F(O)MG
Few-shot Music Generation

Anand, Ben & Suhail

University of British Columbia

April 4, 2018

Generate music matching styles of individual artists with few-shot meta learning.

- Number of songs per artist is limited.

Music generation is a hard problem

- Long sequences / concept of time
- Relatively large vocabulary
- Not all generated sequence produces music

Introduction

Comparison with Text

Text – *Drive Man Car Blue*

Music – $G_{on}\ T_{0.2}\ D_{off}\ A_{off}\ F_{on}\ T_{0.1}$

- A random jumbo of words still conveys meaning
- A random MIDI sequence is usually senseless

Data Processing

MIDI and Vocabulary

Our MIDI format contains,

- 16 instruments
- ON / OFF for each of 128 pitches per instrument
- 32 velocity levels per instrument
- 100 time-shift controls
- 4710 controls in total

To generate a single note, must set velocity, turn note ON, advance time, and turn note OFF.

Model

MAML

Model Agnostic Meta Learning (MAML)[1]

- Learn a good initial guess
- Converge to any solution in a few iterations

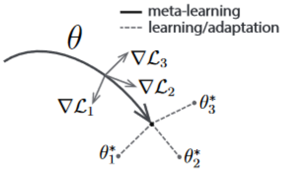

Figure: Illustration of meta learning

[1] https://arxiv.org/pdf/1703.03400.pdf

Model

MAML

Main idea of MAML,

1. Instantiate `meta_net` and `learner_net`
2. Copy params from `meta_net` to `learner_net`
3. Train model as usual with `support` set on `learner_net`
4. Query the model with `query` set on `learner_net`
5. Calculate gradient from (4) and apply to `meta_net`
6. Repeat (2) – (5)

Experiment

Dataset and Environment

Dataset obtained from `freemidi`[2]

- Training/Validation/Testing – 296 / 53 / 53 artists
- 15 songs from each artist
- Divided into support set of 10, and query set of 5
- Each input sequence is truncated to 150

Training is done on `Ubuntu 16.04` with `GTX 1080 Ti`.

- Baseline – 6 to 7 Hours
- MAML – $\sim$ 40 hours

[2] `https://freemidi.org`

Experiment

Training

Training `MAML`

- `meta_net` picks 3 random artists at a time
- `learner_net` trains on support set for 10 iterations
- Return loss on query set to `meta_net`
- 10 epochs in total

Training `Baseline` only involves the second step.
This is just encoder-decoder.

Result

Demo Comparison

We'll play `query` input, `Baseline`, followed by `MAML`.

Result

Loss Comparison

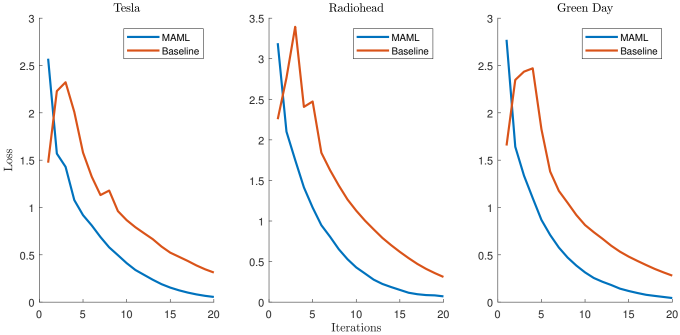

Figure: Evaluation loss comparison for three different artists.

Result

Analysis & Discussion

Some general observations,

- `MAML` converges faster than `Baseline`
- Convergence behaviour is more smooth
- `MAML` often starts at a higher loss, but converges to a lower minimum

We attempted and learned,

- `SNAIL`[3] – Simple neural attentive meta learner.
  We couldn't adapt to fit our task.
- Music and text generation is *waaaay* different.
- Meta-learning rocks!

[3] `https://arxiv.org/pdf/1707.03141.pdf`

Few things we want to try,

- `MIDI-GLoVe`[4] – `word2vec` for MIDI
- `Reptile`[5] – Modified `MAML` with smaller computation and memory footprint

[4] https://github.com/brangerbriz/midi-glove
[5] https://blog.openai.com/reptile/

Conclusion

THE UNIVERSITY OF BRITISH COLUMBIA

Thank you!

References

MIDI GloVe.
https://github.com/brangerbriz/midi-glove.

Reptile: A Scalable Meta-Learning Algorithm.
https://blog.openai.com/reptile/.

C. Finn, P. Abbeel, and S. Levine.
Model-agnostic meta-learning for fast adaptation of deep networks.
CoRR, abs/1703.03400, 2017.