

Musical Artificial Intelligence

Basic Principles +
Machine Learning in Max

Eli Stine \ Oberlin Con \ February 2024

Who I Am / What I Love

_Composer, Programmer, Media Designer, Educator

_Things I Love:

Electroacoustic Music, Spatial Audio, Film/XR Sound Design,

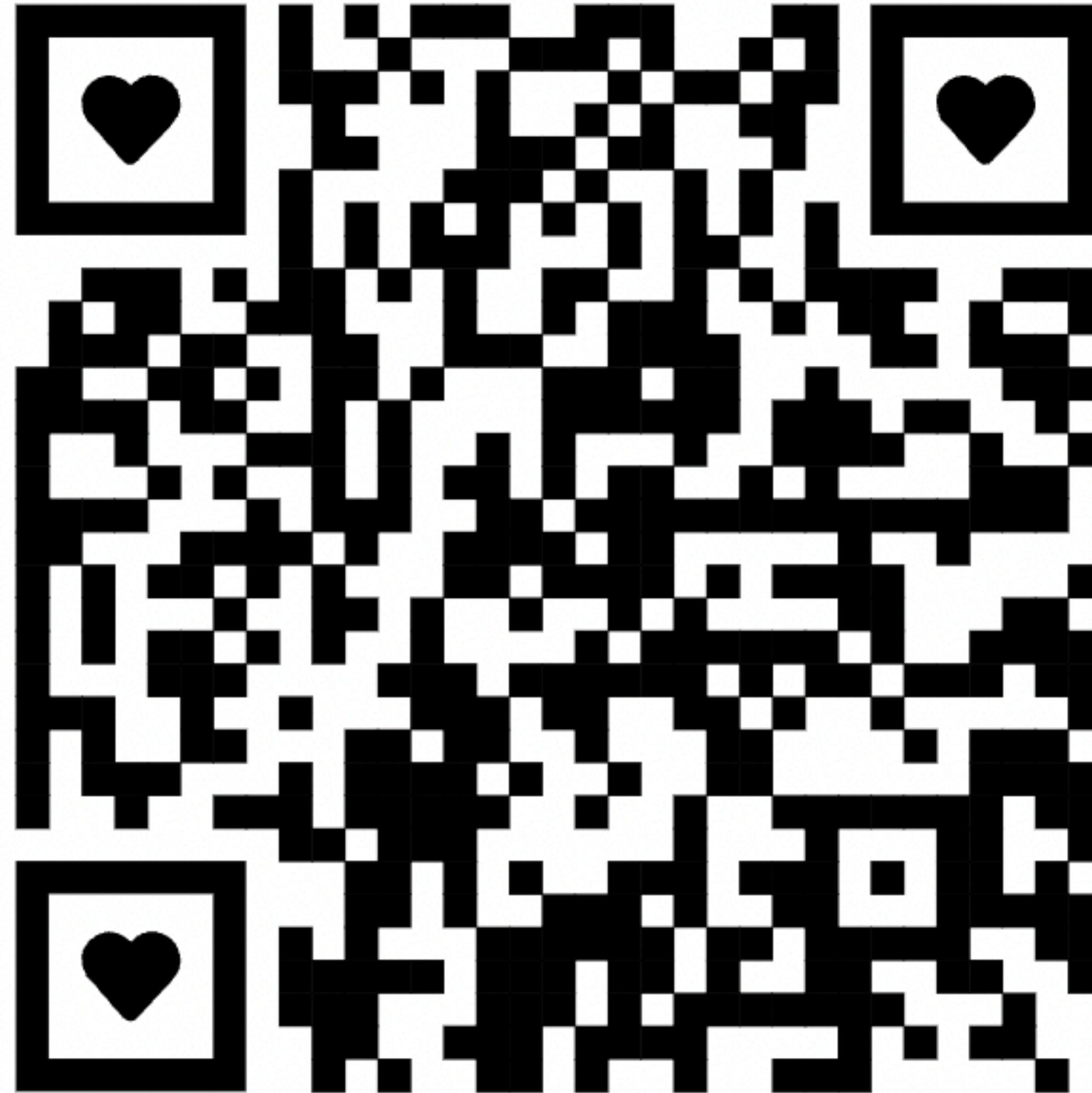
Interactive Media, Computer Graphics

Code for this Talk

No/minimal coding experience required

Requires Max 8 (demo is just fine)

Entirely optional



github.com/estine/mlinmax

Musical AI: Talk Structure / Contents

_ (Personal) History + Definitions

_ Linear Regression + Supervised/Unsupervised Learning

_ Deep Learning

_ Musical Examples + Thoughts

What is Machine Learning (ML)?

