Extended Abstract

Motivation Music creation is an art with infinite possibilities for showcasing one's creativity. However, there are certain rules and preferences that must be obeyed to appeal to one's liking. This makes it an ideal candidate for Reinforcement Learning. This project is interesting, to me at least, because I have been fascinated by music ever since I was a kid and play the piano and guitar myself. I have always wanted to create music but lacked the creative ability to do so, so this time, maybe I can use Reinforcement Learning to help me.

Method We will implement the ideas put forward in the paper Jaques et al. (2016), try to recreate the results and then extend it. We use a pretrained RNN as our baseline model. We then teach the RNN concepts of music theory using Reinforcement Learning, by using Deep Double Q-Learning (Van Hasselt et al. (2016)). The reward is calculated using a combination of the reward from from the baseline RNN model as well as from the music theory rules. The music theory rewards are based on the book Gauldin (1988). We implement several of these and allow various reward functions to be picked during training.

Implementation We implemented the above method in python code files and in jupyter notebooks. We trained the model, generated sample melodies and collected various quantitative and qualitative stats as discussed in the Results section.

Results As a baseline, we simply sampled melodies from the baseline RNN model and then sampled melodies from our trained model. We noticed that the music generated by our model was far superior to the baseline in that it had fewer repeated notes and had a more coherent structure. The probability distribution of the various notes showed that the trained model had a more diverse distribution, which indicates that the melody is more varied. We also collected metrics, for example, how many notes were repeated and saw that it was much fewer in our trained model, which was consistent with the music-theory based reward function we used to train our model.

Discussion The results above show how our model has improved upon the baseline model. Having a music theory based reward function helped the model a lot, as expected. This opens the path for more exploration and tweaking in the exact reward function to use, since each genre of music will have it's weights in the reward function. We originally intended to perform a survey to carry out a more widespread qualitative analysis of our model, however, did not get enough feedback in time for the report. Moreover, unfortunately, we also could not wrap up the RLHF extension in time for the report.

Conclusion The experiments demonstrate that teaching a baseline RNN with music theory based Reinforcement Learning greatly improved the quality of generated music. Further work could experiment with various reward functions in order to improve the model. We can also add a Reinforcement Learning From Human Feedback component to generate music to one's liking. Yet another application could be generating music similar to a known piece of music, by adding the target music in the reward function of our model.

Applications of Reinforcement Learning in Music

Arindam Saha
Center for Global and Online Education (CGOE)
Stanford University
saha2@stanford.edu

Abstract

In this project, we'll be looking into way of generating music using Reinforcement Learning. Music creation is an art with infinite possibilities for showcasing one's creativity. However, there are certain rules and preferences that must be obeyed to appeal to one's liking. This makes it an ideal candidate for Reinforcement Learning. We try to recreate and extend the ideas put forward in the paper Jaques et al. (2016). For anyone who struggles to produce an original piece of amazing music, such as myself, this project can come to the rescue.

1 Introduction

In this project, we'll be looking into ways of generating music using Reinforcement Learning. This project is interesting, to me at least, because I have been fascinated by music ever since I was a kid and play the piano and guitar myself. I have always wanted to create music but lacked the creative ability to do so, so this time, maybe I can use Reinforcement Learning to help me.

We will implement the ideas put forward in the paper Jaques et al. (2016), try to recreate the results and then extend it. The paper proposes using a baseline model trained on melodies using a RNN (Recursive Neural Network) and then further improves it using Reinforcement Learning. The authors claim that this solves various issues with music generation that arise with RNNs, such as the the melody repeating the same note as well as lacking a coherent global structure. Our results show that this is indeed true. By adding a music theory based reward during the Reinforcement Learning training process, we were able to greatly improve the quality of melodies generated. We are also able to tweak the reward function to generate particular types of music that appeal to one's liking.

2 Related Work

Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) networks have been used for generative modeling of music, for example, as shown in Eck and Schmidhuber (2002) and Sturm et al. (2016). In LSTM networks, each recurrent cell controls the storage of information via an input, output and forget gate. This lets them learn long-term dependencies in the data as well as allows them to quickly adapt to new data. A softmax on the final network output, lets us obtain the probabilty of each note. The LSTM is trained using softmax cross-entropy loss and back propagation through time.

The LSTM network is trained on monophonic melodies. In order to generate, it is initialized with a short sequence of notes and then we iteratively generate notes by sampling from the output distribution induced by the model's final softmax layer. The recently sampled note becomes the input for the next step. Although they seem to generate music, they have some major drawbacks in that they repeat a lot of notes as well as produce sequences that very clearly lack a consistent structure.

The Recurrent Neural Networks have no knowledge of music theory, which we know provides various rules that govern music composition. For example, music theory tell us that there are groups of notes that belong to keys, chords form progressions and that songs are made up of music phrases. It also tells us which keys conflict with each other, etc. Hence, we embed a music theory based reward function in our training to learn a model that balances between note probabilities found in the data as well as rules of music theory.

