Sequence Generative Adversarial Networks for Music Generation with Maximum Entropy Reinforcement Learning

Mathias Rose Bjare[1] and David Meredith[2]

[1] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby, Denmark
muthissar@gmail.com
[2] Department of Architecture, Design and Media Technology, Aalborg University, Aalborg, Denmark
dave@create.aau.dk

Abstract. We introduce using maximum entropy reinforcement learning (MERL) for training Sequence Generative Adversarial Networks as a technique for obtaining more diverse samples and alleviate mode drop (MD) and mode collapse (MC) problems. We implemented generators using the ordinary REINFORCE algorithm, REINFORCE with reward baseline and MERL REINFORCE. We trained the models to learn to generate music using the Nottingham data set. We observed that, without pre-training, the algorithms fail to produce high reward trajectories. We showed that the pre-trained REINFORCE models mainly explore trajectories of their initial policies and we argue that the method might be more suitable for fine-tuning models than learning generative distributions from scratch.

Keywords: Neural Network, Generative Adversarial Network, Natural Language Processing, Maximum Entropy Reinforcement Learning, Music Information Retrieval

1 Introduction

This paper builds on the work of Yu et al. (2016) which tokenizes musical examples and uses a REINFORCE-based sequential generator to train a generative adversarial network (GAN) (Goodfellow et al., 2014) to generate music. Yu et al. (2016) primarily reported results on synthetic data generated by a target Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997). However, they also applied the technique to learn the melody tracks of folk music from the Nottingham dataset.[3] The dataset also contains chords and Lee et al. (2017) used the same dataset with a tokenization that stores melody octave, melody pitch-class, chord root octave, and chord class. Lee et al. (2017) reported issues of mode collapse or mode drop. Mode collapse (MC) occurs when

[3] http://www.iro.umontreal.ca/~lisa/deep/data

two modes within the original data distribution merge into one in the generator distribution. In this case, the generator will typically perform poorly on both modes. Mode drop (MD) occurs when the generator removes all probability density from one mode. This can happen if the generator prematurely (over-)fits the discriminator that has not learned all modes of the data.

We implement and discuss a novel approach to obtaining more diverse samples and solving the MD and MC problems by using the MERL variant of the REINFORCE algorithm. It is well-known that the REINFORCE algorithm can be improved using a reward baseline (Sutton and Barto, 2018). We therefore implemented and compared the ordinary REINFORCE algorithm with a version of REINFORCE using a reward baseline.

For the Nottingham dataset, Yu et al. (2016) and Lee et al. (2017) reported the performance of sequence GANs using a sentence version (as opposed to a corpus version) of the bilingual evaluation understudy (BLEU) score where the entire training set is used as reference translations. This is problematic for at least the following three reasons:

- A perfect score can be achieved if the n-grams of a generated example are present with multiplicity across different training examples.
- The score is non-decreasing when the number of samples in the training set is increased.
- The score does not account for the ordering of the n-grams.

Nonetheless we report the BLEU score for our models in order to facilitate comparison with the results obtained by Lee et al. (2017) and Yu et al. (2016).

We will perform a quantitative performance analysis of the implemented models on the Nottingham dataset in terms of negative log-likelihood (NLL). We report quantitative diagnostic GAN metrics. For the generator, we report the mean reward and standard deviation of the reward. For the discriminator, we report the precision and recall. Qualitatively we inspect samples that are: false-positive (FP), true-negative (TN), and generated from MD/MC.

2 Related work

Much work on using neural networks (NNs) for tokenized sequence generated music has been done using Recurrent Neural Networks (RNNs) (Oore et al., 2018). Huang et al. (2018) adopt the popular natural language processing transformer model, which is neither an RNN nor a convolutional neural network (CNN), but uses self-attention to generate symbolic music as a sequence of tokens. Lattner et al. (2016) use a convolutional energy-based model to obtain higher-level structured generated samples. Oord et al. (2016) use convolutional dilation in deep neural networks to generate audio samples of music from large corpora of audio files. Liu et al. (2019) attempt to generate audio samples of singing voices using GANs. Jaques et al. (2016) pretrain a recurrent neural network (RNN) and fine-tune the model using a MERL algorithm. The reward function is human-engineered and based on music-theoretical concepts. Schmidhuber (2010) developed a theory of creativity based on reinforcement learning (RL), where an intrinsic reward is given for the ability to generalize new examples.

