Research Article

Music generation with variational recurrent autoencoder supported by history

Ivan P. Yamshchikov[1] · Alexey Tikhonov[2]

Received: 3 February 2020 / Accepted: 14 October 2020 / Published online: 2 November 2020

Abstract
A new artificial neural network architecture that helps generating longer melodic patterns is introduced alongside with methods for post-generation filtering. The proposed approach, called variational autoencoder supported by history, is based on a recurrent highway gated network combined with a variational autoencoder. The combination of this architecture with filtering heuristics allows the generation of pseudo-live, acoustically pleasing, melodically diverse music.

Keywords Music generation · Discrete sequences generation · Artificial intelligence

Mathematics Subject Classification 68T50 · 68T99

1 Introduction

The rapid progress of artificial neural networks is gradually erasing the border between the arts and the sciences. A significant number of results demonstrate how areas previously regarded as entirely human due to their creative or intuitive nature are now being opened up for algorithmic approaches [24]. Music is one of these areas. Indeed, there were a number of attempts to automate the process of music composition long before the era of artificial neural networks. Well-developed theory of music inspired a number of heuristic approaches to automated music composition. The earliest idea that we know of dates as far back as the nineteenth century, see [15]. In the middle of the twentieth century, a Markov-chain approach for music composition was developed in [8]. Despite these advances, Lin and Tegmark [14] have demonstrated that music, as well as some other types of human-generated discrete time series, tends to have long-distance dependencies that cannot be captured by models based on Markov chains. Recurrent neural networks (RNNs), on the other hand, are better able to process data series with longer internal dependencies [21], such as sequences of notes in a tune [1]. Indeed, a variety of different recurrent neural networks such as hierarchical RNN, gated RNN, long short-term memory (LSTM) network, and recurrent highway network were successfully used for music generation in [4–6, 10, 20, 28] or [23]. Yang et al. [27] use generative adversarial networks for the same task. For a broad overview of generative models for music, we address the reader to [3].

The similarity between the problem setup for note-by-note music generation and the setup used in the word-by-word generation of text makes it reasonable to review some of the methods that proved themselves useful in generative natural language processing tasks. We would like to focus on a variational autoencoder (VAE) proposed in [2, 18]. A VAE makes assumptions concerning the distribution of latent variables and applies a variational approach for latent representation learning. This yields an additional loss component and a specific training algorithm called Stochastic Gradient Variational Bayes (SGVB), see [16] as well as [11]. Thus, a generative

✉ Ivan P. Yamshchikov, ivan@yamshchikov.info; Alexey Tikhonov, altsoph@gmail.com | [1]Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany. [2]Yandex, Berlin, Germany.

VAE obtains examples similar to the ones drawn from the input data distribution. It also gives significant control over the parameters of the generated output, see [13, 26]. This theoretically opens the door for controlled music output and makes the idea of applying VAE-based method to music generation very inviting.

The advantages mentioned above are quite promising, but artificial neural networks also have a well-known problem when applied to music or language generation. A significant percentage of generated sequences, despite their statistical similarity to the training data, are regularly flagged as wrong, boring or inconsistent when reviewed by human peers. This hinders the broader adoption of neural networks in these areas. The contribution of this paper is twofold: (1) we suggest a new architecture for the algorithmic composition of monotonic music called Variational Recurrent Autoencoder Supported by History (VRASH) and (2) we demonstrate that, when paired with simple filtering heuristics, VRASH can generate pseudo-live, acoustically pleasing, melodically diverse melodies.