3 Method

We use a pretrained RNN as our baseline model, which can be downloaded from here. We then teach the RNN concepts of music theory using Reinforcement Learning, by using Deep Double Q-Learning (Van Hasselt et al. (2016)).

In our application of Reinforcement Learning, the state s is the composition so far, action a is the next note to be played and we aim to learn a stochastic policy $\pi(a|s)$ that generates melodious music. Q-learning learns the policy π by maximizing reward over a trajectory (i.e. a sequence of notes), with a discount factor γ applied to future rewards. The optimal policy satisfies the Bellman optimality equation

$$Q(s_t, a_t; \pi^*) = r(s_t, a_t) + \gamma \mathbb{E}_{p(s_{t+1}|s_t, a_t)}[\max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \pi^*)]$$

In Deep Q-learning (Mnih et al., 2013), we use a neural network called the deep Q-network (DQN) to approximate the Q function $Q(s, a; \theta)$ The network parameters θ are updated via stochastic gradient descent (SGD) using the loss function defined below

$$L(\theta) = \mathbb{E}_\beta[((r(s, a) + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2]$$

where β is the exploration policy and θ^- is the Target Q-network, which is held fixed during gradient computation. The ϵ-greedy method is used for exploration. Using a Target Q-network to estimate expected future returns, tackles the problem of Q-learning estimating unrealistically high values, which leads to better performance.

A diagrom of the model is show in 1.

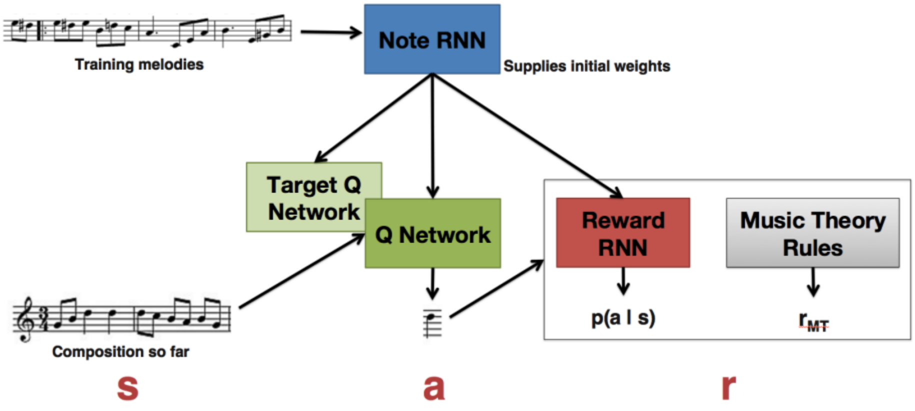


Figure 1: Model (Jacques et al.)

The reward is calculated using a combination of the reward from from the baseline RNN model as well as from the music theory rules, as shown below

$$r(s, a) = \log p(a|s) + \frac{1}{c} r_{MT}(a, s)$$

where r_{MT} is the reward from music theory.

The music theory rewards are based on the book Gauldin (1988). We implement several of these and allow various reward functions to be picked during training. Some of them are:

- **Same Key** - We prefer notes played in the same key
- **Avoid excessive repetition of notes** - We disincentivize the same note being repeated too many times
- **Motifs** - We incentivize motifs i.e. a succession of notes representing a musical idea
- **Repeated motifs** - We incentivize repeated motifs, since it increases likeability of a piece of music.

4 Experimental Setup

We implemented the above method in python code files and in jupyter notebooks. We trained the model, generated sample melodies and collected various quantitative and qualitative stats as discussed in the Results section.

5 Results

First we simply sample melodies from the baseline RNN model.

Figure 2: Baseline model note probabilities

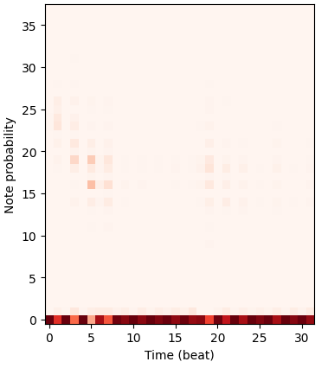


Figure 3: Baseline model sample melody

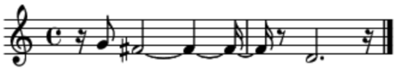


Figure 2 shows the probability distribution of the various notes. Figure 3 shows a sample melody generated. We can see from both these figures the note probabilities graph is quite sparse and the sample melody generated has a lot of repeated notes.

Then we sample melodies from our Reinforcement Learning trained model.

