

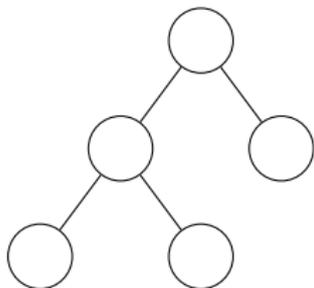
GP-Music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Brad Johanson, Riccardo Poli

Seminarvortrag von Thomas Arnold



1.

- ▶ Generierung von kurzen Musikstücken
- ▶ Einsatz von genetischer Programmierung für hohe Flexibilität
- ▶ Interaktive Evolution - manuelle Bewertung

1.

- ▶ Generierung von kurzen Musikstücken
- ▶ Einsatz von genetischer Programmierung für hohe Flexibilität
- ▶ Interaktive Evolution - manuelle Bewertung

2.

- ▶ Variation und Verbesserung an Algorithmus und Datenstruktur
- ▶ Vereinfachung der manuellen Bewertung

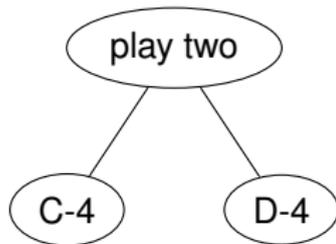
1.
 - ▶ Generierung von kurzen Musikstücken
 - ▶ Einsatz von genetischer Programmierung für hohe Flexibilität
 - ▶ Interaktive Evolution - manuelle Bewertung
2.
 - ▶ Variation und Verbesserung an Algorithmus und Datenstruktur
 - ▶ Vereinfachung der manuellen Bewertung
3.
 - ▶ Einsatz einer automatischen Bewertung

- ▶ Zufällige Generierung einer Grundpopulation mit X Individuen - in diesem Fall monophone Musik-Sequenzen
- ▶ Manuelle Bewertung aller Sequenzen (1 - 100 Punkte)
- ▶ Generierung der nächsten Generation
- ▶ Y Generationen = Wiederholung von Bewertung / Generierung



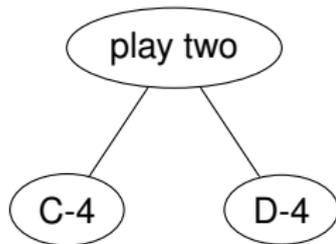
- ▶ Genetische Programmierung = Baumstruktur
- ▶ Terminale sind Noten einer Oktave (C-4, C#4, D-4...) oder Pause (RST)
- ▶ Knoten sind Funktionen auf Noten mit 1-2 Eingaben

- ▶ Genetische Programmierung = Baumstruktur
- ▶ Terminale sind Noten einer Oktave (C-4, C#4, D-4...) oder Pause (RST)
- ▶ Knoten sind Funktionen auf Noten mit 1-2 Eingaben

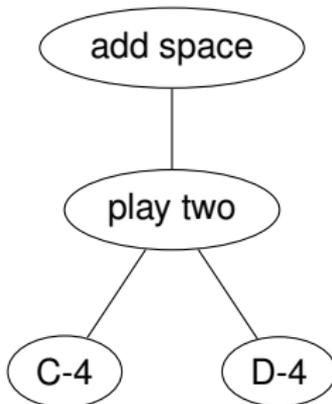


= C-4 D-4

- ▶ Genetische Programmierung = Baumstruktur
- ▶ Terminale sind Noten einer Oktave (C-4, C#4, D-4...) oder Pause (RST)
- ▶ Knoten sind Funktionen auf Noten mit 1-2 Eingaben

