Music Generation Using Machine Learning

Seminar Computer Music SS 2017

Michael Krause

RWTH Aachen

June 27, 2017

The problem

Can a computer create new music?

- Applications: assisted composition, video games, music analysis
- Two broad steps: Composing and creating audio signal
- Composing: melody, rhythm, structure, instrumentation...

Outline of the talk

History

Current approaches

- Unit selection using deep learning
- Evolutionary multi-objective optimization

Conclusion

History

History: AI research and music generation

- General trend in AI: move from symbolic methods (GOFAI) to machine learning / data-based approaches
- Formal grammar approach
  - Create structure through (stochastic) rewriting rules
  - Hand-made or learned from data
  - Terminal symbols can be notes/chords/measures...
- Markov Chains
  - States: notes or measures
  - Transitions: probabilities learned from examples or hand-made
  - No higher level structure
- More recently: Machine Learning (e.g. neural networks)

Current approaches

Now: Two approaches for melody generation

- Melody: essential for most music
- Competing approaches:
  - Bretan et al. (2016): uses deep learning
  - de León et al. (2016): evolutionary algorithm with machine learning based fitness functions

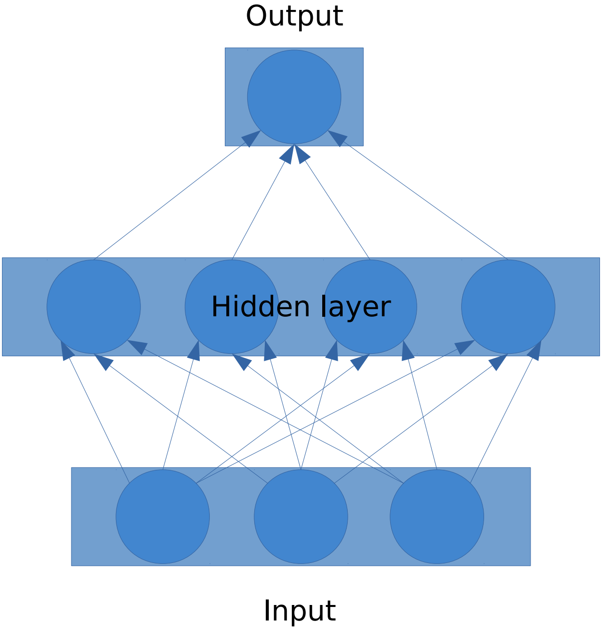
Current approaches:
Unit selection using deep learning

Setup

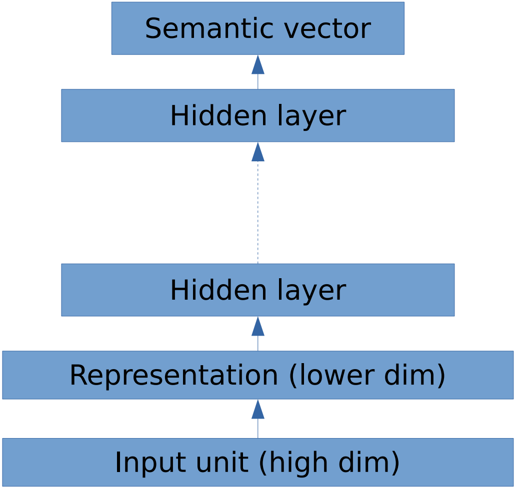
- Given: > 4000 melodies (in particular Jazz)
- Divide into units (1, 2 or 4 measures of melody)
- Create new melodies by concatenating units
- Connected units should
  - share semantic similarity
  - have pleasant transitions

$\rightarrow$ Train two neural nets to encode these properties

Neural nets: Basic example

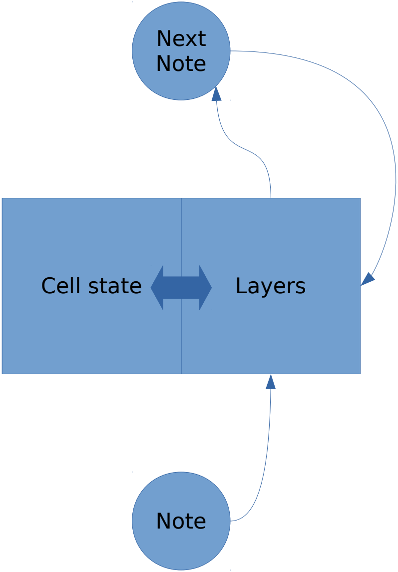


Neural net for semantic similarity



Idea: train semantic vectors for consecutive units to be similar

Neural net for note transitions



Melody generation: Ranking and concatenation

- Start with some measures of music given
- Rank all units given the previous measures
  - Firstly: according to similarity
  - Secondly: according to note transitions
- Concatenate measures with highest ranked unit and repeat

