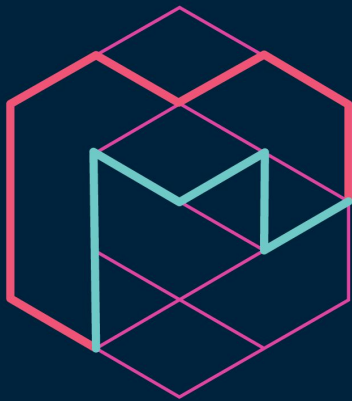


# Magenta



Slides: <http://goo.gl/bXbQox>

[cinjon@google.com](mailto:cinjon@google.com) / [fjord@google.com](mailto:fjord@google.com)

[g.co/magenta](http://g.co/magenta)



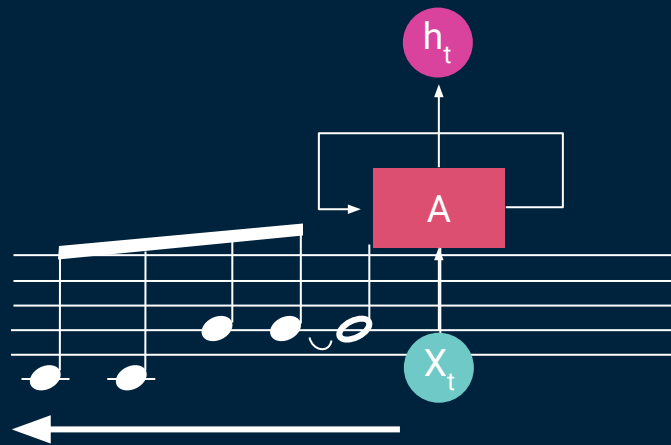
# MACHINES CAN DREAM IMAGES

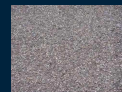
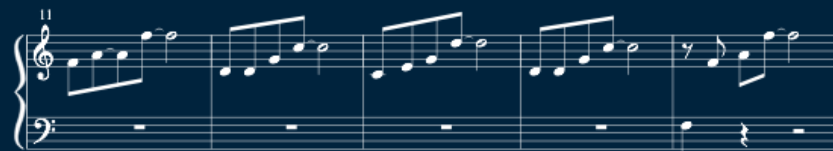






# Prediction









**"THE CYBORG COLLAB" V.2**  
**Google Magenta**

# [github.com/tensorflow/magenta](https://github.com/tensorflow/magenta)

 This repository Search

Pull requests Issues Gist

 tensorflow / magenta

Unwatch 335 Star 2,864 Fork 282

[Code](#) [Issues 4](#) [Pull requests 3](#) [Pulse](#) [Graphs](#) [Settings](#)

Magenta: Music and Art Generation with Machine Intelligence — Edit

101 commits

1 branch


0 releases

17 contributors

Branch: master New pull request

Create new file Upload files Find file Clone or download

README.md

 **magenta**

**Magenta** is a project from the [Google Brain team](#) that asks: Can we use machine learning to create compelling art and music? If so, how? If not, why not? We'll use [TensorFlow](#), and we'll release our models and tools in open source on this GitHub. We'll also post demos, tutorial blog postings, and technical papers. Soon we'll begin accepting code contributions from the community at large. If you'd like to keep up on Magenta as it grows, you can read our [blog](#) and or join our [discussion group](#).



# magenta/models

tensorflow / magenta

Unwatch 331 Star 2,850 Fork 279

Code Issues 5 Pull requests 2 Pulse Graphs Settings

Branch: master magenta / magenta / models /

Create new file Upload files Find file History

adarob committed with fredbertsch Add MIDI interface (#126) Latest commit 66867a6 4 hours ago

..

attention_rnn	Add MIDI interface (#126)	4 hours ago
basic_rnn	Add MIDI interface (#126)	4 hours ago
lookback_rnn	Add MIDI interface (#126)	4 hours ago
shared	Add MIDI interface (#126)	4 hours ago
README.md	Add SequenceGenerator interface (#119)	5 days ago

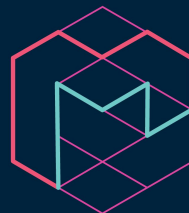
README.md

## Models

This directory contains the various Magenta models.

