ML6

DATA SCIENCE GHENT

OZARK HENRY

boobook

DIGITSER

Agenda

- Today 19.30: explanation of problem, dataset & forming teams
- Today 20.15: official start hackathon!
- Today  21.30: pizza’s
- Tomorrow  09.00: start breakfast @digityser
- Tomorrow  14.30: start lunch
- Tomorrow  20.30: start dinner
- Sunday  09.00: start brunch
- Sunday 12.30: end of hackathon (deadline for submissions)
- Sunday 13.00: start of award ceremony with guest: Ozark Henry!
- Sunday 15.00: end of ceremony & reception

Tweetweet tweet

#MusicHack

@DSGhent
@ML6team
@DigitYser

Practicalities

- Every participant has to sign the general terms and conditions sheet @ reception
- Please respect the premises & other people's stuff
- Smoking is permitted on the roof terrace
- Chill zone in the basement
- Wifi: DIGITYSER-2.4 or 5, password: Easy24Get
- Our phone numbers:

Matthias

+32 498/11.83.09

Sam

+32 485/58.54.52

Xander

+32 473/86.74.89

Hendrik

+32 484/29.50.09

Food

- Food: Register on Eventbrite and pay €30 for all-in food package during the weekend, or bring your own food

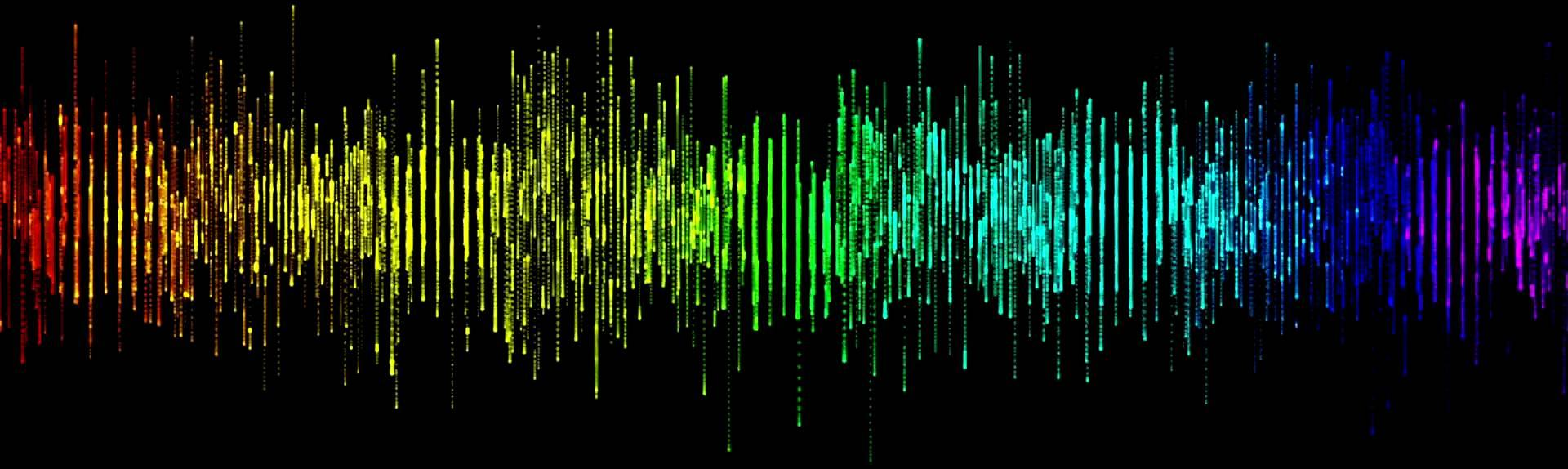
- Water and coffee are free, any other drink can be taken from the HONESTY bar (pay when you drink)

