Berkeley
UNIVERSITY OF CALIFORNIA

CNMAT

CARMINE-EMANUELE CELLA

NEURAL NETWORKS FOR MUSIC APPLICATIONS

MUSIC 159

Geometric approach

OUTLINE

- signals can be represented in vector spaces with inner product;
- projections compute similarity, reconstruction from projections is possible with bases;
- Fourier: a-priori representation for frequency, invariant to time (independent), instable to frequency (dependent variable);
- from Fourier to Wavelets: a-priori representation for time-frequency, invariant to time, stable to frequency, information is lost (not unique), direct reconstruction is not possible;
- scattering: a-priori representation for time-frequency, invariant to time, stable to frequency, information is recovered from upper layers, direct reconstruction is not possible, first layer similar to MFCC;
- joint scattering: as scattering but able to connect different variables; e.g. spiral scattering acts on a third (dependent) variable;
- clustering: unsupervised representation based on geometric proximity;
- MLP: supervised representation that maximises projectors for the specific problem, can act on many variables non joinlty but it is difficult to know what are the variables;
- CNN: as MLP but can act jointly on variables;

PROJECTIONS

2.2. **Projection.** An important application of inner product is to project one vector over another. The projection of x on y is defined as:

$$\mathfrak{P}_x(y) = \frac{\langle x, y \rangle}{\|x\|^2} \cdot x \tag{3}$$

where the ratio between the inner product and the squared norm of x is called *coefficient of projection*.

•: **KEYPONT: projections compute the *similarity* (covariance) of the two vectors**.

2.3. **Reconstruction from projections.** A vector can be reconstructed with a linear combination from its projections on another set of vectors if and only if the set used is a basis.

ANALYSIS AND SYNTHESIS

3.1. **Analysis.** The analysis is the representation ϕ_x of a signal given by the inner product of it by a basis in a vector space; it is therefore given by the projection

$$\phi_x = \sum_t x(t) * \overline{b_k} = \langle x, b_k \rangle \tag{4}$$

where b_k is a given basis and t is time.

3.2. **Synthesis.** The synthesis is the reconstruction of the original signal x by the summation of the products with the representation ϕ_x created by the analysis:

$$x(t) = \sum_k \phi_x b_k(t) = \sum_k \langle x, b_k \rangle b_k(t). \tag{5}$$

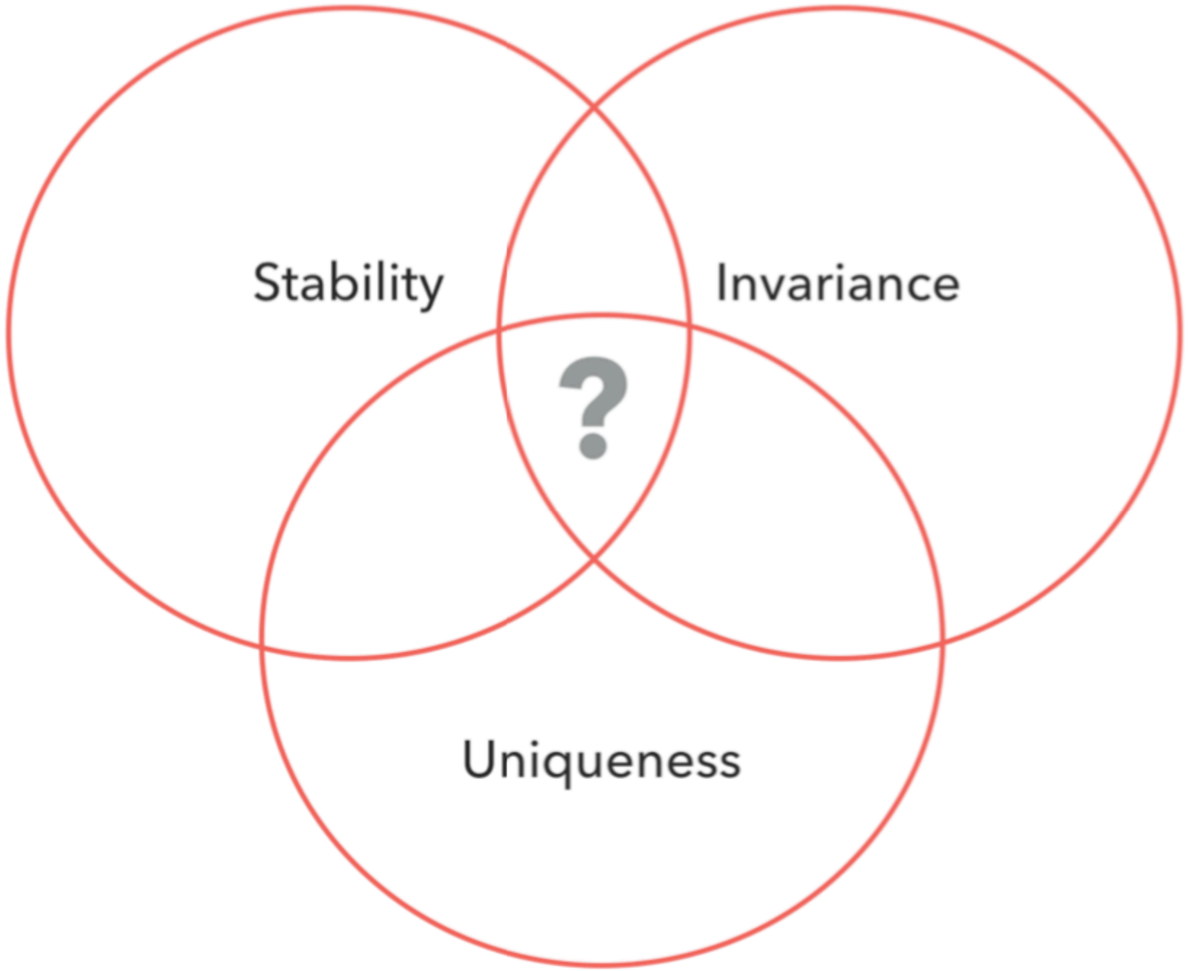
REPRESENTATIONS

Representations can be considered linear operators that need to be invariant to sources of unimportant variability, while being able to capture discriminative information from signals. As such, they must respect four basic properties; being x a signal and Φx its representation:

- *discriminability:* $\Phi x \neq \Phi y \implies x \neq y$;
- *stability:* $\|\Phi x - \Phi y\|_2 \leq C\|x - y\|_2$;
- *invariance (to group of transformations G):* $\forall g \in G, \Phi g.x = \Phi x$;
- *reconstruction:* $y = \Phi x \iff \tilde{x} = \Phi^{-1} y$.

Discriminability means that if the representations of two signals are different than the two signals must be different. Stability means that a small modification of a signal should be reflected in a small modification in the representation and vice-versa. Invariance to a group of transformation G, means that if a member of the group is applied to a signal, than the representation must not change; reconstruction, finally, is the possibility to go back to a signal that is *equivalent* to the original (in the sense of a group of transformations) from the representation. It is possible to divide representations in two major categories: *prior* and *learned*.

REPRESENTATIONS



NO LEARNING: FOURIER

4.1. **Fourier.** The Fourier representation is a specific case of analysis and synthesis, where the basis is given by a set of complex sinusouids: $b_k = e^i 2\pi k$ (where i is the imaginary unit).

