Learning to Play Jazz with Deep Belief Networks

Greg Bickerman, Harvey Mudd College
Sam Bosley, Stanford University
Peter Swire, Brandeis University
Robert Keller, Harvey Mudd College

Motivation

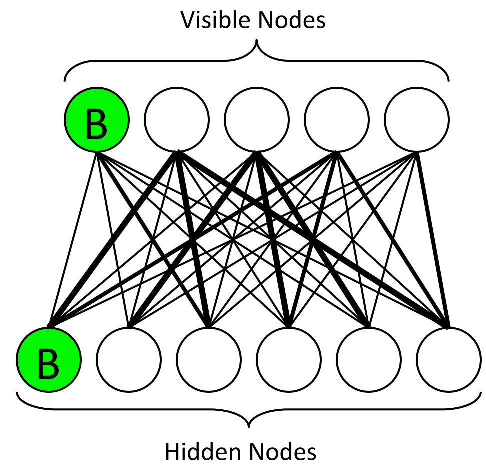
- People are able to improvise jazz on the spot
- Jazz Improvisation
  - Patterned and structured
  - Creative and novel
- Could a machine learn to improvise as well as a human?

Motivation

- Artificial jazz improvisers already exist
  - GenJam
    - Supervised genetic learning
  - Impro-Visor
    - Extensive musical knowledge built in
- Interested in unsupervised learning
- Minimal representational assumptions

Restricted Boltzmann Machines

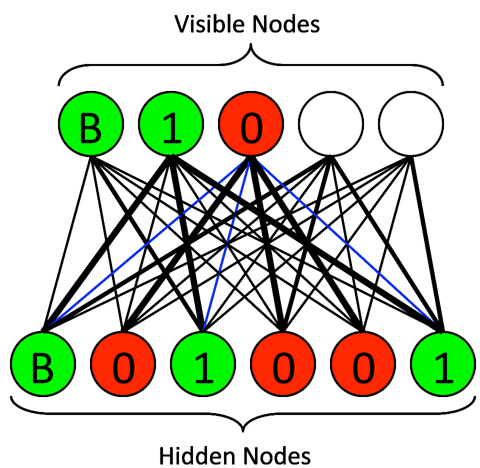
- 2-layer network
  - Visible layer
  - Hidden layer
- Nodes
  - Interconnected
  - Can be set ON or OFF
- Weights
  - Assigned to each connection
  - Symmetric



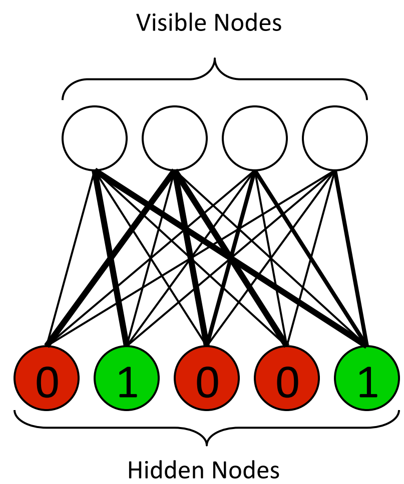
Activation

- Nodes activated **probabilistically** based on activation states of nodes in opposite layer
  - Compute weighted sum of active connections
  - Activation function determines probability of firing

