Acronyms

MLP: multi layer perceptron ≡ feed-forward neural network
RNN: recurrent neural network
LSTM: long-short term memory
CNN: convolutional neural network
BN: batch normalization

..the following slides assume you know these concepts!
Outline

Chronology: the big picture

Audio classification: state-of-the-art review

Music audio tagging as a study case
Outline

Chronology: the big picture

Audio classification: state-of-the-art review

Music audio tagging as a study case
“Deep learning & music” papers: milestones

# papers

- 1986
- 1988
- 1990
- 1992
- 1994
- 1996
- 1998
- 2000
- 2002
- 2004
- 2006
- 2008
- 2010
- 2012
- 2014
- 2016
“Deep learning & music” papers: milestones

RNN from symbolic data for automatic **music composition** (Todd, 1988)

MLP from symbolic data for automatic **music composition** (Lewis, 1988)
"Deep learning & music" papers: milestones

**LSTM** from symbolic data for automatic **music composition** (Eck and Schmidhuber, 2002)

**RNN** from symbolic data for automatic **music composition** (Todd, 1988)

**MLP** from symbolic data for automatic **music composition** (Lewis, 1988)
Deep learning & music” papers: milestones

- **MLP** learns from spectrograms data for note **onset detection** *(Marolt et al, 2002)*
- **LSTM** from symbolic data for automatic **music composition** *(Eck and Schmidhuber, 2002)*
- **RNN** from symbolic data for automatic **music composition** *(Todd, 1988)*
- **MLP** from symbolic data for automatic **music composition** *(Lewis, 1988)*
Deep learning & music papers: milestones

**CNN** learns from spectrograms for music audio **classification** *(Lee et al., 2009)*

**MLP** learns from spectrograms data for note **onset detection** *(Marolt et al., 2002)*

**LSTM** from symbolic data for automatic **music composition** *(Eck and Schmidhuber, 2002)*

**RNN** from symbolic data for automatic **music composition** *(Todd, 1988)*

**MLP** from symbolic data for automatic **music composition** *(Lewis, 1988)*
“Deep learning & music” papers: milestones

End-to-end learning for music audio classification (Dieleman et al., 2014)

CNN learns from spectrograms for music audio classification (Lee et al., 2009)

MLP learns from spectrograms data for note onset detection (Marolt et al, 2002)

LSTM from symbolic data for automatic music composition (Eck and Schmidhuber, 2002)

RNN from symbolic data for automatic music composition (Todd, 1988)

MLP from symbolic data for automatic music composition (Lewis, 1988)
“Deep learning & music” papers: milestones

- **End-to-end learning** for music audio **classification** (Dieleman et al., 2014)
- **CNN** learns from spectrograms for music audio **classification** (Lee et al., 2009)
- **MLP** learns from spectrograms data for note **onset detection** (Marolt et al, 2002)
- **LSTM** from symbolic data for automatic **music composition** (Eck and Schmidhuber, 2002)
- **RNN** from symbolic data for automatic **music composition** (Todd, 1988)
- **MLP** from symbolic data for automatic **music composition** (Lewis, 1988)
“Deep learning & music” papers: data trends

- Symbolic data
- Spectrograms data
- Raw audio data
“Deep learning & music” papers: some references

Dieleman et al., 2014 – **End-to-end learning for music audio**
in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*

Lee et al., 2009 – **Unsupervised feature learning for audio classification using convolutional deep belief networks**
in *Advances in Neural Information Processing Systems (NIPS)*

Marolt et al., 2002 – **Neural networks for note onset detection in piano music**
in *Proceedings of the International Computer Music Conference (ICMC)*

Eck and Schmidhuber, 2002 – **Finding temporal structure in music: Blues improvisation with LSTM recurrent networks**
in *Proceedings of the Workshop on Neural Networks for Signal Processing*

Todd, 1988 – **A sequential network design for musical applications**
in *Proceedings of the Connectionist Models Summer School*

Lewis, 1988 – **Creation by Refinement: A creativity paradigm for gradient descent learning networks**
in *International Conference on Neural Networks*
Outline

Chronology: the big picture

Audio classification: state-of-the-art review

Music audio tagging as a study case
Which is our goal / task?

input \rightarrow \text{machine learning} \rightarrow \text{output}

waveform

or any audio representation!

\textit{deep learning model}

\begin{align*}
\text{phonetic transcription} \\
\text{describe music with tags} \\
\text{event detection}
\end{align*}
The deep learning pipeline

- input
- front-end
- back-end
- output

- waveform
- or any audio representation!
- phonetic transcription
- describe music with tags
- event detection
The deep learning pipeline: input?
How to format the input (audio) data?

