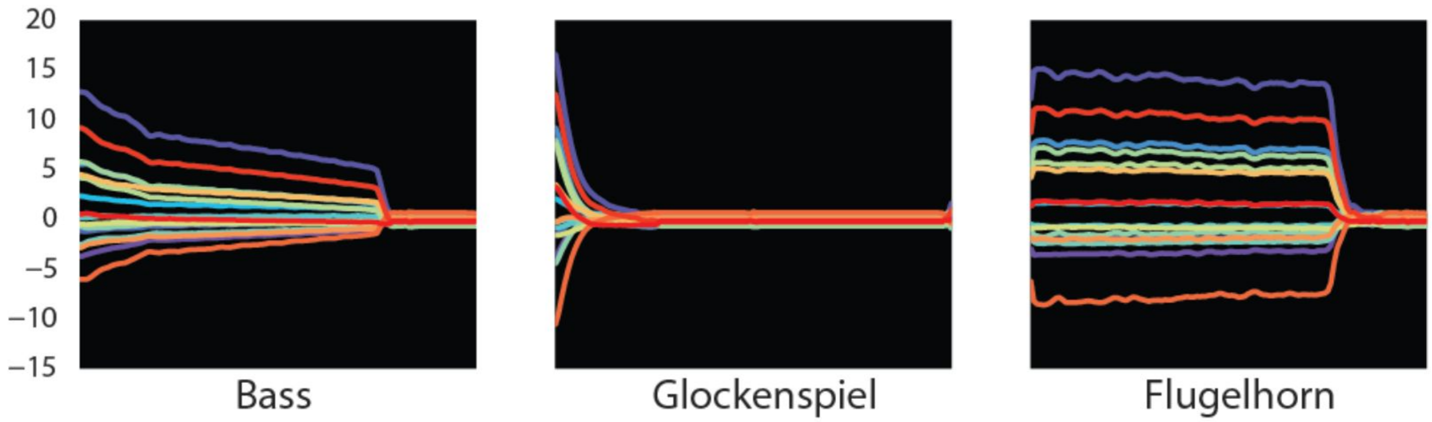
WaveNet Encoding Analysis

Milo Knowles

WaveNet Recap

- Autoencoder Network learns to encode audio as a small vector
- WaveNet uses one **16 element vector for every 32ms of audio!**

Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders



How good is the audio reconstruction?

Can ~64 embeddings x 16 elements really represent 2 sec of audio?

- Example 1: English Horn - C5
- Example 2: Banjo - G
- Example 3: Guitar - D major chord
- Example 4: Flugel Horn - C4

In summary:

- pretty good for monophonic wind instruments
- bad for chords and string instruments

Research Questions

What does each component of the embedding represent?

How do numerical modifications to the embeddings affect qualitative properties of the sound?

What is the neural network learning?

Experiment #1: Setting Components to Zero

- Set one component to zero while keeping the others constant
- What information do we lose as a result?

Example 1: English Horn

0: F6 (4th two octaves up), releases up to G6
1: static
2: B3 (7th), release plays a C#-A#-F# descending arpeggio
3: static
4: Eflat, release on Dflat
5: loud root
6: horrible distortion
7: Eflat, release on Dflat (octave down)
8: static
9: tuned static
10: E4 major third
11: static
12: D4 with screeching overtones
13: static
14: really high pitched noise
15: D4 hold, release on C

Example 2: Banjo

0: F#
1: clipping
2: F#, F low distorted → also a major 7th!
3: noise
4: E octave up, quick whistle down
5: C#
6: low distortion twang
7: high whistling
8: quiet noise
9: noise
10: G#
11: noise
12: D
13: noise
14: high screeching
15: D, release on G → also releases on the root

Experiment 1 Conclusion

- Setting things to zero seems to **remove** certain harmonics and bring out others
- Zeroing some components just creates noise or distortion
- Not much tonal consistency between the english horn and banjo results

Experiment 2: Increase Gain on Each Component

- Multiple the magnitude of a single component by 2x (keeping others constant)
- Are any characteristics **increased**?

