JOHANNES KEPLER
UNIVERSITY LINZ

SPECIAL TOPICS

Audio and Music Processing - Lecture 5: Beats and Tempo Estimation
344.032
KV, 2h, SS2020

Jan Schlüter
Institute of Computational Perception

OVERVIEW

- goals
  - understand beat tracking
  - understand tempo estimation
- topics
  - what are beats?
  - building blocks of a beat detection algorithm
  - two simple beat tracking approaches
  - state-of-the-art approaches
  - tempo estimation: comes for free

BASICS

DEFINITIONS (1)

- **pulse** - the periodic recurrence of strokes, vibrations or undulations
- **tatum** - the period of the fastest pulse train perceived by a listener, or, put differently, the shortest durational values in a music performance that appear on purpose, not randomly
- **tactus** / **beat** - the most prominent metrical level (we tend to tap our feet/clap our hands to the music at this speed); defines tempo
- **measure** - related to the harmonic change rate, or length of a rhythmic pattern

DEFINITIONS (2)

- ▷ **tatum**
- ▷ **tactus**
- ▷ **measure**

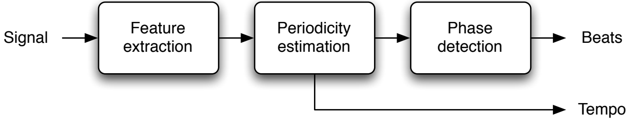
DEFINITIONS (3)

- we focus on the **beat** (“tactus”)
- popular synonyms include “tempo”, “meter”, “rhythm” as well as “groove”
- the **tempo** of a piece of music is determined by the **duration** of the **beat**
- tempo is measured in “beats per minute”, **bpm** for short
- informally, the topic of this lecture are algorithms which “clap their (virtual) hands to the beat”

BEAT TRACKING (1)

- the process of computing the **timing** and **placement** of the beat is called **beat tracking**
- **beat tracking** subdivides into three somewhat related problems:
  - determine the **periodicity**
  - extract the **tempo**
  - determine the **phase**
- these subproblems may be tackled simultaneously, or separately

BEAT TRACKING (2)

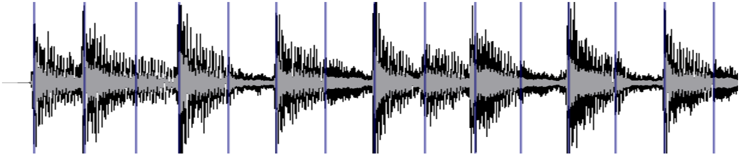


- **feature extraction** (onsets, rhythmic information, chord changes, amplitude envelopes, spectral features)
- **periodicity estimation** (determine the periodicity of the extracted features, via histograms, autocorrelation, comb filters, multi-agent trackers)
- **phase detection** (some methods produce phase information during periodicity estimation already, others need to determine the phase of the periodic signal)

A HISTOGRAM-BASED EXAMPLE BEAT TRACKER

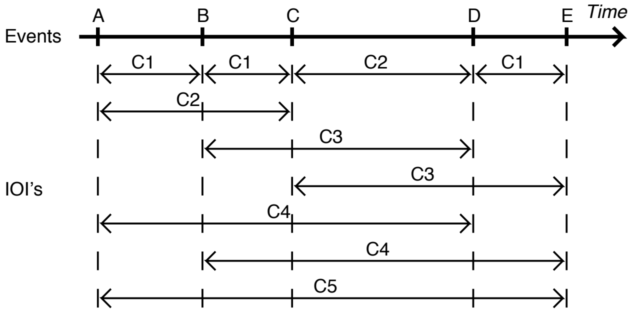
FEATURE EXTRACTION

- our beat tracker will be based on onset times, which we already know how to extract
- a short refresher:
  - compute the STFT
  - calculate spectral flux
  - normalize
  - adaptive peak-picking
- the result is a (more or less accurate) list of onset times

