Improved Reinforcement Learning Game for Score Following

In Partial Fulfillment of the Requirements for the Degree of Master of Science

Module	Dissertation
Major	Social Computing
Author	Minglang Tuo
Teacher	Dr. Shengchen Li
Date	15^{th} / Dec / 2021

Contents

Abstract

Recently years, with the development of artificial intelligence, people tend to insert it into other interdisciplinary fields. In education area, how to judge the level of students' playing music is also an interesting problem. This report will illustrate the score following problem, which means system will score the performer through reinforcement learning according to recognized sound. In this project, the datasets are distinct to train neural network. More specifically, one is from various symphonies songs and another is from single instrument songs. Then the state of art of reinforcement learning technology will be implemented, including policy-based algorithm and actor–critic algorithm. Moreover, the experiments will be discussed to evaluate algorithms from accuracy and robustness. Finally, the relevant parameter will be analysed.

Key Words: Audio Processing, Reinforcement Learning, Score Following, Continuous Control Problem

Chapter 1

Declaration

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.
I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,
Minglang Tuo

Chapter 2

Acknowledgements

The master's career is coming to an end soon. Looking back on the whole process, I feel both joy and regret. First of all, I would like to thank Dr. Li for his encouragement and guidance. During his teaching work, he took time to discuss academic content with me, gave me a lot of references to support my thesis, and discussed relevant career planning with me. In addition, I would like to thank my classmates and friends who have accompanied me for many years. I would also like to thank them for their useful suggestions and comments. With their support, encouragement and help, I can spend one year and a half completely.

Chapter 3

Introduction

3.1 Introduction

As people all known, the task of music score tracking system is to track the information related to its audio signal and image signal. Referring to the previous literature, this problem belongs to audio information retrieval. In the past, the methods to realize relevant tasks are online, such as automatic page turning and automatic accompaniment of music score. For this problem, the traditional solution is to symbolize the music score, that is, manually convert the music score into computer-readable forms, such as musicl-xml format and MIDI format, and then use preety-midi Library converts audio files to the same format and compares them. However, this method is unreliable, not only time-consuming and labor-consuming, but also can not recognize complex music, and the accuracy is very low. In order to optimize related problems, a new multimodal deep neural network is proposed. Compared with the traditional method, it can directly learn to match music score and audio in an end-to-end way. More precisely, through the short extract of a given audio and the corresponding score training, the neural network will try to predict the position where the given score image best matches the current audio extract. However, this method has defects, that is, continuous time processing is independent of each other, which will lead to jump in score tracking and greatly reduce the robustness of prediction.

Inspired by the above literature, researchers try to use a fundamentally different machine learning method, reinforcement learning, to try to solve related problems and see what different changes have been made. researchers will interpret score tracking as a multi-mode control problem. The core idea is that the agent can cope with the current playback performance by reading speed, so as to navigate in the score.Then, researchers try to design experiments of two different reinforcement learning algorithms, and compare their robustness and accuracy to demonstrate that reinforcement learning has a significant improvement in solving this problem. After that, people will design relevant experiments to demonstrate which branch of reinforcement learning algorithm has a more significant effect in solving the problem, and verify that the actor critic has a significant improvement compared with other algorithms. Finally, researchers analyze the relevant parameters.

In Chapter 3, researchers will explain the background knowledge of project implementation. In Chapter 4, researchers will explain how to apply the latest reinforcement learning to the project implementation process. The relevant experiments to compare the tracking effect will be designed in Chapter 5. In Chapter 6, researchers will use the method of relevant data analysis to obtain the optimal parameters. The project and look forward to the future will be summarized in last Chapter.

Chapter 4

Background

4.1 Score Following Problem

The establishment of automatic music score tracking system is the core to solve the problem of music score image tracking. It can represent music performance according to known symbols. The earliest solution is to establish music formats such as musicxml and MIDI. Computers can recognize information by reading relevant symbols. More specifically, the method can create music scores manually or recognize them with optical music software[1]. However, in the face of complex music, such as orchestral music, the accuracy and robustness often decline. In recent years, in order to avoid the relevant situation, someone has proposed a multimodal depth neural network[2], which can recognize music score and audio from end to end to learn and match. Through the excerpts and corresponding scores in different stages, the network will predict which position is the best match with the current audio. This task can be expressed as a multi-mode localization task. However, this method has a major defect, that is, there is no way to deal with the independent continuous time performance frequency bands, which will induce jumps in the tracking process, especially when there are repeated paragraphs. Others have proposed improvements. By learning the joint embedding space of score segments and audio segments, the results of different modes can be observed, such as cosine distance. This method can learn the cross modal similarity measurement and calculate the offline alignment between audio and score through dynamic time warping[3].

4.1.1 Optical Music Recognition

For identifying music score documents, researchers will introduce optical music recognition (OMR), and the format of music score image will change and can be read by computer. However, as researchers all know, the recognition of handwritten music score is still an open problem, such as notation. The literature only focuses on the specific stage of this method. Some people propose to use convolution recurrent neural network, data enhancement and transfer learning to construct music recognition (HMR) system[4].

To solve this problem, because many music scores are written with sticks, Long term

short term memory (LSTM) recurrent neural network (RNN) will read each note sequence in turn[5]. Like text recognition, our architecture can be described in the figure.4.1. The network can extract information directly from image pixels, and uses convolutional neural network (CNN) as image feature extraction tool[6].

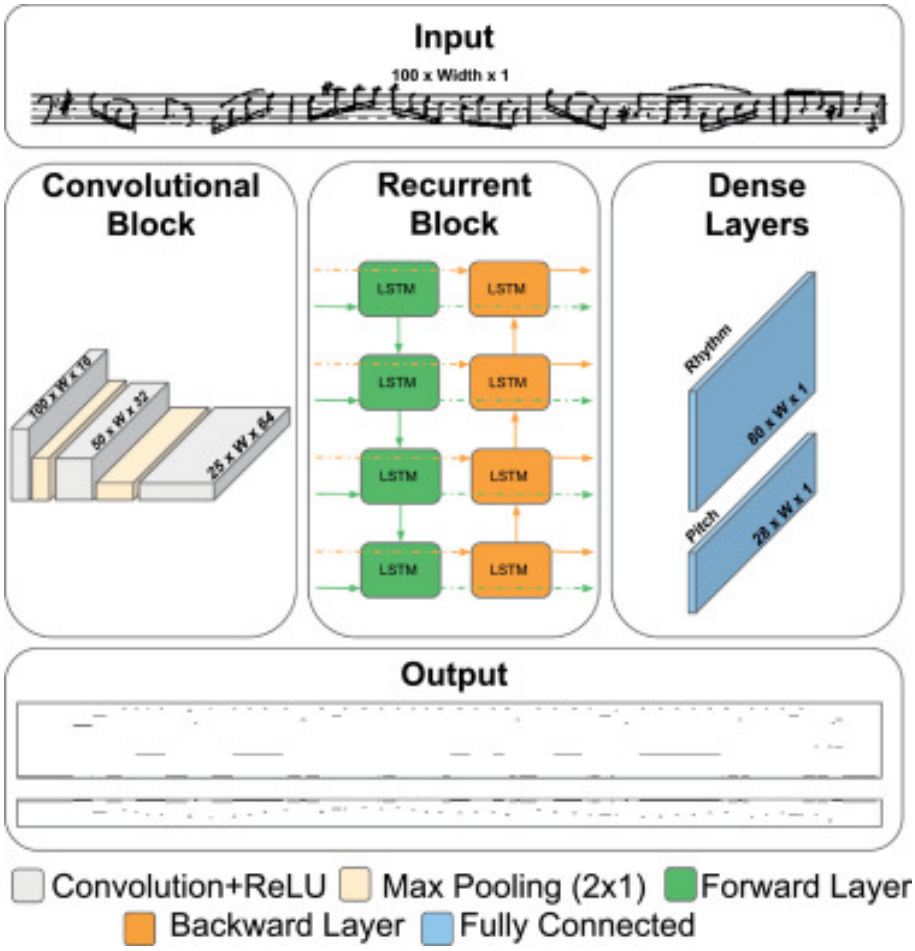


Fig. 4.1: Optical Music

4.1.2 Deep Neural Network Predicting Music

For the matching problem between short music audio clip and corresponding pixel position in music score image. Some people have tried to use a system that can learn to read notes, listen to music and match the currently playing music with its corresponding score notes at the same time. It consists of an end-to-end multimodal convolutional neural network, which takes the score image and the spectrum of the corresponding audio segment as the input. It learns to predict the corresponding position in the corresponding score line for a given invisible audio segment (covering about a bar of Music).

