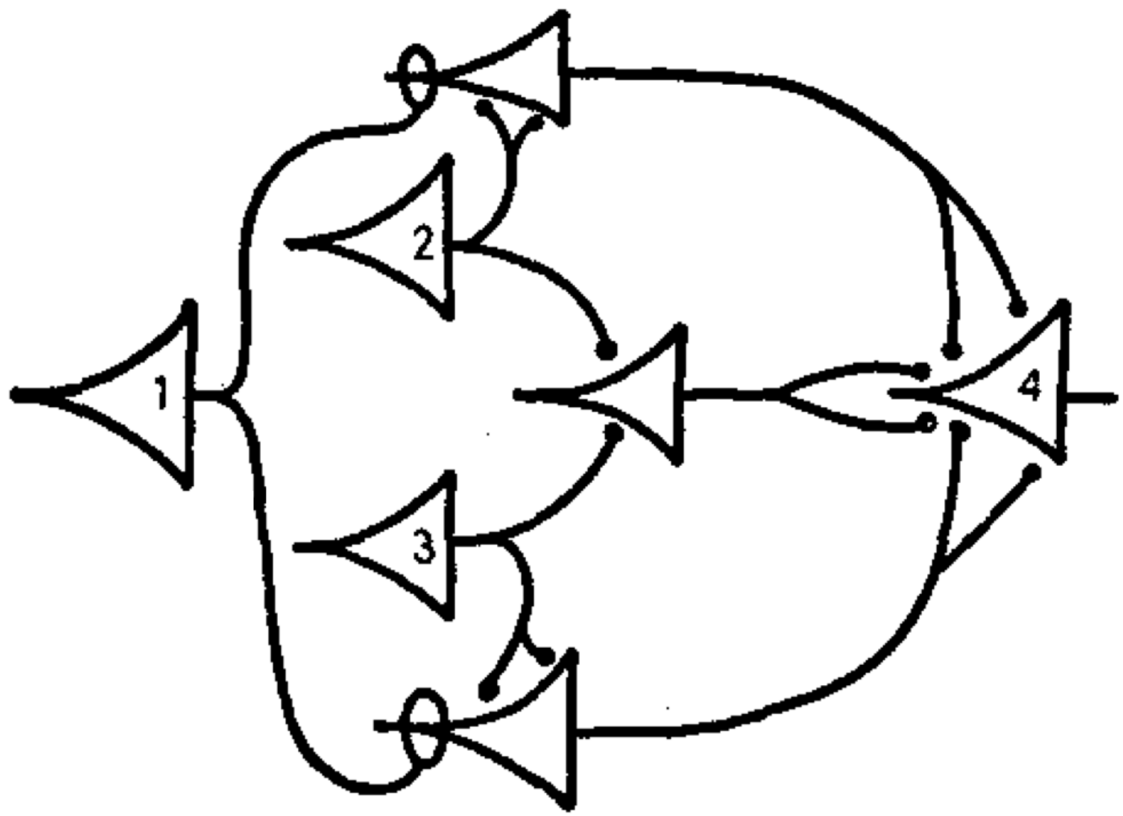
Contemporary Trends in Music AI

The Return of Neural Networks



Feb 5th, 2019, David Kant

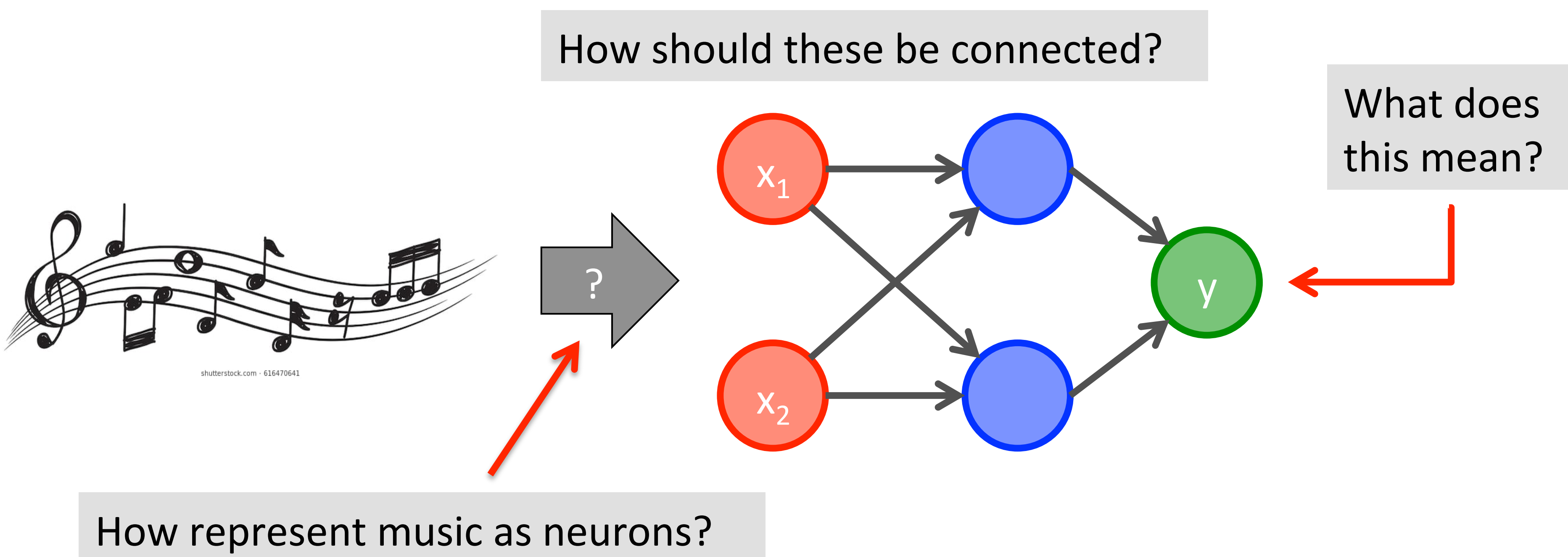
Music and ML Resources

- International Society for Music Information Retrieval (ISMIR)
- Music Information Retrieval Evaluation eXchange (MIREX)
- musicinformationretrieval.com
- Librosa (Columbia)
- Bregman Media Labs (Dartmouth)
- Essentia tools for audio and music analysis (upf Barcelona)