2 Music representation and data

Four gigabytes of midi files that included songs of different epochs and genres formed a proprietary dataset that was used for the experiments. The data was already available but required significant preprocessing. A single midi file can contain several tracks with meaningful information and some tracks of little importance. The files were therefore split into separate tracks. A certain normalization of the data is often needed to facilitate learning, and so the following normalization procedures were applied to every track individually. Each note in midi file is standardly defined with several parameters such as pitch, length and strength plus the parameters of the track (e.g. the instrument that is playing the note) and the parameters of the file (such as tempo). Although nuancing plays an important role in musical compositions, the strength of the notes was omitted in our experiments. This particular paper focuses on the melodic patterns determined by the pitches and by the temporal parameters of the notes and pauses in between. The median pitch of every track was transposed to the 4th octave. The pauses throughout the dataset were also normalized as follows. For each track, a median pause was calculated. It was expected that the absolute majority of the pauses in the track are equal to the median pause multiplied with a rational coefficient (1/2 and 3/2 being especially popular for the majority of the tracks). Tracks with more than eleven different values for the pauses were filtered out. Generally, temporal normalization of midi files can be rather challenging, but the pause filtering trick described above allows us to normalize the obtained tracks using the value of the median pause. Finally, to prevent the model from possible over-fitting and to make the input diverse enough, tracks with exceedingly small entropy were also excluded from the training data. Since tracks are generated on a note-by-note basis, a disproportionate number of tracks with low pitch entropy (say, a house bass line with the same note repeating itself throughout the whole track) would drastically decrease the quality of the output. The final dataset consisted of 15+ thousand normalized tracks and was used for further training.

A concatenated note embedding was constructed for every note in every track. This embedding included the pitch of the note, its octave and a delay that corresponded to the length of the note. Meta-information of a given MIDI track was also embedded for each individual track.

3 Architecture

We have trained three different architectures for the task of melody generation. The baseline for such tasks is usually a classic language model (LM), as shown in Fig. 1. A classic language model tries to predict the next token in a given sequence using information on previous tokens.

A variational autoencoder was originally proposed for the tasks of text generation in [2, 18]. Figure 2 demonstrates this architecture in application to music generation.

A standard language model uses some form of a *state* that represents information on the previous tokens in a sequence. However, the effectiveness of such representations is hard to assess. This is why contrast with the classical Variational Autoencoder, the Variational Recurrent Autoencoder Supported by History shown in Fig. 3 uses previous outputs as additional inputs to build the prediction on. In this way, VRASH 'listens' to the notes that it has already composed and uses them as additional 'historic' input.

In the VRASH scheme, the support by history partially addresses the issue of slow mutual information decline that seems to be typical for natural discrete sequences such as natural language, notes in a composition or even for genes in a human genome, as shown in [14]. Let us now look at this issue a little closer. The following definitions of mutual information I between two random variables X and Y are equivalent

$$\begin{aligned} I(X,Y) \equiv & S(X) + S(Y) - S(X,Y) \\ &= D(p(XY) \| p(X)p(Y)) \\ &= \left\langle \log_2 \frac{P(x,y)}{P(x)P(y)} \right\rangle \\ &= \sum_{x,y} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}, \end{aligned} \tag{1}$$