The following figure.4.2 introduces the multi-modal neural network through the mapping of audio and music score[7]. Two special convolution neural networks are established, one as music score image and the other as audio input. The network will output part of

the prediction classification bucket, that is, the area corresponding to the audio segment in the prediction worksheet image.

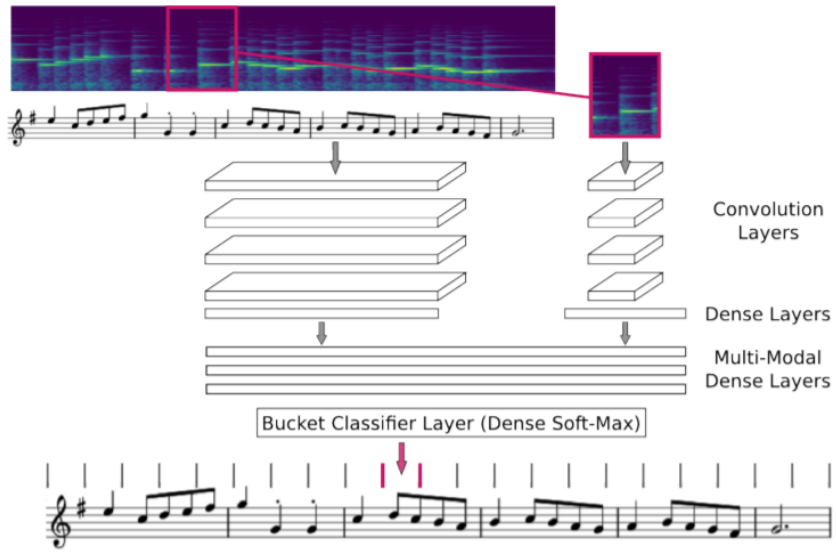


Fig. 4.2: Deep Neural Predict Music

4.2 Markov Chain

This part will talk about the mathematical background of the project named Markov Chain[8], including the reinforcement learning algorithms, encoder and problems three parts. Markov chain, the mathematical basis of reinforcement learning is Markov decision-making process. In this problem, the score tracking agent is an active component interacting with its environment. Interaction occurs in a closed loop in which, The environment makes the agent face a new situation (state st), the agent must make a decision, such as whether to increase, maintain or reduce its scoring speed. After that, the agent will get the next state and reward, representing how well it has done in finishing the whole goal. By running the interaction cycle, researchers get a series of States, actions and rewards $S0, A0, R1, S1, A1, R2, S2, A2, R3...$ Which is a The agent learns from the experience of his behavior.

MDP can be defined as a tuple:

$$M = S, A, R, T, \gamma, H \tag{4.1}$$

1. S - state spaces$\mathbf{s}_t \in S$, state space for score tracking can be defined in two positions. One is the tracking position for audio, another is the score position for image.

2. A - action space, Actions represent increasing, maintaining or reducing its scoring speed.

3. R - The reward function is to give corresponding rewards according to one-to-one correspondence between status and action.

4. $R : S \times A \longrightarrow \mathbb{R}$. In a certain state, choosing a specific action to improve or worsen the current problem is reward.

5. T - transition function$T(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ that governs transition dynamics from one state to another in response to action. In score following setting transition dynamics is usually deterministic and known in advance.

6. γ - scalar discount factor, $0 < \gamma \leq 1$. The of short-term benefits and long-term benefits are affected by agent's discount factors.

7. H - horizon, the sequence $\{s_t, a_t, s_{t+1}, a_{t+1}, s_{t+2}, \dots\}_{t=0}^{H}$ can be defined as the length of the event. The incremental method is used to solve the score tracking problem. The event length has been clearly defined in the operand before the problem. Some criteria have also been introduced for when to end.

In Markov decision making, the agent optimizes the strategy of expected cumulative discount reward according to the policy function $\pi(s)$ through state mapping action:

$$(\pi)^* = \underset{\pi}{argmax} E\,[\,\sum_{t=0}^{H} (\gamma)^t\, R\,(\,s_t, a_t\,)\,]\,, \tag{4.2}$$

In the music score following game, after defining MDP, determine how the agent calculates the maximum value of action through the optimal policyπ^* which is value-based method $Q^\pi(s, a)$. Through the policy π corresponding to the given state s and the expected reward, after performing the relevant actions a, the agent selects the operation to maximize the strategy $Q^\pi(s, a)$ based on the current state. However, Policy-based methods directly models the agent's policy as a parameter function $\pi_\theta(s)$, and maximizes the final reward through the previous decision experience (the result of the agent's previous exploration in the environment), so as to optimize the parameters θ.

As can be seen from the above figure, RL algorithm depends on taking the state of MDP as input and outputting action value or action function. The status represents some information about the problem, such as a given audio signal and picture signal, while the Q value or action is a number. Therefore, the RL algorithm must include an encoder, that is, a function that encodes the state into numbers. To sum up, figure.4.3 shows a pipeline for solving the score tracking problem of RL. The collaborative problem is first reformulated according to the MDP, that is, researchers define the status, action and reward of a given problem. Then, researchers define the state encoder, That is, the input state is encoded and the digital vector is output (Q value or probability of each action). Then there is the RL actual algorithm. The agent who learns the encoder parameters will make a decision on MDP. After the agent selects an operation, the environment will move to a new state,

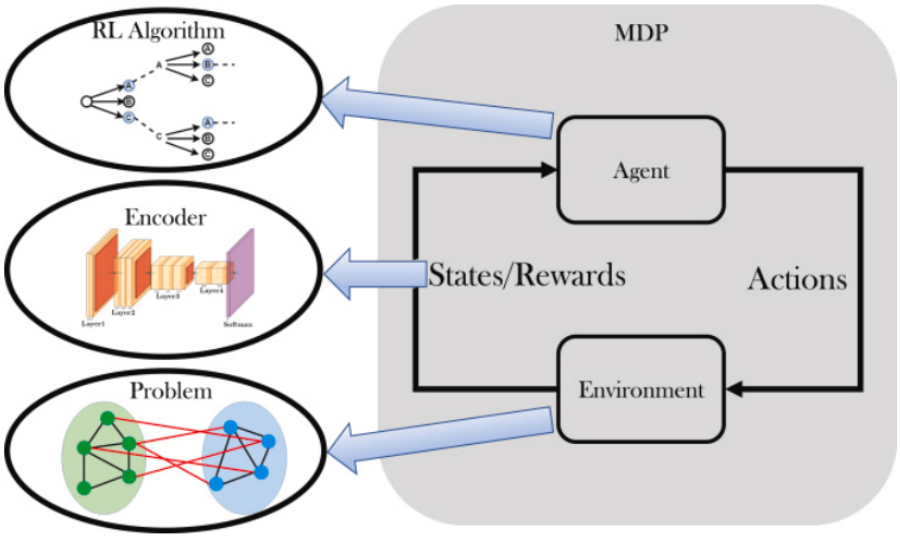


Fig. 4.3: RL Structure

and the agent will receive a reward for its operation. Then, the process will start from the new state within the allocated time budget State repetition. Once the parameters of the model are trained, the agent can search for the solution of the unknown instance of the problem.

4.2.1 Reinforcement Learning Algorithms

For the application analysis of the two algorithms, the reinforce algorithm is mainly policy based algorithm, while A2C is an algorithm combined with policy value Both of them belong to the model free algorithms[9], which uses the experience of previous agent training rather than availability of the transition functions of the environment.

1. Reinforce: Baseline $b(s)$ mitigates the difference in revenue estimates by trying to reduce the difference $A(st, at)$, through the current strategy π_θ can be calculated, but the beginning of training may lead to poor performance because of the initial parameters. However, through the REINFORCE algorithm proposed by Williams (1992)[10], the baseline $b(st)$ can be excluded from the calculation of income estimation. You can also use the average reward on the sampling trajectory and the parameter value function estimator $V_\phi(s_t)$ to calculate the baseline value.