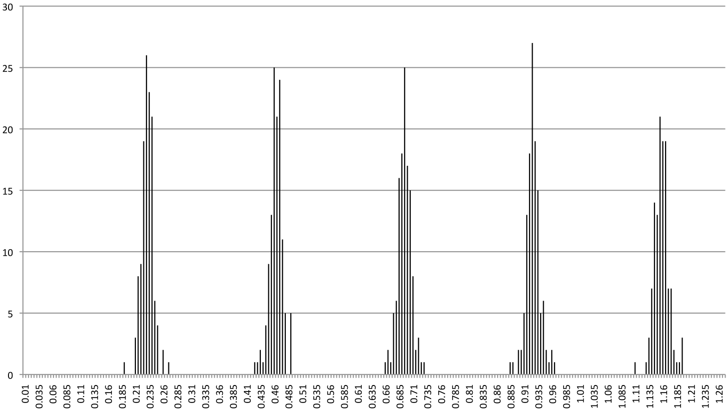


PERIODICITY ESTIMATION

- compute the **Inter Onset Intervals**, "**IOI**" for short
- in most cases, IOIs are multiples of each other
- the periodicity of the beats correspond to one of the IOIs

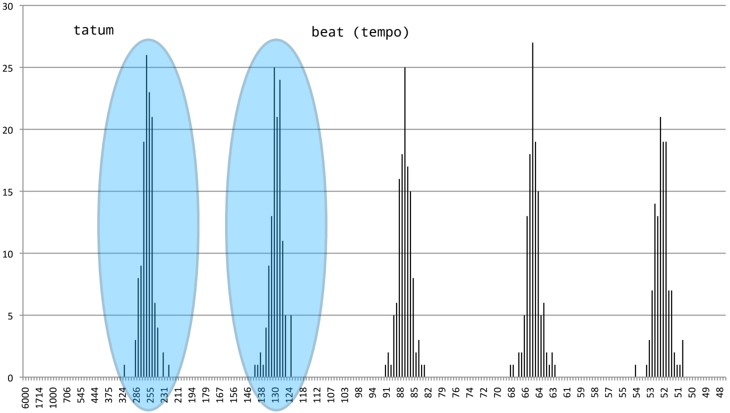


IOI HISTOGRAM (1)



what could be the actual tempo here?

IOI HISTOGRAM (2)

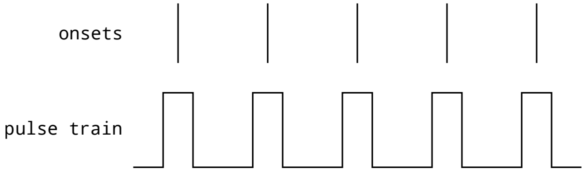


PEAK SELECTION

- this is a problem with beat tracking
- most **common** errors are **octave errors**
- this means reporting a tempo either **half** or **twice** as big as the **correct** tempo
- sometimes even humans disagree on the “correct” tempo
- a simple heuristic is selecting periodicity peaks corresponding to tempi in the range [60, 200] [bpm]
- **instead of just counting** IOIs, also look at the **energy** at each of the two **onsets** for each IOI, weigh the IOIs accordingly

BEAT LOCATION (1)

- create an artificial **pulse train**, based on the extracted tempo hypothesis
- a **pulse train** is very similar to a regular square wave
- **cross-correlate** the pulse train with the result of the onset detection starting at different offsets



BEAT LOCATION (2)

- the offset for the pulse train where the cross-correlation is maximal is taken as the first beat
- for all successive beats:
  - go forward in time one beat period
  - search for an onset around this position in time
  - if we found an onset, select it as the next beat
  - if not, the approximate position is taken as the next beat

DISCUSSION

- we looked at a very (*very*) simple beat tracking algorithm
- it heavily depends on good onset detection
- the following is true for most higher-level algorithms: if your **low-level** features are **sensitive to noise**, everything **built upon them** is likely to be **very sensitive to noise** as well
- it also makes a very **limiting assumption**, namely that the **tempo is constant** for the whole piece, which is not necessarily true

