A Variational Parametric Model for Audio Synthesis

Krishna Subramani
Supervised by Prof. Preeti Rao

Electrical Engineering
Indian Institute of Technology Bombay, India

Audio Synthesis?

- What comes to your mind when you hear 'Audio Synthesis'?

Audio Synthesis?

Figure: One of the early Moog Modular Synthesizers

Audio Synthesis?

- More generally, it involves us specifying controlling parameters to a synthesizer to obtain an audio output
- What are the main parameters which govern the audio generation(at a high level)?

Audio Synthesis?

1. Timbre
2. Pitch
3. Loudness

Audio Synthesis?

- Early analog synthesizers used voltage controlled oscillators, filters, amplifiers to generate the waveform, and ‘envelope generators’ to shape it

Audio Synthesis?

- Early analog synthesizers used voltage controlled oscillators, filters, amplifiers to generate the waveform, and ‘envelope generators’ to shape it
- Data-driven statistical modeling + computing power
  $\implies$ Deep Learning for audio synthesis!

Generative Models for Audio Synthesis

- Rely on ability of algorithms to extract musically relevant information from vast amounts of data
- Autoregressive modeling, Generative Adversarial Networks and Variational Autoencoders are some of the proposed generative modeling methods
- Most methods in literature try to model audio signals directly in the time or frequency domain

Our Nearest Neighbours

- [Sarroff and Casey, 2014] first to use autoencoders to perform frame-wise reconstruction of short-time magnitude spectra

Our Nearest Neighbours

- [Sarroff and Casey, 2014] first to use autoencoders to perform frame-wise reconstruction of short-time magnitude spectra
  - ✓ Learn perceptually relevant lower dimensional representations('latent space') of audio to help in synthesis

Our Nearest Neighbours

- [Sarroff and Casey, 2014] first to use autoencoders to perform frame-wise reconstruction of short-time magnitude spectra
  - ✓ Learn perceptually relevant lower dimensional representations('latent space') of audio to help in synthesis
  - × 'Graininess' in the reconstructed sound

Our Nearest Neighbours

- [Sarroff and Casey, 2014] first to use autoencoders to perform frame-wise reconstruction of short-time magnitude spectra
  - ✓ Learn perceptually relevant lower dimensional representations('latent space') of audio to help in synthesis
  - × 'Graininess' in the reconstructed sound
- [Roche et al., 2018] extended this analysis to try out different autoencoder architectures

Our Nearest Neighbours

- [Sarroff and Casey, 2014] first to use autoencoders to perform frame-wise reconstruction of short-time magnitude spectra
  - ✓ Learn perceptually relevant lower dimensional representations('latent space') of audio to help in synthesis
  - × 'Graininess' in the reconstructed sound
- [Roche et al., 2018] extended this analysis to try out different autoencoder architectures
  - ✓ Helped by the release of NSynth [Engel et al., 2017]

Our Nearest Neighbours

- [Sarroff and Casey, 2014] first to use autoencoders to perform frame-wise reconstruction of short-time magnitude spectra
  - ✓ Learn perceptually relevant lower dimensional representations('latent space') of audio to help in synthesis
  - × 'Graininess' in the reconstructed sound
- [Roche et al., 2018] extended this analysis to try out different autoencoder architectures
  - ✓ Helped by the release of NSynth [Engel et al., 2017]
  - ✓ Usability of the latent space in audio interpolation

Our Nearest Neighbours

- [Sarroff and Casey, 2014] first to use autoencoders to perform frame-wise reconstruction of short-time magnitude spectra
  - ✓ Learn perceptually relevant lower dimensional representations('latent space') of audio to help in synthesis
  - × 'Graininess' in the reconstructed sound
- [Roche et al., 2018] extended this analysis to try out different autoencoder architectures
  - ✓ Helped by the release of NSynth [Engel et al., 2017]
  - ✓ Usability of the latent space in audio interpolation
- [Esling et al., 2018] regularized the VAE latent space in order to effect control over perceptual timbre of synthesized instruments