Activation



Activation

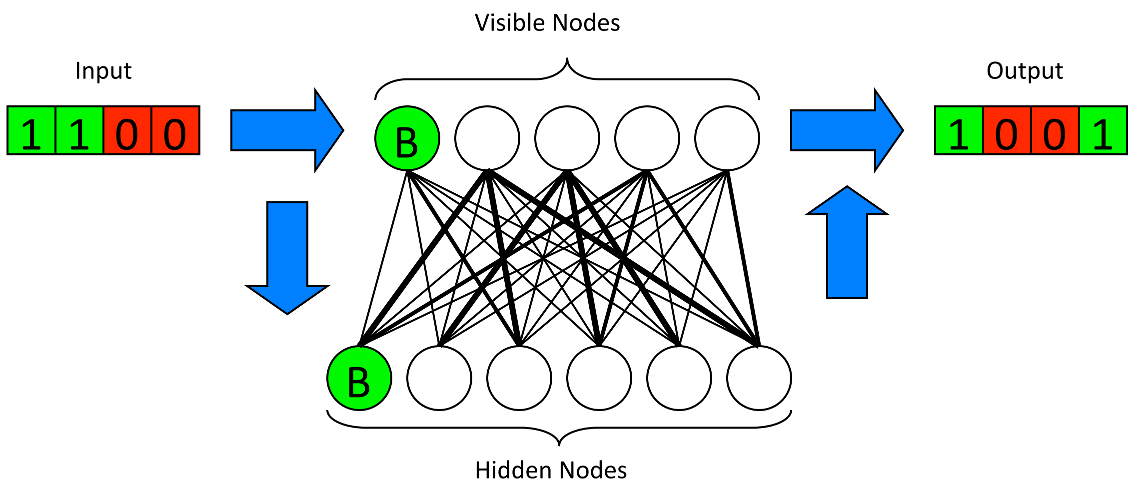


Input / Output

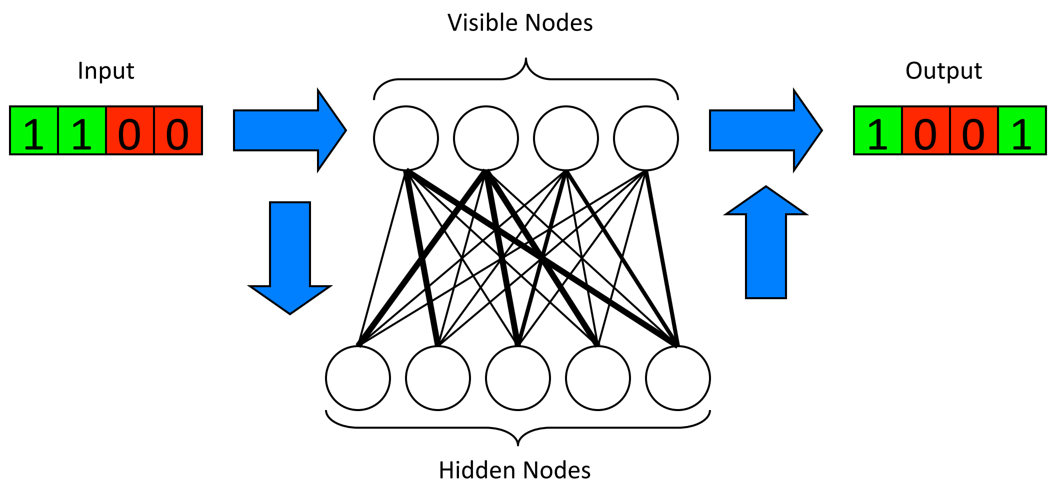
- Input
  - Binary data sequences
  - Mapped onto visible neurons

- Output
  - Identically sized data sequences
  - Read off of visible neurons

Input / Output



Input/Output



Training

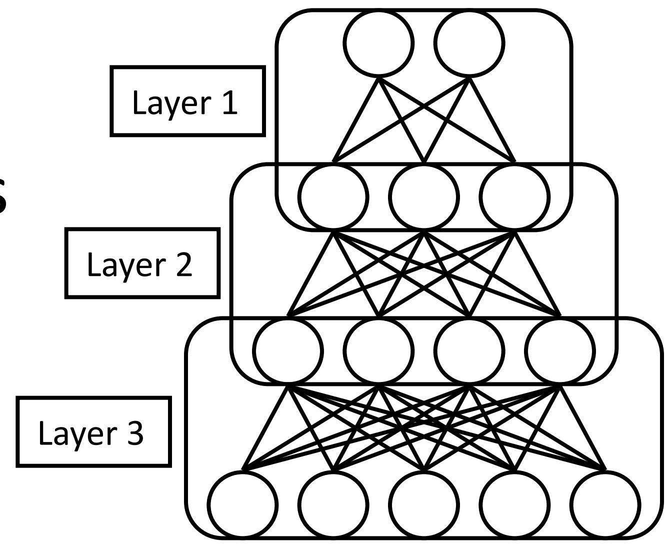
- Contrastive divergence method
  - Activate network normally
  - Activate network with inputs “clamped”
  - Adjust weights to make normal activation behave more like clamped activation

Deep Belief Networks

- Use individual RBMs as layers in a larger network

- Hidden layer of one RBM forms input layer of another

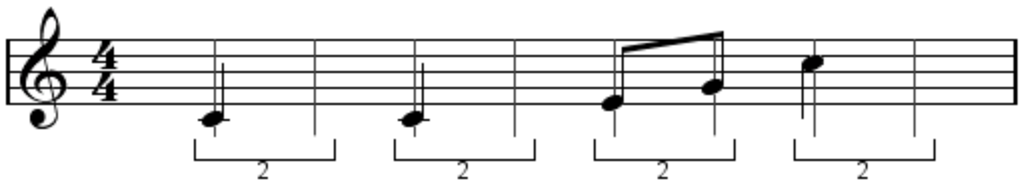
- If single RBMs learn features about data, DBMs learn features about features



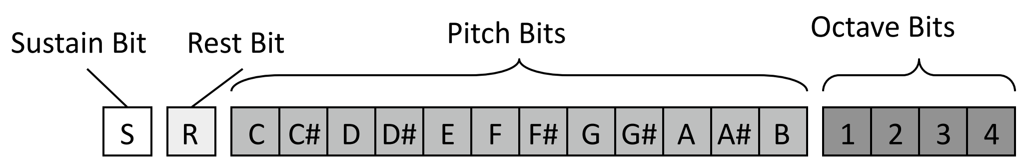
Encoding Scheme

- Requirements
  - Binary encoding
  - Music must be encoded in a string of standard length
  - Each note must be the same “distance” from every other note

