Because Nobody Wants to Edit Drums: Building Trainable Audio Production Tools via Machine Learning

Scott H. Hawley, Ph.D.
Belmont University
@drscotthawley, drscotthawley.github.io

Music City Data, June 2, 2018

About the speaker

2000: Ph.D. in Physics (Numerical Relativity) from University of Texas at Austin
2000-2006: Postdoc at Albert Einstein Institute (Potsdam, Germany), & UT-Austin
2006+: Faculty, Department of Chemistry & Physics, Belmont University

I’m a computational physicist, who used to simulate black holes.

Got tired of research & wanted to make music (but keep a good day job!), so in 2006 got a job teaching in Nashville at Belmont. (Undergrad-only in science)

My students are almost all Audio Engineering Technology (AET) majors, so over time I switched fields, from Numerical Relativity to Musical Acoustics.

Started getting into Machine Learning research* in 2014/2015, after attending the Audio Engineering Society conference.

*And it’s ruining my ‘music career’!

Links & code: drscotthawley.github.io

Meaning of Title

Trainable: Ideally, those than can be trained **by the end user**

- As opposed to pre-trained & ‘deployed’ (e.g. LANDR, Izotope Neutron, ...many others)
- Not always a clear distinction: does “only users with GPUs” count?
  - Musicians with GPUs exist but are a small minority. (Should change w/ time)
  - For now, we’ll say, “trainable by somewhat dedicated individuals” ;-)
- Often the amount of features is the bottleneck for ‘trainable’
  - e.g. Raw audio vs. MIDI (or OSC), End-to-end vs. preprocessed feature extraction

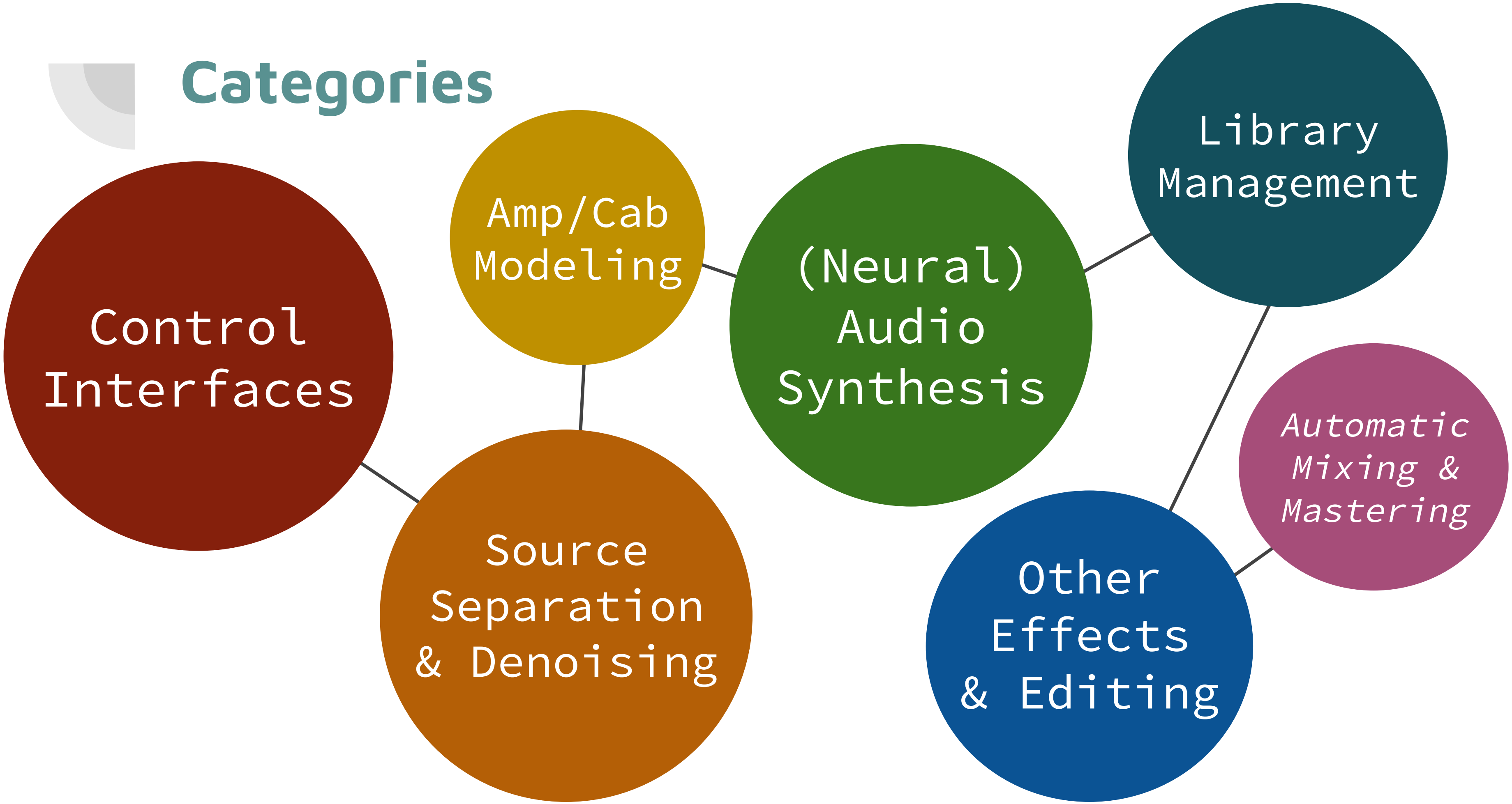
(Musical) Audio:

- Not speech-to-text or vice versa (nor Machine Learning for tracking bands’ fans!)
- My interest is raw audio, but will cover some MIDI/OSC

Production: As in, ***assisting*** the workflow of pro engineers & producers. Will cover some sound generation/synthesis as well, but not algorithmic composition.

Tools: Ease of use, vs. ∃ some grad student’s command-line demo?

Categories


Control Interfaces
Amp/Cab Modeling
Source Separation & Denoising
(Neural) Audio Synthesis
Library Management
Automatic Mixing & Mastering
Other Effects & Editing

Motivation

Use case: How can we develop helpful signal processing tools to empower people to be creative in the (pro) music arena?

Not looking to put anybody out of work (that they want to do)

Scientific: What sorts of models are best for rapidly learning representations of musical production data? What sort of function spaces, numerical issues, etc. are involved?

Educational: This is neat (albeit hard, and uncommon) set of test problems for learning about machine learning, AI, neural networks, etc..

Control Interfaces

Trainable Control Interfaces

General: Wekinator

Voice: Vochlea; Selection for Editing

Drums: Sensory Percussion

Gloves: Mi.Mu

Wekinator, www.wekinator.org

Developed by Rebecca Fiebrink (formerly at Princeton, now at Goldsmiths in London)

"A Meta-Instrument for Interactive, On-the-Fly Machine Learning."
R Fiebrink, D Trueman, PR Cook - NIME, 2009

Wekinator is a trainable ML ‘bridge’ to connect various input devices to output devices by means of OSC codes (similar to MIDI)

Built on Weka ML toolkit, Wekinator allows real-time, interactive training.

Plug: 👍👍 Rebecca’s Free Online Course on **Kadenze**: “Machine Learning for Musicians and Artists” <---- Write this down!

