

Magentaで遊ぼう

2016/09/17

- <https://github.com/PyDataOkinawa/meetup017>
- <http://pydataokinawa.connpass.com/event/39806/>

Magentaとは

- Machine Learning (TensorFlow) + Art
- Official HP
- GitHub
- Douglas Eckさんが発起人
- Douglas Eckさんのインタビュー @ Talking Machines



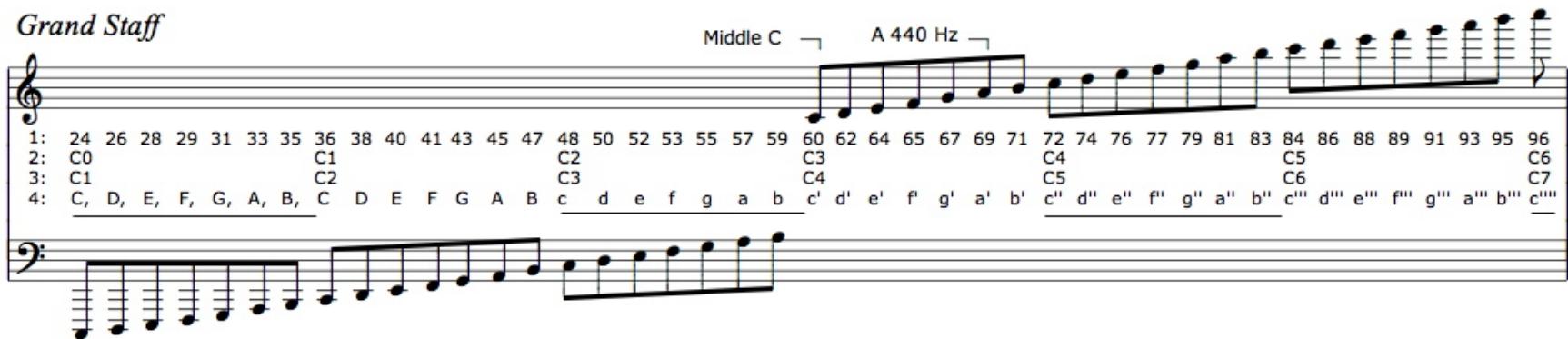
magenta

Magentaとは

- 2016/06/01に始まったばかりのプロジェクト
- 現時点では単旋律のメロディーのみサポート
- 現在サポートされているモデルは3つ
 - Basic RNN
 - Lookback RNN
 - Attention RNN

イベント

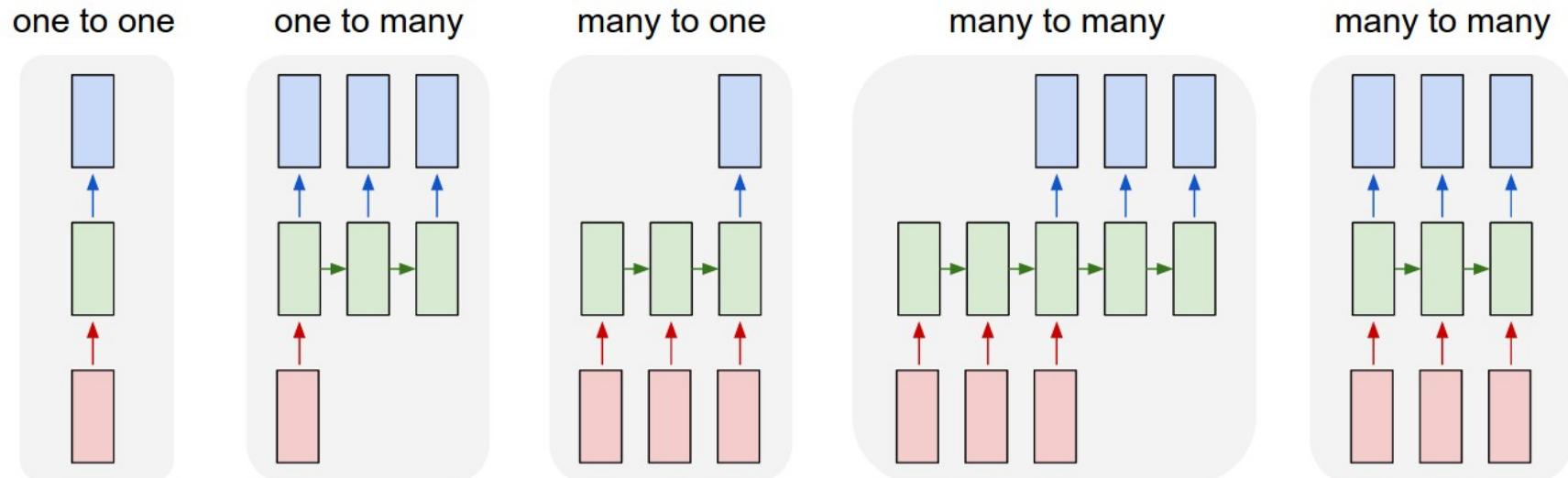
- 有音 : note-on event for each pitch (0-127)
- 消音 : note-off event (-1)
- イベントなし : no event (-2)



