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 featuress 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?
- Dataset size: "*[...] a rough rule of thumb is that a supervised deep learning algorithm will gernerally 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]
- 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 Psychoaccoustics

- 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 noticable difference) beeing around 2-5% [7].

2 Beat Tracking Difficulty of a Song

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

- absence of a clear rhythmic structure (Classical music)
- soft onsets

- heavy syncopation
- expressive timing (e.g. rubato playing)
- the problem of 'octave errors' (detecting double or half time the rate of the ground truth)

Quantify the difficulty of beat tracking with indicators such as

- beat strength [Tzanetakis2002]
- pulse clarity [Lartillot2008]
- entropy of a cyclic tempogram [Grosche2010] as an indicator of the tempo salience

3 Signal processing

3.1 Time-Frequency Reassignment

- audio signals have a distribution of energy that varies in time and frequency
- a spectrogram is constrained by an unfortunate tradeoff between resolution in time and frequency
- time-frequency reassignment: mapping the data to time-frequency coordinates that are nearer to the true region of support of the analyzed signal
- time-frequency reassignment has been used in a variety of applications for obtaining improved time and frequency estimates for time-varying spectral data [cite... 16-18]

4 Feature extraction

Potential features: Zero Crossing Rate, Energy, Danceability, Entropy of Energy, Spectral Centroid, Spectral Spread, Spectral Entropy, Spectral Flux, Spectral Roll off, MFCC, Chroma Vector, Chroma Deviation

5 Clustering

The whole training data gets separated in different clusters. Each cluster represents a homogenous subset of the whole data set, i.e., the multiple models are trained for specific aspects (e.g. genre, timbre, rhythmical patterns, instrumentation, etc.).

Figure 1: Beat tracking system signal flow

6 Beat Tracking System

7 Preprocessing

- "*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, Goodfellow2016]
- Two different types of input features: Onset events and harmonic changes
- Assumtion: Most harmonic changes occur inside a piece of music are located at metric bars [5, Khadkevich2012]

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 $f_s = 44.1\,\text{kHz}$ and converted to a monaural signal by averaging both stereo channels.

- Slice audio $x(t)$ into frames $x_n(t)$, $n = 1, 2, \ldots, N$, 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 $M(n, m) = \log\left(S(n, k) \cdot F(m, k)^T + 1.0\right)$

Short-time Fourier transform:

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

Spectrogram:

$$S(n, k) = |X(n, k)|^2 \tag{2}$$

Separation of magnitude and phase:

$$X(n, k) = |X(n, k)|\, e^{j\phi(n,k)} \tag{3}$$

7.1 Chroma representation

- CRP (chroma DCT-reduced log-pitch) features have significant amount of robustness to changes in timbre an instrumentation [8, Mueller2010]

8 Machine learning model

Seuqence-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)

8.1 Sequence modeling

We are given an input sequence $x_1, \ldots, x_T$ and wish to predict some corresponding outputs $y_1, \ldots, y_T$ at each time. The key constraint is that to predict the output y_t for some time t, we are constrained to only use those inputs that have been previously observed: $x_1, \ldots, x_t$. Formally, a seqeunce modeling network is any function $f : \mathcal{X}^T \to \mathcal{Y}^T$ that produces the mapping

$$\hat{y}_1, \ldots, \hat{y}_T = f(x_1, \ldots, x_T) \tag{4}$$

The architecture should take a sequence of any length and map it an output seqence of any length and map it to an output sequence of the same length.

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, $L(y_0, \ldots, y_T, f(x_0, \ldots, x_T))$, where the sequences and outputs are drawn according to some distribution [Bai2018].

8.2 Data representation

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 [9].

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.

General: 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. As elements the set cointains:

- feature vector: $\mathbf{x} \in \mathbb{R}^{(\text{sequence length, input size})}$
- true label: $\mathbf{y}_T \in \{0, 1, \ldots, (\text{number of classes} - 1)\}^{(\text{sequence length})}$
- Data augmentation (transformation or adding noise) $\Rightarrow$ prediction gets robust against transformation and noisy signals
- output: y probability of beat instant per time frame
- $p = 698$
- $\mathbf{x} \in \mathbb{R}^{(3015,\, 120)}$
- $\mathbf{y}_T \in \{0, 1\}^{(3015)}$

8.3 Recurrent Networks (RNNs)

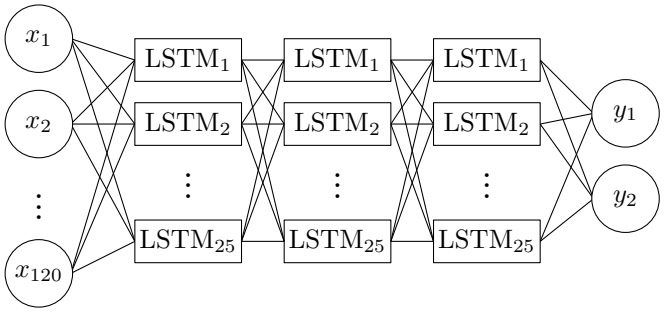

Figure 2: Model architecture in [10][Boeck2011]

8.4 Temporal Convolutional Netwoks (TCNs)

- the convolutions are casual, i.e, 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
- TCNs do not use gated mechanisms
- 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
- employ dilated convolutions that enable an exponentially large receptive field
- for a 1-D sequence input $\mathbf{x} \in \mathbb{R}^T$ and a filter $f : \{0, \dots, k-1\} \to \mathbb{R}$, the delated convolution operation F on element s of the sequence is defined as

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i)\, \mathbf{x}_{s-d\,i} \tag{5}$$

  where $d = 2^\nu$ is the delation factor, ν is the level of the network and k is the filter size.