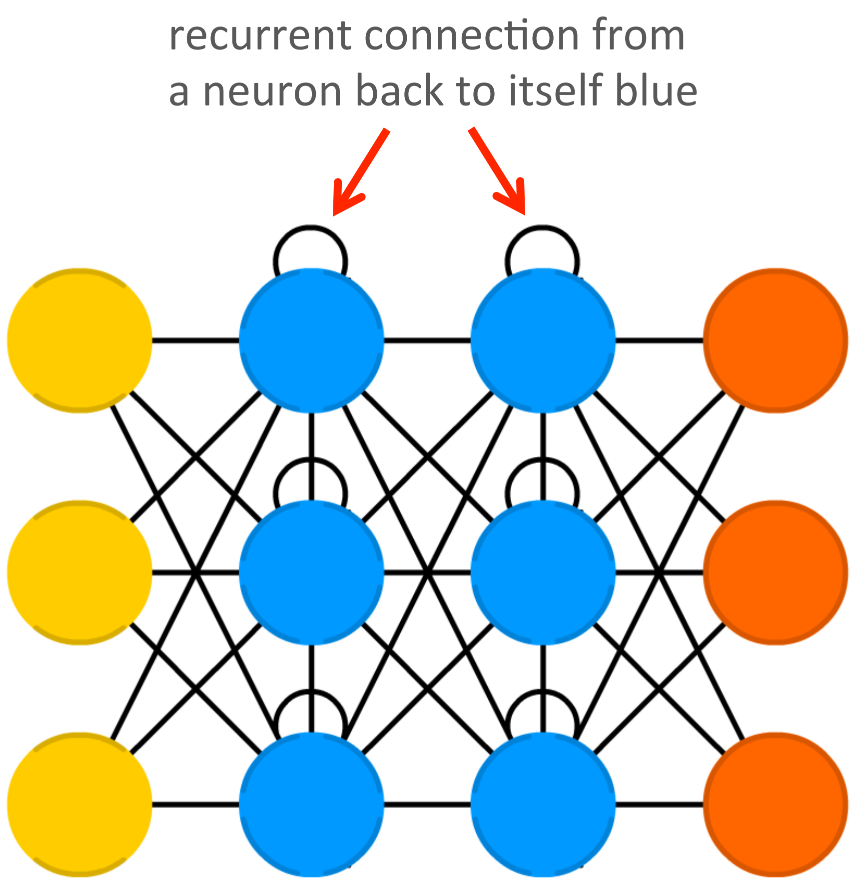
Two key questions

1) How do you represent music to a neural network?

2) Which network architectures as best for musical tasks?



Recurrent Neural Network (RNN)



"feedforward neural nets represent functions, where as recurrent neural nets represent programs…"

- a neural network in which the hidden layers have recurrent connections
- network output depends not only on the current input but by the entire history of inputs
- history is maintained by the state of the recurrent neurons, giving the network ***memory***
- recurrence allows the network to learn ***sequences***, not just input / output pairs

Generating Sequences with RNNs

- output represents the probability distribution of the next element in the sequences given the sequence of previous inputs
- sample from this distribution and feed that result right back in as the next input

THE RAiMONES

making punk rock intelligent, artificially intelligent.

- THE RAiMONES (2018) by Matthias Frey
- generates guitar and bass lines plus lyrics in the style of The Ramones
- recurrent neural network (RNN) trained on a symbolic *musical* representation (MIDI) of songs by the Ramones
- trained on 130 songs + lyrics of all 178 songs
- Long Short-Term Memory Recurrent Neural Network
- Separate networks used for instruments and lyrics

http://raimones.komakino.ch/#audiooutput

Music Representation

- all songs transposed to the same key (C Major)
- only the guitar and bass lines are used
- each songs is serially concatenated into one big list
- rhythms quantized to sixteenth notes
- input: matrix one row for bass, six for guitar, two extra bits for each mute and hold
- sequence length 32 or 64 (2 or 4 bars)

Guitar	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45
	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52
	57	57	57	57	57	57	57	57	57	57	57	57	57	57	57	57
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Bass	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Time: 1/16th Quantization

