Deep Learning for MIR

Part III

Sebastian Böck
OFAI, TU Wien

Outline

- MIR primer: onset detection, beat and downbeat tracking
- recurrent neural networks (RNNs)
- some extensions to them
- how they can be used for MIR tasks

Introduction

Music is generally organised in a hierarchical way.
Metrical structure is defined at multiple levels:

- tatum or onsets level (e.g. 16th or 32nd notes)
- beats are at the level we usually nod our heads
- downbeats are the first beats of a measure

Beats and downbeats are integer multiples of tatum.

Metrical structure

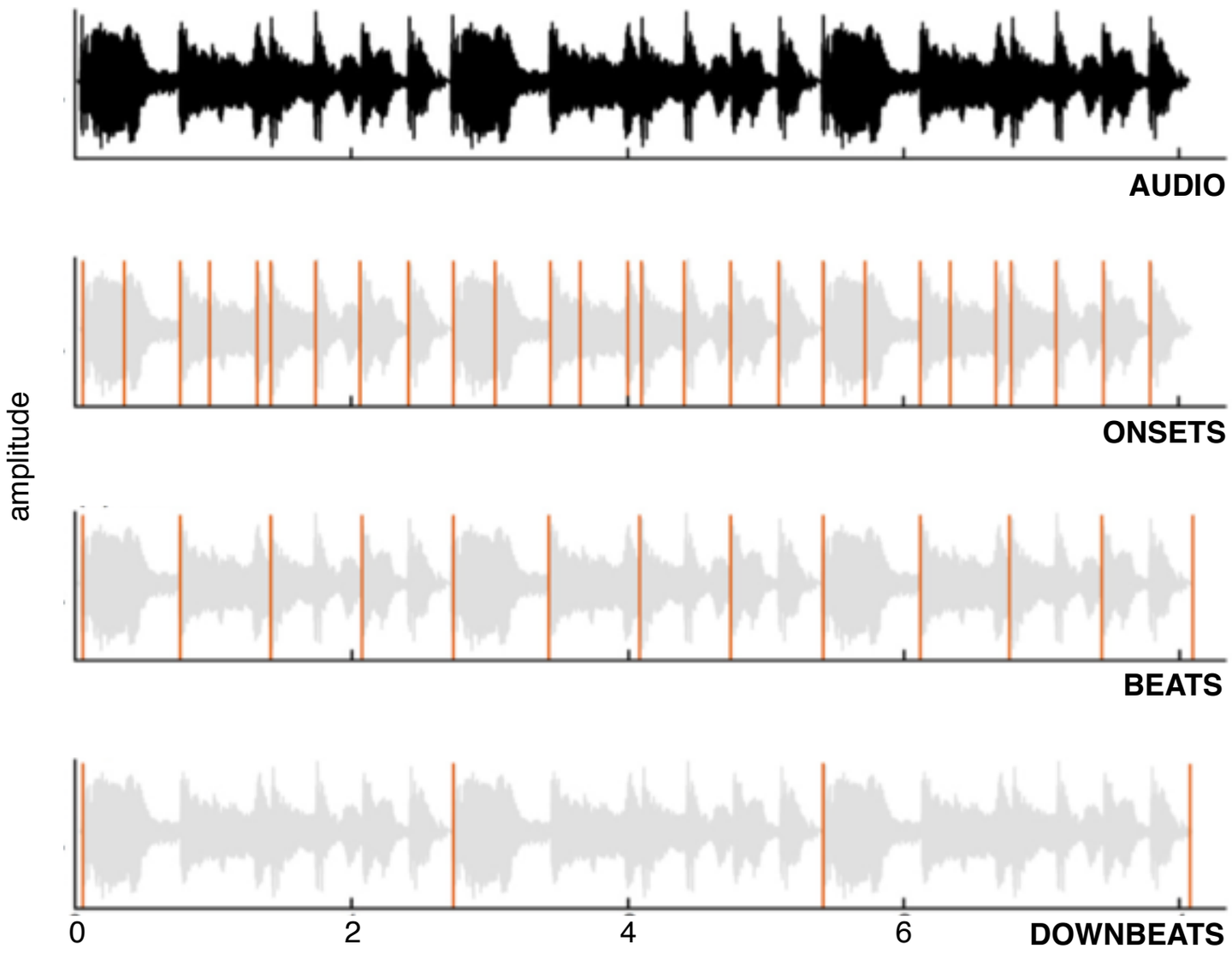


figure by Jason Hockman

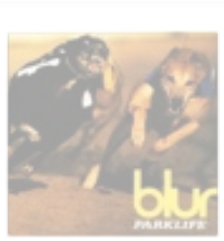
Motivation

Why do we want automatic rhythm analysis?
What can these MIR systems be used for?

- automatic music analysis and transcription
- music transformation
- automated accompaniment
- audio software, e.g. DJ-tools
- ...

Applications

Automatic rhythmic transformation
(example by Jason Hockman)



source **audio**
SONG A

target **rhythm**
SONG B

transform audio
RESULT

Applications

Automatic rhythmic transformation
(example by Jason Hockman)

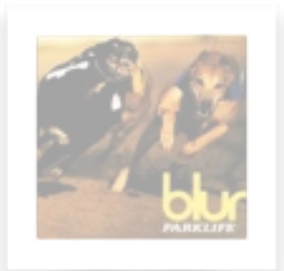
source **audio**
SONG A

target **rhythm**
SONG B

transform audio
RESULT

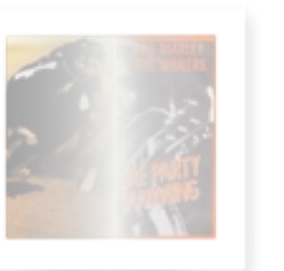
Applications

Automatic rhythmic transformation
(example by Jason Hockman)



source **audio**
SONG A

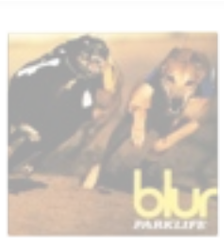
target **rhythm**
SONG B



transform audio
RESULT

Applications

Automatic rhythmic transformation
(example by Jason Hockman)



source **audio**
SONG A

target **rhythm**
SONG B

transform audio
RESULT

Applications

Automatic accompaniment

Onset detection

Task: detect all events in music (e.g. notes played by instruments, percussive sounds, singing voice, etc.)

Onsets usually coincide with rise in energy, thus this feature is exploited by most algorithms.

Early systems operated directly on time domain signals. Newer systems usually operate on time-frequency representations.

Onset detection is often used as a first step for higher level tasks such as beat tracking or tempo estimation.

Demo 1

http://localhost:8888/notebooks/Part_3a_Onset_Detection.ipynb

Beat and downbeat tracking

Beat and downbeat tracker often operate “layer-wise”:

- first detect the onsets
- decide which onsets are beats
- decide which beats are downbeats

Beat tracking

Common approach:

- compute novelty function
  (e.g. continuous onset detection function)
- detect periodicity (i.e. the tempo)
- detect phase (i.e. beat positions)

Hundreds of methods proposed in literature

Beat tracking

Periodicity analysis:

- **histogram**: inter onset intervals (IOIs) aggregated in bins representing beat periods, e.g. Dixon 2007

- **autocorrelation**: sliding dot product of a signal with a time-shifted version of itself, e.g. Ellis 2007

- **comb filters**: measure resonance of signal when passing it through a bank of comb filters with different time lags, Klapuri et al. 2006

Beat tracking

Phase selection:

After selecting the dominant period (tempo) beat positions must be found

- **cross-correlation** of detection function with pulse train of determined tempo, e.g. Davies & Plumbley 2007

- **dynamic programming**: maximise likely beat positions in detection function wrt. estimated tempo, e.g. Ellis 2007

Downbeat tracking

Detect the “one” in a sequence of beats and the length of one cycle (a bar / measure).