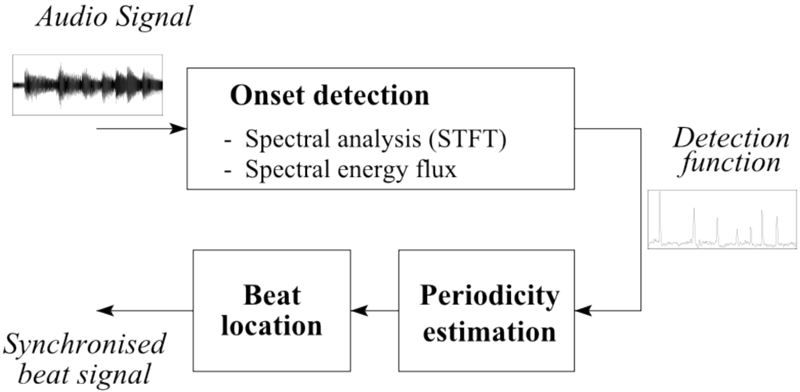
RELAXATION

LOCAL LINK
YOUTUBE

AUTOCORRELATION

- a beat tracker based on autocorrelation [1]
- the following slides describe a beat tracker that may constitute a good basis for the exercise track, combined with an onset detection approach of your choice

SYSTEM OVERVIEW



ONSET DETECTION

- we will not go into too much detail here
  (last lecture covered that very extensively)
- spectral differencing is used as the detection function
- logarithmic **perceptual correction** is applied to the amplitude
- the detection function can be median filtered - all **values below the median** in a window are **set to zero** to cope with noise
- you may also **use** the **detection** function **directly**, if the signals you process are not too noisy

PERIODICITY ESTIMATION (1)

- $d[t]$ is the detection function, $r[\tau]$ the autocorrelation, τ is also called “lag”

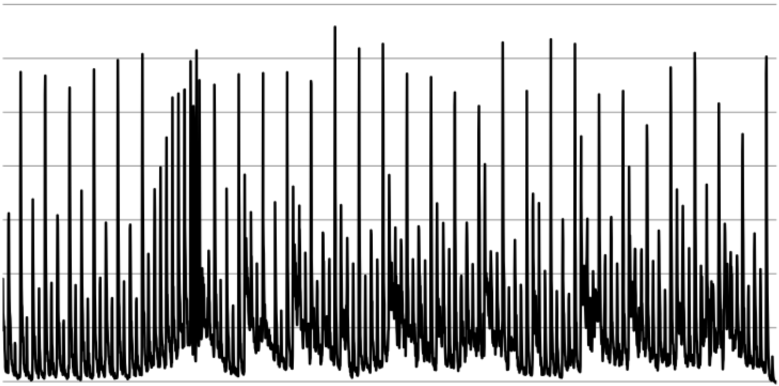
$$
\begin{aligned}
r[\tau] = &\quad \sum_{t=0}^{N} d[t+\tau] \cdot d[t] \\
\tau \in &\quad [\tau_{start}, \tau_{end}]
\end{aligned}
$$

- autocorrelation is used to detect periodicities in the detection function
- the detection function can be seen as a **quasi-periodic** and **noisy pulse-train**
- under the assumption that the tempo is in the range [60, 200] [bpm], we only have to compute the autocorrelation for all τ falling in this range

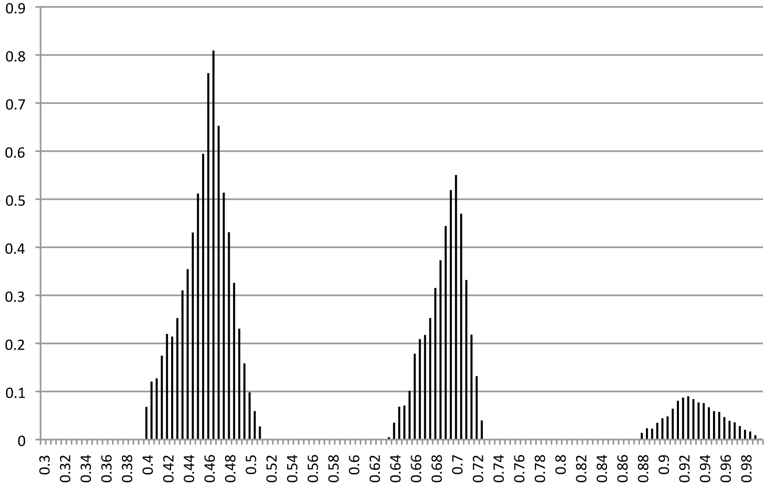
PERIODICITY ESTIMATION (2)