Example 1: English Horn

0: Breathy, barely audible overtones
1: Static
2: 1 octave above root
3: F# (the #4)
4: both an octave below and above
5: 5th, but with vibrato
6: root
7: really breathy octave above
8: noise
9: higher noise
10: vibrato root
11: higher octave, like a train whistle
12: two octaves up, really unstable , like tea kettle
13: noise
14: higher noise
15: screeching octave up

Example 2: Voice Lead Synth

0: Noise
1: crackling noise
2: weak root
3: F#
4: low octave
5: 5th
6: root
7: electrical octave above
8: noise
9: higher noise
10: vibrato root
11: higher octave, like train whistle
12: octave up, really unstable note
13: noise
14: higher noise
15: screeching unstable octave up

These are exactly the same as the English horn! But they’re similar instruments playing the same pitch...

Example 3: Banjo (G3)

Banjo

0: noise
1: crackling noise
2: really high root
3: F# →no longer a #4, but the **same pitch**
4: low C → no longer a root, but the **same pitch**
5: G# ???
6: G, release to G#
7: Out of tune G
8: noise
9: higher noise
10: vibrato high root
11: weird root
12: whistling, unstable
13: noise
14: higher noise
15: screeching

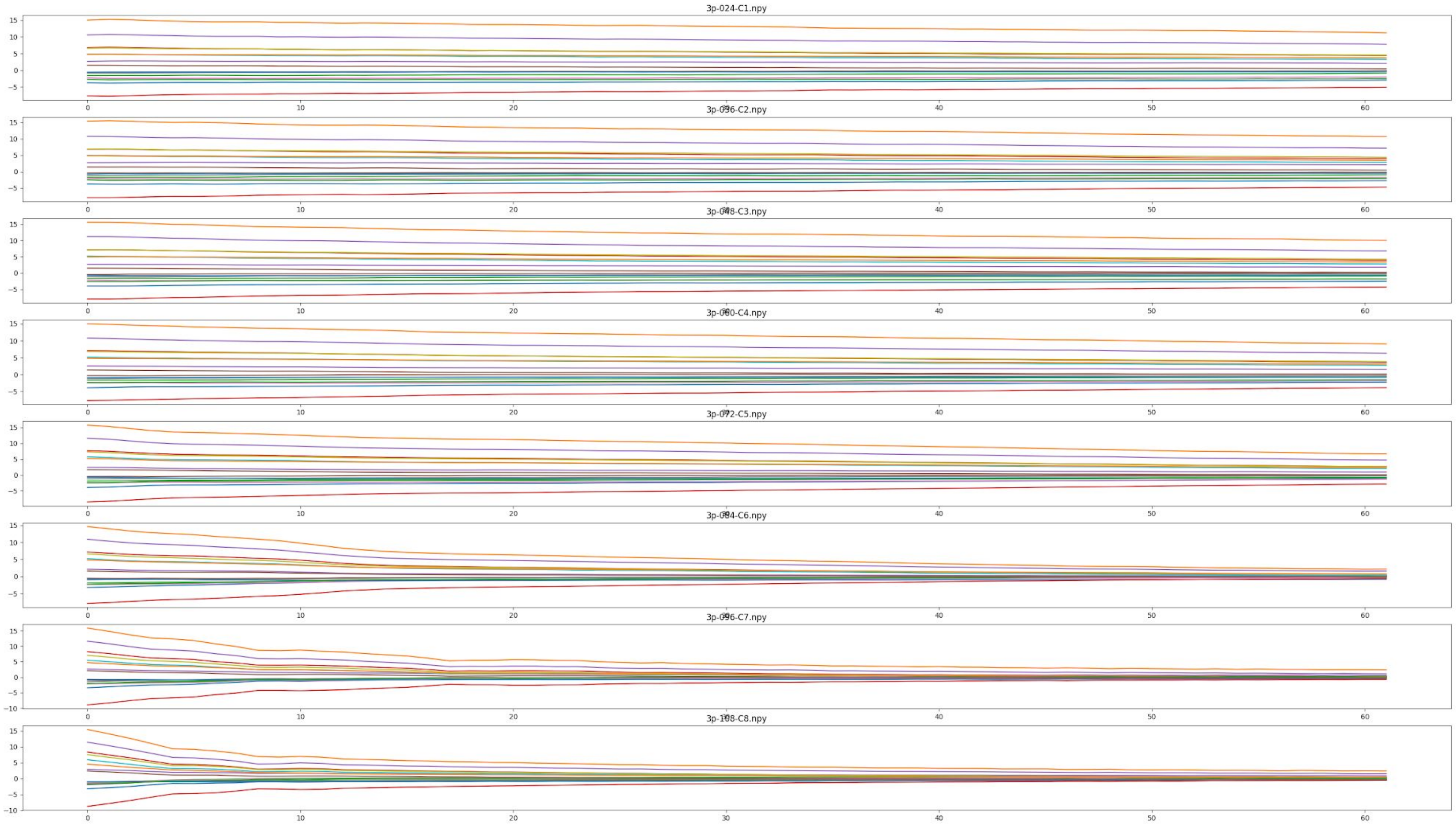
Voice Lead Synth (previous slide)