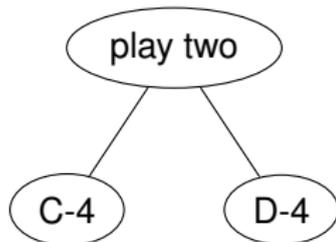


= C-4 D-4

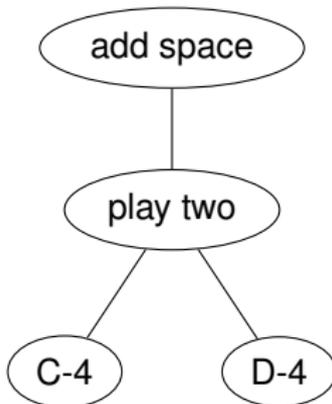


= C-4 RST D-4 RST

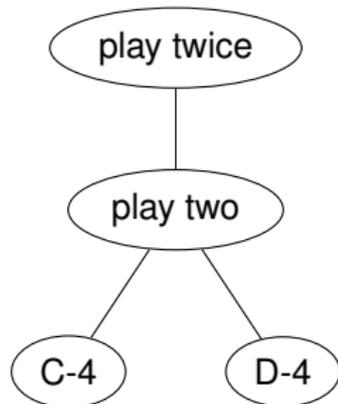
- ▶ Genetische Programmierung = Baumstruktur
- ▶ Terminale sind Noten einer Oktave (C-4, C#4, D-4...) oder Pause (RST)
- ▶ Knoten sind Funktionen auf Noten mit 1-2 Eingaben



= C-4 D-4

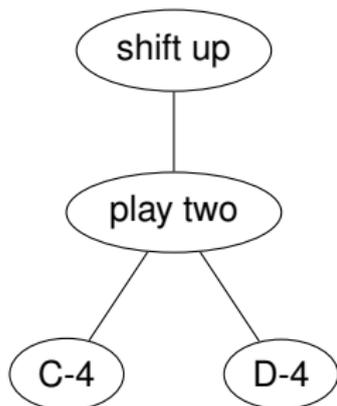


= C-4 RST D-4 RST



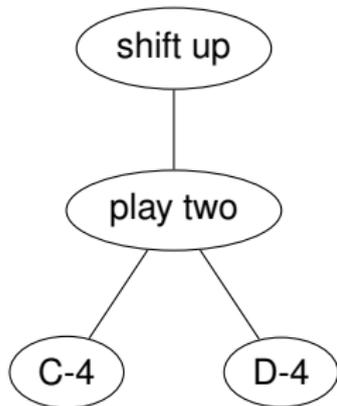
= C-4 D-4 C-4 D-4

- ▶ Genetische Programmierung = Baumstruktur
- ▶ Terminale sind Noten einer Oktave (C-4, C#4, D-4...) oder Pause (RST)
- ▶ Knoten sind Funktionen auf Noten mit 1-2 Eingaben

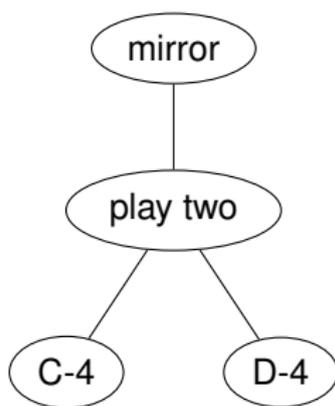


= C#4 D#4

- ▶ Genetische Programmierung = Baumstruktur
- ▶ Terminale sind Noten einer Oktave (C-4, C#4, D-4...) oder Pause (RST)
- ▶ Knoten sind Funktionen auf Noten mit 1-2 Eingaben