2. Actor–critic algorithms: The family of Actor–Critic (A2C, A3C) (Mnih et al., 2016)[11] are proposed recently. The most typical example of the algorithm is to calculate the return estimation of different steps through the parameter value function to score the previous state, and complete the update of the state value through self supervision,which means estimating from the values of the subsequent states. It is an

extension of the baseline reinforcement learning algorithm within baseline:

$$\hat{A}\left(s_t, a_t\right) = r\left(s_t, a_t\right) + V_\phi\left(s_t^{'}\right) - V_\phi\left(s_t\right) \tag{4.3}$$

Although this approach introduces bias to the gradient estimates, it often reduces variance even further. Moreover, the actor–critic methods can be applied to the online and continual learning, as they no longer rely on Monte-Carlo rollouts, i.e. unrolling the trajectory to a terminal state.

4.2.2 Encoder

PyTorch (PyTorch 2021) is a machine learning library for neural network in Python[12]. It is open-source used for applications in natural language processing (NLP) and computer vision. To define a neural network in PyTorch, it is pretty convenient to create a class that inherits from nn.Modules within several rows of code. In the created class, user can define the layers of the network in the init function and indicates how data will pass through the network in the forward function. The specific neural network construction will be described in the next chapter.

4.2.3 Score Following Problem

The mathematical foundation for reinforcement learning is Markov Decision Process in figure.4.4. In this problem, the score following agent is the active component that interacts

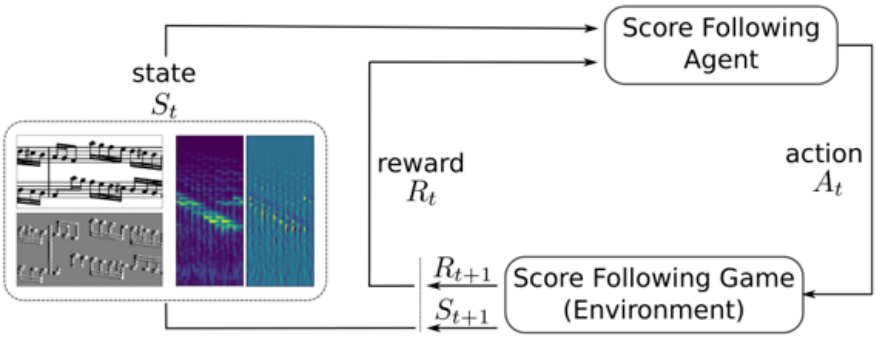


Fig. 4.4: Score Following Problem1

with its environment. In the process of reinforcement learning interaction, a closed loop is formed, and the agent will encounter a new situation (state st). In the environment, the agent must immediately make corresponding decisions to solve the problem, like deciding whether to increase, keeping or decreasing its speed in the score. After that, the agent will get the next state and reward indicating how well it's doing in achieving the overall goal. In a series of interactive cycles, $S0, A0, R1, S1, A1, R2, S2, A2, R3,$, the agent has a more corresponding state and learns to make decisions by itself to obtain feedback. Then it affects the environment, and the environment will feed back.

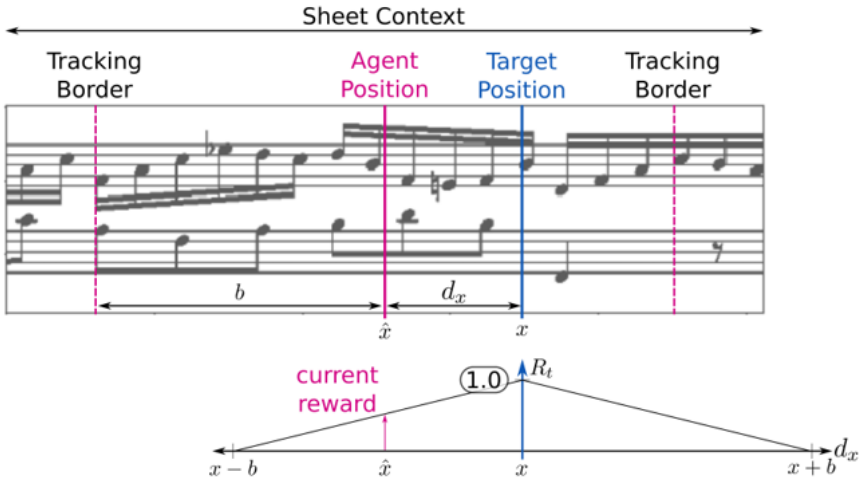


Fig. 4.5: Score Following Problem2

In the process of score tracking, now, the most important part of Markov process is reward. This figure.4.5 summarizes our reward calculation for each start time in the audio in the score after MDP, that is, the real target position X in the score. Based on the agent's current location x, researchers calculate the current tracking error as a formula, The reward signals RB to $x+B$ near the target position X are defined as a formula in the predefined tracking window between ($r = 1.0|dx|/b$). Therefore, when the agent's position is the same as the target position, the reward for each time step reaches its maximum value of 1.0. As the reward decays linearly to 0.0, the tracking error will also reach the maximum value B allowed by the window size. When the absolute tracking error exceeds B (the agent exits the window), researchers reset the score after the game (back to the beginning of the score, the first audio frame). The sole goal of the agent is to maximize the cumulative return in the future. It will learn to match the correct position in the score and will not lose its goal by jumping out of the window.

4.3 FluidSynth

Fluidsync is a real-time software synthesizer. It is a powerful API based on soundfont 2. It has no graphical user interface. You can write corresponding programs to call it. For example, in embedded systems, you can use its methods in some mobile applications. researchers will need it to synthesise the audios from MIDI[13]. Synthesising the audios and computing spectrograms will take a while but it is done only once when training an agent for the first time.

4.4 Evaluation Standard of Experiment

As for the criteria for experiment, researchers divided into two aspects. One is test the accuracy and another is robustness. researchers use some statistics calculation to compare two algorithms.

4.4.1 Accuracy

From the figure.4.5, based on the agent's position x and the ground truth position x, the tracking error dx will be calculated. In fact, two parameters are introduced into the final experimental evaluation system, namely absolute tracking error $|dx|$ and standard deviation tracking error $std(|dx|)$, which can be used to evaluate the agent's tracking accuracy.

4.4.2 Robustness

As for the Robustness, researchers will mainly introduce two groups of formulas.

$$R_{\text{on}} = \frac{\#onesets\ tracked\ within\ window}{\#\text{onsets}} \tag{4.4}$$

$$R_{\text{tue}} = \frac{\#piece\ tracked\ until\ the\ end}{\#\text{pieces}} \tag{4.5}$$

The start of the trace is the start of the onset count, if the agent does not exit the trace window. In a fragment, if all onsets are tracked, which means the agent does not exit the tracking window, the fragment is tracked to the end. For the robustness of the agent, researchers use two parameters to track, like the ratio of Ron $\in$ [0, 1] of overall tracked onsets and the ratio of pieces Rtue $\in$ [0, 1] to the whole song.

4.5 Code Metadata

This project was built under ubuntu environment with Python 3.6. The information of libraries and their usages are listed in Table 2. In the GitHub repository, The source code files (.py) are stored in the src directory, including two main parts, which are training reinforcement agent(experiment.py) and evaluating agent($evaluate_agent.py$). An additional file $test_agent$(.py) for visualization is used for some figures in this report. The dataset files (.mid) generated from feature extraction part are located at data directory. The models (.pt) generated from prediction part are located at the model directory. Their envioniment can be viewed in the figure.4.6.