Wekinator overview

Image source: wekinator.org

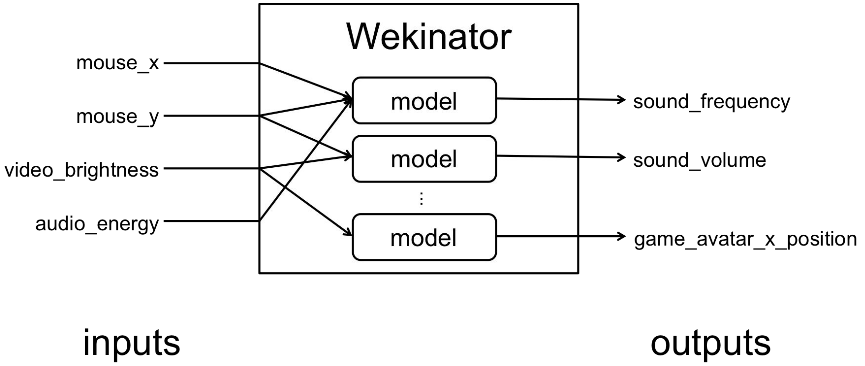
Uses supervised learning approach to map input features, e.g.,

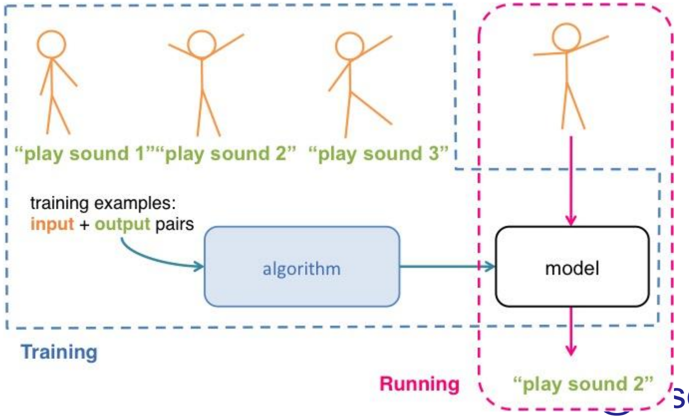
- Mouse position
- Game controller buttons
- Hand position (video)
- Facial features (pre-proc’d video)
- Microphone audio

…to outputs, e.g.,

- Pitch
- Volume
- Sample selection
- Loop start/stop

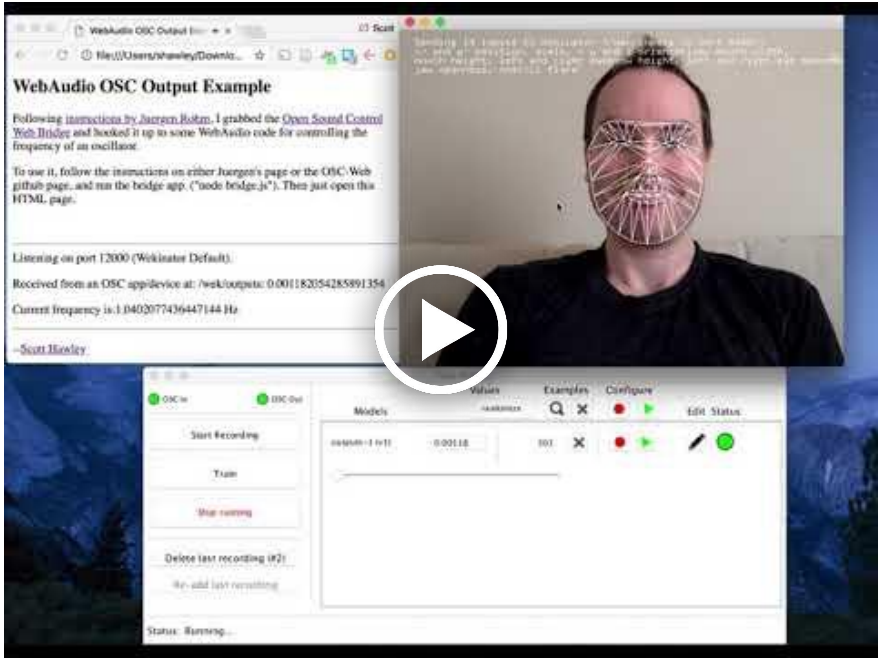
Various models & methods available for classification & regression





Wekinator, live demo?

Mayyybe. Video link as backup: Mouth opening controls pitch of WebAudio app.
This & lots more sample code & projects at www.wekinator.org/examples



Voice: Vochlea

Voice: Selection for Editing

Variety of demos on audio-AI topics by Paris Smaragdis’ group: http://paris.cs.illinois.edu/demos

“In this demo we present an audio-driven interface which allows a user to vocalize the sound they want to select and an automatic process matches that input to the most appropriate sound.”

(How ‘trainable’ is this?)

Related: “Joint Optimization of Masks and Deep Recurrent Neural Networks for Monaural Source Separation,” Huang, P.-S., M. Kim, M. Hasegawa-Johnson, P. Smaragdis, in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol.23, no.12, pp.2136-2147, Dec. 2015. https://arxiv.org/abs/1502.04149

Drums: Sensory Percussion by Sunhouse

Gloves: Mi.Mu

Brown, D., Nash, C. and Mitchell, T. (2018) "Understanding user-defined mapping design in mid-air musical performance." In: *Proceedings of the 5th International Conference on Movement Computing (MOCO 2018)*, Genoa, Italy, 28 - 30 June 2018. Available from: http://eprints.uwe.ac.uk/36127

Source Separation & Denoising

Source Separation & Denoising

Source Separation is a ***huge*** field and we will only touch on a few items that I'm most familiar with. This means we're leaving out ***many*** significant results. *Apologies all around.*

DrumAtom

Products: (company-trainable rather than user-trainable, still worthy of mention)

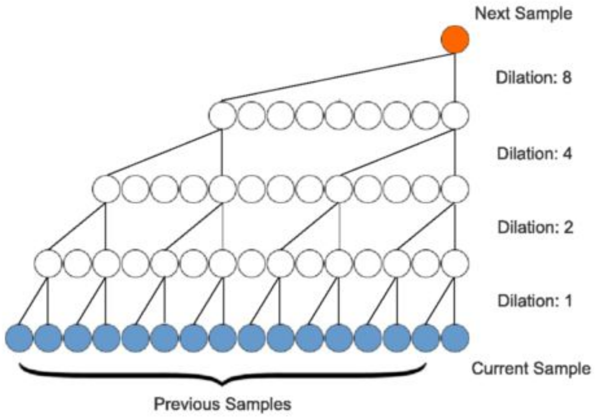
- DrumAtom by Accusonus. Uses NMF*.
- XTRAX by Audionamix ("One song in. Three stems out.")

*Notable talk: "NMF? Neural Nets? It's all the same..." by P. Smaragdis, SANE 2015 workshop, https://www.youtube.com/watch?v=wfmpViJljWw

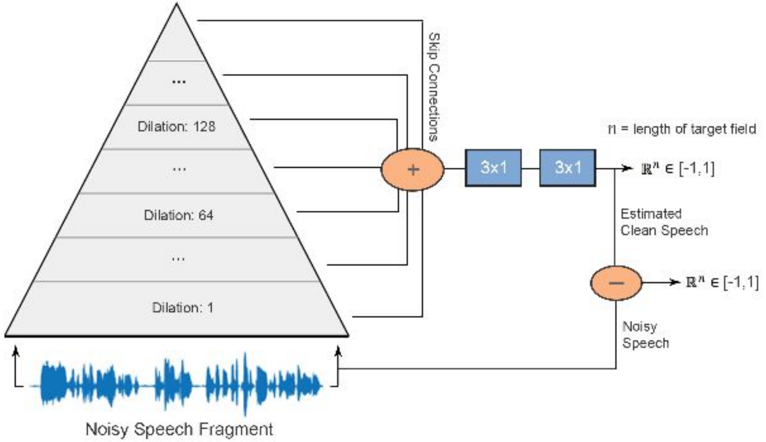
@drscotthawley

Recent research:

- "An Overview of Lead and Accompaniment Separation in Music," Z. Rafii, A. Liutkus, F.-S. Stöter, S.I. Mimilakis, D. Fitzgerald, B. Pardo IEEE/ACM Transactions on Audio, Speech and Language Processing, 2018.
- "A Wavenet for Speech Denoising," by D. Rethage, J. Pons, X. Serra, in proceedings of the 43rd IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP2018), April 2018.
  Code: https://github.com/drethage/speech-denoising-wavenet
  Examples: http://jordipons.me/apps/speech-denoising-wavenet
  - Built on WaveNet (van den Oord et al., Google), a stacked set of dilated ('atrous') 1D convolutional layers with skip connections; orig. used for speech synthesis



Source: WaveNet paper



Source: Rethage et al.