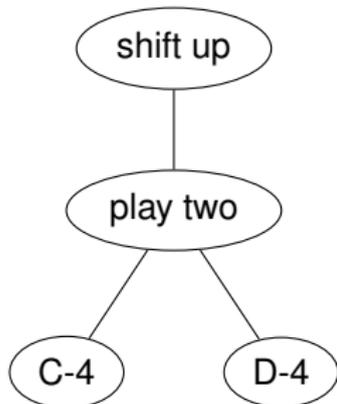


= C#4 D#4

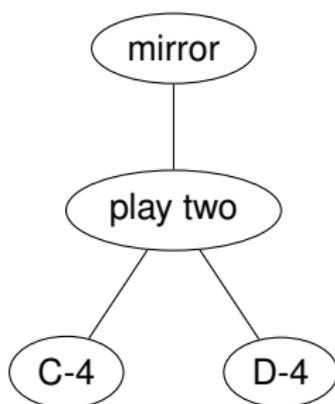


= D-4 C-4

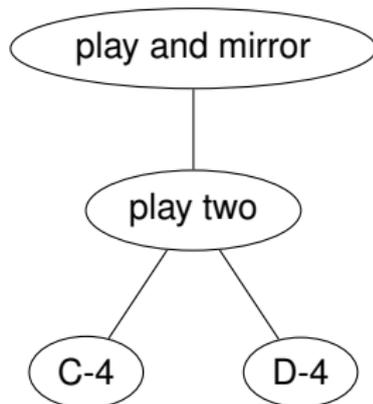
- ▶ Genetische Programmierung = Baumstruktur
- ▶ Terminale sind Noten einer Oktave (C-4, C#4, D-4...) oder Pause (RST)
- ▶ Knoten sind Funktionen auf Noten mit 1-2 Eingaben



= C#4 D#4



= D-4 C-4



= C-4 D-4 D-4 C-4

Bewertung - Simple Interface

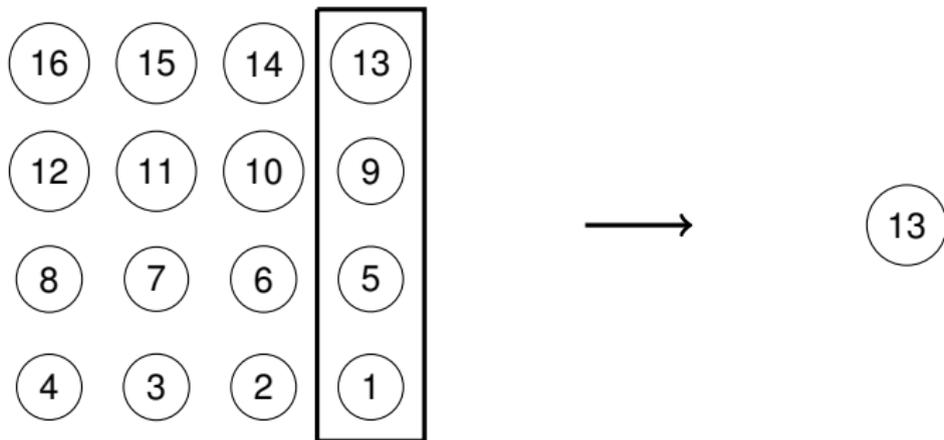
- ▶ Ein Individuum wird abgespielt und bewertet
- ▶ Es folgt das nächste Individuum
- ▶ Generationenwechsel im Ablauf nicht spürbar



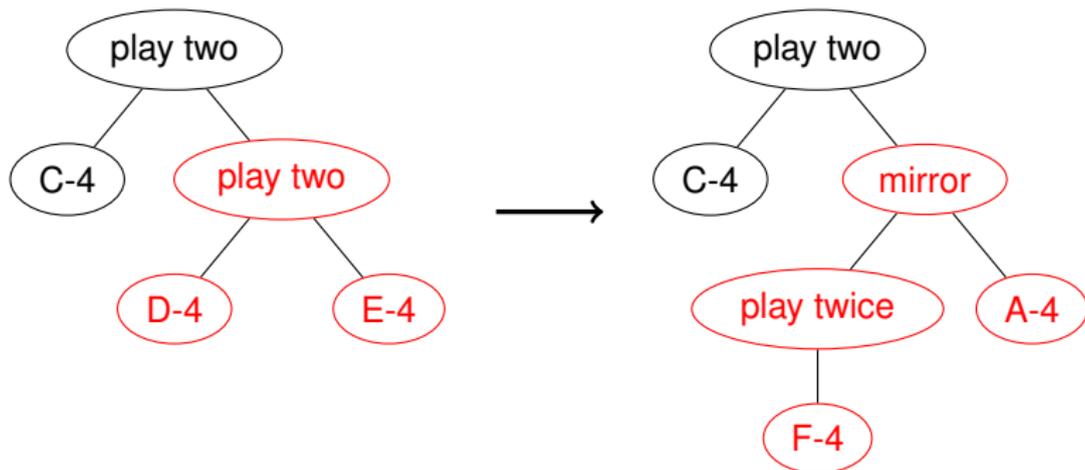


- ▶ 70 % Reproduktion
- ▶ 15 % Mutation
- ▶ 15 % Rekombination
- ▶ Chance jedes Individuums, ausgewählt zu werden, steigt mit der Bewertung