Module	Version	Purpose
Scipy	1.1.0	Scientific computing tools, used to optimize multi-dimensional arrays and matrices, linear operations, etc
Numpy	1.14.3	
Torch	1.3.1	
Seaborn	0.8.1	Draw statistical pictures, such as box chart, bar chart, tree chart, etc
Matplotlib	2.2.2	
OpenCv	3.4.3.18	Computer vision processing library plays a role in music score recognition
Gym	0.15.4	Reinforcement Learning Algorithm library generate alignment environiment
PySoundFile	0.9.0	Audio Libraries. Functions are music and sound signal processing and analysing.
Madmom	0.15.1	
Librosa	0.6.3	
PyFluidSynith	1.2.5	
Pretty_midi	0.2.8	
Nose	1.3.7	Python Test library, including unit test and interface test for system.
Mock	2.0.0	

Fig. 4.6: Library

Chapter 5

Method

5.1 Data Extraction

First, researchers downloaded the Nottingham database, which contains 296 monophonic melody songs. researchers extracted some relevant data sets from Eric Foxley and published them in Eric Foxley's music database, which contains the collection of British and American Ballads (Hornpipe, jig, etc.). The music format is ABC music symbol format and published in SourceForge. I tried to correct the beat lost during transition and repetition in the data set. The structure of data set can be viewed in figure.5.1.

The format of the data can be regarded as a directory, in which there are directories of various music clips. The name comes from mutopia. For each directory, there are two subdirectories performance / and scores /. More specifically, performance and score are subdirectories of the corresponding music directory, with coding information and derived features. The score is in PDF format and the performance is midi.

5.2 Alignment of score

The file coding of the data set can be divided into multiple parts. The abstract music entity can extract lilypond's file coding from mutopia. The coding reflects music score (visual dynamics) and audio (music score), from which researchers can extract the attempt and performance characteristics of relevant scores. In the end, noteheads in the score to note events in the performances are aligned. The figure.5.2 reflects the whole process of processing.

For each sub music directory, there is a basic Lavender * Ly file, a MIDI generated file, and a configuration file YML, which contains the information of sub music clips, such as the number of aligned notehead / Note event pairs Then, for the subdirectories in performance / and scores /, the fragment saves the relevant generated performance and scores. For MIDI files derived from MIDI of songs, performance authorization codes are generated. In the performance, the audio file is composed of piano sound font, and the spectrum diagram is calculated from the audio. Then the audio file is discarded, and it

```
BachCPE__cpe-bach-rondo__cpe-bach-rondo/
  BachCPE__cpe-bach-rondo__cpe-bach-rondo.ly
  BachCPE__cpe-bach-rondo__cpe-bach-rondo.norm.ly
  BachCPE__cpe-bach-rondo__cpe-bach-rondo.midi
  meta.yml
  scores/
    BachCPE__cpe-bach-rondo__cpe-bach-rondo_ly/
      BachCPE__cpe-bach-rondo__cpe-bach-rondo_ly.pdf
      coords/
        notes_01.npy
        notes_02.npy
        notes_03.npy
        systems_01.npy
        systems_02.npy
        systems_03.npy
      img/
        01.png
        02.png
        03.png
      mung/
        01.xml
        02.xml
        03.xml
  performances/
    BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-900_YamahaGrandPiano/
      BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-900_YamahaGrandPiano.midi
      features/
        BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-900_YamahaGrandPiano_midi.npy
        BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-900_YamahaGrandPiano_notes.npy
        BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-900_YamahaGrandPiano_onsets.npy
        BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-900_YamahaGrandPiano_spec.npy
    BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-1000_ElectricPiano/
      BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-1000_ElectricPiano.midi
      features/
          BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-1000_ElectricPiano_midi.npy
          BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-1000_ElectricPiano_notes.npy
          BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-1000_ElectricPiano_onsets.npy
          BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-1000_ElectricPiano_spec.npy
```

Fig. 5.1: data structure

will not exist in the file structure. After that, the performance file will be added with spectrum, rhythm change beat and sound font After that, the audio sound features and playing sound features will be stored in the file directory: features / subdirectory The following relevant parameters researchers calculated:

1. Spectrogram
2. Onsets list
3. Note events list
4. MIDI matrix

For the core frame by frame function (spectrum diagram to track MIDI matrix), 20 frames per second is the set frame rate. The binary matrix of 128 x n frames is defined as the MIDI matrix. If the pitch is active in the frame, the matrix unit contains 1 After pairing the corresponding note-on and note-off events, the performance MIDI derive the note event, which is a N_ EVENTS x 5 numpy array. Pitch, duration(in seconds), track and channel and onset time(in seconds) are the columns of numpy. The vector with length n is onesets, and the annotation event will be mapped to the start frame For the spectrum, it is a 92 x n frame matrix calculated from the synthetic audio. About note events, the relationship between MIDI matrix and spectrum diagram is about this. The calculated

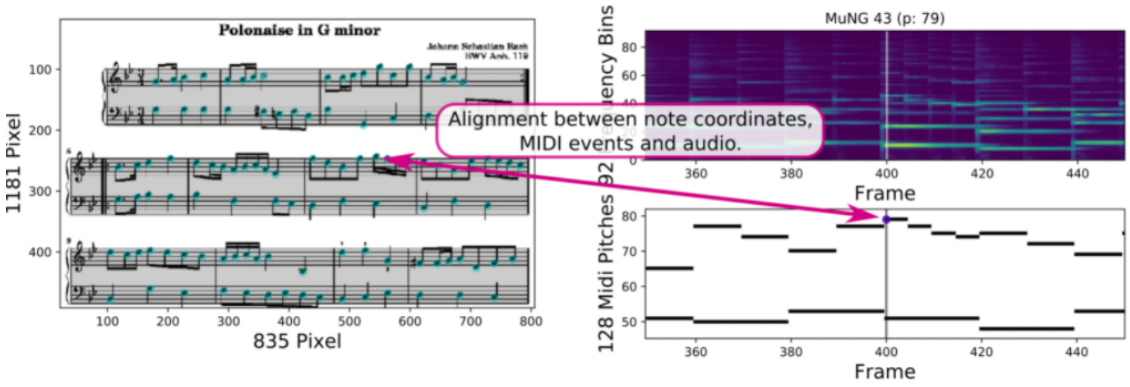


Fig. 5.2: Aligment

sampling rate of 22050hz and 2048 samples are used as the size of FFT window[14]. For subsequent dimensionality reduction operations, researchers use a standardized 16 band logarithmic filter bank to generate 92 frequency slots with allowable frequencies of 30Hz to 16KHz Each score is a subdirectory of the piece's scores/ subdir. For the score generated using lilypond (PDF format), which is stored in the score directory, researchers generated the following information:

1. MuNG (MUSCIMA++ Notation Graph) holds alignment to performances
2. Coordinates of noteheads and systems
3. Page images

Notehead and system coordinates for each songs are stored in the cords/ subdirectory of the score. Notepad coordinates are used as the origin to establish complete alignment for the coordinates of the whole system.

Then researchers established musima + + symbol diagram (mung), which is an XML format to describe music in figure.5.3. Each part records the information grouped into the system by Notepad, which means the consistency between score and performance. Individual noteheads are stored and relevent important alignment between a performance and a score. Next, researchers will introduce how to compare the two kinds of information.

The identifier for the entire dataset is similar to $XML : ID$ That is, for the identifier in the score, mung is stored in each separate file. It can be viewed in the code or work across pages. The relevant border is divided by elements. How to correctly align notepad with Notepad, and the most important thing is how to expand the integral. researchers use elements to store mung object names and group Notepad into the system. Mung format consists of these elements and no other descriptors are required.

1. The name of a performance always begins with the "key" attribute of the element to point performance.

```
<CropObject xml:id="msmd_aug___BachCPE__cpe-bach-rondo__cpe-bach-rondo_ly-P00___0">
 <Id>0</Id>
 <ClassName>notehead-full</ClassName>
 <Top>118</Top>
 <Left>598</Left>
 <Width>9</Width>
 <Height>7</Height>
 <Mask>0:0 1:63</Mask>
 <Outlinks>1824</Outlinks>
 <Data>
                <DataItem key="BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-
1000_ElectricPiano_onset_frame" type="int">255</DataItem>
                <DataItem key="tied" type="int">0</DataItem>
                <DataItem key="BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-
1000_ElectricPiano_note_event_idx" type="int">48</DataItem>
                <DataItem key="BachCPE__cpe-bach-rondo__cpe-bach-rondo_tempo-
1000_ElectricPiano_onset_seconds" type="float">12.727274</DataItem>
                <DataItem key="midi_pitch_code" type="int">68</DataItem>
                <DataItem key="ly_link"
type="str">textedit:///media/matthias/Data/msmd_aug/BachCPE__cpe-bach-rondo__cpe-bach-
rondo/BachCPE__cpe-bach-rondo__cpe-bach-rondo.norm.ly:704:15:16</DataItem>
 </Data>
</CropObject>
```

Fig. 5.3: XML format

2. The key element of music score and audio (spectrum diagram) alignment is the particular notehead corresponding to the specific note event list element in performance PERF_NAME[15].