Our Nearest Neighbours

- [Sarroff and Casey, 2014] first to use autoencoders to perform frame-wise reconstruction of short-time magnitude spectra
  - ✓ Learn perceptually relevant lower dimensional representations(‘latent space’) of audio to help in synthesis
  - × ‘Graininess’ in the reconstructed sound
- [Roche et al., 2018] extended this analysis to try out different autoencoder architectures
  - ✓ Helped by the release of NSynth [Engel et al., 2017]
  - ✓ Usability of the latent space in audio interpolation
- [Esling et al., 2018] regularized the VAE latent space in order to effect control over perceptual timbre of synthesized instruments
  - ✓ Enabled them to ‘generate’ and ‘interpolate’ audio in this space

Our Nearest Neighbours

- Major issue with previous methods was frame-wise analysis-synthesis based reconstruction which fails to model temporal evolution

Our Nearest Neighbours

- Major issue with previous methods was frame-wise analysis-synthesis based reconstruction which fails to model temporal evolution
- [Engel et al., 2017] inspired by Wavenets [Oord et al., 2016] autoregressive modeling capablities for speech extended it to musical instrument synthesis

Our Nearest Neighbours

- Major issue with previous methods was frame-wise analysis-synthesis based reconstruction which fails to model temporal evolution
- [Engel et al., 2017] inspired by Wavenets [Oord et al., 2016] autoregressive modeling capablities for speech extended it to musical instrument synthesis
  - ✓ Generation of realistic and creative sounds

Our Nearest Neighbours

- Major issue with previous methods was frame-wise analysis-synthesis based reconstruction which fails to model temporal evolution
- [Engel et al., 2017] inspired by Wavenets [Oord et al., 2016] autoregressive modeling capablities for speech extended it to musical instrument synthesis
  - ✓ Generation of realistic and creative sounds
  - × Complex network which was difficult to train and sample from

Our Nearest Neighbours

- Major issue with previous methods was frame-wise analysis-synthesis based reconstruction which fails to model temporal evolution
- [Engel et al., 2017] inspired by Wavenets [Oord et al., 2016] autoregressive modeling capablities for speech extended it to musical instrument synthesis
  - ✓ Generation of realistic and creative sounds
  - × Complex network which was difficult to train and sample from
- [Wyse, 2018] also autoregressively modelled the audio, albeit by conditioning the waveform samples on additional parameters like pitch, velocity(loudness) and instrument class

Our Nearest Neighbours

- Major issue with previous methods was frame-wise analysis-synthesis based reconstruction which fails to model temporal evolution
- [Engel et al., 2017] inspired by Wavenets [Oord et al., 2016] autoregressive modeling capablities for speech extended it to musical instrument synthesis
  - ✓ Generation of realistic and creative sounds
  - × Complex network which was difficult to train and sample from
- [Wyse, 2018] also autoregressively modelled the audio, albeit by conditioning the waveform samples on additional parameters like pitch, velocity(loudness) and instrument class
  - ✓ Was able to achieve better control over generation by conditioning

Our Nearest Neighbours

- Major issue with previous methods was frame-wise analysis-synthesis based reconstruction which fails to model temporal evolution
- [Engel et al., 2017] inspired by Wavenets [Oord et al., 2016] autoregressive modeling capablities for speech extended it to musical instrument synthesis
  - ✓ Generation of realistic and creative sounds
  - × Complex network which was difficult to train and sample from
- [Wyse, 2018] also autoregressively modelled the audio, albeit by conditioning the waveform samples on additional parameters like pitch, velocity(loudness) and instrument class
  - ✓ Was able to achieve better control over generation by conditioning
  - × Unable to generalize to untrained pitches

Why Parametric?

- Rather than generating new timbres('interpolating' across sounds), we consider the problem of synthesis of a given instrument sound with flexible control over the pitch and loudness dynamics
- Pitch shifting without timbre modification uses a source-filter model with the filter(spectral envelope) being kept constant [Roebel and Rodet, 2005]
- A powerful parametric representation over raw waveform or spectrogram has the potential to achieve high quality with less training data

Why Parametric?