The discrete Fourier analysis (DFT) will be therefore:

$$\hat{x}(k) = \sum_t x(t) e^{\frac{-i2\pi kt}{T}} \tag{6}$$

and, in the same way, the reconstruction (or inverse Fourier transform, IDFT) is given by:

$$x(t) = \frac{1}{T} \sum_k \hat{x}(k) e^{\frac{i2\pi kt}{T}}. \tag{7}$$

This can be thought of as a convolution with the Fourier basis

NO LEARNING: WAVELET/SCATTERING TRANSFROM

4.3. **Scattering.** The information lost in the wavelet representation is recovered by the following cascaded multi-layer representation:

$$S_1 x(t, \lambda_1) = |x \star \psi_{\lambda_1}| \star \phi(t). \tag{8}$$

These are called first-order scattering coefficients; second order scattering coefficients are defined as:

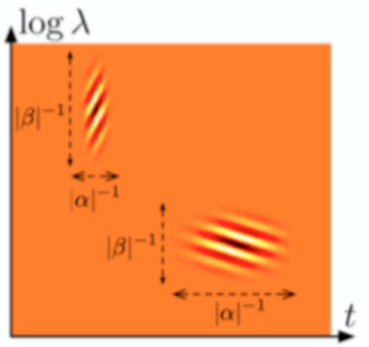
$$S_1 x(t, \lambda_1, \lambda_2) = ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \phi(t). \tag{9}$$

Iterating this process defines scattering coefficients at any order m:

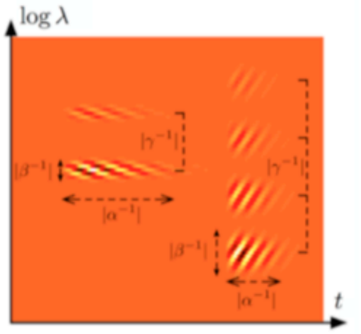
$$\boxed{S_m x(t, \lambda_1, \ldots, \lambda_m) = || \ldots |x \star \psi_{\lambda_1}| \star \ldots | \ldots \star \psi_{\lambda_m}| \star \phi(t)}. \tag{10}$$

NO LEARNING: JOINT SCATTERING TRANSFROM

- **joint-scattering**: $||x \overset{t}{\star} \psi_\lambda| \overset{t}{\star} \psi_\alpha| \overset{log\lambda}{\star} \psi_\beta|$



- **spiral-scattering**: $||x \overset{t}{\star} \psi_\lambda| \overset{t}{\star} \psi_\alpha| \overset{log\lambda}{\star} \psi_\beta| \overset{oct}{\star} \psi_\gamma|$



SUPERVISED LEARNING: MULTI-LAYER PERCEPTRON

5.2. **Multi-layer perceptrons.** Set of linear transformations followed by non-linearities (like scattering) in which the projectors are learned with supervision with backpropagation [...]:

$$MLP_1 = \rho(Wx + b) \tag{11}$$

where:

- W is a a linear transformation made of weights found during learning;
- b is a translation vector;
- is a point-wise application of a non-linearity.

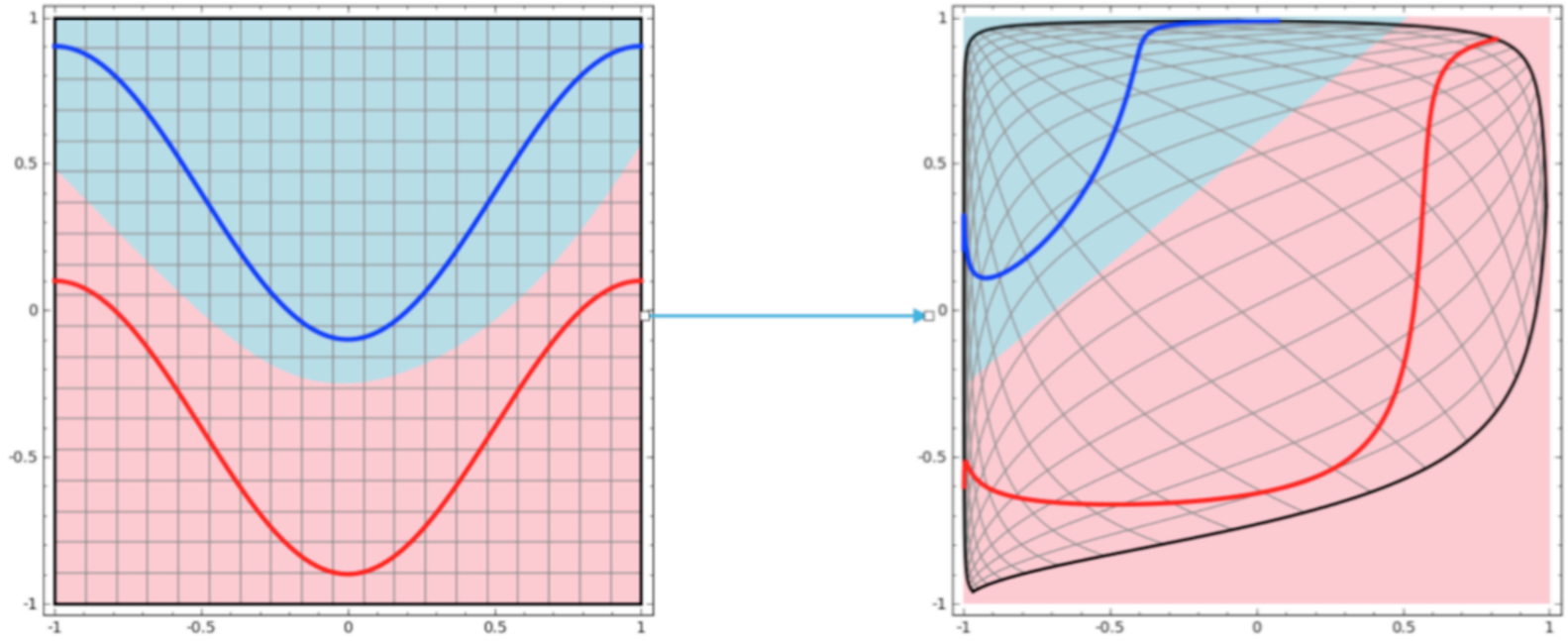
This structure can be repeated in a cascaded structure, creating invariances for different variables.

SUPERVISED LEARNING: CNN

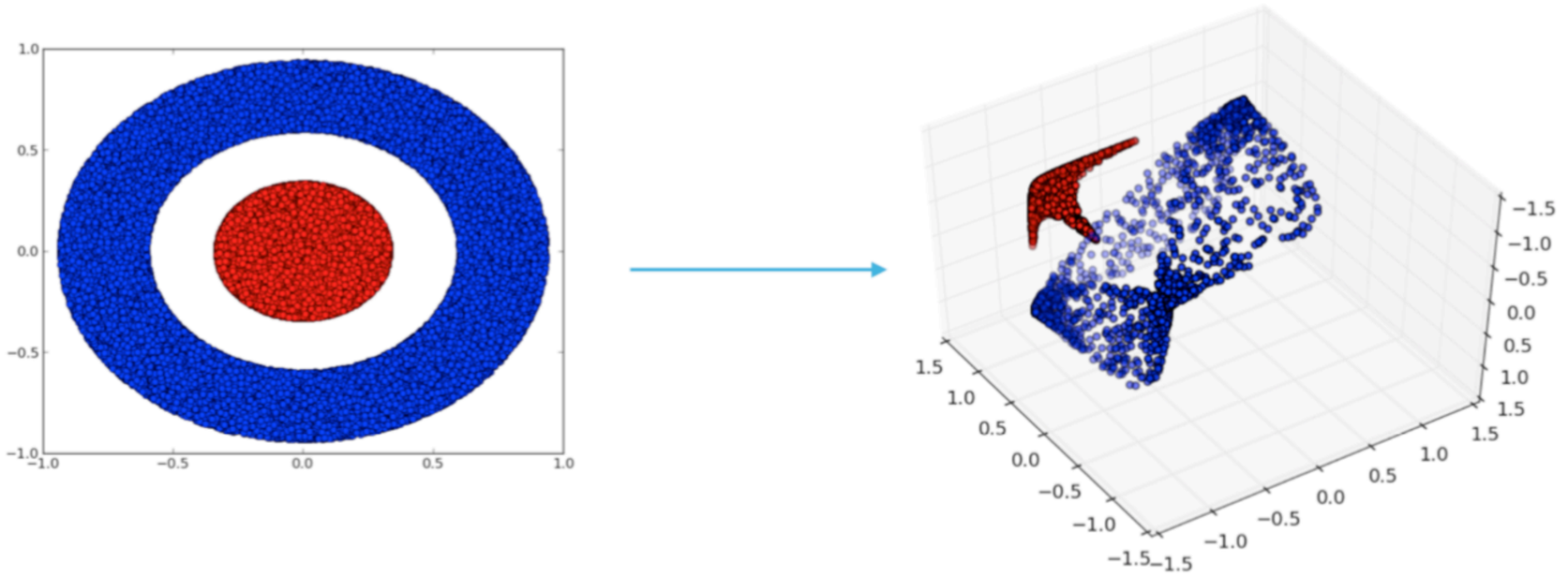
CNN are a supervised representations as MLP that can apply invariances and stabilize any number of variables and is also able to act *jointly* on the variable as joint scattering [...].

