Recurrent networks and music analysis

Brian McFee
https://bmcfee.github.io/
NYU MARL / CDS

Outline

1. Overview of the domain
2. Input representation
3. Convolutional and recurrent models
4. Examples

About me

https://bmcfee.github.io/

- Ph.D. in Computer Science
  UCSD, 2012
- Postdoc in EE & Jazz
  Columbia, 2012-2014
- Data science fellow
  NYU, 2014-2018
- Asst. Prof of Music Tech & Data Science
  NYU, 2018-

What is music information retrieval (MIR)?

1. Extracting and summarizing the **semantic content of music**

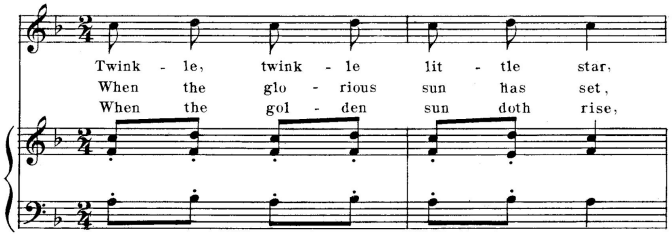
2. Making this content **accessible** and **useful to people**

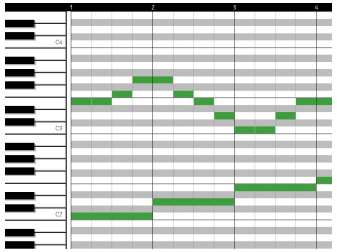
Applications of MIR

- Recommender systems
- Scholarship (musicology, informatics, ...)
- Music education
- Interactive display
- Machine-assisted creativity

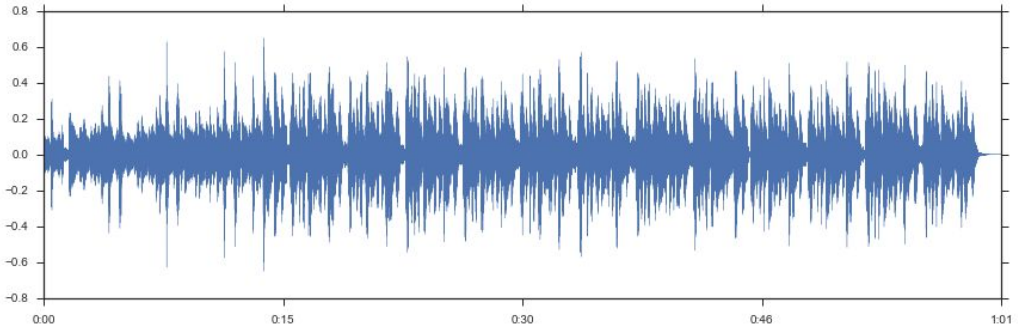
What do we mean by music?

- Score / symbolic representation

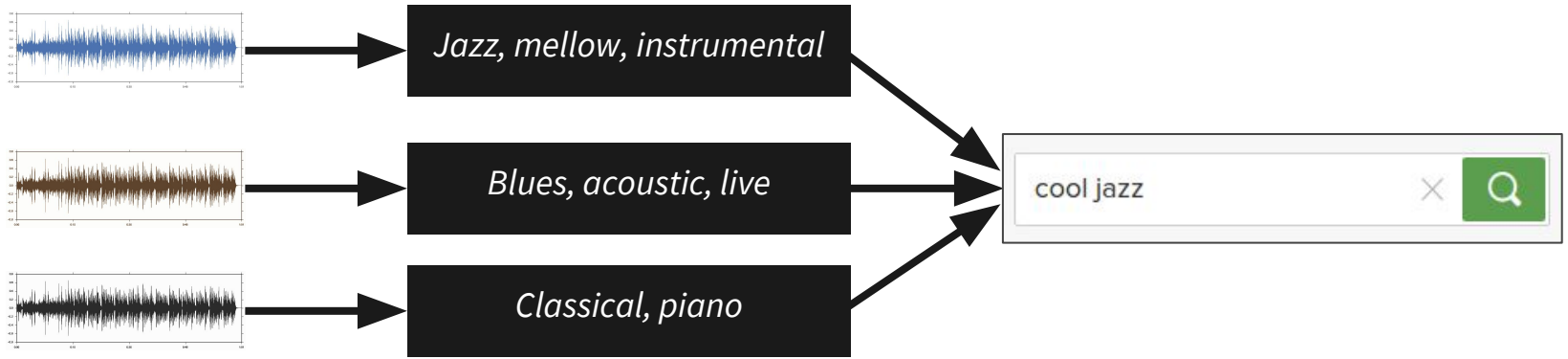




- Audio



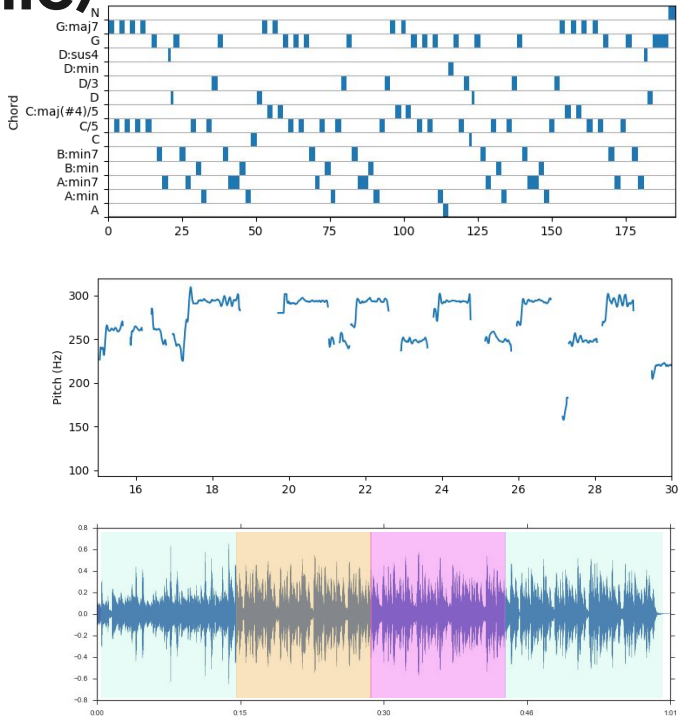
Track-level applications (static)



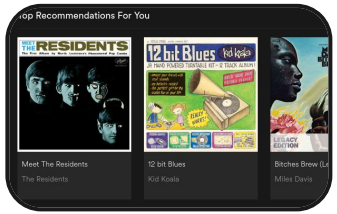
- Tagging, search, and retrieval

Track-level applications (dynamic)

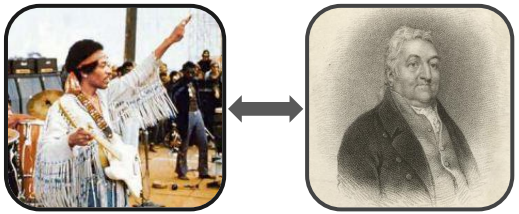
- Instrument detection
- **Chord recognition**
- Melody extraction
- **Beat and meter tracking**
- Structural segmentation
- Source / channel separation
- Automatic transcription



Corpus-level applications



Recommendation



Composition / cover identification

Fingerprinting and duplicate detection

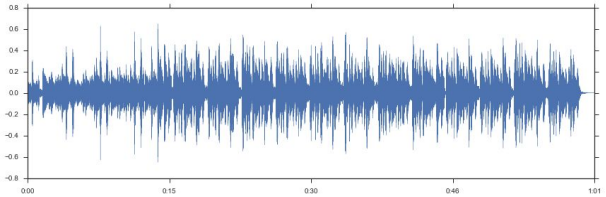
Soft similarity

Exact similarity

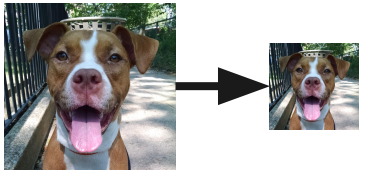
What's so special about music?

- Structure of the audio
- Structure of the concepts
- *Structure is intentional!*
- *And hierarchical!*

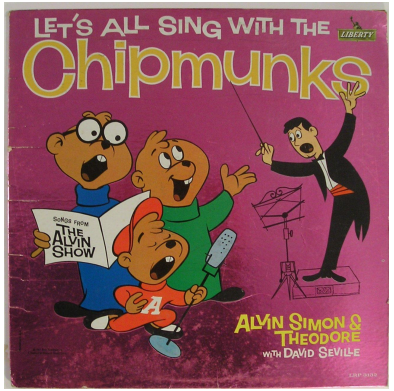
Audio vs. Images



- 1D Input: (mono) amplitude over time



- Interactions can be both **local** and **long-range**
  - Do Markov assumptions make sense here?



- ***Not scale-invariant***
  - Unlike in images, downsampling does not preserve structures
  - Models should support variable-length input

Music vs. Speech

- High variability of observations
- Sources overlap in time and frequency
- Ambiguity and subjectivity are huge factors!
- ***Ground truth*** usually doesn't exist, so act accordingly

How does structure help?

We often know things about the input, e.g.:

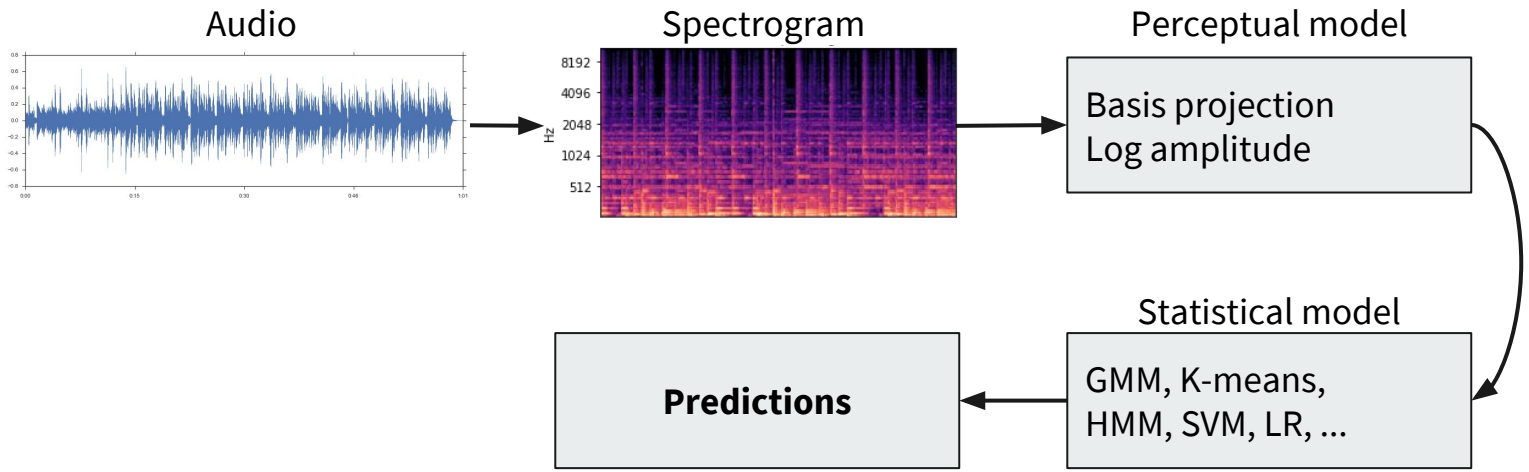
- Most music has sustained tones and/or rhythmic structure
- Hierarchical organization
  - Recording -> Sections -> Bars -> Beats -> ...
- We can leverage work in auditory perception and cognition
- Physics can help too!