- Rather than generating new timbres('interpolating' across sounds), we consider the problem of synthesis of a given instrument sound with flexible control over the pitch and loudness dynamics
- Pitch shifting without timbre modification uses a source-filter model with the filter(spectral envelope) being kept constant [Roebel and Rodet, 2005]
- A powerful parametric representation over raw waveform or spectrogram has the potential to achieve high quality with less training data
    1. [Blaauw and Bonada, 2016] recognized this in context of speech synthesis and used a vocoder representation to train a generative model, achieving promising results along the way

Dataset

- Good-sounds dataset [Romani Picas et al., 2015], consisting of individual note and scale recordings for 12 different instruments
- We work with the ‘Violin’ played in mezzo-forte loudness, and choose the 4^{th} octave(MIDI 60-71)

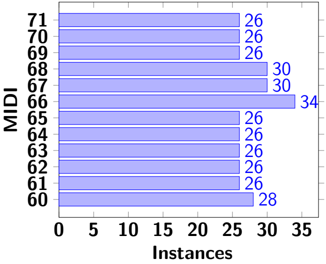


Figure: Instances per note in the overall dataset

- Average note duration for the chosen octave is about 4.5s per note

Dataset

Why we chose Violin?

Popular in Indian Music, Human voice-like timbre

Ability to produce continuous pitch!

- Go … sisting of inc … truments
- We work with the 'Violin' played in mezzo-forte loudness, and choose the 4^{th} octave(MIDI 60-71)

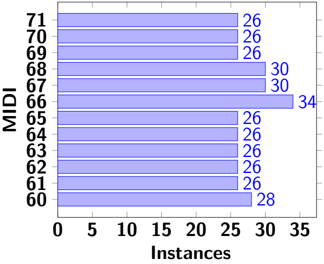


Figure: Instances per note in the overall dataset

- Average note duration for the chosen octave is about 4.5s per note

Dataset

- We split the data to train(80%) and test(20%) instances across MIDI note labels
- Our model is trained with frames(duration 21.3ms) from the train instances, and system performance is evaluated with frames from the test instances.

Non-Parametric Reconstruction

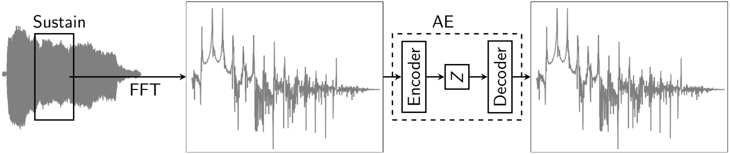
- Framewise magnitude spectra reconstruction procedure in [Roche et al., 2018] cannot generalize to untrained pitches

Non-Parametric Reconstruction

- Framewise magnitude spectra reconstruction procedure in [Roche et al., 2018] cannot generalize to untrained pitches
- We consider two cases - Train including and excluding MIDI 63, along with its 3 neighbouring pitches on either side

Non-Parametric Reconstruction

- Framewise magnitude spectra reconstruction procedure in [Roche et al., 2018] cannot generalize to untrained pitches
- We consider two cases - Train including and excluding MIDI 63, along with its 3 neighbouring pitches on either side



- Repeat above across all input spectral frames, and invert obtained spectrogram using Griffin-Lim [Griffin and Lim, 1984]

Non-Parametric Reconstruction

Figure: Input MIDI 63, 1 [1]

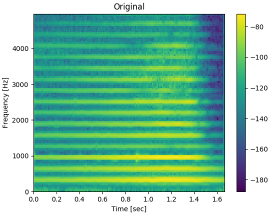


Figure: Including MIDI 63, 2 [2]

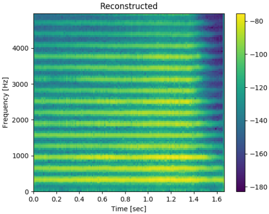
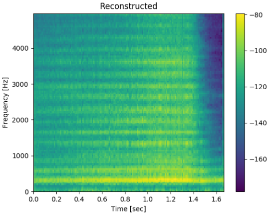


Figure: Excluding MIDI 63, 3 [3]

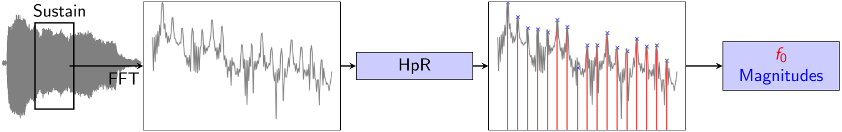


Parametric Model

1. Frame-wise magnitude spectrum $\rightarrow$ harmonic representation using Harmonic plus Residual(HpR) model [Serra et al., 1997](currently, we neglect the residual)