- downbeats often coincide with harmonic changes (e.g. western music, Pop, Rock, EDM)

- bars are often defined by the boundaries of rhythmic patterns (e.g. non-western music, HJDB)

Downbeat tracking

Most existing systems exploit either one or both of these features to track the downbeats, e.g.:

- Krebs at al. 2013 jointly models bar position, tempo and rhythmic patterns with a dynamic Bayesian network (DBN)

- Papadopoulos & Peeters 2011 jointly estimate chords and downbeats by integrating knowledge of mutual dependencies between chords and metric structure

Deep Learning for MIR

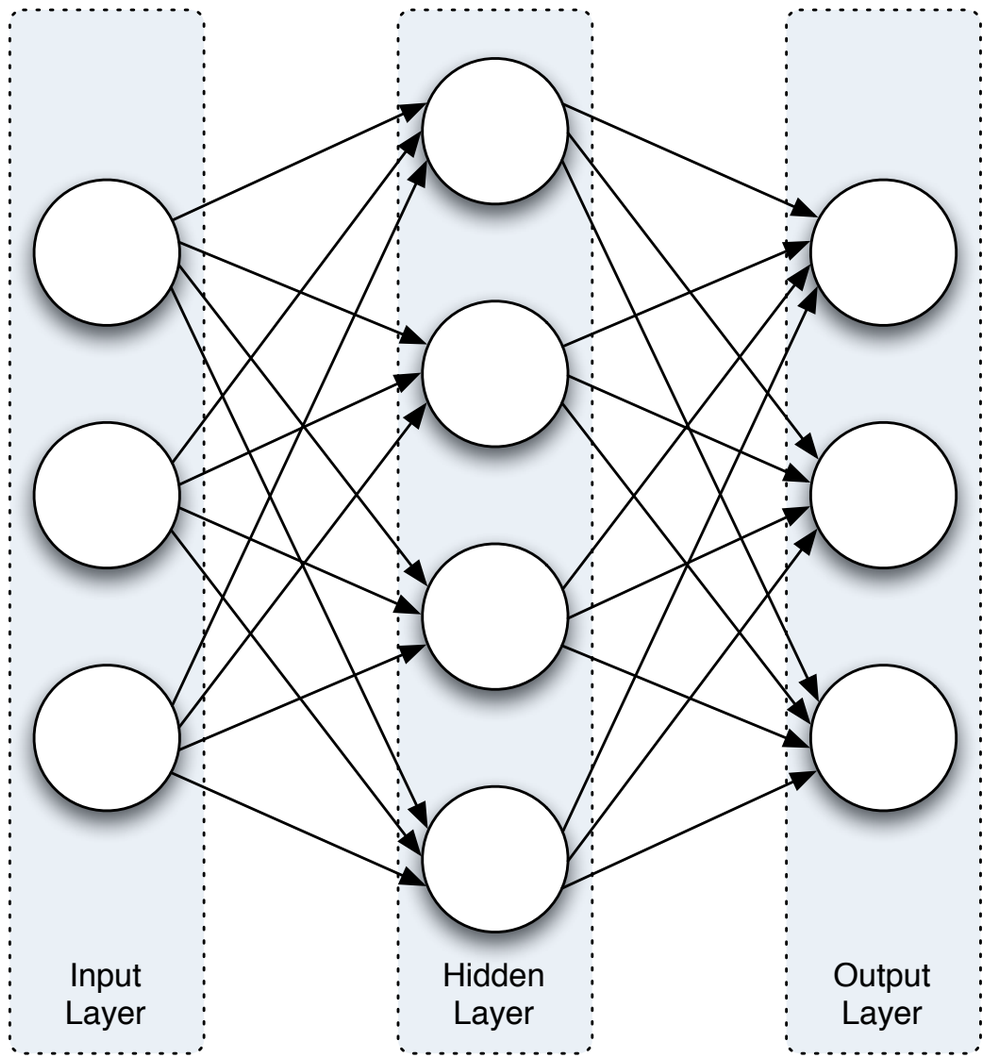
- deep learning is prevalent in MIR nowadays
- often it is deployed only for certain sub-tasks (e.g. Durand et al. 2016 use individual networks for melody, harmony, and rhythm to infer downbeats)
- but systems can also be learned end-to-end

Deep Learning for MIR

- learn features instead of hand-crafting them
- replace manually defined rules with statistics
- many different technologies can be used
- RNNs are just one of them

Feed Forward Neural Networks

- input Layer, hidden layer(s) output layer
- output computed on current input
- no temporal context