0: Noise
1: crackling noise
2: weak root
3: F# (the #4 of C)
4: low octave (C)
5: 5th (G)
6: root
7: electrical octave above
8: noise
9: higher noise
10: vibrato root
11: higher octave, like train whistle
12: octave up, really unstable note
13: noise
14: higher noise
15: screeching unstable octave up

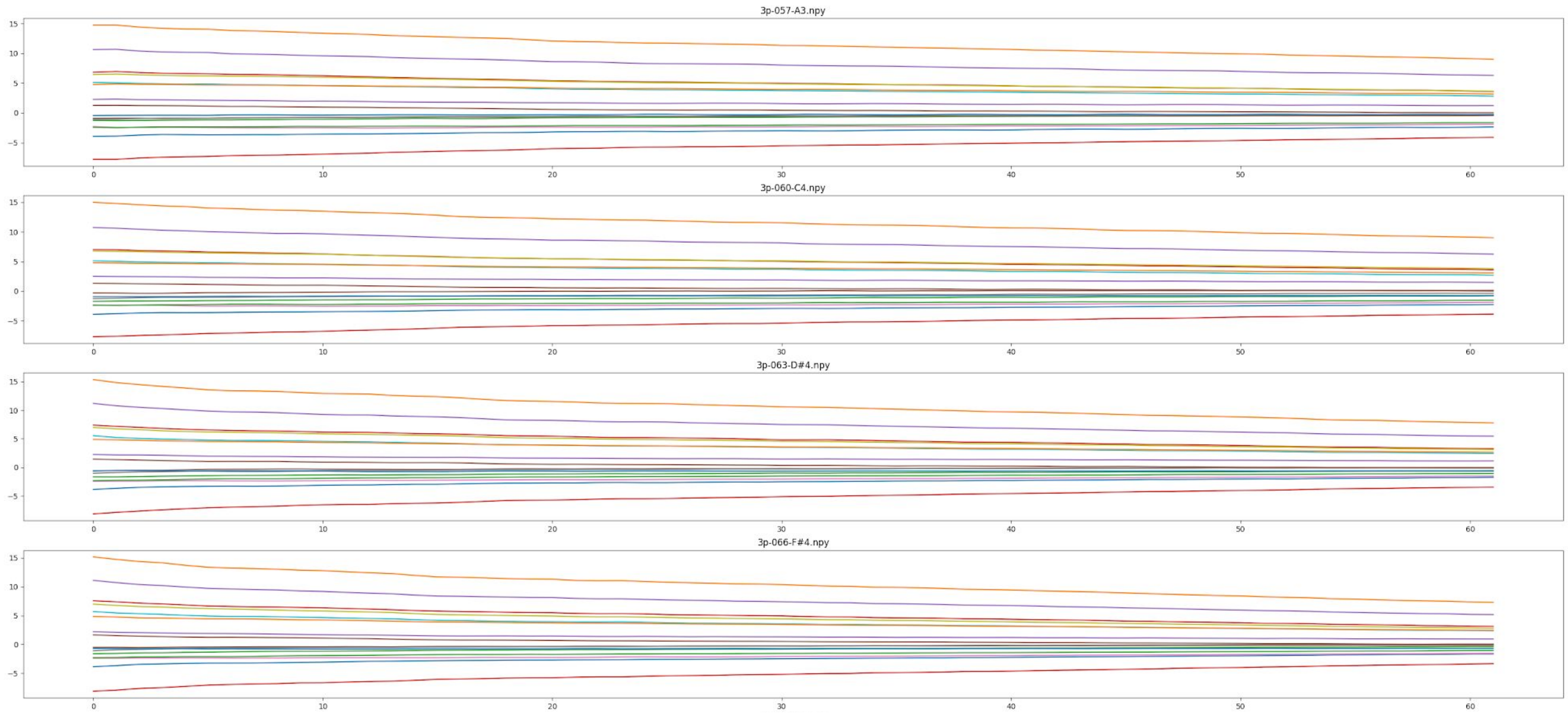
Experiment 2 Conclusion

- the same components tend to create noise
