Artificial Intelligence in music composition

Jiménez Castro Bruno Isaac, Federico Michelotto.

Advanced Techniques in Artificial Intelligence, Computer engineering

Universidad del País Vasco

Campus Gipuzkoa

bruno.bj93@gmail.com, federico.mikelotto@gmail.com

***Abstract*- Algorithmic composition has been thought since Mozart's time; more recently contemporary musicians such as Arnold Schoenberg, Anton Webern and John Cage all have composition based on different algorithms.**
Even though music composition is a creative process of humanity, it seems that artificial intelligence has been designed enough to develop the same capability. Could be this a mathematical approach to human creativity?
***Keywords*-. AI, music composition, neural networks.**

1. Introduction

Artificial Intelligence has been rather successful in many aspects, including creating music.
Basically, an AI is a mathematical approach of a human behaviour within an environment, and the interesting idea about applying an AI on musical composition is that music, as one of the beauty arts, depends mostly on human creativity.
It is not so difficult to represent a physical phenomenon mathematically, but how can math or computing represent creativity? And even more difficult, how can creativity be reproduced algorithmically?
For such an ambitious solution have been already designed many different algorithms and structures. Recurrent, Deep Learning, Convolutive and Long Short-term Neural Networks.
Over time, it is harder for amateurish musicians to differentiate between AI and human compositions, and this tends to become as hard as nobody could recognize the difference in the future. Maybe this future is not so far if it has not arrived yet.
Before thinking about the algorithmically composition of a masterpiece of music we must think first in the algorithmic creation of music. As long as the music is an integration of harmony, melody and rhythm, the algorithmic composition must merge all those too in a single audio output.

2. History

In the 18th century was invented the first algorithmic music generator, *Musikalisches Würfelspiel* ("musical dice game"), a game that generates short piano compositions from fragments, with choices made by dice.
In the late 1950s there were the firsts attempts to generate music using Markov chains[1], as a formalization of the primitive dice game.
While Markov chain's generator are able to produce just subsequences of the original data, Recurrent Neural Networks (RNNs), developed in the 1980s, try to generate music without the constraint to choose from a set of subsequences. Recurrent neural networks are Neural Networks (NNs) with loops in them. In this way, information produced in output from a NN can be passed as input to the next NN.

[1] A Markov chain is "a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event"

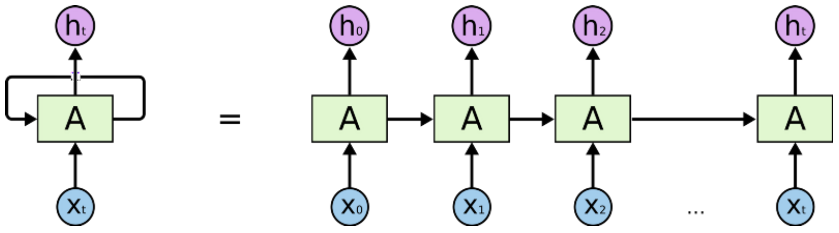


Figure 1: Recurrent neural network unfolded

Doing this, the results are much less predictable, and this feature of the RNN is also the most problematic characteristic of this technology. RNNs are used in many contexts as speech recognition, translation, image captioning and music generation. However, traditional RNNs suffer from short-term coherence; it means that RNNs have difficulties to constrain long-term dependencies.
In 2002 Douglas Eck updated this approach from standard RNN cell, to "Long Short-term memory" (LSTMs) cells. Now Douglas leads the Magenta project at Google Brain where they use machine learning to generate creative content as art, text and music.