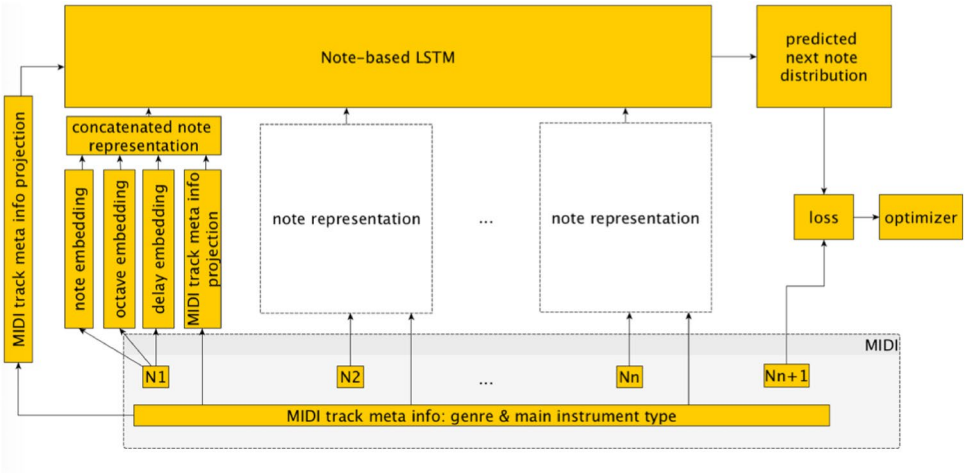

Fig. 1 Language model scheme for music generation

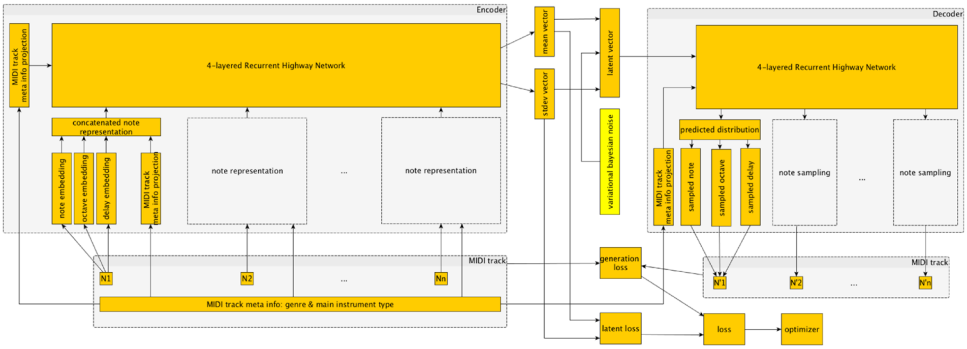

Fig. 2 Variational autoencoder scheme for music generation. Bottleneck between decoder and encoder aims to compress the macrostructure of the melody effectively and obtain a diverse melody with a human-like macrostructure. The variational Bayesian noise highlighted with light yellow color

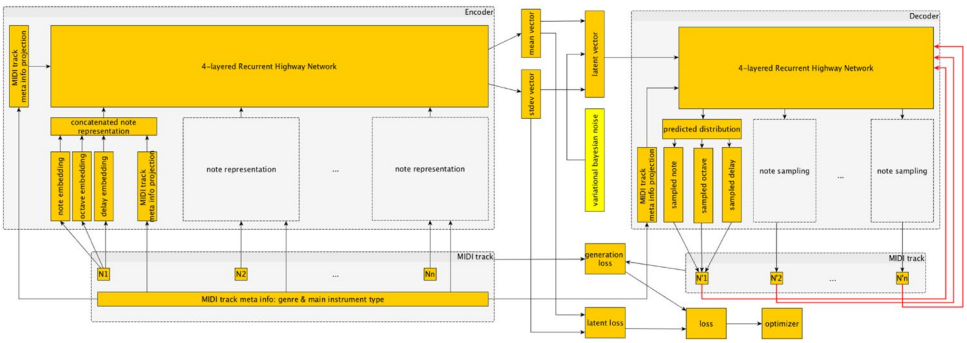

Fig. 3 Variational recurrent autoencoder supported by history (VRASH) scheme for music generation. Previously generated notes are used for the generation of further notes

where $S = \langle -\log_2 P \rangle$ is a Shannon entropy measured in bits, see [19], and D is the Kullback–Leibler divergence, see [12]. Indeed, Lin and Tegmark [14] show that in a number of natural datasets, mutual information between such tokens declines relatively slowly. VRASH addresses this problem specifically, trying to compensate for slow mutual information decline with the *history* support mechanism. Contrary to the approach proposed in [17], where a network generates short loops and then connects them in longer patterns, thus providing a possibility to control melodic variation, we focus on whole-track melody generation. Let us now describe the experimental results obtained.

