Musical Audio Beat Tracking

With temporal convolutional networks

– Ekko Wu, Sebastian Xu

About beat tracking

What is beat tracking:

- Beat tracking is a process that identifies the beat positions in a music recording. It's a key task in Music Information Retrieval (MIR).

Why important:

- Build basis for all beat related tasks
  - Provide structural insight
  - Basis for Downbeat detection
  - Drum generation based on drumless music

Application:

- Chord recognition
- Drum beat generation

Reproduce Paper:

Title: Temporal convolutional networks for musical audio beat tracking

Author: Matthew E. P. Davies, Sebastian Bock

Publication: 2019 27th European Signal Processing Conference (EUSIPCO)

What interesting about this paper:

- Using Temporal Convolutional Networks instead of RNN

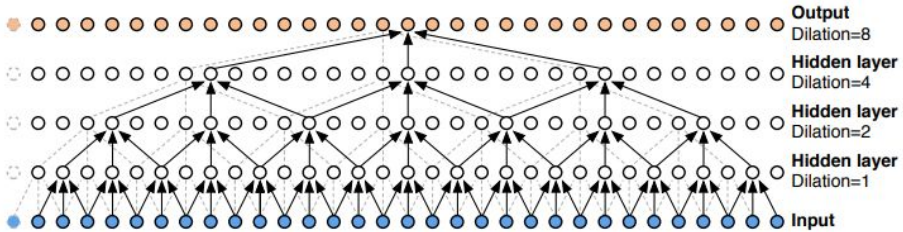


Fig. 2. Overview of the TCN structure (adapted from the original version [13]) to demonstrate non-causal operation. The grey dashed lines show the network connections shifted back one time step.

State of Art Method - BLSTM

Spectrogram:

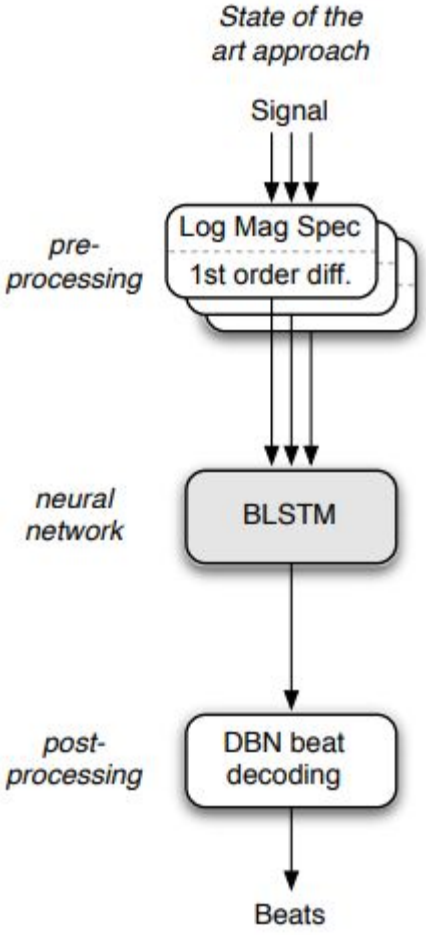
- hop size: 10 ms
- Three window sizes of 23.2 ms, 46.4 ms and 92.9 ms

BLSTM:

- Three layers of BLSTM

Beat Location:

- Peak picking (OR in this case)
- Dynamic Bayesian Networks (DBNs) via HMM



Paper Proposed Method part1

Signal Conditioning	
Audio sample rate	44.1 kHz
Window shape	*Hann*
Window & FFT size	2048 *samples*
Hop size	10 ms
Filterbank freq. range	30 . . . 17000 Hz
Sub-bands per octave	12
Total number of bands	81

Spectrogram:

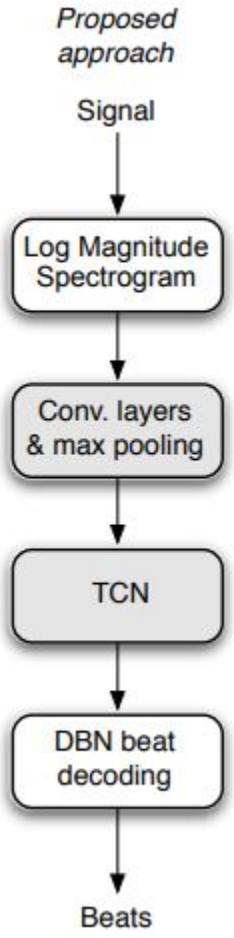
- single log magnitude spectrogram (mel spectrogram)
  - a hop size of 10 ms and a window size of 46.4 ms (2048 samples)
  - A logarithmic grouping of frequency bins with 12 bands per octave input representation: total of 81 frequency bands from 30 Hz up to 17 kHz

Conv. Block	
Number of filters	16, 16, 16
Filter size	3×3, 3×3, 1×8
Max. pooling size	1×3, 1×3, —
Dropout rate	0.1
Activation function	*ELU*

Convolutional Block:

- three convolutional layers
  - 16 filters of kernel size 3 × 3 -> max pooling over 3 bins(frequency direction)
  - 16 filters of kernel size 3 × 3 -> max pooling over 3 bins(frequency direction)
  - 16 filters of kernel size 1 × 8 -> no pooling
  - Dropout
  - Activation: ELU (Exponential Linear Unit)
  - Output 16-dimensional feature vector

$$\mathrm{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases}$$



Proposed Method part2

Temporal Convolution Network:

- One stack
- Dilation to 2^10
- 16 filters
- Filter size 5
- Dropout
- ELU

TCN	
Number of stacks	1
Dilations	$2^{0,\ldots,10}$
Number of filters	16
Filter size	5
Spatial dropout rate	0.1
Activation function	*ELU*

Output activation function:

- Sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Convolution + TCN + outputActivation = BeatNet:

- Example input is (3000,81) -> output (3000,) (activation function of 3000 samples long)

DBN decoding

- Take activation function -> generate beat prediction

Proposed Method part3

Optimizer and Criterion:

- Optimizer Adam
- Learning rate 0.001
- Batch size 1
- Output activation function sigmoid
- Loss function binary cross-entropy

When to stop:

- The validation loss does not improve for 50 epochs.

Evaluation Method

F-measure:

- The F-measure combines both precision and recall into a single metric by taking their harmonic mean.

Formula: $$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Simplified Formula: $$F1 = \frac{2TP}{2TP + FP + FN}$$

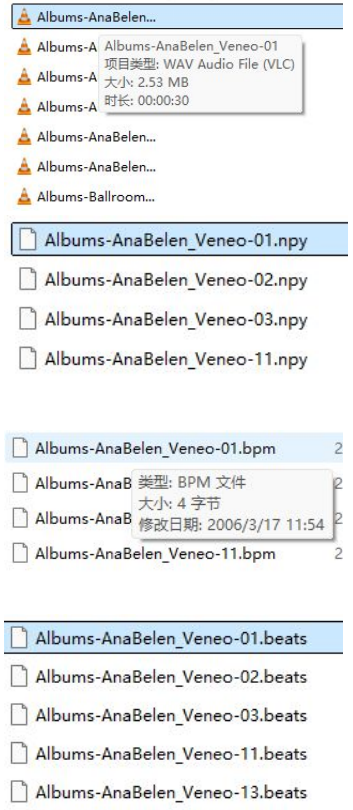
Implementation: data preparation

Ballroom dataset: UPF

- Training data from UPF is in .wav format. Wave file is converted to .npy spectrogram by pre-processing.
- UPF provides .bpm file only not .beats.