- as the detection function represents a function of the magnitude at onset times, autocorrelation should find the most prevalent periodicity
- further analysis regarding multiplicity relationships between peaks in the autocorrelation may improve the results

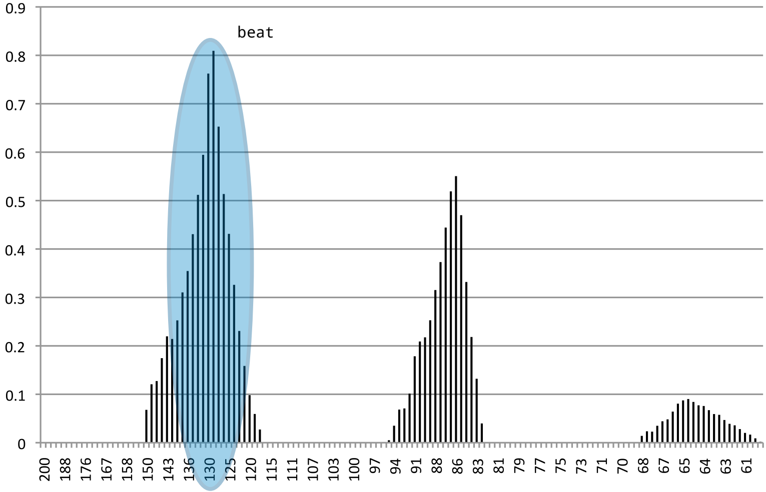
EX: DETECTION FUNCTION $d(\cdot)$



EX: AUTOCORRELATION OF $d(\cdot)$



EX: AUTOCORRELATION OF $d(\cdot)$



WINDOWED BEAT LOCATION

- **only the first few seconds** are cross-correlated with a pulse train of the extracted tempo
- the **time-index** where the **cross-correlation is maximal** is taken as the **first beat location**
- successive beats are computed by adding a beat period and searching for a peak in the detection function near this location in time; if none is found, the approximate position is taken directly
- **after** placing the **last beat** in this window, the tempo **on the next few seconds** is calculated, and the **tracking continues** with this **new beat period**

WHAT WE LEARNED SO FAR

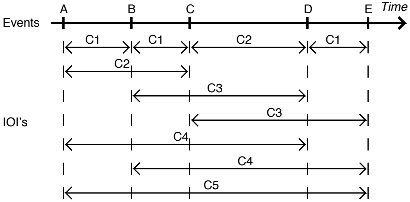
- using autocorrelation on a (median filtered) detection function improves the situation over IOI histograms
- using autocorrelation on a (median filtered) detection function directly, emphasizes more salient events, and is slightly more noise-tolerant
- use a windowed approach in a first attempt to cope with changing tempo
- still, more sophisticated methods are needed for beat tracking robust to changing tempo

RELAXATION

YOUTUBE

MULTIPLE AGENTS

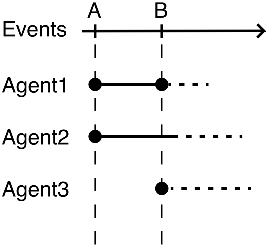
- onset detection based on the signal envelope [4]
- this detects only very salient onsets, which are more likely to correspond to beats
- the original focus of this system was on symbolic data
- later refinements used more sophisticated onset detection functions



- compute IOIs
- cluster IOIs
- after clustering we are left with a set of different tempo hypotheses