(listen to audio example)

Current approaches: Evolutionary multi-objective optimization

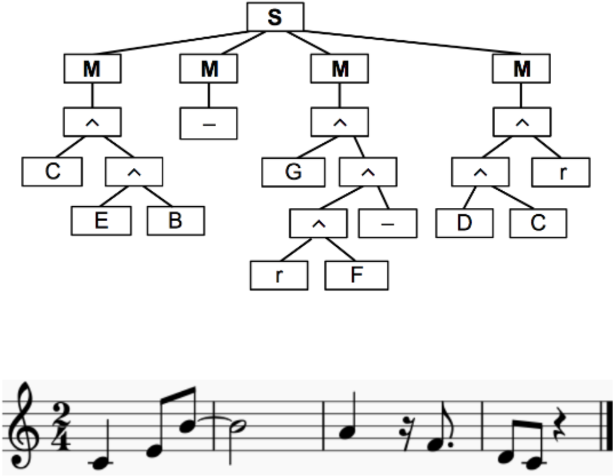
Evolutionary optimization overview

- Optimization problem in general: candidate solutions and objective function
- Here: candidates set is "population", objective is high "fitness"
- The phenotype (appearance) of a candidate may be different from its genotype (encoding/representation)
- Initial candidates are repeatedly evolved through
  - Variation: modify candidates through crossover and mutation
  - Selection: retain only candidates with good objective value

Evolutionary multi-objective optimization

- Challenges for evolutionary approach to music
  - How to encode genotype?
  - How to generate candidates?
  - How to evaluate fitness? User interaction?
- In the present paper
  - Melodies are trees
  - Initial melodies are generated randomly
  - Fitness consists of multiple measures, partly learned from existing melodies of a given style

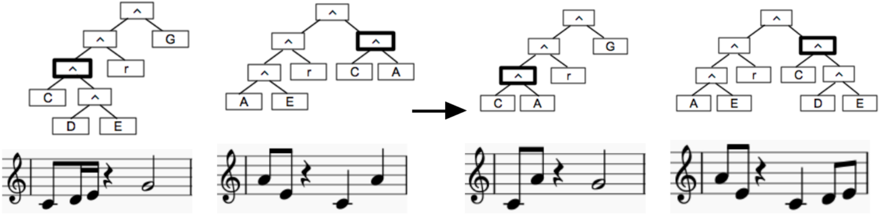
Melodies as trees



(Figure 1 from the paper)

Crossover and mutation

- "Fittest" candidates are admitted to crossover and mutation
- Mutation: With small probability a node is randomly replaced
- Crossover: Nodes swapped between pairs of candidates (figure)



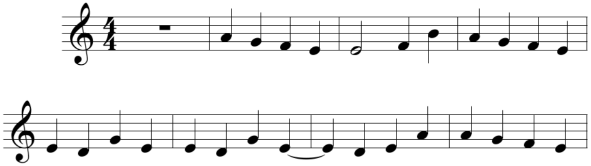
(Figure 2 from the paper)

Fitness functions

- 12 separate ones used
- Function parameters based on body of melodies in a style
- Examples:
  - Global descriptors (number of notes, most repeated interval, average pitch and deviation etc.)
  - Language model (probability of note after previous 4 notes)
  - Music analysis (e.g. number of segments in melody)
- Candidate A *pareto-optimal*: for all candidates B with higher fitness than A in some function, B must have lower fitness than A in another function

Results (fitness trained for different styles)

Celtic #3



Pre-bop jazz #1

(listen to audio example)

Results (fitness trained for different styles)

(listen to audio example)

Conclusion

Conclusion

Take-home concepts:

- Move from symbolic to machine learning methods
- Unit selection using neural networks
- Melody tree as result of evolutionary optimization

Possible further work:

- Variety vs. higher level structure
- Extend approaches to generate harmonization
- Quality measures?

Papers

Mason Bretan, Gil Weinberg, and Larry Heck.
A Unit Selection Methodology for Music Generation Using Deep Neural Networks.
arXiv:1612.03789 [cs.SD], 2016.

Pedro J. Ponce de León, José M. Iñesta, Jorge Calvo-Zaragoza, and David Rizo.
Data-based melody generation through multi-objective evolutionary computation.
Journal of Mathematics and Music, 10(2):173–192, 2016.

Jose David Fernández and Francisco Vico.
AI Methods in Algorithmic Composition: A Comprehensive Survey.
Journal Of Artificial Intelligence Research, 48:513–582, 2013.

More sources

- Example music composed using the DNN approach
- Example music composed using the evolutionary approach
  I have arranged some of these in MuseScore here, so they can be heard