

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>                                | <b>3</b>  |
| 1.1      | Difficulties of Beat Tracking . . . . .            | 3         |
| 1.2      | Applications of Beat Tracking . . . . .            | 5         |
| <b>2</b> | <b>History of Beat Tracking</b>                    | <b>7</b>  |
| <b>3</b> | <b>Basic Principles of Human Rhythm Perception</b> | <b>11</b> |
| <b>4</b> | <b>Foundations of Machine Learning</b>             | <b>16</b> |
| 4.1      | Sequence Modeling . . . . .                        | 16        |
| 4.2      | Data Representation . . . . .                      | 16        |
| 4.3      | Architectures . . . . .                            | 16        |
| 4.3.1    | Recurrent Neural Networks . . . . .                | 17        |
| 4.3.2    | Convolutional Neural Networks . . . . .            | 17        |
| 4.3.3    | Temporal Convolutional Networks . . . . .          | 17        |
| 4.3.4    | Trellis Networks . . . . .                         | 20        |
| 4.4      | Performance Measure . . . . .                      | 20        |
| 4.5      | Regularization Techniques . . . . .                | 21        |
| 4.6      | Optimization . . . . .                             | 21        |
| 4.7      | Hyperparameter Optimization . . . . .              | 21        |
| 4.8      | Multi Task Learning . . . . .                      | 21        |
| 4.9      | Valididation . . . . .                             | 21        |
| <b>5</b> | <b>Beat Tracking System</b>                        | <b>22</b> |
| 5.1      | Dataset . . . . .                                  | 22        |
| 5.2      | Data Preprocessing . . . . .                       | 23        |
| 5.2.1    | Chroma Representation . . . . .                    | 24        |
| 5.3      | Feature Learning . . . . .                         | 25        |
| 5.4      | Temporal Decoding . . . . .                        | 25        |
| 5.4.1    | Transition Model . . . . .                         | 25        |
| 5.4.2    | Observation model . . . . .                        | 26        |
| <b>6</b> | <b>Evaluation</b>                                  | <b>27</b> |
| 6.1      | Performance measures . . . . .                     | 27        |
| 6.2      | Saliency maps . . . . .                            | 28        |
| 6.3      | Network training . . . . .                         | 28        |
| 6.4      | Labeling . . . . .                                 | 28        |
| <b>7</b> | <b>Conclusion</b>                                  | <b>28</b> |
| <b>8</b> | <b>Appendix</b>                                    | <b>29</b> |
| 8.1      | Hyperparameter . . . . .                           | 29        |

## General notes

- Goal of supervised learning: predict not yet observed input
- Motivation: “*It is probable that no universal beat tracking model exist which does not utilise a switching model to recognize style and context prior to application.*” [1, Collins2006]
- Instead of calculating the STFT as for input features discover a good set of features by representation learning. (“*Learned representations often result in much better performance than can be obtained with hand-designed representations.*” [2, Goodfellow2016])
- Common association between sequence modeling and recurrent networks: “*Given a new sequence modeling task or dataset, wich architecture should one use?*” [3, Bai2018]
- Certain convolutional architectures can reach state-of-the-art accuracy in audio synthesis (e.g. Google WaveNet [4, Oord2016])
- Do TCNs outperform LSTM architectures in this particular sequence modeling task?
- Contrast of the two statements:
  - “*Feature extraction is an essential step towards efective and accurate beat/downbeat positions extraction.*” [5, Khadkevich2012]
  - “*The system shows state-of-the-art beat and downbeat tracking performance on a wide range of different musical genres and sryles. It does so ny avoiding hand-crafted features such as harmonic changes, or rhythmic patterns, but rather leans the relevant features directly from audio.*” [6, Boeck2016b]

# 1 Introduction

- A fundamental research topic in music information retrieval is the automatic extraction of beat locations from music signals.
- The aim of a beat tracker is to recover a sequence of time instants from a musical input that are consistent with the times when a human might tap their foot. [7]
- Beat tracking is an important initial step in computer emulation of human music understanding, since beats are fundamental to the perception of (Western) music.
- The goal of beat tracking is to construct a computational algorithm capable of extracting a symbolic representation which corresponds to the phenomenal experience of beat or pulse in a human listener.
- The primary information required for beat tracking is the onset times of musical events, i.e., musical notes and percussive sounds, and this is sufficient for music of low complexity and little variation in tempo.
- The interpretation of beat is one of the most fundamental aspects of musical intelligence.
- A naive approach to describe the rhythm of musical data is to specify an exhaustive and accurate list of onset times, maybe together with some other musical features characterising those events, e.g., durations, pitches and intensities. However, such a representation lacks abstraction. Diverse media used for rhythm transmission suffer a trade-off between the level of abstraction and the comprehensiveness of the representation. Standard (Western) music notation provides an accepted method for communicating a composition to a performer, but it has little value in representing the interpretation of a work as played in a concert. On the other hand, an acoustic signal implicitly contains all rhythmic aspects but provides no abstraction whatsoever [8].
- The main goal in automatic rhythm description is the parsing of acoustic events that occur in time into the more abstract notions of metrical structure, tempo and timing. A major difficulty is the inherent ambiguity of rhythm, as discussed in section 1.1.
- How well does the proposed model generalise to different musical styles?

## 1.1 Difficulties of Beat Tracking

It is difficult to reliably extract high-level rhythm related features from musical excerpts having properties such as [Quinton2016]

- The principal reason that beat tracking is intrinsically difficult is that it is the problem of inferring an original beat structure that is not expressed explicitly. The

degree of beat tracking difficulty is therefore not determined simply by the number of musical instruments performing a musical piece it depends on how explicitly the beat structure is expressed in the piece. However, it is very difficult to measure its explicitness because it is influenced from various aspects of the songs.

- The larger the number of syncopation, the greater the difficulty of beat tracking (quantitative measure: power-difference measure [9])
- The main reason that different tendencies with regard to the explicitness with which their beat structure is indicated.
- There is not necessarily a specific sound that directly indicates the position of beats. In fact, a musical beat may not directly correspond to a real sound, there may even be no signal on a beat.
- absence of a clear rhythmic structure (Classical music)
- soft onsets
- blurred note transitions (e.g. classical music dominated by string instruments)
- heavy syncopation
- expressive timing (e.g. rubato playing)
- the problem of 'octave errors' (detecting double or half time the rate of the ground truth)
- acoustic signals consist of sound of various kinds of instruments
- the onsets of notes are difficult to obtain, unlike the case of MIDI signals, where there is no such problem.
- a beat may not directly correspond to a real sound. It is a perceptual concept that human feels in music.
- Multiple interpretations of beats are possible at any given time
- There is no simple relationship between polyphonic complexity - the number and timbres of notes played at a single time - in a piece of music, and its rhythmic complexity or pulse complexity [10]. There are pieces and styles of music which are texturally and timbrally complex, but have straightforward, perceptually simple rhythms; and there also exist music which deal in less complex textures but are more difficult to rhythmically understand and describe.
- The complexity of grouping harmonic partials together to form notes, and determining the onset times of those.
- For this reason Cemgil et al. [11] define music transcription as the extraction of an *acceptable* music notation.