And the output space, e.g.:

- Regularity in event timing (beats, onsets)
- Structure of chords (from music theory)
- Melody is piece-wise continuous
- Some instrument combinations are more likely than others

Input representations

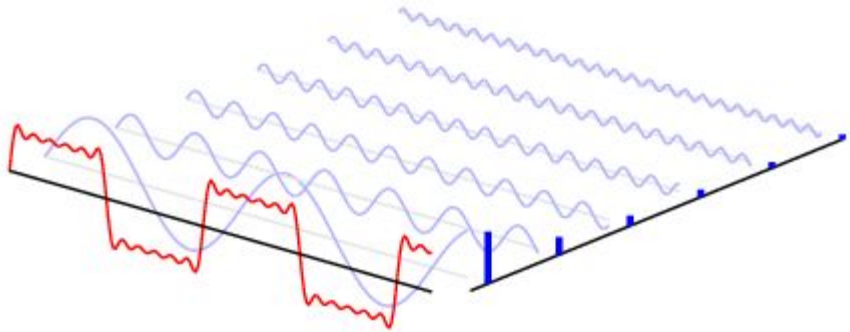
A standard pipeline, ca. 2002-2011



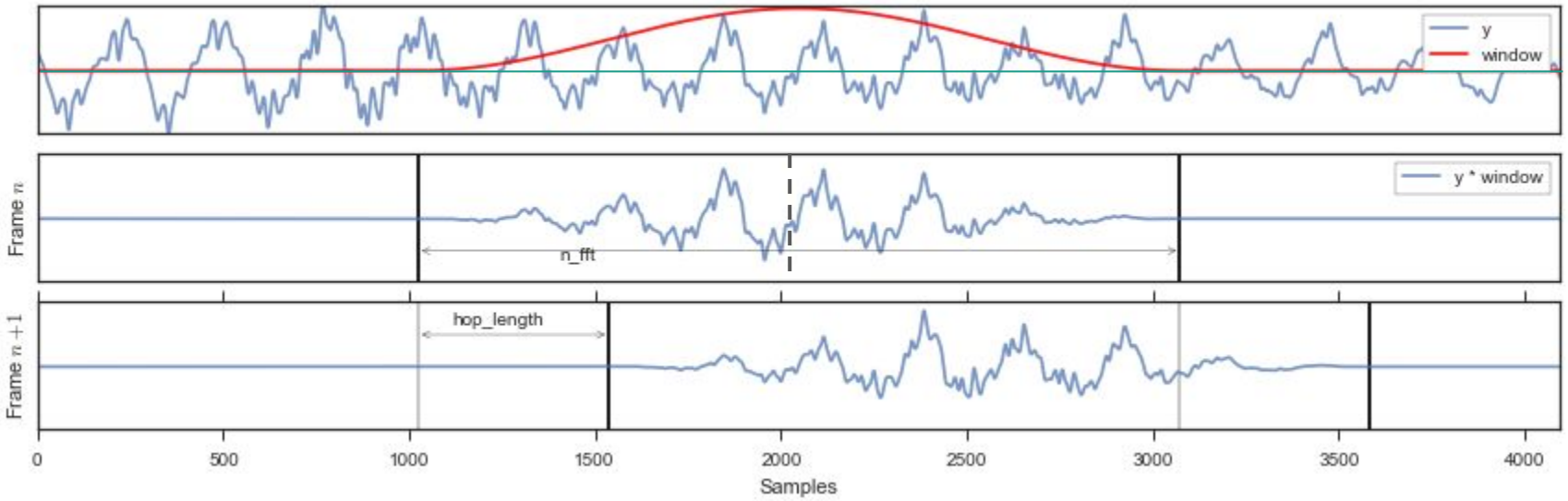
(e.g., instruments, tags, beats, ...)

Spectrograms

- Break sound apart into different frequencies
  - **Fourier analysis**
- Measure how energy changes over time
  - **Short-time Fourier transform**
- Break audio into small, overlapping **frames**

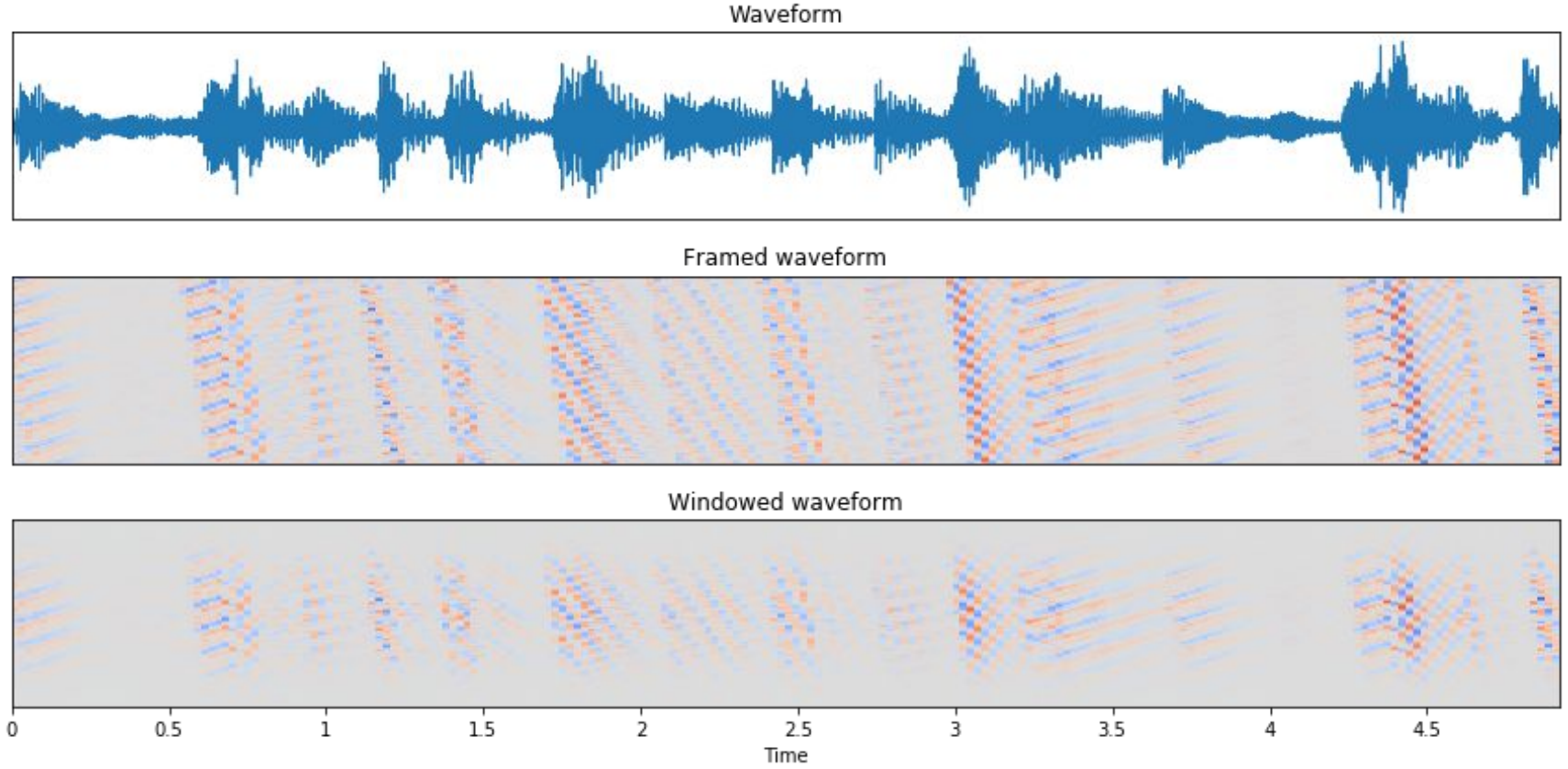


http://pgfplots.net/tikz/examples/fourier-transform/

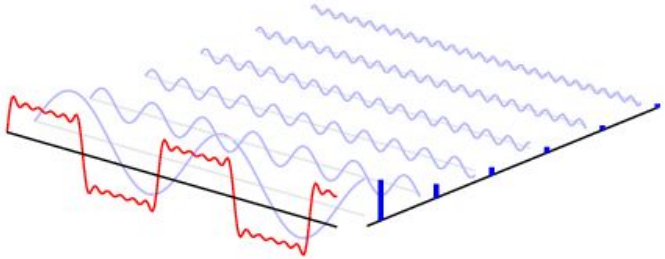


frame[n] → frame[n] * window

Neighboring frames differ by shifts when the signal is stationary



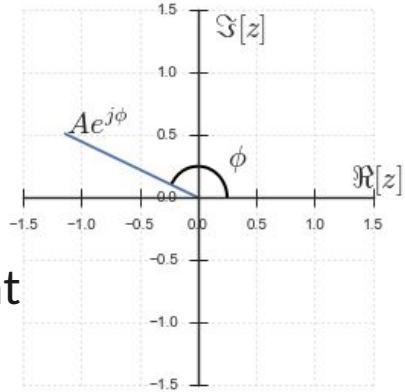
DFT



- **Complex**-valued coefficient for each frequency

  $$X(f) = Ae^{j\varphi}$$

  - A = Magnitude (energy)
  - φ = Phase (shift)

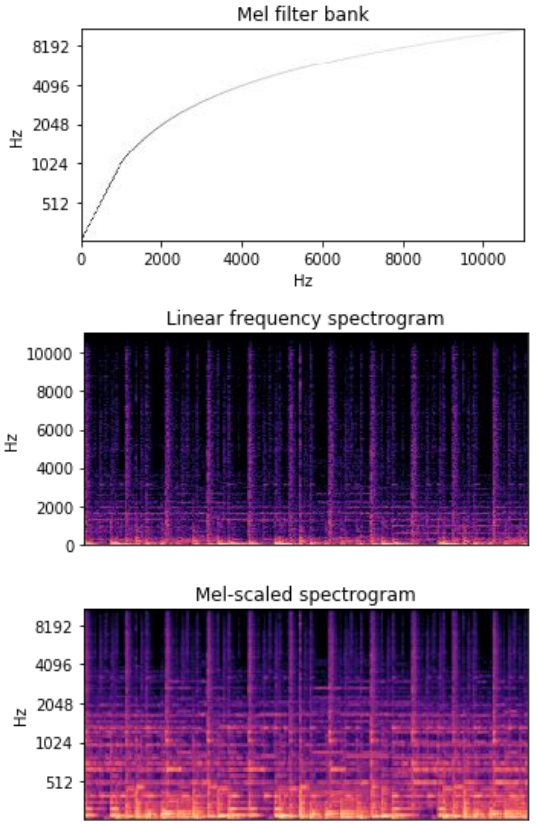


- Usually we discard phase
  - Phase is non-linear and depends on frame alignment
  - Complex values are a nuisance
  - Energy carries a lot of information anyway

Stevens, S. S., Volkmann, J., & Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, *8*(3), 185-190.