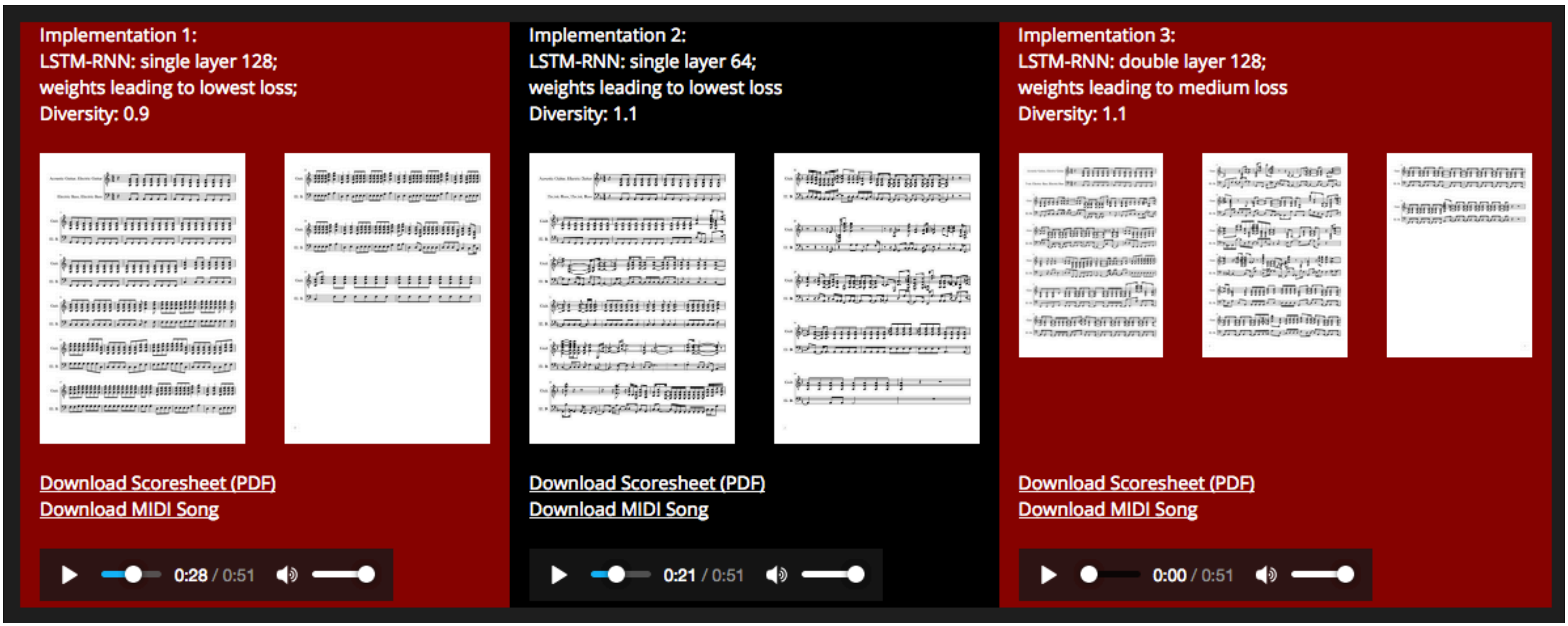
http://raimones.komakino.ch/#music

THE RAiMONES

making punk rock intelligent, artificially intelligent.

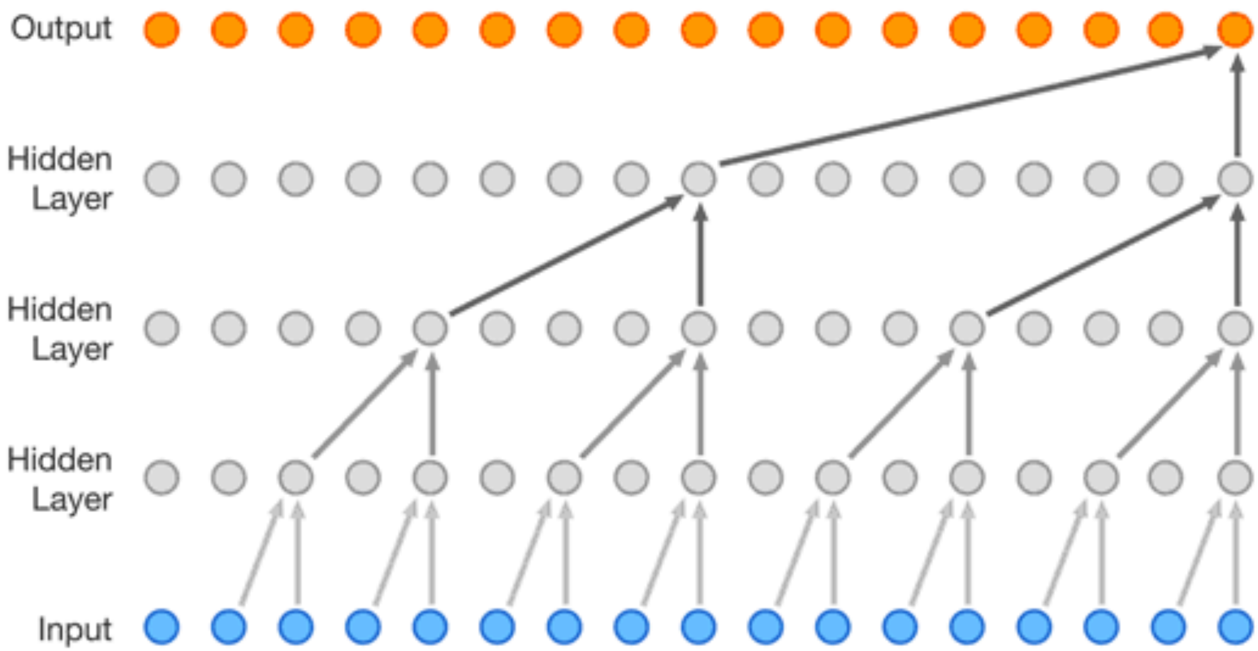
Generated MIDI outputs

- the *temperature or* “diversity” controls how “chaotic” or “creative” the output --- a variety of outputs can be generated from the same model by adjusting this parameter



three example outputs given the same initial seed: two bars of “Beat on the Brat” http://raimones.komakino.ch/#midioutput