@urscotthawley

My own experiments:

A “vanilla” LSTM encoder-decoder model can do a good job at ‘simple’ denoising. Trained network using synthetic data of signal+noise in, clean signal out.   Got reduction of 10-12dB.

Q: “Yea, but what about more general noisy audio?”
A: *Exactly*. We used a collection that included other simple non-sine signals, but still didn’t try general audio because of limitations that became evident in the model re. performing more general audio effects (later in talk).

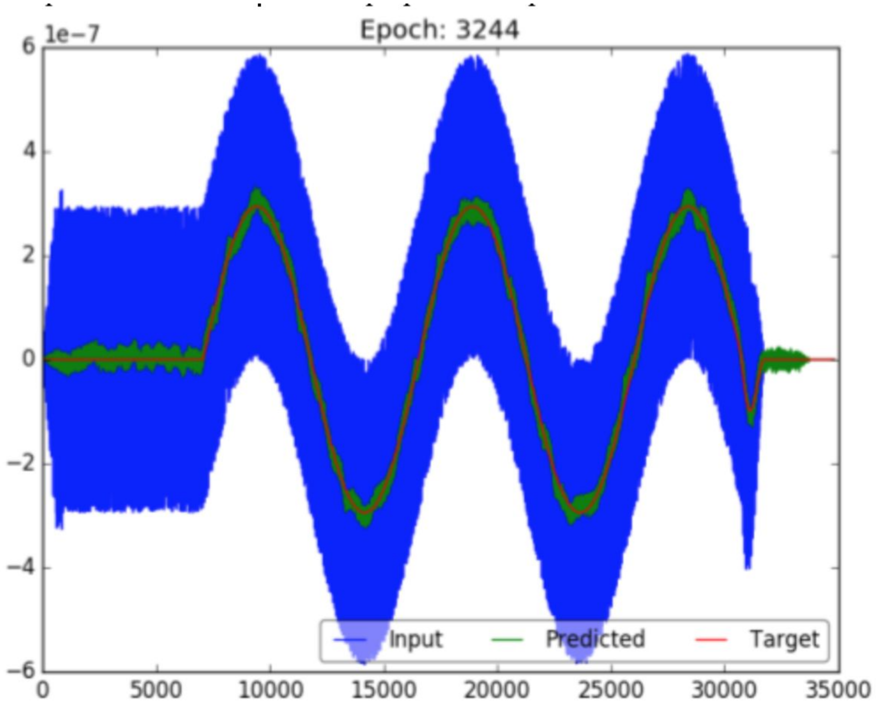


Figure 1: De-Noising. Example waveform for a de-noising filter, learned by Hawley’s neural network by training over 3000+ “epochs” of arbitrary input signals with noise added, in which the network was presented with the original pre-noise signal as the “target” to produce. A reduction of 10-12dB is achieved.   *The key point: This was achieved without knowing how to write a ‘proper’ denoising filter!*

- S.H. Hawley & J. Dowse, Les Paul Music Innovation Award Finalist Submission, Dec 2016, w/ S.I. Mimilakis

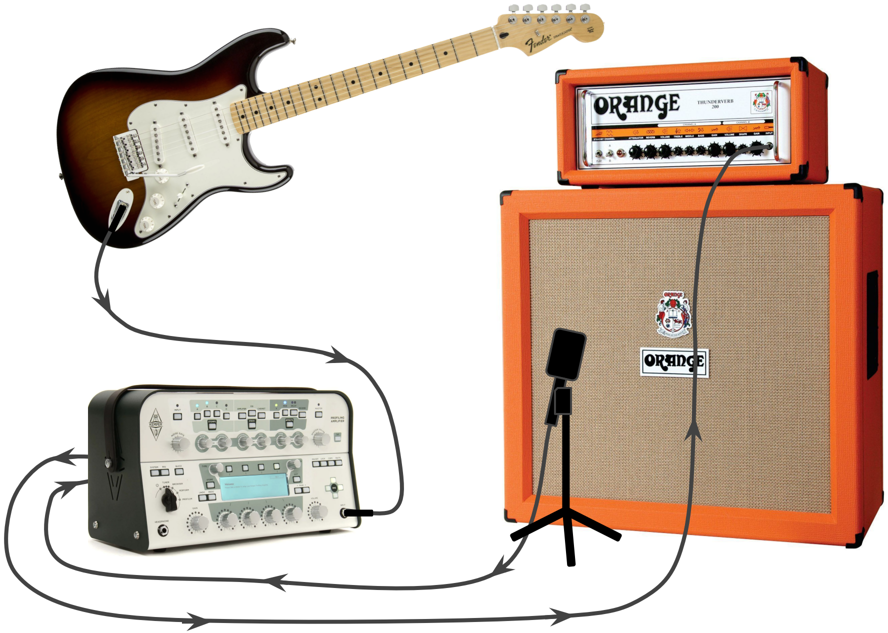
@drscotthawley

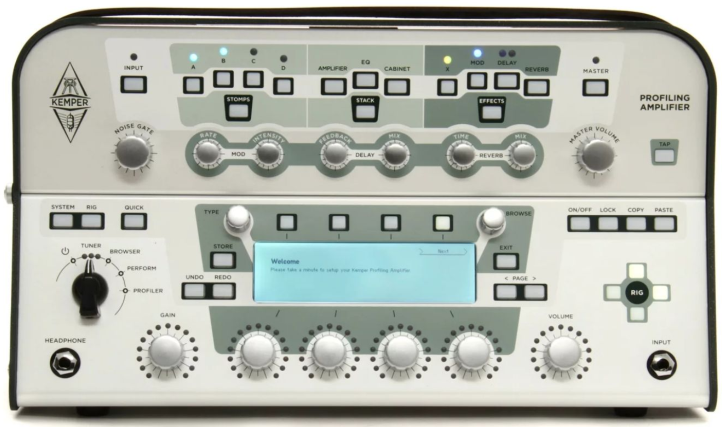
Amplifier & Cabinet Modeling

Kemper Profiler Amplifier

Guitar amp & cabinet modeling. “Profiles” the sound of an existing amp & cabinet by training approx. **5 minutes**, comparing clean in to mic’d out.

Allows guitarist/producers to ‘collect’ many amps & cabinets into one portable unit.





Algorithm: Proprietary
Runs several test tones, grabs EQ curve & impulse response. Further trains as you play. (Knobs work!)

Seems to have a finite number of tunable pre-fab modules **specifically for amp & cabinet modeling** among which it selects, & adjusts parameters.

Analysis paper: “A Discussion in Machine Learning: The Kemper Profiler,” by Lucas Novick, Jan 2017
https://github.com/aspirecoop/papers/blob/master/LDNovickPHY3990_KemperProfiler.pdf

@drscotthawley

Axe-FX (Fractal Audio Systems)

Similar to Kemper, but rack mount & more effects.

Matches tone by adjusting parameters of a variety of pre-fab 'blocks' which model amps, cabinets, effects. Includes IR capture.