Def.: Broad definition: we want a computer to learn how to emulate, or model, some data or process

Implementation: given an input, we want to build a model that gives us the correct output for that input

Some useful terminology that will show up: features, training, classification, unsupervised/supervised/deep learning

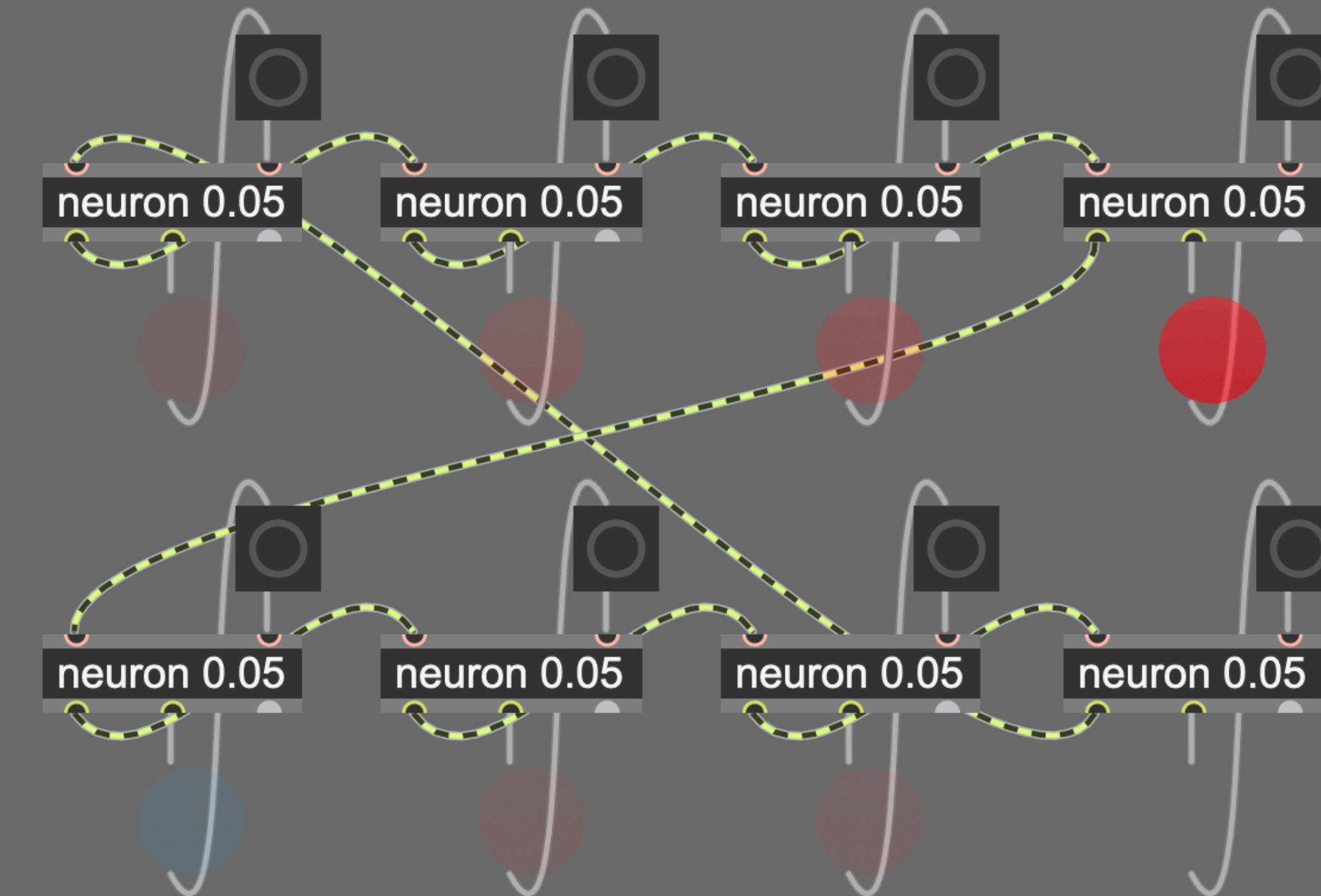
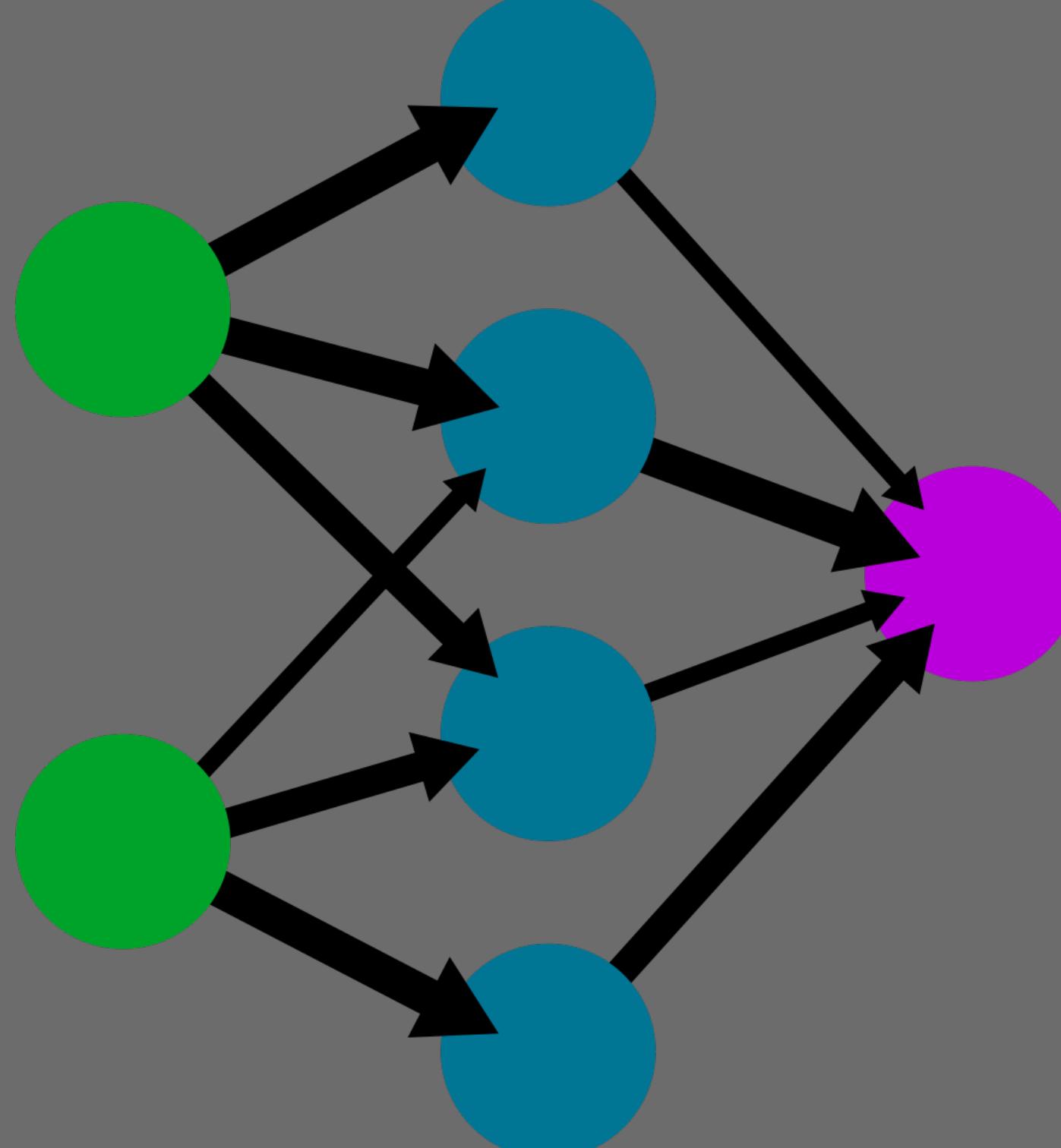
My Entry + Interests