WaveNet: A Generative Model for Raw Audio



https://deepmind.com/blog/wavenet-generative-model-raw-audio/

DADABOTS
A Platform For Artificial Artists

Home / Albums

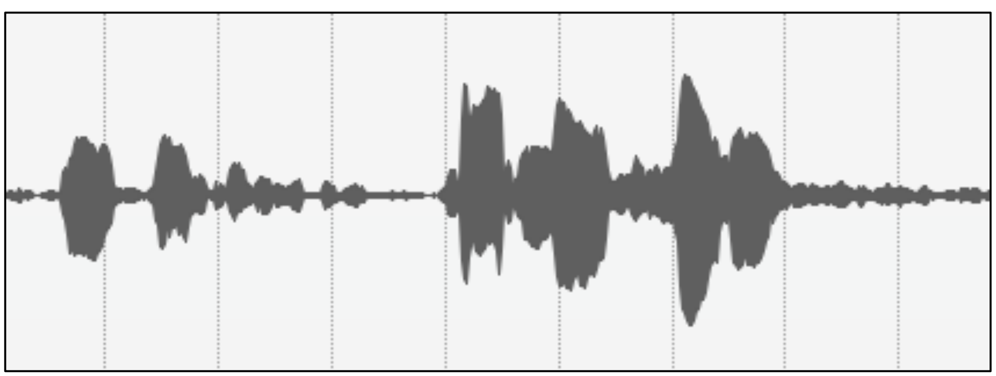
Generating Black Metal and Math Rock

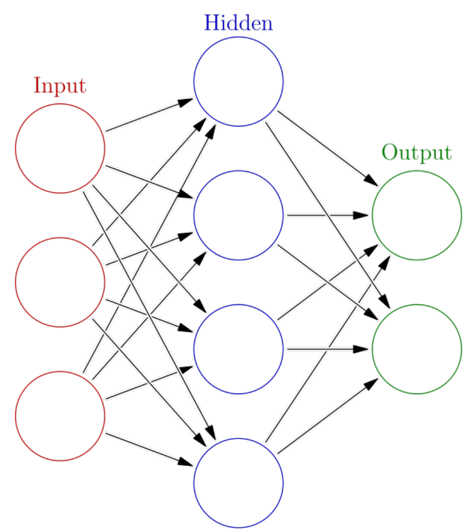
- DADABOTS (2018) by CJ Carr and Zack Zukowski
- generates music in modern genres such as black metal and math rock where timbre is an important compositional feature
- recurrent neural network (SampleRNN) trained on *waveform* (sample-by-sample) representation of songs
- unlike MIDI and symbolic models, SampleRNN generates raw audio in the time domain.
- an application of neural synthesis (similar to WaveNet) to musical audio rather than speech

http://dadabots.com/#new-albums

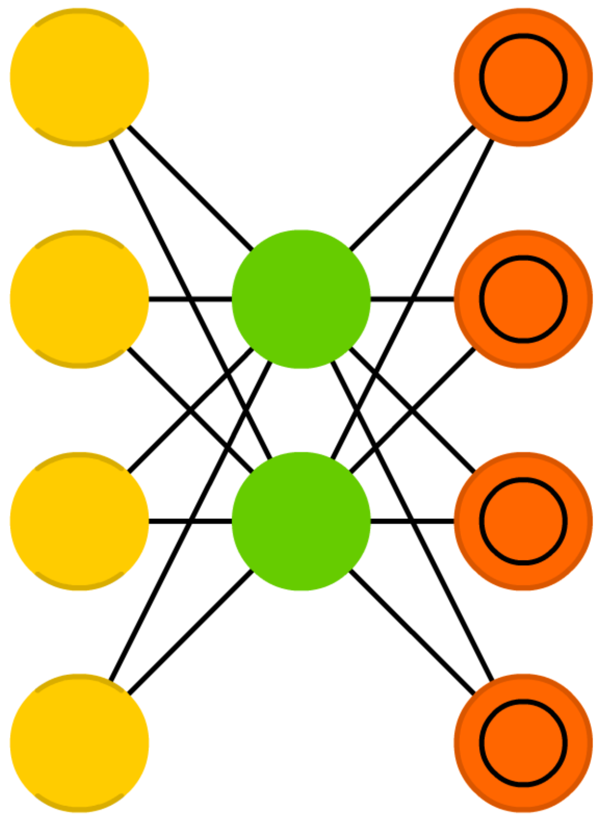
Generating Black Metal and Math Rock

- pre-process each audio dataset into 3,200 eight second chunks of raw audio data
- 2-tier SampleRNN with 256 embedding size, 1024 dimensions, 5 to 9 layers, LSTM
- audio downsampled to 16 kHz (that’s why it sounds not so great)
- trained for about 3 days, generating audio intermittently