- There is no canonical form for representing rhythm, and lacking this ground truth, it is difficult, if not impossible, to provide a meaningful quantitative comparison of the various computer systems. Therefore we need a common database of test music labelled with the ground truth on which the systems are tested.

## 1.2 Applications of Beat Tracking

- Beat tracking can be used to automate the time-consuming tasks that must be completed in order to synchronize events with music.
- Video and audio editing (visual track can be automatically synchronized with an audio track using beat tracking)
- Stage light control. In live performances, beat tracking is useful in the control of stage lighting by a computer. For instance, various properties of lighting such as color, brightness, direction, and effect can be changed in time to the music.
- Visualization for example time-grid in audio sequencers.
- Musical interaction systems [12]
- Content-based audio effects, for multimedia or interactive performances or studio post-production
- Beat-driven real-time computer graphics.
- Temporal segmentation to higher level MIR tasks such as chord estimation for harmonic description [13].
- Long-term structural segmentation of audio [14]
- performance analysis investigates the interpretation of musical works, e.g., the performer's choice of tempo and expressive timing.
- In audio content analysis beat tracking is important for automatic indexing and content-based retrieval of audio data, such as in multimedia databases and libraries.
- Automatic transcription and score extraction from performance data
- Music similarity
- Beat tracking can provide computers the ability to participate intelligently in live performances in real time and join the ensemble.
- synchronisation of a musical performance with computers or other devices
- Commercial devices already exist that attempt to extract a MIDI clock from an audio signal, indicating both the tempo and the actual location of the beat. Such MIDI clocks can be used to synchronize other devices such as drum machines or audio effects, enabling a new range of beat-synchronized audio processing.

- Automatic playlist generation, where a computer is given the task to choose a series of audio tracks from a track database in a way similar to what a human deejay would do. The track tempo is a very important selection criterion in this context, as deejays will tend to string tracks with similar tempi back to back. Furthermore, deejays also tend to perform beat-synchronous crossfading between successive track manually, slowing down or speeding up one of the tracks so that the beat in the two tracks line up exactly during the crossfade. This can easily be done automatically once the beats are located in the two tracks.

## 2 History of Beat Tracking

The following chapter is a summary (background) of related work that fostered the development of automatic beat tracking in musical audio.

- Overview: Signal processing → data-driven approaches
- The task of beat tracking has been well established over the last twenty-five years.
- The first analysis step is the creation of a feature list, i.e. the parsing, of filtering, of the data at hand into a sequence of features. These features range from note onset features (as time, duration and pitch), to frame-based signal features, and they convey the predominant information relevant to rhythmic analysis.
- The extraction of note onsets times for rhythmic analysis is ubiquitous in the literature. An exhaustive overview of musical onset detection can be found in [15].
- Some systems do not focus on note onsets and their features, but refer to a data granularly of a lower level of abstraction, namely frames. A frame is a short chunk, typically 10 ms of audio, from which both time and frequency domain features can be computed. Consecutive frames are usually considered with some overlap for smoother analyses, and the analysis step, the hop size, equals the frame size minus the overlap.

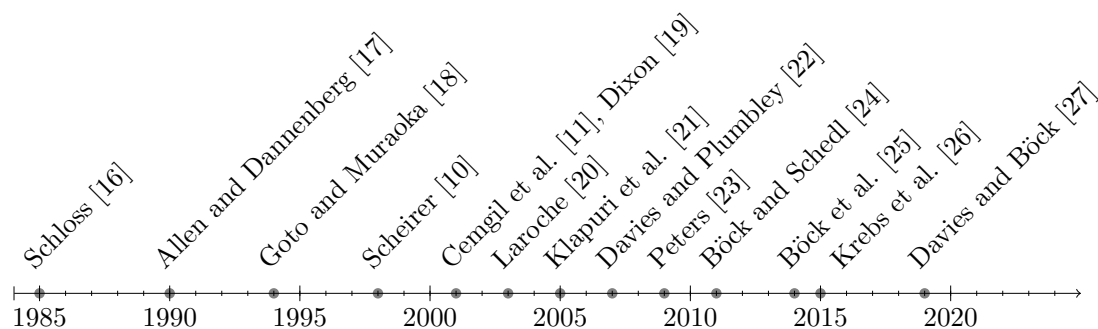


Figure 1: Timeline with the history of beat tracking.

The earliest work on automatic extraction of rhythmic content from audio is published 1985 in the percussion transcription system of Schloss [16]. Onsets are detected as peaks in the slope of the amplitude envelope, where the envelope is defined to be equal to the maximum amplitude in each period of the high-pass filtered signal, and the period defined as the inverse of the lowest frequency expected to be present in the signal. The main limitation of the system is that it requires parameters to be set interactively.

Allen and Dannenberg [17] extended the musical concept of beat to include two aspects, namely period and phase. Based on that concept, they built a method that uses real-time beam search to allow the beat tracker to consider several possible stages at

once. They use a credibility measure so that at any given time there is a set of active states that represent the most credible interpretations for the performance encountered so far. However, the system's reliance on MIDI limited the input source to electronic instruments, and moreover limited its application.

Goto and Muraoka [18] introduce the first worth mentioning beat tracking system which could process music played on ensembles of a variety of instruments. However, they restricted their system to rock and pop music in which drums maintain the beats. The system leverages the fact that for a large class of popular music, a bass drum and a snare drum usually occur on the strong and weak beats, respectively. It manages multiple agents that track beats according to different strategies in order to examine multiple hypothesis in parallel. All hypotheses are gathered and the most reliable one is selected as the output. This enables the system to follow beats without losing track if them, even if some hypothesis became wrong. Assumptions were made; between 65 and 185 BPM, time-signature is 4/4, tempo stays almost constant. In following developments, Goto present a beat tracking system for both music with and without drum-sounds [28]. It uses frequency-domain analysis to detect chord changes, which are assumed to occur in metrically strong positions. This is the first system to demonstrate the use of high level information in directing the lower-level beat tracking process. The high level information is specific to the musical style, which is a major limitation of the system.

Scheirer [10] figured out from psychoacoustic demonstration on beat perception, that amplitude envelopes from a small number of broad frequency channels are sufficient information to rhythmically analyse musical content. He concludes that a rhythmic processing algorithm should treat frequency bands separately, combining results at the end, rather than attempting to perform beat tracking on the sum of filterbank outputs. This leads him to the use of a small number of bandpass filters and banks of parallel comb filters, which function as tuned resonators, to perform periodicity analysis. In the next processing step, the phase of the musical signal is extracted by examining the internal state of the delays of the comb filters. Finally, the phase and the period is used to estimate the beat times as far into the future as desired. One problem with the system is that in order to track tempo changes, the system must repeatedly change its choice of filter, which implies the filters must be closely spaced to be able to smoothly track tempo variations. However, the system applies no continuity constraint when switching between filters.

Dixon [19] processes a sequence of note onset times either extracted from an audio signal or from a symbolic representation within a multi-agent system. Likely tempo hypotheses are derived from clustering inter-onset intervals (IOI), thus encoding aspects of the metrical hierarchy. The hypothesis are used to form multiple beat agents using a paradigm, where each agent has a state consisting of the period and the phase of the beat. The sequence of beat times with the best score to date is selected by the agent. The observations are only processed if they occur around the predicted beat locations, i.e., within a window whose width depends on the pulse period. The algorithm is designed to track beats in expressively performed music.