3. Performance PERF_NAM is for this particular notehead. The name of spectrum and the frames in the MIDI matrix are specified by this element. This is derived from the alignment operation, which simplifies the operation of storing it in mung

4. The notehead is interpreted when the time in audio of the performance file are exacted from element points.(researchers will discard the audio when the relevant operations are completed, but the element can make the alignment of noteheads easier through the re-rendering of performance MIDI)

5. Item holds a reference to the exact location in the normalized lilypond file from which the lilypond sculpting engine renders the Notepad. It helps us restore the pitch associated with this notepad.

6. The MIDI pitch code associated with Notepad will exist in the and element and be extracted from the original lilypond file

5.3 Neural Network Model

For the theoretical implementation of the algorithm, because the extension of the reinforcement learning baseline algorithm is the actor critic algorithm, researchers focus on the synchronous advantage actor critic method. In the previous chapter, researchers introduced the policy Θ (determining the agent's behavior) and the value function v (s), predicting the quality of a specific state s relative to the cumulative future reward. The actor critic method uses these two concepts. The participants are composed of strategy Θ And is responsible for selecting the suitable operation in each state. After the agent completes the corresponding action according to its own judgment, the critic helps the agent improve the performance according to the value function v (s). In the context of depth RL, these two functions are approximately simulated by depth neural network (DNN), which is called strategy and value network. researchers use it below Θ Parameter representing the policy network.

Figure.5.4 shows a schematic diagram of this structure of network. As people known, music score and audio at the same time are operated by multimodal convolutional neural network. The input of the network is the Markov state of MDP introduced in Chapter 2. The left side of the network processes the paper image, The right side processes spectral extracts (including pictures). The two modes are combined in series and further handling using dense layers after low-level representation learning. This architecture means that the parameters of the lower layer is shared by the strategy and value network, which is a common choice in RL. For the last two output layers, we make analysis respectively. The first layer will predict the action according to our strategy, and the guess probability is Θ (a|s), in which there are three output neurons, which can be converted into effective probability distribution through soft maximum activation. The second output layer consists of a linear output neuron that predicts the current state value V (s). Table.5.5 lists the exact architecture used in our experiment. In addition to the two output layers, researchers use exponential linear units.

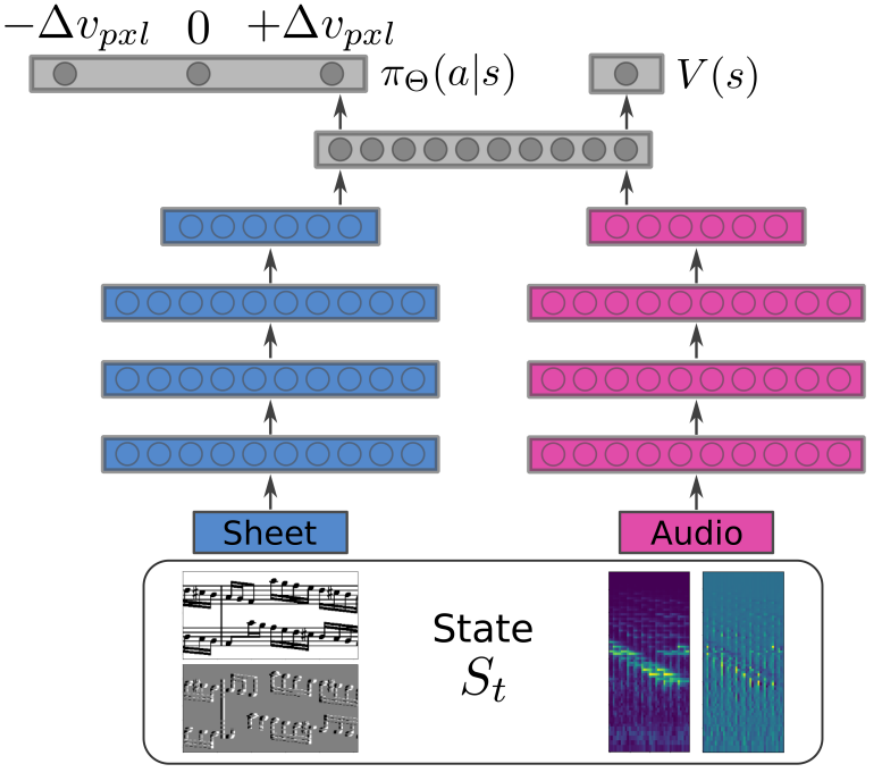


Fig. 5.4: network

Table 1. Network architecture. DO: Dropout, Conv(3, stride-1)-16: 3×3 convolution, 16 feature maps and stride 1.

Audio (Spectrogram) 78 × 40	Sheet-Image 80 × 256
Conv(3, stride-1)-32	Conv(5, stride-(1, 2))-32
Conv(3, stride-1)-32	Conv(3, stride-1)-32
Conv(3, stride-2)-64	Conv(3, stride-2)-64
Conv(3, stride-1)-64 + DO(0.2)	Conv(3, stride-1)-64 + DO(0.2)
Conv(3, stride-2)-64	Conv(3, stride-2)-64
Conv(3, stride-2)-96	Conv(3, stride-2)-64 + DO(0.2)
Conv(3, stride-1)-96	Conv(3, stride-2)-96
Conv(1, stride-1)-96 + DO(0.2)	Conv(1, stride-1)-96 + DO(0.2)
Dense(512)	Dense(512)
Concatenation + Dense(512)	
Dense(256) + DO(0.2)	Dense(512) + DO(0.2)
Dense(3) - Softmax	Dense(1) - Linear

Fig. 5.5: Network architecture

Chapter 6

Implementation

The section describes the implementation of each algorithm and programming technology in detail.

6.1 Score Following visulization

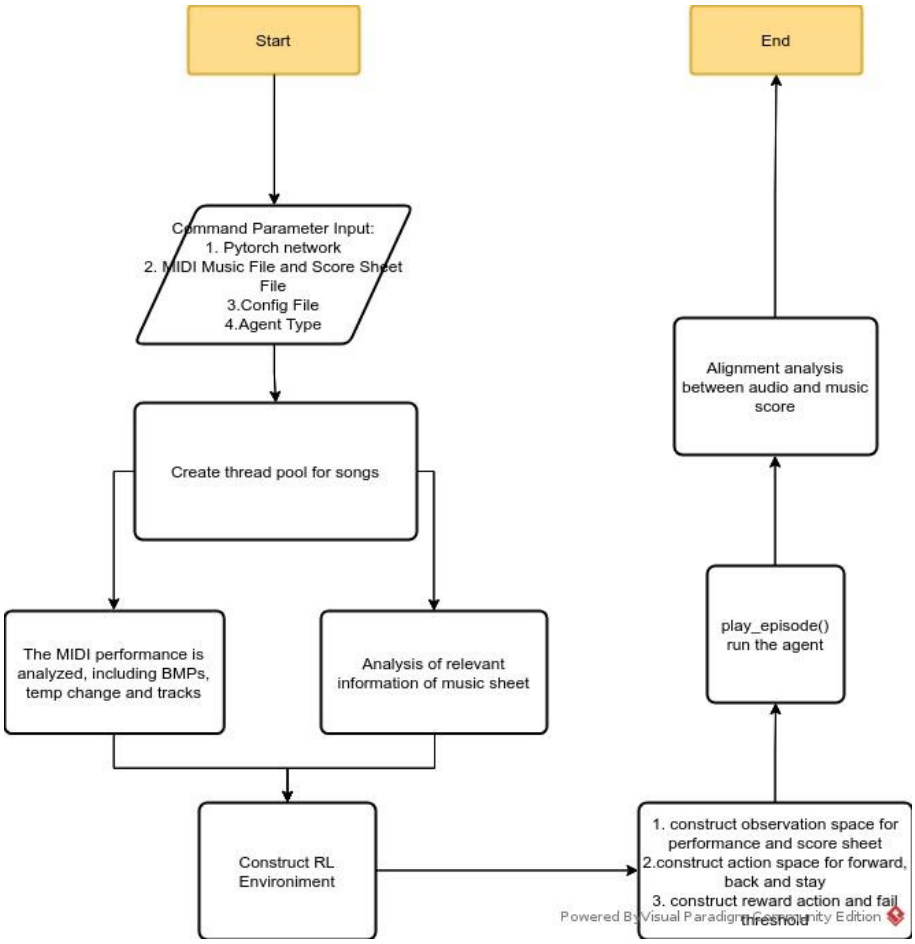


Fig. 6.1: flow chart2

Once you have installed all packages and downloaded the data everything should be

ready to train the models. To check if the game works properly on your system you can run the following script and play the game on your own.

```
python test_agent.py --data_set ./data/test_sample --piece
                                        Anonymous__lesgraces__lesgraces --
                                        game_config game_configs/mutopia_lchs1.
                                        yaml --agent_type human
```

