Georgia Tech Center for Music Technology
College of Design

SeqGAN for Music Generation

Benjamin Genchel[1], Ryan Rose[2]

[1] bgenchel3@gatech.edu

[2] rrose37@gatech.edu

Contents

- What is SeqGAN?
  - Why was it developed?
  - How does it work?
- SeqGAN for Music Generation
  - Inputs
  - Baseline
- Evaluation
- Next Steps

Georgia Tech Center for Music Technology
College of Design

What/Why is SeqGAN?

Georgia Tech Center for Music Technology
College of Design

- DNN architecture aimed at solving the problem of using **Generative Adversarial Networks** (GAN) for sequence generation.

- Issues with standard GAN models in this domain:
  - **Designed for continuous outputs** – the loss from the discriminator is used to guide the generator to slightly adjust its output in a continuous space towards something more realistic. **How to adjust 'slightly' when only certain values are allowed?**

  - **Designed to evaluate a complete output** – in sequence generation, a loss (and thus a discriminator judgement) is needed at each step of the generation. **How can a discriminator make a judgement on the realism of something incomplete?**

- **SeqGAN** solves these issues by treating the generator as a stochastic policy (probabilistically choosing 'actions' in an 'environment') in a reinforcement learning framework.

How does SeqGAN work?

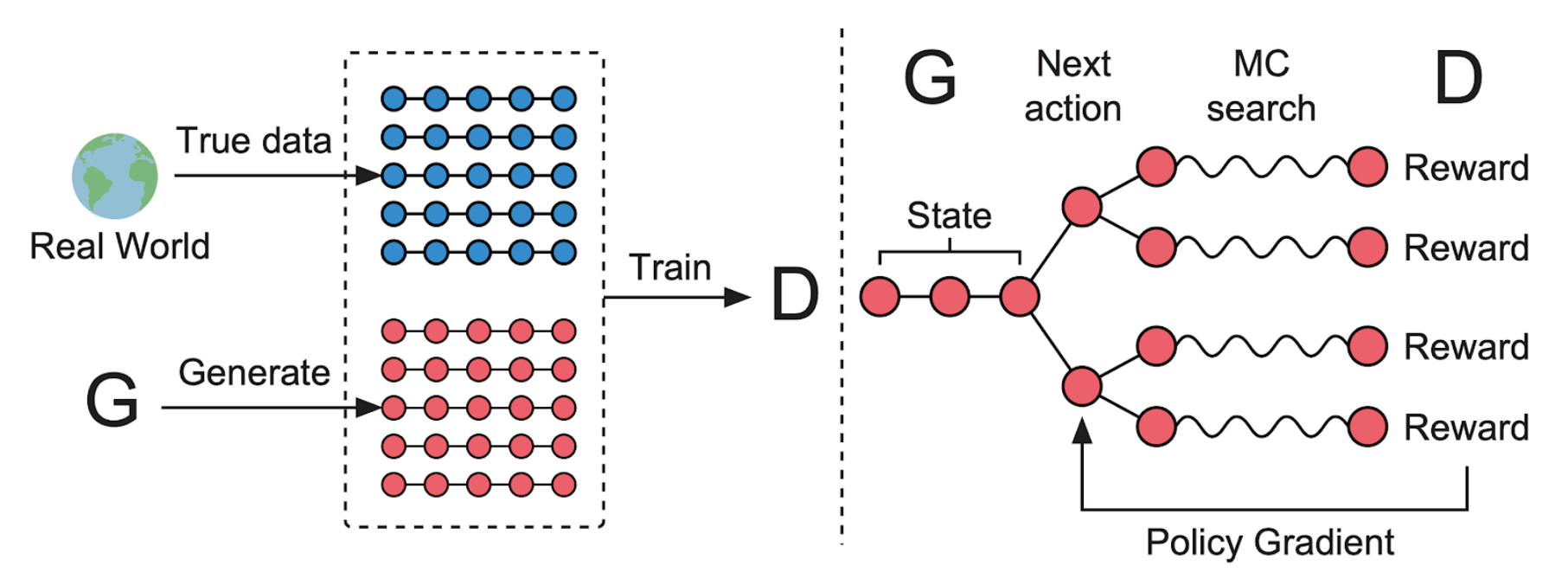
Georgia Tech Center for Music Technology
College of Design

- Generator – Long Short Term Memory Recurrent Neural Network (LSTM-RNN)
- Discriminator – Convolutional Neural Network (CNN)
- Need to provide a reward (for a full sequence) at every time. How?
  - **Monte Carlo Rollouts** (Just like AlphaGo, which makes this project inherently successful by association) – at every intermediate time step, sample unknown tokens based on the current policy to create a full sequence.
  - Current policy is modeled by the Generator G_θ
  - Final action-value function is the average of Discriminator output from final sequence and all rollouts.

How does SeqGAN work?

Georgia Tech Center for Music Technology
College of Design

- Pre train Generator using MLE (standard sequence model procedure)
- Pre train Discriminator with Pre trained Generator
- Adversarially Train Generator and Discriminator using policy gradient updates



SeqGAN for Music

Georgia Tech Center for Music Technology
College of Design

- Model music as a sequence of ticks
  - 24 ticks per quarter note = 96 ticks per bar
  - 1 tick = length 37 vector
    - 36 note possibilities spanning 3 octaves + rest
- Add conditioning information by appending to each tick
- Train SeqGAN to generate music 4 to 8 bars at a time
- **Original Paper**
  - Trained on Nottingham Dataset 'sampling' each song at 1 tick = 0.4 seconds.
  - Evaluated using BLEU score, and shown to perform better than standard MLE sequence learning, though not much information is provided

Dataset

Georgia Tech Center for Music Technology
College of Design

- 200+ Bebop (Jazz, Late 1930s – 1970s) Lead Sheets
  - Originally in musicXML format, converted to tick format.
- Why Lead Sheets?
  - Monophonic Melody + Chord Information
    - monophonic melody much easier to deal with both conceptually for us developers and practically for the network.
    - chord information gives useful musical context for notes.

Current Progress

Georgia Tech Center for Music Technology
College of Design

- Original model implemented and running.
- Dataset half finished
  - Transcribing 200 Lead Sheets from the Real Bebop Book
  - Supplementing with the Charlie Parker Omnibook
- Data pipeline fully implemented and tested.

Next Steps

Georgia Tech Center for Music Technology
College of Design

- Continue/Finish creating dataset
- Confirm model results on Nottingham Dataset match up with original paper.
- Adapt model for Bebop dataset.
- Perform original evaluation on Bebop dataset.
  - BLEU Score.
  - **MSE** (Mean Squared Error)
- Perform Statistical Evaluation (Yang, Lerch 2018)
  - Compares distribution of generated data to training data along a number of different dimensions.