Cemgil et al. [11] formulate beat tracking in a Bayesian framework where tempo and beat is modeled as a stochastic dynamical system. The system is defined with two hidden state variables, namely the period and the phase of beat. To this deterministic model, they add a Gaussian random vector whose covariance matrix models the likely tempo variations. State transitions are defined by a simple set of equations that describe how state variables evolve with time. Because all noises are assumed to be Gaussian and all relationships between variables are linear, the covariance matrix can be efficiently estimated by a Kalman filter. They also define the tempogram representation which includes a probability distribution over the period and phase given a list of onset. This probability distribution is proportional to the likelihood of the observed onsets under given period and phase hypotheses, weighted by prior distribution, which is equally distributed, as they consider all tempi to be initially equiprobable. For given periods and phases, the likelihood is computed as the integral over all onsets of the product of a constant pulse track and a continuous representation of the onsets. This implements the assumption that a good pulse track is one which matches all onsets well. The tempogram marginal probability distribution  $p(\omega|t)$  provides a 1-D representation of periodicities resembling those aforementioned.

Laroche [20] initially finds salient features like note onsets, note changes, and percussion hits by calculating the Fourier transform of an audio signal. A nonlinear monotonic compression function is applied to the amplitude spectrum, so high-frequency components are not masked by higher amplitude low-frequency components. To locate fast variations in the frequency domain contents, a first-order difference is calculated. All bins are summed together, and the result is half-wave rectified to obtain a positive energy flux signal. A least-squares approach is used to determine the best candidates for the tempo and beat locations. The final step consists of going through the successive tempo analysis frames and finding in each frame the best candidates. To that effect a dynamic programming technique is used. This entails continuity and non-syncope constraints.

Klapuri et al. [21] expand upon Scheirer’s amplitude envelope and comb filter model. They adopt a more robust regstral accent signal across four parallel analysis bands as the input to their system and use comb filterbanks within a probabilistic framework to simultaneously track three metrical levels. These correspond to the tatum, tactus and measure, which are explained in section 3. Analysis can be performed causally and noncausally, and is not restricted to any particular genre, tempo or time-signature. The robustness of the analysis model is due to the probabilistic modelling of the temporal evolution and interaction between each of the three metrical levels analysed. In a study of audio tempo induction algorithms [29], this approach was shown to be most accurate.

Davies and Plumbley [22] adopt a simpler and more efficient, heuristic approach than the system of Klapuri by embedding context-dependent information directly into the beat period and alignment estimation processes. They use two state model; the first state performs tempo induction and tracks tempo changes, while the second maintains contextual continuity within a single tempo hypothesis. The first state, also called general state, operates in a memoryless fashion, extracting the beat period and beat

alignment through a process of repeated induction. In this manner, the two-state model can explicitly model tempo discontinuities while smoothing out odd beat errors during each consistent beat period hypothesis.

Peeters [23] approach is based on a probabilistic framework, in which the beat tracking problem is formulated in a hidden Markov model, that can be efficiently solved with the Viterbi algorithm [30]. An onset-energy-function, time-variable tempo, and meter serves as an input to the system. Beat times are decoded over beat-numbers according to observation and transition probabilities. A beat-template is used to derive the observation probabilities from the signal. For this purpose, a linear discriminant analysis finds the most discriminative beat-template.

### 3 Basic Principles of Human Rhythm Perception

- Beat, as a phenomenon, refers to the perceived pulses which are approximately equally spaced and define the rate at which notes in a piece of music are played. For a specific performance, the beat is defined by the occurrence times, beat times, of these pulses.
- Western music notation provides an objective regular temporal structure underlying musical event occurrences and organising them into a hierarchical metrical structure [8].
- Many works in musicpsychology attempt to group musical onsets together into a rhythmic context, i.e., to construct a model which subsues multiple onsets separated in time into a rhythmic clock, hierarchy, grouping, or oscillatory model.
- A rhythmic event is characterised by an onset time and a salience. They represent the most basic unit of rhythmic information, from which all beat and tempo information is derived.
- The more salient events are the more likely to correspond to beat times than the less salient events. This tendency for events with greater perceptual salience to occur in stronger metrical positions has been noted by various authors [31, 32, 33].
- Each pair of events in a rhythmic sequence initially contributes to the salience of a single pulse sensation (emphasis occurs), and later that pulse sensations can enhance the salience of other consonant pulse sensations [33]. One may understand the initially above as an indication not to implement influential schemes between metrical levels in the induction process, but indeed to do it in the tracing process, which is also in agreement with the Dynamic Attending Theory [34].
- Perception research has shown that with up to 40 ms difference in onset times, two tones are heard as synchronous, and for more than two tones, the threshold is up to 70 ms [35].
- Rhythmic information is provided by IOIs in the range of approximately 50 ms to 2 s [35].
- Perception of beat is a prerequisite to rhythm perception, which in turn is a fundamental part of music perception.
- Experiments on the tempo sensitivity on humans have shown that the ability to notice tempo changes is proportional to the tempo, with the JND (just noticeable difference) being around 2-5% [36].
- Although in the brains of performers music is temporally organized according to its hierarchical beat structure, this structure is not explicitly expressed in music; it is implied in the relations among various musical elements which are not fixed and which are dependent on musical genres or pieces.

- A listener who cannot identify chord names can nevertheless perceive chord changes.
- The experience of rhythm involves movement, regularity, grouping, and yet accentuation and differentiation [35].
- There is no ground truth for rhythm to be found in simple measurement of an acoustic signal. The only ground truth is what human listeners agree to be the rhythmic aspects of the musical content of that signal
- As contrasted with rhythm in general, beat and pulse correspond only to the sense of equally spaced temporal units [35]. Where meter and rhythm associate with qualities of grouping, hierarchy and a strong/weak dichotomy, pulses in a piece of music are only periodic at a simple level.
- The Generative Theory of Tonal Music (GTTM) [31] formalises the distinction objective structure and the performers conceptualisation of music by defining rules for a “musical grammar” which deals separately with grouping structure and metrical structure. While the grouping structure deals with time spans, the metrical structure deals with durationless points in time, the beats, which obey the following rules. Beats must be equally spaced. A division according to a specific duration corresponds to a metrical level. There must be a beat of the metrical structure for every note in a musical sequence. A beat at a high level must also be a beat at each lower level. At any metrical level, a beat which is also a beat at the next higher level is called a downbeat, and other beats are called upbeat. Beats obey a discrete time grid, with time intervals all being multiples of a common duration the smallest metrical level.
- Lerdahl and Jackendoff [31] describe meter perception as the progress of finding periodicities in the phenomenal and structural accents in a piece of music. They propose a set of metrical preference rules, based on musical intuitions, which are assumed to guide the listener to plausible interpretations of rhythms. The rules prefer structures where: beats coincide with note onsets; string beats coincide with onsets of long notes; parallel groups receive parallel metrical structure; and the strongest beat occurs early in the group.
- Lerdahl and Jackendoff [31] classify musical accents into three types: phenomenal accents, which come from physical attributes of the signal such as amplitude and frequency; structural accents, which arise from perceived points of arrival and departure such as cadences; and metrical accents, points in time which are perceived as accented due to their metrical position.
- A major weakness of the GTTM is that it does not deal with the departures from strict metrical timing which are apparent in almost all styles of music. Thus it is only really suitable for representing the timing structures of musical scores, or as an abstract representation of a performance, where the expressive timing is not represented.