3 Model

3.1 Sequence GAN

Sequence GAN (Yu et al., 2016) is a framework proposed as a way of training GANs (Goodfellow et al., 2014) for sequences of discrete tokens, $x = \{x_t : x_t \in V\}_{t=1}^{T}$, using RL, where, at step t', the Markov decision process (MDP) actions are V, the states are the long short-term memory (LSTM)-compressed context $\{x_t : x_t \in V\}_{t=1}^{t'}$ and the rewards are

$$R_{t'} = \begin{cases} D(x) & \textit{if } t' = T \\ 0 & \textit{otherwise} \end{cases}$$

where $D(x)$ is the output of a discriminator network.

3.2 Generator Algorithms

REINFORCE (Policy Gradient) Policy gradients express a policy π by a differentiable function (e.g. NN), which can be optimized using gradient descent (GD). The gradient is given by (Sutton and Barto, 2018):

$$\nabla_\theta J(\theta) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,x_t)\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(x_t|s_t)\left(\sum_{t'=t+1}^{T} R_{t'}\right)\right] \tag{1}$$

Where Eq. 1 is approximated by Monte Carlo sampling N trajectories.

REINFORCE with reward baseline If the discriminator dominates the GAN min-max game (Goodfellow et al., 2014) then for many trajectories $i \in I \subseteq 1, ..., N$, we have $D(\{x_{i,t}\}_{t=1}^{T}) \in [0, 1]$ is close to zero. The contribution of these actions $x_{i,t}$ will not affect the gradient, when computing Eq. 1. Therefore the agent is not discouraged from picking actions that are producing poor samples, but is only encouraged to pick actions that are producing good results. A way to mitigate this problem is by using a baseline. Let $b(s_{t'})$ be a state-dependent baseline. It can be shown (Sutton and Barto, 2018) that if Eq. 1 is replaced by:

$$\sum_{t=0}^{T} \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(x_t|s_t)\left(\sum_{t'=t+1}^{T} R_{t'} - b(s_{t'})\right)\right]$$

then the expected value of the gradient step is unchanged. A simple choice of a baseline is the mean reward:

$$b(s_{t'}) = \frac{1}{N}\sum_{i=1}^{N} D(\{x_{i,t}\}_{t=1}^{T})$$

Then $R_{t'} - b(s_{t'})$ can be perceived as an advantage (possibly negative) of selecting this action against a mean action.

MERL REINFORCE. When MDs and MCs occur, the entropy of the distribution typically decreases. MERL (Ziebart et al., 2008) deals with maximizing the expected reward while keeping the entropy of the policy high. A MERL version of the REINFORCE gradient can be derived as (Levine, 2018):

$$\nabla_\theta J_{MERL}(\theta) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,x_t)\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(x_t|s_t) \sum_{t'=t}^{T}\left(R_{t'} - \alpha \log \pi_\theta(x_{t'}|s_{t'}) - 1\right)\right] \tag{2}$$

Where α is a temperature constant controlling the relative importance of the entropy.

3.3 Discriminator

As this work focuses on generator algorithms, we shall use the discriminator from Yu et al. (2016) based on Kim et al. (2015). Tokens are one-hot encoded. 2-d convolution is performed with different filter sizes and finally max pooling over time resulting in one feature per filter. A single highway-layer (Srivastava et al., 2015) is used and finally, a logistic function.