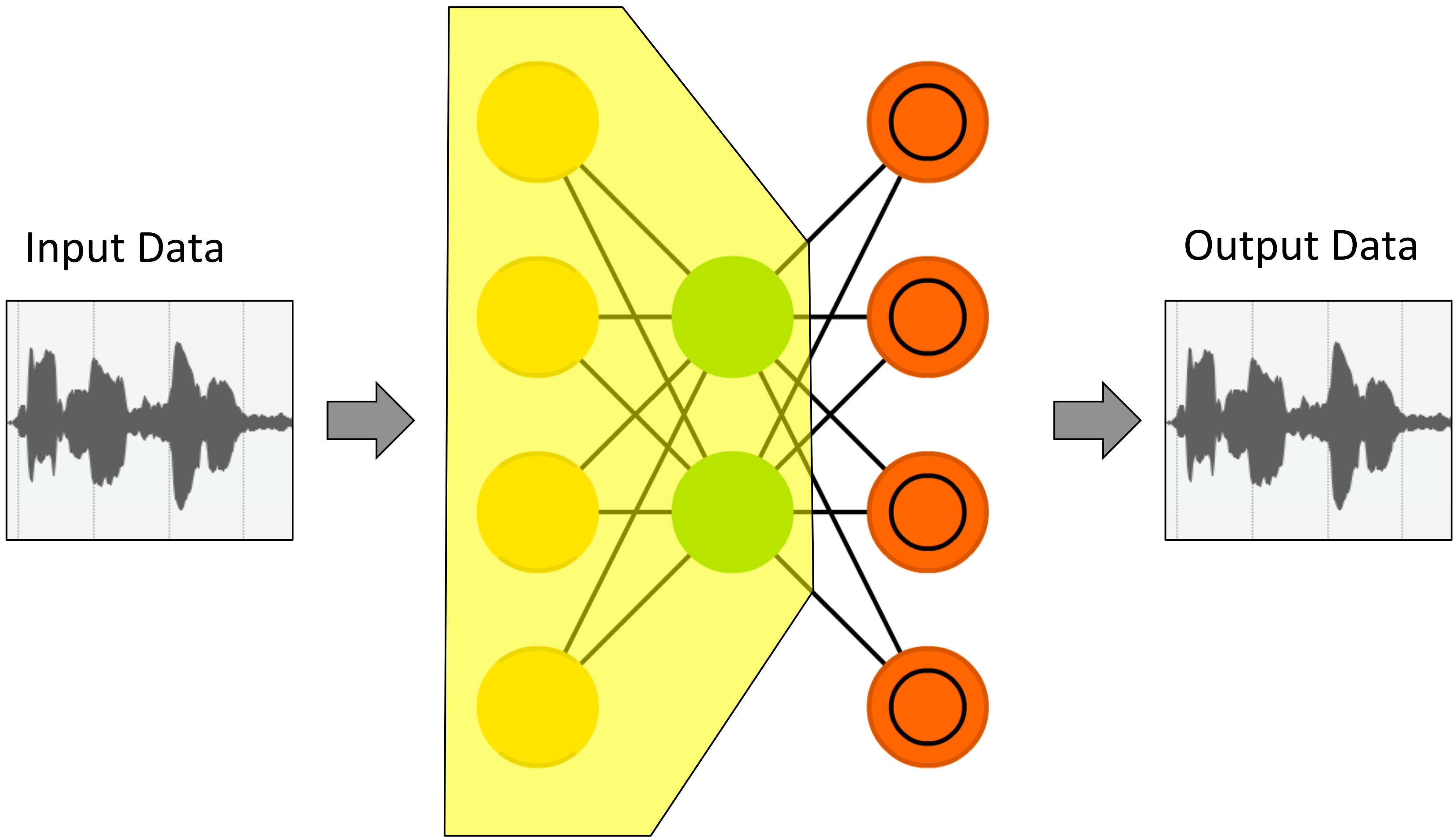


Semantic (Latent) Spaces
using Autoencoders



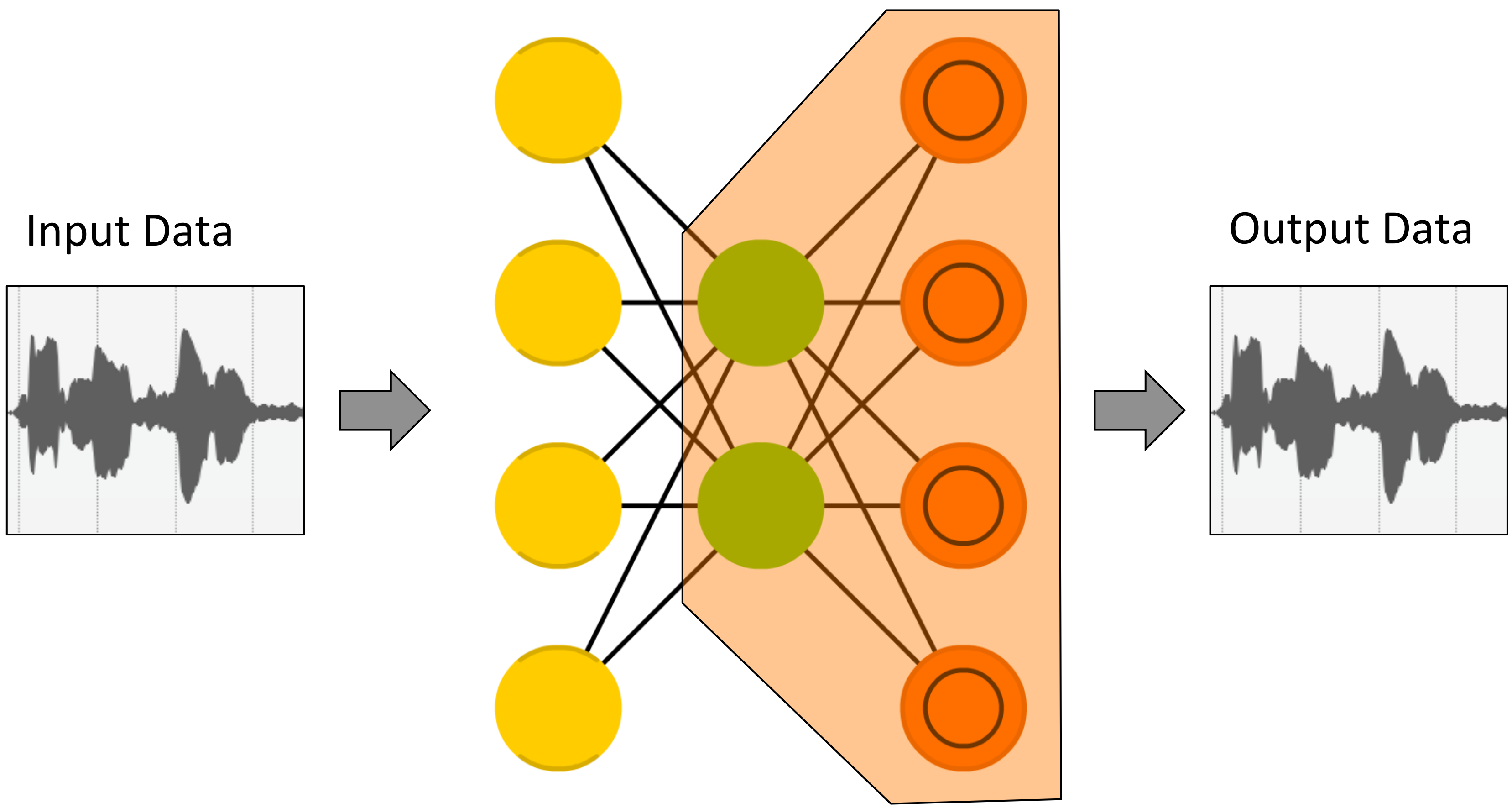
1) Encoder

maps the input data to the activations in the hidden layer



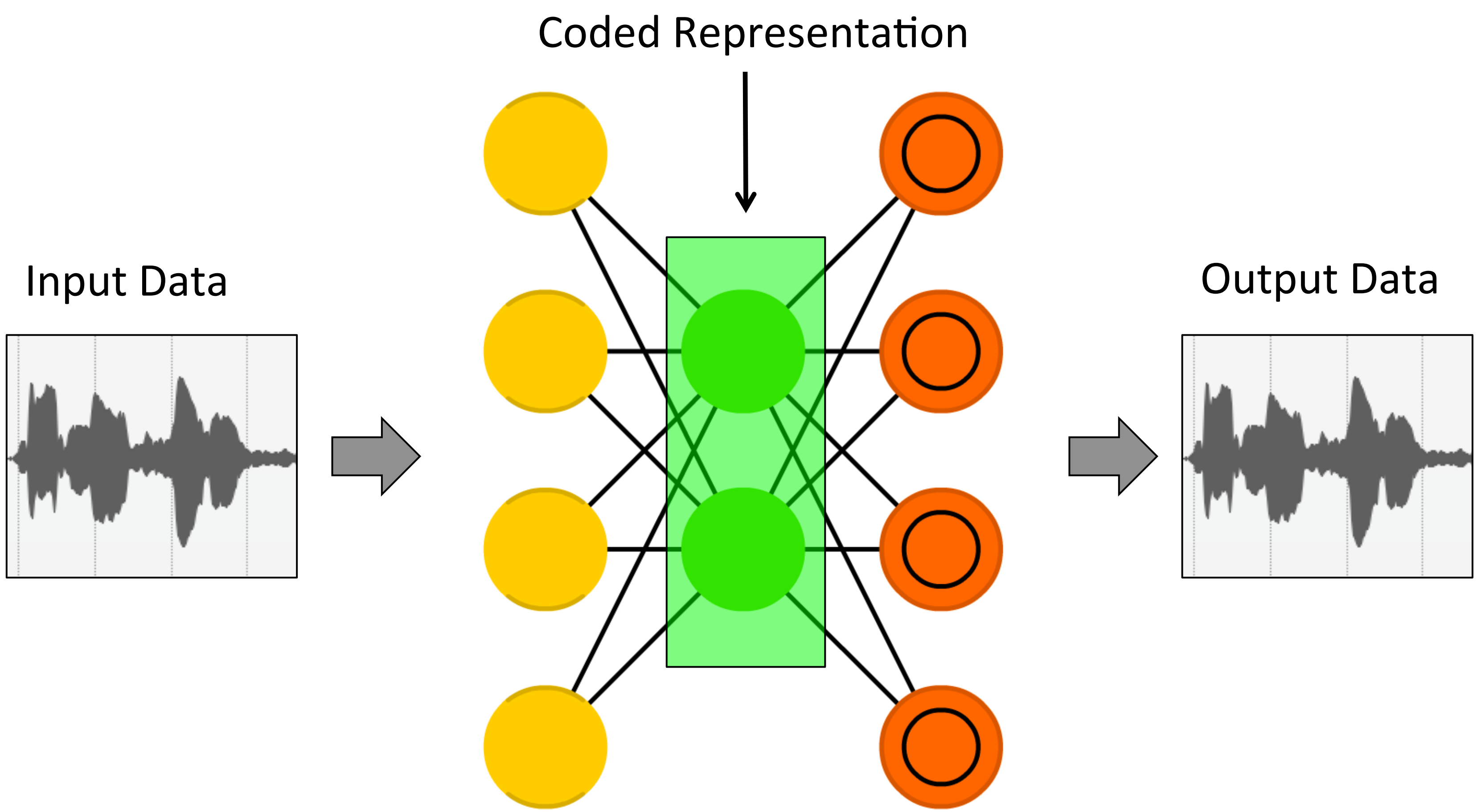
2) Decoder

reconstructs the output data from the activations in the hidden layer



3) Latent Space