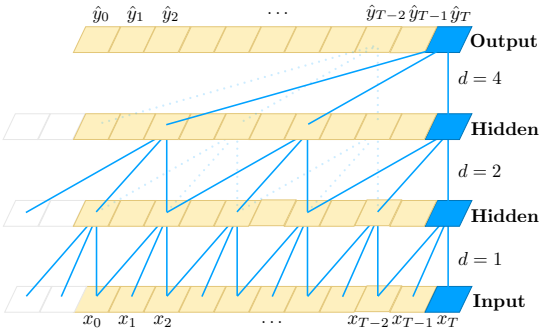

Figure 3: Dilated casual convolution. Source: [3][Bai2018]

- In place of a concolutional layer TCNs employ a gneric residual module.
- Each residual block contains a branch leading out to a series of transformations (dilated causal convolution, weight normalization, rectified linear transfer function, spatial dropout), whose outputs are added to the input $\mathbf{x}$ of the block
- 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) [11, Nair2010] as a non-linearity
- for normalization a weight normalization [12, Salimans2016] is applied to the convolutional filters
- additionaly spatioal dropout [13, Srivastava2014] is added after each dilated convolution for regularization (at each training step, a whole channel is zeroed out)

8.5 Trellis networks (TrellisNets)

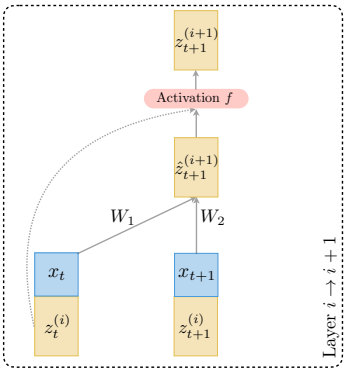

Figure 4: TrellisNet at an atomic level.

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

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

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 differnent classes.

$$\text{LogSoftmax}(x_i) = \log\left(\frac{\exp(x_i)}{\sum_j \exp(x_j)}\right) \tag{6}$$

8.6 Performance measure

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

$$H(p, q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x). \tag{7}$$

In machine learning cross entropy can be used as a loss funciton to measures 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.

8.7 Regularization techniques

8.8 Optimization

- Regularization with Dropout or Weight Decay
- check out cyclical learning rates [14][Smith2017]

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

8.9 Valididation

Test set method Split the observarions into to disjunct subsets

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

n-fold cross-validation

9 Postprocessing

We use a probabilistic dynamic model to exploit the seqential structure og music.

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

Hidden variables:

- ω - the tempo
- ϕ - 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

9.1 Transition model

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

$$M(T) = \frac{60}{T} f_r \quad \left[\frac{\text{frames}}{\text{beat}}\right] \tag{8}$$

- The beat position state is dependent on the tempo by using exactly one state per audio frame.
- $\Phi \in \{1, 2, \ldots, M(T)\}$ 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

$$N_{\max} = M(T_{\min}) - M(T_{\max}) + 1 \tag{9}$$

  different tempo states.
- The tempo in beat positions per time frame $\dot{\Phi} \in \{M(T_{\max}), M(T_{\max}) + 1, \ldots, M(T_{\min})\}$
- For example:

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

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

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

$$\omega_k = \begin{cases} \omega_{k-1}, & P(\omega_k|\omega_{k-1}) = 1 - p_\omega \\ \omega_{k-1} + 1, & P(\omega_k|\omega_{k-1}) = \frac{p_\omega}{2} \\ \omega_{k-1} - 1, & P(\omega_k|\omega_{k-1}) = \frac{p_\omega}{2} \end{cases} \tag{10}$$

- The probability of tempo change is heuristically set to $p_\omega = 0.002$

9.2 Observation model

10 Evaluation

10.1 Performance measures

Set of annotated beat times $\mathcal{A} = \{a_1, \ldots, a_A\}$
Set of predicted beat times $\mathcal{B} = \{b_1, \ldots, b_B\}$

True positive if beat is in ± 70 ms rage 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:

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

F-measure:

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

P-score: Define two sequences T_a and T_b as

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

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

11 Appendix

11.1 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] Carolyn Drake and Marie-Claire Botte. Tempo sensitivity in auditory sequences: Evidence for a multiple-look model. *Perception & Psychophysics*, 54(3):277–286, 1993.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.