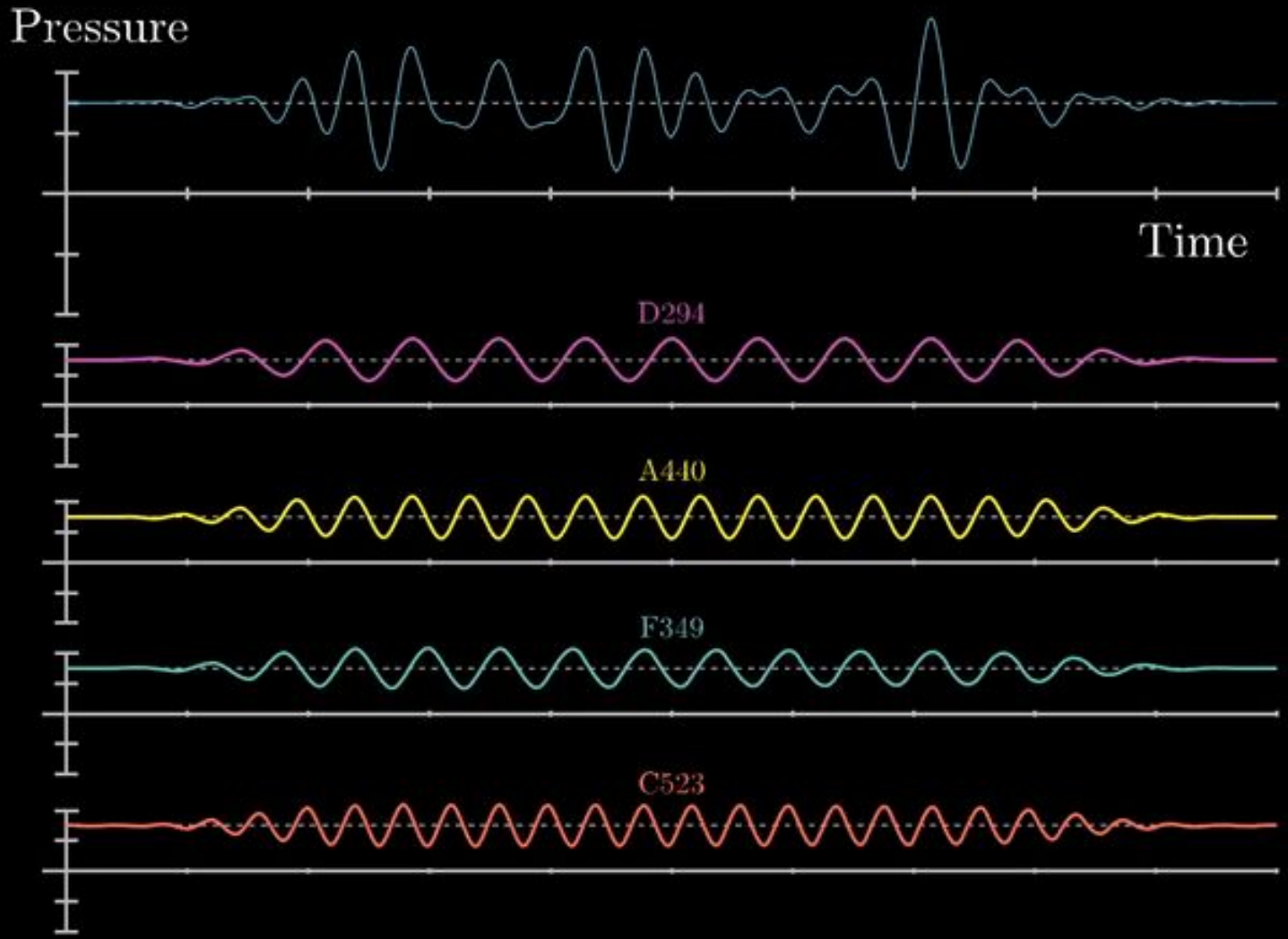
Hackathon Flow

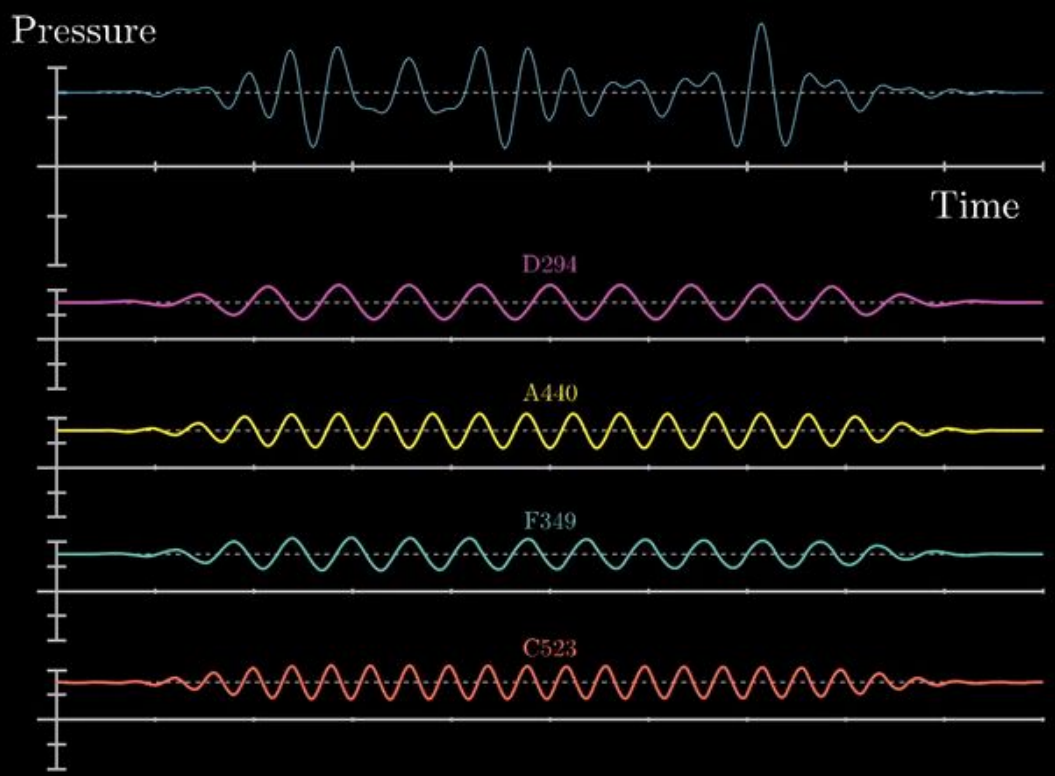
1. Form team and register on: link to spreadsheet
2. Hack your way towards an awesome, NN-generated MP3 file!
3. Send us your solution (before Sunday 12.30)
    - MP3 (max 3) upload to: google form
    - Add your slide to slidedeck: final presentation
        - Up to 3 solutions allowed per team: ~5 slides per solution