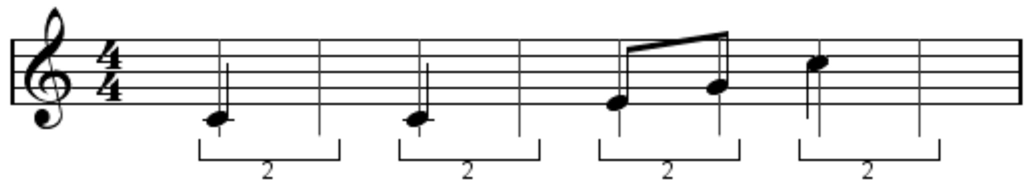
Encoding Scheme



- Break melody into beat subdivisions
- Each subdivision contains 18 bits
  - 1 Sustain Bit
  - 1 Rest Bit
  - 12 Pitch Bits
  - 4 Octave Bits



Encoding Scheme



Sustain Bit | Rest Bit | Pitch Bits | Octave Bits

S	R	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	1	2	3	4
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Chord Encoding Scheme

Chord Bits

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
CM	1	0	0	0	1	0	0	1	0	0	0	0
	1	0	0	0	1	0	0	1	0	0	0	0
FM	1	0	0	0	0	1	0	1	0	0	0	0
	1	0	0	0	0	1	0	1	0	0	0	0
GM	0	0	1	0	0	0	1	0	0	0	0	1
	0	0	1	0	0	0	1	0	0	0	0	1
CM	1	0	0	0	1	0	0	1	0	0	0	0
	1	0	0	0	1	0	0	1	0	0	0	0

Initial Dataset

- Children's songs
  - 2 measures
  - 8^{th} note resolution
  - Simple chords
  - 14 melodies
- Generated similar songs

Main Dataset

- Jazz licks
  - 4 measures
  - 12 beat subdivisions
    - (32^{nd} triplet note resolution)
  - ii-V^7-I-VI^7 chord progression
  - 100+ licks
    - Transcribed
    - Handwritten

Windowing

- Rather than training on entire piece at once, break music into “windows”

- Start with the first measure of a lick and gradually move window forward

- Allows learning/generating arbitrary length music sequences with a fixed size network

Windowing

Windowing

Windowing

Windowing

Windowing

Generating New Melodies

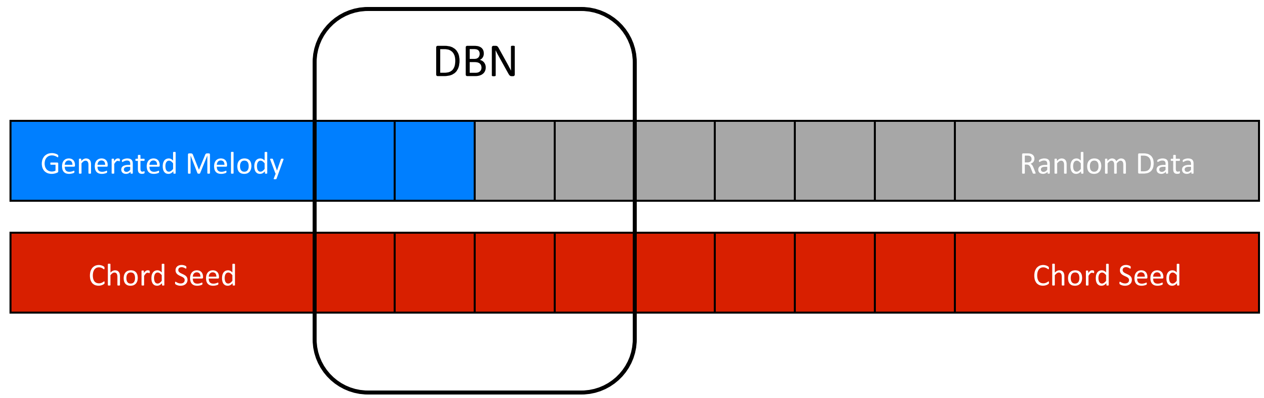
- Need to specify a chord sequence over which to generate a new melody.