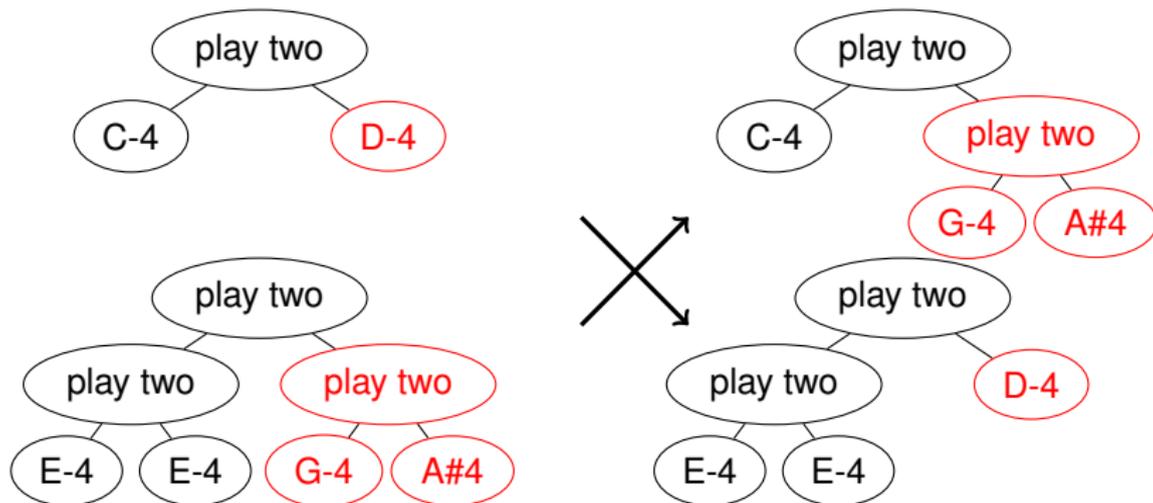
- ▶ 4 individual tournament selection
- ▶ 4 Individuen werden zufällig aus der Population ausgewählt
- ▶ Das Individuum mit der höchsten Bewertung wird in die nächste Generation übernommen



- ▶ Ein Individuum wird ausgewählt
- ▶ Höhere Bewertung = Höhere Wahrscheinlichkeit
- ▶ Zufällige Veränderung an zufälliger Stelle

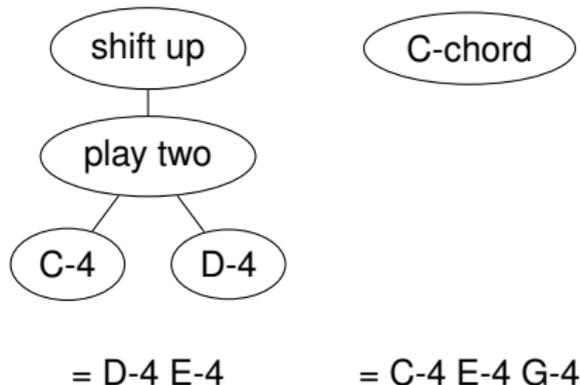


- ▶ Zwei Individuen werden ausgewählt
- ▶ Höhere Bewertung = Höhere Wahrscheinlichkeit
- ▶ An zufälliger Stelle neu kombiniert



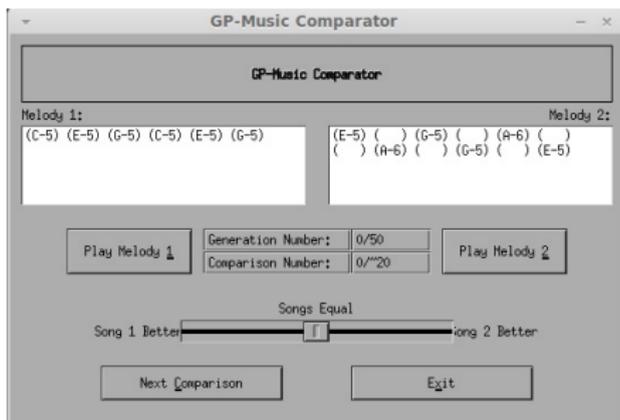
- ▶ Problem: Sehr kurze Sequenzen / sehr lange Sequenzen / sehr viele Pausen
- ▶ Änderung: Feste minimale Länge / maximale Länge / maximale Pausen