Music is complicated

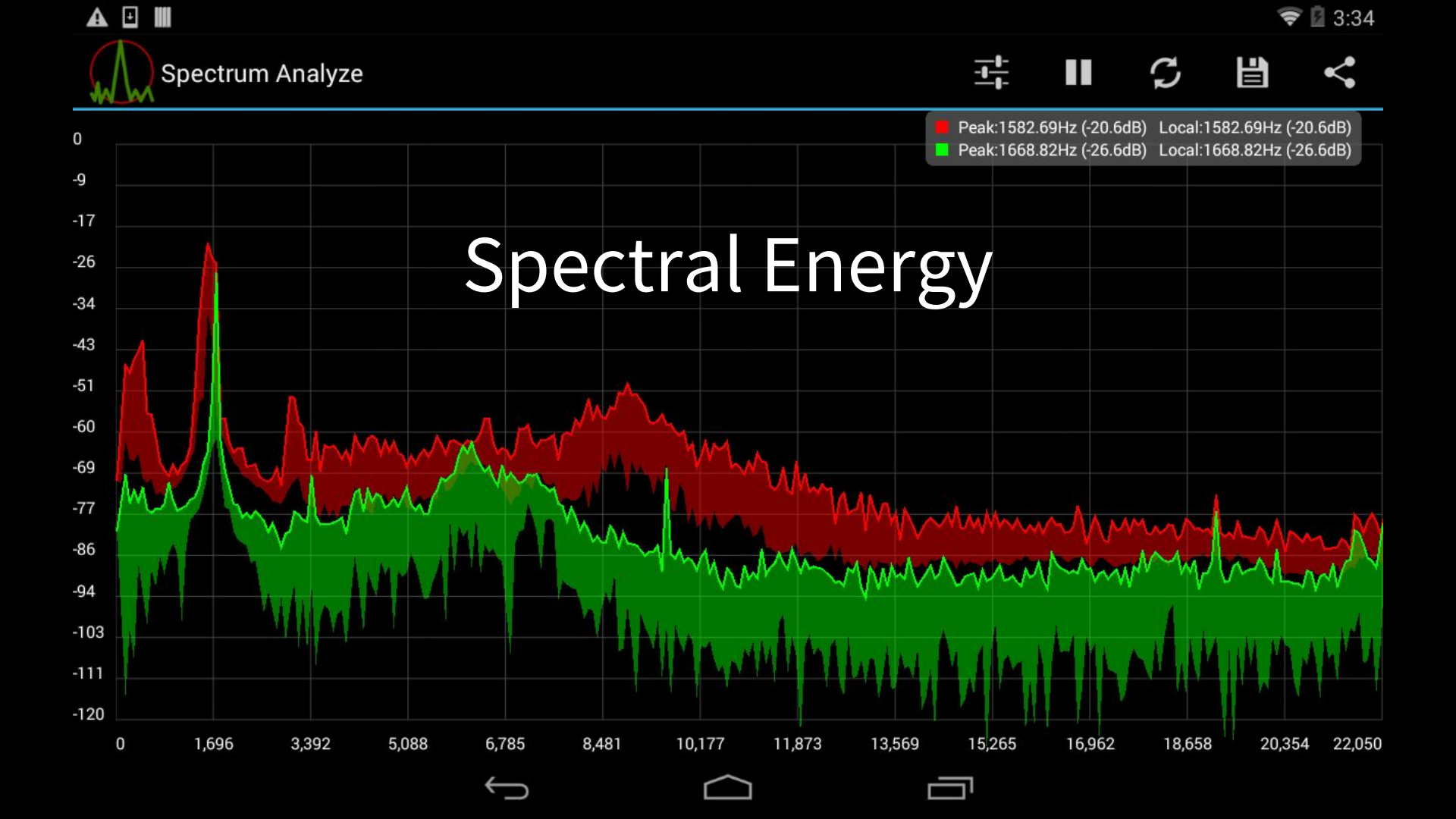
Raw waveform

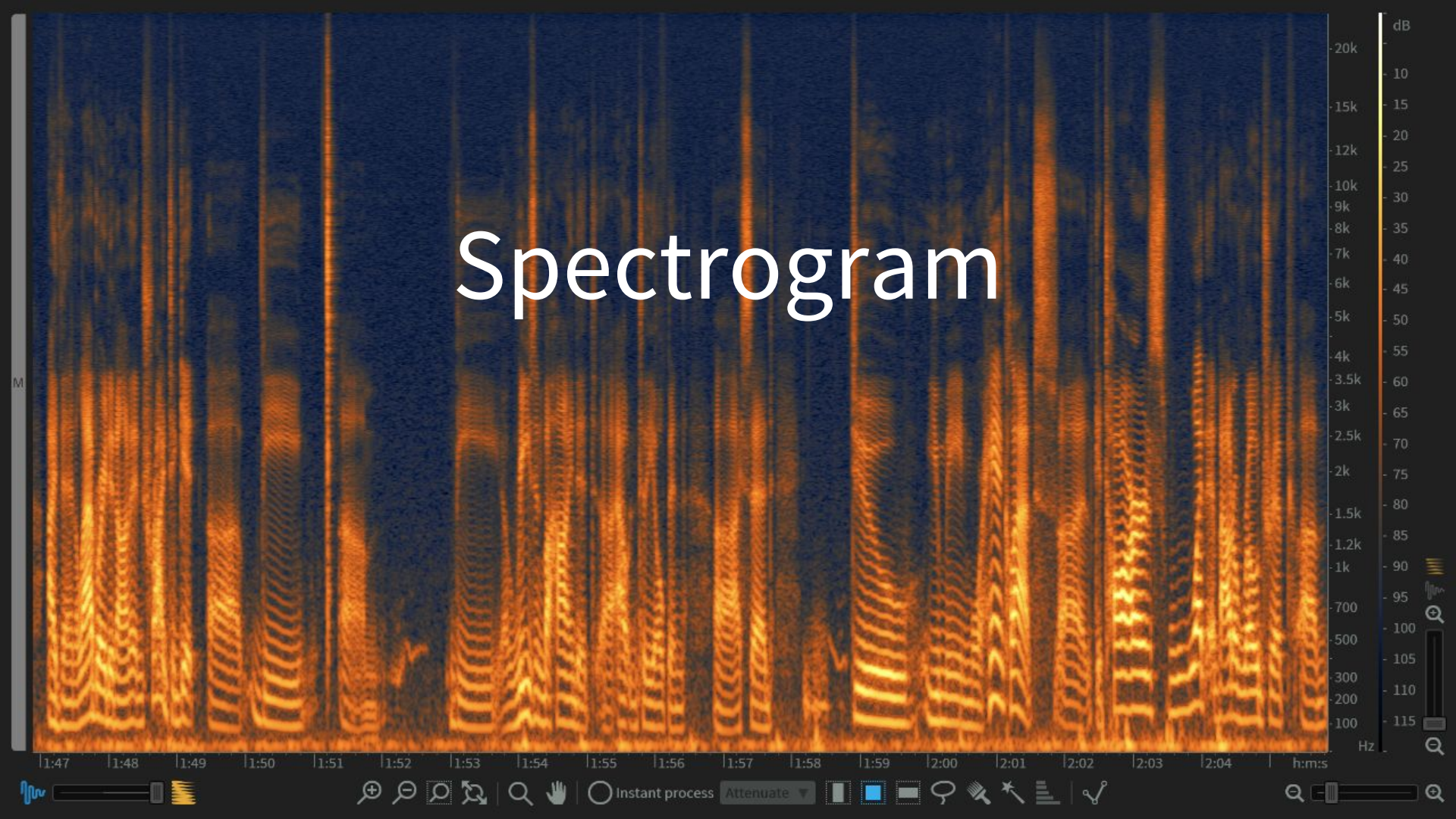



Pressure
Time
D294
A440
F349
C523



$$f(t) = A_0 + \sum_{n=1}^{\infty} (A_n cos(nt) + B_n sin(nt))$$

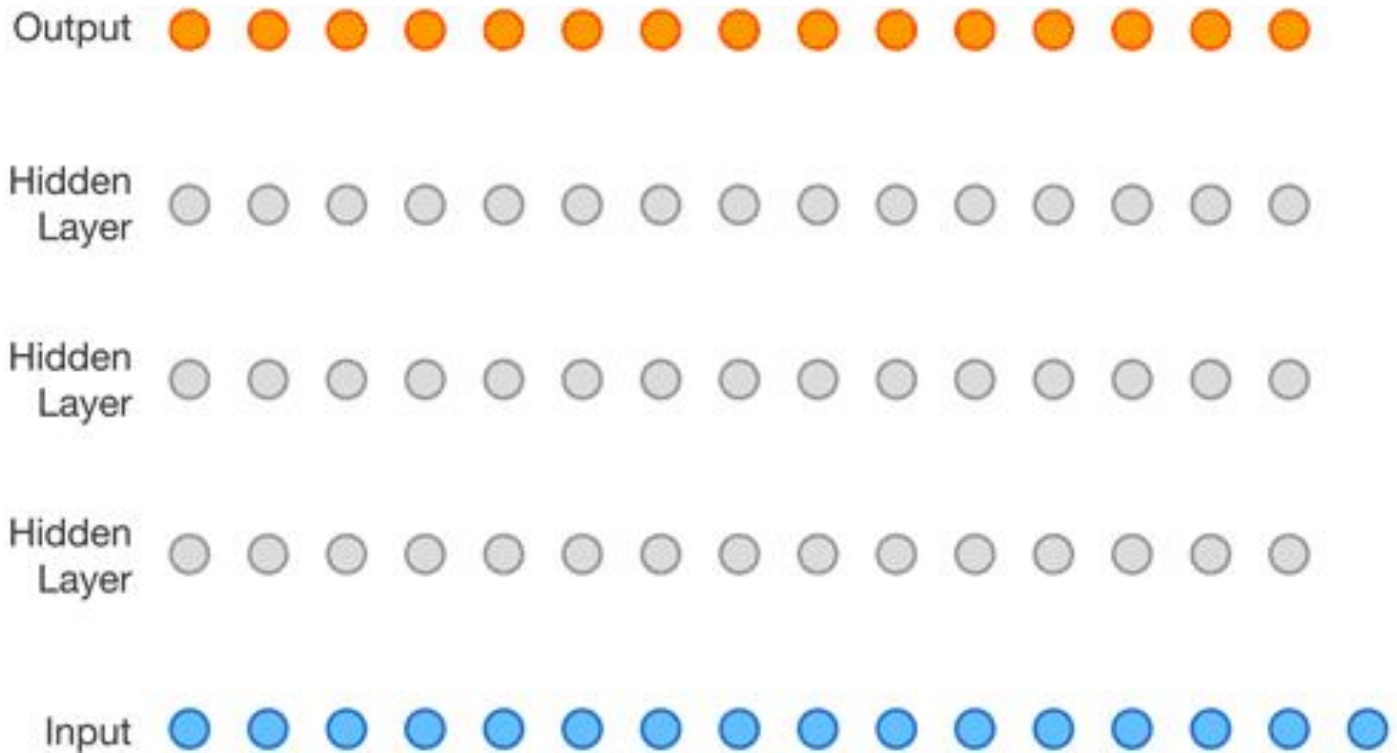

3:34
Spectrum Analyze