INITIALIZATION

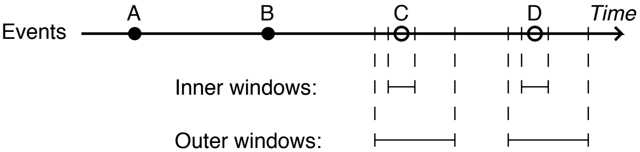
- for each event in the first few seconds, as well as for each tempo hypothesis, an agent is created
- if a beat would be predicted by two or more agents, only the one with the higher score is retained
- in the figure on the right, there are two assumptions:
  - the initialization period spans events A and B
  - there are two tempo hypotheses, T_1, T_2



- Agent 1 is created at Event A with Tempo T_1
- Agent 2 is created at Event A with Tempo T_2
- Agent 3 is created at Event B with Tempo T_2
- $\dots$

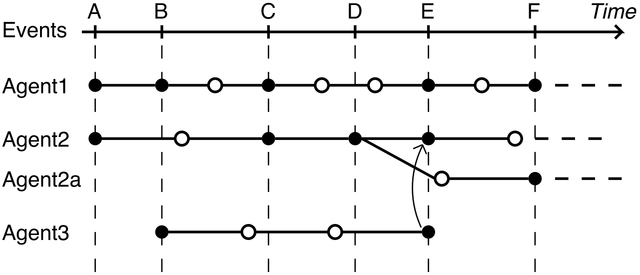
EVENT PROCESSING

- after initialization, each event is processed by each of the agents, allowing each to consider the event as a beat
- each agent has a prediction of the next beat time, because of its own tempo hypothesis
- predicted beats are enclosed by two windows:
  - inner window: the deviation the agent will accept without hesitation
  - outer window: the deviation the agent will accept as an additional possibility



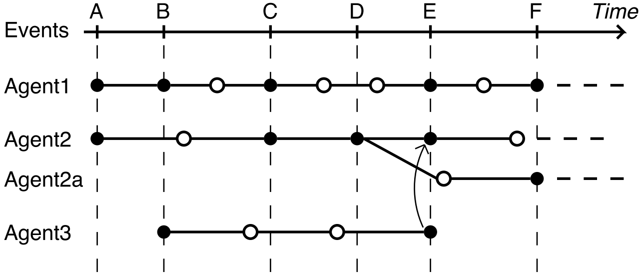
JKU

SCENARIO #1



- if an event falls **outside both** windows, it is simply **ignored**

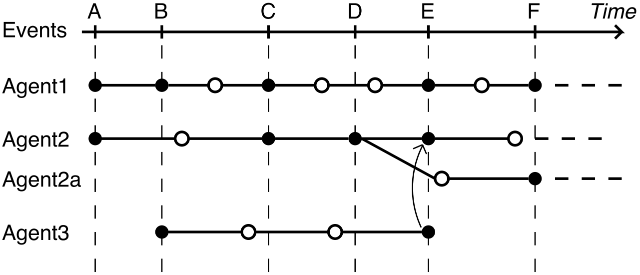
SCENARIO #2



- if an event falls in the **inner window**, it is **accepted** as a beat
- the tempo hypothesis is updated as a fraction of the difference between predicted and accepted beat time
- if an event does **not** fall into the **first** predicted beat window, but a later one, missing beats are **interpolated** (hollow circles)

JKU

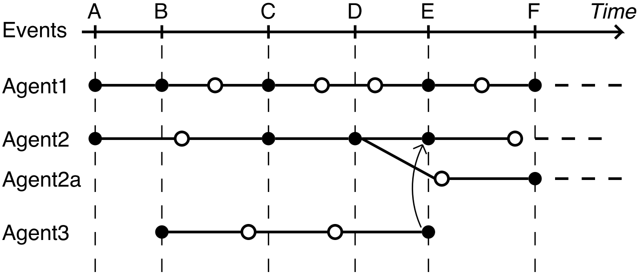
SCENARIO #3



- if the event falls in the **outer window**, the event is **accepted** as a beat, as in scenario #2
- additionally, a **new agent** that does **not accept** the event as a beat is created, to include both possibilities (above: Agent 2a)

JKU

DEDUPLICATION



- agents that approximately agree in **both** tempo and phase need to be **pruned**
- only the agent with **higher** evaluation **score** is **retained** (Agent2 $\|$ Agent3) $\rightarrow$ Agent2