The program block diagram in figure.6.1 reveals that researchers try to align audio and music score manually without using any reinforcement learning algorithm. researchers first create a relevant thread pool to store music, then researchers analyze audio files, such as BMPs and tracks, and then researchers analyze the information of music score. Then researchers created a comparison environment. Different from the reinforcement learning environment, researchers can operate the left and right of the keyboard to control the speed of audio alignment, and then researchers set a certain space beyond the accurate audio range to end the game.

6.2 Train Agents

To train a model on a specific data set, learning algorithm and network architecture you can start with our suggested commands below. First one is about training Nottingham (monophonic) dataset and the second is about MSMD (polyphonic) by actor critic method.

```
python experiment.py --net ScoreFollowingNetNottinghamLS --train_set <PATH-TO-
                                        DATA-ROOT>/nottingham/nottingham_train
                                        --eval_set <PATH-TO-DATA-ROOT>/
                                        nottingham/nottingham_valid --
                                        game_config game_configs/nottingham_ls1
                                        .yaml --log_root <LOG-ROOT> --
                                        param_root <PARAM-ROOT> --agent a2c
```

```
python experiment.py --net ScoreFollowingNetMSMDLCHSDeepDoLight --train_set <
                                        PATH-TO-DATA-ROOT>/msmd_all/
                                        msmd_all_train --eval_set <PATH-TO-DATA
                                        -ROOT>/msmd_all/msmd_all_valid --
                                        game_config game_configs/mutopia_lchs1.
                                        yaml --log_root <LOG-ROOT> --param_root
                                         <PARAM-ROOT> --agent a2c
```

The flow chat can be viewed in the following figure.6.2.The input of the program is the appropriate training time and relevant training data set. Through the relevant game configuration file, the relevant neural network framework will be created, and then researchers will create an evaluation pool. If it is in the evaluation pool, the alignment training experiment is completed and the results will be output; If not, train the music through previous statistical data and output the results. By using Tensorboard, it's easily to watch our training progress.

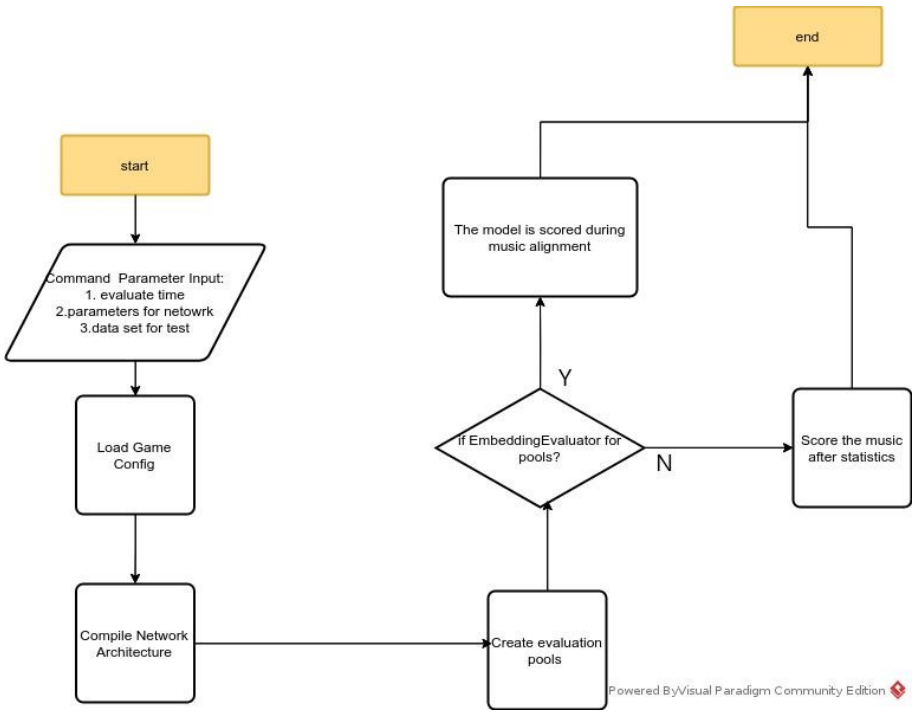


Fig. 6.2: flow chart1

6.3 Evaluate Agents

To investigate the performance of your trained agents you have the following two options:

6.3.1 An Audio-Visual Quality Check

To get an intuition on how well your agents work you can visualize its performance on a particular piece. Just run the following command and you will get a rendering videos to visulization following game processing.

```
python test_agent.py --params <PARAM-ROOT>/<run_id>/best_model.pt --data_set <
                                PATH-TO-DATA-ROOT/nottingham/
                                nottingham_test --piece <PIECE-NAME> --
                                game_config game_configs/nottingham_ls1
                                .yaml --agent_type rl
```

6.3.2 Computing the Numbers

To compute the performance measures over the entire training, validation or test set you can run the following command.

```
python evaluate_agent.py --trials 1 --params <PARAM-ROOT>/<run_id>/best_model.
                                pt --data_set <PATH-TO-DATA-ROOT/
                                nottingham/nottingham_test
```

The program block diagram of this subroutine is shown below figure.6.3. Firstly, the input is based on the relevant neural network model (. PT), the data to be aligned

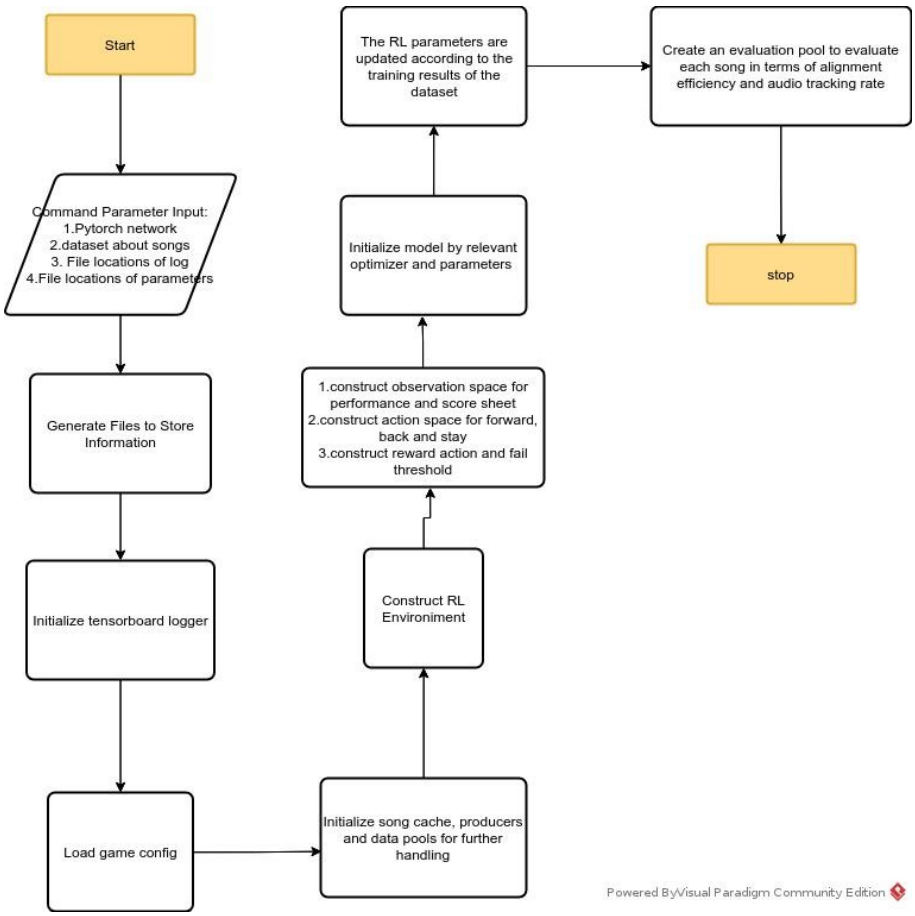


Fig. 6.3: flow chart

(including audio MIDI file and score file) and the parameters of the whole configuration file (. Yaml). After that, researchers will generate the operation data of relevant files. researchers can use the tensorboard to view the relevant data. With the establishment of the program song pool, researchers will create relevant reinforcement learning environment (including relevant observation space, viewing the efficiency of comparison, action space to decide whether to accelerate, decelerate or maintain the current speed for tracking, and the final reward mechanism).