@drscotthawley

(Neural) Audio Synthesis

WaveGAN & SpecGAN, @chrisdonahuey

“Synthesizing Audio with Generative Adversarial Networks,” C. Donahue, J. McAuley, M. Puckette, ICLR 2018 accepted paper, https://openreview.net/forum?id=r1RwYIJPM

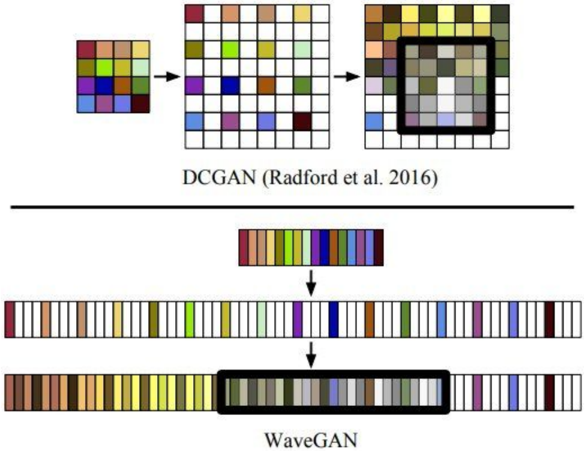
Code: https://github.com/chrisdonahue/wavegan

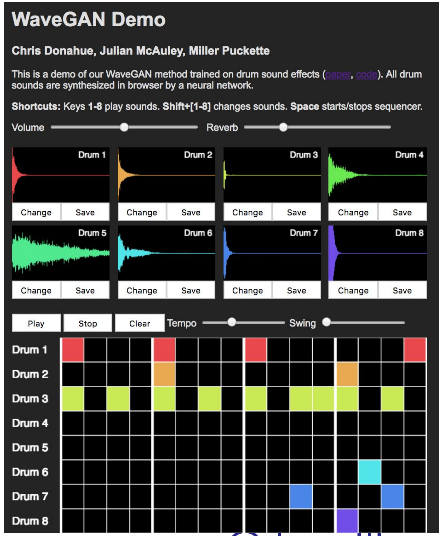
Jupyter notebook example: https://colab.research.google.com/drive/1e9o2NB2GDDjadptGr3rwQwTcw-IrFOnm

Used time-domain (WaveGAN) or spectral domain (SpecGAN) approach for generating sounds of speech, piano, birds, & drums.

Sound examples: http://wavegan-v1.s3-website-us-east-1.amazonaws.com

Demo (drum machine): https://chrisdonahue.github.io/wavegan



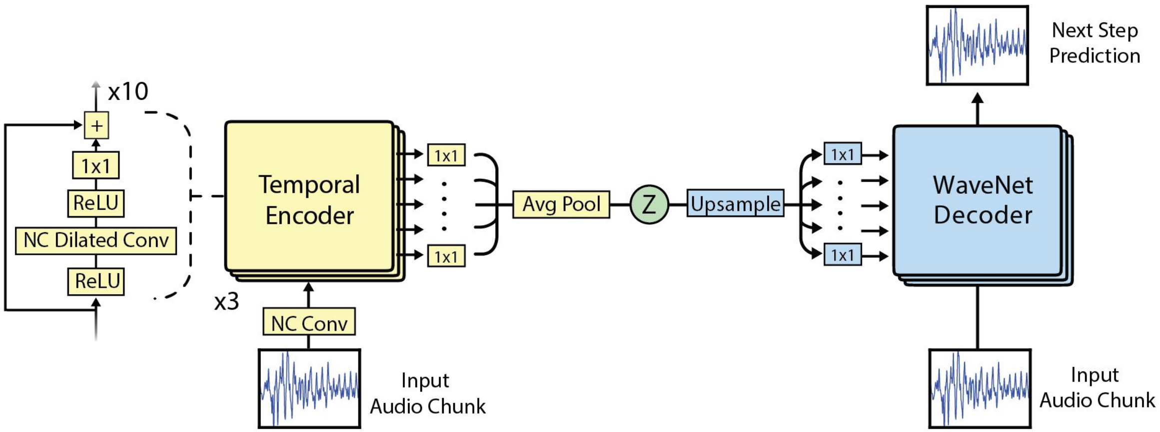


NSynth (Magenta: Google Brain & DeepMind)

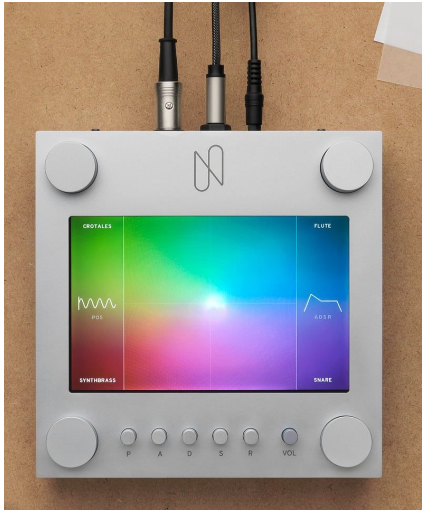
https://magenta.tensorflow.org/nsynth

Neural synthesis of a variety of sounds:

"NSynth uses deep neural networks to generate sounds at the level of individual samples. Learning directly from data, NSynth provides artists with intuitive control over timbre and dynamics and the ability to explore new sounds that would be difficult or impossible to produce with a hand-tuned synthesizer."



"NSynth Super" Device:



Free plans to build your own device: https://github.com/googlecreativelab/open-nsynth-super

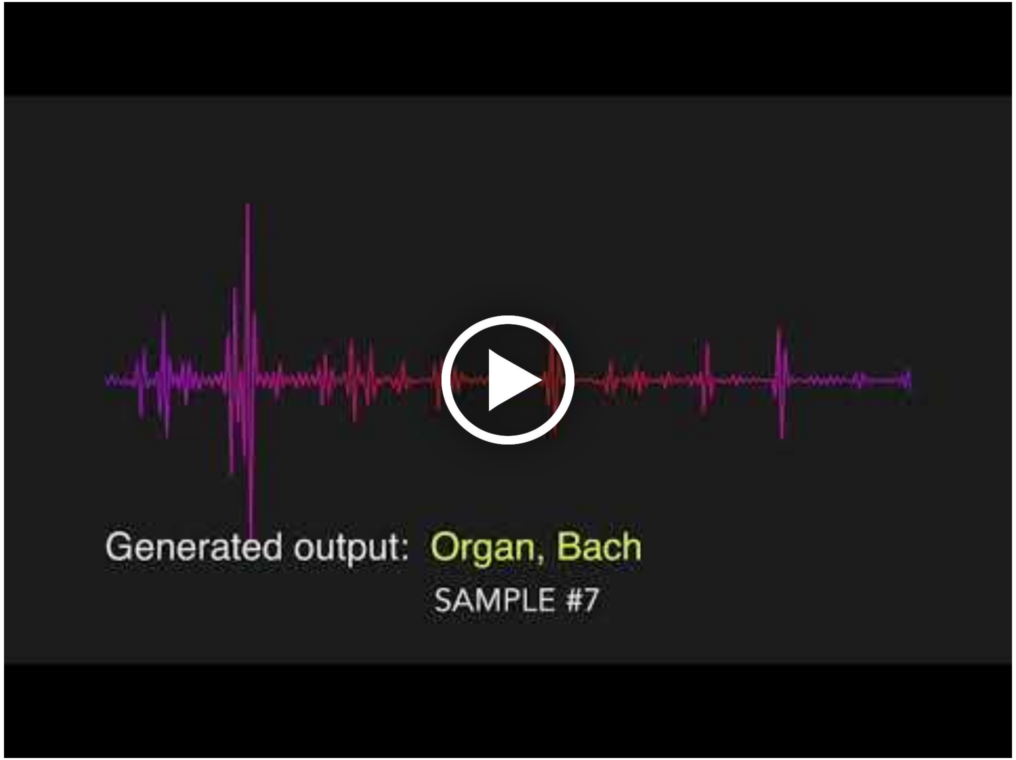
@drscotthawley

“A Universal Music Translation Network”, Mor et al (Facebook AI), May 21 2018, https://arxiv.org/abs/1805.07848

Automatic Sample Re-Synthesis

“Sample replacement” is a common technique used by producers, such as replacing ‘real’ drums (usually toms) with drum *samples*.

This **new** result ***re-synthesizes*** audio and changes one (group of) instrument(s) into another.



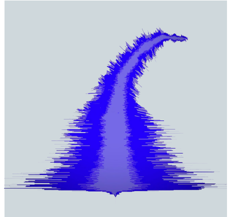
@drscotthawley

Library Management

Of Loops & Samples

S.H. Hawley & (your name here!)