4 Experiments and discussion

Before discussing the proposed architectures, we feel it is necessary to make the following remarks. It is still not clear how one could compare the results of generative algorithms that work in the area of the fine arts. Indeed, since music, literature, cinema, etc., are intrinsically subjective, it is rather difficult to approach them with truly rigorous metrics. The majority of approaches is usually based on peer-review systems where the number of human peers can vary significantly. For example, in [9] the authors refer to the subjective opinions of only 26 peers, whereas in [7]

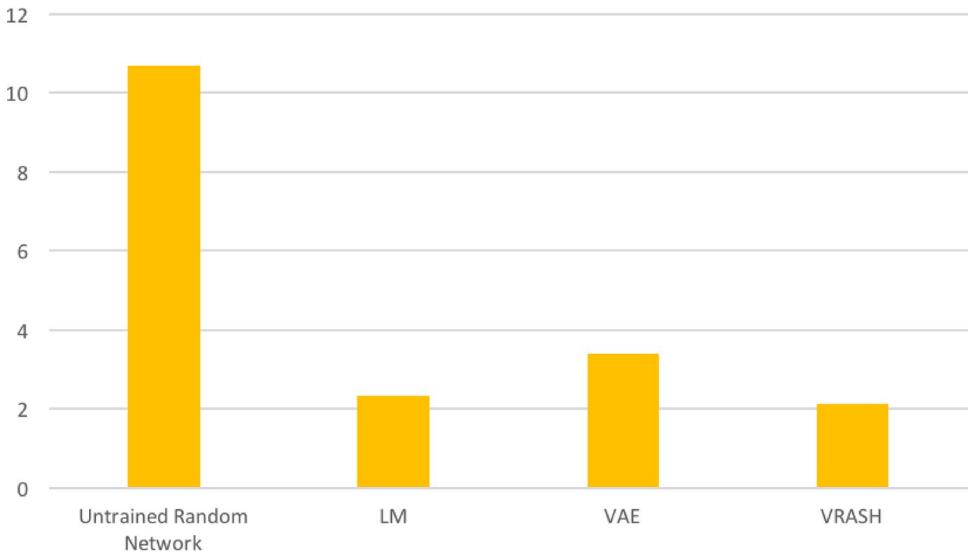

Fig. 4 Cross-entropy of the proposed architectures near the saturation point. The untrained random network is used as a reference baseline

more than 1200 peer responses are analyzed. Such collaborative approaches based on individual subjective assessments could be used to evaluate the quality of the output, but they are typically costly and can hardly be used to obtain scalable results. The number of peers required to compare several different architectures and obtain rigorous quantitative differences between them drastically exceeds the ambition of this particular work. Keeping these remarks in mind, we would like to further discuss possible objective metrics that are frequently used to compare generative models; we would also like to suggest a simple yet useful workaround for quality assessment.

Figure 4 shows the cross entropy of the language model (LM), VAE and VRASH architectures near the saturation point. The untrained random network is used as a reference baseline. The LM and VRASH models demonstrate comparable cross entropy.

Formally speaking, VRASH demonstrates only marginally better performance in comparison with the language model, but we claim that the results produced by VRASH are more subjectively interesting. Further development of this architecture in context of music generation looks promising. After a subjective assessment of the tracks produced by different algorithms, we find that VRASH yields the highest percentage of tracks with qualitative interesting temporal and melodic structures. In [24] the artistic applications of the VRASH architecture are highlighted along with positive feedback from listeners as well as from professional musicians.

All three of the proposed architectures work relatively well and generate music that is diverse and sufficiently interesting as long as the training dataset is large enough and of high quality. Still the architectures do have certain important differences. The first general problem that occurs in many generative models is the tendency to repeat a certain note. This issue is more prominent for the language Model, whereas VAE and specifically VRASH tend to deal with this challenge more successfully.

Another issue is the macrostructure of the track. Throughout the history of music, a number of standard music structures have been developed, starting with a relatively simple song structure (characterized by a repetitive chorus that is divided with verses) and finishing with symphonies that comprise a number of different, less sophisticated forms. Despite the fact that VAE (and VRASH, specifically) have been developed to capture the macrostructures of the track, they do not always provide the distinct structural dynamics which characterizes a number of human-written musical tracks. However, VRASH seems to be the step in the right direction.