**Waveform**
end-to-end learning

**Pre-processed waveform**
e.g.: spectrogram
The deep learning pipeline: front-end?

input → front-end

waveform

spectrogram
<table>
<thead>
<tr>
<th>based on domain knowledge?</th>
<th>filters config?</th>
<th>input signal?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>waveform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pre-processed waveform</td>
</tr>
</tbody>
</table>
CNN front-ends for audio classification

Waveform end-to-end learning

Pre-processed waveform e.g.: spectrogram

Sample-level

Small-rectangular filters
Based on domain knowledge?

No

Filters config?

Minimal filter expression

Input signal?

Waveform

Sample-level

Pre-processed waveform

Small-rectangular filters
Domain knowledge to design CNN front-ends

Waveform
dend-to-end learning

Pre-processed waveform
e.g.: spectrogram
Domain knowledge to design CNN front-ends

Waveform end-to-end learning

- filter length: 512
- stride: 256

Pre-processed waveform e.g.: spectrogram

- window length? 
- hop size?

Explicitly tailoring the CNN towards learning temporal or timbral cues

frame-level

vertical or horizontal filters
<table>
<thead>
<tr>
<th>based on domain knowledge?</th>
<th>filters config?</th>
<th>input signal?</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>minimal filter expression</td>
<td>waveform</td>
</tr>
<tr>
<td>yes</td>
<td>single filter shape in 1st CNN layer</td>
<td>pre-processed waveform</td>
</tr>
</tbody>
</table>

- sample-level:
  - 3x1 3x1 ... 3x1 3x1

- frame-level:
  - vertical OR horizontal
  - vertical or horizontal

- small-rectangular filters:
  - 3x3 3x3 ... 3x3 3x3
DSP wisdom to design CNN front ends

Waveform
end-to-end learning

Pre-processed waveform
e.g.: spectrogram

Efficient way
to represent
4 periods!

Explicitly tailoring the CNN towards
learning temporal and timbral cues

Frame-level (many shapes!)

Vertical and/or horizontal
<table>
<thead>
<tr>
<th>Based on domain knowledge?</th>
<th>Filters config?</th>
<th>Waveform</th>
<th>Pre-processed waveform</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>minimal filter expression</td>
<td>sample-level</td>
<td>small-rectangular filters</td>
</tr>
<tr>
<td></td>
<td>3x1 3x1 ... 3x1 3x1</td>
<td>frame-level</td>
<td>vertical OR horizontal</td>
</tr>
<tr>
<td>yes</td>
<td>single filter shape in 1\textsuperscript{st} CNN layer</td>
<td>frame-level</td>
<td>vertical AND/OR horizontal</td>
</tr>
<tr>
<td></td>
<td>3x1 3x1 ... 3x1 3x1</td>
<td>frame-level</td>
<td>or</td>
</tr>
<tr>
<td>yes</td>
<td>many filter shapes in 1\textsuperscript{st} CNN layer</td>
<td>frame-level</td>
<td>vertical AND/OR horizontal</td>
</tr>
</tbody>
</table>
CNN front-ends for audio classification

**Sample-level:** Lee et al., 2017 – Sample-level Deep Convolutional Neural Networks for Music Auto-tagging Using Raw Waveforms in Sound and Music Computing Conference (SMC)


**Frame-level (single shape):** Dieleman et al., 2014 – End-to-end learning for music audio in International Conference on Acoustics, Speech and Signal Processing (ICASSP)

**Vertical:** Lee et al., 2009 – Unsupervised feature learning for audio classification using convolutional deep belief networks in Advances in Neural Information Processing Systems (NIPS)

**Horizontal:** Schluter & Bock, 2014 – Improved musical onset detection with convolutional neural networks in International Conference on Acoustics, Speech and Signal Processing (ICASSP)

**Frame-level (many shapes):** Zhu et al., 2016 – Learning multiscale features directly from waveforms in arXiv:1603.09509

**Vertical and horizontal (many shapes):** Pons, et al., 2016 – Experimenting with musically motivated convolutional neural networks in 14th International Workshop on Content-Based Multimedia Indexing
The deep learning pipeline: back-end?

- Input
- Front-end: waveform, several CNN architectures
- Back-end

?
What is the back-end doing?

Back-end *adapts* a variable-length feature map to a fixed output-size.
Back-ends for variable-length inputs

- **Temporal pooling**: max-pool or average-pool the temporal axis
  Pons et al., 2017 – *End-to-end learning for music audio tagging at scale*, in proceedings of the ML4Audio Workshop at NIPS.