Sorting H.A.T. (Hosted Audio Tagger)



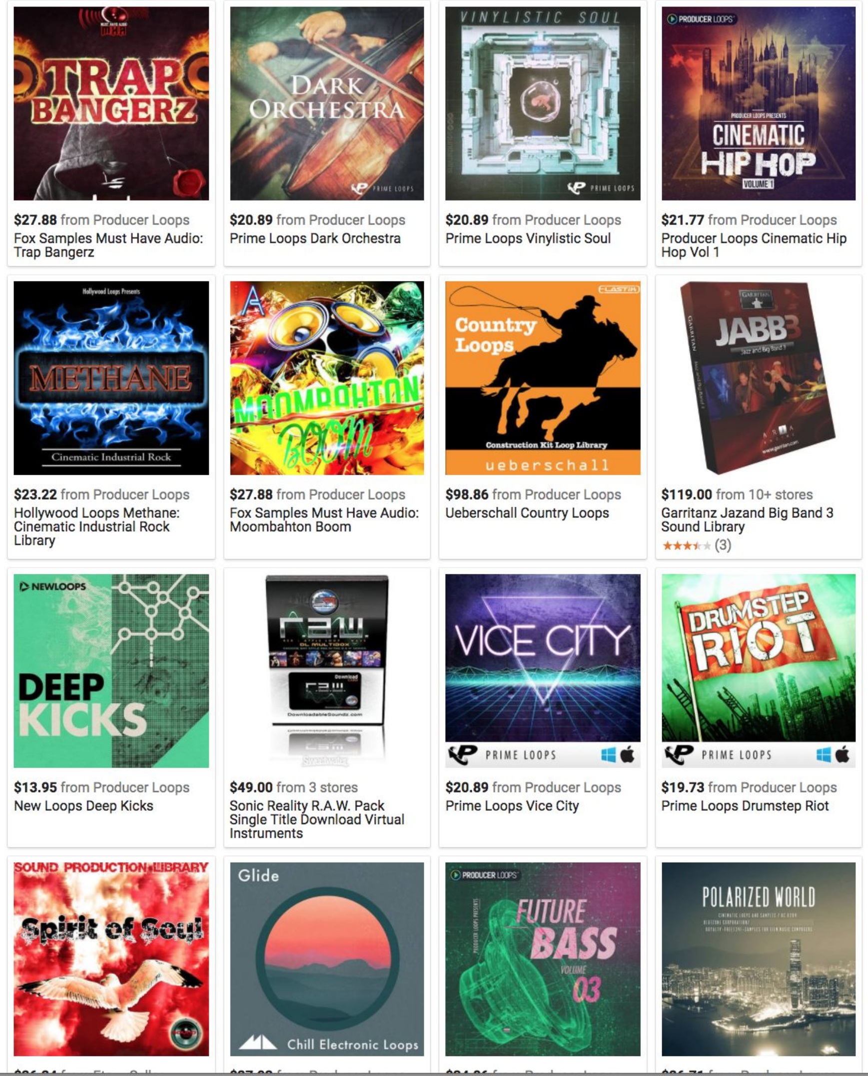
Origin: @aspirecoop

"A loose collective of scientists, engineers, artists and developers who collaborate on bringing their innovative audio ideas to life!"

Ethan Henley (producer): "I'd love to have a way to re-index my library of audio loops and samples, using my own custom tags."

Hawley: Well, I wrote a simple NN audio classifier*...

*"Panotti: A Convolutional Neural Network classifier for multichannel audio waveforms" http://github.com/drscotthawley/panotti
Built using Keras, with the Tensorflow backend.



Background:

Producers & composers maintain huge libraries of audio samples and loops.

Sometimes these audio files come with metadata ‘tags’ (genre, feel, instrument,...) sometimes not.

Finding the loop or sample you want can be a challenge.

Idea

:

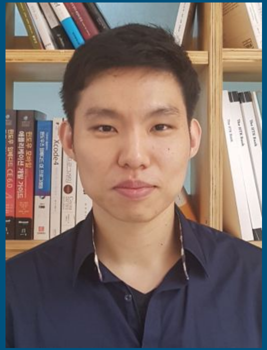
Instead of relying on “pre-made” tags supplied by the manufacturer, give users the power to tag according to their own preferred keywords.

Audio Classification

- Lots of history of using various kinds of feature extractors (tempo, pitch,...) for audio classification, e.g. for recommendation systems (Pandora, ...)
- Recent advances in Machine Learning have yielded a variety of methods which *learn* features *from the data itself*

...as Image Classification

- One common approach is to take a **spectrogram** of the audio, and use image classification methods
- Image classification is a mature & active field

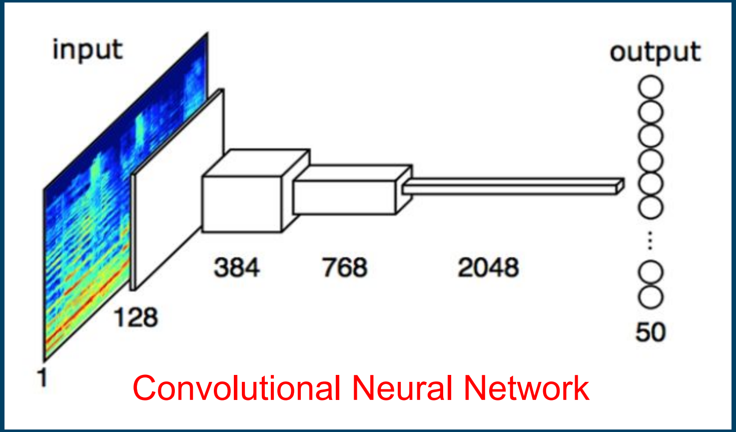


I've been following* the work of

KEUNWOO CHOI

a Ph.D. student specializing in Music Information Retrieval (MIR) at the Center for Digital Music (C4DM) of Queen Mary University of London.

*and ripping off



kid's got skillz

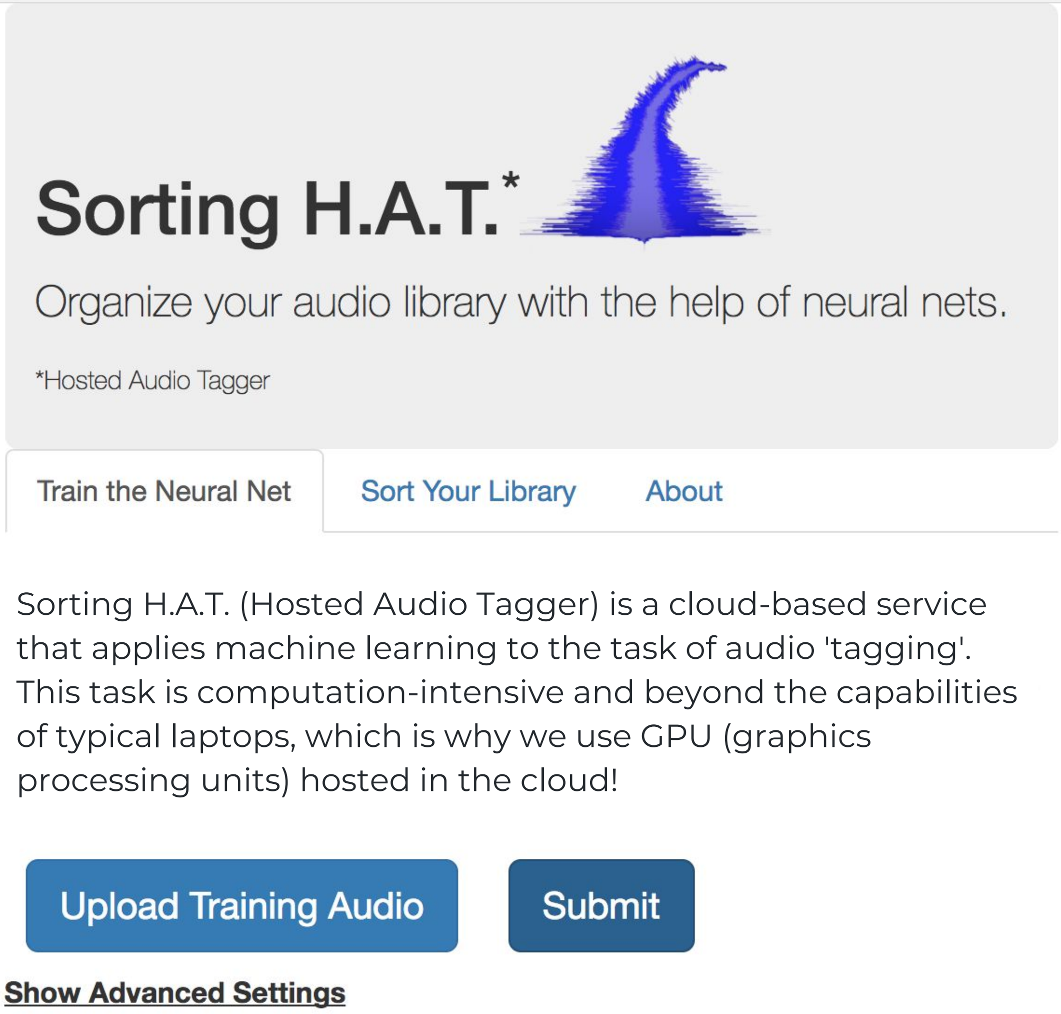
Making it ‘practical’