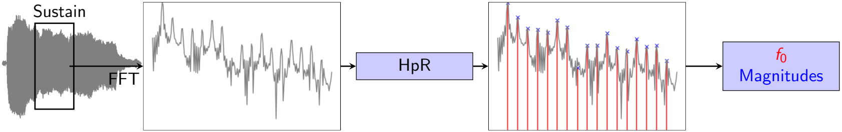
Parametric Model

1. Frame-wise magnitude spectrum → harmonic representation using Harmonic plus Residual(HpR) model [Serra et al., 1997](currently, we neglect the residual)



Parametric Model

1. Frame-wise magnitude spectrum → harmonic representation using Harmonic plus Residual(HpR) model [Serra et al., 1997](currently, we neglect the residual)



- Output of HpR block $\implies$ log-dB magnitudes + harmonics

Parametric Model

2. log-dB magnitudes + harmonics $\rightarrow$ TAE algorithm
   [Roebel and Rodet, 2005, IMAI, 1979]

Parametric Model

2. log-dB magnitudes + harmonics → TAE algorithm [Roebel and Rodet, 2005, IMAI, 1979]

- TAE $\implies$ Iterative Cepstral Liftering

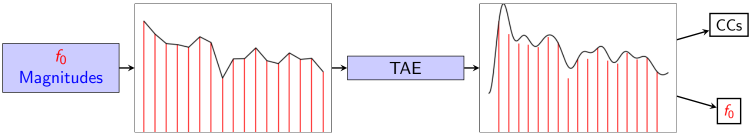
$$
\begin{aligned}
&A_0(k) = log(|X(k)|), V_0 = -\infty \\
&while(A_i - A_0 < \Delta)\{ \\
&A_i(k) = max(A_{i-1}(k), V_{i-1}(k)) \\
&V_i = FFT(lifter(A_i)) \\
&\}
\end{aligned}
$$

Parametric Model

2. log-dB magnitudes + harmonics → TAE algorithm [Roebel and Rodet, 2005, IMAI, 1979]

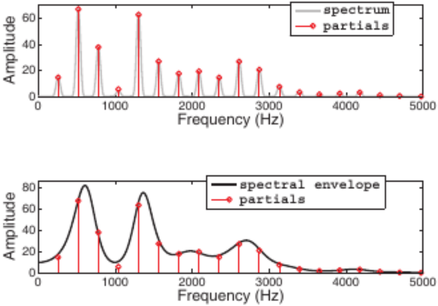
- TAE $\implies$ Iterative Cepstral Liftering

$$
\begin{array}{l}
A_0(k) = log(|X(k)|), V_0 = -\infty \\
while(A_i - A_0 < \Delta)\{ \\
A_i(k) = max(A_{i-1}(k), V_{i-1}(k)) \\
V_i = FFT(lifter(A_i)) \\
\}
\end{array}
$$



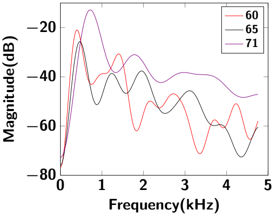
Parametric Model

- No open source implementation available for the TAE, thus we implemented it following procedure highlighted in [Roebel and Rodet, 2005, Caetano and Rodet, 2012]



- Figure shows a TAE snap from [Caetano and Rodet, 2012]
- Similar to results we get! 1 [4] 2 [5]

Parametric Model



- Spectral envelope shape varies across pitch
  1. Dependence of envelope on pitch [Slawson, 1981, Caetano and Rodet, 2012]
  2. Variation due the TAE algorithm
- TAE → smooth function to estimate harmonic amplitudes

Parametric Model

- No. of CCs(Cepstral Coefficients,K_{cc}) depends on Sampling rate(F_s),pitch(f_0)

$$K_{cc} \leq \frac{F_s}{2f_0}.$$

- For our network, we choose the maximum K_{cc}(lowest pitch) $=$ 91, and zero pad the high pitch K_{cc} to 91 dimension vectors

Generative Models

- Autoencoders [Hinton and Salakhutdinov, 2006] - Minimize the MSE(Mean Squared Error) between input and network reconstruction

Generative Models

- Autoencoders [Hinton and Salakhutdinov, 2006] - Minimize the MSE(Mean Squared Error) between input and network reconstruction
  - Simple to train, and performs good reconstruction(since they minimize MSE)

Generative Models

- Autoencoders [Hinton and Salakhutdinov, 2006] - Minimize the MSE(Mean Squared Error) between input and network reconstruction
  - Simple to train, and performs good reconstruction(since they minimize MSE)
  - Not truly a generative model, as you cannot generate new data

Generative Models

- Autoencoders [Hinton and Salakhutdinov, 2006] - Minimize the MSE(Mean Squared Error) between input and network reconstruction
  - Simple to train, and performs good reconstruction(since they minimize MSE)
  - Not truly a generative model, as you cannot generate new data
- Variational Autoencoders [Kingma and Welling, 2013] - Inspired from Variational Inference, enforce a prior on the latent space.