Chapter 7

Experiment

7.1 Experimental Data Set Processing and Experimental Design

Our experiment will use two different data sets. The Nottingham dataset includes 285 monophonic melodies of folk music (training: 167, verification: 43, test: 40); it has been used in to evaluate score tracking in score images. The second dataset contains 479 classical works created by different composers such as Beethoven, Mozart and Bach, which are from the free mutopia project 4 (training: 300, verification: 20, test: 80). It covers polyphonic music, which is a very difficult challenge for score followers. In both cases, the score is typeset with lilypond, and the audio is synthesized from MIDI using the original piano sound font. For audio processing, the researchers set the calculation rate to 20 FPS and set the sampling rate of 22.05khz to calculate the logarithmic spectrum., This automatic rendering pass provides accurate audio score alignment for training. FFT uses the window size of 2048 samples for calculation and post-processing using a logarithmic filter bank, which only allows frequencies from 50Hz to 6KHz (78 frequency units)[16].

The spectrum graph context visible to the agent is set to 40 frames (2 seconds of audio), and the sliding window page image covers 150 frames $\times$ 512 pixels and further tripled before rendering to the network.The researchers used Adam update rules to process the optimizer so that the running average coefficients were 0.5 and 0.999, and the learning rate was set to 12. Then, researchers train the model until there is no improvement in the number of tracking starts of 50 previous times on the verification set, and reduce the learning rate by 10 times. Rhythm change action vpxl in Nottingham is 0.5 and polyphonic music is 1.0.

However, compared with training, in our evaluation, researchers only consider the time steps of the actual attack in the audio. Although inserting an intermediate time step helps to produce a stronger learning signal, it has no musical significance. Specifically, researchers will report the evaluation statistical mean absolute tracking error | DX | and its standard deviation STD for all samples (| DX |). The accuracy of score followers will be quantified by these two measures. In order to measure their robustness, researchers calculated the ratio of the whole tracking set from start to end and the ratio of tracking

segments.

7.2 Experimental Results and Data Analysis

Method	R_{tue}	R_{on}	$\overline{\lvert d_x \rvert}$	$std(\lvert d_x \rvert)$
	Nottingham (monophonic, 46 test pieces)			
REINFORE	0.43	0.65	3.15	13.15
REINFORCE$_{bl}$	0.94	0.96	4.21	4.59
A2C	**0.96**	**0.99**	**2.17**	**3.53**
	Mutopia (polyphonic, 100 test pieces)			
REINFORE	0.61	0.72	62.34	298.14
REINFORCE$_{bl}$	0.20	0.35	48.61	41.99
A2C	**0.74**	**0.75**	**19.25**	**23.23**

Fig. 7.1: Experiment Result

Table.7.1 summarizes the experimental results. Looking at the Nottingham dataset, researchers find that there is a big gap in the performance of different methods. These two RL based methods can almost completely track all specimens. In addition, the average tracking error of A2C is low and shows a significantly lower standard deviation. The high standard deviation of REINFORE is more obvious in polyphonic music. For predicting the position probability distribution on the score image of a given current audio, REINFORE will be defined as a positioning task. Repeated music paragraphs may lead to multiple patterns in the location probability distribution. At the same time, the probability of each pattern is the same. When the REINFORE tracker follows the mode with the highest probability, it begins to jump between such fuzzy structures, resulting in a high standard deviation of tracking error and, in the worst case, losing the target.

Our MDP score tracking formula solves this problem because the agent controls the travel speed of its navigation in the worksheet image. This limits the agent because it does not allow large jumps in the score. In addition, it is closer to the actual performance of the music(for example, top to bottom and left to right when excluding duplicates). This theoretical advantage is reflected by our results, especially A2C

However, in the case of complex polyphonic score, researchers also observed that the performance of BL decreased completely. The figures reported are the results of more than five days of training. researchers have mentioned that the known strategy optimization has a high square difference in gradient estimation, which is what researchers observed in the experiment. A2C algorithm trains the same data set, Nottingham data set, with

the same reinfocus algorithm. Even if reinfocus learns a useful strategy, it takes about 20 times as long as A2C, that is, it takes 5 days to train successfully, while A2C takes 6 hours. For mutopia (120 samples), 70% of the samples will track closely during the tracking process without losing the target. The average error of this result is only 20 pixels, which is about 5mm in the standard A4 paper Western score, which is three times higher than the baseline average error of 62 pixels.

researchers also report the results of enhancing REINFORE with a view to enhancing the potential of RL in this case. Recall that the basic MDP is the same for both REINFORE and A2C. The only change is a stronger learner. All other components, including network architecture, optimization algorithms and environment, remain unchanged. Considering that deep learning is one of the most deeply studied fields in machine learning, as long as the learning itself makes progress, researchers can look forward to the further improvement of score tracking task.

Then, I did some comparison experiments to adjust the parameters, hoping to obtain the best parameters for same dataset. Now researchers comparing the parameter of Computation rate and Spectrograms rate. As the figures shown, the left picture is the REINFORCE algorithm and the right picture is advanced acter-critiers algorithm.

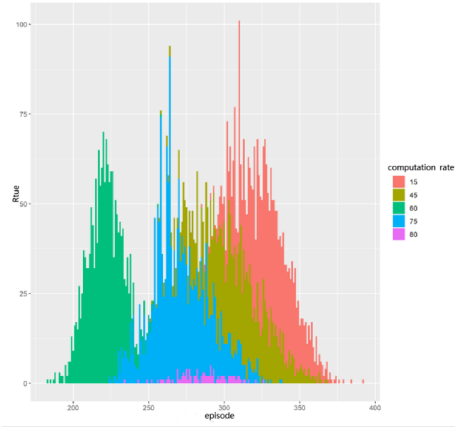


Fig. 7.2: data analyse1

researchers reset the computation rate from 15 to 80 and researchers found the red one has bigger R-tue, which means when the computation rate is 15, the robustness of model is more higher in both algorithms. Looking at these bar-charts, researchers set different parameters to comparing the accuracy of model.

researchers found that the higher computation rate that the model is more accuracy for both algorithms. And for these box diagrams, researchers found different between the two algorithms. As for REINFORCE algorithm, researchers set spectrograms rate as 175, the R-thu is the highest which means the system is most robust. However, for advanced-acter-critiors, the parameter is 145,the system is most robust.