- Training a classifier requires a large, well-labelled dataset, and powerful computers -- with Graphics Processing Units (GPUs) -- to crunch the numbers.
- Once trained, classifiers can be deployed to do inference in apps such as Izotope Neutron.
- But no consumer-level apps existed which would allow Ethan & others to do what they might want -- i.e. *train.*

Hence...

@HackMT

HackMT 2018 Hackathon Team 15:
Scott Hawley, Braden Carei, Daniel Ellis, Will Haase, Braiden King, Tyler Thomas.



Built it on Flask

Flask a framework that allows you to write a web-based application (server) using Python.

(All our other code was in Python, so...)

It's simple to use, and similar to writing a GUI in that it's all event-driven and (web-)callbacks, specified by "@" decorations.

Only one problem for our app...

Source: http://flask.pocoo.org

Difficulty:

Filesystem Access

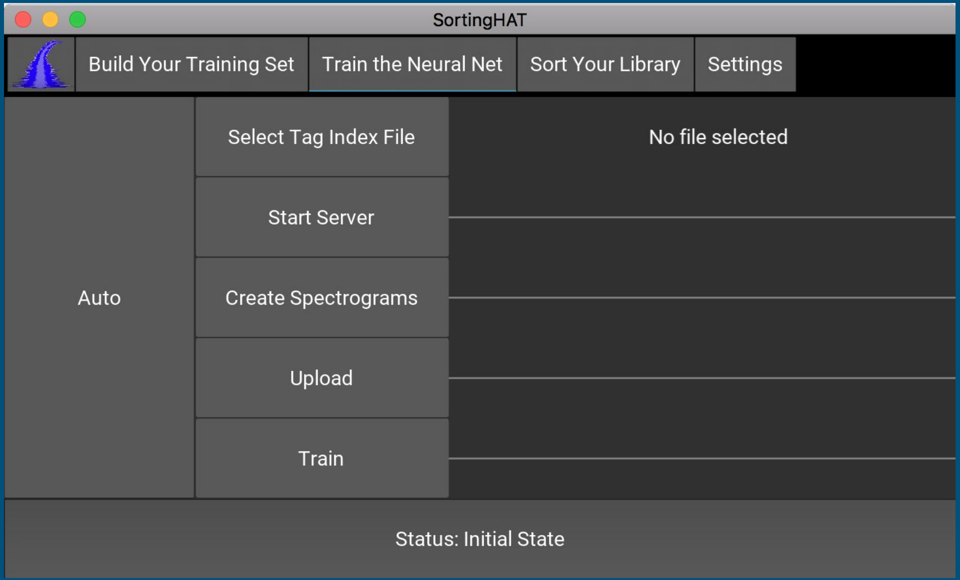
The sorting program is ultimately supposed to move (or create links to) files around on your local computer.

But we built the first Sorting HAT to be a browser app -- which for security has *no* access to your local file system.

So, rewrite. Desktop App + GUI

Desktop App

NEW!



- GPU (cloud) server is only required for training
- So everything else can be done locally, e.g. on laptop
- Thus we upload only spectrograms:
  - *Huge* reduction in data (e.g., 10 MB for 1 GB of audio)
  - No re-distribution of proprietary audio (IP/lawyer friendly)
  - Could host a (collaborative) database of spectrograms

Federated, Encrypted Deep Learning: OpenMined

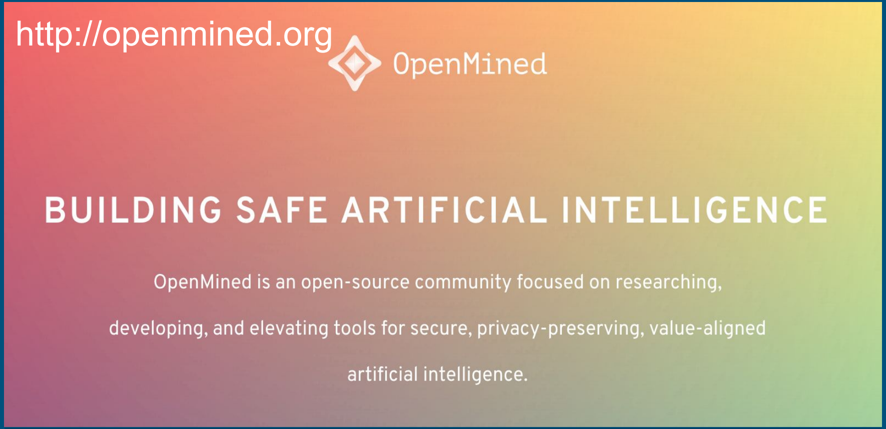
Project Lead: Andrew Trask @iamtrask
(Oxford/DeepMind, formerly at Belmont and Nashville's Digital Reasoning)

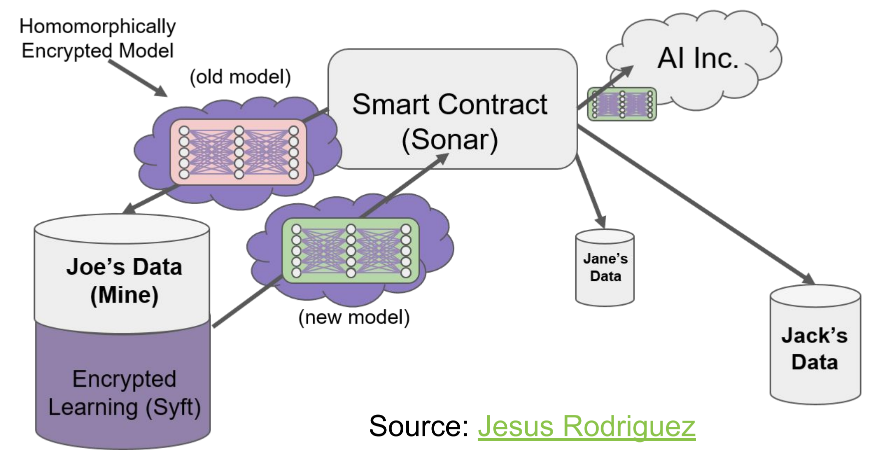
Tutorials:
https://github.com/iamtrask/Tutorials