Perceptual model: frequency

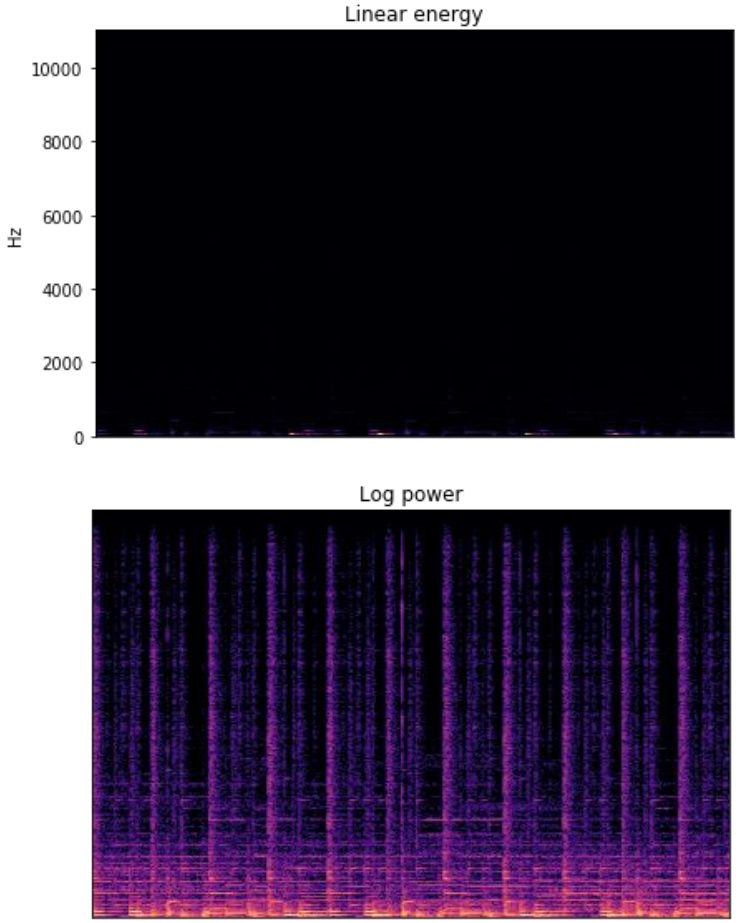
- FFT uses **linearly**-spaced frequencies
- Mel scale* is **log**-spaced above a threshold frequency
  - Tuned by psychoacoustic experiments to match humans
- Often used as **dimensionality reduction**
  - Eg, 1025 FFT bins -> 128 Mel bins
- Decent representation for timbre, not so great for pitch



*There are several numerical approximations to the Mel scale. Pick your favorite.

Perceptual model: amplitude

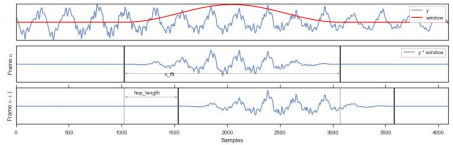
- Perception of “loudness” is approximately logarithmic, not linear
- Convert $|A|$ to $\log |A|^2$
- Linearizes energy ratio comparisons:
  $\log |A|^2/|B|^2 \rightarrow 2(\log|A| - \log|B|)$

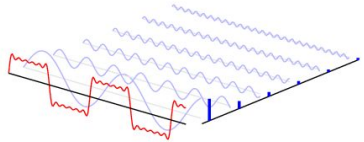


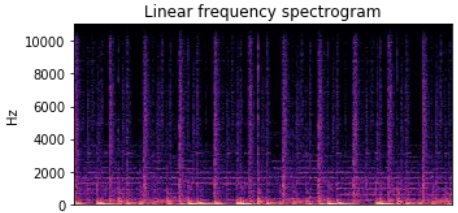
What did all of this do?

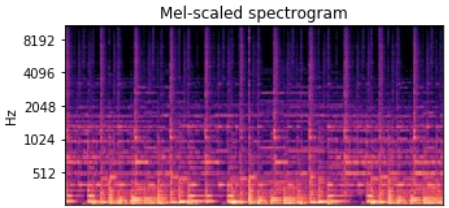
- Framing
- Discrete Fourier transform
- Magnitude spectrum
- Mel-scaling
- Log amplitude
- Downsampling (by frame advance)
- Linear change of basis
- Non-linear transform
- Dimensionality reduction
- Non-linear range compression

Result: time-series of feature vectors

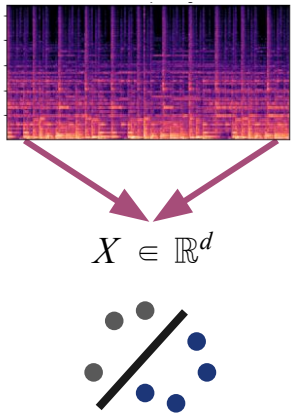


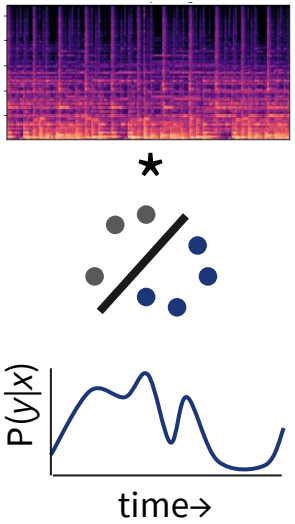


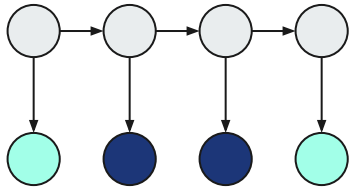




	Static	Dynamic (local)	Dynamic (long-range)
Example	*Tag prediction*	*Instrument activations*	*Chord recognition*
Method	• Summarize features over time • Model the summary vector	• Predict on local windows • I.e., convolve with classifier	• Hidden Markov Model, RNN...







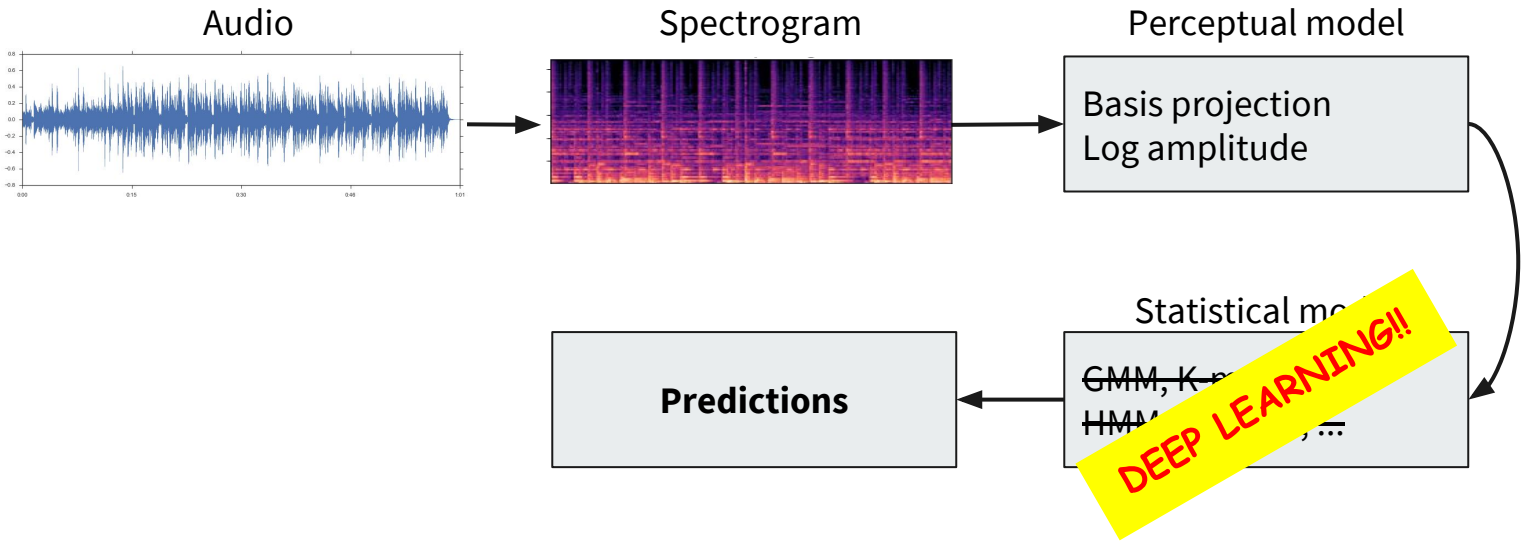
Prediction time

Modern MIR

… aka, the deep learning takeover

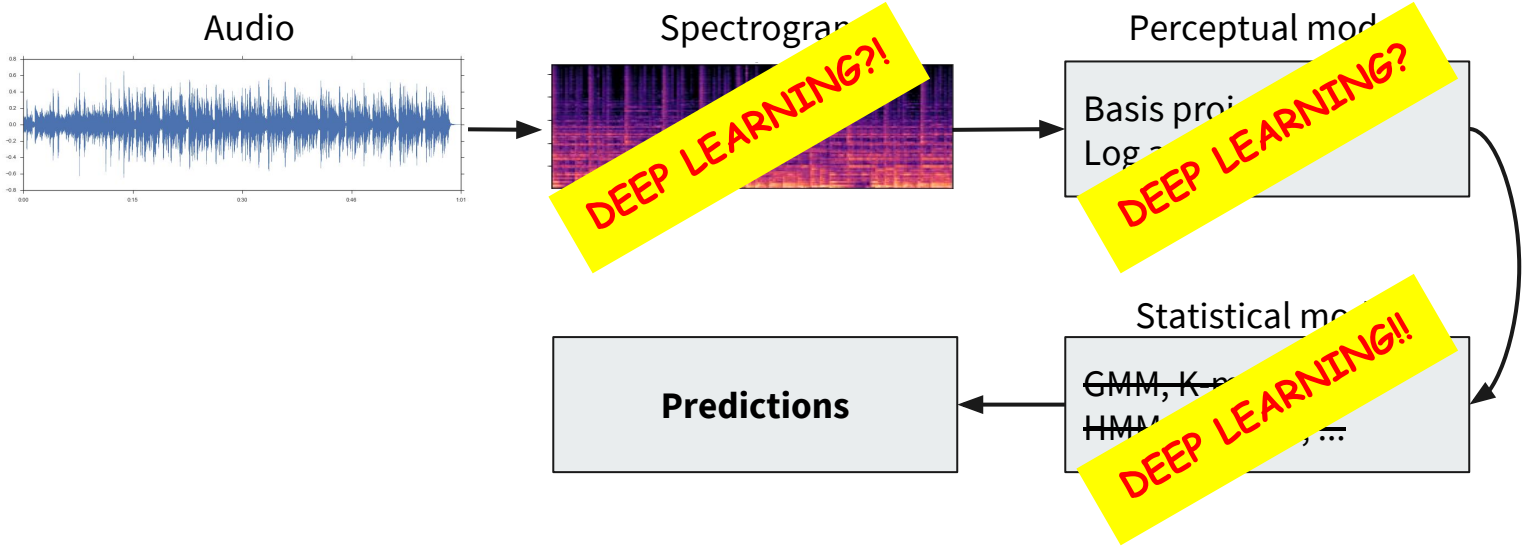
Going deep

- Stick a deep network on top of the existing feature stack
- Add time-convolution or recurrence to exploit temporal dependencies



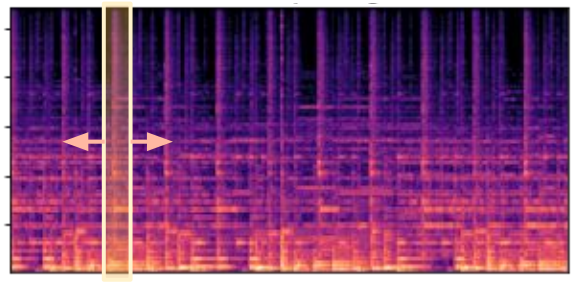
Going deeper...