4 Evaluation

The models are tested on the Nottingham dataset[4] consisting of 1037 traditional songs originally encoded in ABC notation, that have been re-encoded in MIDI format. In order to facilitate comparison with previous work, we use the tokenization procedure used by Lee et al. (2017), where tokens are created by combining melody pitch-class, melody octave, chord pitch-class, chord root octave and duration. Music containing tokens appearing less than 10 times in the data set was removed. The resulting vocabulary has $|V| = 3216$. The tokenized pieces are split into sequences of length 100. The tokens are embedded in a vector space using a random normal distribution.

[4] `http://abc.sourceforge.net/NMD/`

The qualitative performance, as measured by the NLL score and the BLEU score, are reported in Table 1. The table also shows diagnostic performance metrics of the trained generator and discriminator. For the generators, the mean reward $\bar{R}_T$ and standard deviation of the reward S_T are reported. For the discriminators, the precision p_d and recall r_d are reported. p_d is the proportion of samples judged to be real that are actually real. p_d therefore decreases as the number of FPs increases (an FP being a case where a generated sample is judged by the discriminator to be real). r_d is the proportion of real samples that are correctly judged by the discriminator to be real. r_d therefore decreases as the number of false-negatives (FNs) increases (an FN being a case where a real sample is judged by the discriminator to be generated). Of p_d and r_d we shall mainly focus on p_d since it drives the GAN learning.

Experiments are conducted on the following GAN trained models: ordinary REINFORCE (ordinary), REINFORCE extended with reward baseline (reward baseline), and MERL REINFORCE with an entropy temperature constant of $\alpha = 1$ and reward baseline (entropy $\alpha = 1$). Furthermore, for comparison, experiments are conducted on a NLL model trained using back-propagation in time (Rumelhart et al., 1986).

Like Yu et al. (2016) and Lee et al. (2017) we found that, without pre-training, the generator never generated samples that could fool the discriminator. Therefore the NLL model was used as an initialization for the generator, and the discriminator was pre-trained until convergence against the NLL model.

Table 1: Performance of the ordinary model, the model with reward baseline, and the MERL model.

Model	$H(p_{data}, p_{gen})$	**BLEU**	$H(p_{gen})$	$\bar{R}_T$	S_T	p_d	r_d
NLL	**2.58**	**0.375**	3.17	-	-	-	-
Ordinary	3.81	0.241	1.54	**0.016**	0.100	**0.993**	1.00
Reward baseline	4.39	0.146	4.11	0.012	0.077	1.00	**0.993**
MERL $\alpha = 1$	5.66	0.082	**5.26**	0.000	0.001	1.00	1.00

Table 1 shows that the likelihood and BLEU score are not improved by the GAN training. From the GAN diagnostic scores, it is found that the discriminator dominates the min-max game. The mean reward is highest for the ordinary model, and for all GAN models, except for the ordinary, we have $p_d = 1$. This explains that S_T is higher for the ordinary model, since for some sample $x \sim G_{org}(z)$ we have $D(x) > 0.5$. A good sample x_g from the ordinary model with $D(x_g) = 0.947$ is shown in Figure 1, which consists of a motif repeated twice with variation on the later repetition. The simple motif and chord progression subjectively sounds folky, suggesting that some stylistic traits of the training set have been learned.

The recall, r_d, was found to be equal to 1 for all except the reward baseline model. The MERL version model has the highest entropy at the expense of all other performance metrics. Experiments with $\alpha < 1$ did not result in any significant improvements. The ordinary model has the lowest entropy.

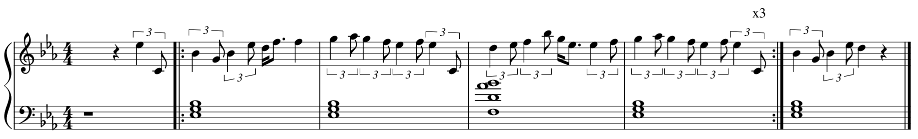


Fig. 1: sample x_g. A four bar structure is repeated 3 times. The four bar structure is build from a 2-bar motif-like structure, which is repeated with some variation. The tonic and the dominant is played appropriately during the motif and its variation.