- Drake and Bertrand [37] advocate a universal predisposition toward simple duration ratio, and claim that humans tend to hear a time interval as twice as long or short as previous intervals. The Dynamic Attending Theory [34] proposes that humans spontaneously focus on a reference level of periodicity, and they can later switch to other levels to track events occurring at different time spans, e.g., longer span harmony changes, or a particular shorter-span fast motive.
- They list the following types of phenomenal accent, which they consider incomplete: note onsets, sforzandi, sudden dynamic or timbral changes, long notes, melodic leaps and harmonic changes. However, they give no indication as to how these factors might be compared or combined, either quantitatively (absolute values) or qualitatively (relative strengths).
- Longer notes tend to be perceived as accented.
- The beat of a piece of music is the sequence of equally spaced phenomenal impulses which define a tempo for the music.
- The period of a beat is the time duration between two successive beats, i.e., the reciprocal of the tempo.
- The phase of a beat determines where a beat occurs with respect to performance time.
- The frequency of the pulse in a rhythmic musical signal is the tempo or the rate of the rhythm, and the phase of the pulse indicates where the downbeat of the rhythm occurs. That is, the times at which a pulse occurs can be defined to have zero phase, and thus the points in time exactly in-between pulses have phase of  $\pi$  radians.
- In order to represent changing tempi, various approaches can be used. If tempo is considered as an instantaneous value, it can be calculated as the inter-beat interval measured between each pair of successive beat. A more perceptual plausible approach is to take an average tempo measured over a longer period of time. A measure of central tendency of tempo over a complete musical excerpt is called the basic tempo [38], which is the implied tempo around which the expressive tempo varies.
- The value of tempo as a function of time is called a tempo curve.
- Tempogram [?]
- Given a metrical structure, tempo is defined as the rate of the beats at a given metrical level, for example the quarter note level in the score.
- There is usually a preferred or primary metrical level, which corresponds to the rate at which most people would tap or clap in time with the music, and this is

commonly used to define the tempo, expressed either as a number of beats per minute, or as the time interval between beats, the inter-beat interval (IBI).

- The perception of tempo exhibits a degree of variability. It is not always correct to assume that the denominator of the time signature corresponds to the “foot-tapping” rate, nor to the actual “physical tempo” that would be an inherent property of audio flows [39].
- Differences in human perception of tempo depend on age, musical training, musical preferences and general listening context, e.g., tempo of a previously heard sequence, listener’s activity, instant of the day [36, 40, 41]
- Differences in tempo perception are nevertheless far from random. They most often correspond to a focus on a different metrical level, e.g., differences of half or twice the inter-beat interval or one-third or three times the inter-beat interval.
- Human pitch recognition is only sensitive to signal phase under certain unusual conditions. rhythmic response is crucially a phased phenomenon - tapping on the beat is not at all the same as tapping against the beat. or slightly ahead of or behind the beat, even if the frequency of tapping is accurate [10]
- From psychoacoustic demonstration on beat perception it can be shown that certain kinds of signal manipulations and simplifications can be performed without affecting the perceived pulse content of a musical signal. An amplitude-modulated noise constructed by vocoding a white noise signal with the subband envelopes of a musical signal is sufficient to extract pulse and meter. The simplified signal is created by performing a frequency analysis of the original signal by processing it through a filterbank of bandpass filters, or grouping FFT bins together. Thus it seems that separating the signal into subbands and maintaining the subband envelopes separately is necessary to do accurate rhythmic processing. This fact leads to the hypotheses that some sort of cross-band rhythmic integration, not simply summation across frequency bands, is performed by the auditory system to perceive rhythm [10].
- Studies of Povel and Essnes [32] have demonstrated that beat perception may be explained with a model in which a perceptual clock is aligned with the accent structure of the input. The model relies heavily on structural qualities of the input, such as a sophisticated model of temporal accent, to function.
- They propose a model of perception of temporal patterns, based on the idea that a listener tries to induce an internal clock which matches the distribution of accents in the stimulus and allows the pattern to be expressed in the simplest possible terms. They use patterns of identical tone bursts at precise multiples of 200 ms apart to test their theory.
- Expressive timing is generated from performers’ understanding of the musical structure and general knowledge of music theory and musical style [42]. However, there

is no precise mathematical model of expressive timing, and the complexity of musical structure from which timing is derived, coupled with the individuality of each performer and performance, makes it impossible to capture musical nuance in the form of rules [19].

- Expressive reductions in tempo are more common and more extreme than tempo increases [38].

## 4 Foundations of Machine Learning

### 4.1 Sequence Modeling

Given an input sequence  $x_{1:T} = x_1, \dots, x_T$  with sequence length  $T$ , a sequence model is any function  $f : \mathcal{X}^T \rightarrow \mathcal{Y}^T$  such that

$$y_{1:T} = y_1, \dots, y_T = f(x_1, \dots, x_T) \quad (1)$$

where  $y_t$  should only depend on  $x_{1:t}$  and not on  $x_{t+1:T}$ , i.e., no leakage of information from the future. This causality constraint is essential for autoregressive modeling. We use  $x_{1:T} = x_1, \dots, x_T$  to denote a input sequence of length  $T$ , where vector  $x_t \in \mathbb{R}^p$  is the input at time step  $t$ . Thus  $x_{1:T} \in \mathbb{R}^{T \times p}$ .

The goal of learning in the sequence modeling setting is to find a network  $f$  that minimizes some expected loss between the actual outputs and the predictions,

$$L(y_1, \dots, y_T, f(x_1, \dots, x_T)) \stackrel{!}{=} \min \quad (2)$$

where the sequences and outputs are drawn according some distribution.

For most deep learning practitioners, sequence modeling is synonymous with recurrent networks. For example the sequence modeling chapter in a standard reference on deep learning is titled “Sequence Modeling: Recurrent and Recursive Nets” [2] capturing the common association of sequence modeling and recurrent architectures. Recent results indicate that convolutional architectures can outperform recurrent networks. Concluding from an empirical evaluation of generic convolutional and recurrent networks for sequence modeling, Bai et al. [3] assert that the common association between sequence modeling and recurrent networks should be reconsidered, and convolutional networks should be regarded as a natural starting point for sequence modeling tasks.

- Sequence modeling include polyphonic music modeling[4], word- and character-level language modeling.
- settings such as auto-regressive prediction, where we try to predict some signal given its past, by setting the target output to be simply the input shifted by one time step

### 4.2 Data Representation

- Many artificial intelligence tasks can be solved by designing the right set of features to extraxt for that task, then providing theses features to a simple machine learning algorithm. [2]

### 4.3 Architectures

- Given a sequence modeling task or dataset, which architecture should one use?
- Bai et al. [3] show that a simple convolutional architectures outperforms canonical recurrent networks such as LSTMs across a diverse range of tasks and datasets, while demonstrating longer effective memory. Is it also the case in beat tracking?