- _ Harvestworks ↡ Neural Networks / AI Class ↡ Markov Models*
- _ Fascination with modeling nuanced and fragile natural/organic/chaotic systems using a completely discrete and stable tool (Digital Computer)*
- _ Renewed cultural interest in audio AI/generation systems
⇒ more engagement/funds/development in tools, etc.*

A quick divergence: Neural Networks

A simple neural network

input layer hidden layer output layer



Max patch with a set of neurons that are triggered by, and generate, audio

Musical Machine Learning: Some Historical Context

- _Musical Dice Game* (compose a minuet from random numbers) ~1750s
- _Dreaming of Computer Music* - Ada Lovelace 1843
- _ILLIAC Suite* (computer-composed string quartet) - Lejaren Hiller 1958
- _Computer Music* - Max Mathews ~1960s
- _Voyager System* (real-time digital improvisation) - George Lewis 1986
- _Experiments in Musical Intelligence* (style matching) - David Cope 1991
- _Deep Learning* (modern ML revolution) - Geoffrey Hinton 2016

Raw vs. Symbolic Musical AI

Def.: Music may be defined in many ways, but a common formal distinction in musical AI systems is made between generative systems that are

symbolic (note-based) or
raw (digital audio sampled-based)

Symbolic task: generate notes in fugue in the style of Bach
Raw task: generate digital audio samples at 44.1kHz, 16 bit

Simple Linear Regression

linearregression (presentation)

ML in Max | Simple Linear Regression

1 Inputting the Data:

Scatter plot of data

Concerts attended

Reported level of happiness

Blue = data
Green = fitted line
Red = error

2.327 Total Error

2 Matching to a Line By Hand:

$$y = 0.6x + 1.90$$

m: 0.558 (slope) b: 1.891 (intercept)

3 Replacing with Machine Learning:

X js linearregression.js

4 Using the Model for Prediction:

Concerts Attended: Estimated Happiness

1 2.448

Supervised Learning

ML in Max | Supervised Learning (SVM)

*Requires: ml-lib

1 Input Training Data, Labelled by Class (which is what makes this supervised)

Switch to Training Mode

KICKS

- ▶ Alma Mater [KIC...]
- ▶ Chop Won't ...
- ▶ Gummo [Kick]...
- ▶ Michigan [KIC...]
- ▶ Old Kick - 2 [KI...]

SNARES

- ▶ 40 Clip [SNAR...]
- ▶ All Night [SNA...]
- ▶ Barbeque [SN...]
- ▶ Boomerang [S...]
- ▶ Cashin' [SNAR...]

Feature Visualization

0.59 1.00 0.54

2 Train on the Input Data TRAIN CLEAR

3 Give it new, unclassified input, get classification

Switch to Query Mode

KICKS

- ▶ 100 Clip [KICK...]
- ▶ Bet Cool [KICK...]
- ▶ EA [Kick].wav
- ▶ Hell Shell [KIC...]
- ▶ Old Kick - 1 [KI...]

SNARES

- ▶ 10 Pacc [SNA...]
- ▶ 100 Racks [SN...]
- ▶ Backroom [SN...]
- ▶ Bet Cool [SNA...]
- ▶ Brentrambo [S...]

Answer: Hey, that's a snare!

Unsupervised Learning

ML in Max | Unsupervised Learning (Fuzzy Clustering)

Requires: ml.

1 Input Training Data, completely unlabeled (which makes it unsupervised)

Switch to Training Mode

KICKS

- ▶ ⏪ Alma Mater [KIC...]
- ▶ ⏪ Chopa Won't ...
- ▶ ⏪ Gummo [Kick]...
- ▶ ⏪ Michigan [KIC...]
- ▶ ⏪ Old Kick - 2 [Kl...]

SNARES

- ▶ ⏪ 40 Clip [SNAR...]
- ▶ ⏪ All Night [SNA...]
- ▶ ⏪ Barbeque [SN...]
- ▶ ⏪ Boomerang [S...
- ▶ ⏪ Cashin' [SNAR...

Feature Visualization

0.32 1.00

2 Give it new, unclassified input, get classification

Switch to Query Mode

KICKS

- ▶ ⏪ 100 Clip [KICK...]
- ▶ ⏪ Bet Cool [KICK...]
- ▶ ⏪ EA [Kick].wav
- ▶ ⏪ Hell Shell [KIC...]
- ▶ ⏪ Old Kick - 1 [Kl...]

SNARES

- ▶ ⏪ 10 Pacc [SNA...]
- ▶ ⏪ 100 Racks [SN...]
- ▶ ⏪ Backroom [SN...]
- ▶ ⏪ Bet Cool [SNA...]
- ▶ ⏪ Brentrambo [S...