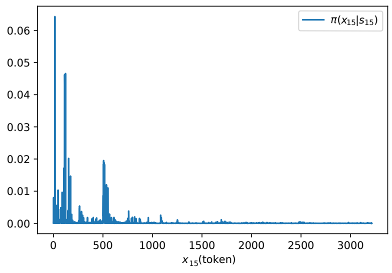
5 Discussion

The GAN-trained models failed to produce high reward trajectories (FPs) without pretraining similarly to what is reported by Yu et al. (2016) and Lee et al. (2017). The MERL model has the lowest mean reward, which might be caused by the $\log(\pi(x_t|s_t))$ term in Eq. 2 weakening the importance of the action's reward.

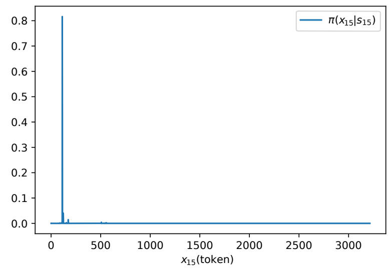
Reward baseline did not seem to improve the quantitative results significantly. A possible explanation is that the mean and variance of the reward (Table 1) are already very low. Therefore the discriminator is confident that all trajectories are generated, and arguably no trajectory is better than the others; hence the discrimination between good and poor trajectories by the mean reward baseline is artificial.

For a fixed degenerate sample trajectory, $s_0, x_0, r_0, ..., s_{T-1}, x_{T-1}, r_{T-1}, s_T$, generated by the ordinary model, $\pi(x_t|s_t)$ and $\mathcal{H}(\pi(\cdot|s_t))$ are shown for $t = 15$ for all models in Figure 2. This illustrates that the GAN-trained models are obtained from the LL distribution by dropping most modes. Similar results are obtained for $t > 15$. As the agent samples from the initial maximum likelihood distribution, the agent will tend to drop modes of the maximum likelihood estimate corresponding to TN trajectories and amplify modes corresponding to FP trajectories. This fundamental on-policy behaviour of the REINFORCE algorithm is not mitigated by its MERL variant.

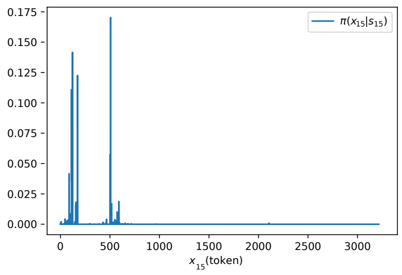
The GAN-trained models do not increase the likelihood as reported by Yu et al. (2016) on synthetic data. A likely explanation is that NLL is reported by $\mathcal{H}(p_{gen}, p_{data})$ instead of $\mathcal{H}(p_{data}, p_{gen})$ as reported in Table 1. No performance gain in the BLEU score is observed as reported by Lee et al. (2017) and Yu et al. (2016), but for the reasons mentioned in Section 1, the BLEU score is a poor metric for music generation.



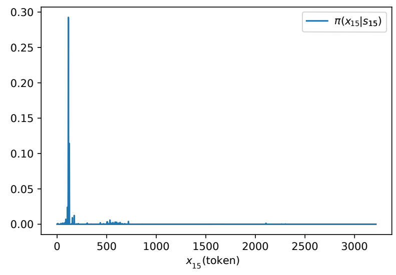
(a) NLL model.
$\mathcal{H}(\pi(\cdot|s_{15})) = 5.302$



(b) Ordinary model.
$\mathcal{H}(\pi(\cdot|s_{15})) = 0.9276$



(c) Reward baseline model.
$\mathcal{H}(\pi(\cdot|s_{15})) = 2.937$



(d) MERL $\alpha = 1$ model.
$\mathcal{H}(\pi(\cdot|s_{15})) = 2.479$

Fig. 2: Shows $\pi(\cdot|s_{15})$ for the trained models. A few modes from the NLL model are expanded in all the GAN trained models and the others are dropped. This is also the case for the MERL model.