Encrypted: Homomorphic Encryption allows for training on encrypted data

Federated: A decentralized network of devices operates on the data

Data Ownership: Computations are mined via blockchain, data owners get rewarded





Source: Jesus Rodriguez

So this means we're doing

Cloud Computing / SaaS*

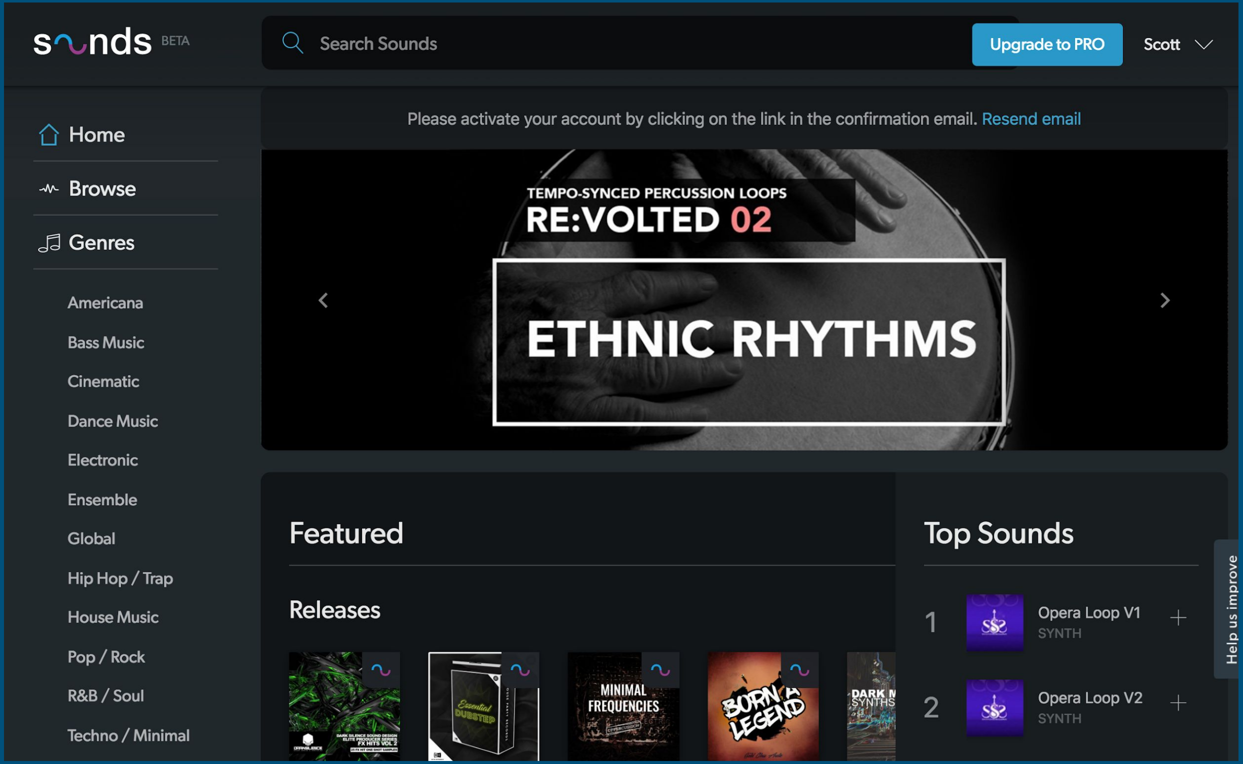
* "Software as a Service"

- Setting up Amazon EC2 or Google Cloud Compute is 'hard' for 'typical' users
- Should 'we' offer this as a service & manage accounts?
- If so, should this (Open Source) project now become a 'startup'? ($$ in <---> $$ out)
- Not looking to get rich, just get app "done" and "usable"

Aside: No, AWS Lambda won't work (no GPUs)

Also in the works: Native Instruments' sounds.com

- Uses ML to classify & organize samples & loops.
- Had hoped to interest NI & Izotope to support us. So, hmm. ?
- But Sounds.com is a "closed ecosystem", not user-trainable
- Can't help with your 'old' samples, already on your hard drive



Other Effects & Editing

S.H. Hawley & S.I. Mimilakis (& soon B. Colburn!)

Not published yet!

Learning ‘Generic’ Audio Effects (S.H. Hawley & S.I. Mimilakis)

Inspired by Kemper & Axe-FX, and by denoising efforts such as the LSTM encoder-decoder mentioned earlier, as well as works such as

- “Learning to Execute” by Zaremba & Sutskever, https://arxiv.org/abs/1410.4615
- Style transfer for image processing (review: https://tinyurl.com/y9x6ez4s)
- Variational autoencoders (VAEs) (e.g. https://distill.pub/2017/aia)

...which NN architectures best facilitate what we might call “Learning to Modulate” or “Audio Style Transfer”, i.e. to be able to ***profile*** ‘generic’ audio effects, in a (preferably) **end-to-end** fashion?

- Impulse responses can give you linear, time dependent (LTD) effects.
- My own experiments showed that an LSTM ‘Seq2Seq’-like encoder-decoder can do nonlinear time-independent effects (e.g., distortion), and some LTD effects (e.g. echo/delay),
- but what about ***nonlinear time-dependent*** effects, such as compressors?

@drscotthawley

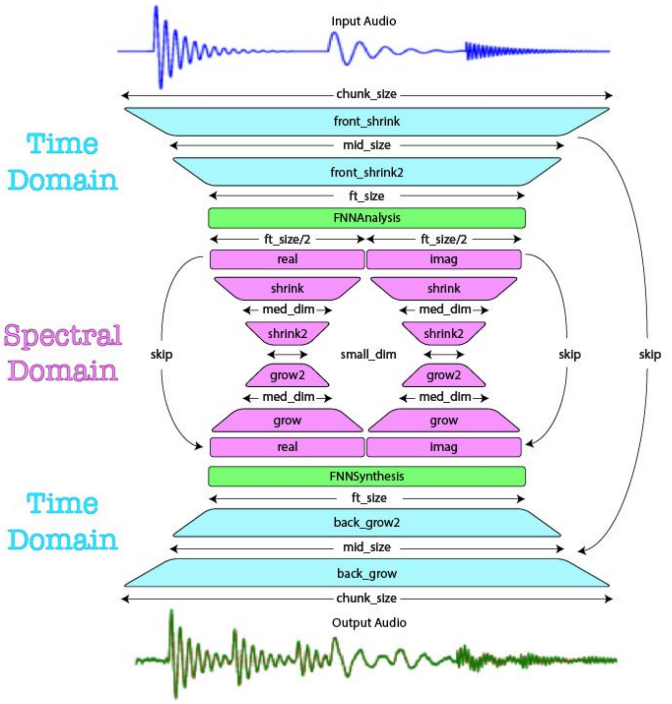
Further goals

- **'Knobs':** How to best learn the parameterized controls of the effect?
  - Experience w/ Wekinator suggests that for *N* 'knobs,' learning the 2^N 'boundary points' of parameter 'hypercube' may be sufficient.
  - Where do the 'knob' values get injected into the model? Idea: Borrow from VAEs
- **New effects:** Empower users to create their own effects, by assembling their own input->output pair datasets
- **Generality:** These 'new effects' could in principle encompass other tasks including 'editing' processes, e.g. time alignment or 'drum editing' -- and ***nobody*** wants to do drum editing! ;-)

@drscotthawley

Current Model

- (So this is supervised learning, regression, with Deep Neural Networks)
- As noted, LSTM never learned compression -- either gave zeros, or followed the input, or ‘split the difference’ between input & ‘target’
- Tried a lot of things, considered memory-enhanced overkill such as Neural Turing Machines, ...have yet to try WaveNet but it looks promising
- Current using a ***trainable*** Short-Time Fourier Transform, supplemented with pre- and post-processing in the time domain, plus a ‘bottleneck’ in the spectral domain, and some skip connections.
- Doesn’t account for ‘knobs’ yet.