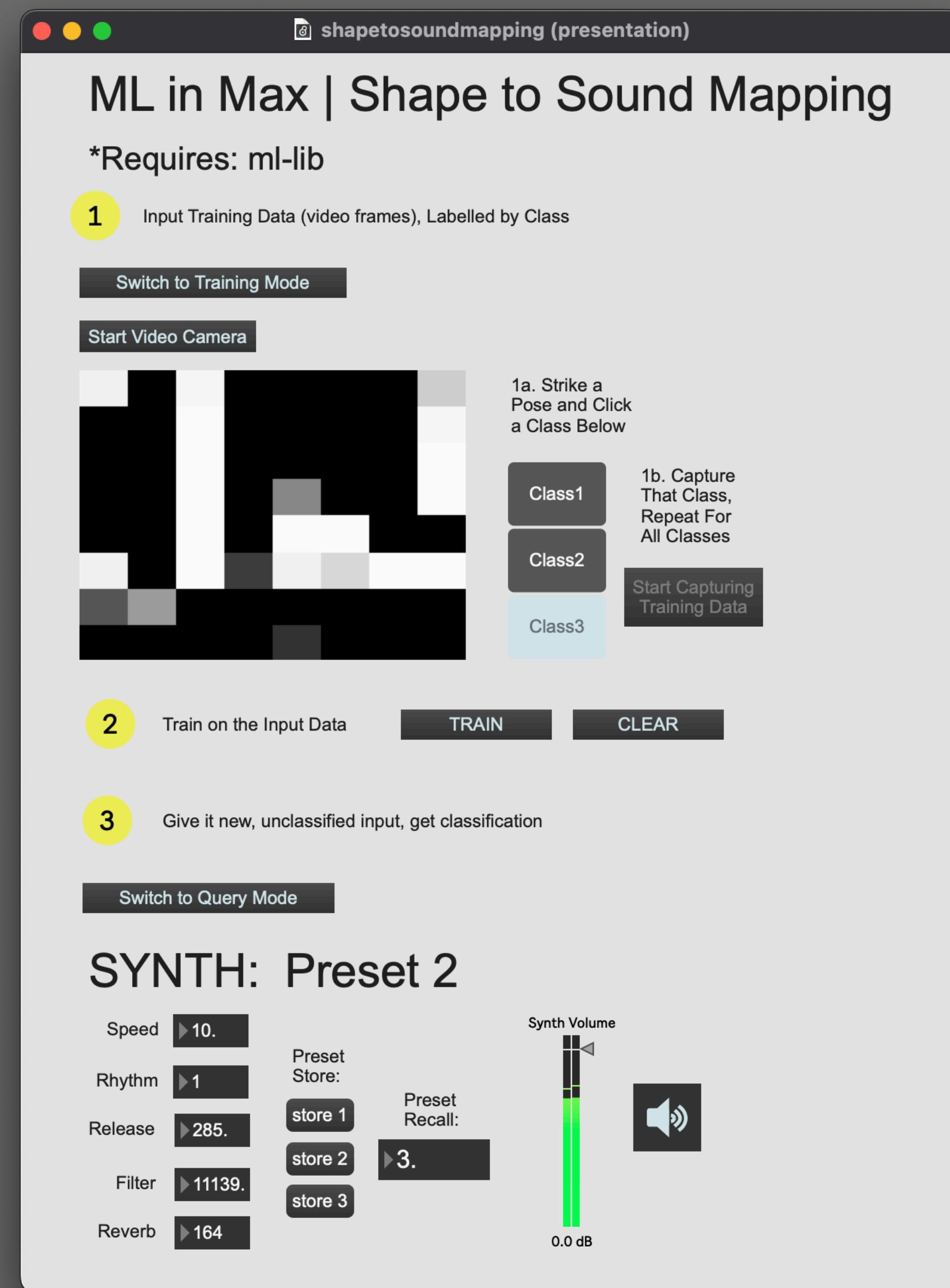
Data Visualization + Clusters

CLEAR

Answer: Hey, that's in cluster 2!