3. Architecture

In reinforcement learning (RL) the goal of the agent, is to maximize the reward function over a sequence of actions. The agent takes an action a in according to its policy π based where the agent s is at the moment, that is:

$$s' = \pi(a|s)$$

The optimal deterministic policy π^* is known to satisfy the following Bellman optimality equation:

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

Where the Q function is equal to the summation of immediate reward after performing action a while in state s and the discounted expected future reward after transition to a next state s', in fact the Q function is recursive, and γ is a constant between 0 and 1 that has the scope to give less importance to the future rewardings.
And if we know the optimal Q^* function, the optimal policy π^* is:

$$\pi^*(a|s) = arg\ max_a\ Q^*(s, a)$$

The algorithms that learn this Q^* function are called Q-learning algorithms and they perform this task by minimizing the Bellman residual, i.e the difference between the first member and the second member of the Bellman optimality equation.
This is slightly different from the *TD*[2] *error*. The TD error is the difference between the two sides of the equation without the expected value, evaluated after just a single transition for (s, a). So the Bellman residual is the expected value of the TD error.

[2] Temporal difference (TD) learning is an approach to learning how to predict a quantity that depends on future values of a given signal. The name TD derives from its use of changes, or differences, in predictions over successive time steps to drive the learning process.

The LSTM network, called Note RNN, use Reinforcement learning to train the model. The LSTM network learns to control how to storage past information, and for this scope a LSTM cell has an input gate, an output gate and a forget gate.
In figure 2 we can see that after the Note RNN was trained it supplies the initial weights to the Reward RNN, the Q-Network and the Target-Q-Network. The latter has the scope to stabilize the training of the Q-Network during the phase of calculation of the loss function (the calculation made to minimize the Bellman residual where Q-Network and Target-Q-Network are involved).

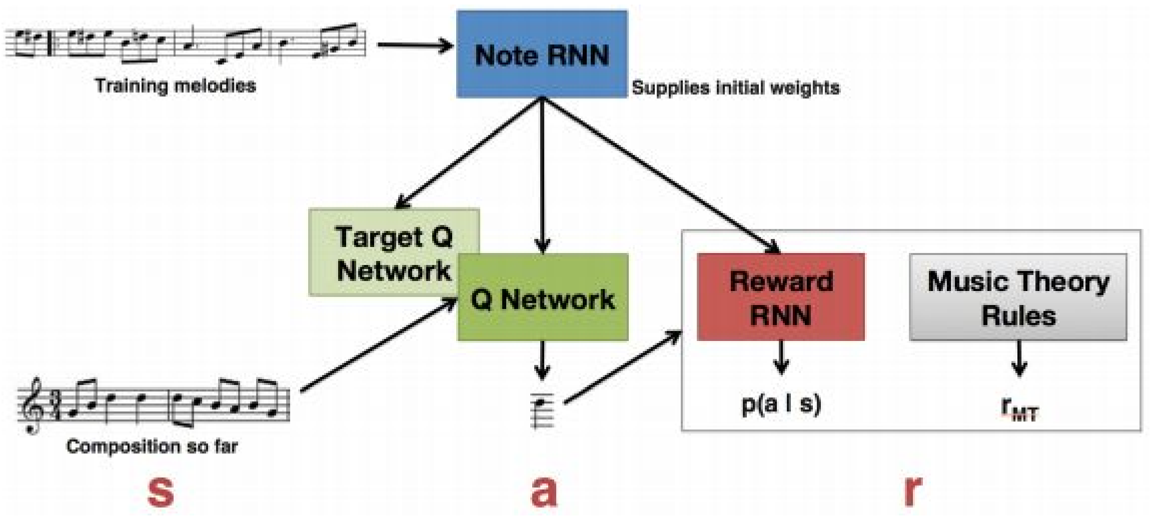


Figure 2: A Note RNN is trained on MIDI files and supplies the initial weights for the Q-network, Target-Q-network, and Reward RNN

During the phase of training, the music theory rules, in addiction with the policy learned until that moment are combined such that the total reward at time t is:

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

where c is a constant controlling the emphasis placed on the music theory reward.

4. Data input

To feed all this information into the model have been used MIDI (Musical Instrument Digital Interface) files. A MIDI file contain encoded information of *musical events,* for example to sound a particular note we need to encode the message (*note on*) and to stop it we have to use an analog message (*note off*); in fact, the activation and the release of a specific note are two different events.
A MIDI channel voice message consists of a status Byte (e.g *note on*) followed by one byte followed by two data bytes, which specify key number (indicating which key was pressed) and velocity (how hard the key was pressed).
To train the Note RNN there have been used a total of 30,000 songs.