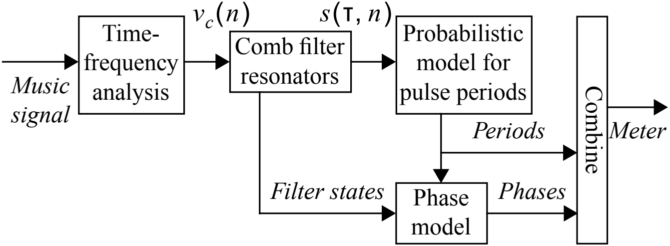
JKU

AGENT SELECTION

- agents have a high score if their predictions are good
- the closer a prediction is to an actual event, the better it is
- depending on the **salience** of the onset, the score is higher
- “saliency” refers to how noticable / pronounced an item is
- the agent with the highest final score will be returned in the end

COMB FILTERS

an approach based on comb filters and hidden markov models [6]

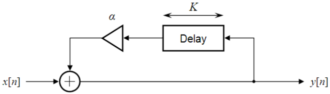


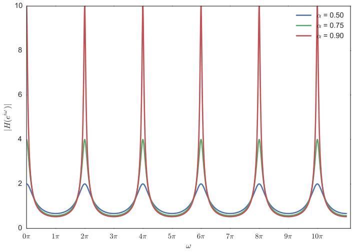
ACCENTS/ONSETS

- STFT with window size 23ms and 50% overlap
- filterbank with 36 triangular filters distributed on a critical band scale between [50, 20000] [Hz]
- non-linear compression (μ-law, similar to logarithm)
- low-pass filtering over time (cutoff at 10 Hz)
- differentiation over time
- half-wave rectification
- features try to **measure** the **degree of accentuation** in the musical signal

PERIODICITY (1)

- a comb filter **adds** part of a **delayed** version of **itself** to itself
- the delay causes destructive and constructive **interference**
- the frequency response looks a bit like the teeth of a **comb**, hence the name
- its effects are very **similar** to **auto-correlation**, **but cheaper** to compute





JKU

PERIODICITY (2)

- **multiple** comb filters with different delays corresponding to **different tempi** are used to find the delay that elicits the **strongest** comb-filter **response**
- choosing the right periodicity estimation method is **not** the **key** problem in beat tracking
- more important are measuring the **degree** of accentuation, as well as modelling **higher level** musical knowledge

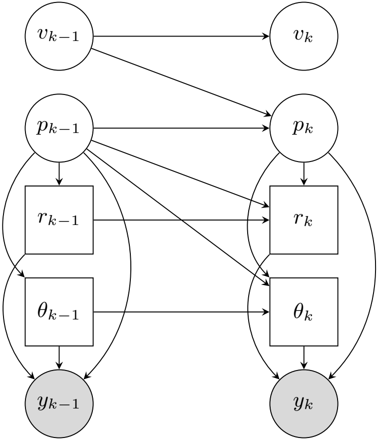
ESTIMATION

- use a **HMM** (Hidden Markov Model) for estimation
- HMMs are simple dynamic bayes nets
- HMMs are widely used for temporal pattern recognition
- here a HMM is used to describe the simultaneous evolution of four processes:
  - **hiddens** - periods of tatum, tactus and meter
  - **observables** - vector of energies of comb-resonators
- the phases are estimated after periods have been established

PROBABILISTIC MODELS

other probabilistic models can be used for beat tracking as well [7]

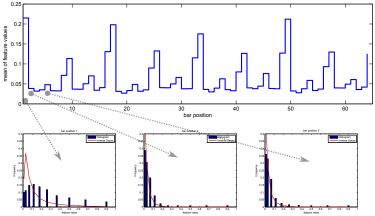
- y: observations, based on "SuperFlux" features
- v: tempo
- p: position inside the bar
- r: rhythmic pattern
- θ: meter (duple/triple)