- some components control **relative harmonics**
- other components control **absolute pitches**
- Pitch is probably encoded as a nonlinear function of many components, and would be hard to isolate

Sidenote: Octaves have the same ordering



Sidenote: Even different pitches have same ordering



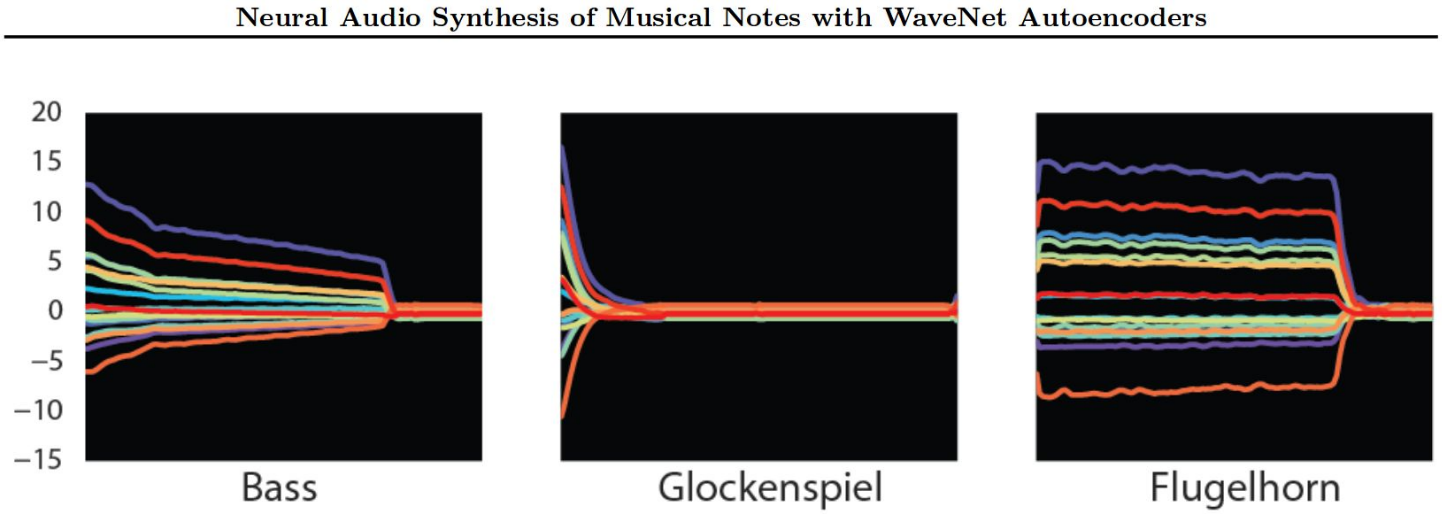
Note: I synthesized audio from these embeddings, and they do have the correct pitch!

Experiment 3: Sign Flip (failed)