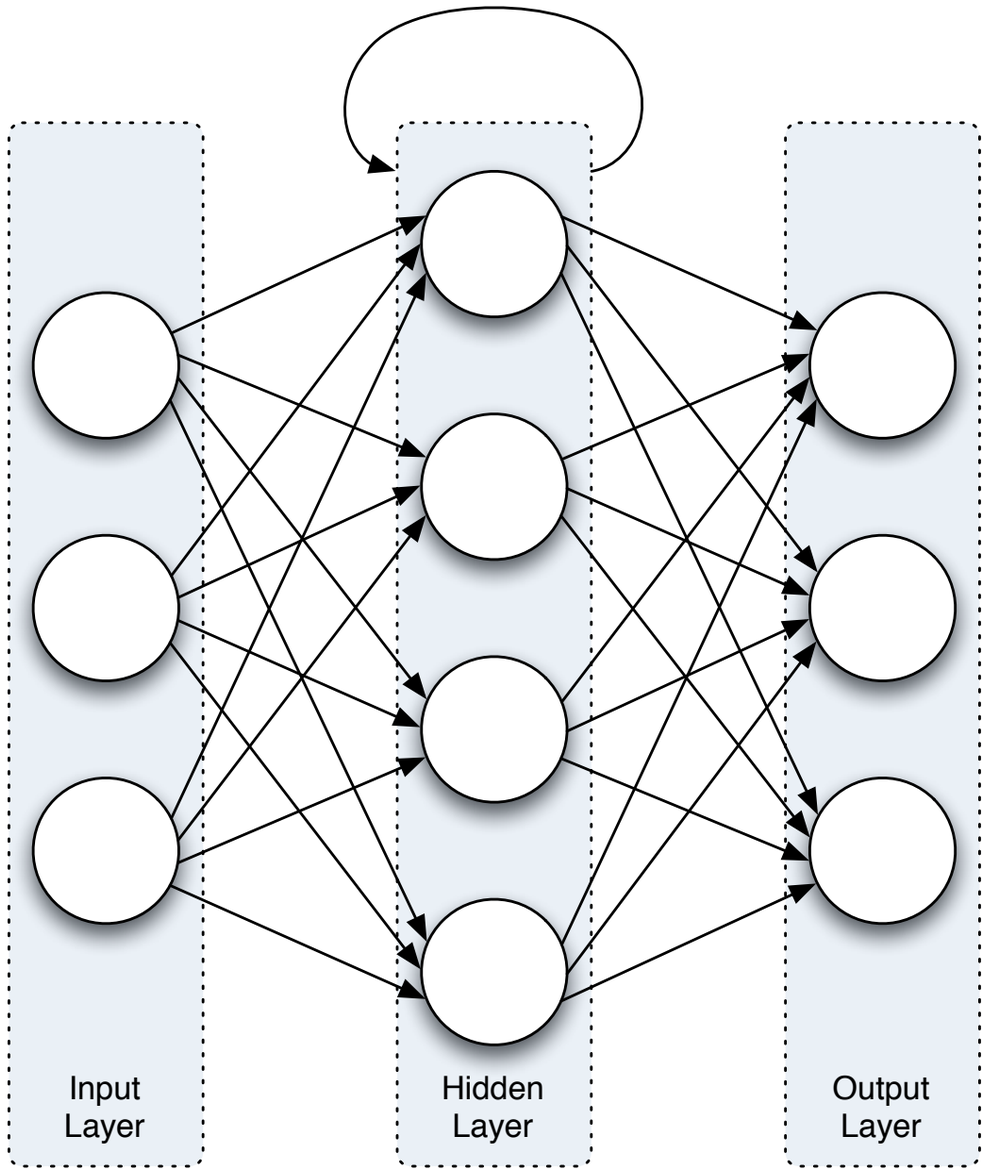


Recurrent Neural Networks

- add recurrent connections in the hidden layer

- output computed on current input and previous inputs

- can model temporal context



Recurrent Neural Networks

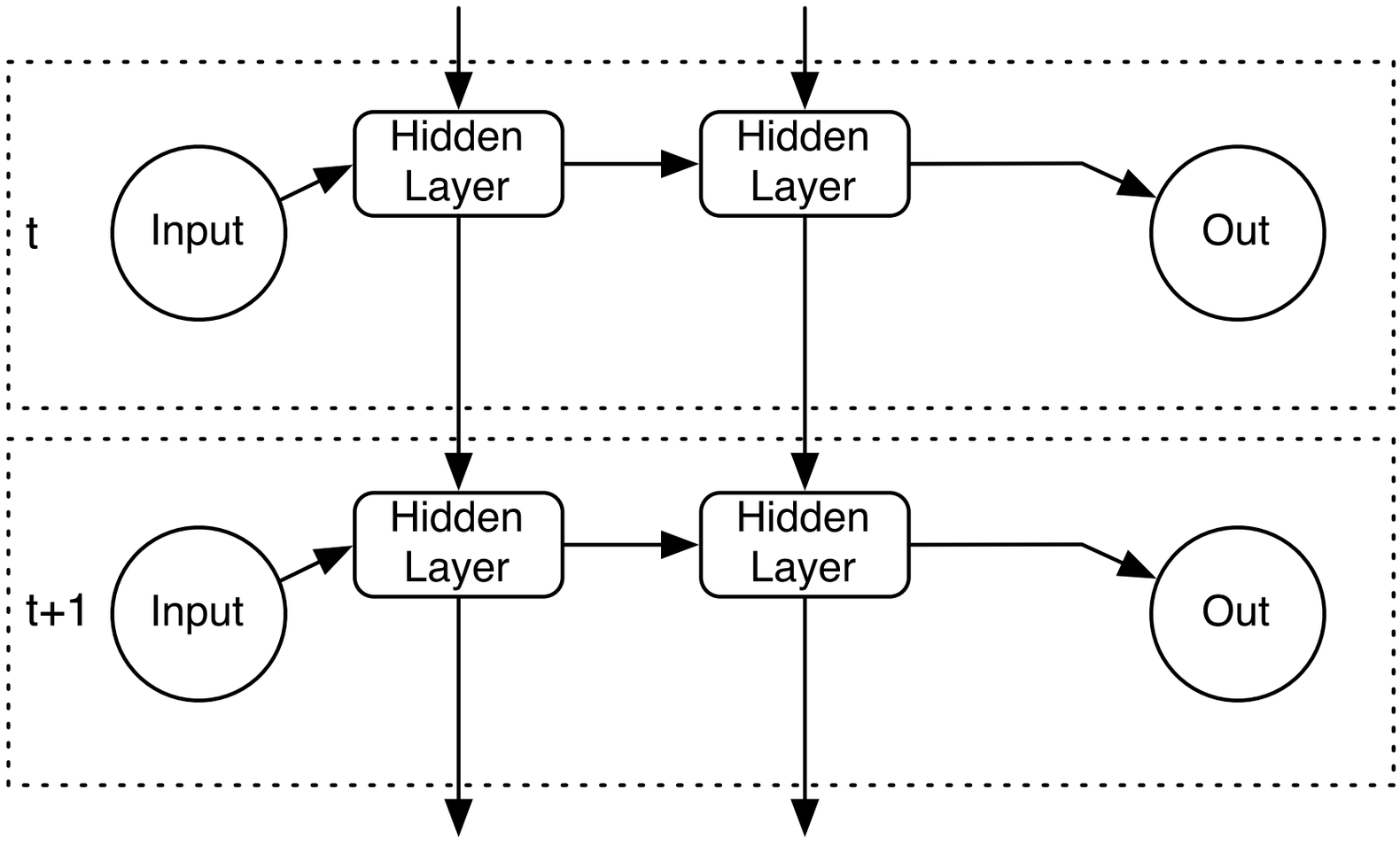
Advantages:

- good for modelling time series (audio signals!)

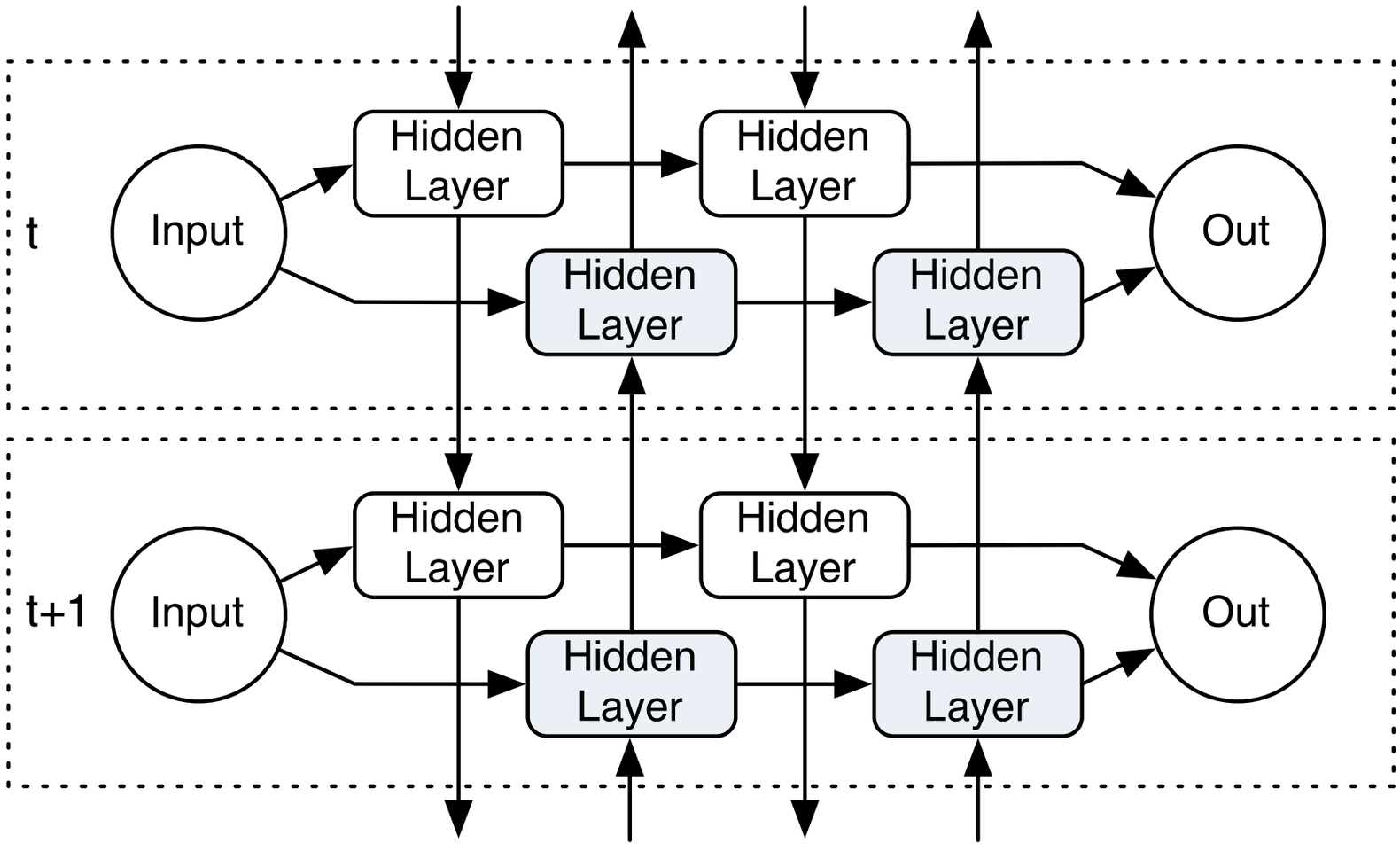
Disadvantages:

- vanishing gradient problem
- hard to train for large number of units/layers
- limited context (~10 time steps, ~100ms@100fps)
- can only access past information (unidirectional RNN)

Unidirectional RNN



Bidirectional RNN



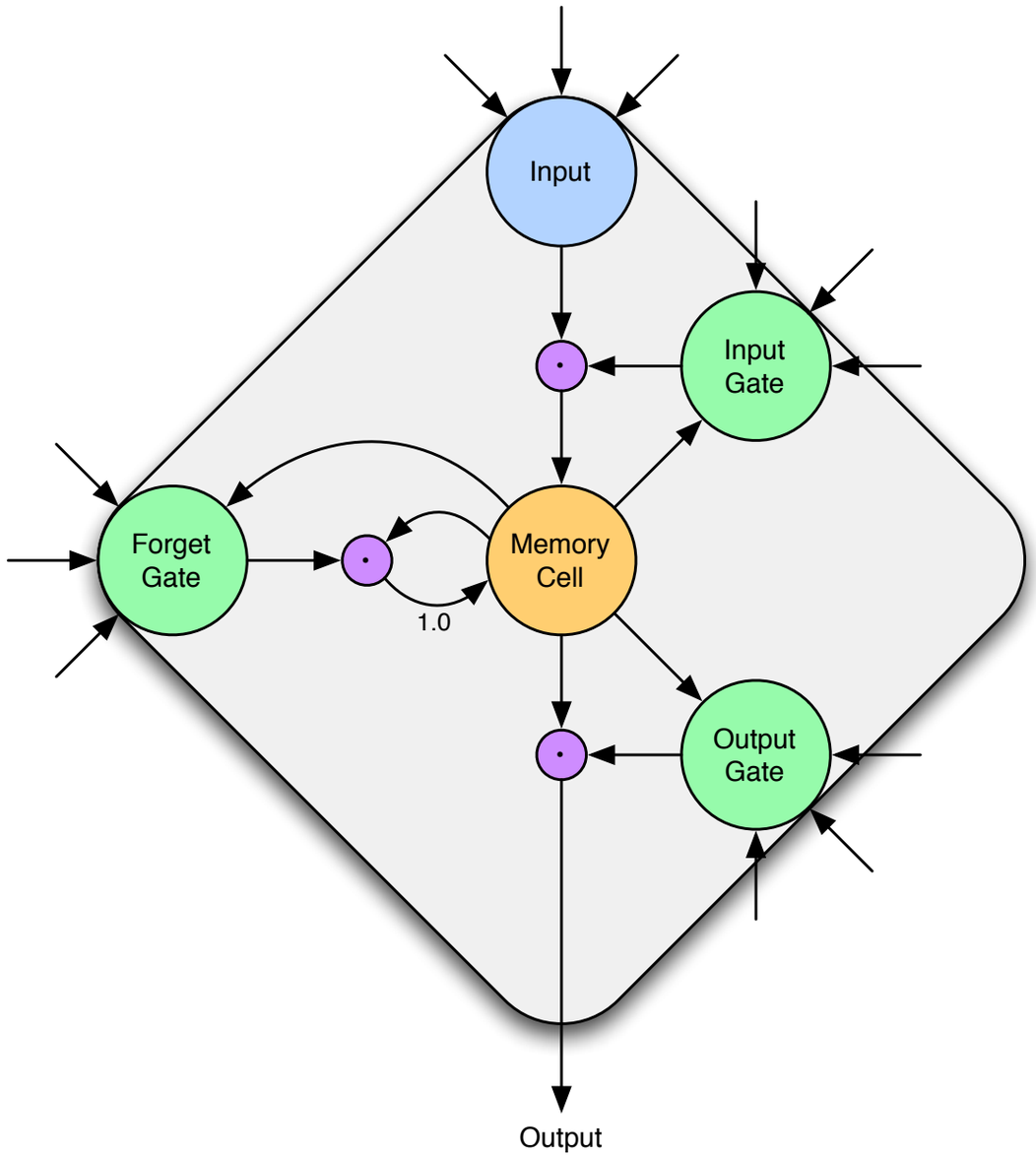
Bidirectional RNN

- proposed by Schuster and Paliwal in 1997
- add a second set of hidden layers
- feed these hidden layers with the input sequence in reverse temporal order
- network has access to past and “future” information (advantageous for modelling musical events)
- still limited context (enough for onset detection)

LSTM

- Long Short-Term Memory
- proposed by Hochreiter and Schmidhuber in 1997
- overcome the vanishing gradient problem
- have a “memory cell” which stores information
- is able to model long term context

LSTM


Input
Input Gate
Forget Gate
Memory Cell
1.0
Output Gate
Output

RNNs for onset detection

- spectrogram input
- spectrogram diffs as auxiliary input
- three hidden bidirectional recurrent layers
- single sigmoid output unit
- one hot encoding of targets
- stochastic gradient descent with momentum
- minimising binary cross entropy

Network training

- many hyper-parameters to select/adjust/tune, the single most important one being the learn rate

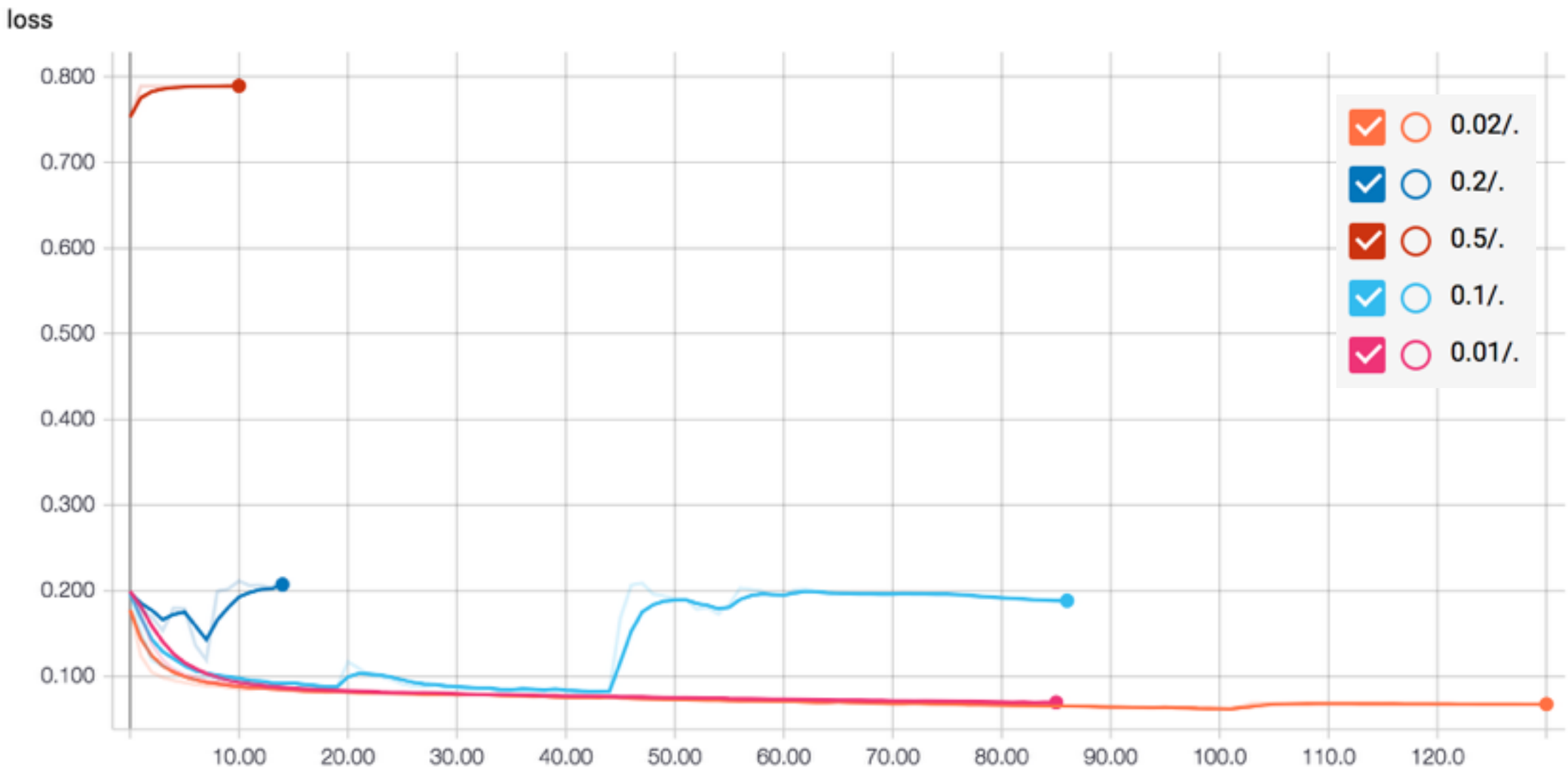
- high learn rates preferable in order to:
  - achieve good generalisation capabilities
  - minimise training time