To date, every generative model based on artificial neural networks has had problem of low-quality output. Currently, among the melodically diverse and acoustically pleasing tracks which could be generated, we also inevitably hear tracks with annoyingly simple recurrent patterns, off-beat sequences, obscure macrostructures, etc. Faced with this problem, we proposed the following workaround. Alongside the generative VRASH-based model, we used a set of automated filtering heuristics that allowed to obtain a pseudo-real-time non-stop stream of generated music with very limited computational power, for example, we have managed to run a pseudo-real-time generation of non-repeated tunes on Raspberry Pi (Fig. 5).[1]

The heuristics were obtained in a straightforward manner yet turned out to be extremely effective. Using human assessment for 1000+ tracks, we trained a classifier to predict whether or not a track would be acoustically pleasant. Human peers were asked to evaluate tracks on a scale

[1] To hear pseudo-live generation by the VRASH-based model working on Raspberry Pi go to https://youtu.be/Yu8iXOyG8kE.

Fig. 5 VRASH accompanied with heuristic filters is compact enough to run pseudo-real-time music on Raspberry Pi

from 1 to 5, where 5 was the highest mark. Then we split the evaluated tracks into two categories: those that had a mark of 4 or 5 were considered acceptable, whereas the tracks were marked with 3, 2 or 1 were to be detected and removed by the filtering algorithm. For each track in the training dataset, we calculated the following set of theoretic informational features:

- entropy of notes without octave information;
- entropy of changes between consecutive notes without octave information;
- entropy of notes lengths;
- entropy of changes between consecutive notes lengths;
- entropy of notes with octave information;
- entropy of changes between consecutive notes with octave information;
- minimal entropies for sliding windows that were 8, 16, 32, 64 and 128 notes long;
- average entropy for sliding windows that were 8, 16, 32, 64 and 128 notes long;
- coordinates of the sampling vector.

Due to the size of the dataset, we were limited in our choices of methods. Table 1 shows how different methods perform depending on the size of the test dataset.

If the filtering needs to be done faster, the obtained classifier can be replaced with a set of manually constructed empirical heuristics. Due to the fact that we are not interested in the recall of the obtained classifier (when working with neural generative models one often faces an excessive amount of generated melodies, yet wants to filter more pleasing ones), one can make such heuristics even more strict so that 100 % accuracy is achieved. A similar approach was used in [25] for text generation and in [22] for drum pattern sampling and proved itself useful. We believe that such filtering could be adopted across various generative tasks and can significantly improve the resulting quality at a relatively low development cost.

Another way for comparing generated music with real tracks is to build mutual information plots analogous to the ones shown in [14]. We have written above that VRASH is designed to capture long-distance dependencies between the notes in a track. Figure 6 shows how mutual information in terms of Equation 1 declines with distance in different types of VRASH-generated tracks.

Looking closely at Fig. 6, several interesting details are worth mentioning. First of all, Bach-stylized VRASH-generated music tends to have higher mutual information between notes that are further apart. Similar to real tracks, mutual information declines slowly (if at all) in Bach-stylized VRASH-generated music. Its higher values might explain the feedback which we often received from human peers: they noticed that the music was harmonious yet somehow "mechanical". Higher levels of mutual information between distant notes can partially account for that. Second, jazz-stylized VRASH-generated music demonstrates a mutual information profile that is closest to the profile of real tracks. However, as the distance between the notes gets longer, mutual information in generated tracks tends to decrease faster than in real data. This also corresponds with the qualitative feedback of human peers who generally characterized jazz-stylized music as diverse and more human-like.

Table 1 Accuracy of the filtering mechanism varies across different test sets and methods but allows up to 87% of tracks classified as good are also positively evaluated by human

Method	Share of test	# of bad tracks	# of good tracks	Good tracks recall on test
Logistic regression	0.5	207	44	0.79
	0.4	189	35	0.81
	0.3	162	24	0.85
	0.2	128	18	0.86
SVC with linear kernel	0.5	238	64	0.74
	0.4	199	43	0.78
	0.3	138	22	0.84
	0.2	60	9	0.85
SVC with rbf kernel	0.5	229	53	0.77
	0.4	205	39	0.81
	0.3	162	30	0.81
	0.2	86	11	0.87