Generative Models

- Autoencoders [Hinton and Salakhutdinov, 2006] - Minimize the MSE(Mean Squared Error) between input and network reconstruction
  - Simple to train, and performs good reconstruction(since they minimize MSE)
  - Not truly a generative model, as you cannot generate new data
- Variational Autoencoders [Kingma and Welling, 2013] - Inspired from Variational Inference, enforce a prior on the latent space.
  - Truly generative as we can ‘generate’ new data by sampling from the prior

Generative Models

- Autoencoders [Hinton and Salakhutdinov, 2006] - Minimize the MSE(Mean Squared Error) between input and network reconstruction
  - Simple to train, and performs good reconstruction(since they minimize MSE)
  - Not truly a generative model, as you cannot generate new data
- Variational Autoencoders [Kingma and Welling, 2013] - Inspired from Variational Inference, enforce a prior on the latent space.
  - Truly generative as we can ‘generate’ new data by sampling from the prior
- Conditional Variational Autoencoders [Doersch, 2016, Sohn et al., 2015] - Same principle as a VAE, however learns the conditional distribution over an additional conditioning variable

Generative Models

- Why VAE over AE?

Generative Models

- Why VAE over AE?
  - Continuous latent space from which we can sample points(and synthesize the corresponding audio)

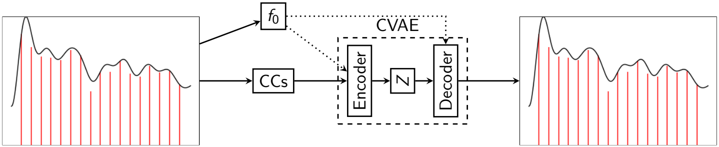
Generative Models

- Why VAE over AE?
  - Continuous latent space from which we can sample points(and synthesize the corresponding audio)
- Why CVAE over VAE?

Generative Models

- Why VAE over AE?
  - Continuous latent space from which we can sample points(and synthesize the corresponding audio)
- Why CVAE over VAE?
  - Conditioning on pitch $\implies$ Network captures dependencies between the timbre and the pitch $\implies$ More accurate envelope generation + Pitch control

Network Architecture



- Network input is CCs $\rightarrow$ MSE represents perceptually relevant distance in terms of squared error between the input and reconstructed log magnitude spectral envelopes
- We train the network on frames from the train instances
- For evaluation, MSE values calculated ahead is the average reconstruction error across all the test instance frames

Network Architecture

- Main hyperparameters -
  1. β - Controls relative weighting between reconstruction and prior enforcement

Network Architecture

$L \propto \text{MSE} + \beta.\text{KLD}$

- Main hyperparameters -

  1. β - Controls relative weighting between reconstruction and prior enforcement

Network Architecture

- Main hyperparameters -
  1. β - Controls relative weighting between reconstruction and prior enforcement

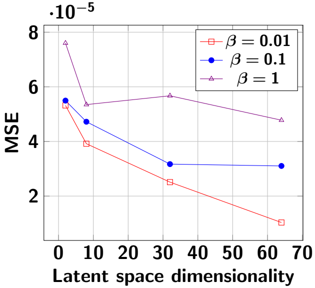


Figure: CVAE, varying β

Network Architecture

- Main hyperparameters -
  1. β - Controls relative weighting between reconstruction and prior enforcement

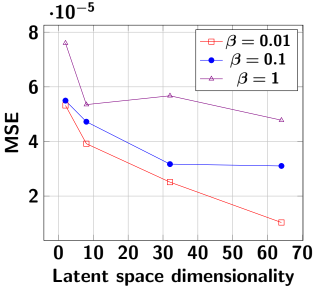


Figure: CVAE, varying β

- Tradeoff between both terms, choose $\beta = 0.1$

Network Architecture

- Main hyperparameters -
  2. Dimensionality of latent space - networks reconstruction ability