LINEARIZATION: 2D

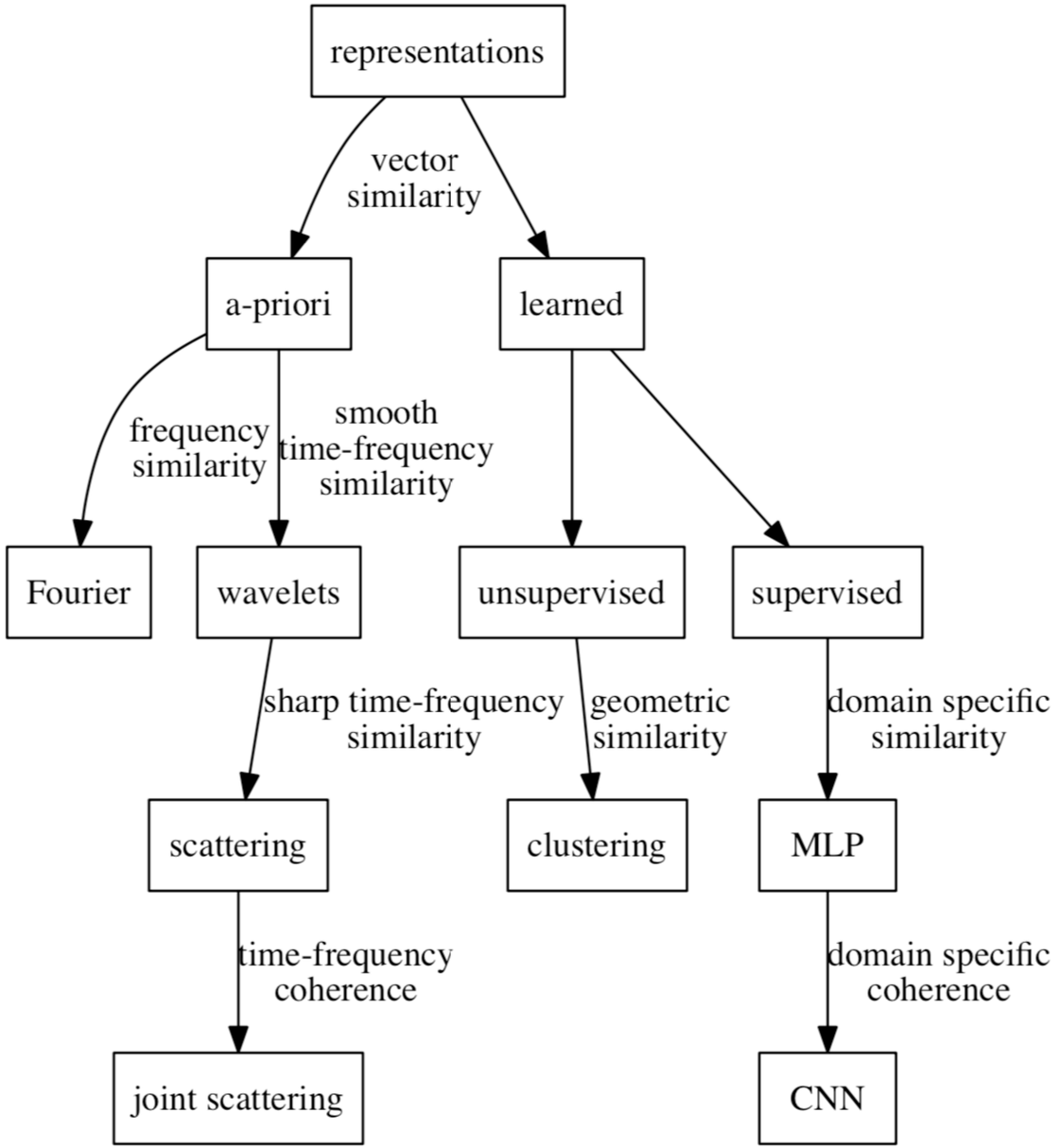
5.3. **Linearization.** We can think that the role of projectors and non-linearities in MLP is to *linearize* the feature space [...]. See figures 5 and 6.



LINEARIZATION: 3D



UNIFIED OVERVIEW



Applications

Musical instrument classification: classical approach

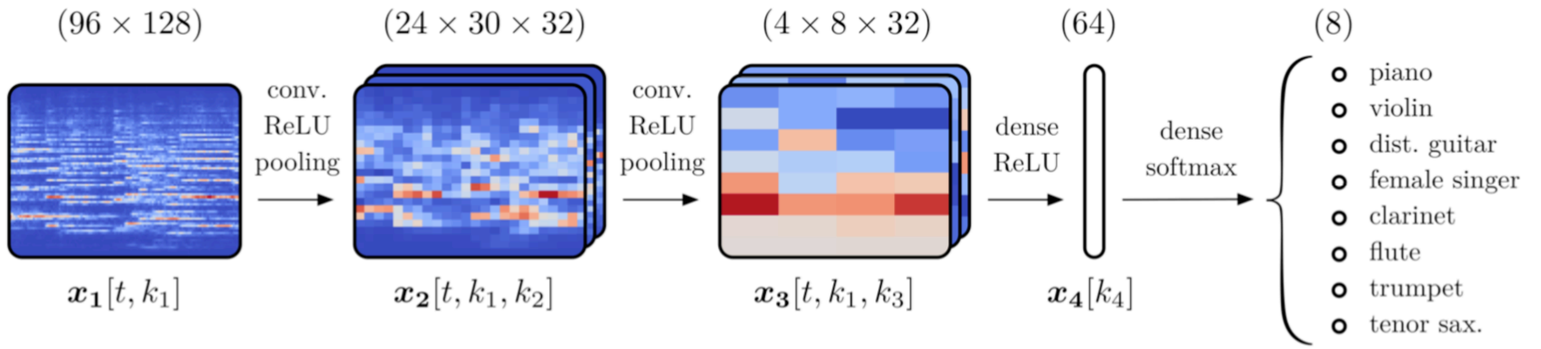
```
clfs = []

clfs.append(LogisticRegression(solver='lbfgs', max_iter=5000, multi_class='auto' ))
clfs.append(SVC())
clfs.append(KNeighborsClassifier(n_neighbors=5))
clfs.append(RandomForestClassifier(n_estimators=10))

print ("\nRunning classifications...")
for classifier in clfs:
    pipeline = Pipeline([
        ('normalizer', StandardScaler()),
        ('clf', classifier)
    ])
    print('---------------------------------')
    print(str(classifier))
    print('---------------------------------')
    shuffle = KFold (n_splits=5, random_state=5, shuffle=True)
    scores = cross_val_score (pipeline, X, y, cv=shuffle)

    print("model scores: ", scores)
    print("average score: ", scores.mean ())

    pipeline.fit (X_train, y_train)
    ncvscore = pipeline.score(X_test, y_test)
    print("non cross-validated score: ", ncvscore)

```

Musical instrument classification: deep learning approach (Lostanlen and Cella, 2016)

Deep convolutional networks (ConvNets) owe their success to two assumptions:
1. locality of correlations, and
2. stationarity of statistics.