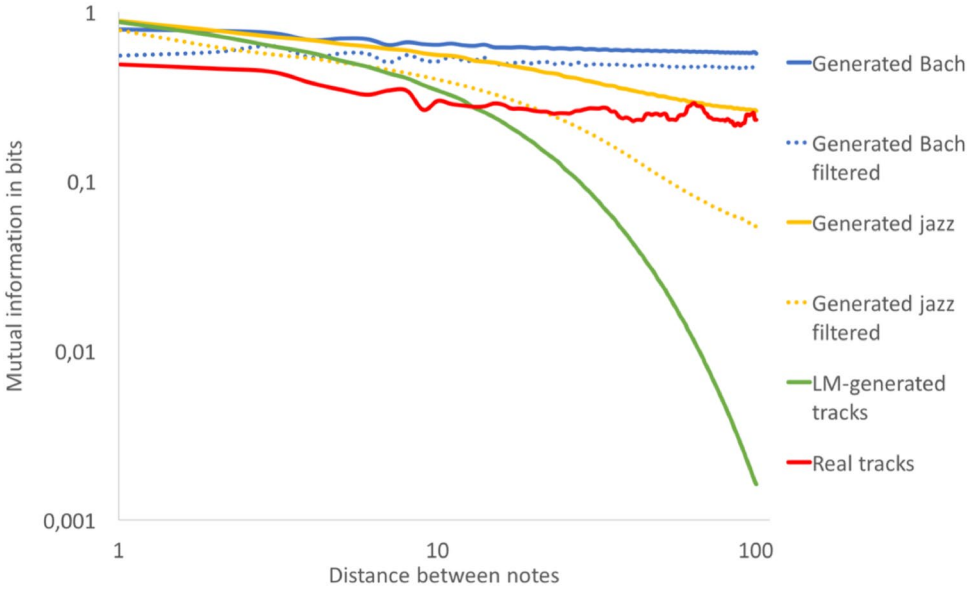

Fig. 6 Mutual information defined in Eq. 1 as a function of distance between two notes in real musical tracks. The figure shows VRASH-generated and automatically filtered Bach-stylized tracks, VRASH generated jazz-stylized tracks, VRASH generated automatically filtered jazz-stylized tracks, and tracks generated by a language model shown in Fig. 1

Filtering jazz-stylized music significantly affects the decline of mutual information between the notes. This could be ascribed to the fact that the filter was trained on Bach-stylizations. A filter that manages to provide a high-quality melody stream for a certain style of music needs to be retrained for other different styles of music in order to guarantee that it will preserve the complexity needed for the music to stay entertaining. Finally, Fig. 6 shows that VRASH-generated melodies tend to demonstrate a slower decline of mutual information than music generated by a language model.

5 Conclusion

In this paper, we described several architectures for monotonic music generation. We compared the Language Model, the Variational Autoencoder and the Variational Recurrent Autoencoder Supported by History (VRASH). This is the first application of VRASH to music generation that we know of. There are several compelling advantages of this model that make it especially useful in context of automated music generation. First of all, VRASH provides a good balance between the global and the local structure of the track. VAE allows to partially reproduce

macrostructure, but VRASH is able to generate more locally diverse and interesting patterns. Second, VRASH is relatively easy to implement and train. Finally, VRASH allows to control the style of the output (through the latent representation of the input vector) and to generate tracks corresponding to the given parameters. Beyond this, we proposed a simple filtering method to deal with the problem of inconsistent generative output. We also proposed an information theoretic approach to compare different generative architectures output with empirical data.

Funding Open Access funding enabled and organized by Projekt DEAL.

Compliance with ethical standards

Conflict of interest The authors declare that they have no competing interests.