- no perfect way of judging what a good learn rate is, thus testing different learn rates is recommended

- early stopping can be used to prevent overfitting

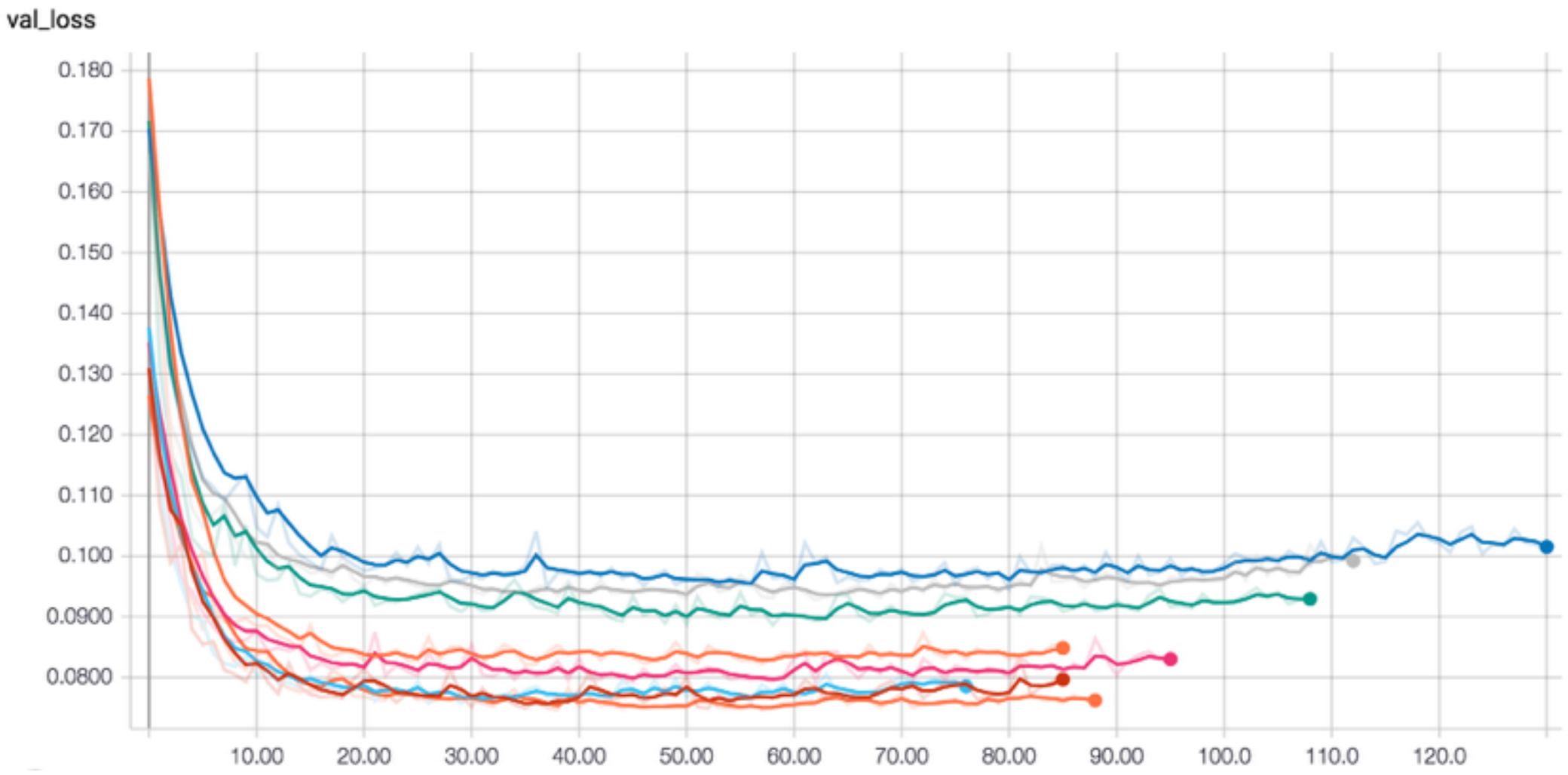
Network training

different learn rates



Network training

overfitting / early stopping



Demo 2

http://localhost:8888/notebooks/Part_3b_RNN_Onset_Detection.ipynb

RNNs for beat tracking

- beats can also be inferred with a RNN
- network trained to learn the characteristics of beats instead of onsets

- more temporal context needed to model beats (a beat period at 120 bpm spans 500 ms, a whole bar in 4/4 time signature 2 s)

RNNs for beat tracking

Example: RNN used in Böck et al. 2014

- network architecture similar to onset detection
- LSTM instead of tanh-units
- advantageous to widen targets
  (hot encode 2 consecutive frames)

Post-processing

Often state-of-the-art results can only be obtained with appropriate post-processing:

- simple methods, e.g. thresholding
- advanced methods, e.g. dynamic programming
- specialised methods incorporating domain knowledge or constraints, e.g. dynamic bayesian networks (DBNs) in Böck et al. 2014