Yet, the constant-Q transform (CQT) does not comply with them over the full hearing range.

What convolutional architectures for time-frequency representations ?



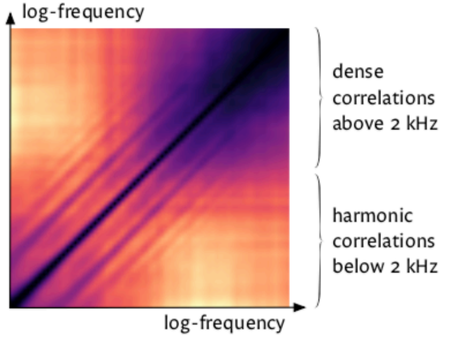
Two convolutional layers and two densely connected layers: a commonly used ConvNet architecture for MIR.
Trained with Adam optimizer, on stochastic cross-entropy, over normalized batches of size 32.
Dropout of 50% of the activations is applied at the last two layers.

Musical instrument classification: deep learning approach

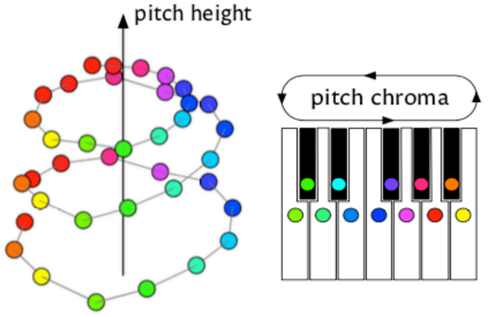
Problem 1: Locality of correlations ?

« Local neighborhoods in frequency do not share the same relationship » [Humphrey 2013].

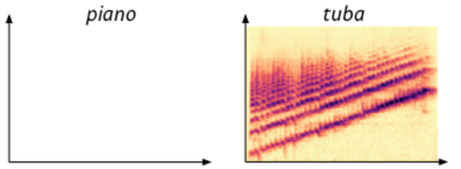
We computed the covariance matrix between CQT coefficients in the RWC dataset of isolated notes.



Isomap embedding [Le Roux 2007] of harmonic correlations reveals the pitch helix [Shepard 1965].



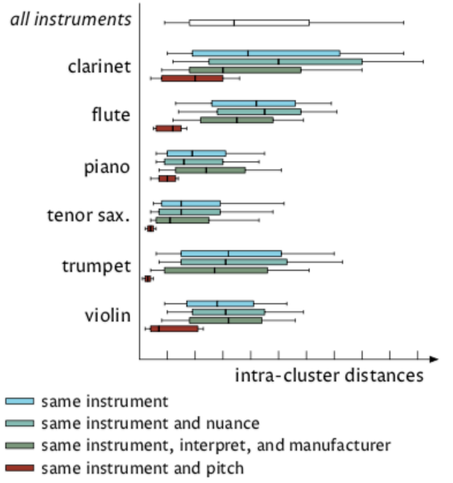
Problem 2: Stationarity of statistics ?



Source-filter interpretation:
the source is transposed by pitch shift while the overall spectral envelope remains unchanged.

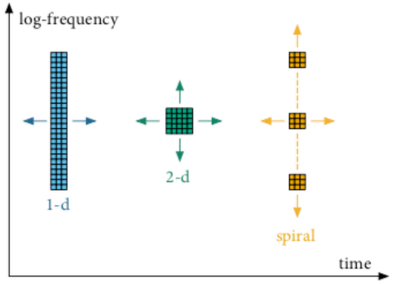
We computed pairwise distances in mel-frequency cepstral coefficients (MFCC) of isolated notes in the RWC dataset.

The DCT involved in MFCC yields the optimal basis under the assumption of stationarity.



Yet, MFCC are affected by realistic pitch shifts despite being designed to be invariant to frequency transposition of pure tones.

Solution: improved weight sharing



Distance ~1/n, unevenness ~1/n^2
At high frequencies, transposed pitches have similar spectra up to some additive bias.
We use 1-d convolutions above 2 kHz.

The probability of two randomly chosen partials between 1 and n to be in octave relationship is ~1/n.
We use spiral convolutios below 2 kHz.

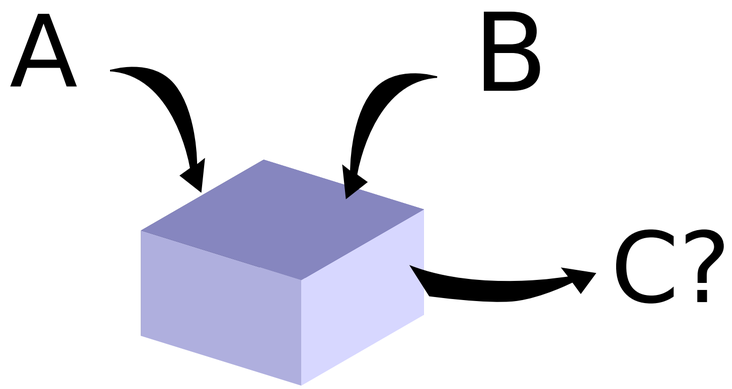
Experiments in musical instrument classification with MedleyDB [] for training and solosDB [Joder] for testing.

MFCC, random forest classifier	38.6
2-d ConvNet	30.9
& spiral ConvNet	28.3
& 1-d ConvNet	26.0
1-d scattering [Andén 2014]	32.0
2-d scattering [Andén 2015]	22.0
spiral scattering [Lostanlen 2015]	19.9

Hybridyzing convolutional layers with multiple weight sharing strategies improves classification accuracy of ConvNets with respect to the traditional 2-d architecture.
However, the state of the art is obtained by a deep scattering network, in which learned convolutional kernels are replaced by wavelets.

Style transfer: deep learning approach

Several techniques have been proposed to synthesize novel audio content owing its properties from two audio sources.



Traditional Approaches

Morphing vs. Hybridization [Caetano et al., 2012]:

- Morphing: C interpolates features from A and B
- Hybridization: C inherits features from A and B

Popular Terms:

- Cross-Synthesis = Vocoding = algorithms where the spectral envelope of one sound is impressed on the flattened spectrum of another
- Analysis-Resynthesis = algorithms to describe the timbre and manipulate it (e.g. sinusoidal modelling)
- Many other approaches exist...

Traditional Approaches

Morphing vs. Hybridization [Caetano et al., 2012]:

- Morphing: C interpolates features from A and B
- Hybridization: C inherits features from A and B

Popular Terms:

- Cross-Synthesis = Vocoding = algorithms where the spectral envelope of one sound is impressed on the flattened spectrum of another
- Analysis-Resynthesis = algorithms to describe the timbre and manipulate it (e.g. sinusoidal modelling)
- Many other approaches exist...