Peak:1582.69Hz (-20.6dB) Local:1582.69Hz (-20.6dB)
Peak:1668.82Hz (-26.6dB) Local:1668.82Hz (-26.6dB)
Spectral Energy
0
-9
-17
-26
-34
-43
-51
-60
-69
-77
-86
-94
-103
-111
-120
0
1,696
3,392
5,088
6,785
8,481
10,177
11,873
13,569
15,265
16,962
18,658
20,354
22,050

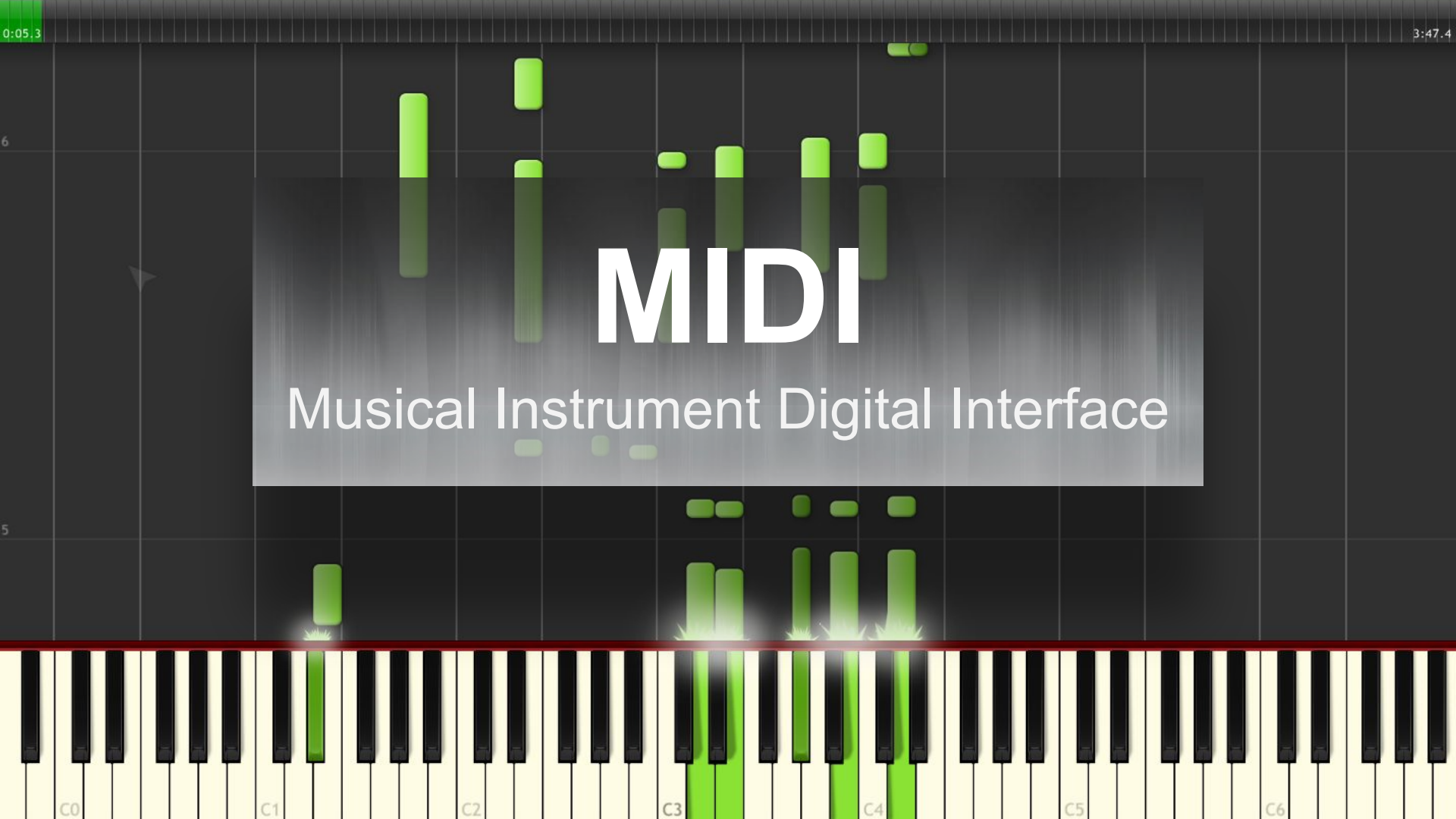

Spectrogram
M
20k
15k
12k
10k
9k
8k
7k
6k
5k
4k
3.5k
3k
2.5k
2k
1.5k
1.2k
1k
700
500
300
200
100
Hz
dB
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
1:47
1:48
1:49
1:50
1:51
1:52
1:53
1:54
1:55
1:56
1:57
1:58
1:59
2:00
2:01
2:02
2:03
2:04
h:m:s
Instant process
Attenuate