<http://www.theoreticallycorrect.com/Helmholtz-Pitch-Numbering/>

Basic RNN

- 基本となるモデル
- 入力
 - 直前のイベント（ワンホットベクトル）
- 出力
 - 現在のイベント（ワンホットベクトル）



The Unreasonable Effectiveness of Recurrent Neural Networks

全体の流れ

1. MIDIファイルを `NoteSequence` protoに変換
2. 学習用・検証用のデータを作成
3. モデルの学習と検証
4. モデルから音楽生成

全体の流れがわかるスクリプトファイル

- `run_basic_rnn_train.sh`
- `run_beatles.sh`

初期のディレクトリ構成

```
|- midi
  |- beatles
    |- The_Beatles_-_Let_It_Be.mid
|- script
  |- build_dataset.sh
  |- generate_melodies.sh
  |- midi.sh
  |- run_beatles.sh
  |- train_rnn.sh
```

Step 1: MIDIファイルをNoteSequence protoに変換

MIDIを NoteSequence protosに変換

build_dataset.sh

```
bazel build magenta/scripts:  
convert_midi_dir_to_note_sequences  
  
. /bazel-bin/magenta/scripts/  
convert_midi_dir_to_note_sequences  
--midi_dir=$MIDI_DIR  
--output_file=$SEQUENCES_TFRECORD  
--recursive
```

`./build_dataset.sh` 後の構成

MIDI files -> NoteSequence protos

```
.  
└── midi  
    └── beatles  
        └── The_Beatles_-_Let_It_Be.mid  
└── script  
    ├── build_dataset.sh  
    ├── generate_melodies.sh  
    ├── midi.sh  
    ├── run_beatles.sh  
    └── tmp  
        └── notesequences.tfrecord  
    └── train_rnn.sh
```

Step 2: 学習用・検証用のデータを作成

学習用・検証用データセットを作成

train_rnn.sh

```
bazel run //magenta/models/$RNN_TYPE:  
${RNN_TYPE}_create_dataset -- \  
--input=$SEQUENCES_TFRECORD \  
--output_dir=$DATASET_DIR \  
--eval_ratio=$EVAL_RATIO
```

