. Lee, A Deep Bag-of-Features Model for Music Auto-Tagging, arXiv preprint:1508.04999 (2015).
[23] K. Choi, G. Fazekas, M. Sandler, Automatic Tagging Using Deep Convolutional Neural Networks, arXiv preprint:1606.00298 (2016).
[24] C. Hua, S. Wu, X. Guan, New robust stability condition for discrete-time recurrent neural networks with time-varying delays and nonlinear perturbations, Neurocomputing 219 (2017) 203–209, doi:10.1016/j.neucom.2016.09.024.
[25] Z. Wang, J. Wang, Y. Wu, State estimation for recurrent neural networks with unknown delays: a robust analysis approach, Neurocomputing 227 (2017) 29–36, doi:10.1016/j.neucom.2016.07.061.

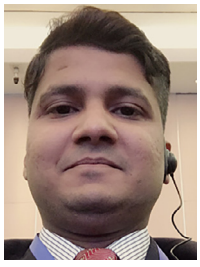
[26] S. Mallat, Group invariant scattering, Commun. Pure Appl. Math. 65 (10) (2012) 1331–1398, doi:10.1002/cpa.21413.
[27] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, arXiv preprint:1406.1078 (2014).
[28] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, doi:10.1162/neco.1997.9.8.1735.
[29] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, A. Courville, Recurrent Batch Normalization, arXiv preprint:1603.09025 (2016).
[30] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Netw. 5 (2) (1994) 157–166, doi:10.1109/72.279181.
[31] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks., ICML 28 (3) (2013) 1310–1318.
[32] J. Martens, I. Sutskever, Learning recurrent neural networks with hessian-free optimization, in: Proceedings of the Twenty-Eighth International Conference on Machine Learning (ICML-11), 2011, pp. 1033–1040.
[33] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint:1412.3555 (2014).
[34] E. Law, L. Von Ahn, Input-agreement: a new mechanism for collecting data using human computation games, in: Proceedings of the 2009 SIGCHI Conference on Human Factors in Computing Systems, ACM, 2009, pp. 1197–1206, doi:10.1145/1518701.1518881.
[35] S. Dieleman, B. Schrauwen, End-to-end learning for music audio, in: Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2014, pp. 6964–6968, doi:10.1109/icassp.2014.6854950.
[36] C. Schŋrkhuber, A. Klapuri, Constant-q transform toolbox for music processing, in: Proceedings of the Seventh Sound and Music Computing Conference, Barcelona, Spain, 2010, pp. 3–64.
[37] E.C. Smith, M.S. Lewicki, Efficient auditory coding, Nature 439 (7079) (2006) 978–982, doi:10.1038/nature04485.
[38] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving Neural Networks by Preventing Co-adaptation of Feature Detectors, arXiv preprint:1207.0580 (2012).
[39] M. Abadi, A. Agarwal, P. Barham, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467 (2016).
[40] D. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, arXiv preprint: 1412.6980 (2014).
[41] A. Van Den Oord, S. Dieleman, B. Schrauwen, Transfer learning by supervised pre-training for audio-based music classification, in: Proceedings of the 2014 Conference of the International Society for Music Information Retrieval (ISMIR 2014), 2014.

Guangxiao Song received the M.S. degree in Computer Science and Technology from Yangtze University, China, in 2016. Since September 2016, he has been a Ph.D. candidate in information science and technology at Donghua University, China. His research interests include deep learning and music information retrieval.

Zhijie Wang received his B.E., M.E. and Ph.D. degrees from College of Information Science and Technology, Donghua University, Shanghai, China, in 1991, 1994 and 1997, respectively. He did his Post Doc from the University of Tokyo, Tokyo, Japan in 2002. He is currently a Professor with the College of Information Science and Technology, Donghua University, Shanghai, China. His main research interests are computational neuroscience, neural network, and deep learning.

Fang Han received her B.S. and M.S. degrees from Beijing Jiaotong University, Beijing, China, in 2003 and 2006, respectively, and the Ph.D. degree from Beihang University, Beijing, China, in 2009. She visited Aberdeen University, Aberdeen, UK for one year as a visiting Ph.D. student in 2008 and New York University, New York, USA as a visiting scholar for one year in 2016, respectively. She is currently an Associate Professor with the College of Information Science and Technology, Donghua University, Shanghai, China. Her main research interests are computational neuroscience and deep learning.

Shenyi Ding received the B.S. degree from College of Information Science and Technology, Donghua University. Since September 2015, he take a successive postgraduate and doctoral program in information science and technology at Donghua University, China. His research interests include deep learning and dynamic system modeling and control.



Muhammad Ather Iqbal received his B.E. and M.E. degrees in Electronics Engineering and Industrial Controls and Automation from Usman Institute of Technology (UIT), Karachi, Pakistan with high honors, in 2009 and 2012, respectively. He is currently enrolled as a Ph.D. student in the College of Information Science and Technology at Donghua University, Shanghai, China. He is currently working on developing computer vision and deep learning algorithms. His main research interests are image processing and analysis, pattern recognition, and deep neural networks for machine learning.