- End-to-end learning?
- Mostly doesn't improve over spectrograms
- But it can be more storage-efficient in large-data regimes



Common architectures: convolutional networks

- Input whitening (or batch-norm) after log-scaling
- Convolve filters along time axis
- For global prediction tasks, temporal pooling is fine
  - You can also learn to pool! [M., Salamon, Bello, IEEE-TASLP 2018]
- For time-varying tasks, time-pooling reduces output resolution



$$f_i = h(w_i * x + b)$$

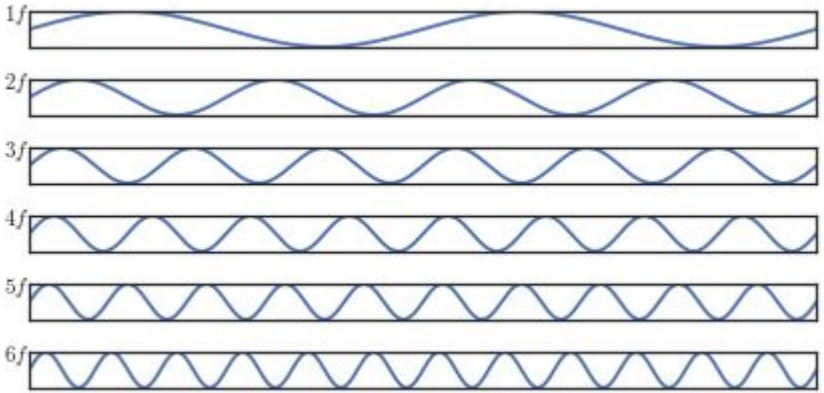
time→

Pitch invariant features

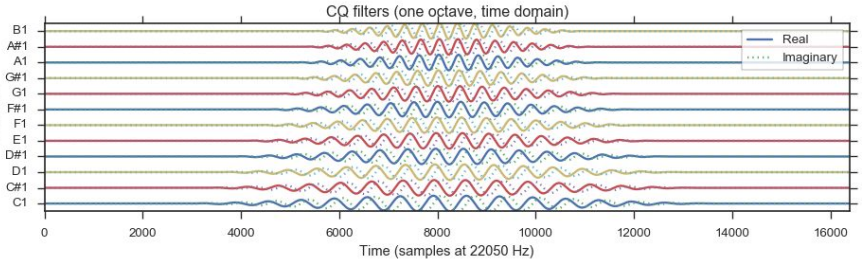
- Constant vertical shift = constant change in pitch
- This requires **log-spaced** frequencies
  - E.g., equal temperament scale:

$$f_{n+1} = 2^{1/12} f_n$$

- But the Fourier basis is **linearly-spaced**
- Mel is only logarithmic at the high end



Constant-Q Transform

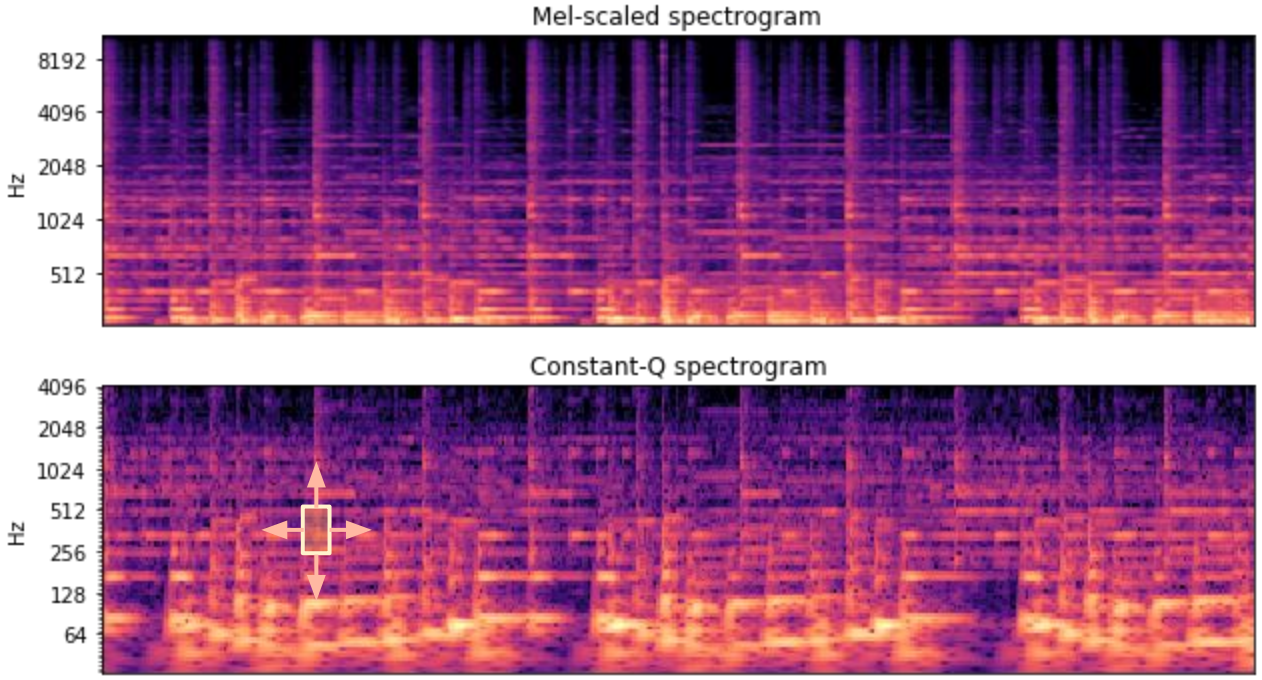


- **Log-scaled** frequency representation

$$f_n = 2^{n/B} f_0$$

- Slightly lossy (compared to FFT)
  - but still ~invertible
  - B bins per octave for analysis (typically $B=12,\ 36$ or 60)
- Window length inversely related to frequency
  - Fixed number of cycles for each basis waveform
- Good representation for both pitch and timbre
  - Not so great for micro-timing and transients

Brown, Judith C. "Calculation of a constant Q spectral transform."
The Journal of the Acoustical Society of America 89.1 (1991): 425-434.


Mel-scaled spectrogram
8192
4096
2048
1024
512
Hz
Constant-Q spectrogram
4096
2048
1024
512
256
128
64
Hz

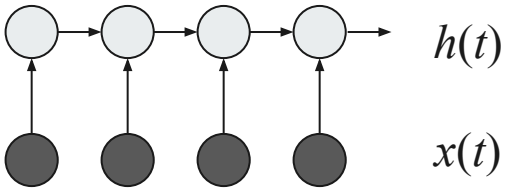
Mel vs CQT

Convolutional and recurrent networks

- Convolutional filters have a fixed (finite) receptive field
  - Finite impulse response (FIR) filter
- Music can have arbitrary time dependencies
  - Depends on tempo, sustain, composition idiom...
- Recurrent networks (RNNs) have unbounded receptive fields
  - Infinite impulse response (IIR) filter
  - Gated models (LSTM, GRU) can improve stability

RNN dynamics

$$h(t) = \sigma(Wh(t\text{-}1) + Ux(t) + b)$$



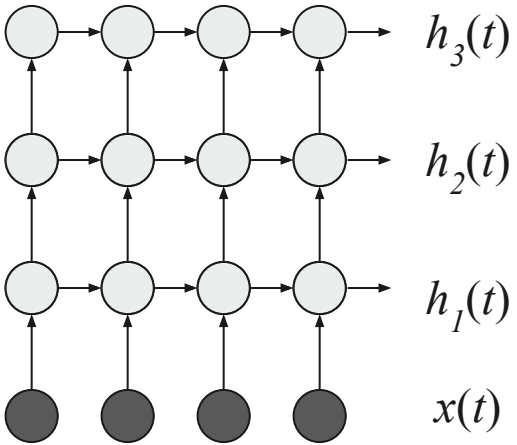
Choice of model should reflect target concepts

- Localized in time? E.g., instrumentation
  - Conv-net
  - Pitch-invariant? Conv2D
- Self-referential in time? E.g., rhythm
  - Recurrent net
- Global? E.g., tagging
  - Dense/MLP?
  - Or CNN/RNN/CRNN + pooling

- Conv layers up front are usually a good idea
- Recurrent models can hide precise timing!
- But RNNs don’t solve everything…
  - Decoding can still be problematic
  - Repetition and long-range interaction are hard

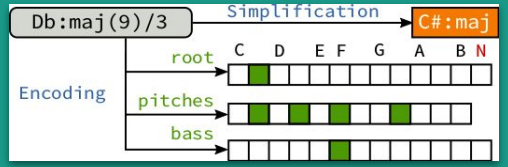
How to use RNNs?

- Locally integrating context?
  - E.g., for rhythm or harmony
  - Use h(t) as a time-series representation
  - This allows stacking: $RNN_1 \rightarrow RNN_2 \rightarrow RNN_3 \rightarrow ...$
- Summarizing global structure?
  - E.g., for encoding variable-length sequences
  - Use h(T) (last frame) as a fixed-dimensional representation
- Does bidirectionality make sense?
  - Usually it's okay!
  - Not viable for causal models (e.g. realtime prediction)

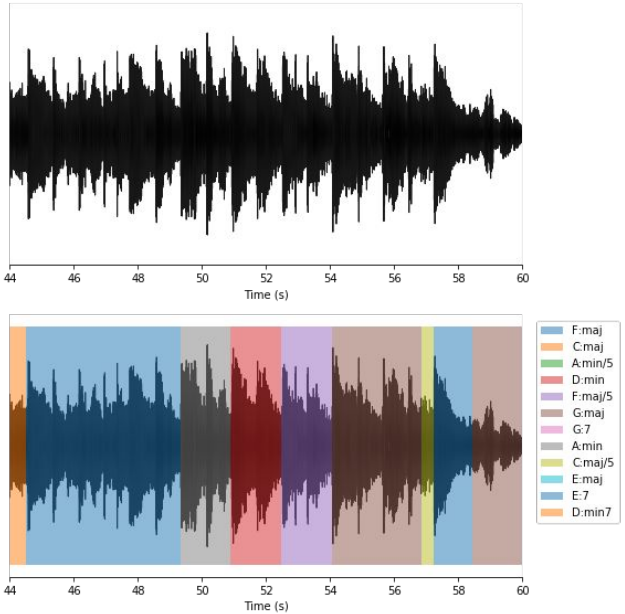


Example 1: Chord estimation

[M. & Bello, ISMIR 2017]



Automatic chord estimation



- Transcribe audio as symbolic chord labels (*A:min*, *C:7*, etc)
- Why RNNs for this task?
  - Chords have variable duration
  - Harmony can be implied over time (e.g. by arpeggio)
  - Decoding ambiguous chords (C:sus4 vs F:sus2) depends on neighbors

Small chord vocabularies

N	
C:maj	C:min
C#:maj	C#:min
D:maj	D:min
…	…
B:maj	B:min

- Standard approach: ***1-of-K*** classification
  - 25 classes: N + (12 × min) + (12 × maj)
  - Frames → chord labels → labeled segments
  - Hidden Markov Models, convolutional networks, etc.
- Classes are approximately balanced
- Implicit assumption:

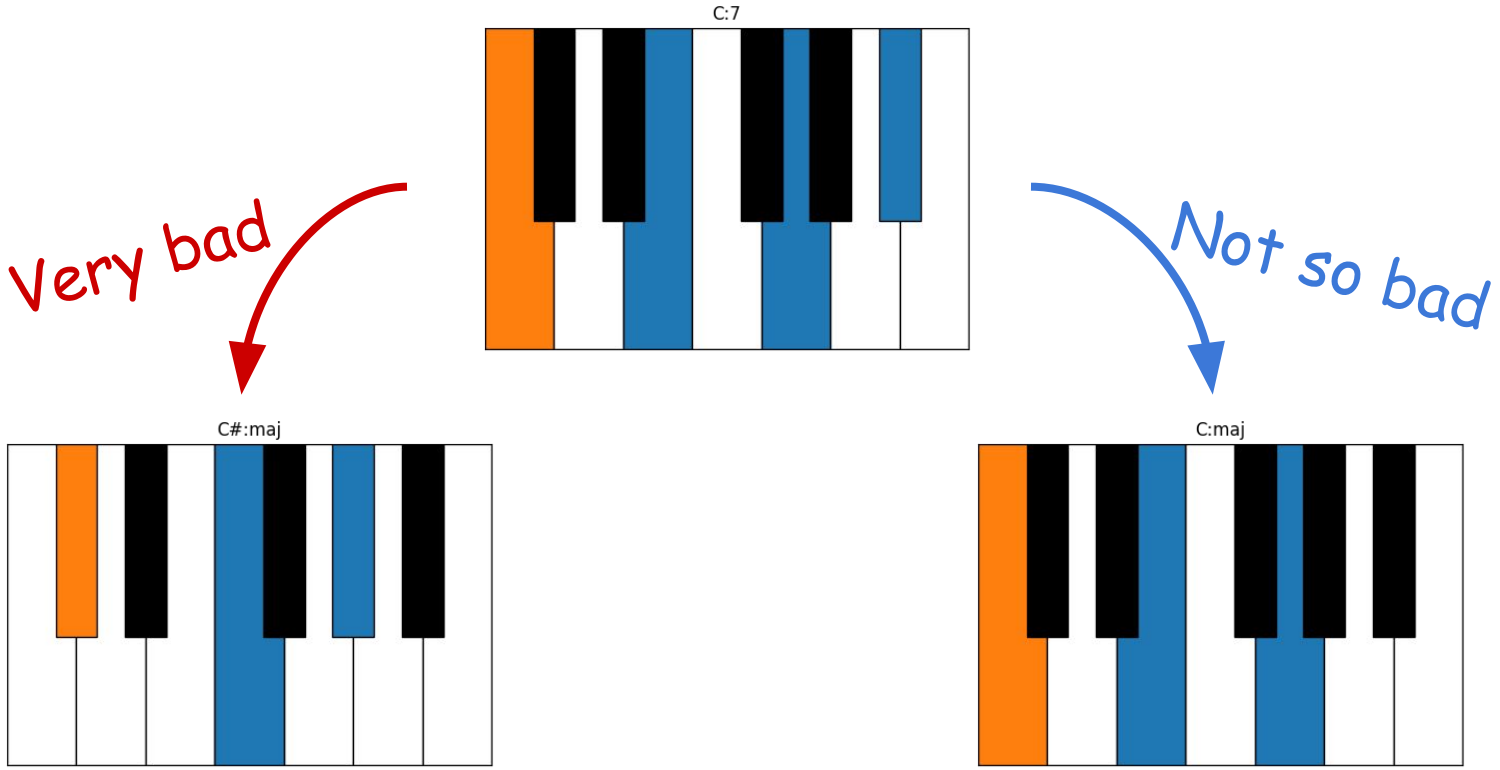
All mistakes are equally bad

Large chord vocabularies

Chord quality	Frequency
maj	53%
min	14%
7	10%
...	
hdim7	0.17%
dim7	0.07%
minmaj7	0.04%

Distribution of the 1217 dataset

- Class distribution is **highly skewed**
- Classes are **not well-separated**
  - Overlapping pitch classes: *C:7 = C:maj + m7*
  - Identical pitch classes: *C:sus4* vs. *F:sus2*
- Improving on **rare classes** often hurts **common classes**

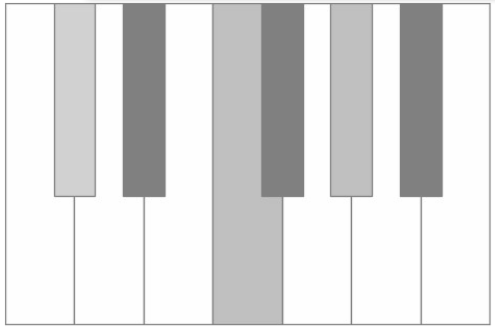

C:7
Very bad
Not so bad
C#:maj
C:maj

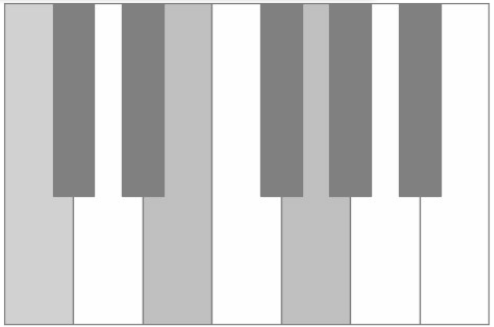
C:7

Ve

bad

Chord space is *structured*

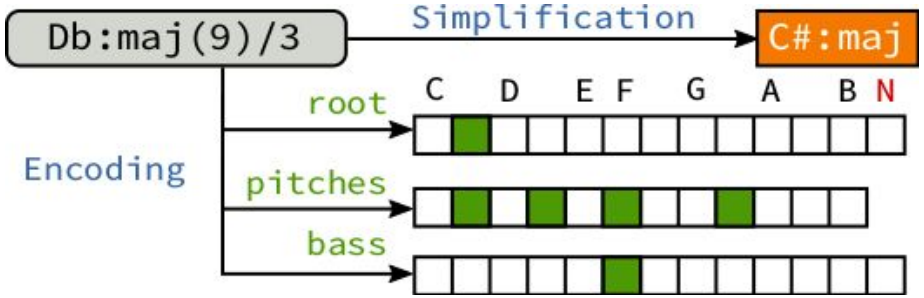




Standard classification setup

- Discard inversions **/3 → NUL**
- Discard added/suppressed notes: **(9) → NUL**
- Simplify above-octave extensions: **G:9 → G:7**
- Resolve enharmonic equivalence: **D ♭ → C#**
- Learn to predict the **simplified chord label**

Structured training

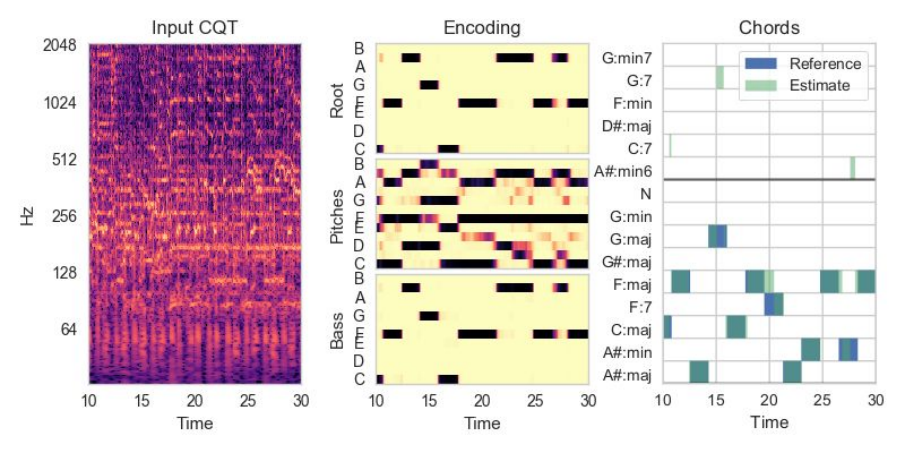


- Represent **chord labels** as **binary encodings**

- Representation is **structured**:
  - **Similar chords** with **different labels** will have **similar encodings**
  - **Dissimilar chords** will have **dissimilar encodings**

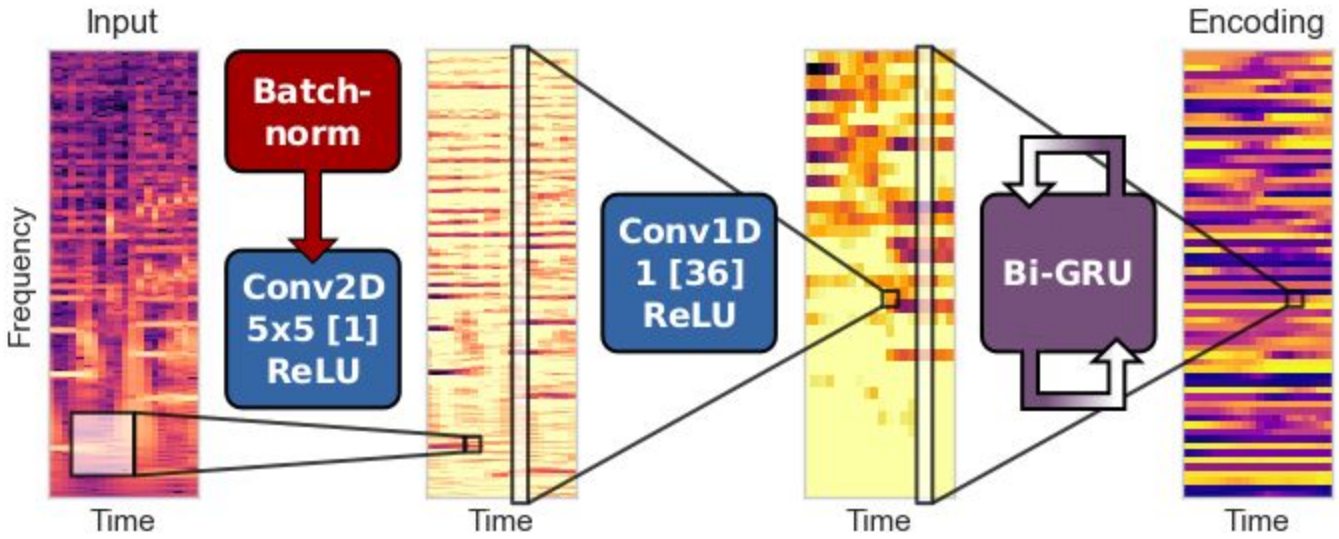
- Predict the **binary encoding**
- Learn to decode into **chord labels**
- Jointly optimize all four outputs:
  root + **pitches** + **bass** + **label**

Model architectures



- **Input**: constant-Q spectral patches
  - 6 octaves, 3 bins per semitone
  - ~10 Hz frame rate

- **Outputs**: (per-frame)
  - **Root** **[multiclass, 13]**
  - **Pitches** **[multilabel, 12]**
  - **Bass** **[multiclass, 13]**
  - **Chords** **[multiclass, 170]**



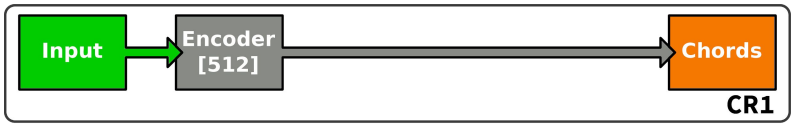
Suppress transients

Encode pitch content

Contextual smoothing

Hidden state at frame t:

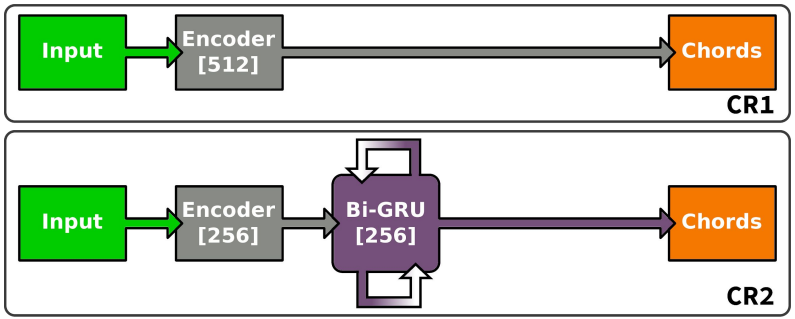
$h(t) \in [-1, +1]^D$



Chords = Logistic regression from GRU state

Frames are independently decoded:

$$\boldsymbol{y(t)} = \text{softmax}(W\, h(t) + \beta)$$



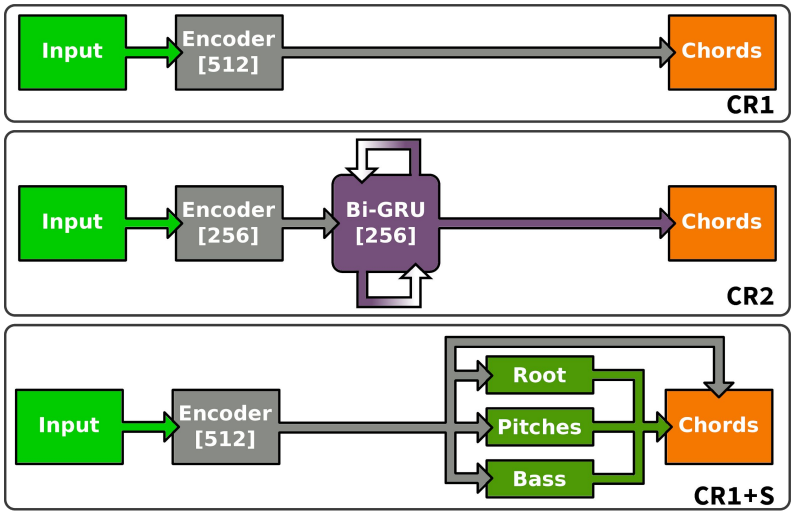
Chords = Logistic regression from GRU state

Chords = GRU + LR

Frames are recurrently decoded:

$$h_2(t) = \text{Bi-GRU}[h](t)$$

$$y(t) = \text{softmax}(W\, h_2(t) + \beta)$$



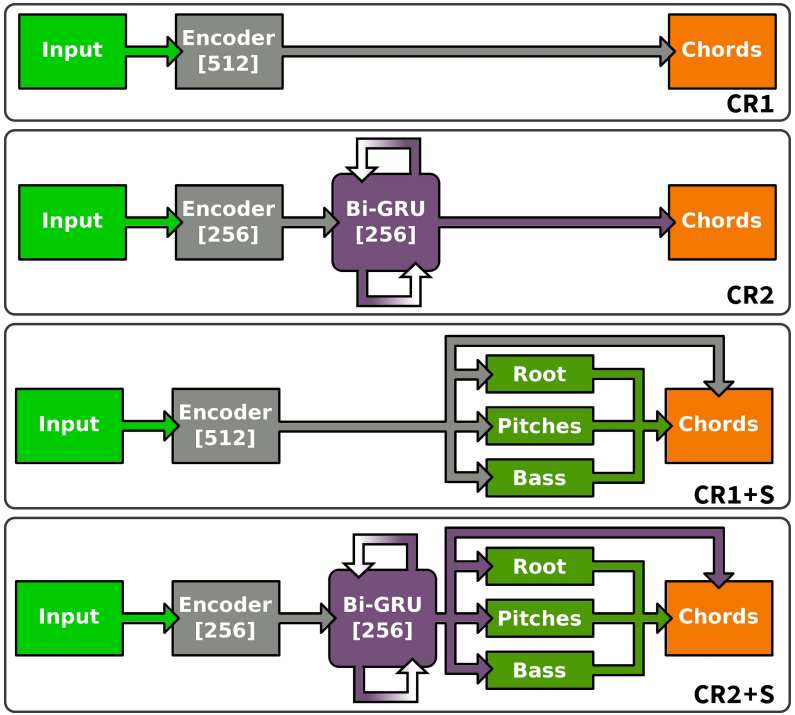
Chords = Logistic regression from GRU state

Chords = GRU + LR

Chords = LR from GRU state + root/pitch/bass

Frames are independently decoded with structure:

$$\mathbf{y(t)} = \text{softmax}(W_r\,\mathbf{r(t)} + W_p\,\mathbf{p(t)} + W_b\,\mathbf{b(t)} + W_h\,h(t) + \beta)$$



Chords = Logistic regression from GRU state

Chords = GRU + LR

Chords = LR from GRU state + root/pitch/bass

All of the above

What about root bias?

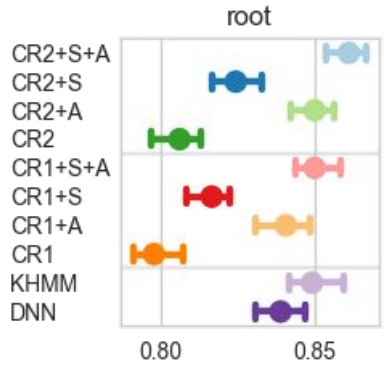
- Quality and root should be independent
- But the data is **inherently biased**
- Solution: **data augmentation**!
  - **muda** [M., Humphrey, Bello, ISMIR 2015]
  - Jointly deform *audio* with *annotations*
- Each training track → ± 6 semitone shifts
  - All qualities are observed in all root positions
  - All roots, pitches, and bass values are observed

http://photos.jdhancock.com/photo/2012-09-28-001422-big-data.html

Evaluation

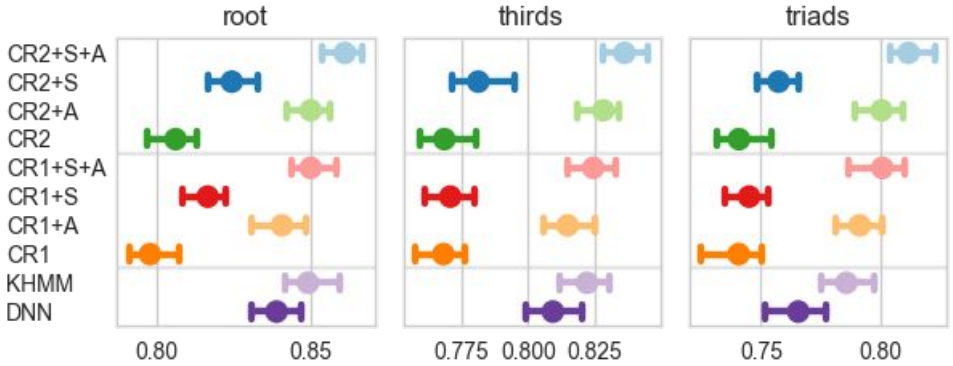
- 8 configurations
  - ± data augmentation
  - ± structured training
  - 1 vs. 2 recurrent layers
- 1217 full-length recordings
  (Billboard + Isophonics + MARL corpus)
  - 5-fold cross-validation
- Baseline models:
  - DNN [Humphrey & Bello, 2015]
  - KHMM [Cho, 2014]

CR1: 1 recurrent layer
CR2: 2 recurrent layers
+A: data augmentation
+S: structure encoding



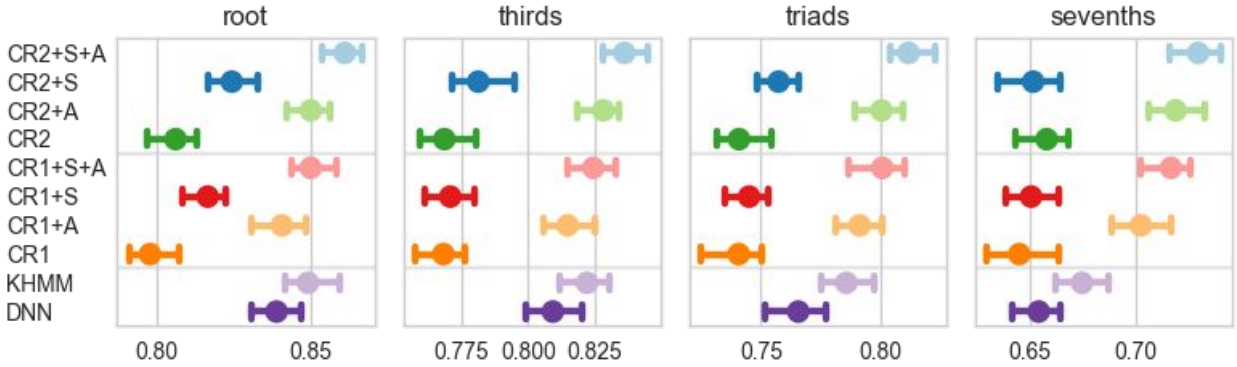
Data augmentation (+A) is necessary to match baselines.

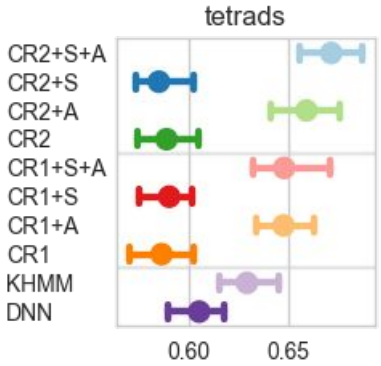
CR1: 1 recurrent layer
CR2: 2 recurrent layers
+A: data augmentation
+S: structure encoding



Structured training (+S) and deeper models improve over baselines.