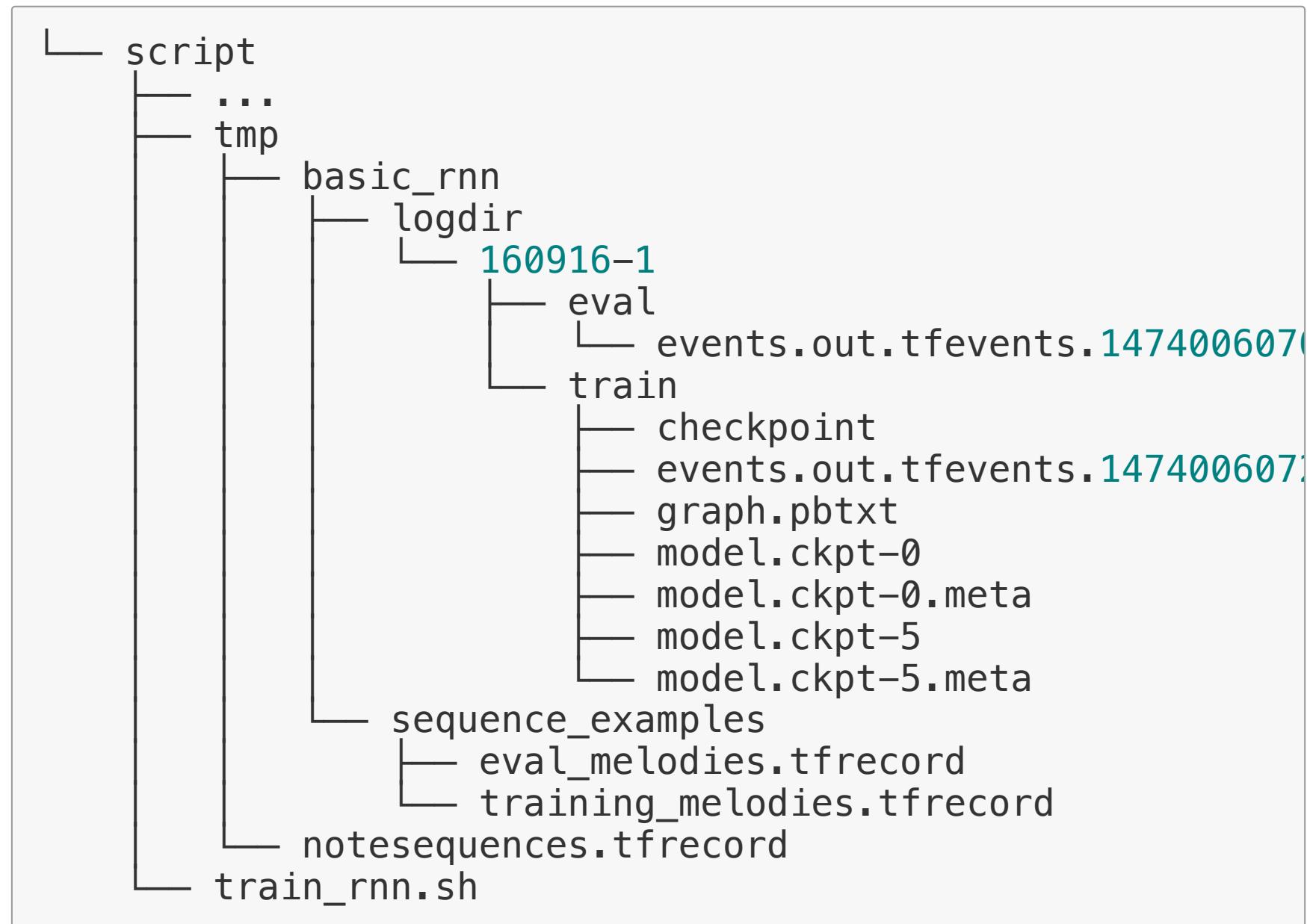
Step 3: モデルの学習と検証

モデルの学習と検証

train_rnn.sh

```
./bazel-bin/magenta/models/${RNN_TYPE}/${RNN_TYPE}_train  
--run_dir=$RUN_DIR  
--sequence_example_file=$TRAIN_DATA  
--hparams=$HPARAMS  
--num_training_steps=$NUM_TRAINING_STEPS  
--eval=false &  
  
./bazel-bin/magenta/models/${RNN_TYPE}/${RNN_TYPE}_train  
--run_dir=$RUN_DIR  
--sequence_example_file=$EVAL_DATA  
--hparams=$HPARAMS  
--num_training_steps=$NUM_TRAINING_STEPS  
--eval=true &  
  
tensorboard --logdir=$LOG_DIR --port=$TB_PORT &
```