Membership Grade: 99%

Musical Example: Shape to Sound Mapping



Musical Example: Real-Time Concatenative Synthesis

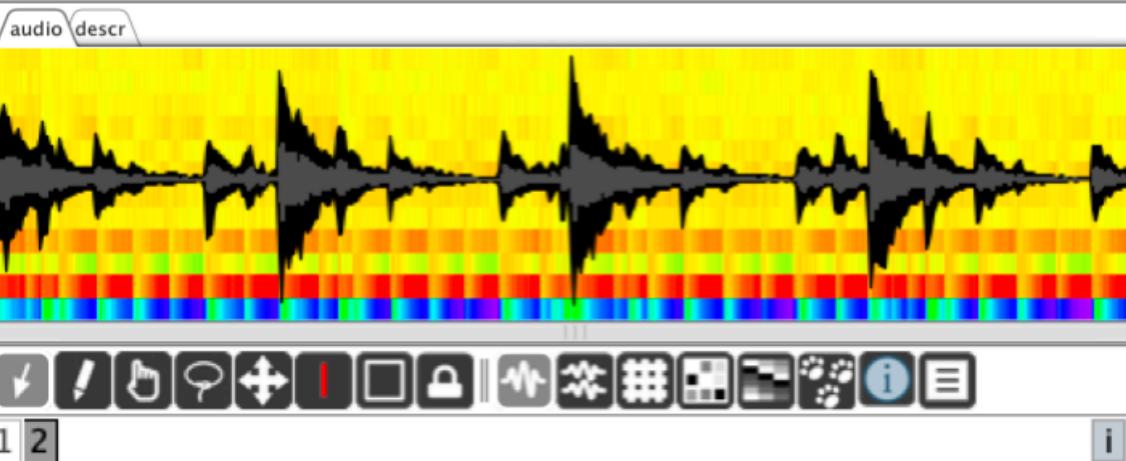
ML in Max | Real-Time Concatenative Synthesis

*Requires: MuBu 

1 Load our destination sound(s), which is automatically sliced into frames and analyzed (the "descr" tab below contains these)

```
readappend duduk.aif @name audio  
readappend brushes.aif @name audio  
readappend sho0630.aif @name audio
```

clearall



2 Pick a source sound, which is analyzed in real-time and ML picks the closest match in the destination corpus

Stop File cherokee.aif drumLoop.aif jongly.aif

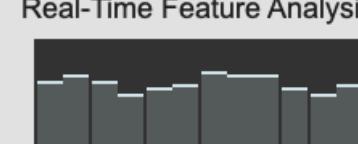
Mic Input Soundfile Input Soundfile Dry Which Destination Sound(s) To Use? current

-inf dB 0.0 dB 0.0 dB

Audio Heuristic Weights



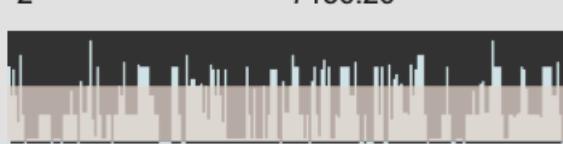
Real-Time Feature Analysis



Granular Synthesizer Settings

period	5.	0.
duration	0.	8.
positionvar	3.	