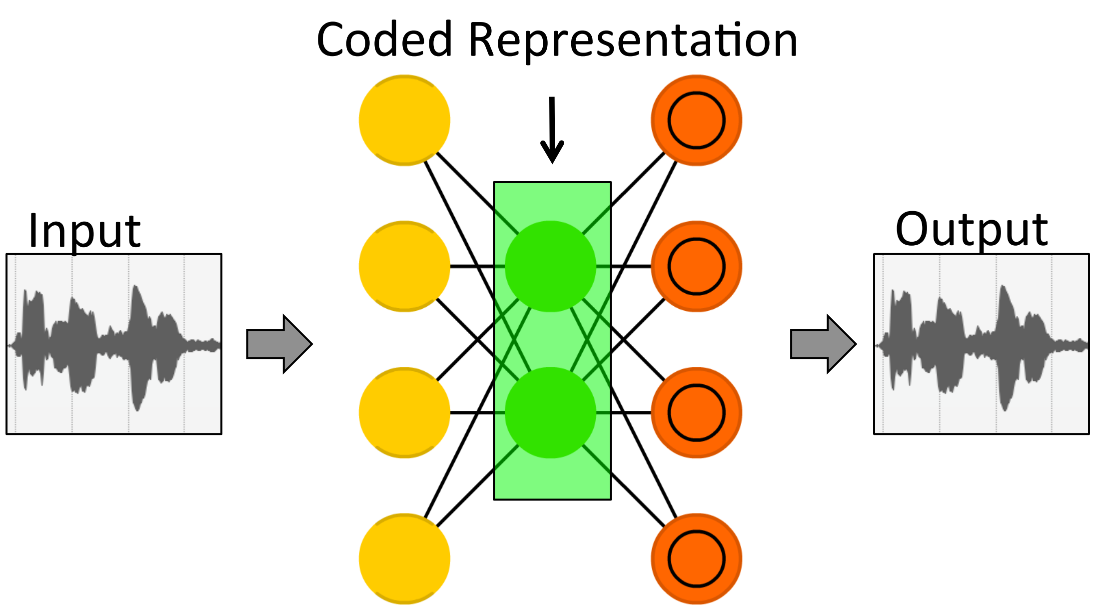
activations of the hidden layer is a coded representation of the data



- the hidden layer is called the *coded representation* or *latent space*. the activations of the neurons in the hidden layer represent the original input data according to whatever encoding / decoding scheme the network has learned through training.