## Generators

All generators for NoteSequence-based models should expose their generator functionality by implementing the `BaseSequenceGenerator` abstract class defined in `lib/sequence_generators.py`. This allows all generators to communicate

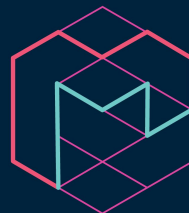


# Your Turn



# Using a model

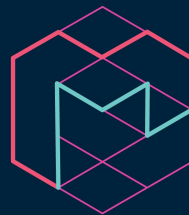
- Training a model is basically asking the computer to find patterns in data
- Get a bunch of midi files
- Turn those files into a dataset for the model
  - Translates midi into a format that's easier to work with (NoteSequence)
  - Extracts melody information from the NoteSequences
- Train the model to predict the next note in a given sequence
- Prime the model with a starting sequence
- Ask the model to continue the sequence



# Getting MIDI files

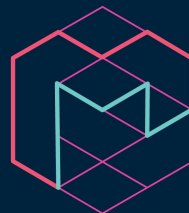
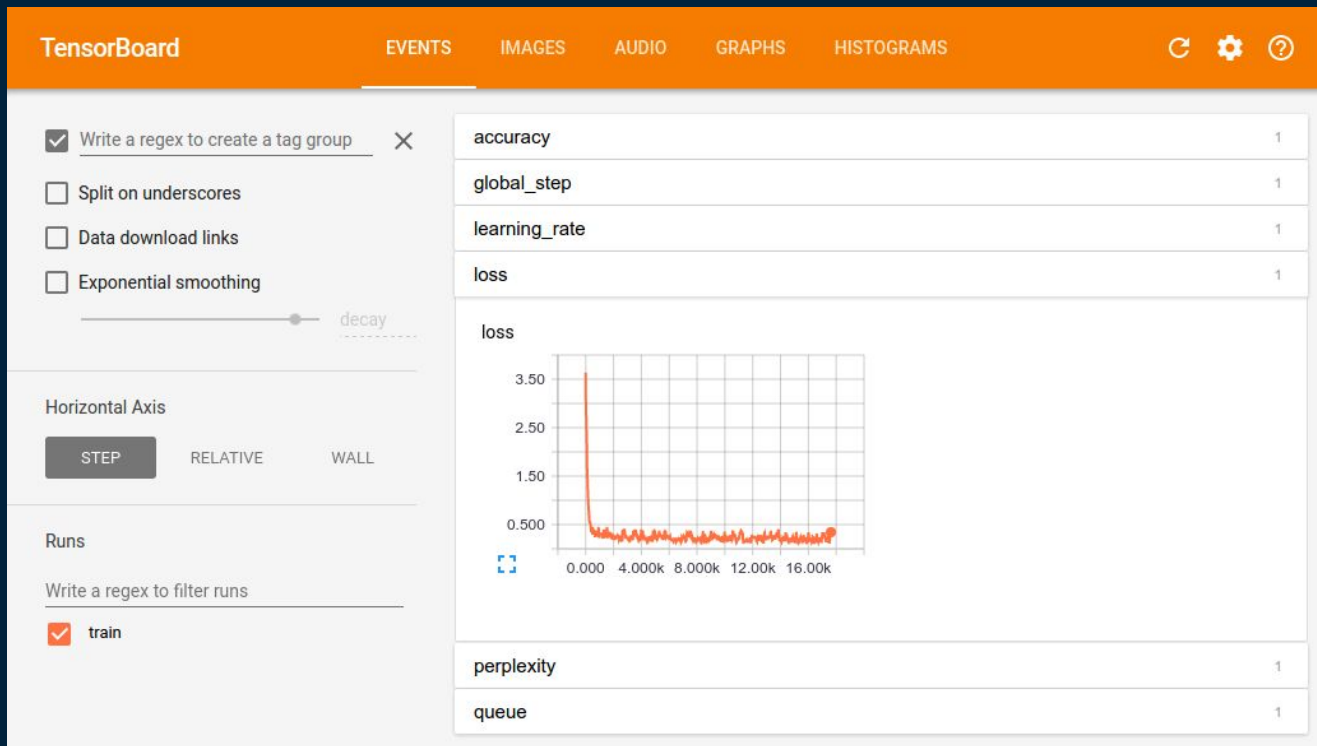
- Find a collection that matches the music you want to generate.
- Get a large quantity of files (at least 50).
  - Colin Raffel has a collection he'd be happy to share with you
- Run the commands to create your dataset.
- Example:

```
mkdir /tmp/midiworld  
cd /tmp/midiworld/  
wget -r --accept "*.mid" --level 1 "http://www.midiworld.com/classic.htm/"
```