Figure 4: RL Trained model note probabilities

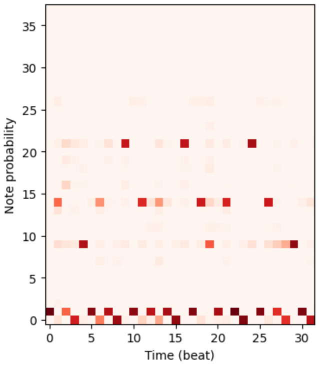


Figure 5: Trained model sample melody

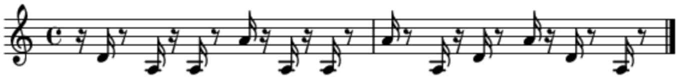
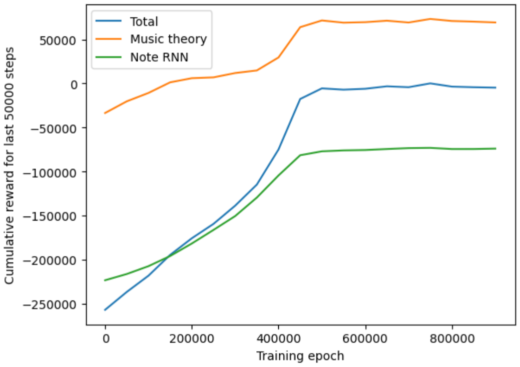


Figure 6: Training plot



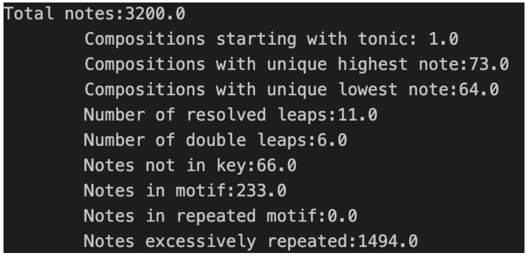
We can clearly see from Figures 4 and 5 that the trained model has a more spread-out probability graph and the sample melody is dense and has a repeating motif in it, aided by our reward function

Figure 6 shows how the reward was split between the RNN and music theory, during training.

5.1 Quantitative Evaluation

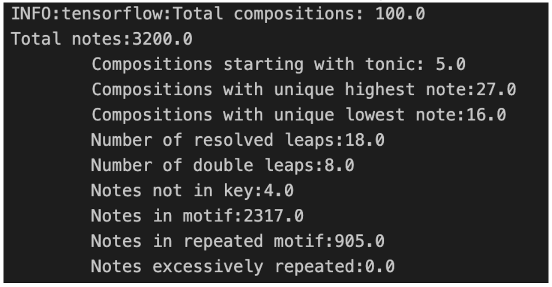
For quantitative analysis, we generated 100 melodies from each model and collected music-theory related stats. Figures 7 and 8 clearly show the stark difference. For example, the baseline model excessively repeated 1494 notes, whereas the trained model never did. The Notes not in key is also greatly less, hence, increasing the quality of music generated.

Figure 7: Baseline evaluation stats



```
Total notes:3200.0
        Compositions starting with tonic: 1.0
        Compositions with unique highest note:73.0
        Compositions with unique lowest note:64.0
        Number of resolved leaps:11.0
        Number of double leaps:6.0
        Notes not in key:66.0
        Notes in motif:233.0
        Notes in repeated motif:0.0
        Notes excessively repeated:1494.0
```

Figure 8: Trained evaluation stats



```
INFO:tensorflow:Total compositions: 100.0
Total notes:3200.0
        Compositions starting with tonic: 5.0
        Compositions with unique highest note:27.0
        Compositions with unique lowest note:16.0
        Number of resolved leaps:18.0
        Number of double leaps:8.0
        Notes not in key:4.0
        Notes in motif:2317.0
        Notes in repeated motif:905.0
        Notes excessively repeated:0.0
```

5.2 Qualitative Analysis

For qualitative analysis, we listened to the music generated by the model and it was indeed much more pleasing to hear. The melody displayed in 5 is one such example and it is easy to see that it has quite a few notes to make the music interesting. We originally intended to perform a survey, however, could not get enough feedback in time for this report.

6 Discussion

The results above show how the model has improved upon the baseline model.

7 Conclusion

The experiments demonstrate that teaching a baseline RNN with music theory based Reinforcement Learning greatly improved the quality of generated music. Further work could experiment with various reward functions in order to improve the model. We can also add a Reinforcement Learning From Human Feedback component to generate music to one's liking. Yet another application could be generating music similar to a know piece of music, by adding the target music in the reward function of our model.

8 Team Contributions

This was a solo project.

Changes from Proposal There was not enough time to include the RLHF (Christiano et al. (2017)) component in the report.

References

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems* 30 (2017).

D. Eck and J. Schmidhuber. 2002. Finding temporal structure in music: blues improvisation with LSTM recurrent networks. In *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*. 747–756. https://doi.org/10.1109/NNSP.2002.1030094

Robert Gauldin. 1988. *A practical approach to eighteenth-century counterpoint*. Prentice-Hall.

Natasha Jaques, Shixiang Gu, Richard E. Turner, and Douglas Eck. 2016. Generating Music by Fine-Tuning Recurrent Neural Networks with Reinforcement Learning. In *Deep Reinforcement Learning Workshop, NIPS*.

Bob L Sturm, Joao Felipe Santos, Oded Ben-Tal, and Iryna Korshunova. 2016. Music transcription modelling and composition using deep learning. *arXiv preprint arXiv:1604.08723* (2016).

Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.