- ▶ Problem: Sehr kurze Sequenzen / sehr lange Sequenzen / sehr viele Pausen
- ▶ Änderung: Feste minimale Länge / maximale Länge / maximale Pausen
- ▶ Problem: Kaum harmonische Klänge
- ▶ Änderung: Optionale Festlegung auf C-Dur (C-4, D-4, E-4... statt C-4, C#4, D-4...), Akkord-Terminale

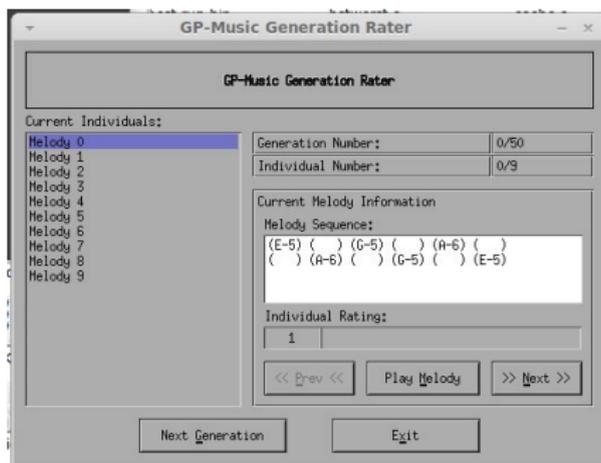


Bewertung - Tournament Interface

- ▶ Immer zwei Individuen im direkten Vergleich
- ▶ A viel besser / A besser / gleich gut / B besser / B viel besser
- ▶ Programm ermittelt aus Vergleichen selbst die Punktzahl 1 - 100
- ▶ "Spannung" durch direkten Wettbewerb, allerdings extrem langwierig (Jeder gegen Jeden!)



- ▶ Alle Individuen der aktuellen Generation in einer Liste
- ▶ Bewertung von reproduzierten Individuen bleibt sichtbar
- ▶ Vorteil: Besserer Überblick, Vergleichbarkeit = Komfort für Bewerter



- ▶ Hauptproblem: Manuelle Bewertung
- ▶ Sehr zeitaufwändig, eintönig, erfordert Konzentration
- ▶ Ansatz: Bewertungen des Menschen in neuronales Netz einfüttern, um so eine Bewertungsfunktion zu lernen
- ▶ Weitere Bewertungen automatisch durchführen lassen

- ▶ Für jede Sequenz wird ein separates, neurales Netzwerk aufgebaut
- ▶ Einen Eingabeknoten für jede Note bzw. Pause
- ▶ Sequenzen beliebiger Länge sind möglich

49 49 -1 51 51 -1 49 49

C-4, C-4, RST, D-4, D-4, RST, C-4, C-4

- ▶ Verbindung zur nächsthöheren Ebene mit 2 Variablen:
- ▶ Level X Spread: Anzahl der Verbindungen jedes Knoten der Ebene X mit tieferen Knoten
- ▶ Level X Delta: Abstand der Verbindungen (bestimmt Überlappung)
- ▶ Beispiel: Level 1 Spread = 4, Level 1 Delta = 2