`./train_rnn.sh -r bash_rnn` 後の構成



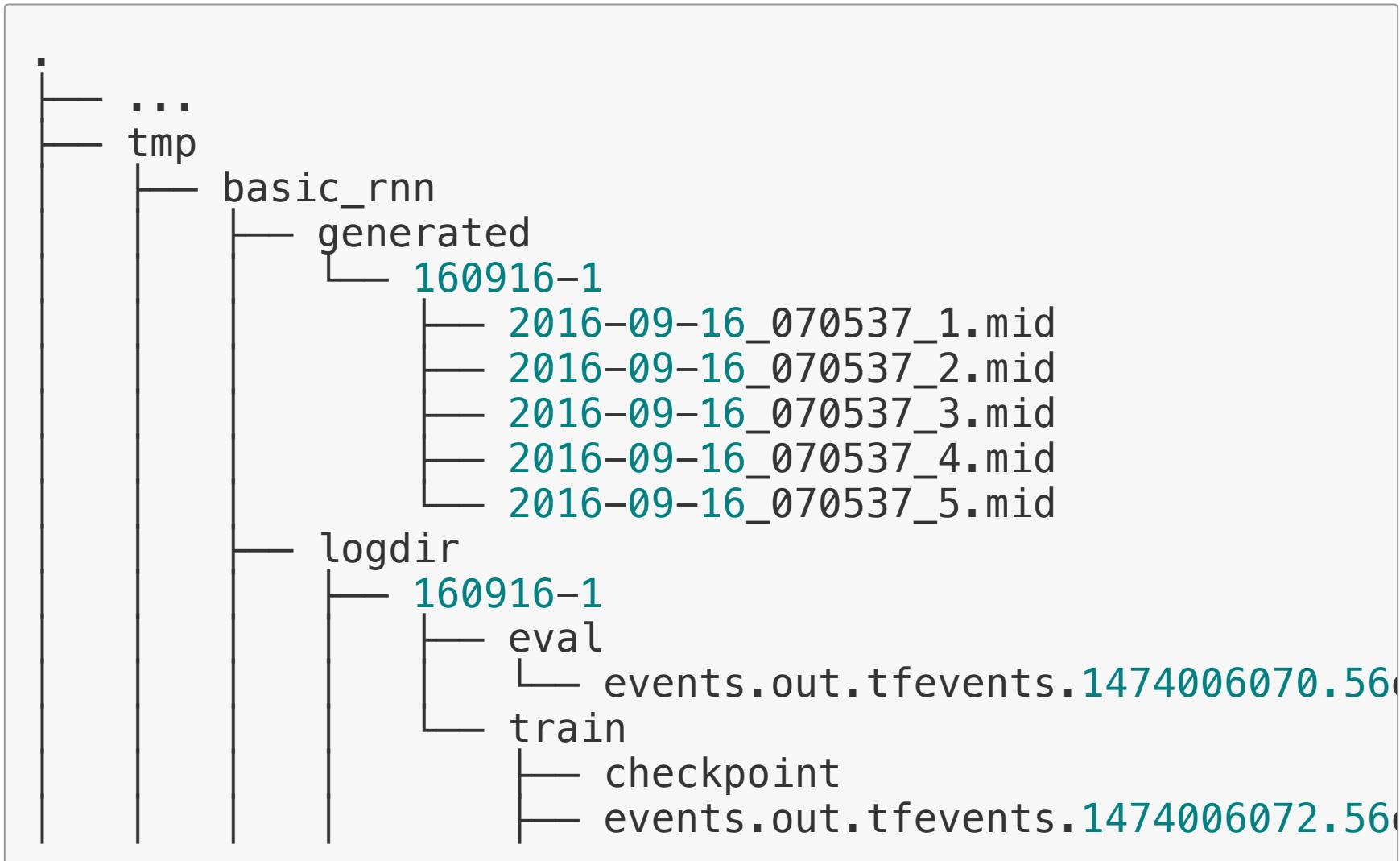
Step 4: モデルから音楽生成

モデルから音楽生成

```
bazel run //magenta/models/${RNN_TYPE}:
${RNN_TYPE}_generate -- \
--run_dir=$RUN_DIR \
--hparams=$HPARAMS \
--output_dir=$OUTPUT_DIR \
--num_outputs=5 \
--num_steps=512
#--primer_midi=${PRIMER_PATH}
#--primer_melody="[60]"
```

`./generate_melodies.sh` 後の構成

```
./generate_melodies.sh -r basic_rnn -n 160916-1
```



Lookback RNN

カスタム入力（3種類）

- 1小節、2小節前のイベント
- 直前が1小節、2小節前のイベントの繰り返しかどうか
- 小節の中の位置情報

```
Step 1: [0,0,0,0,1]
Step 2: [0,0,0,1,0]
Step 3: [0,0,0,1,1]
Step 4: [0,0,1,0,0]
```

カスタムラベル（2種類）

- 2小節前の繰り返し (repeat-2-bars-ago)
 - 1小節前の繰り返し (repeat-1-bar-ago)
- RNNのセルを使わず情報を保持。名前の由来。

Attention RNN

- 最近のDeep Learning (NLP) 界隈におけるAttention事情
- Generating Long-Term Structure in Songs and Stories
- Attention and Augmented Recurrent Neural Networks

おまけ：音楽用語英日対応表

英語	日本語
Note	音符
Bar	小節
Pitch	音の高さ