Machine Learning Approach

- It should learn automatically what features to morph/hybridize
- Hierarchical representation

Previous Works

Image Processing:

- Style Transfer: [Gatys, 2016]

Previous Works

Image Processing:

- Style Transfer: [Gatys, 2016]
- The same approach fails on audio spectrogram used as images [Foote, 2016; Ulyanov, 2016; Geng, 2016]

Previous Works

Image Processing:

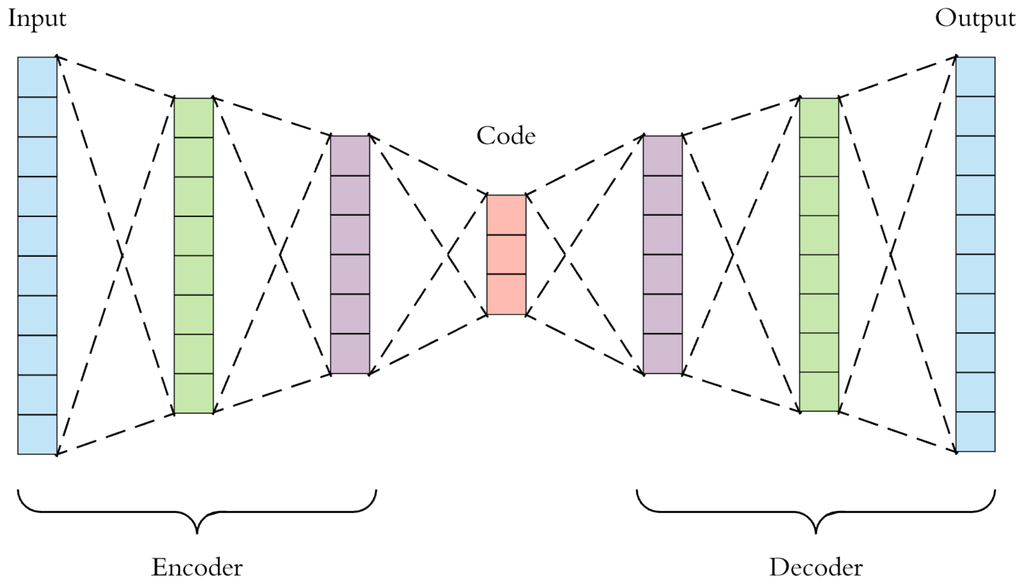
- Style Transfer: [Gatys, 2016]
- The same approach fails on audio spectrogram used as images [Foote, 2016; Ulyanov, 2016; Geng, 2016]

- Thesis: audio spectrograms not isotropic, time and frequency have different scales, structure and repetition. Audio data is very fine grained (max pooling is dangerous)

Autoencoding Approach

Autoencoding Neural Network

- Neural net composed of an encoding and decoding section.
- **Hp**: each layer applies a nonlinear geometric transformation
- Reducing the dimensionality and imposing a minimum reconstruction loss at the end obtains a compression, i.e. dimensionality reduction with minimum loss
- The hidden layer gives the *latent code*, or *latent variables* i.e. the learned information, while the other layers perform a compressive transform and its inverse



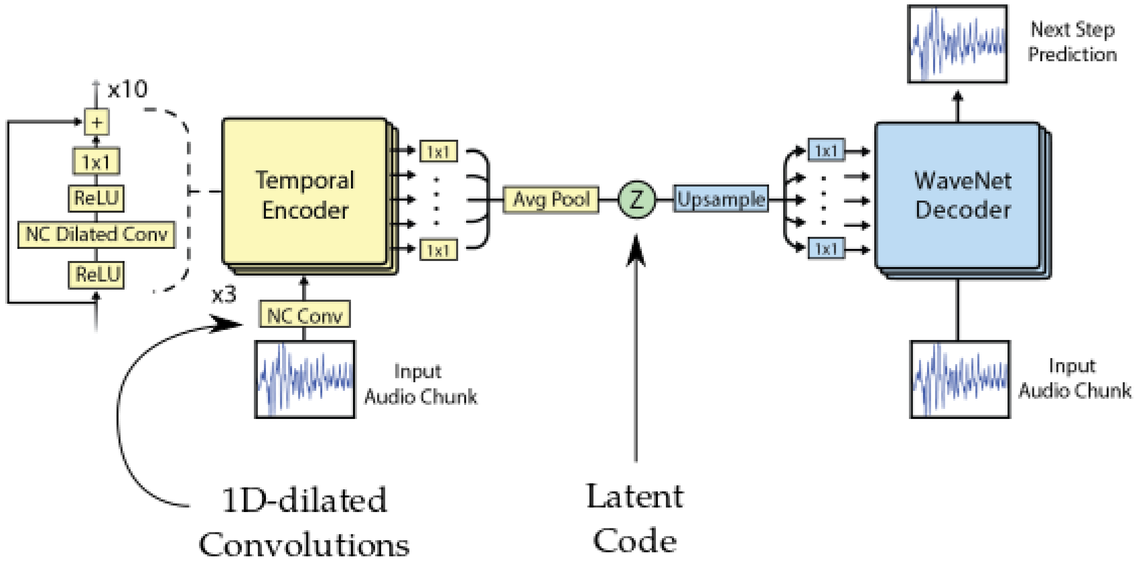
Idea

- **Starting Point** (early y2017): if the network can learn to reconstruct the main features of the training signal it may be able to filter/modify a novel signal imposing the learned features (hybridization).

The aim of this work is to test whether this happens and observe what kind of features are learned/retained during this process.

Previous Works

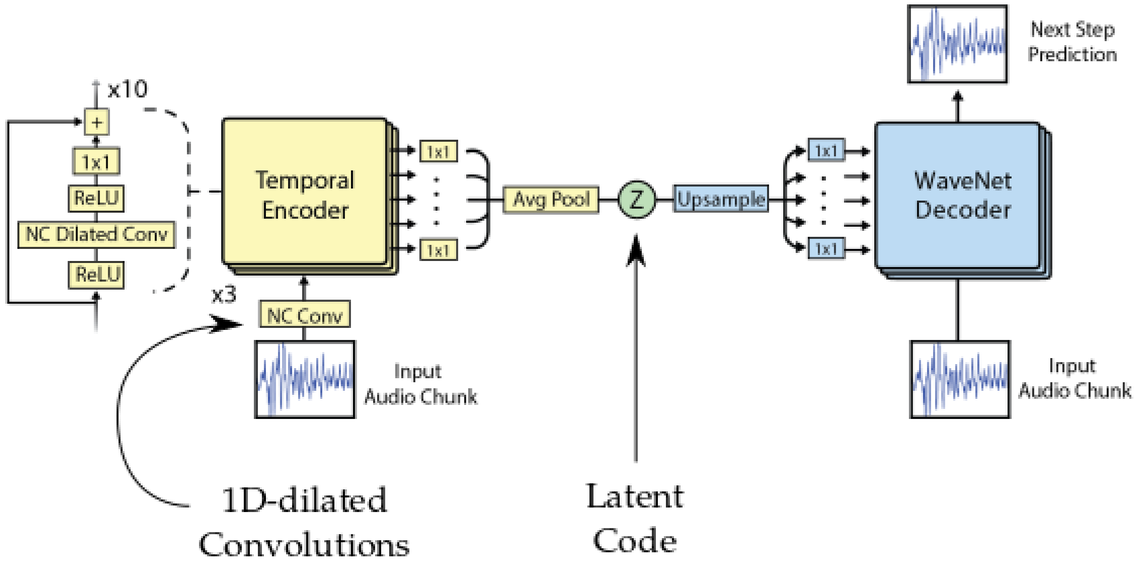
- WaveNet Autoencoder [Engel, 2017]