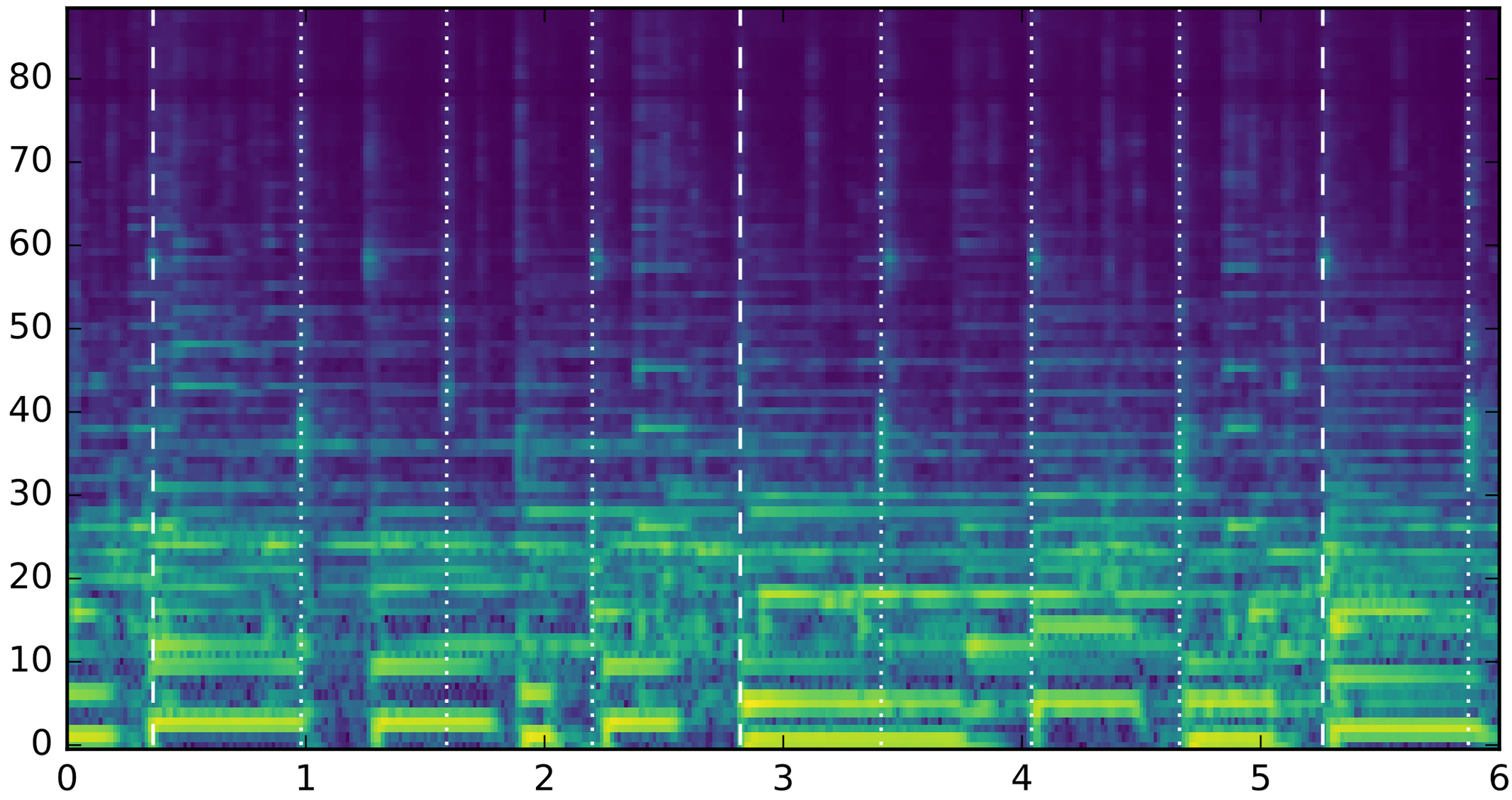
RNNs for downbeat tracking

Example: RNN used in Böck et al. 2016

- network architecture similar to beat tracking
- advantageous to widen targets
  (hot encode 2 consecutive frames)
- softmax output layer containing 3 units:
  no_beat, *beat*, *downbeat*

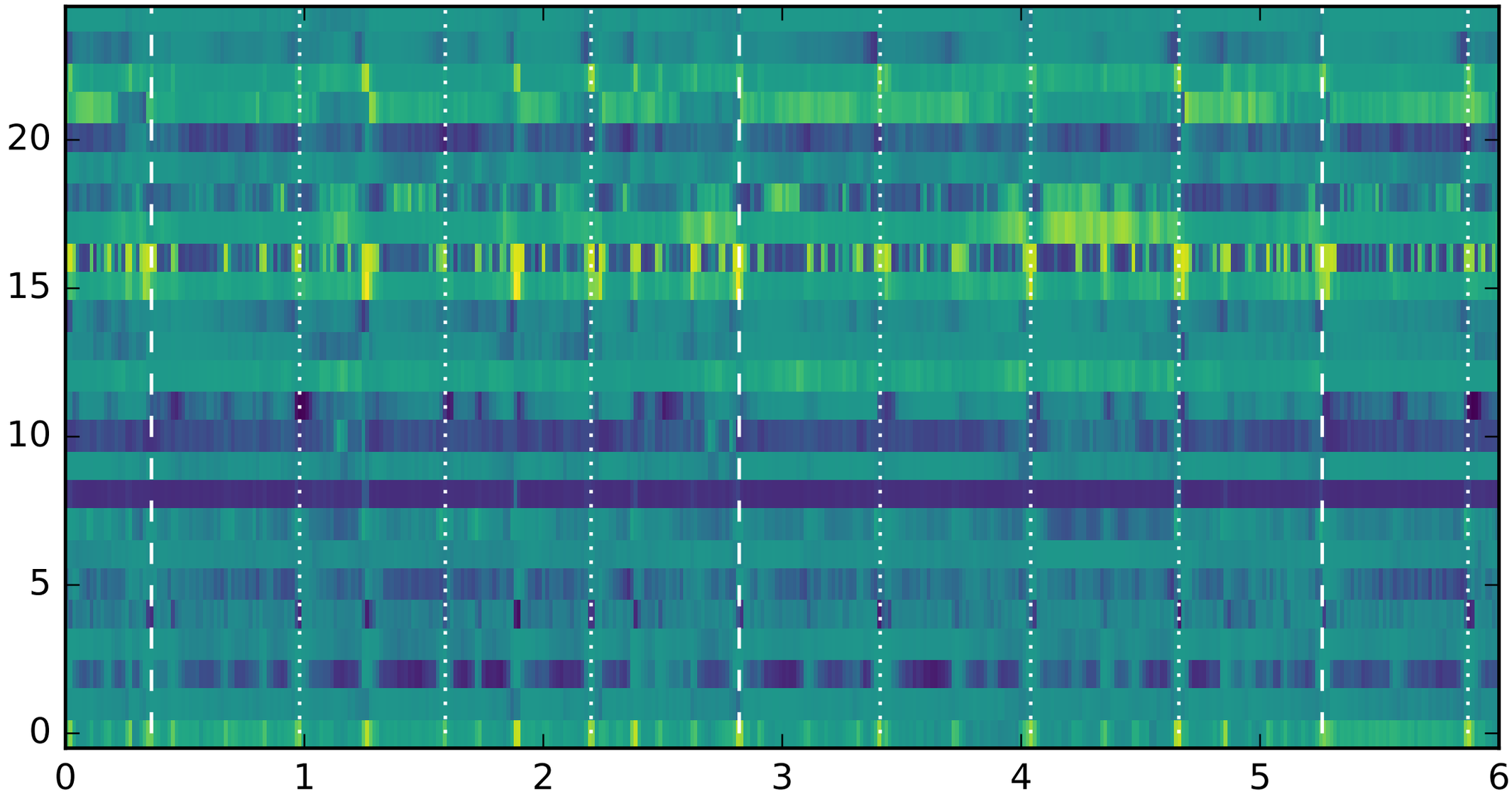
RNNs for downbeat tracking

Input features (only spectrogram of 1 FFT size shown)