Concatenative Synthesizer Output



Destination Buffer + Location in File

2 7156.26

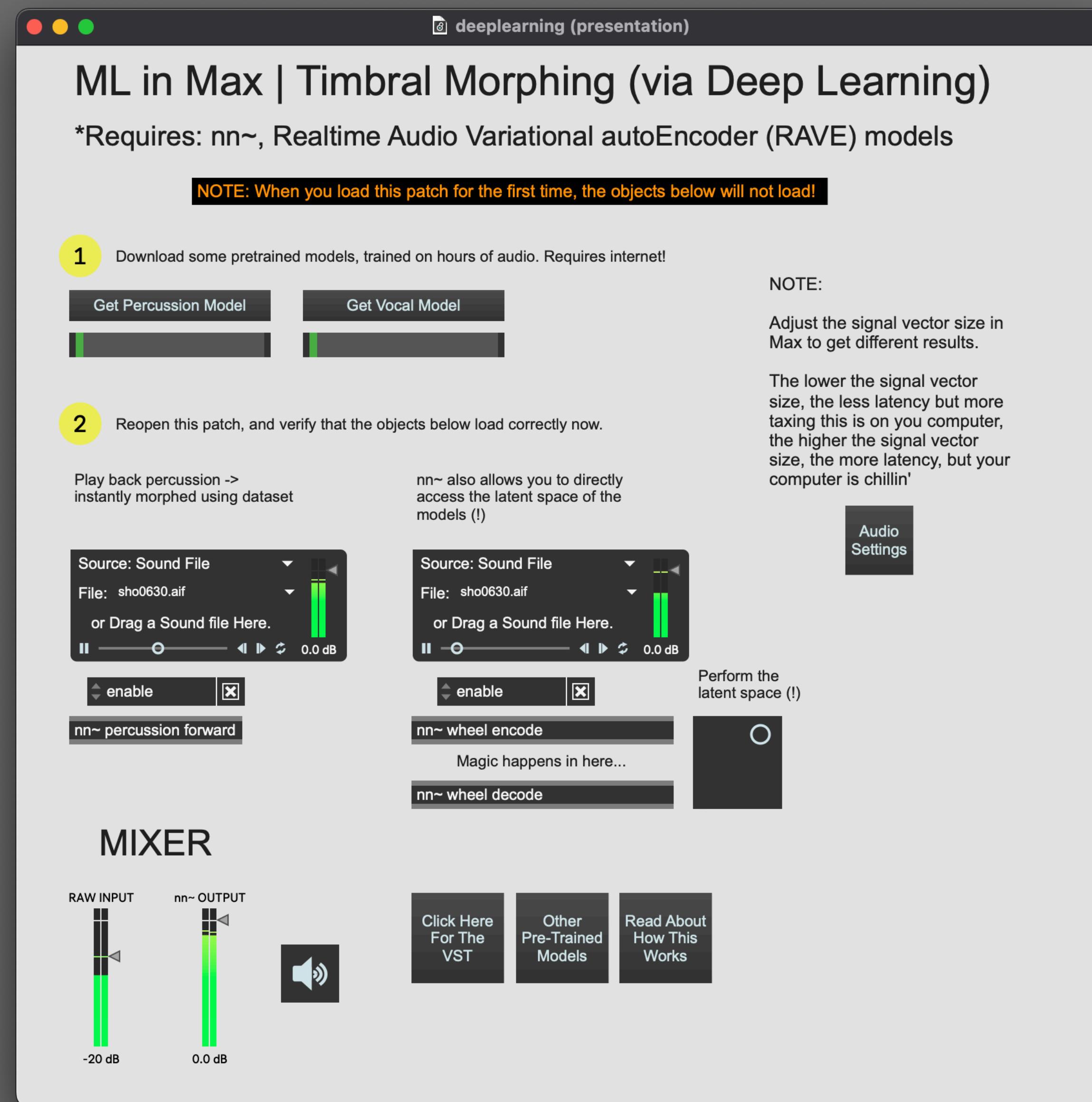
What is going on here?

Our goal is to map from a source sound to a destination sound in real-time.

First, we select our destination audio file(s), split it into little chunks (called windows) and analyze each chunk. Here, we are analyzing the windows using Mel-Frequency Cepstral Coefficients (MFCCs), which give us robust spectral information about a sound in just 12 features. After this step, we have a database of windows and their corresponding features.

Second, we select a file that will be our input. The windows of this sound are analyzed in real-time and using a supervised learning technique called k-Nearest Neighbors (or k-NN) we take each source window's features and get the most similar window from the destination database. We then pass this window's location to a granular synthesizer, which plays back that window as a grain.

Musical Example: Timbral Morphing (via Deep Learning)



No-Input Raw Generative Audio

Weird Neural Net Playground (presentation)

The interface consists of several sections:

- ALL INPUT:** A spectrogram showing multiple vertical bands of energy.
- Mic Input:** A vertical slider labeled "Mic" with a value of "1.4 dB".
- Synthesized Input:** A section with two buttons: "Tone Off" and "Click".
- Input:** A vertical slider labeled "Input" with a value of "0.0 dB".
- ALL OUTPUT:** A spectrogram showing multiple vertical bands of energy.
- Raw:** A vertical slider labeled "Raw" with a value of "0.0 dB".
- 8va:** A vertical slider labeled "8va" with a value of "0.0 dB".
- 15va:** A vertical slider labeled "15va" with a value of "0.0 dB".
- Feedback Network:** A section with a "Feedback" slider set to "-12 dB" and a dropdown menu "Type of Feedback" set to "Pitch-Shifted". Below it is a circular icon labeled "Transp" with "1200.00 ct".
- Delay Line Length:** A slider with a value of "511.0 ms".
- Delay Line Feedback:** A slider with a value of "0.393".

Requirements:
This patch requires nn~ and also the wheel.ts model to be downloaded in an accessible location

By default the models are OFF (turn on toggles at bottom)

Directions:
The top section is the input, which can be synthetic (a 440 hertz tone, a click) or analog (microphone input)

The bottom section is the output, which is a vocal model in raw form or pitch shifted up one or two octaves, respectively.

The right section is a simple audio feedback network, which can result in some really fascinating results

(Try some no-input neural network playing...)

Thoughts on Musical AI/Generative Audio Systems

Creative Friction

Originality + Ownership

Access + Ethics

How does a music technology-focused artist navigate this?