Ebene 1

Ebene 1

Ebene 1

49

49

-1

51

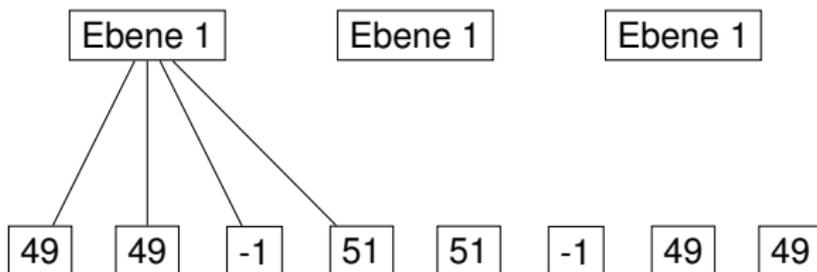
51

-1

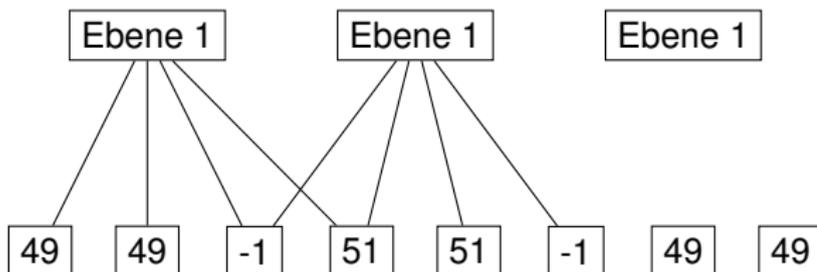
49

49

- ▶ Verbindung zur nächsthöheren Ebene mit 2 Variablen:
- ▶ Level X Spread: Anzahl der Verbindungen jedes Knoten der Ebene X mit tieferen Knoten
- ▶ Level X Delta: Abstand der Verbindungen (bestimmt Überlappung)
- ▶ Beispiel: Level 1 Spread = 4, Level 1 Delta = 2

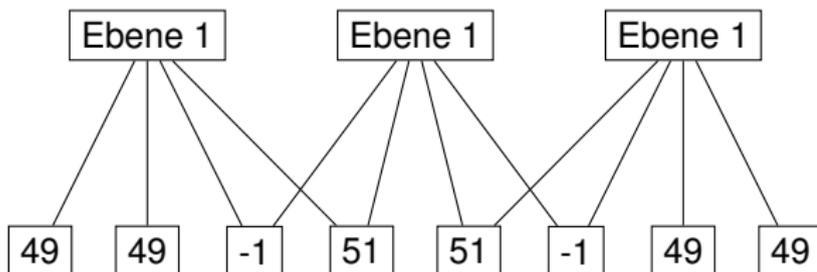


- ▶ Verbindung zur nächsthöheren Ebene mit 2 Variablen:
- ▶ Level X Spread: Anzahl der Verbindungen jedes Knoten der Ebene X mit tieferen Knoten
- ▶ Level X Delta: Abstand der Verbindungen (bestimmt Überlappung)
- ▶ Beispiel: Level 1 Spread = 4, Level 1 Delta = 2



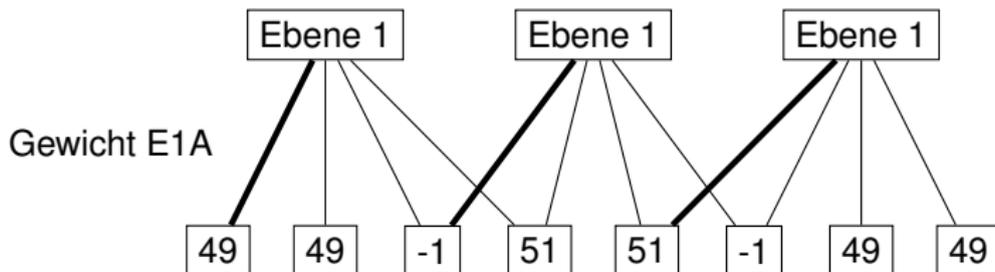
2 Knoten Abstand = Delta

- ▶ Verbindung zur nächsthöheren Ebene mit 2 Variablen:
- ▶ Level X Spread: Anzahl der Verbindungen jedes Knoten der Ebene X mit tieferen Knoten
- ▶ Level X Delta: Abstand der Verbindungen (bestimmt Überlappung)
- ▶ Beispiel: Level 1 Spread = 4, Level 1 Delta = 2