DeepMind WaveNet

1 Second

DeepMind WaveNet




0:05.3
3:47.4
MIDI
Musical Instrument Digital Interface
C0
C1
C2
C3
C4
C5
C6

Get started with MIDI

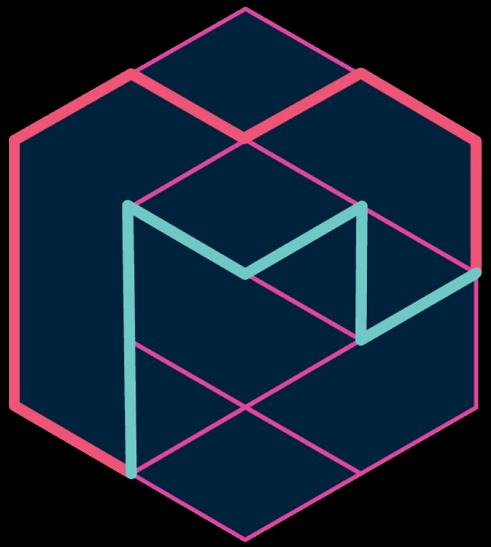
Lakh MIDI Dataset Tutorial

This IPython notebook demonstrates how to use the data in the Lakh MIDI Dataset. It shows how the dataset is organized and gives examples of how to use annotations extracted from `LMD-aligned` (the collection of MIDI files which have been matched and aligned to entries in the Million Song Dataset). We will use `pretty_midi` for parsing the MIDI files, `mir_eval` for sonification and visualization, and `librosa` for audio analysis.

In [1]:

```
# Imports
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import pretty_midi
import librosa
import mir_eval
import mir_eval.display
import tables
import IPython.display
import os
import json

# Local path constants
DATA_PATH = 'data'
RESULTS_PATH = 'results'
# Path to the file match_scores.json distributed with the LMD
SCORE_FILE = os.path.join(RESULTS_PATH, 'match_scores.json')

# Utility functions for retrieving paths
def msd_id_to_dirs(msd_id):
```

http://bit.ly/2FAHuv3



magenta

https://magenta.tensorflow.org/

Start your own Linux VM on gcp

☆ ☆ ☆ ☆ ☆

FEEDBACK VERZENDEN

Quickstart Using a Linux VM

This page explains how to create a Linux virtual machine instance in Compute Engine using the Google Cloud Platform Console.

Before you begin

1. Select or create a Cloud Platform project.

   GO TO THE MANAGE RESOURCES PAGE

2. Facturering voor uw project inschakelen.

   FACTURERING INSCHAKELEN

Create a virtual machine instance

1. In the GCP Console, go to the VM Instances page.

   GO TO THE VM INSTANCES PAGE

Install Magenta Development Version

- sudo apt-get update
- sudo apt-get install libasound2-dev libjack-dev python-pip htop
- pip install --upgrade pip
- sudo pip install matplotlib scipy bokeh IPython pandas jupyter
- git clone https://github.com/tensorflow/magenta.git

Install Bazel:

- sudo apt-get install openjdk-8-jdk
- echo "deb [arch=amd64] http://storage.googleapis.com/bazel-apt stable jdk1.8" | sudo tee /etc/apt/sources.list.d/bazel.list
  curl https://bazel.build/bazel-release.pub.gpg | sudo apt-key add -
- sudo apt-get update && sudo apt-get install bazel
- sudo apt-get upgrade bazel
- sudo pip install magenta

Generate your first MIDI-song:

Download pretrained RNN:

- cd magenta/magenta/models/melody_rnn
- mkdir trained_models
- mkdir generated_midis
- curl -o trained_models/attention_rnn.mag http://download.magenta.tensorflow.org/models/attention_rnn.mag

Generate MIDI using attention_rnn: *(replace ".." with the correct directories)*

```
python melody_rnn_generate.py \
--config='attention_rnn' \
--bundle_file= /home/../magenta/magenta/models/melody_rnn/trained_models/attention_rnn.mag \
--output_dir=generated_midis \
--num_outputs=10 \
--num_steps=128 \
--primer_melody="[60]"
```

Generate MIDI to mp3: http://mp3-tools.com/free-midi-to-mp3-converter.html

MIDI Synthesiser

Magenta has a custom MIDI synthesiser that allows you to play with various types of MIDI-to-sound scripts:

https://github.com/tensorflow/magenta/tree/master/magenta/interfaces/midi

Get + explore MIDI data: (includes very useful notebooks!)