The Coded Representation

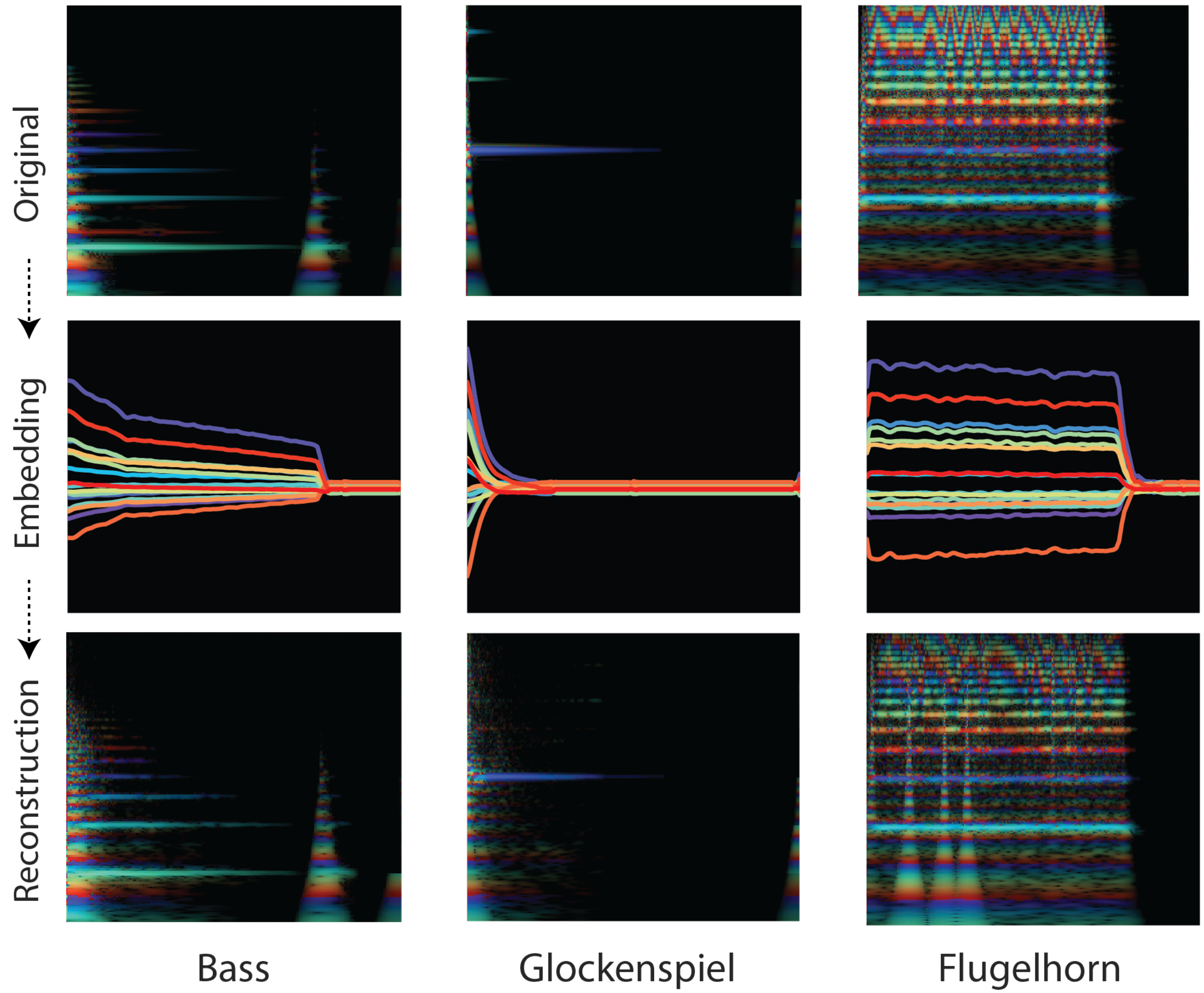
- often the hidden layer has *fewer* neurons than the input/output layers, forcing the network to learn a compressed representation
- hidden layer can be *equal to* or *larger* than the input/output layers and subject to additional constraints. this leads to a few variants…



Autoencoder Variants

- *compression autoencoder*
  - fewer hidden neurons than the input/output layers
- *sparse autoencoder*
  - the hidden layer has more neurons that the input/output layers, BUT only a few of the hidden neurons are allowed to be active at the same time
- *denoising autoencoder*
  - able to recover original signal from noisy or distorted input
  - instead of training on X -> X, train on X' -> X where X' has introduced distortion
- *variational autoencoder*
  - enforces additional constraints in the form of the activation values of the hidden layer, often as statistical distributions