- Morphing occurs between latent variables from two input sources

Previous Works

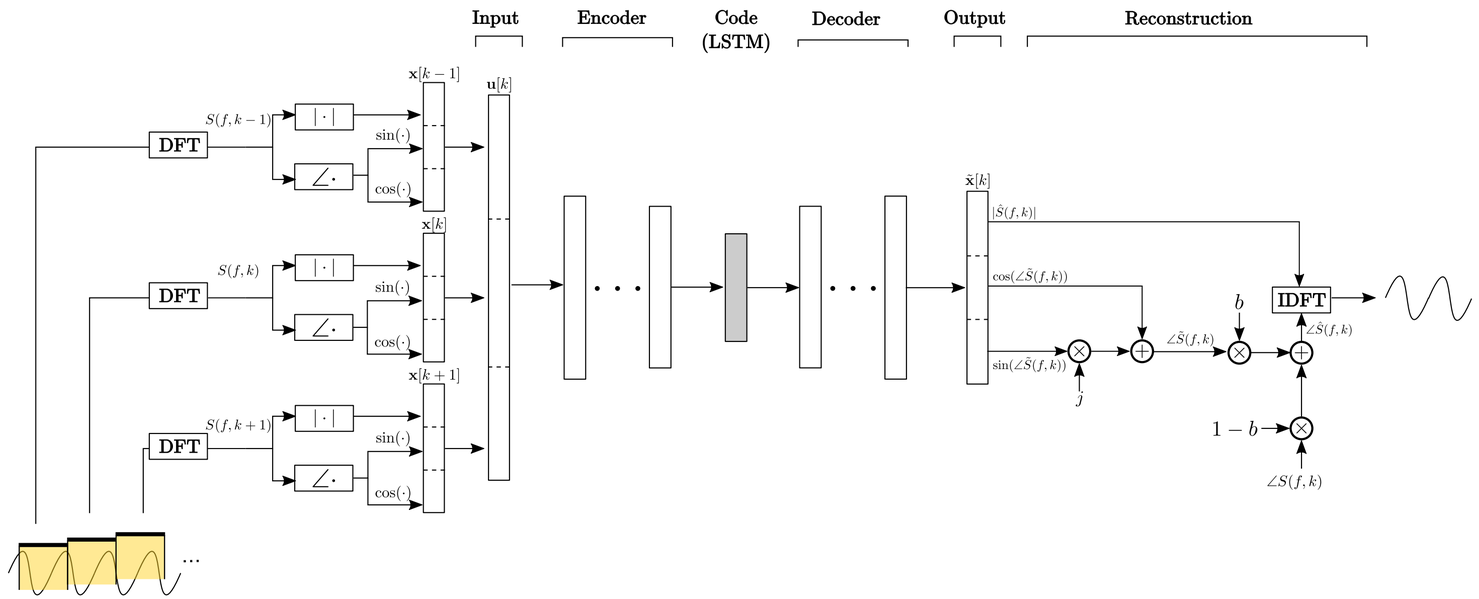
- WaveNet Autoencoder [Engel, 2017]



- Morphing occurs between latent variables from two input sources
- Training a wavenet on commercial GPUs takes weeks

A possible architecture (Gabrielli, Cella, et al., 2017)

Proposed Architecture



Notes

- A simple STFT autoencoder did not get quite close to the point

Notes

- A simple STFT autoencoder did not get quite close to the point
- Two modifications to the network improved the results:
  - Input context
  - LSTM hidden layer

Notes

- A simple STFT autoencoder did not get quite close to the point
- Two modifications to the network improved the results:
  - Input context
  - LSTM hidden layer
- Results will be provided later

Experiments

Comparative Methods

- Spectral Envelope Hybridization (cepstral domain):
  - Spectral Envelope computed for both inputs as

$$E = DFT(W_{LP}(DFT^{-1}(log(|DFT(s)|)))) \quad (1)$$

    with W_{LP} *liftering* filter
  - Target envelope is flattened by deconvolution
  - Product of the Flattened envelope and the Source signal
  - outcome length = length of the longest, the other is repeated

---

- MFCC-based Mosaicing [Burred, 2014]:
  - Replaces frames of the source signal by frames of the target signal (used as a *dictionary*)
  - match is done by k-nn in the MFCC domain
  - outcome length = source length

Dataset 1

- Internal dataset of recorded solo instrumental or vocal tracks
- Length: 30-120s per file
- Instruments: clean/dst guitar, synth pad, trumpet, electric piano, male and female voice
- recordings of performance on a single tonality. They contain notes, chords, legatos, glissandi, etc.

Dataset 2

- Magenta NSynth dataset (http://magenta.tensorflow.org/nsynth)
- single notes, spanning pitch and dynamic
- 11 classes of musical instruments
- we randomly selected 3500 files for each class for balanced training

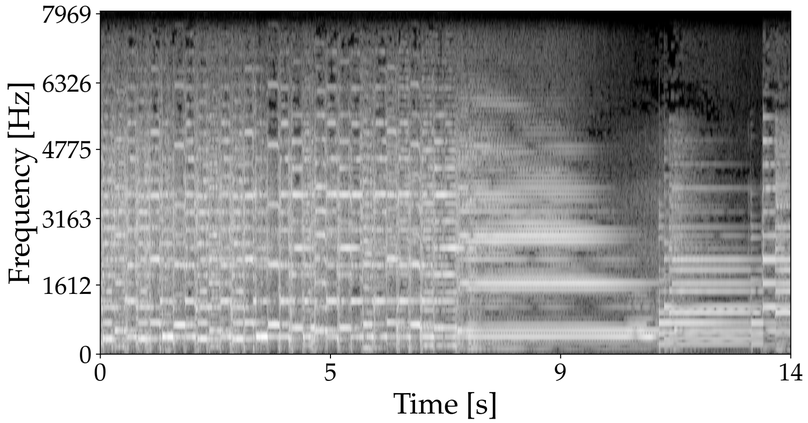
Dataset 1

- Internal dataset of recorded solo instrumental or vocal tracks
- Length: 30-120s per file
- Instruments: clean/dst guitar, synth pad, trumpet, electric piano, male and female voice
- recordings of performance on a single tonality. They contain notes, chords, legatos, glissandi, etc.

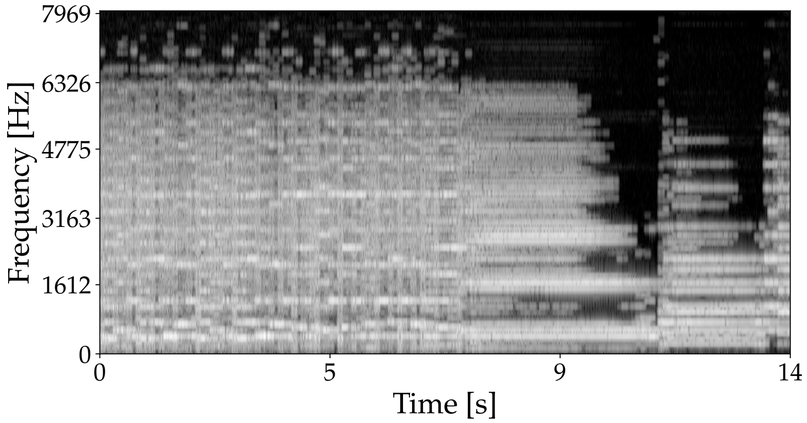
Dataset 2

- dataset (http://magenta.tensorflow.org/nsynth)
- ning pitch and dynamic
- sical instruments
- selected 3500 files for each class for balanced training