# Training the model

- Use TensorBoard to monitor your progress



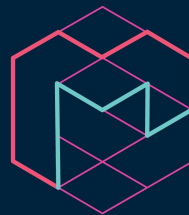


# Or, just use our pre-trained model

Download the checkpoint file: [goo.gl/fuygaK](https://goo.gl/fuygaK)

Run the generator with `--checkpoint_file` instead of `--run_dir`:

```
bazel run //magenta/models/lookback_rnn:lookback_rnn_generate -- \
--checkpoint_file=/tmp/lookback_rnn.checkpoint \
--hparams="{ 'batch_size':64, 'rnn_layer_sizes':[64,64] }" \
--output_dir=/tmp/lookback_rnn/generated \
--num_outputs=10 \
--num_steps=128 \
--primer_melody="[60]"
```

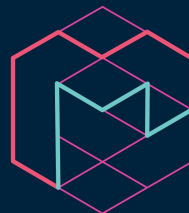


# Generating a new sequence: command line

```
bazel run //magenta/models/lookback_rnn:lookback_rnn_generate -- \  
  --run_dir=/tmp/lookback_rnn/logdir/run1 \  
  --hparams="{ 'batch_size':64, 'rnn_layer_sizes':[64,64] }" \  
  --output_dir=/tmp/lookback_rnn/generated \  
  --num_outputs=10 \  
  --num_steps=128 \  
  --primer_melody="[60]"
```

...

```
INFO:tensorflow:Wrote 10 MIDI files to /tmp/lookback_rnn/generated
```



# Generating a new sequence - MIDI Interface

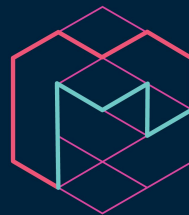
[magenta/interfaces/midi](https://github.com/tensorflow/magenta/tree/master/interfaces/midi)

The screenshot shows the GitHub repository page for `tensorflow/magenta`, specifically the `interfaces/midi` directory. The repository has 333 Unwatch, 2,854 Stars, and 280 Forks. The current branch is `master`. The commit history shows a recent commit by `cghawthorne` adding a sequence diagram for the MIDI interface. The file list includes `BUILD`, `README.md`, `midi.png`, and `midi.py`. The `README.md` file is expanded, showing the title **Magenta MIDI Interface** and a description: "This interface allows you to connect to a model generator via a MIDI controller and synthesizer. These can be either 'hard' or 'soft' components. Note that you can only interface with a trained models that have a `SequenceGenerator` defined for them."

```
graph LR
    MIDI[MIDI Instrument] -- MIDI Events --> Magenta[Magenta MIDI Interface]
    Magenta --> SeqGen[Sequence Generator]
    SeqGen --> TF[TensorFlow]
```



# How does it work?



# Sequence Generators

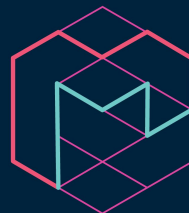
## GenerateSequenceRequest

```
input_sequence {
  ticks_per_beat: 96
  tempos {
    bpm: 120.0
  }
  notes {
    pitch: 60
    velocity: 100
    start_time: 0
    end_time: 0.125
  }
}
generator_options {
  generate_sections {
    start_time_seconds: 0.25
    end_time_seconds: 16.0
  }
}
```

## GenerateSequenceResponse

```
generated_sequence {
  ticks_per_beat: 96
  tempos {
    bpm: 120.0
  }
  notes {
    pitch: 60
    velocity: 100
    start_time: 0
    end_time: 0.125
  }
  notes {
    pitch: 55
    velocity: 100
    start_time: 1.625
    end_time: 1.875
  }
}
```

<https://github.com/tensorflow/magenta/blob/master/magenta/protobuf/generator.proto>





# Sequence Generator Interface

```
class BaseSequenceGenerator(object):
```

```
    @abc.abstractmethod
```

```
    def _initialize(self, checkpoint):
```

```
        pass
```

```
    @abc.abstractmethod
```

```
    def _close(self):
```

```
        pass
```

```
    @abc.abstractmethod
```

```
    def _generate(self, generate_sequence_request):
```

```
        """
```

```
        Args:
```

```
            generate_sequence_request: The request for  
                                        generating a sequence
```

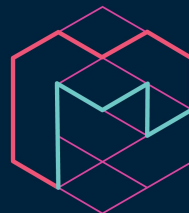
```
        Returns:
```

```
            A GenerateSequenceResponse proto.
```