Explore the **‘Lakh MIDI Dataset v0.1’:**

http://colinraffel.com/projects/lmd/

Download and store the MIDI dataset on the VM:

mkdir midi_data
curl -o midi_data/lmd_aligned.tar.gz http://hog.ee.columbia.edu/craffel/lmd/lmd_aligned.tar.gz
curl -o midi_data/clean_midi.tar.gz http://hog.ee.columbia.edu/craffel/lmd/clean_midi.tar.gz
cd midi_data
tar -xvzf lmd_aligned.tar.gz
tar -xvzf clean_midi.tar.gz

Download Tfrecords files: (avoid 4 hours of waiting)

clean_midi.tfrecords

sequence_data.zip

Generate TF_Records file from MIDI-data:

```
cd ../scripts/

INPUT_DIRECTORY=/home/../midi_data/clean_midi/
SEQUENCES_TFRECORD=/home/../midi_data/clean_midi.tfrecord


#Run script headless (takes a really long time!!)
nohup python convert_dir_to_note_sequences.py \
  --input_dir=$INPUT_DIRECTORY \
  --output_file=$SEQUENCES_TFRECORD \
  --recursive &> clean_midi_log.out&
```

To check that your script is running you can type “htop” in the command line

The output of the script will be written to “clean_midi_log.out”

Create Train/Validation/Test sets

Go back to root folder

```
mkdir midi_data/sequence_data

cd /../models/melody_rnn

nohup python melody_rnn_create_dataset.py \
--config='attention_rnn' \
--input=/home/../midi_data/clean_midi.tfrecord \
--output_dir=/home/../midi_data/sequence_data \
--eval_ratio=0.10 &> generate_train_data.out&
```


Are
You
Ready?

Create Train/Validation/Test sets

```
python melody_rnn_train.py \
--config=attention_rnn \
--run_dir=/tmp/melody_rnn/logdir/run1 \
--sequence_example_file=/home/../midi_data/sequence_data/
training_melodies.tfrecord \
--hparams="batch_size=64,rnn_layer_sizes=[64,64]" \
--num_training_steps=20000
```

How to register your team

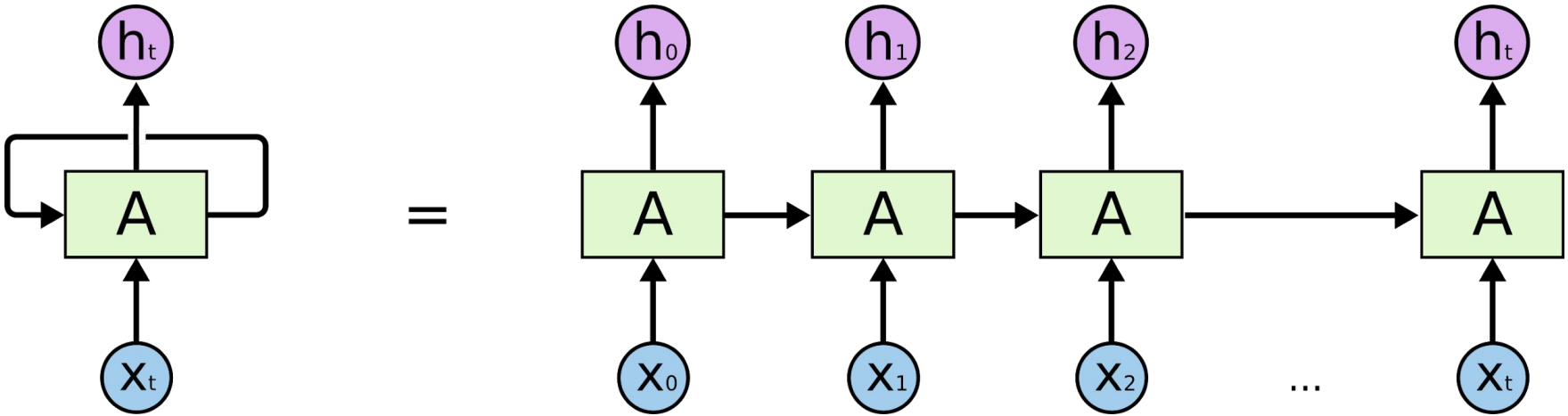
Go to

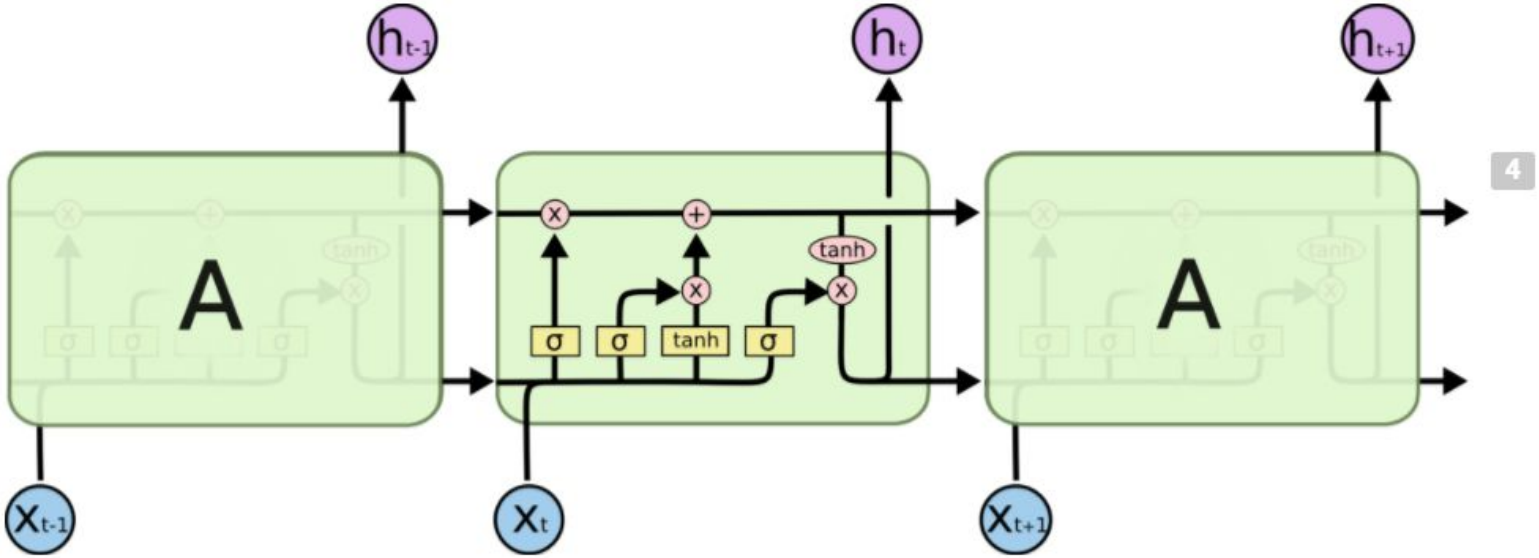
https://docs.google.com/spreadsheets/d/1Tdgm-vsX8hMVRVjfD9FLSs3kuQNwuCKITdcA8pbHDt4/edit#gid=0