JKU

PROBABILISTIC MODELS

- observation model is learned from data
- it models the expected feature values for each bar position (64 per bar)
- for each feature observation, it gives the likelihood for each position in the bar, depending on the rhythmic pattern
- tracking demo video



- the **top** curve depicts the mean feature value for each bar position
- the histograms and curves in the **bottom** depict histograms and fitted inverse gaussian distributions for bar positions 1, 2 and 5

RELAXATION

YOUTUBE

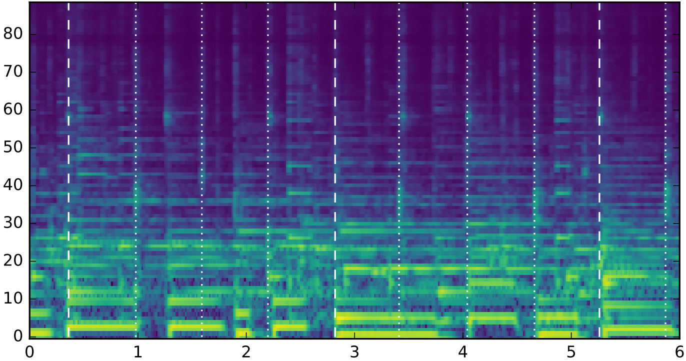
LEARNING A BEAT TRACKER

- a machine learning approach [3]
- don't define the tracker manually
- very similar to the learned onset detector from last lecture
- slightly different features:
  - 3 STFTs instead of 2, with different window sizes
  - first order spectral differences are computed and given to the network as additional inputs
  - logarithmic filterbanks with 3, 6, and 12 bands per octave
- **beat and downbeat locations** are output by the system directly
- **meter, tempo and phase** are computed by a dynamic bayesian network that infers these quantities jointly
- after training, the network has seen $\sim 65[\mathrm{h}]$ of music

JKU

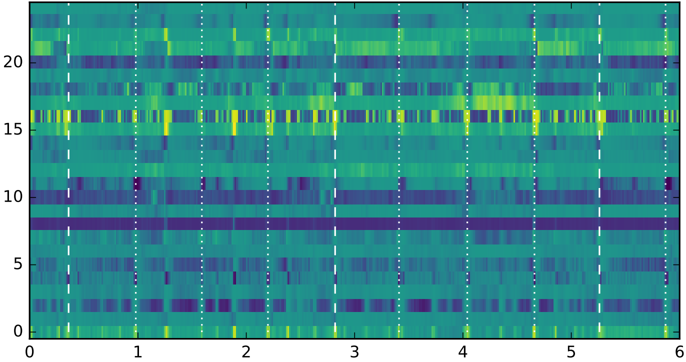
WHAT IS LEARNED ?

One of the inputs, with beats and downbeats annotated:



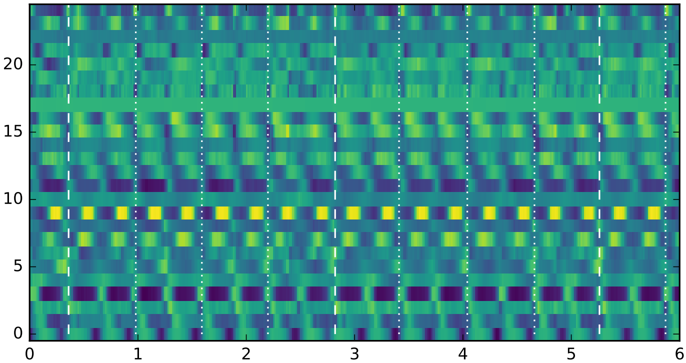
WHAT IS LEARNED ?

The unit activations after the first hidden layer:



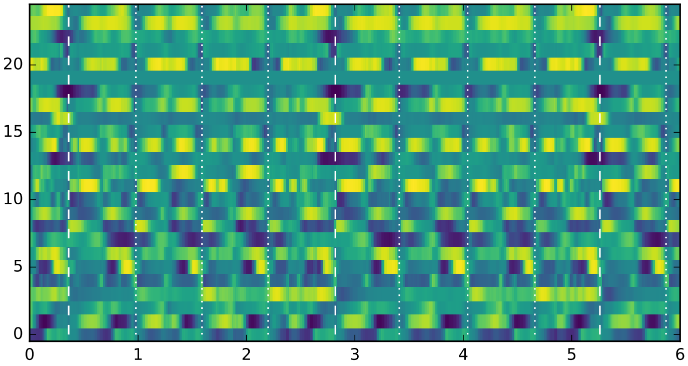
WHAT IS LEARNED ?