5. Music theory

To train the RNNs Google Magenta team followed the directives that can be found in the book written by Robert Gauldin,"A practical approach to eighteenth-century counterpoint":

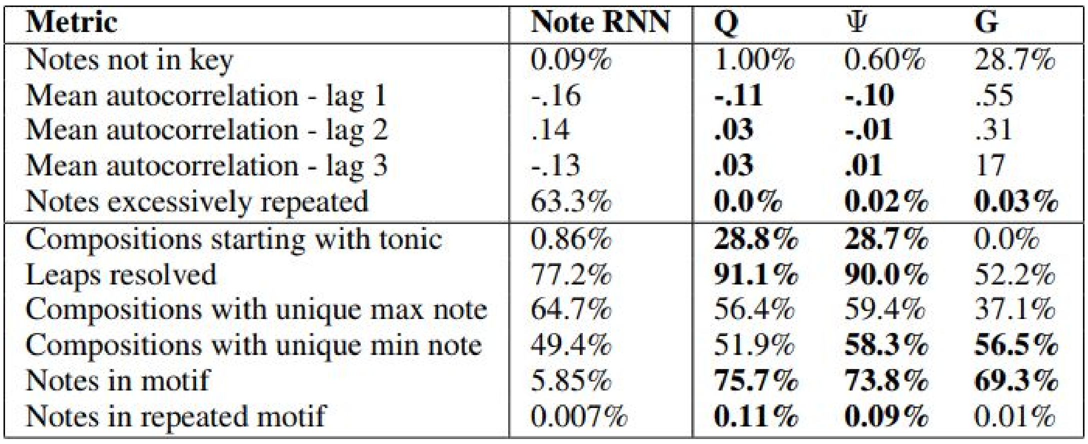
- All notes should belong to the same key;

- The composition should begin and end with the tonic note of the key and this note should occur in the first beat and last 4 beats of the composition;
- Unless a rest is introduced or a note is held, a single tone should not be repeated more than four times in a row;
- To encourage variety, they penalize the model if the composition is highly correlated with itself at a lag of 1, 2, or 3 beats. The penalty is applied when the auto-correlation coefficient is greater than .15.
- The composition should avoid awkward intervals like augmented 7ths, or large jumps of more than an octave.
- Gauldin also indicates good compositions should move by a mixture of small steps and larger harmonic intervals, with emphasis on the former;
- When the composition moves with a large interval (a 5th or more) in one direction, it should eventually be resolved by a leap back or gradual movement in the opposite direction.
- Leaping twice in the same direction is negatively rewarded
- The highest note of the composition should be unique, as should the lowest note
- The model is rewarded for playing motifs, which are defined as a succession of notes representing a short musical "idea"; in our implementation, a bar of music with three or more unique notes. Since repetition has been shown to be key to emotional engagement with music, they also sought to train the model to repeat the same motif within a composition.

Given all these rules as input to the RNN, the Magenta project gives as an output a completely musical composition. Its improvement is determined by the reinforcement learning (rewards and penalties).

6. Results

In this case to measure the performance of the RNN cannot be used likelihood or MSE (Mean Squared Error) because you can't define "good" music or bad "music", but you can compare the results obtained before and after training the LSTM model (called *Note RNN*) with respect to the rules imposed at the beginning.