and register your team and its members

Attention RNN:

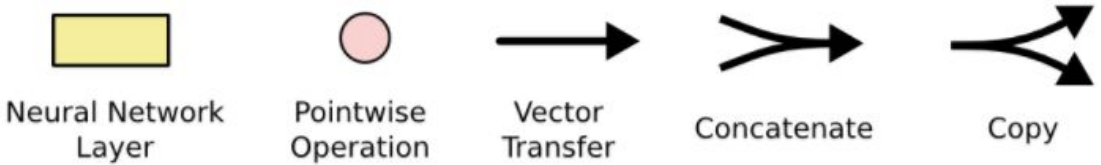
```
DEFAULT_MIN_NOTE = 48
DEFAULT_MAX_NOTE = 84
DEFAULT_TRANSPOSE_TO_KEY = 0

'attention_rnn': MelodyRnnConfig(
        magenta.protobuf.generator_pb2.GeneratorDetails(
            id='attention_rnn',
            description='Melody RNN with lookback encoding and attention.'),
        magenta.music.KeyMelodyEncoderDecoder(
            min_note=DEFAULT_MIN_NOTE,
            max_note=DEFAULT_MAX_NOTE),
        tf.contrib.training.HParams(
            batch_size=128,
            rnn_layer_sizes=[128, 128],
            dropout_keep_prob=0.5,
            attn_length=40,
            clip_norm=3,
            learning_rate=0.001))
```


h_t
A
x_t
=
h_0
h_1
h_2
h_t
A
A
A
A
x_0
x_1
x_2
...
x_t



The repeating module in an LSTM contains four interacting layers.

Don't worry about the details of what's going on. We'll walk through the LSTM diagram step by step later. For now, let's just try to get comfortable with the notation we'll be using.



Interesting Links:

- Music Generation Challenge on CrowdAI
- More info on the MIDI file format
- Other GitHub Repo's for music generation
- How LSTM's work (Colah's Blog)
- 2 Episodes from Siraj' on music generation:
  - Generate Music in TensorFlow
  - How to Generate Music with Tensorflow (LIVE)

Slightly related Links *(if you need some chill-out)*

- Rising pitch sound effect in Dunkirk

Useful Linux commands:

Run Jupyter notebook @VM: (paste this in SSH terminal and copy the login token)
jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser & disown

Run Python script headless & append to log.out:
sudo nohup python run.py &> log.out&

List (& kill) all running jupyter servers:
jupyter notebook list
kill $(pgrep jupyter)

Check running python jobs:
pgrep -af python

Check disk & CPU usage:
htop
df

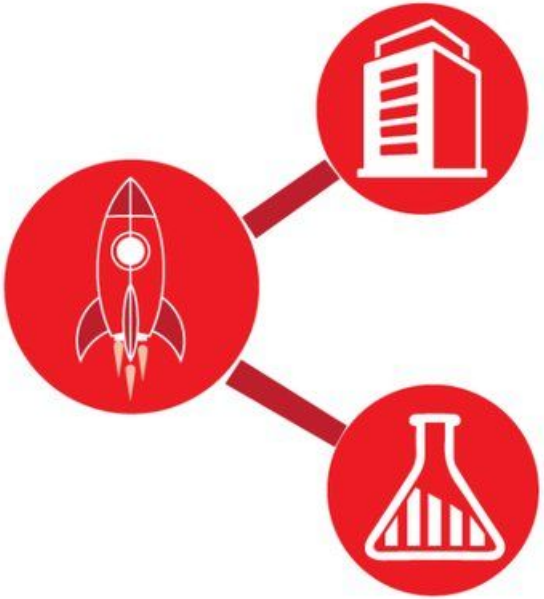
Evaluation

1. MP3 + Technical presentation
2. Public voting (1 - 2 weeks after Hackathon)

Many thanks to

Link to this slidedeck:
goo.gl/8Gt87S

ML6



DIGITYSER