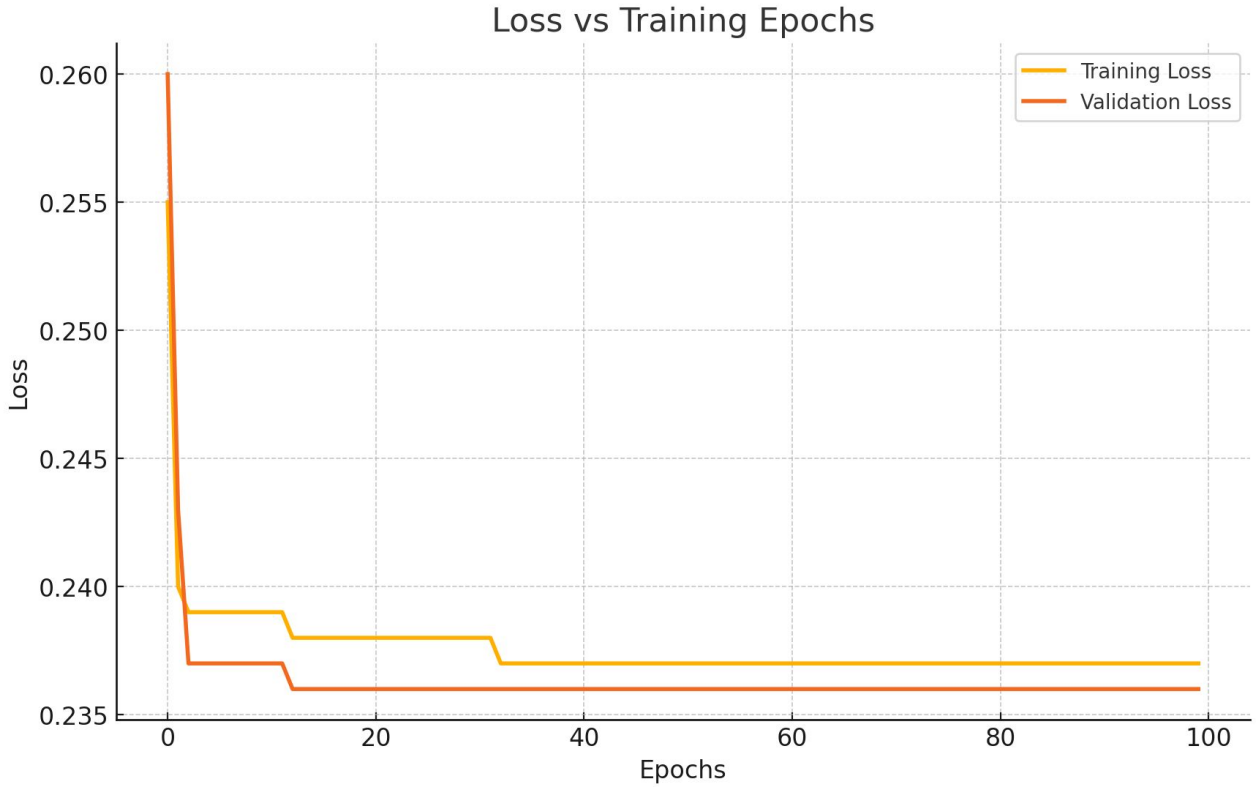
Table 1: Style distribution of the ballroom dance music excerpts

Cha Cha	111
Jive	60
Quickstep	82
Rumba	98
Samba	86
Tango	86
Viennese Waltz	65
Slow Waltz	110



Training:

The model is trained for 99 epochs and stops.



Model Evaluation: not successfully implemented

Evaluate_model.py:

- This function is provided by the paper but it requires to use madmom.

Madmom: (state of art model for beat and tempo tracking)

- Madmom was not successfully installed in my python3.12 environment.

Why evaluation implementation not successful:

1. The madmom model I found is asking for python3.8 but 3.8 is out of date and does not have installer available.
2. Madmom installing file requires lower version of numpy=1.18.x or 1.19.x, while these version is not compatible with python3.12.x. For example, Numpy uses np.int and np.float but they no longer exist in after numpy=1.23
3. A lot files of this project is written under python3.8, but in order to reproduce this paper, python 3.12 is use and a lot of code are modified to make all file in the project compatible with each other.

Conclusion:

The reproduction process in only half successful.

Thank you for your listening.