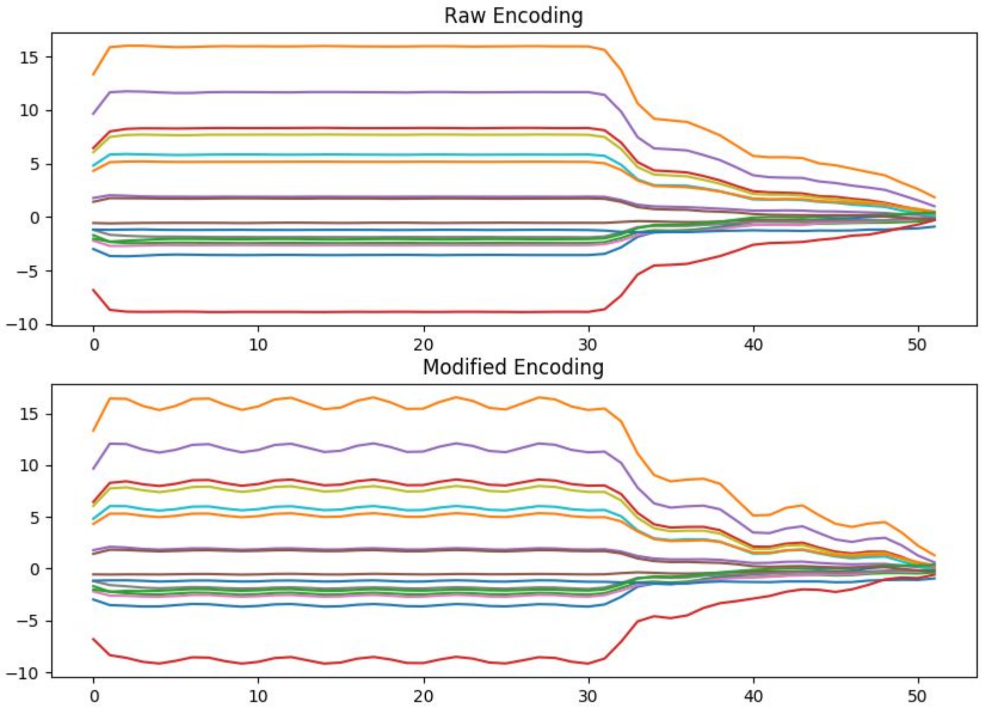
- Reverse the sign of one component, while keeping others the same
- results are really distorted
- many more of the components produce noise
- no shared characteristics with the gain analysis from the previous experiment

Experiment 4: Vibrato

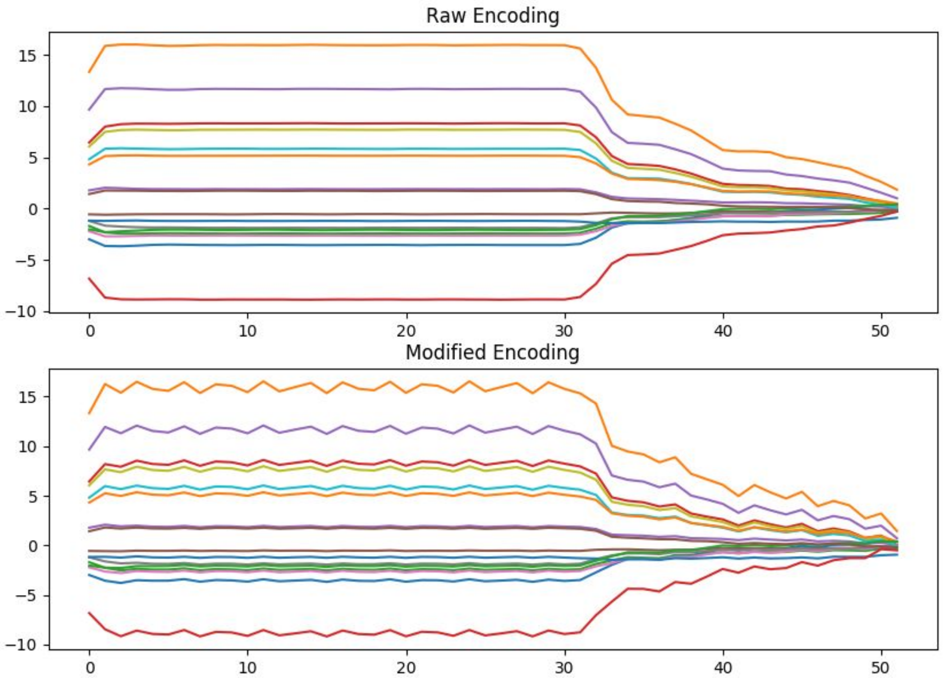
- Add an LFO to all of the components



Example 1: English Horn (10Hz Vibrato)



Example 2: English Horn (20Hz Vibrato)



Conclusions

- Difficult to interpret the embeddings
- The neural network is learning an optimal way to compress audio - there’s no reason why it should make sense to us