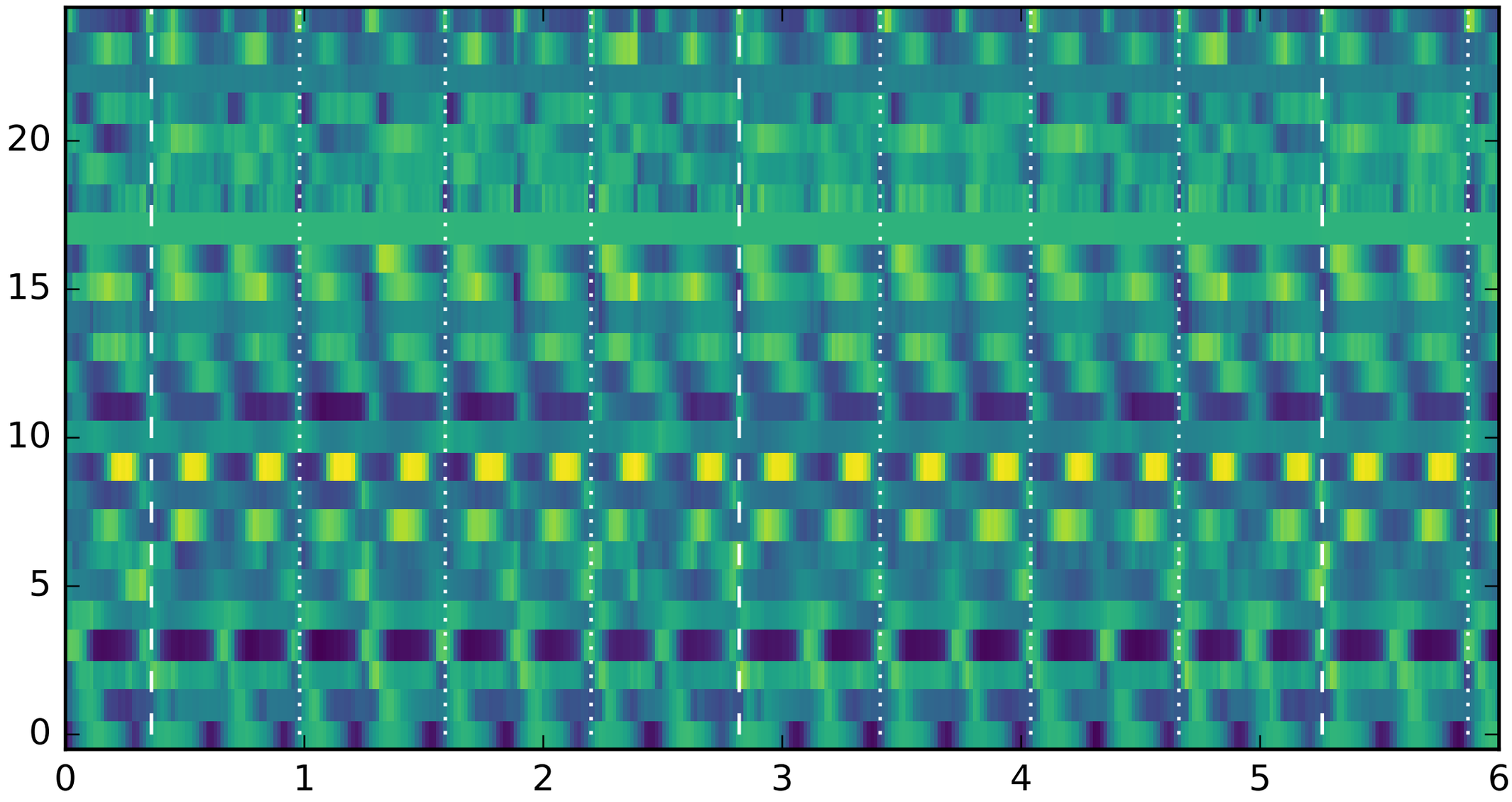
RNNs for downbeat tracking

Activations of first hidden layer



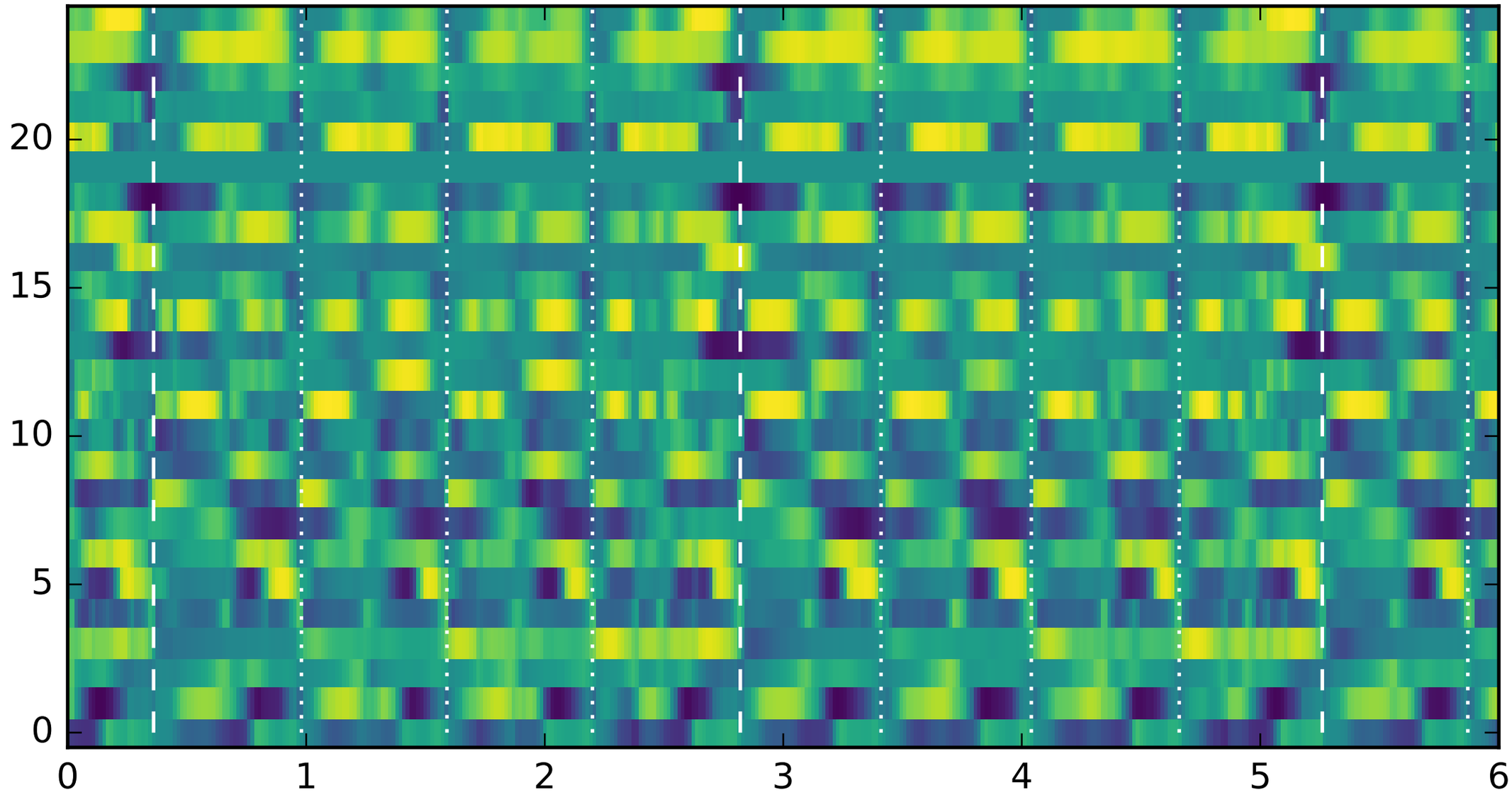
RNNs for downbeat tracking

Activations of second hidden layer



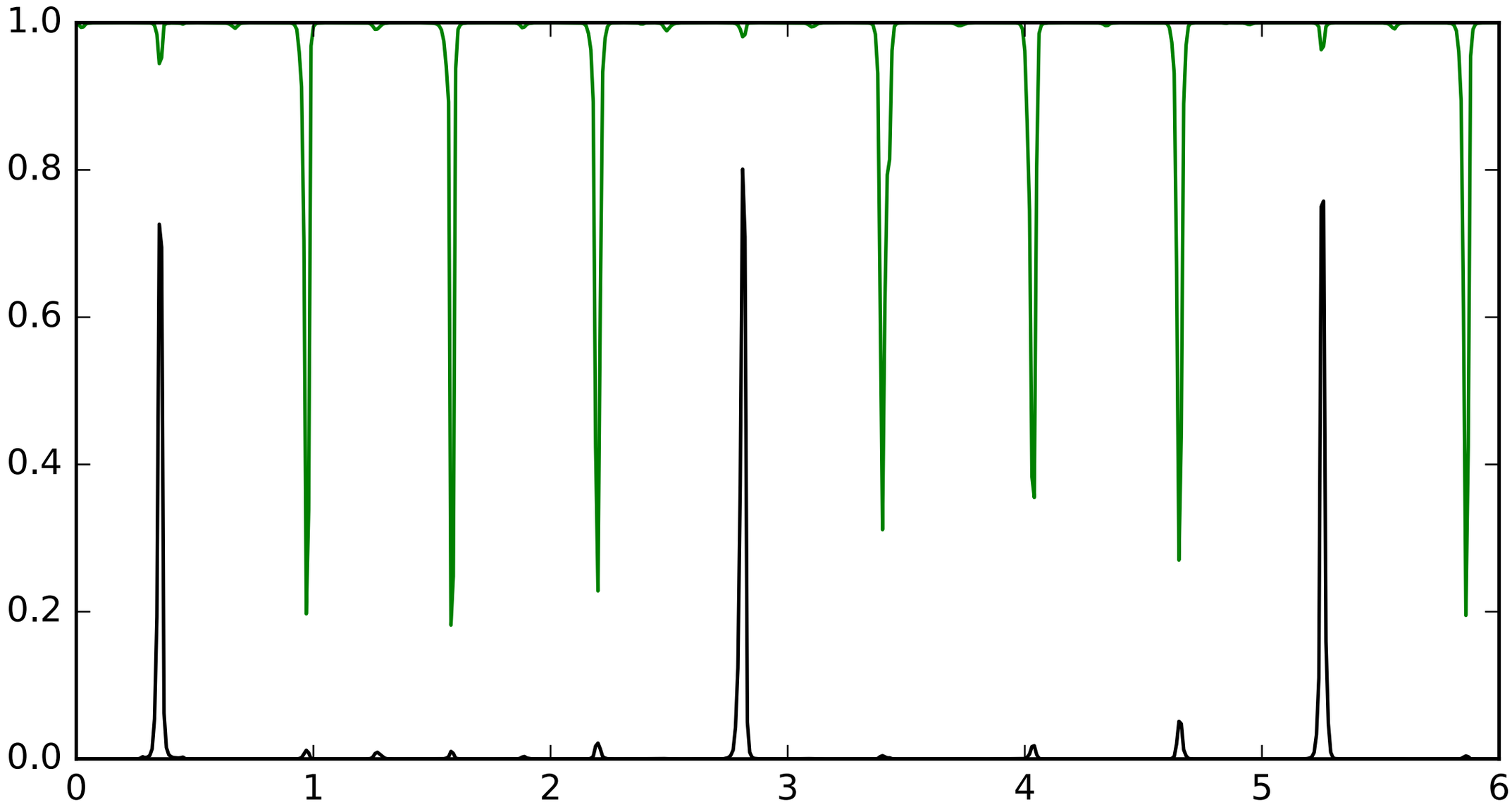
RNNs for downbeat tracking

Activations of third hidden layer



RNNs for downbeat tracking

Activations of output layer



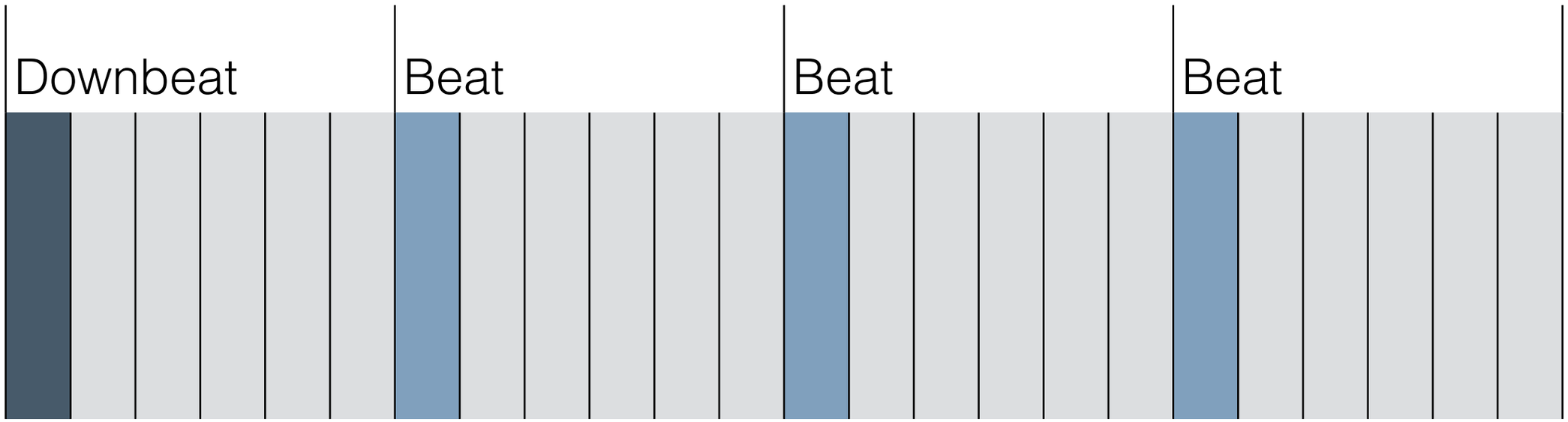
Post-processing

DBN used in Böck et al. 2016 for decoding the output of an RNN predicting beat and downbeat positions

- models bars with arbitrary number of beats
- uses state space proposed by Krebs et al. 2015
- mutually infers tempo, meter and phase of sequence
- tempo changes are allowed at the bar level
- different meters modelled with multiple state spaces

Post-processing

State space of the DBN used in Böck et al. 2016 (simplification)



4/4 time signature

Tips & tricks

- always check your data!
- start simple:
    - use proven input features, e.g. 2048 FFT size (@44.1kHz), filter the magnitudes, scale them logarithmically
    - small network (few layers & units per layer), standard parametrisation, e.g. SGD with momentum
- test different learn rates
- always check what you're interested in (not just the loss)
- add complexity only if everything is working as expected

Common problems

- model not converging: learn rate too high?
- low performance: learn rate too low?
- only predicting one class: unbalanced targets, widen them?
- unstable behaviour: clip gradients?