Sanity Check: reconstruction test

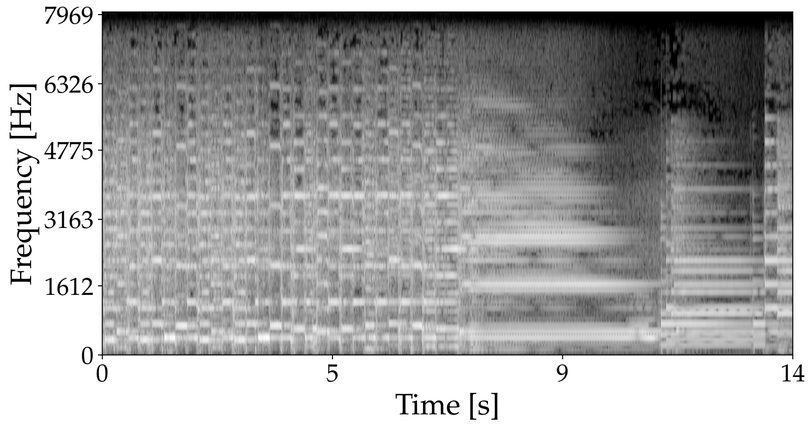


(a) Original electric guitar track (input and target).

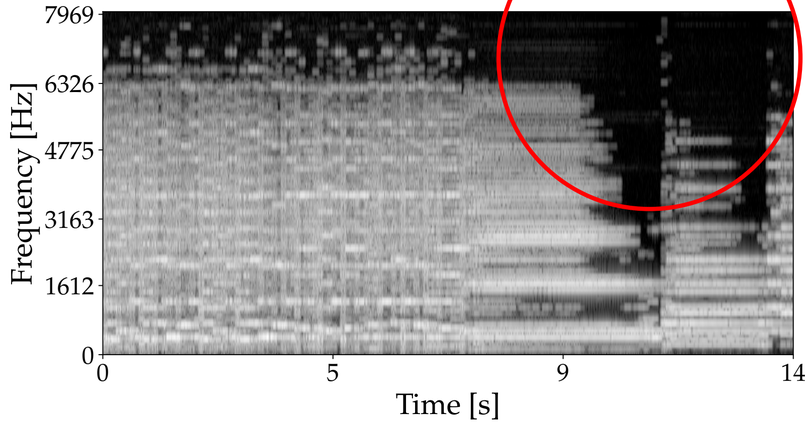


(b) Reconstruction.

Sanity Check: reconstruction test



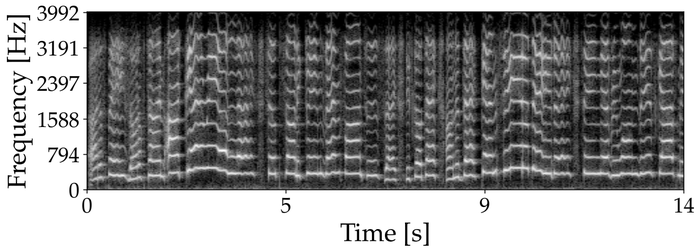
(a) Original electric guitar track (input and target).



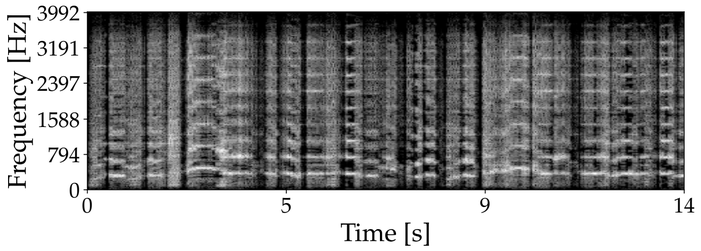
(b) Reconstruction.

Experiments w/ Dataset1

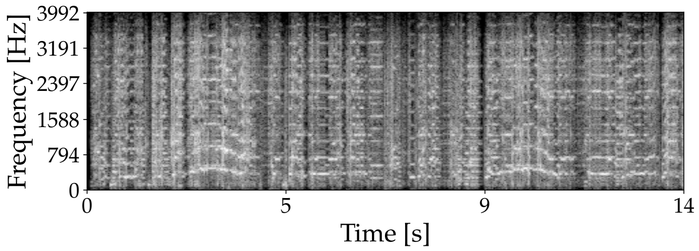
- Training: a single file from the dataset for each experiment
- Testing: Female Singing voice input: FV1 (all sort of pitched, unpitched, unvoiced, pitch variations, etc.)



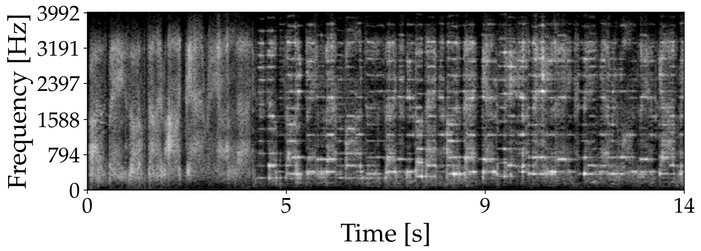
(a) Female voice FV1 (input).



(b) The proposed approach.



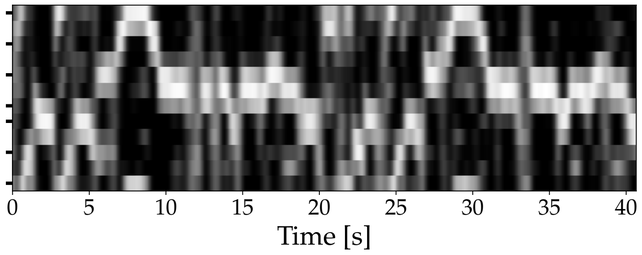
(c) k-nn matching techinque.



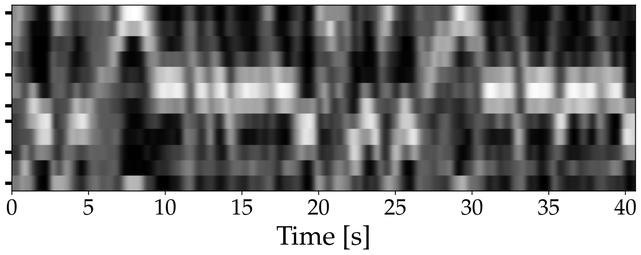
(d) Spectral flattening technique.

Experiments w/ Dataset1

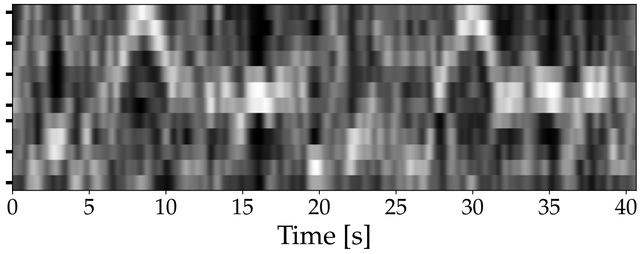
- Training: a single file from the dataset for each experiment
- Testing: Female Singing voice input: FV1 (all sort of pitched, unpitched, unvoiced, pitch variations, etc.)



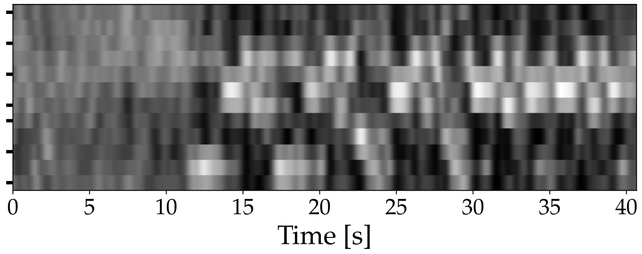
(a) Female voice FV1 (input).