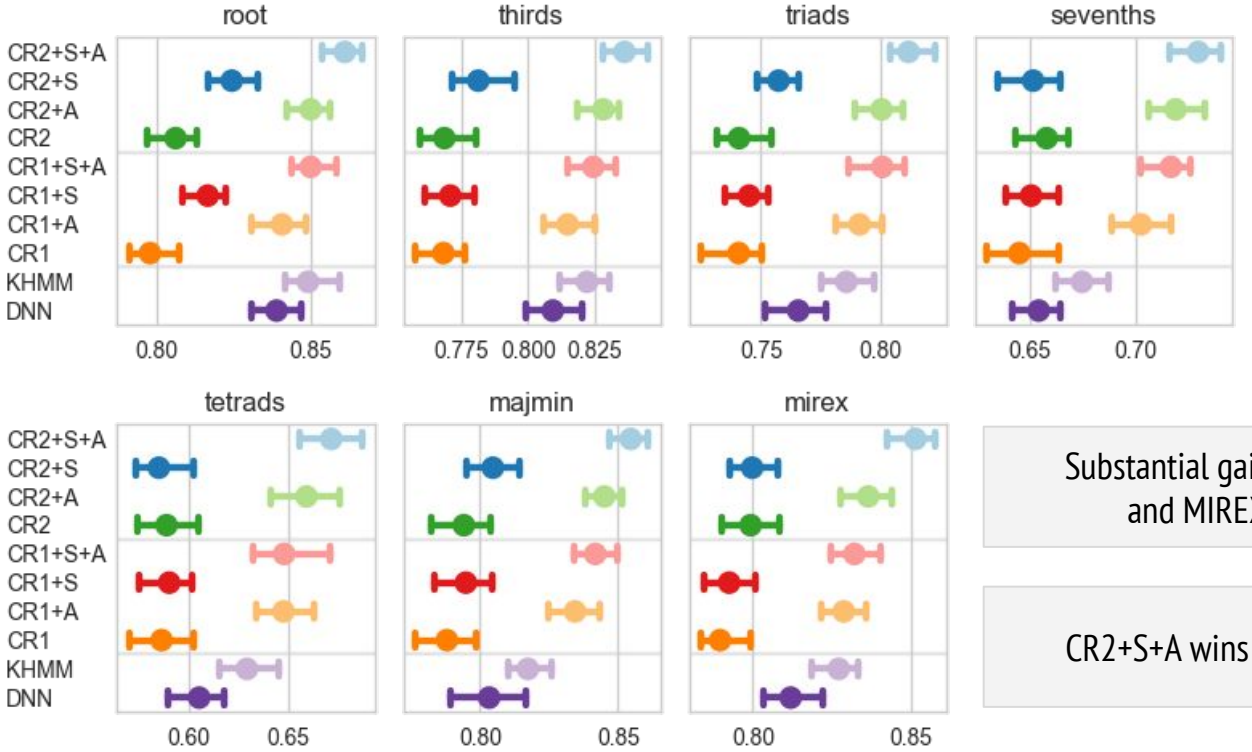
CR1: 1 recurrent layer
CR2: 2 recurrent layers
+A: data augmentation
+S: structure encoding





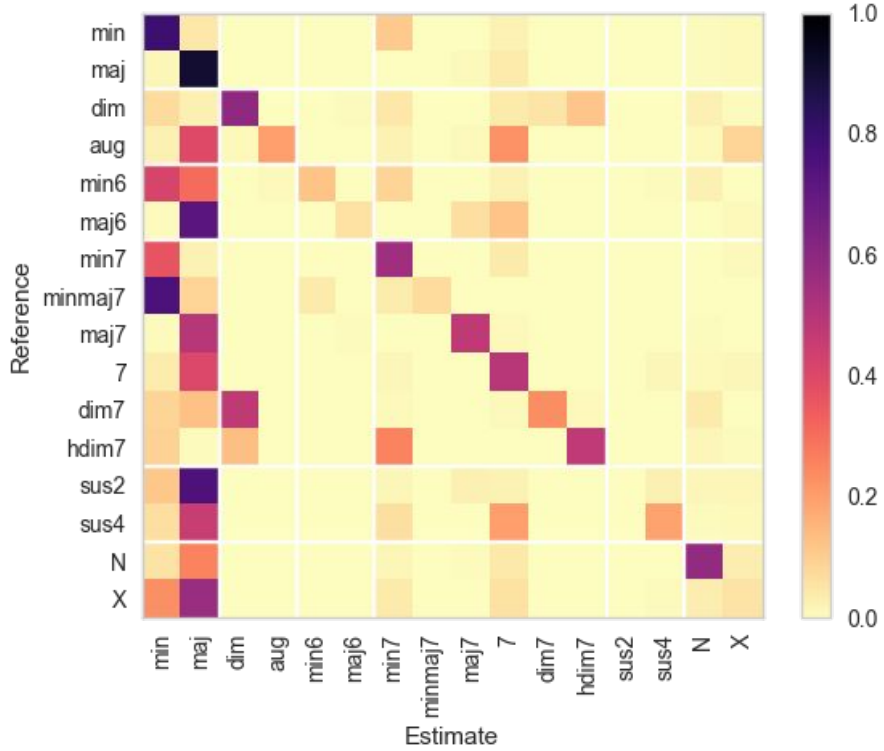
Improvements are bigger on the harder metrics
(7^{th}s and tetrads)

CR1: 1 recurrent layer
CR2: 2 recurrent layers
+A: data augmentation
+S: structure encoding



Substantial gains in maj/min and MIREX metrics

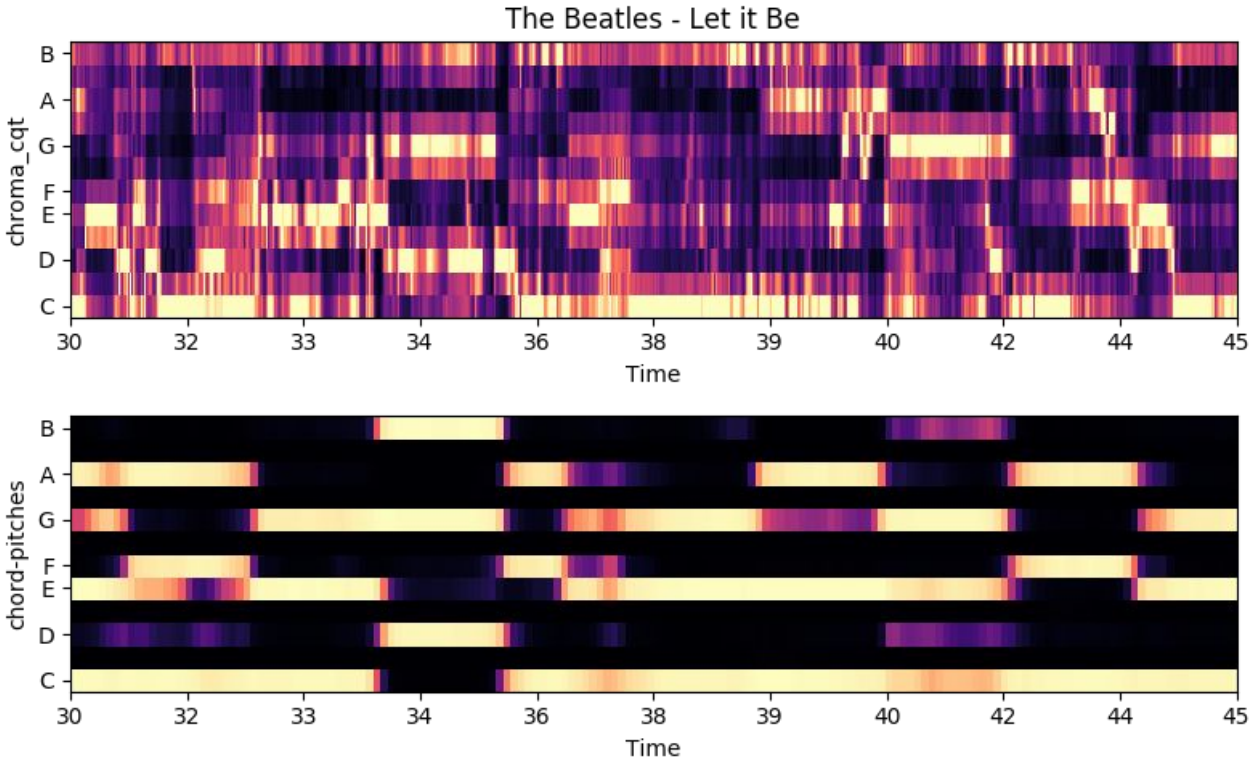
CR2+S+A wins on all metrics



Errors tend toward simplification

Reflects maj/min bias in training data

Simplified vocab. accuracy: 63.6%


The Beatles - Let it Be
chroma_cqt
B
A
G
F
E
D
C
30
32
33
34
36
38
39
40
42
44
45
Time
chord-pitches
B
A
G
F
E
D
C
30
32
33
34
36
38
39
40
42
44
45
Time

Frames are independently decoded with structure:

$$\mathbf{y}(t) = \text{softmax}(W_r\, \mathbf{r}(t) + W_p\, \mathbf{p}(t) + W_b\, \mathbf{b}(t) + W_h\, h(t) + \beta)$$

RNNs don't solve everything!

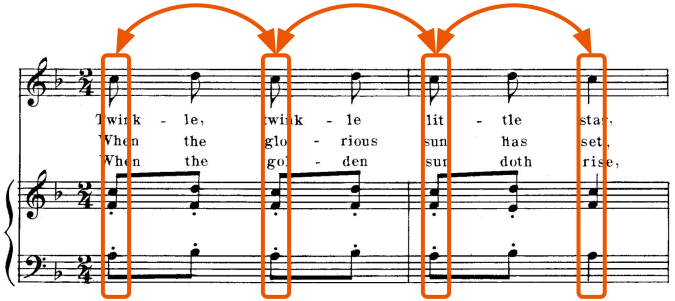
- This model produces conditionally independent class likelihoods for each frame:
  $p(Y_t \mid \boldsymbol{x}) \perp p(Y_{t+1} \mid \boldsymbol{x})$
- Decoding into labels was done independently for each frame
  $y(t) = \text{argmax}_y\, p(Y_t = y \mid \boldsymbol{x})$
- This looks good by the numbers, but can be bad in practice if a sequence is unstable
  ⇒ Flapping between classes: [**C:maj**, **C:7**, **C:maj**, **C:7**, **C:maj**, **...**]
- Quick fix: use **Viterbi** decoding to make piecewise constant predictions

Example 2: Meter tracking

[Fuentes, M., Crayencour, Essid, & Bello, ISMIR 2018]

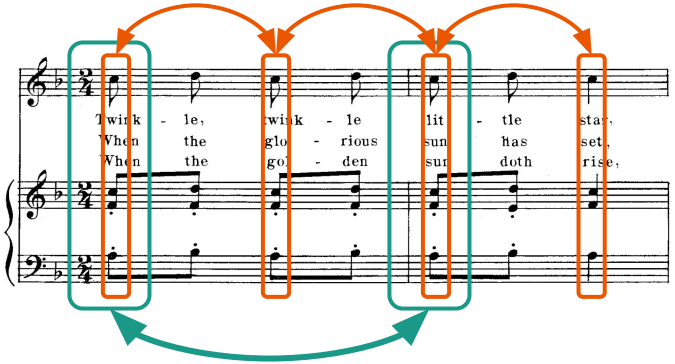
Meter tracking

- **Beat tracking**: find the ¼ notes
  - In most genres, this is well solved by engineering + heuristics
  - Not terribly difficult as long as you can find note onsets



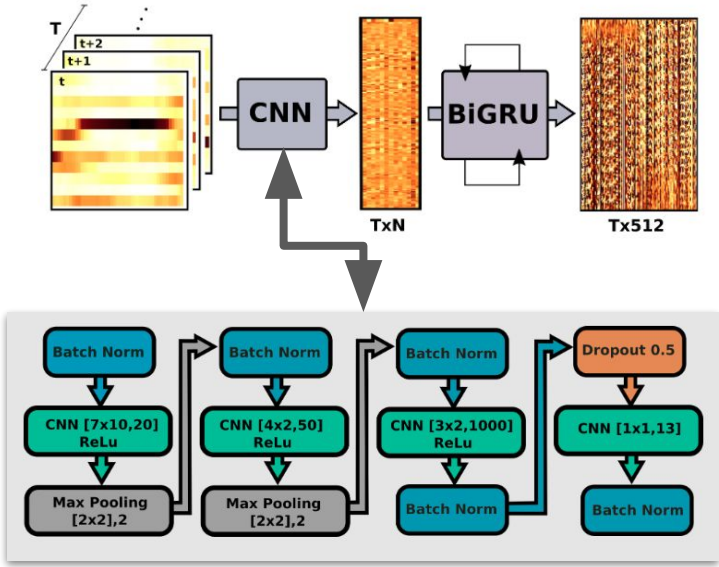
Meter tracking

- **Beat tracking**: find the ¼ notes
  - In most genres, this is well solved by engineering + heuristics
  - Not terribly difficult as long as you can find note onsets
- **Meter tracking**: find the bar lines (**downbeats**)
  - Implicitly: detect the time signature
  - Much harder problem! Depends on long-term structure, local changes, etc.
  - Errors in **beat tracking** can propagate upward



General framework (Fuentes et al., 2018)

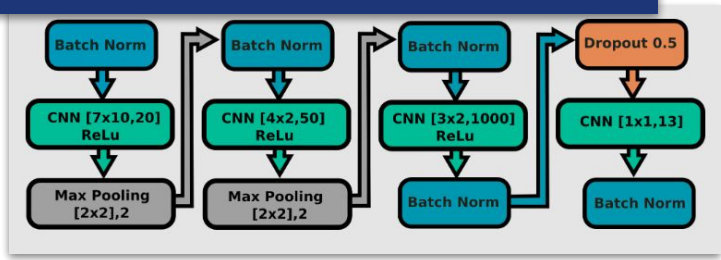
1. Use a pretty good beat tracker
   a. Optionally: sub-divide to 1/16 notes
2. Compute features along the estimated time grid
3. Model architecture
   a. RNN vs CRNN
4. Decoding
   a. Independent vs Viterbi



General framework (Fuentes et al., 2018)

1. Use a pretty good beat tracker
   a. Optionally: sub-divide to 1/16 notes
2. Compute features along the estimated time grid
3. Model architecture
   a. RNN vs CRNN
4. Decoding
   a. Independent vs Viterbi

All of these had been tried independently before, but the full space of combinations had not been evaluated rigorously!