- not generalising well: network too large?
- not learning anything meaningful: remove LSTM units,
  try a vanilla RNN with tanh-units

References

S. Böck, F. Krebs, and G. Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. *ISMIR*, 2014.

S. Böck, F. Krebs, and G. Widmer. Joint Beat and Downbeat Tracking with Recurrent Neural Networks. *ISMIR*, 2016.

S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. madmom: a new Python Audio and Music Signal Processing Library. *ACMMM,* 2016.

M. E. P. Davies and M. D. Plumbley. Context-Dependent Beat Tracking of Musical Audio. *IEEE TASLP*, 2007.

S. Dixon. Evaluation of the Audio Beat Tracking System BeatRoot. *JNMR*, 2007.

S. Durand, J. P. Bello, B. David, and G. Richard. Feature Adapted Convolutional Neural Networks for Downbeat Tracking. *ICASSP*, 2016.

S. Dixon. Evaluation of the Audio Beat Tracking System BeatRoot. *JNMR*, 2007.

D. Ellis. Beat tracking by dynamic programming. *JNMR*, 2007.

J. Hockman, M. E. P. Davies, and I. Fujinaga. One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass. *ISMIR*, 2012.

A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the meter of acoustic musical signals. *IEEE TASLP*, 2006.

F. Krebs, S. Böck, and G. Widmer. Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio. *ISMIR*, 2013.

F. Krebs, S. Böck, and G. Widmer. An Efficient State Space Model for Joint Tempo and Meter Tracking. *ISMIR*, 2015.

H. Papadopoulos, and G Peeters. Joint Estimation of Chords and Downbeats From an Audio Signal. IEEE TASLP, 2011.

Thanks!

Questions?