6 Conclusion

The performance of sequence GAN on the Nottingham dataset using the ordinary REINFORCE, REINFORCE with reward baseline, and MERL REINFORCE was evaluated. Without using pre-training all three generators failed to raise the mean reward (lower the discriminator precision). All three generators were then pre-trained using the NLL estimate, which allows the generator to generate FPs, but the min-max game was still dominated by the discriminators. Ordinary REINFORCE achieved the highest mean reward and lowest NLL of the GAN-trained models. Since the mean and variation of the reward baseline REINFORCE is very low, arguably no trajectories were better or worse than any others, which could explain the lack of performance gain by the reward baseline REINFORCE algorithm as opposed to the original model. MERL REINFORCE achieved the lowest mean reward and highest NLL score. A likely explanation is that the added entropy further weakens the low learning signal. The MERL REINFORCE-trained model achieved the highest entropy but failed to fundamentally solve the MC/MD problem on selected modes. It is reasoned that the Monte Carlo trajectories of REINFORCE do not deviate too much from their pretrained modes. Therefore if the reward is high the mode will be amplified; if it is low, it will be dropped. This makes the agents reluctant to explore new modes and sequence GAN using REINFORCE variants might be better suited for fine-tuning generative models rather than learning generative distributions from scratch.

7 Future work

As the generators carry out Monte Carlo sampling of full trajectories, reducing the number of tokens could severely reduce the variance of the trajectories, especially for models trained without pre-training. Some options could be to (1) transpose all training examples to a fixed key, or (2) to drop the chord octave, or (3) to interleave chord and melody events in the sequence instead of combining them in a token (as is done in Huang et al. (2018)). It is possible to consider other RL algorithms, such as actor critique methods or value-based methods. However, if the reward signal changes too much, it is likely that also a value-based network would perform poorly. In order to better understand the possibilities of using reinforcement learning for GAN training on sequences, it is crucial to determine how the reward function changes during training. The learning signal is weak since it is only nonzero on the last state transition. As a convolutional network is used for the discriminator, it is likely that the smaller filters find low-scale features, that could be used to produce a reward signal for states corresponding to sequence prefixes. This could potentially reduce the variance of the reward signals, allow discounting and online learning.

References

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. *arXiv e-prints*, page arXiv:1406.2661.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Huang, C. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Hawthorne, C., Dai, A. M., Hoffman, M. D., and Eck, D. (2018). An improved relative self-attention mechanism for transformer with application to music generation. *CoRR*, abs/1809.04281.

Jaques, N., Gu, S., Turner, R. E., and Eck, D. (2016). Tuning recurrent neural networks with reinforcement learning. *CoRR*, abs/1611.02796.

Kim, Y., Jernite, Y., Sontag, D. A., and Rush, A. M. (2015). Character-aware neural language models. *CoRR*, abs/1508.06615.

Lattner, S., Grachten, M., and Widmer, G. (2016). Imposing higher-level structure in polyphonic music generation using convolutional restricted boltzmann machines and constraints. *CoRR*, abs/1612.04742.

Lee, S., Hwang, U., Min, S., and Yoon, S. (2017). A seqgan for polyphonic music generation. *ArXiv*, abs/1710.11418.

Levine, S. (2018). Reinforcement learning and control as probabilistic inference: Tutorial and review. *CoRR*, abs/1805.00909.

Liu, J.-Y., Chen, Y.-H., Yeh, Y.-C., and Yang, y.-h. (2019). Score and lyrics-free singing voice generation.

Oord, A. van den., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499.

Oore, S., Simon, I., Dieleman, S., Eck, D., and Simonyan, K. (2018). This time with feeling: Learning expressive musical performance. *CoRR*, abs/1808.03715.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247.

Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Highway networks. *CoRR*, abs/1505.00387.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction.* The MIT Press, second edition.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2016). Seqgan: Sequence generative adversarial nets with policy gradient. *CoRR*, abs/1609.05473.

Ziebart, B. D., Maas, A., Bagnell, J., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. *AAAI Conference on Artificial Intelligence (AAAI)*, page 1433–1438.