(b) The proposed approach.



(c) k-nn matching technique.



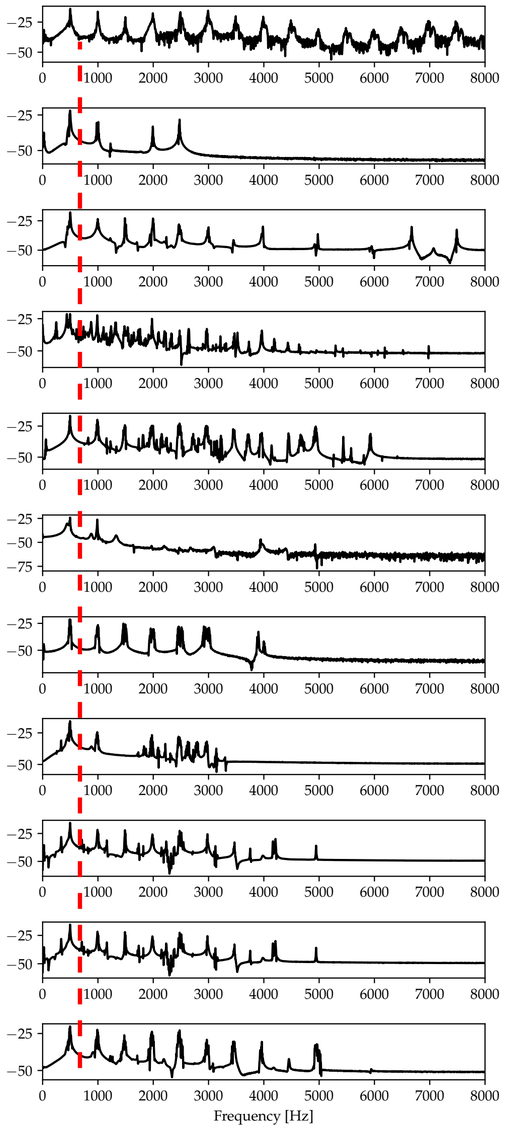
(d) Spectral flattening technique.

Considerations

- Timbre is quite coherent from frame to frame if compared, for example, to the MFCC-based method
- The spectral flattening method has a recognizable vocoder-like timbre, with the musical structure of the target file appearing together with its spectral content (undesirable)
- Pitch trajectories of the proposed approach are more similar to the testing input
- We observe that the network fails to follow the pitch of the input when it is outside the range learned during the training phase
  Example:
  - The high pitch of the input file reaches a B4 which the network cannot match, due to the lack of notes above G#4 in the training set

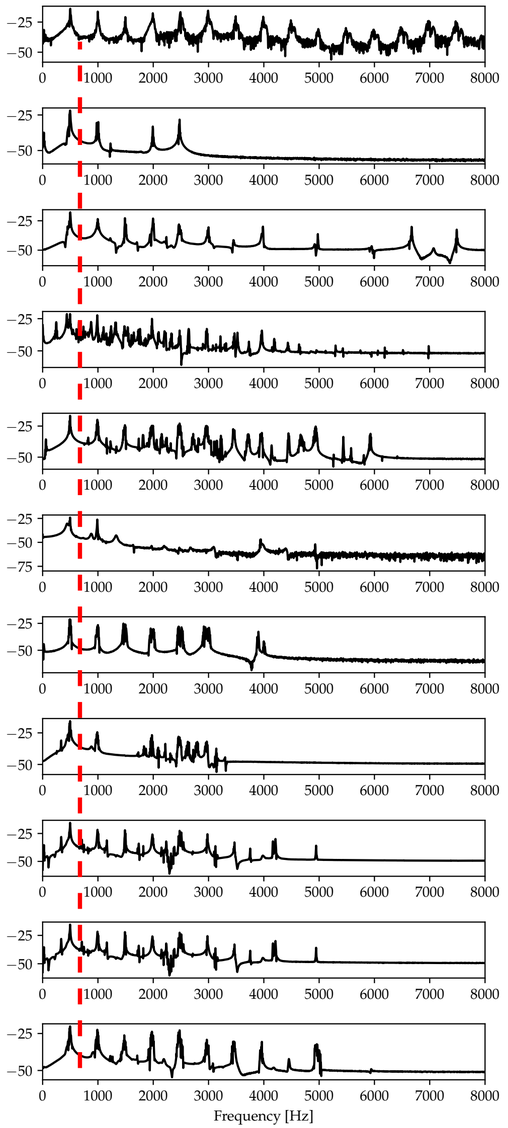
Experiments w/ Dataset2

- Training: an entire instrument class (3500 tones)
- Testing: FV1 as before



Experiments w/ Dataset2

- Training: an entire instrument class (3500 tones)
- Testing: FV1 as before
- DFT (4096 points) calculated for each file at a specific time interval, where the input file shows a pitched */i:/* phoneme at frequency 497.6 Hz (B4 + 13 cents)



Considerations

- we observed that the network **cannot** follow glissando, pitch bending and vibrato from the input because the dataset has no time-varying pitch to be learned
- applying FV1 as input results in lower-quality note transitions compared to the autoencoder trained on Dataset 1.
- The network cannot learn all the timbres of the different instruments in a class, but it learns instead an averaged spectrum of the whole class.
- The bandwidth of the output is related to that of the class used for training

Pitch Tracking Experiments

Method:

- Pitch accuracy evaluation to assess the pitch tracking accuracy of the architecture
- 5 MIDI files $\times$ 20 random MIDI notes each $\times$ 10 instrument classes $=$ 1000 notes

Results:

- Only 12% of the 1000 notes lost the original pitch

Conclusions

Discussion

- Pitch is taken from the source signal
- works also for multiple pitches, providing harmonization
- unpitched frames map to unpitched (if in training set)
- some timbre learning and transferring (highly dependent of the training set)
- spectral continuity sort of OK (compared to Mosaicing) (LSTM has a role)
- input energy not transferred to the output (depends on training set dynamic levels)... For now handled outside the neural network
- morphing is possible

Conclusions

- Explorative study: although simple, this autoencoder approach seems to work
- More *expressive* models need to be built able to generalize and learn a larger dataset
- convolutional techniques should be experimented with (WaveNet paved the way...)
- Still a lot to do on feature learning and convolutional techniques for timbre learning (e.g. instrument classification tasks)

https://gitlab.com/a3labPapers/CompanionFiles/tree/master/AES-XSynth

Game over?

A Universal Music Translation Network

Noam Mor, Lior Wolf, Adam Polyak, Yaniv Taigman
Facebook AI Research

Abstract

We present a method for translating music across musical instruments, genres, and styles. This method is based on a multi-domain wavenet autoencoder, with a shared encoder and a disentangled latent space that is trained end-to-end on waveforms. Employing a diverse training dataset and large net capacity, the domain-independent encoder allows us to translate even from musical domains that were not seen during training. The method is unsupervised and does not

STYLE TRANSFER (?)

The following audio samples were transferred between:

- Symphony, Mozart
- String Quartet, Haydn
- Cantata (Chorus opera & Orchestra), Bach
- Organ, Bach
- Piano, Bach
- Harpsichord, Bach

❖ Electric Guitar, Charlie Christian
❖ Electric Guitar, Metallica
❖ Classical Guitar & Orchestra , Jazz
❖ Trumpet & Orchestra, Jazz
❖ Midi samples of Piano & Trumpet, Elvis Presley, Rihanna
❖ Music of Africa
❖ Whistling (Human)

- **In training distribution**

❖ **Out of training distribution**

THANK YOU!

Suggested exercise: try to implement your own network for creative applications!