### 4.3.1 Recurrent Neural Networks

In deep learning, a recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence.

- Recurrent networks are dedicated sequence models that maintain a vector of hidden activations that are propagated through time [43, 44, 45].
- This family of architectures has gained tremendous popularity due to prominent applications to language modeling and machine translation.
- The intuitive appeal of recurrent modeling is that the hidden state can act as a representation of everything that has been seen so far in the sequence.
- Basic RNN architectures are notoriously difficult to train and more elaborate architectures are commonly used instead, such as the LSTM [46] and the GRU [47].

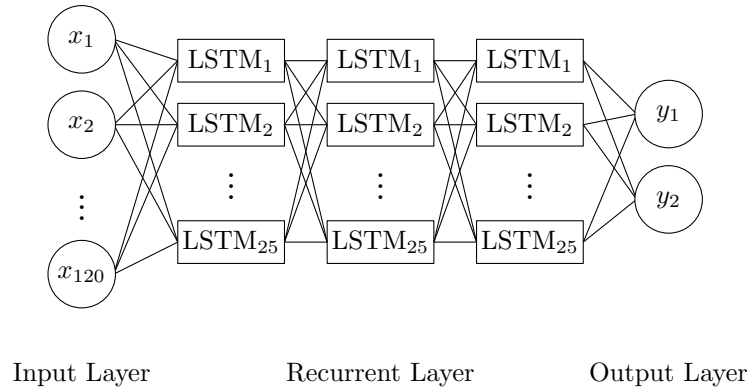


Figure 2: Model architecture in [24].

### 4.3.2 Convolutional Neural Networks

A convolutional neural network (CNN) is a class of deep neural networks, most commonly applied to analyzing visual imagery.

- Convolutional neural networks (CNNs) [48] have been applied to sequences for decades. They were used prominently for speech recognition in the 80s and 90s.

### 4.3.3 Temporal Convolutional Networks

A temporal convolutional network (TCN) represents a special kind of convolutional neural network and is informed by recent convolutional architectures for sequential data [3]. It is designed from first principles and combines simplicity, autoregressive prediction,

and very long memory. In comparison to WaveNet [4], the TCN does not employ skip connections across layers (no conditioning, context stacking, or gated activations).

The TCN is based upon two principles: 1) the convolutions are casual, i.e., no information leakage from future to past; 2) the architecture can take a sequence of any length and map it to an output sequence of the same length just as with an RNN. To achieve the first point, the TCN uses a 1D fully-convolutional network architecture [49], where each hidden layer is the same length as the input layer. To accomplish the second point, the TCN uses causal convolutions, i.e., convolutions where an output at time  $t$  is convolved only with elements from time  $t$  and earlier in the previous layer.

Simple causal convolutions have the disadvantage to only look back at history with size linear in the depth of the network. To circumvent this fact, the architecture employs dilated convolutions that enable an exponentially large receptive field. More formally, for a 1-D sequence input  $\mathbf{x} \in \mathbb{R}^T$  and a filter  $f : \{0, \dots, k-1\} \rightarrow \mathbb{R}$ , the delated convolution operation  $F$  on element  $s$  of the sequence is defined as

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i) \mathbf{x}_{s-d \cdot i} \quad (3)$$

where  $d = 2^\nu$  is the dilation factor, with  $\nu$  the level of the network, and  $k$  is the filter size. The term  $s - d \cdot i$  accounts for the direction of the past. Dilation is equivalent to introducing a fixed step between every two adjacent filter taps, as it can be seen in Fig. 3. Using larger dilation enables an output at the top level to represent a wider range of inputs, thus effectively expanding the receptive field of a CNN. There are two ways to increase the receptive field of a TCN: choosing lager filter sizes  $k$  and increasing the dilation factor  $d$ , since the effective history of one layer is  $(k-1)d$ .

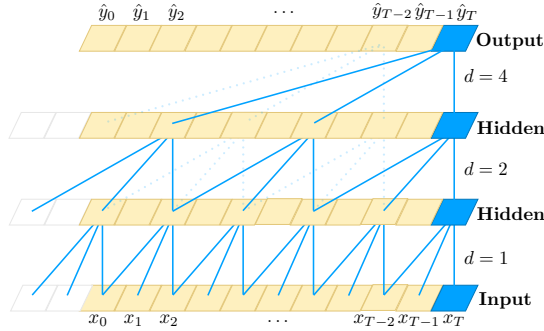


Figure 3: A dilated casual convolution with dilation factors  $d = 1, 2, 4$  and filter size  $k = 3$  [3].

Another architectural element of a TCN are residual connections. In place of a convolutional layer, TCNs employ a generic residual module. Each residual block contains a branch leading out to a series of transformations  $\mathcal{F}$ , whose outputs are added to the input  $\mathbf{x}$  of the block,

$$o = \text{Activation}(\mathbf{x} + \mathcal{F}(\mathbf{x})). \quad (4)$$

This effectively allows layers to learn modifications to the identity mapping rather than the entire transformation, which has been shown to benefit deep neural networks [50]. Especially for very deep networks stabilization becomes important, for example, in the case where the prediction depends on a large history size ( $> 2^{12}$ ) with a high-dimensional input sequence.

A residual block has two layers of dilated causal convolutions and rectified linear units (ReLU) as non-linearity, shown in Fig. 4. For normalization, weight normalization [51] is applied to the convolutional filters. In addition, a spatial dropout [52] is added after each dilated convolution for regularization, i.e., at each training step, a whole channel is zeroed out.

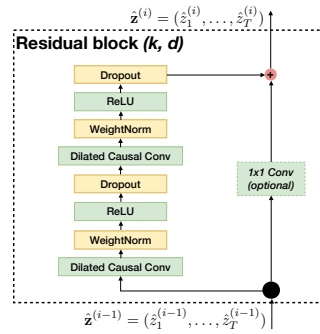


Figure 4: TCN residual block [3].

- the convolutions are casual, that means no information leakage from future to past (compare to bidirectional RNN).
- long effective history sizes, i.e., the ability for networks to look very far into the past to make a prediction (receptive field?), by using a combination of very deep networks (augmented with residual layers) and dilated convolutions
- this allows layers to learn modifications to the identity mapping rather than the entire transformation, which has been shown to benefit deep neural networks (where?)
- within a residual block there exist two layers of dilated causal convolution and a rectified linear unit (ReLU) [53, Nair2010] as a non-linearity
- for normalization a weight normalization [51, Salimans2016] is applied to the convolutional filters
- additionally spatioal dropout [52, Srivastava2014] is added after each dilated convolution for regularization (at each training step, a whole channel is zeroed out)
- The TCN architecture appers not inly more accurate than canonical recurrent networks such as LSTMs and GRUs, but also simpler and clearer.