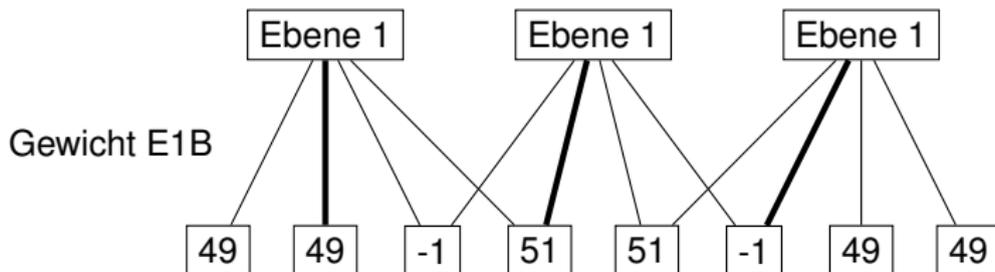


2 Knoten Abstand = Delta

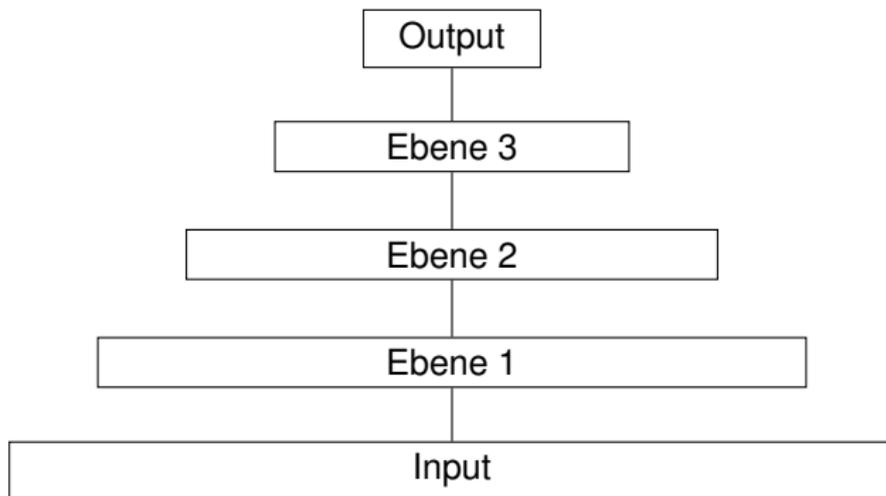
- ▶ Jeweils X-te Verbindung pro Ebene hat gleiche Gewichte
- ▶ Werte über alle Netzwerke (Sequenzen) übergreifend



- ▶ Jeweils X-te Verbindung pro Ebene hat gleiche Gewichte
- ▶ Werte über alle Netzwerke (Sequenzen) übergreifend



- ▶ Insgesamt 3 Ebenen zwischen Input und Output
- ▶ Output liefert wieder Bewertung zwischen 1 und 100





- ▶ Zuerst manuelle Testreihen:
- ▶ Funktionen nur 'play two' und 'add space', keine Akkorde:
Schlechte, unstrukturierte Ergebnisse (Beispiel Sequenz)



- ▶ Zuerst manuelle Testreihen:
- ▶ Funktionen nur 'play two' und 'add space', keine Akkorde:
Schlechte, unstrukturierte Ergebnisse (Beispiel Sequenz)
- ▶ Alle Funktionen, keine Akkorde
Besser, gleichmäßig harmonischer (Beispiel Sequenz)



- ▶ Zuerst manuelle Testreihen:
- ▶ Funktionen nur 'play two' und 'add space', keine Akkorde:
Schlechte, unstrukturierte Ergebnisse (Beispiel Sequenz)
- ▶ Alle Funktionen, keine Akkorde
Besser, gleichmäßig harmonischer (Beispiel Sequenz)
- ▶ Alle Funktionen, mit Akkorden, Limitierung auf C-Dur:
Erheblich interessantere, wohlklingendere Sequenzen (Beispiel Sequenz)



- ▶ Weitere, lange manuelle Testreihe mit 200 Bewertungen
- ▶ 100 davon als Trainings-Set für das neuronale Netz
- ▶ Die anderen 100 als Kontroll-Set



- ▶ Weitere, lange manuelle Testreihe mit 200 Bewertungen
- ▶ 100 davon als Trainings-Set für das neuronale Netz
- ▶ Die anderen 100 als Kontroll-Set
- ▶ Mittlerer Fehler des neuralen Netzwerks am Kontroll-Set nach Optimierung:
7.16
- ▶ Bei Bewertungs-Intervall {1 - 100} wurde mittlerer Fehler von ca. 7 als Erfolg betrachtet