Network Architecture

- Main hyperparameters -
  2. Dimensionality of latent space - networks reconstruction ability

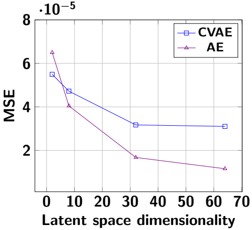


Figure: CVAE($\beta = 0.1$) vs AE

Network Architecture

- Main hyperparameters -
  2. Dimensionality of latent space - networks reconstruction ability

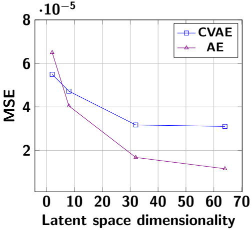


Figure: CVAE($\beta = 0.1$) vs AE

- Steep fall initially, flatter later. Choose dimensionality $= 32$

Network Architecture

- Network size : $[91, 91, 32, 91, 91]$
  - 91 is the dimension of input CCs, 32 is latent space dimensionality
- Linear Fully Connected Layers + Leaky ReLU activations
- ADAM [Kingma and Ba, 2014] with initial lr $= 10^{-3}$
- Training for 2000 epochs with batch size 512

Experiments

- Two kinds of experiments to demonstrate networks capabilities

Experiments

- Two kinds of experiments to demonstrate networks capabilities
    1. Reconstruction - Omit pitch instances during training and see how well model reconstructs notes of omitted target pitch

Experiments

- Two kinds of experiments to demonstrate networks capabilities
  1. Reconstruction - Omit pitch instances during training and see how well model reconstructs notes of omitted target pitch
  2. Generation - How well model 'synthesizes' note instances with new unseen pitches

Reconstruction

- Two training contexts -

Reconstruction

- Two training contexts -
  1. T($\times$) is target, $\checkmark$ is training instances

MIDI	T - 3	T - 2	T - 1	T	T + 1	T + 2	T + 3
Kept	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$

Reconstruction

- Two training contexts -
  1. T($\times$) is target, $\checkmark$ is training instances

MIDI	T - 3	T - 2	T - 1	T	T + 1	T + 2	T + 3
Kept	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$

  2. Octave endpoints

MIDI	60	61	62	63	64	65
Kept	$\checkmark$	$\times$	$\times$	$\times$	$\times$	$\times$
MIDI	66	67	68	69	70	71
Kept	$\times$	$\times$	$\times$	$\times$	$\times$	$\checkmark$

Reconstruction

- Two training contexts -
  1. T($\times$) is target, $\checkmark$ is training instances

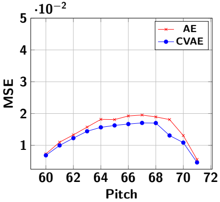
MIDI	T - 3	T - 2	T - 1	T	T + 1	T + 2	T + 3
Kept	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$

  2. Octave endpoints

MIDI	60	61	62	63	64	65
Kept	$\checkmark$	$\times$	$\times$	$\times$	$\times$	$\times$
MIDI	66	67	68	69	70	71
Kept	$\times$	$\times$	$\times$	$\times$	$\times$	$\checkmark$

- In each of the above cases, we compute the MSE as the frame-wise spectral envelope match across all frames of all the target instances.

Reconstruction



- Both AE and CVAE reasonably reconstruct the target pitch 1[6] 2[7] 3[8]
- CVAE produces better reconstruction, especially when the target pitch is far from the pitches available in the training data(plot above) 4[9] 5[10] 6[11]
- Conditioning helps to capture the pitch dependency of the spectral envelope more accurately

Reconstruction

- To emulate the effect of pitch conditioning with an AE, we train the AE by appending the pitch to the input CCs and reconstructing this appended input as shown below

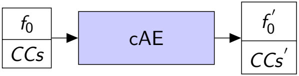
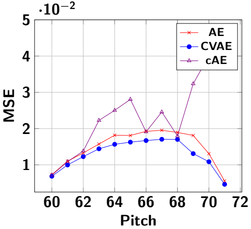


Figure: 'Conditional' AE(cAE)

- [Wyse, 2018] followed a similar approach of appending the conditional variables to the input of his model
- the 'cAE' is comparable to our proposed CVAE in that the network might potentially learn something from f_0

Reconstruction



- We train the cAE on the octave endpoints and evaluate performance as shown above
- Appending f_0 does not improve performance. It is infact worse than AE!