References

1. Boulanger-Lewandowski N, Bengio Y, Vincent P (2012) Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. In: In Proceedings of the 29th international conference on international conference on machine learning. Citeseer
2. Bowman S, Vilnis L, Vinyals O, Dai A, Jozefowicz R, Bengio S (2016) Generating sentences from a continuous space. In: Proceedings of the 20th SIGNLL conference on computational natural language learning, pp 10–21
3. Briot JP, Hadjeres G, Pachet FD (2017) Deep learning techniques for music generation–a survey. arXiv preprint arXiv:1709.01620
4. Choi K, Fazekas G, Sandler M (2016) Text-based LSTM networks for automatic music composition. arXiv preprint arXiv:1604.05358
5. Chu H, Urtasun R, Fidler S (2016) Song from pi: a musically plausible network for pop music generation. arXiv preprint arXiv:1611.03477
6. Colombo F, Muscinelli SP, Seeholzer A, Brea J, Gerstner W (2016) Algorithmic composition of melodies with deep recurrent neural networks. arXiv preprint arXiv:1606.07251
7. Hadjeres G, Pachet F, Nielsen F (2017) DeepBach: a steerable model for Bach chorales generation. In: Proceedings of the 34th international conference on machine learning, proceedings of machine learning research, vol 70. PMLR, pp 1362–1371
8. Hiller LA, Isaacson LM (1979) Experimental music; composition with an electronic computer. Greenwood Publishing Group Inc, Westport
9. Huang A, Wu R (2016) Deep learning for music. arXiv preprint arXiv:1606.04930
10. Johnson DD (2017) Generating polyphonic music using tied parallel networks. In: International conference on evolutionary and biologically inspired music and art. Springer, pp 128–143
11. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114
12. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86
13. Larsen ABL, Sønderby SK, Larochelle H, Winther O (2016) Autoencoding beyond pixels using a learned similarity metric. In: International conference on machine learning, pp 1558–1566
14. Lin HW, Tegmark M (2016) Critical behavior from deep dynamics: a hidden dimension in natural language. arXiv preprint arXiv:1606.06737
15. Menabrea LF, Lovelace AA (1843) Sketch of the analytical engine invented by charles babbage, esq., by lf menabrea, of turin, officer of the military engineers. Translated and with notes by AA L. Taylor's Scientific Memoirs 3:666–731
16. Rezende DJ, Mohamed S, Wierstra D (2014) Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31st international conference on international conference on machine learning, vol 32, pp II–1278
17. Roberts A, Engel J, Eck D (2017) Hierarchical variational autoencoders for music. In: NIPS workshop on machine learning for creativity and design, vol 3
18. Semeniuta S, Severyn A, Barth E (2017) A hybrid convolutional variational autoencoder for text generation. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 627–637
19. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27(3):379–423
20. Sigtia S, Benetos E, Dixon S (2016) An end-to-end neural network for polyphonic piano music transcription. IEEE ACM Trans Audio Speech Lang Process 24(5):927–939
21. Sundermeyer M, Schlüter R, Ney H (2012) LSTM neural networks for language modeling. In: Thirteenth annual conference of the international speech communication association
22. Tikhonov A, Yamshchikov IP (2020) Drum beats and where to find them: sampling drum patterns from a latent space. arXiv preprint arxiv.org/abs/2007.06284
23. Waite E, Eck D, Roberts A, Abolafia D (2016) Project magenta: generating long-term structure in songs and stories. https://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn. Accessed 15 July 2016
24. Yamshchikov IP, Tikhonov A (2018) I feel you: what makes algorithmic experience personal? Polit Mach Art After 7:1–6
25. Yamshchikov IP, Tikhonov A (2019) Learning literary style end-to-end with artificial neural networks. Adv Sci Tech Eng Sys J 4(6):115–125
26. Yan X, Yang J, Sohn K, Lee H (2016) Attribute2image: conditional image generation from visual attributes. In: European conference on computer vision. Springer, pp 776–791
27. Yang LC, Chou SY, Yang YH (2017) Midinet: a convolutional generative adversarial network for symbolic-domain music generation. arXiv preprint arXiv:1703.10847
28. Zilly JG, Srivastava RK, Koutnık J, Schmidhuber J (2017) Recurrent highway networks. In: International conference on machine learning, pp 4189–4198