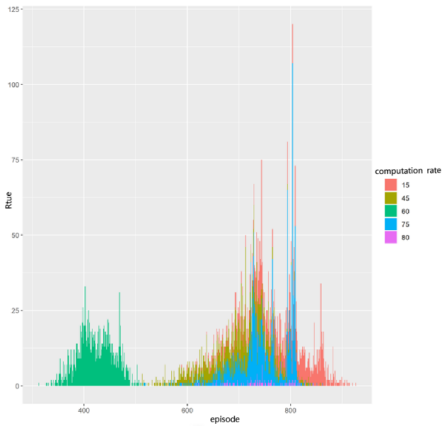
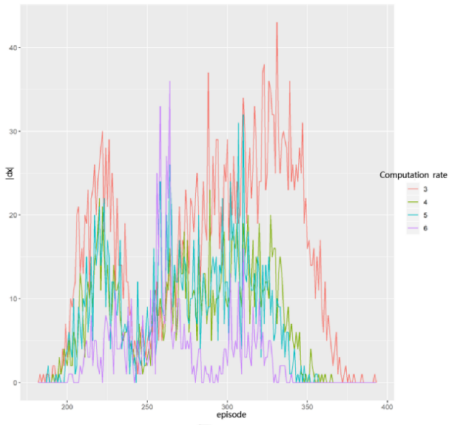
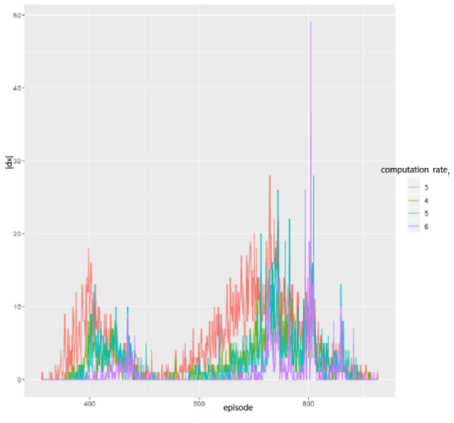


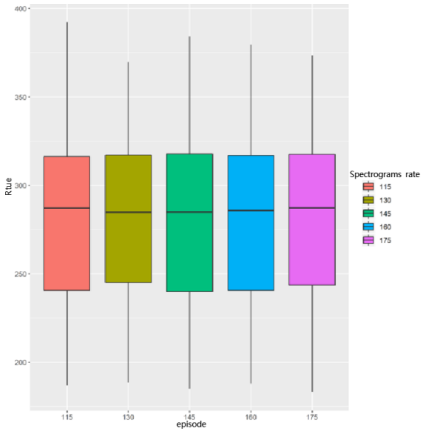
Fig. 7.3: data analyse2



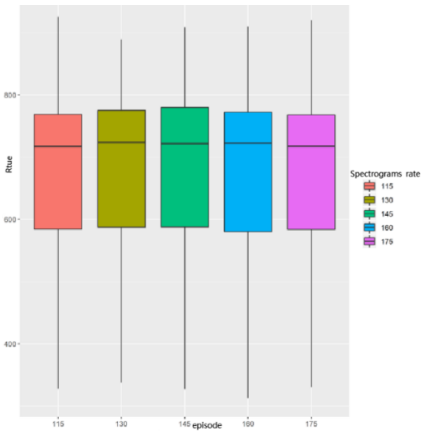
(a) data analyse3



(b) data analyse4



(c) data analyse5



(d) data analyse6

Fig. 7.4: the order of optimal parameters

Chapter 8

Conclusion

In conclusion, researchers propose a Markov decision process for score tracking in score images, and show how to solve this problem through the most advanced deep reinforcement learning. By designing the comparison experiment of single tone and polyphonic piano music, we find that the verification of the algorithm is very competitive. Finally, researchers would like to discuss some specific aspects and point out interesting future prospects.

First, researchers use a continuous reward signal to train all agents, The signal is calculated by inserting the target (real ground) position between the continuous attack set and the note head. Of course, reinforcement learners can also learn from the delayed signal (for example, the non-zero reward is only for the actual top chord, or even the horizontal or bottom chord). This further means that, for example, researchers can use one of our models to train synthetic audio and annotate a set of real performance audio at the bar level (this is completely feasible)[17], and then fine tune the model using the very same algorithm. The only difference is that for the time point without annotation, the environment only returns zero neutral reward.

Secondly, researchers have begun to experiment with continuous control agents, which directly predict the required rhythm changes rather than relying on a set of discrete actions. After that, researchers analyzed the data and obtained the best parameters tested, such as calculation rate and spectrum rate. In the future, these improvements will extend other algorithms, such as policy optimization. It has been proved that continuous control is very successful in other fields and can realize the perfect combination of music score and audio.

Researchers believe that all these make the music score after MDP a very promising and exciting playground, which can be further studied in music information retrieval and reinforcement learning.

Chapter 9

Professional Issues

I confirm that this project obey the Code of Practice and Code of Conduct issued by British Computer Society, and I believe that the whole project is designed to serve the public interest. I guarantee that the whole project is completed by myself. Furthermore, this project is believed to have the ability to benefit other people. In the process of carrying out this project, I have followed the ”Key IT practices” stated in the code of practices.

Reference

[1] A. Baró, P. Riba, J. Calvo-Zaragoza, and A. Fornés, "From optical music recognition to handwritten music recognition: A baseline," Pattern Recognition Letters, vol. 123, pp. 1–8, 2019.

[2] X. Wang, X. Hu, Y. Li, and C. Jiang, "Multi-modal human pose estimation based on probability distribution perception on a depth convolution neural network," Pattern Recognition Letters, vol. 153, pp. 36–43, 2022.

[3] J. Zhang, J. Ren, L. Li, J. Gu, and D. Zhang, "Defect identification of layered adhesive structures based on dynamic time warping and simulation analysis," Infrared Physics and Technology, vol. 120, p. 103943, 2022.

[4] C. Wen, A. Rebelo, J. Zhang, and J. Cardoso, "A new optical music recognition system based on combined neural network," Pattern Recognition Letters, vol. 58, pp. 1–7, 2015.

[5] C. Zhang, S. Li, M. Ye, C. Zhu, and X. Li, "Learning various length dependence by dual recurrent neural networks," Neurocomputing, vol. 466, pp. 1–15, 2021.

[6] V. Leonhardt, F. Claus, and C. Garth, "Pen: Process estimator neural network for root cause analysis using graph convolution," Journal of Manufacturing Systems, 2021.

[7] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza, "Data representations for audio-to-score monophonic music transcription," Expert Systems with Applications, vol. 162, p. 113769, 2020.

[8] M. Hu, W. Liu, K. Xue, L. Liu, H. Liu, and M. Liu, "Comparing calculation methods of state transfer matrix in markov chain models for indoor contaminant transport," Building and Environment, vol. 207, p. 108515, 2022.

[9] M. Biemann, F. Scheller, X. Liu, and L. Huang, "Experimental evaluation of model-free reinforcement learning algorithms for continuous hvac control," Applied Energy, vol. 298, p. 117164, 2021.

[10] A. Kaveh and S. Rezazadeh Ardebili, "An improved plasma generation optimization algorithm for optimal design of reinforced concrete frames under time-history loading," Structures, vol. 34, pp. 758–770, 2021.

[11] A. Kathirgamanathan, E. Mangina, and D. P. Finn, "Development of a soft actor critic deep reinforcement learning approach for harnessing energy flexibility in a large office building," Energy and AI, vol. 5, p. 100101, 2021.

[12] Y. Pan, M. Wang, and Z. Xu, "Tednet: A pytorch toolkit for tensor decomposition networks," Neurocomputing, vol. 469, pp. 234–238, 2022.

[13] C. Prados-Roman, M. Fernández, L. Gómez-Martín, E. Cuevas, M. Gil-Ojeda, N. Marusczak, O. Puentedura, J. E. Sonke, and A. Saiz-Lopez, "Atmospheric formaldehyde at el teide and pic du midi remote high-altitude sites," Atmospheric Environment, vol. 234, p. 117618, 2020.

[14] S. Lucarini, L. Cobian, A. Voitus, and J. Segurado, "Adaptation and validation of fft methods for homogenization of lattice based materials," Computer Methods in Applied Mechanics and Engineering, vol. 388, p. 114223, 2022.

[15] A. Jablonski and K. Dziedziech, "Intelligent spectrogram – a tool for analysis of complex non-stationary signals," Mechanical Systems and Signal Processing, vol. 167, p. 108554, 2022.

[16] M. Sharma, S. Patel, and U. R. Acharya, "Automated detection of abnormal eeg signals using localized wavelet filter banks," Pattern Recognition Letters, vol. 133, pp. 188–194, 2020.

[17] F. Rumsey and T. McCormick, "Chapter 14 - midi and synthetic audio control," in Sound and Recording (Sixth Edition), sixth edition ed., F. Rumsey and T. McCormick, Eds. Boston: Focal Press, 2010, pp. 397–452.