Generation

- We are interested in 'synthesizing' audio!
- We see how well the network can generate an instance of a desired pitch(which the network has not been trained on)

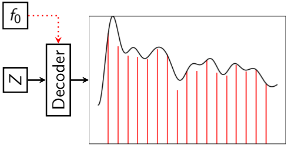


Figure: Sampling from the Network

- We follow the procedure in [Blaauw and Bonada, 2016] - a random walk to sample points coherently from the latent space

Generation

- We train on instances across the entire octave sans MIDI 65, and then generate MIDI 65
- On listening, we can see that the generated audio still lacks the soft noisy sound of the violin bowing
  1 [12] 2 [13] 3 [14]
- For a more realistic synthesis, we generate a violin note with vibrato 4 [15]

Putting it all together

- We explored autoencoder frameworks in generative models for audio synthesis of instrumental tones

Putting it all together

- We explored autoencoder frameworks in generative models for audio synthesis of instrumental tones
- Our parametric representation decouples ‘timbre’ and ‘pitch’, thus relying on the network to model the inter-dependencies

Putting it all together

- We explored autoencoder frameworks in generative models for audio synthesis of instrumental tones
- Our parametric representation decouples 'timbre' and 'pitch', thus relying on the network to model the inter-dependencies
- Pitch conditioning allows to generate the learnt spectral envelope for that pitch, thus enabling us to vary the pitch contour continuously

Putting it all together

- Our model is still far from perfect however, and we have to improve it further

Putting it all together

- Our model is still far from perfect however, and we have to improve it further
    1. An immediate thing to do will be to model the residual of the input audio. This will help in making the generated audio sound more natural

Putting it all together

- Our model is still far from perfect however, and we have to improve it further
    1. An immediate thing to do will be to model the residual of the input audio. This will help in making the generated audio sound more natural
    2. We have not taken into account dynamics. A more complete system will involve capturing spectral envelope(timbral) dependencies on both pitch and loudness dynamics

Putting it all together

- Our model is still far from perfect however, and we have to improve it further
  1. An immediate thing to do will be to model the residual of the input audio. This will help in making the generated audio sound more natural
  2. We have not taken into account dynamics. A more complete system will involve capturing spectral envelope(timbral) dependencies on both pitch and loudness dynamics
  3. Conducting more formal listening tests involving synthesis of larger pitch movements or melodic elements from Indian Classical music

Putting it all together

- Our model is still far from perfect however, and we have to improve it further
  1. An immediate thing to do will be to model the residual of the input audio. This will help in making the generated audio sound more natural
  2. We have not taken into account dynamics. A more complete system will involve capturing spectral envelope(timbral) dependencies on both pitch and loudness dynamics
  3. Conducting more formal listening tests involving synthesis of larger pitch movements or melodic elements from Indian Classical music
- Our contributions

Putting it all together

- Our model is still far from perfect however, and we have to improve it further
  1. An immediate thing to do will be to model the residual of the input audio. This will help in making the generated audio sound more natural
  2. We have not taken into account dynamics. A more complete system will involve capturing spectral envelope(timbral) dependencies on both pitch and loudness dynamics
  3. Conducting more formal listening tests involving synthesis of larger pitch movements or melodic elements from Indian Classical music
- Our contributions
  1. To the best of our knowledge, we have not come across any work using a parametric model for musical tones in the neural synthesis framework, especially exploiting the conditioning function of the CVAE!

Putting it all together

- Our model is still far from perfect however, and we have to improve it further
  1. An immediate thing to do will be to model the residual of the input audio. This will help in making the generated audio sound more natural
  2. We have not taken into account dynamics. A more complete system will involve capturing spectral envelope(timbral) dependencies on both pitch and loudness dynamics
  3. Conducting more formal listening tests involving synthesis of larger pitch movements or melodic elements from Indian Classical music
- Our contributions
  1. To the best of our knowledge, we have not come across any work using a parametric model for musical tones in the neural synthesis framework, especially exploiting the conditioning function of the CVAE!
  2. The TAE method has no known PYTHON implementation, so we plan to make our code open-source to the MIR community for research

References I

[Blaauw and Bonada, 2016] Blaauw, M. and Bonada, J. (2016).
Modeling and transforming speech using variational autoencoders.
In *Interspeech*, pages 1770–1774.