- Chord bits are “clamped” during generation so that they can influence the melody being generated without changing themselves

Generating New Melodies

- Use a windowing strategy analogous to our windowed training method
- As each successive beat is generated the whole melody and chord sequence shifts forward to make room for the next beat

Generating New Melodies



Results

- Rhythmically stable
- Respects chord tones
- Occasional color tones
- Very few foreign tones

Results

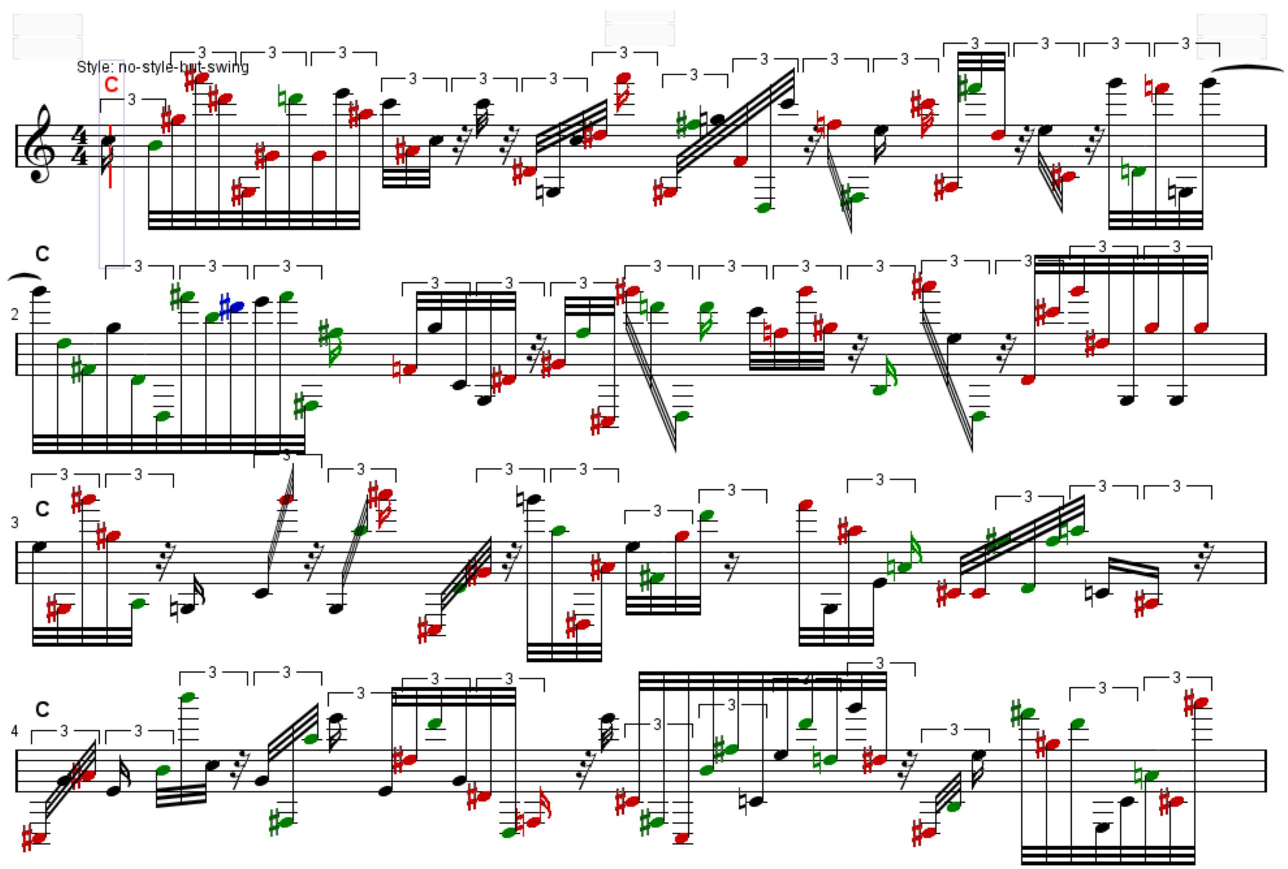
- Trained on transpositions as well
  - Generated music following key of given chord progression
  - Succeeded with up to four transpositions

Example Training Licks

Example Generated Licks

More Generated Examples

Random Music



Future Work – Repeated Notes

- Our machines produced disproportionate numbers of repeated notes

- Can sound static or too immobile for jazz

Future Work – Repeated Notes

- Possible solution: post processing
  - Merge repeated notes together
  - Results in a smoother output, but starts to cross line of unsupervised learning

- Ideally, machine should avoid repeated notes in the first place

Future Work – Training Algorithm

- Slow
  - Optimization
  - Parallelization
  - Adaptive termination

- Sensitive to training presentation order
  - Randomize training inputs

Future Work – Chord Inference

- We believe our work naturally lends itself to the open problem of inferring unknown chords for a melody
  - Currently we provide a chord seed to generate a melody.
  - If we instead provide a melody as input, we could determine which chords fit that melody

Conclusion

- Unsupervised learning algorithm
- Based on probabilistic neural network theory
- Able to create novel jazz licks based on an existing corpus
- Minimal assumptions about musical knowledge