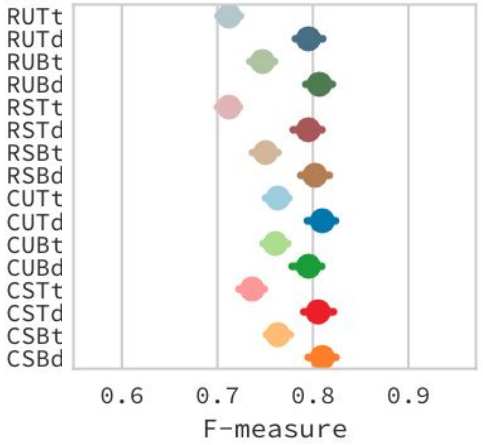


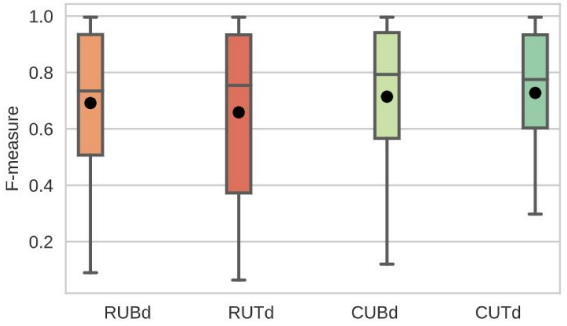
Experiment

- Combined 8 datasets of western popular/jazz/classical music (~50 hours of audio)
- Leave-one-dataset-out evaluation paradigm
- Extra comparison:
  - Can we use a structured output representation instead of binary **{downbeat, not downbeat}**?
  - Encode each time point by its position within the bar: {¼, 2/4, ¾, 4/4, ⅓, ⅔, 3/3, …}
  - Intuition: this could give the RNN more structure to work with ⇒ more discrimination and gradient flow

Meter tracking findings

- RNN was sensitive to the input timing grid, CRNN not so much **(T vs B)**
  - RNN did worse with subdivided inputs (longer sequences between downbeats)
  - CRNN performed comparably with both
- CRNN generally outperformed the pure RNN **(R vs C)**
  - Differences on average were small, but the CRNN was more robust
- Viterbi decoding always helped **(t vs d)**
- Structured output did not help, but it didn't hurt either **(U vs S)**





Some other cool examples

FolkRNN - https://folkrnn.org/

- Sturm, et al. 2015-
- 3-level LSTM model on “ABC” notation of songs
- Essentially a language model on melody transcriptions
- Recorded and published album of synthetic folk tunes:
  - https://soundcloud.com/oconaillfamilyandfriends

```
X: 1
T: Scottish Horse, The
Z: Weejie
S: https://thesession.org/tunes/12696#setting21449
R: jig
M: 6/8
L: 1/8
K: Amix
A|c3 ecA|d>ef a2 f|edc cAe|B3 B<BA|
c3 ecA|d>ef afd|ecA B2e|A3 A<A:|
|:e|a2 e A<Ae|d2 f a2 f|e>dc cAe|B3 B<Be/d/|
[1 ceA eaA|faA dfA|ceA B2 e|A3 A<A:|
[2 c3 ecA|d>ef afd|ecA B2 e|A3 A<A||
```

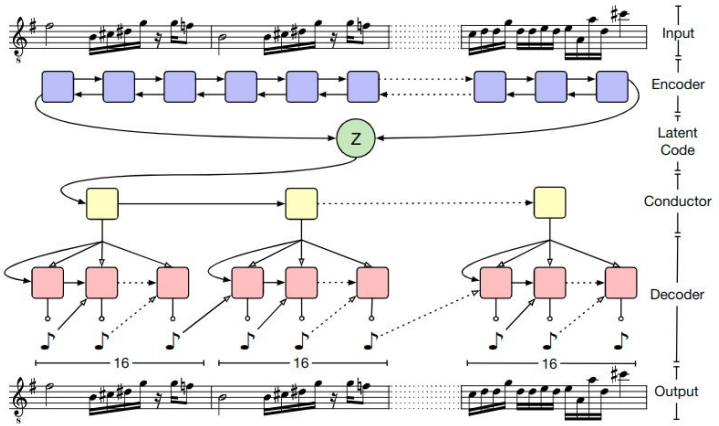
After the album was mastered, we printed several dozen "white-label" CDs to send to reviewers. We wanted the album reviewed as if it were a standard album of folk music, and to avoid the bias that can result when a person believes a creative product comes from a machine.[8] With ethics approval granted from the Faculty Research Ethics committee of Kingston University, UK,[9] we created the following story about the album, and printed it on stickers applied to the while-label CDs:

> During the Summer of 2017, three generations of the Ó Conaill family gathered at the family home in Roscommon to celebrate the life and legacy of Dónal Ó Conaill. The late father and grandfather to the Ó Conaill family, Dónal was quietly dedicated to the tradition, and known for collecting local tunes without names, which he passed on to his family. His daughters, Caitlín and Ùna, are joined by their children and family friends to make a recording of the best of these tunes, along with some of Dónal's personal favourites.
>
> contact: caitlinoconaill@gmail.com
>
> 11 tracks (digital only) Promo Use Only. Not for resale. Release date: July 1, 2018.

http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1248565&dswid=-3767

MusicVAE

- A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music
  - Roberts, Engel, Raffel, Hawthorne, & Eck, 2018
- Learn a fixed-dimensional embedding of symbolic melody
- Decoder has multiple hierarchical levels to account for multi-level structure
- Variational auto-encoder turns the embedding into a probability distribution ⇒ can sample new melodies



Closing remarks

- Music is inherently sequential, and recurrent models are a natural choice
- Music is not Markovian, so you successful application takes some ingenuity
- Long-range interactions are challenging to model: hierarchy may help!
- Most of the work goes into properly packaging your input / output representations to make learning easy

Thanks!

Questions?

brian.mcfee@nyu.edu

https://bmcfee.github.io

Supplementary material

Can’t stop, won’t stop...

Resources

Things you can use!

Software

(shameless self-promotion)

- Librosa - Audio feature extraction in python
- Mir_eval - Standard metrics for music tasks
- Muda - Musical data augmentation
- JAMS - JSON annotated music specification
- Pescador - Data streaming for iterative learning
- Pumpp - audio features / label encoding for deep learning

Software

- Sonic-visualiser - Desktop app for viewing / analyzing audio content
- MSAF - Music structure analysis framework
- Madmom - Music analysis / high-level pre-trained models (beats, chords, etc)

Data sets

With audio

- MedleyDB (2014-)
  - Separated sources (stems) with audio
  - Melody and instrument annotations
- Free Music Archive (2016/7-)
  - Pre-computed features and audio (106K tracks)
  - Genre tags and play counts
- MTG Datasets
  - Too many to keep track of…
  - Instruments, transcription, query-by-humming…
- MIREX
  - Annual evaluation campaign
  - Development sets are available

Data sets

With features

Without audio*

- ISOPhonics (2009-)
  - Chords, keys, beats, structure for Beatles + a few others
- Million Song Dataset (2011)
  - Pre-computed features
  - Tags, Lyrics, Collaborative filter, Cover songs, Pairwise similarities, …
- Lakh Midi Dataset (2016)
  - 177K MIDI files
  - 45K linked to MSD
- SALAMI (2011-)
  - Structure annotations from multiple annotators

Where to find MIR

- The International Society for Music Information Retrieval
  - http://ismir.net
  - Links to mailing list, satellite group, journal, etc.
- On Slack:
  - https://mircommunity.slack.com/

What makes tagging hard?

- Open vocabulary
  - Spelling errors
  - Concept overlap
- Weak annotation (1)
  - Some concepts are localized (instruments)
  - Some are global (genre)
  - Some are totally irrelevant (*favorite*, *seen live*)
- Weak annotation (2)
  - You never directly observe negative examples
  - Annotators may be precise, but not complete
  - Discriminative learning algorithms are weak in this setting

What makes instrument detection hard?

- Can be perceptually difficult for humans
  - Was that a tenor or baritone sax?
  - A viola or a cello?
- Some classes have ill-defined boundaries
  - Electric guitar?
  - Synthesizer?!
- Requires careful annotation
- And large evaluation sets to test for spurious correlations
  - if **genre=rock** then
    instruments=(voice,guitar,bass,drums)

What makes chord recognition hard?

- Large vocabulary space
  - Major, minor
  - augmented, diminished
  - [sixths], [sevenths]
  - [suspended chords]
  - Extensions
  - Inversions
- Inter-annotator disagreement
- Biased observation
  - Maj > 50% of observations, minmaj7 < 0.01%
- Structured output space
  - C:maj <-> C:7 vs. C:maj <-> C#:maj
  - C:maj <-> C:maj/3

What makes beat tracking hard?

- Handling irregular timing, tempo drift
- Soft-attack instruments
  - E.g., bowed strings
- Implied pulse
- Biased training data
- Too many evaluation criteria

What makes structure analysis hard?

- Severe inter-annotator disagreement
- Probably ill-defined across genres
  - Structure in classical is very different from jazz
  - Or pop, or hip-hop...
- Lack of support for hierarchy
- Evaluation criteria don’t quite capture everything we want

What makes cover detection hard?

- Performers vary significantly
- Much of the signal is carried by lyrics (for vocal music)
  - But sung lyric analysis is a hard nut to crack
- Or melody
  - See above
- Limited access to audio

What makes melody extraction hard?

- What distinguishes melody from any other pitch contour?
- Polyphony can be deadly
- Annotation is expensive and difficult
- Again, access to data...

What makes automatic transcription hard?

- Everything that's hard about melody, chords, rhythm
- Plus source separation
- Plus access to large evaluation collections
- And not all music makes sense to transcribe
  - In fact, a large portion of modern music was never transcribed to begin with