#### 4.3.4 Trellis Networks

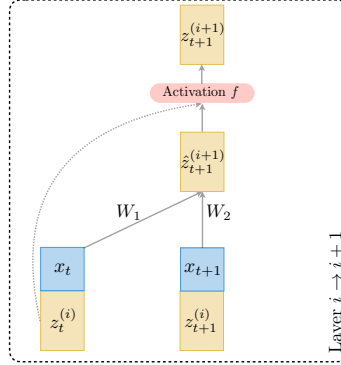


Figure 5: TrellisNet at an atomic level.

**Input:**  $x_{1:T} = x_1, \dots, x_T$ ,  $x_t \in \mathbb{R}^p$ ,  $x_{1:T} \in \mathbb{R}^{T \times p}$   
with sequence length  $T$  input dimensionality  $p$

**Output:**  $y_{1:T} = y_1, \dots, y_T = G(x_1, \dots, x_T)$   
with function  $G : \mathcal{X}^T \rightarrow \mathcal{Y}^T$

**Hidden:**  $z_t^{(i)} \in \mathbb{R}^q$

**LogSoftmax:** A LogSoftmax normalization is added the last layer of the neural network to obtain the log-probabilities for the different classes.

$$\text{LogSoftmax}(x_i) = \log \left( \frac{\exp(x_i)}{\sum_j \exp(x_j)} \right) \quad (5)$$

#### 4.4 Performance Measure

**Loss Function** Cross entropy is defined for two probability distributions  $p$  and  $q$  as

$$H(p, q) = - \sum_{x \in \mathcal{X}} p(x) \log q(x). \quad (6)$$

In machine learning cross entropy can be used as a loss function to measure the performance of a classification model. The probability  $p_i$  is the true label (binary indicator 0 or 1), where as the distribution  $q_i$  is the predicted value of the current model.

## 4.5 Regularization Techniques

## 4.6 Optimization

- Regularization with Dropout or Weight Decay
- check out cyclical learning rates [54][Smith2017]

Estimate generalization error with a validation set during training. Stop training when the error on the validation set rises (early stopping).

## 4.7 Hyperparameter Optimization

- optimize hyperparameters with nevergrad
- instrumentation: turn a piece of code with parameters into a function defined on an n-dimensional continuous data space

## 4.8 Multi Task Learning

- It is the information in these extra training signals that helps the hidden layer learn a better internal representation for the door recognition domain, and this better representation in turn helps the net better learn to recognize [Caruana1997]
- Information in the extra tasks is helping the hidden layer learn a better internal representation
- 

## 4.9 Validation

**Test set method** Split the observations into two disjoint subsets

$$\text{observations} \left\{ \begin{array}{l} \text{training data } \left\{ \left( \mathbf{x}^{(\alpha)}, \mathbf{y}_T^{(\alpha)} \right) \right\}, \alpha \in \{1, \dots, p\} \\ \rightarrow E^T \text{ selects model parameters} \\ \text{test data } \left\{ \left( \mathbf{x}^{(\beta)}, \mathbf{y}_T^{(\beta)} \right) \right\}, \beta \in \{1, \dots, q\} \\ \rightarrow \hat{E}^G \text{ estimates generalization error} \end{array} \right.$$

**n-fold cross-validation**

## 5 Beat Tracking System

From an overall perspective, the proposed automatic beat tracking system comprises three major phrases: data preprocessing, feature learning and temporal decoding.

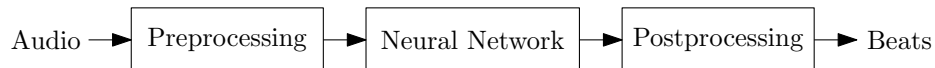


Figure 6: Beat tracking system signal flow

Sequence-to-sequence prediction:

- the entire input sequence can be used to predict each output (not feasible in real time processing)

Binary classification problem:

- beat (class 1)
- no beat (class 0)

### 5.1 Dataset

For training and evaluation we use the datasets listed in Table 1.

The Hainsworth dataset [55] contains 222 musical audio files, with the following genre breakdown: Rock/Pop (68), Dance (40), Jazz (40); Classical (30), Folk (22), and Choral (22). Each file is between 30 and 60 s in length, mono and sampled at 44.1 kHz with 16-bit resolution.

Table 1: Datasets used for training.

| <b>Dataset</b>    | <b>files</b> | <b>length</b> |
|-------------------|--------------|---------------|
| Ballroom [29, 56] | 685          | 5 h 57 m      |
| GTZAN [57, 58]    | 1000         | 8 h 20 m      |
| Hainsworth [55]   | 222          | 3 h 19 m      |
| SMC [59]          | 217          | 2 h 25 m      |
| Total             |              | h m           |

- The ideal outcome of the annotation is an unambiguous representation of the start points of musical events
- Due to uncertainties in the annotation process, for many types of input signals (especially multi-instrument excerpts) it may not be possible to determine onset locations with greater precision than 50 ms. [60]
- The input data format is uncompressed linear pulse code modulated (PCM) signals. Single channel 16 bit linear PCM with a sampling rate of 44100 Hz. The data is created by averaging the two channels of the original stereo recordings.

## 5.2 Data Preprocessing

- “Is it important to use several features?” [Durand2015]
- Two different types of input features: Onset events and harmonic changes
- Assumption: Most harmonic changes occur inside a piece of music are located at metric bars [5, Khadkevich2012]
- in complex polyphonic mixtures of music, simultaneously occurring events of high intensities lead to masking effects that prevent any observation of an energy increase of a low intensity onset. To circumvent these masking effects, detection functions were proposed that analyze the signal in a bandwise fashion to extract transients occurring in certain frequency regions of the signal [Grosche2011]

The Beat Tracking task requires annotations in the form of time instants of beats from a musical excerpt. Additionally, the tempo, metrical level of the beat and downbeat positions can also be annotated for related tasks like downbeat tracking/ meter tracking/ tempo tracking etc.

The method for obtaining ground truth annotations depends on whether the aim is to identify descriptive beat locations or to replicate a human tapping response. In the former case, an initial estimate of the beat locations can be obtained by recording tap times for a given musical excerpt and iteratively modifying and auditing the obtained beat positions while in the latter case, the ground truth can be completely defined by the tapping response [61].

The audio excerpt in time domain and its spectrogram can be visualised using tools like Sonic Visualizer. Beat locations can first be obtained by recording the tap locations the musical excerpt and then manually correcting these locations for exact onsets. Annotators can also follow a semi-automatic process, for example in [2], where beats and downbeats were first obtained through the ircambeat software and then errors were manually corrected by human annotators.

After preprocessing the audio we obtain the observation set  $O = \left\{ \mathbf{x}^{(\alpha)}, \mathbf{y}_T^{(\alpha)} \right\}_{\alpha=1}^p$ , with  $p$  samples in total.

- Data augmentation (transformation or adding noise)  $\Rightarrow$  prediction gets robust against transformation and noisy signals
- output:  $y$  probability of beat instant per time frame

The first step in the approach consists of preprocessing the original data. Data preprocessing refers to all transformations on the raw data before the resulting training set is fed to the machine learning algorithm. It includes different methods such as normalization, transformation and feature extraction.

The data set contains raw pulse code modulated (PCM) audio signals stored as WAV files. For the sake of consistency and also to reduce computational complexity the audio signal is resampled at a sampling rate  $f_s = 44.1$  kHz and converted to a monaural signal by averaging both stereo channels.