- **Attention**: weighting latent representations to what is important

- **RNN**: summarization through a deep temporal model
  Vogl et al., 2018 – *Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks*, In proceedings of the ISMIR conference.

..*music is generally of variable length!*
Back-ends for fixed-length inputs

*Common trick: let's assume a fixed-length input*

- **Fully convolutional stacks**: adapting the input to the output with a stack of CNNs & pooling layers.


- **MLP**: map a *fixed-length* feature map to a *fixed-length* output

  Schluter & Bock, 2014 – *Improved musical onset detection with convolutional neural networks* in proceedings of the ICASSP.

  ..such trick works very well!
The deep learning pipeline: output

- **Input**: waveform, spectrogram
- **Front-end**: several CNN architectures
- **Back-end**: MLP, RNN, attention
- **Output**: phonetic transcription, describe music with tags, event detection
The deep learning pipeline: output

- **Input**: waveform, spectrogram
- **Front-end**: several CNN architectures
- **Back-end**: MLP, RNN, attention
- **Output**: phonetic transcription, describe music with tags, event detection
Outline

Chronology: the big picture

Audio classification: state-of-the-art review

Music audio tagging as a study case

Pons et al., 2017. End-to-end learning for music audio tagging at scale, in ML4Audio Workshop at NIPS  
Summer internship @ Pandora
The deep learning pipeline: input?

describe music with tags
How to format the input (audio) data?

**waveform**

already: zero-mean & one-variance

**log-mel spectrogram**

- **STFT & mel mapping**
  reduces size of the input by removing perceptually irrelevant information

- **logarithmic compression**
  reduces dynamic range of the input

- **zero-mean & one-variance**

**NO pre-processing!**
The deep learning pipeline: input?

input → front-end → back-end → output

- waveform
- log-mel spectrogram
- describe music with tags
The deep learning pipeline: front-end?

input → front-end → back-end → output

waveform
log-mel spectrogram

? describe music with tags
<table>
<thead>
<tr>
<th>based on domain knowledge?</th>
<th>filters config?</th>
<th>input signal?</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>minimal filter expression</td>
<td>waveform</td>
</tr>
<tr>
<td></td>
<td>3x1 3x1 ... 3x1 3x1</td>
<td>sample-level</td>
</tr>
<tr>
<td>yes</td>
<td>single filter shape in 1st CNN layer</td>
<td>pre-processed waveform</td>
</tr>
<tr>
<td></td>
<td>vertical OR horizontal</td>
<td>vertical AND/OR horizontal</td>
</tr>
<tr>
<td>yes</td>
<td>many filter shapes in 1st CNN layer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
</tbody>
</table>

- **Waveform**
  - Sample-level: 3x1 3x1 ... 3x1 3x1
  - Frame-level: vertical OR horizontal
  - Frame-level: vertical AND/OR horizontal

- **Pre-processed waveform**
  - Small-rectangular filters: 3x3 3x3 ... 3x3 3x3
Studied front-ends: waveform model

(sample-level) (Lee et al., 2017)
Studied front-ends: spectrogram model

vertical and horizontal
musically motivated CNNs

(Pons et al., 2016 – 2017)
The deep learning pipeline: front-end?

- **Input**: waveform
  - log-mel spectrogram
- **Front-end**: sample-level
  - vertical and horizontal
- **Back-end**: describe music with tags
- **Output**
The deep learning pipeline: back-end?

- **Input**: waveform, log-mel spectrogram
- **Front-end**: sample-level, vertical and horizontal
- **Back-end**: describe music with tags
- **Output**
Studied back-end: music is of variable length!

Temporal pooling

(Dieleman et al., 2014)
The deep learning pipeline: back-end?

- **Input**: waveform, log-mel spectrogram
- **Front-end**: sample-level, vertical and horizontal
- **Back-end**: temporal pooling
- **Output**: describe music with tags
MagnaTT

Million song dataset

1M songs

25k songs

250k songs
MagnaTT

Million song dataset

spectrograms > waveforms

25k songs

250k songs

1M songs
MagnaTT

Million song dataset

25k songs

1M songs

250k songs

spectrograms > waveforms

waveforms > spectrograms
Let’s listen to some music: our model in action

acoustic
string ensemble
classical music
period baroque
compositional dominance of lead vocals
major
Deep learning architectures for music audio classification: a personal (re)view

Jordi Pons

jordipons.me – @jordiponsdotme

Music Technology Group
Universitat Pompeu Fabra, Barcelona