Resources

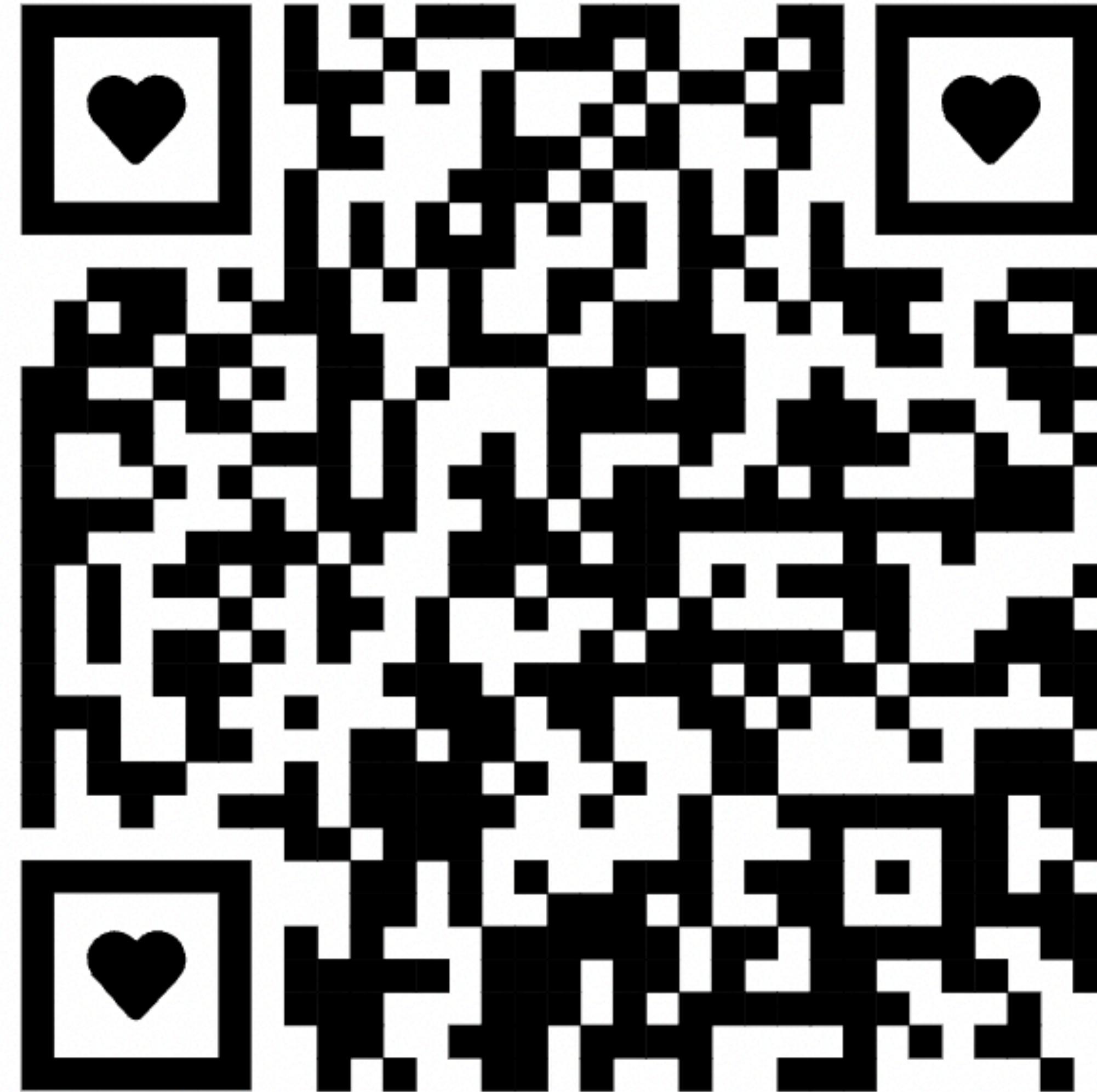
- _MusicLM: <https://google-research.github.io/seanet/musiclm/examples/>
- _DDSP: <https://magenta.tensorflow.org/ddsp-vst>
- _Mawf: <https://mawf.io/>
- _Neutone by Qosmo: <https://neutone.space/>
- _Wavenet: <https://deepmind.google/technologies/wavenet/>

Code for this Talk

No/minimal coding experience required

Requires Max 8 (demo is just fine)

Entirely optional



github.com/estine/mlinmax