- Slice audio  $x(t)$  into frames  $x_n(t)$ ,  $n = 1, 2, \dots, N$ , where  $N$  is the number of frames
- Compute complex spectrogram  $X(n, k)$  with FFT
- Complex spectrogram is converted to the power spectrogram  $S(n, k) = |X(n, k)|^2$
- Mel-spectrogram  $M(n, m) = \log (S(n, k) \cdot F(m, k)^T + 1.0)$

Short-time Fourier transform:

$$X(n, k) = \sum_{m=0}^{N-1} w(m) x(m + nh) e^{-2\pi j k / N} \quad (7)$$

Spectrogram:

$$S(n, k) = |X(n, k)|^2 \quad (8)$$

Separation of magnitude and phase:

$$X(n, k) = |X(n, k)| e^{j\phi(n, k)} \quad (9)$$

### 5.2.1 Chroma Representation

- Chords are more likely to change in beat times than on other positions. Chords are more likely to change at the beginnings of measures than at other positions of the beat.
- A chromagram comprises a time-series of chroma vectors. which represent harmonic content at a specific time in the audio.
- Chromagrams are concise descriptors of harmony because they encode tone quality and neglect tone height.
- CRP (chroma DCT-reduced log-pitch) features have significant amount of robustness to changes in timbre an instrumentation [62, Mueller2010]
- Employ data-driven approach to extract chromagrams that specifically encode content relevant to harmony
- Chroma features are noisy in their basic formulation because they are affected by various interferences: musical instruments produce overtones in addition to the fundamental frequency; percussive instruments pollute the spectrogram with broadband frequency activations (e.g. snare drums) and/or pitch-like sounds (tom-toms, bass drums); different combinations of instruments (and different, possibly genre-dependent mixing techniques) create different timbres and thus increase variance [7,20] [63]



### 5.3 Feature Learning

### 5.4 Temporal Decoding

We use a probabilistic dynamic model to exploit the sequential structure of music, generally resulting in a more robust estimation.

As a postprocessing step we make use of a dynamic Bayesian network (DBN) which jointly infers tempo and phase of the beat.

Hidden variables:

- $\omega$  - the tempo
- $\phi$  - the position inside the beat period

In order to infer the hidden variables from an audio signal, we specify three entities:

- **transition model**: describes the transitions between the hidden variables
- **observation model**: takes the beat activations from the neural network
- **initial distribution**: encodes prior knowledge about the hidden variables
  - test probability model for given tempo estimation

#### 5.4.1 Transition Model

- The number of observations  $M$  per beat at tempo  $T$  in BPM is defined as

$$M(T) = \frac{60}{T} f_r \left\lceil \frac{\text{frames}}{\text{beat}} \right\rceil \quad (10)$$

- The beat position state is dependent on the tempo by using exactly one state per audio frame.
- $\Phi \in \{1, 2, \dots, M(T)\}$  denotes the position inside the beat period (pib)
- The tempo space corresponds to integer valued beat positions in the interval  $[M(T_{\max}), M(T_{\min})]$ , with

$$N_{\max} = M(T_{\min}) - M(T_{\max}) + 1 \quad (11)$$

different tempo states.

- The tempo in beat positions per time frame  $\dot{\Phi} \in \{M(T_{\max}), M(T_{\max})+1, \dots, M(T_{\min})\}$

- For example:

$$\begin{aligned} T_{\min} = 56 \text{ BPM} &\rightarrow M(T_{\min}) = 107 \\ T_{\max} = 215 \text{ BPM} &\rightarrow M(T_{\max}) = 28 \end{aligned}$$

$$\begin{aligned} \Phi &\in \{1, 2, \dots, 107\} \\ \dot{\Phi} &\in \{28, 29, \dots, 107\} \end{aligned}$$

- Transitions to each tempo is allowed but only at beat times

$$\omega_k = \begin{cases} \omega_{k-1}, & P(\omega_k|\omega_{k-1}) = 1 - p_\omega \\ \omega_{k-1} + 1, & P(\omega_k|\omega_{k-1}) = \frac{p_\omega}{2} \\ \omega_{k-1} - 1, & P(\omega_k|\omega_{k-1}) = \frac{p_\omega}{2} \end{cases} \quad (12)$$

- The probability of tempo change is heuristically set to  $p_\omega = 0.002$
- Higher-level or domain specific knowledge could be used to set this parameter. For example in rock or pop music, the beat is usually quite steady, so a small value for  $p_\omega$  would be quite appropriate, while for classical music, particularly styles including many tempo changes, a higher value would be more optimal.

#### 5.4.2 Observation model

## 6 Evaluation

- annotations tapped at different metrical levels in beat tracking
- a evaluation method should adequately contend with the inherent uncertainty and/or ambiguity while providing a measurement of performance which is both meaningful and easy to interpret. [61]
- objective methods for beat tracking evaluation compare the output beat times from a beat tracking algorithm against one or more sequences of ground truth annotated beat times. [61]
- We must question the importance of strict continuity when evaluating a beat tracking system. A single misplaced beat, in the context of a sequence of otherwise accurate beats, is a far less disturbing error than the accuracy calculated with the continuity requirement would suggest.

### 6.1 Performance measures

Set of annotated beat times  $\mathcal{A} = \{a_1, \dots, a_A\}$

Set of predicted beat times  $\mathcal{B} = \{b_1, \dots, b_B\}$

True positive if beat is in  $\pm 70$  ms range of annotated beat

$tp$  = number of true positives

$fp$  = number of false positives (extra detections)

$fn$  = number of false negatives (missed detections)

**Precision and recall:**

$$\text{precision} = \frac{tp}{tp + fp}, \quad \text{recall} = \frac{tp}{tp + fn}$$

**F-measure:**

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2tp}{2tp + fp + fn}$$

**P-score:** Define two sequences  $T_a$  and  $T_b$  as

$$T_a(n) = \begin{cases} 1, & \text{if } n \in \mathcal{A} \\ 0, & \text{otherwise} \end{cases}, \quad T_b(n) = \begin{cases} 1, & \text{if } n \in \mathcal{B} \\ 0, & \text{otherwise} \end{cases}$$

$$\text{P-score} = \frac{\sum_{m=-w}^w \sum_n T_a(n) T_b(n+m)}{\max(A, B)}, \quad \text{where } w = 0.2 \text{ median}(\Delta_a)$$

## 6.2 Saliency maps

- Artificial neural networks often give good results, but it is difficult to understand what they learned, or on which basis they generate their output.
- compute saliency maps using guided back-propagation [25]

## 6.3 Network training

- To achieve better results, we could use DNN ensembles instead of a single DNN. We could ensure that the network sees data for which its predictions are wrong more often during training, or similarly, we could simulate a more balanced dataset by showing the net super-frames of rare chords more often [63]

## 6.4 Labeling

- In an editor (Sonic Visualizer) that enables a user to mark beat positions in a digitalized audio signal while listening to the audio and watching its waveform.
- The positions can be finely adjusted by playing back the audio with click tones at beat times.