Implementation Details

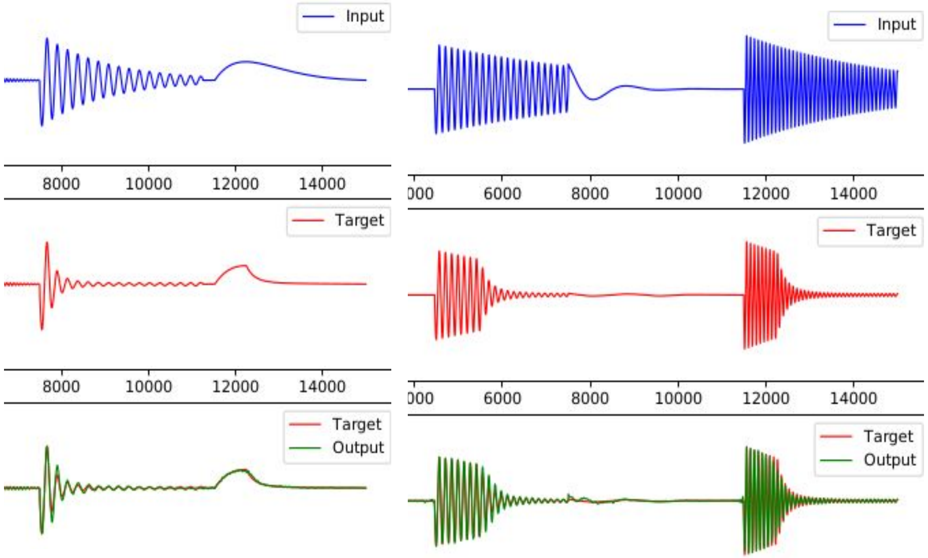
- We switched from Keras to PyTorch
  - Keras is easy to get started with (e.g. @ChaseAucoin's talk yesterday)
  - PyTorch allows (/forces) you to easily go 'deeper' into what the code is really doing.
  - Tensorflow is fine too; some of our switch was sociological
  - I still use Keras for other projects, e.g. an object-detection code*

- Supervised learning, using MSE loss.
  - Have also started a WaveNet-like model for comparison. (CE-loss)

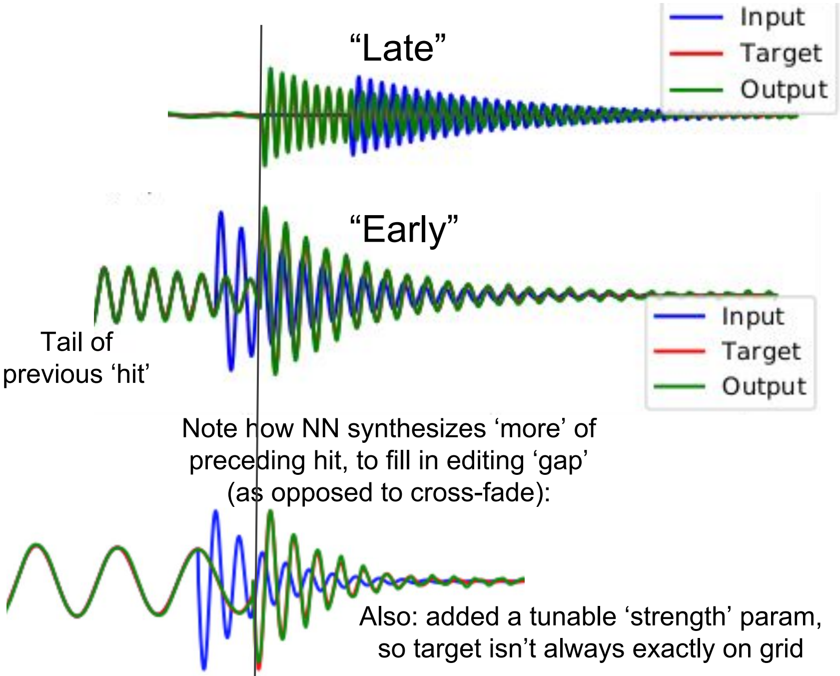
- Training on NVIDIA Titan X Pascal GPUs
  - 12 GB VRAM is key: Allows a larger 'window' of ~15000 samples
    - Also have a GTX 1080, 8 GB VRAM not as useful
  - Can run data parallel but find it doesn't offer speedup.

Early Results

Compressor:



Time Alignment:



Limitations:

- Regularity of dataset: ‘decaying sine waves’, random amp & decay rate, but same length, on a ‘grid’
- Currently the FNN model exploits this regularity; unclear if it’ll generalize to real audio; that’s next!

Automatic Mixing & Mastering

Automatic Mixing & Mastering

Not covering much in this talk because

1. "*Them's fightin' words*" in Nashville (e.g., fear of job loss.).
2. Not aware of much 'user-trainable' app model.

Majority of work done by Josh Reiss's "Intelligent Sound Engineering" group at Queen Mary University of London (QMUL) @IntelSoundEng

Commercial AI-mastering service LANDR emerged from QMUL.

See "Ten Years of Automatic Mixing," by Brecht de Mann, Josh Reiss & Ryan Stables, Proceedings of the 3rd Workshop on Intelligent Music Production, Salford, UK, 15 September 2017, http://www.brechtdeman.com/publications/pdf/WIMP3.pdf

Other recent work: "Deep Neural Networks for Dynamic Range Compression in Mastering Applications," by S. Mimilakis, K. Drossos, T. Virtanen, & G. Schuller, paper for 140th convention of the Audio Engineering Society, 2016. http://www.aes.org/e-lib/browse.cfm?elib=18237

Aside: Image methods vs. audio methods

Applying Image Methods to Audio? Observations

1. Noise. Humans are much less sensitive to visual noise than they are to audio noise.

   …would like add more studies on this, & methods for dealing with it.

2. Method: Using 2D ConvNets on audio spectrograms “shouldn’t” work very well, because of the asymmetry between time & frequency, e.g. lack of vertical (frequency) translation invariance in the dataset

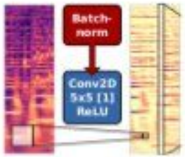
   ….and yet they do!   Even better perhaps: alternative representation (HCQT)…

Re. use of “Harmonic Constant Q Transform”, this slide by Brian McFee et al:

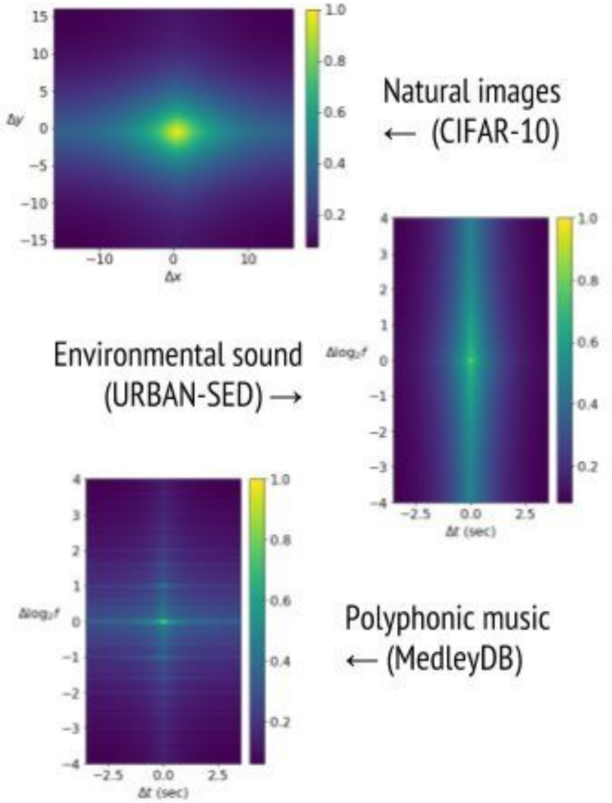
Modeling harmonics for pitch tracking

[with Rachel Bittner, Justin Salamon, Juan Bello, ISMIR 2017++]

- We often treat **spectrograms** like **images**, but is this *justified*?



- Convolutional networks exploit **local statistics** and **(2D) translational symmetry**
- We developed a ***Harmonic CQT*** representation to **allow convnets to easily model** *harmonics*
- **Insight**: treat *harmonics* like *color channels*!



@drscotthawley

More Links

More Links

Check out ASPIRE: @aspirecoop
Next meeting ~June 11: report from NIME.

Some links & news: http://www.creativeai.net/

Online ML Courses:

- Rebecca Fiebrink's: https://www.kadenze.com/courses/machine-learning-for-musicians-and-artists-v
- Andrew Ng's: https://www.coursera.org/learn/machine-learning, https://www.coursera.org/learn/neural-networks-deep-learning