[Caetano and Rodet, 2012] Caetano, M. and Rodet, X. (2012).
A source-filter model for musical instrument sound transformation.
In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 137–140. IEEE.

[Doersch, 2016] Doersch, C. (2016).
Tutorial on variational autoencoders.
arXiv preprint arXiv:1606.05908.

[Engel et al., 2017] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D., and Simonyan, K. (2017).
Neural audio synthesis of musical notes with wavenet autoencoders.
In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1068–1077. JMLR. org.

[Esling et al., 2018] Esling, P., Bitton, A., et al. (2018).
Generative timbre spaces: regularizing variational auto-encoders with perceptual metrics.
arXiv preprint arXiv:1805.08501.

References II

[Griffin and Lim, 1984] Griffin, D. and Lim, J. (1984).
Signal estimation from modified short-time fourier transform.
IEEE Transactions on Acoustics, Speech, and Signal Processing, 32(2):236–243.

[Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R. (2006).
Reducing the dimensionality of data with neural networks.
science, 313(5786):504–507.

[IMAI, 1979] IMAI, S. (1979).
Spectral envelope extraction by improved cepstrum.
IEICE, 62:217–228.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014).
Adam: A method for stochastic optimization.
arXiv preprint arXiv:1412.6980.

[Kingma and Welling, 2013] Kingma, D. P. and Welling, M. (2013).
Auto-encoding variational bayes.
arXiv preprint arXiv:1312.6114.

[Oord et al., 2016] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016).
Wavenet: A generative model for raw audio.
arXiv preprint arXiv:1609.03499.

References III

[Roche et al., 2018] Roche, F., Hueber, T., Limier, S., and Girin, L. (2018).
Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models.
arXiv preprint arXiv:1806.04096.

[Roebel and Rodet, 2005] Roebel, A. and Rodet, X. (2005).
Efficient Spectral Envelope Estimation and its application to pitch shifting and envelope preservation.
In *International Conference on Digital Audio Effects*, pages 30–35, Madrid, Spain. cote interne IRCAM: Roebel05b.

[Romani Picas et al., 2015] Romani Picas, O., Parra Rodriguez, H., Dabiri, D., Tokuda, H., Hariya, W., Oishi, K., and Serra, X. (2015).
A real-time system for measuring sound goodness in instrumental sounds.
In *Audio Engineering Society Convention 138*. Audio Engineering Society.

[Sarroff and Casey, 2014] Sarroff, A. M. and Casey, M. A. (2014).
Musical audio synthesis using autoencoding neural nets.
In *ICMC*.

[Serra et al., 1997] Serra, X. et al. (1997).
Musical sound modeling with sinusoids plus noise.
Musical signal processing, pages 91–122.

References IV

[Slawson, 1981] Slawson, W. (1981).
The color of sound: a theoretical study in musical timbre.
Music Theory Spectrum, 3:132–141.

[Sohn et al., 2015] Sohn, K., Lee, H., and Yan, X. (2015).
Learning structured output representation using deep conditional generative models.
In *Advances in neural information processing systems*, pages 3483–3491.

[Wyse, 2018] Wyse, L. (2018).
Real-valued parametric conditioning of an rnn for interactive sound synthesis.
arXiv preprint arXiv:1805.10808.

Audio examples description I

1. Input MIDI 63 to Spectral Model
2. Spectral Model Reconstruction(trained on MIDI63)
3. Spectral Model Reconstruction(not trained on MIDI63)
4. Input MIDI 60 note to Parametric Model
5. Parametric Reconstruction of input note
6. Input MIDI 63 Note
7. Parametric AE reconstruction of input
8. Parametric CVAE reconstruction of input
9. Input MIDI 65 note(endpoint trained model)
10. Parametric AE reconstruction of input(endpoint trained model)
11. Parametric CVAE reconstruction of input(endpoint trained model)
12. CVAE Generated MIDI 65 Violin note
13. Similar MIDI 65 Violin note from dataset

Audio examples description II

14. CVAE Reconstruction of the MIDI 65 violin note
15. CVAE Generated MIDI 65 Violin note with vibrato