## 7 Conclusion

- We have presented an efficient audio based beat tracking algorithm able to produce equivalent results to the current state-of-the-art.
- Limitations in both the amount of training data and the variable quality of the annotations. Perform data augmentation.
- Dataset size: “[...] a rough rule of thumb is that a supervised deep learning algorithm will generally achieve acceptable performance with around 5000 labeled examples per category and will match or exceed human performance when trained with a dataset containing at least 10 million labeled examples.” [2, Goodfellow2016]

## **8 Appendix**

### **8.1 Hyperparameter**

- window function (e.g. Hamming window)

## References

- [1] Nick Collins. Towards a style-specific basis for computational beat tracking. 2006.
- [2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [3] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.
- [4] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [5] Maksim Khadkevich, Thomas Fillon, Gaël Richard, and Maurizio Omologo. A probabilistic approach to simultaneous extraction of beats and downbeats. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 445–448. IEEE, 2012.
- [6] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *ISMIR*, pages 255–261, 2016.
- [7] Daniel P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, mar 2007.
- [8] Fabien Gouyon and Simon Dixon. A review of automatic rhythm description systems. *Computer music journal*, 29(1):34–54, 2005.
- [9] Masataka Goto and Yoichi Muraoka. Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions. *Speech Communication*, 27(3-4):311–335, 1999.
- [10] Eric D. Scheirer. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601, 1998.
- [11] A. T. Cemgil, H. J. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram representation and kalman filtering. *Journal of New Music Research*, 28:4:259–273, 2001.
- [12] Andrew Robertson and Mark D. Plumbley. B-keeper: A beat-tracker for live performance. In *NIME*, 2007.
- [13] Juan Pablo Bello and Jeremy Pickens. A robust mid-level representation for harmonic content in music signals. In *ISMIR*, volume 5, pages 304–311. Citeseer, 2005.
- [14] Mark A Bartsch and Gregory H Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on multimedia*, 7(1):96–104, 2005.

- [15] Juan Bello. Towards the automated analysis of simple polyphonic music: A knowledge-based approach (ph. d. thesis). 2003.
- [16] W. Andrew Schloss. *On the Automatic Transcription of Percussive Music - From Acoustic Signal to High-Level Analysis*. PhD thesis, Stanford University, 05 1985.
- [17] Paul E Allen and Roger B Dannenberg. Tracking musical beats in real time. In *ICMC*, 1990.
- [18] Masataka Goto and Yoichi Muraoka. A beat tracking system for acoustic signals of music. In *Proceedings of the second ACM international conference on Multimedia*, pages 365–372. ACM, 1994.
- [19] Simon E. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58, 08 2001.
- [20] Jean Laroche. Efficient tempo and beat tracking in audio recordings. *Journal of the Audio Engineering Society*, 51(4):226–233, 2003.
- [21] Anssi P Klapuri, Antti J Eronen, and Jaakko T Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, 2005.
- [22] Matthew E. P. Davies and Mark D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 15(3):1009–1020, March 2007.
- [23] Geoffroy Peeters. Beat-tracking using a probabilistic framework and linear discriminant analysis. *12th International Conference on Digital Audio Effects (DAFx-09)*, 2009.
- [24] Sebastian Böck and Markus Schedl. Enhanced beat tracking with context-aware neural networks. In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, 2011.
- [25] Sebastian Böck, Florian Krebs, and Gerhard Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. In *ISMIR*, 2014.
- [26] Florian Krebs, Sebastian Böck, and Gerhard Widmer. An efficient state-space model for joint tempo and meter tracking. In *ISMIR*, pages 72–78, 2015.
- [27] Matthwe E. P. Davies and Sebastian Böck. Temporal convolutional networks for musical audio beat tracking. *European Association for Signal Processing*, 2019.
- [28] Masataka Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001.

- [29] Fabien Gouyon, Anssi Klapuri, Simon Dixon, Miguel Alonso, George Tzanetakis, Christian Uhle, and Pedro Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.
- [30] Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, 13(2):260–269, 1967.
- [31] Fred Lerdahl and Ray S Jackendoff. *A generative theory of tonal music*. MIT press, 1985.
- [32] Dirk-Jan Povel and Peter Essens. Perception of temporal patterns. *Music Perception: An Interdisciplinary Journal*, 2(4):411–440, 1985.
- [33] Richard Parncutt. A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception: An Interdisciplinary Journal*, 11(4):409–464, 1994.
- [34] Mari R Jones and Marilyn Boltz. Dynamic attending and responses to time. *Psychological review*, 96(3):459, 1989.
- [35] Stephen Handel. Listening. *An introduction to the perception of auditory events*, Cambridge, MA, 1989.
- [36] Carolyn Drake and Marie-Claire Botte. Tempo sensitivity in auditory sequences: Evidence for a multiple-look model. *Perception & Psychophysics*, 54(3):277–286, 1993.
- [37] Carolyn Drake and Daisy Bertrand. The quest for universals in temporal processing in music. *PsychoL Sci*, 13:71–4, 2001.
- [38] Bruno H Repp. On determining the basic tempo of an expressive music performance. *Psychology of Music*, 22(2):157–167, 1994.
- [39] Carolyn Drake, L Gros, and Amandine Penel. How fast is that music? the relation between physical and perceived tempo. *Music, mind, and science*, pages 190–203, 1999.
- [40] Carolyn Drake, Mari Riess Jones, and Clarisse Baruch. The development of rhythmic attending in auditory sequences: attunement, referent period, focal attending. *Cognition*, 77(3):251–288, 2000.
- [41] Carolyn Drake, Amandine Penel, and Emmanuel Bigand. Why musicians tap slower than nonmusicians. *Rhythm perception and production*, pages 245–248, 2000.
- [42] Eric F Clarke. Rhythm and timing in music. In *The psychology of music*, pages 473–500. Elsevier, 1999.



- [43] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.
- [44] Paul J Werbos et al. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [45] Alex Graves. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*, pages 5–13. Springer, 2012.
- [46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [47] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [48] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- [49] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [51] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.
- [52] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [53] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [54] Leslie N Smith. Cyclical learning rates for training neural networks. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 464–472. IEEE, 2017.
- [55] Stephen W Hainsworth and Malcolm D Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Advances in Signal Processing*, 2004(15):927847, 2004.

- [56] Florian Krebs, Sebastian Böck, and Gerhard Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *ISMIR*, pages 227–232, 2013.
- [57] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.
- [58] Ugo Marchand and Geoffroy Peeters. Swing ratio estimation. pages 423–428, 2015.
- [59] A. Holzapfel, M. Davies, J. R. Zapata, J. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 20(9), 2012.
- [60] Pierre Leveau and Laurent Daudet. Methodology and tools for the evaluation of automatic onset detection algorithms in music. In *In Proc. Int. Symp. on Music Information Retrieval*. Citeseer, 2004.
- [61] Norberto Degara Matthew E. P. Davies and Mark D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical report, Centre for Digital Music, Queen Mary University of London, 2009.
- [62] Meinard Muller and Sebastian Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):649–662, 2010.
- [63] Filip Korzeniowski and Gerhard Widmer. Feature learning for chord recognition: The deep chroma extractor. *arXiv preprint arXiv:1612.05065*, 2016.