Nsynth (2017), Google Magenta



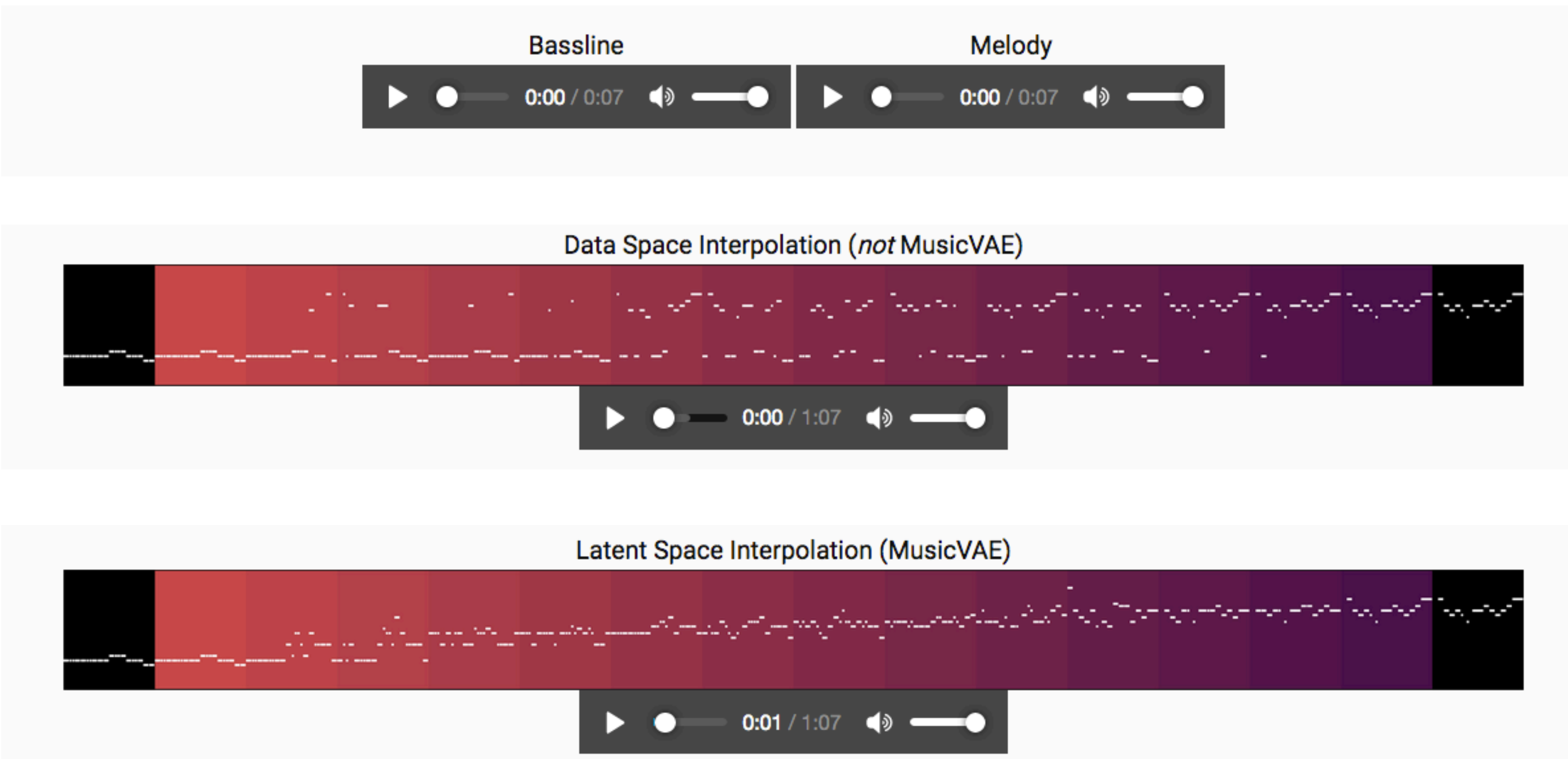
https://magenta.tensorflow.org/nsynth

MusicVAE (2018), Google Magenta

- a collection of Machine Learning tools that allow artists to explore, blend, and mix musical ideas
- uses *Variational Autoencoder* or a *latent space model* to represent high dimensional musical materials in a lower dimensional code that can be manipulated
- the latent code learns structural characteristics of the dataset, and semantically meaning transformations, such as “note density,” lie along vector operations

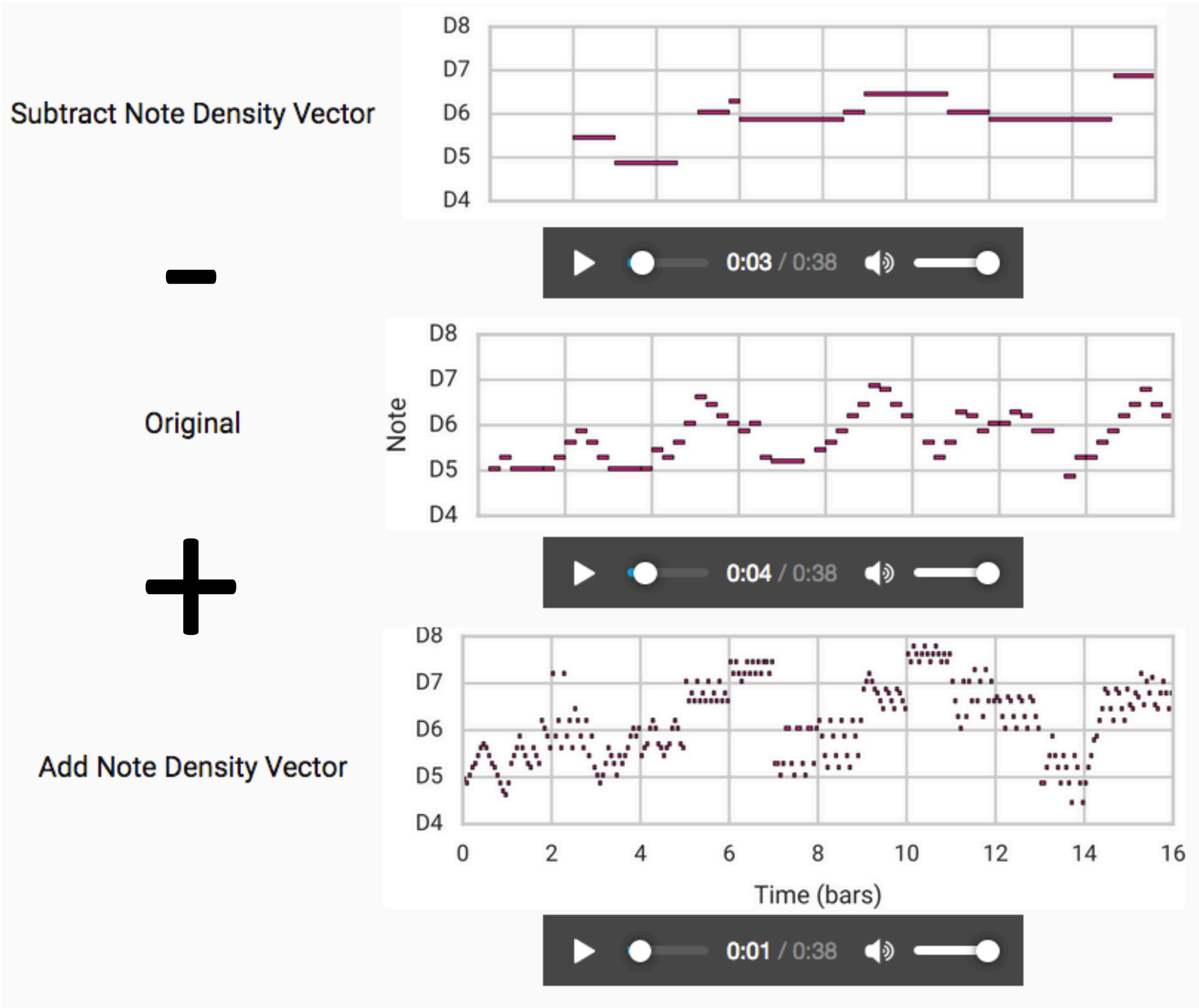
MusicVAE (2018), Google Magenta

- Interpolate between two melodies in latent space



https://magenta.tensorflow.org/music-vae

MusicVAE (2018), Google Magenta



- attribute vector arithmetic to control semantic qualities of generated music