- ▶ Weitere, lange manuelle Testreihe mit 200 Bewertungen
- ▶ 100 davon als Trainings-Set für das neuronale Netz
- ▶ Die anderen 100 als Kontroll-Set
- ▶ Mittlerer Fehler des neuralen Netzwerks am Kontroll-Set nach Optimierung: 7.16
- ▶ Bei Bewertungs-Intervall $\{1 - 100\}$ wurde mittlerer Fehler von ca. 7 als Erfolg betrachtet
- ▶ Weitere Testreihen mit Hilfe dieses automatischen Bewerterers
- ▶ Variation bei Population, Generationenanzahl



- ▶ Weitere, lange manuelle Testreihe mit 200 Bewertungen
- ▶ 100 davon als Trainings-Set für das neuronale Netz
- ▶ Die anderen 100 als Kontroll-Set
- ▶ Mittlerer Fehler des neuralen Netzwerks am Kontroll-Set nach Optimierung: 7.16
- ▶ Bei Bewertungs-Intervall {1 - 100} wurde mittlerer Fehler von ca. 7 als Erfolg betrachtet
- ▶ Weitere Testreihen mit Hilfe dieses automatischen Bewerterers
- ▶ Variation bei Population, Generationenanzahl
- ▶ Ergebnisse teils gut, teils schlecht, merklich schlechter als bei manueller Bewertung (Beispiel Sequenz)
- ▶ Nicht konsistent besser bei längeren Testreihen



- ▶ Paper wurde 1998 veröffentlicht
- ▶ Automatisch generierte Musik wurde bereits mehrfach versucht
- ▶ Automatische Bewertung durch neurales Netz wurde bereits versucht
- ▶ Dieses Paper stellt zweiten Ansatz mit Genetischer Programmierung vor
- ▶ Erstmals kombiniert mit neuralem Netz
- ▶ Denkanstoß für weitere Forschungen - Ergebnisse aber ohne Signifikanz

- ▶ Paper wurde und wird in vielen Arbeiten zitiert - 11 Zitate in Werken von 2012 (Google Scholar)
- ▶ Automatisch generierte Musik auch heute nicht "gelöst"
- ▶ Verschiedene Forschungsansätze:
- ▶ Bessere manuelle Bewertungen
- ▶ Bessere automatische Bewertungen
- ▶ Weniger Algorithmisch - mehr Musik

- ▶ Ansatz Genetische Programmierung mit interaktiver Evolution oft wiederholt
- ▶ z.B. Maximus A. Kaliakatsos–Papakostas 'Interactive Evolution of 8-bit melodies with Genetic Programming towards finding aesthetic measures for sound' (2012)
- ▶ Ausblick: Manuelle Bewertung in größeren Experimenten über Web-Plattform untersuchen



- ▶ Für automatische Bewertung: Ansatz neurales Netzwerk wird selten versucht
- ▶ Für lange Sequenzen / komplette Lieder: Zu viele Neuronen, zu viele Parameter, zu geringe Chancen, Fitness-Funktion zu finden
- ▶ Meist werden andere Wege versucht, um eine Fitness-Funktion zu vermeiden oder fest zu bestimmen
- ▶ z.B. Master-Arbeit von 2011: Johannes Høydahl Jensen 'Evolutionary Music Composition (A Quantitative Approach)'
- ▶ Häufigkeitesverteilungen (Zipfsches Gesetz) in bekannten Werken analysiert, damit Fitness-Funktion bestimmt

- ▶ AI Biles GenJam
- ▶ Jazz-Soli: Genetischer Algorithmus mit manueller Bewertung
- ▶ Jazz-Wechselspiel: Notenerkennung mit Mutation
- ▶ Entwicklung des System 1993 bis... heute!
- ▶ Mittlerweile wengier Fokus auf den Algorithmus - mehr Fokus auf die Musik
- ▶ (Link zu YouTube-Video)

...und fertig!



TECHNISCHE
UNIVERSITÄT
DARMSTADT

**Vielen Dank für die
Aufmerksamkeit!**

Fragen?