Metric	**Note RNN**	**Q**	**Ψ**	**G**
Notes not in key	0.09%	1.00%	0.60%	28.7%
Mean autocorrelation - lag 1	-.16	**-.11**	**-.10**	.55
Mean autocorrelation - lag 2	.14	**.03**	**-.01**	.31
Mean autocorrelation - lag 3	-.13	**.03**	**.01**	17
Notes excessively repeated	63.3%	**0.0%**	**0.02%**	**0.03%**
Compositions starting with tonic	0.86%	**28.8%**	**28.7%**	0.0%
Leaps resolved	77.2%	**91.1%**	**90.0%**	52.2%
Compositions with unique max note	64.7%	56.4%	59.4%	37.1%
Compositions with unique min note	49.4%	51.9%	**58.3%**	**56.5%**
Notes in motif	5.85%	**75.7%**	**73.8%**	**69.3%**
Notes in repeated motif	0.007%	**0.11%**	**0.09%**	0.01%

Discussion

After all the examples of algorithmic musical compositions, another valid question could be how can we discern between human and AI compositions if a specialized musician cannot do it?

Iris Yuping in her work about David Cope's Emmy System[2], demonstrates how an entropy analysis could evidence if a composition belongs to a human or it has been generated by a computer.
Her work consists in compare both the real Bach's chorales with the Bach style chorales composed by the Emmy System applying a probabilistic analysis.

She found that the entropy of the AI compositions tends to be considerably higher than the Bach's compositions, thus a classifier could be designed, said Iris.
It is evident that Artificial Intelligence is a potential solution for algorithmic music composition, even knowing music evolves a purely human creative process. A mathematical approach of creativity could have as many different possible results as human composers in the world.
Now it is not difficult to think about a personal music composer, which objectives are to fill a personal need or preference every time you wish it; all contained in your smartphone.

References:

[1] Jaques, N. (2018). *Generating Music by Fine-Tuning Recurrent Neural Networks with Reinforcement Learning*. [online] Storage.googleapis.com. Available at: https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45871.pdf [Accessed 8 Oct. 2018].

[2] Medium. (2018). *RL — DQN Deep Q-network – Jonathan Hui – Medium*. [online] Available at: https://medium.com/@jonathan_hui/rl-dqn-deep-q-network-e207751f7ae4 [Accessed 8 Oct. 2018].

[3] Medium. (2018). *Deep Reinforcement Learning Demysitifed (Episode 2) — Policy Iteration, Value Iteration and….* [online] Available at: https://medium.com/@m.alzantot/deep-reinforcement-learning-demysitifed-episode-2-policy-iteration-value-iteration-and-q-978f9e89ddaa [Accessed 8 Oct. 2018].

[4] leemon@cs.cmu.edu, L. (2018). *What Are Residual Algorithms?*. [online] Cs.cmu.edu. Available at: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/learn-43/lib/idauction2/.g/web/glossary/residual.html [Accessed 8 Oct. 2018].

[5] O'Reilly | Safari. (2018). *Statistics for Machine Learning*. [online] Available at: https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/9651d5a3-22fc-494d-ba92-5a39de916ee0.xhtml [Accessed 8 Oct. 2018].

[6] En.wikipedia.org. (2018). *MIDI*. [online] Available at: https://en.wikipedia.org/wiki/MIDI [Accessed 8 Oct. 2018].

[7] Cs224d.stanford.edu. (2018). [online] Available at: https://cs224d.stanford.edu/reports/allenh.pdf [Accessed 8 Oct. 2018].

[7] Neuro.cs.ut.ee. (2018). [online] Available at: https://neuro.cs.ut.ee/wp-content/uploads/2018/02/MIDI_music.pdf [Accessed 8 Oct. 2018].

[8] Indiana.edu. (2018). *Chapter Three: How MIDI works 4*. [online] Available at: http://www.indiana.edu/~emusic/etext/MIDI/chapter3_MIDI4.shtml [Accessed 8 Oct. 2018].

[9] Douglas, E. (2018). *Project Magenta: Music and Art with Machine Learning (Google I/O '17)*. [online] YouTube. Available at: https://www.youtube.com/watch?v=2FAjQ6R_bf0 [Accessed 8 Oct. 2018].

[10] Anon, (2018). [online] Available at: https://www.library.yale.edu/musiclib/exhibits/beyondpaper/gilbert_musical_game.html [Accessed 8 Oct. 2018].