The unit activations after the second hidden layer:



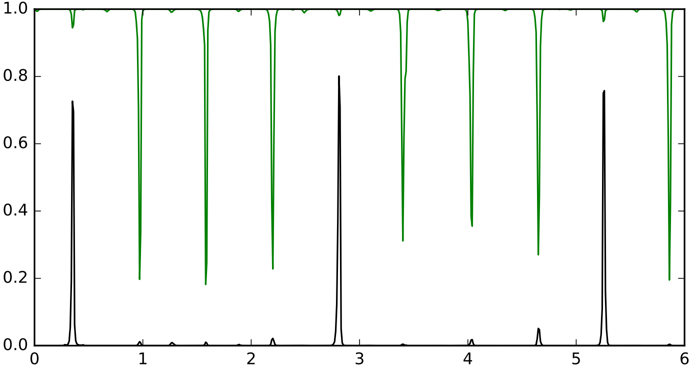
WHAT IS LEARNED ?

The unit activations after the third hidden layer:



WHAT IS LEARNED ?

The activations of the final hidden layer:



APPLICATION

- one of the applications of the learned beat tracker is control
- it has been used to control a robotic drummer
  ROBOD, Signal Processing Cup
- what we will see in the next video is the network's responses to the input at different layers
- we will also see the estimated tempo of the dynamic bayesian network
- ▷ **demo**

CONCLUSIONS

CONCLUSIONS

- beat tracking methods are utilized in a variety of applications, such as:
  - automatic accompaniment
  - beat-informed effects processing
  - automated alignment of two musical pieces
  - score alignment
  - music classification
- **offline** is much **easier** than **online**
- **non-causal** vs. **causal**

JKU

MAIN SOURCES

- some of the main sources for this lecture were not explicitely cited, because they would have to be cited everywhere
- there are **lots** of references in the cited paper's own reference sections
- don't forget about the papers with state-of-the-art approaches!

REFERENCES I

[1] Miguel A. Alonso, Gaël Richard, and Bertrand David.
Tempo and beat estimation of musical signals.
In *ISMIR 2004, 5th International Conference on Music Information Retrieval, Barcelona, Spain, October 10-14, 2004, Proceedings*, 2004.

[2] Sebastian Böck.
Onset, beat, and tempo detection with artificial neural nets.
Master's thesis, TU München, 2010.

REFERENCES II

[3] Sebastian Böck, Florian Krebs, and Gerhard Widmer.
Joint beat and downbeat tracking with recurrent neural networks.
In *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, pages 255–261, 2016.

[4] Simon Dixon.
Automatic extraction of tempo and beat from expressive performances.
Journal of New Music Research, 30(1):39–58, 2001.

REFERENCES III

[5] Fabien Gouyon, Anssi Klapuri, Simon Dixon, M. Alonso, George Tzanetakis, C. Uhle, and Pedro Cano.
An experimental comparison of audio tempo induction algorithms.
IEEE Trans. Audio, Speech & Language Processing, 14(5):1832–1844, 2006.

[6] Anssi Klapuri, Antti J. Eronen, and Jaakko Astola.
Analysis of the meter of acoustic musical signals.
IEEE Trans. Audio, Speech & Language Processing, 14(1):342–355, 2006.

REFERENCES IV

[7] Florian Krebs, Sebastian Böck, and Gerhard Widmer.
Rhythmic pattern modeling for beat and downbeat tracking in musical audio.
In *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4-8, 2013*, pages 227–232, 2013.

[8] Eric D Scheirer.
Tempo and beat analysis of acoustic musical signals.
The Journal of the Acoustical Society of America, 103(1):588–601, 1998.