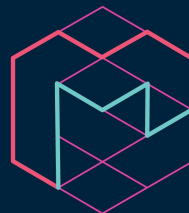
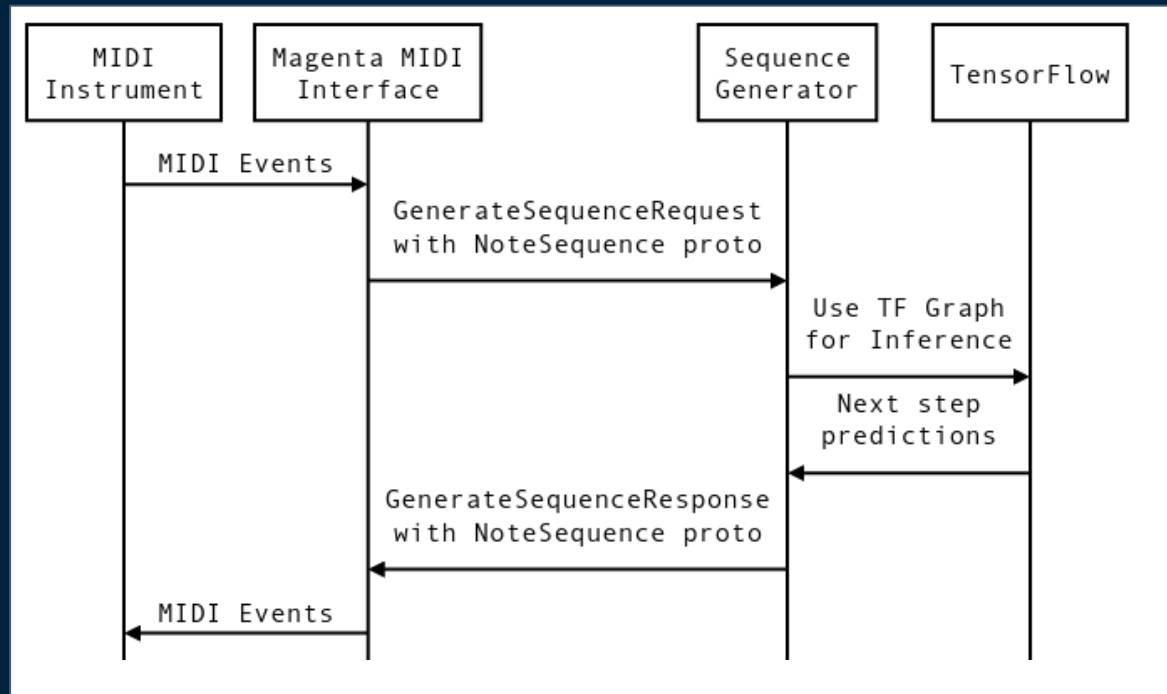
```
        """
```

```
        pass
```

[https://github.com/tensorflow/magenta/blob/master/magenta/lib/sequence\\_generator.py](https://github.com/tensorflow/magenta/blob/master/magenta/lib/sequence_generator.py)

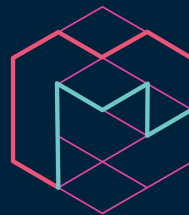


# Sequence Generators



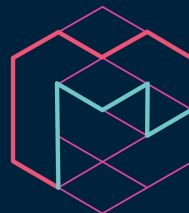
# Project ideas

- Create a new interface.
  - Rather than a keyboard for input, why not take input from another data source?
  - You could create priming melodies based on the current weather!
  - As long as your interface speaks the Sequence Generator protocol, you can connect it.
- Create a new model.
  - Implement BaseSequenceGenerator and *any* interface can connect to your model.



# Important Links

- GitHub: <https://github.com/tensorflow/magenta>
- Blog: <https://magenta.tensorflow.org/>
- MIDI Interface instructions:  
<https://github.com/tensorflow/magenta/tree/master/magenta/interfaces/midi>
- BaseSequenceGenerator class:  
[https://github.com/tensorflow/magenta/blob/master/magenta/lib/sequence\\_generator.py](https://github.com/tensorflow/magenta/blob/master/magenta/lib/sequence_generator.py)
- Sequence Generator protocol:  
<https://github.com/tensorflow/magenta/blob/master/magenta/protobuf/generator.proto>
- Pre-trained lookback\_rnn checkpoint: <https://goo.gl/fuygaK>



# Magenta



Slides: <http://goo.gl/bXbQox>

cinjon@google.com / fjord@google.com

[g.co/magenta](http://g.co/magenta)

